

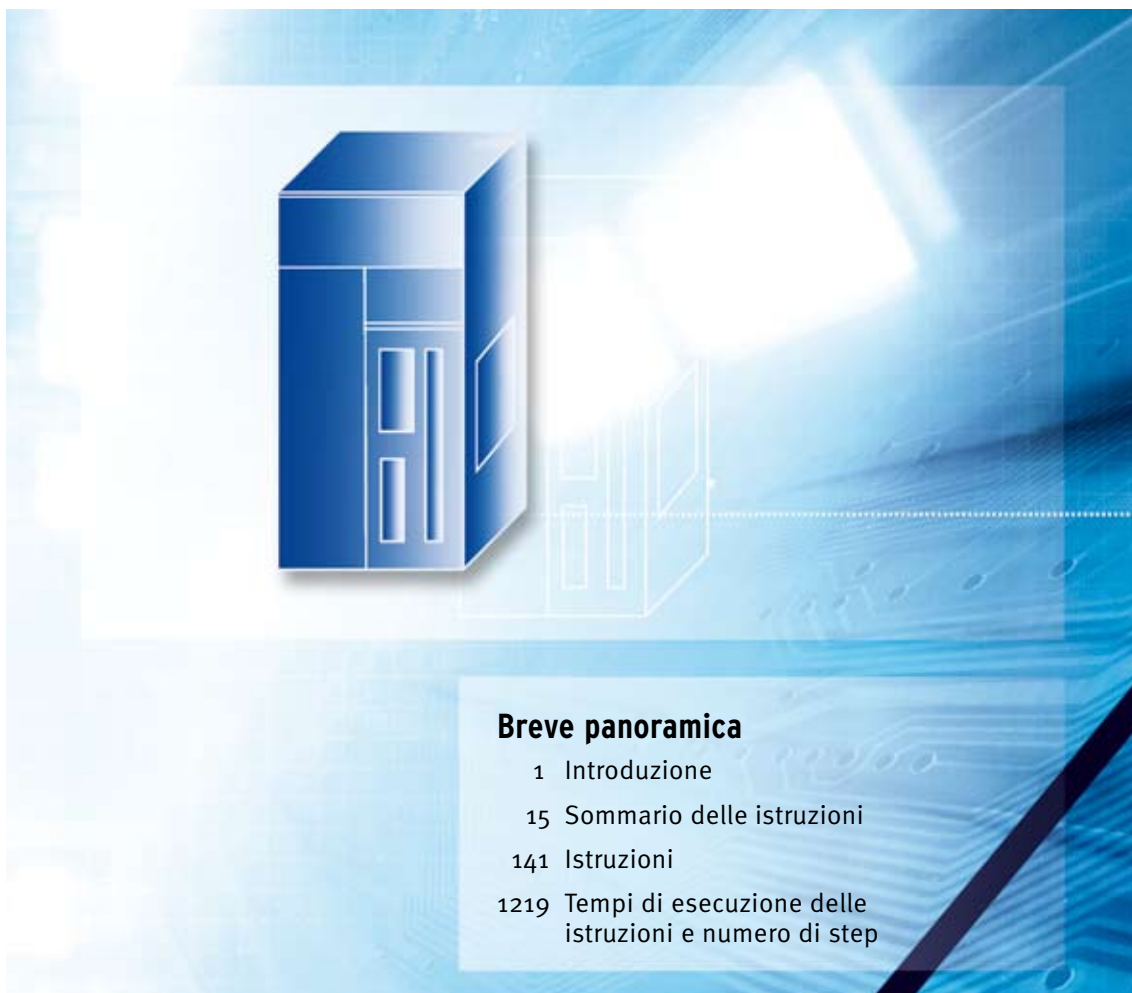


Controllori programmabili

Serie SYSMAC CS

Serie SYSMAC CJ

MANUALE DI RIFERIMENTO PER LE ISTRUZIONI



Breve panoramica

- 1 Introduzione
- 15 Sommario delle istruzioni
- 141 Istruzioni
- 1219 Tempi di esecuzione delle istruzioni e numero di step

Serie SYSMAC CS

CS1G/H-CPU□□-EV1

CS1G/H-CPU□□H

CS1D-CPU□□H

CS1D-CPU□□S

Serie SYSMAC CJ

CJ1G-CPU□□

CJ1G/H-CPU□□H

CJ1G-CPU□□P

CJ1M-CPU□□

Controllori programmabili

Manuale di riferimento per le istruzioni


Revisione: luglio 2004


Avviso

I prodotti OMRON sono destinati all'uso in accordo con le procedure appropriate da parte di un operatore qualificato e solo per gli scopi descritti in questo manuale.

In questo manuale vengono utilizzate le seguenti convenzioni per indicare e classificare le precauzioni. Attenersi sempre alle istruzioni fornite. La mancata osservanza di tali precauzioni potrebbe causare lesioni a persone o danni a proprietà.

 **PERICOLO** Indica una situazione di immediato pericolo che, se non evitata, sarà causa di lesioni gravi o mortali.

 **AVVERTENZA** Indica una situazione di potenziale pericolo che, se non evitata, può essere causa di lesioni gravi o mortali.

 **Attenzione** Indica una situazione di potenziale pericolo che, se non evitata, può essere causa di lesioni non gravi a persone o danni a proprietà.

Riferimenti ai prodotti OMRON

L'abbreviazione "Ch", che compare su alcuni display e prodotti OMRON, spesso corrisponde a "canale", termine che viene in alcuni casi abbreviato come "Cnl" nella documentazione.

L'abbreviazione "PLC" indica un controllore programmabile. È tuttavia possibile che in alcuni dispositivi di programmazione venga visualizzata l'abbreviazione "PC" ad indicare il controllore programmabile.

Supporti visivi

Nella colonna sinistra del manuale sono riportate le seguenti intestazioni per facilitare l'individuazione dei diversi tipi di informazioni.

Nota Indica informazioni di particolare rilevanza per un efficiente e vantaggioso utilizzo del prodotto.

1,2,3... 1. Indica un qualche tipo di elenco, quali procedure, liste di controllo, ecc.

© OMRON, 1999

Tutti i diritti riservati. Nessuna parte della presente pubblicazione può essere riprodotta, memorizzata in un sistema, trasmessa in qualsivoglia formato o mezzo, meccanico, elettronico, tramite fotocopia, registrazione o altro, senza previo consenso scritto di OMRON.

Non viene assunta alcuna responsabilità palese in relazione all'uso delle informazioni contenute nel presente manuale. Inoltre, poiché OMRON è alla costante ricerca della migliore qualità per i propri prodotti, le informazioni contenute nel presente manuale sono soggette a modifiche senza preavviso. Il presente manuale è stato redatto con la massima attenzione e tuttavia OMRON non assume alcuna responsabilità in relazione ad eventuali errori od omissioni, né assume alcuna responsabilità in relazione ad eventuali danni derivanti dalle informazioni in esso contenute.

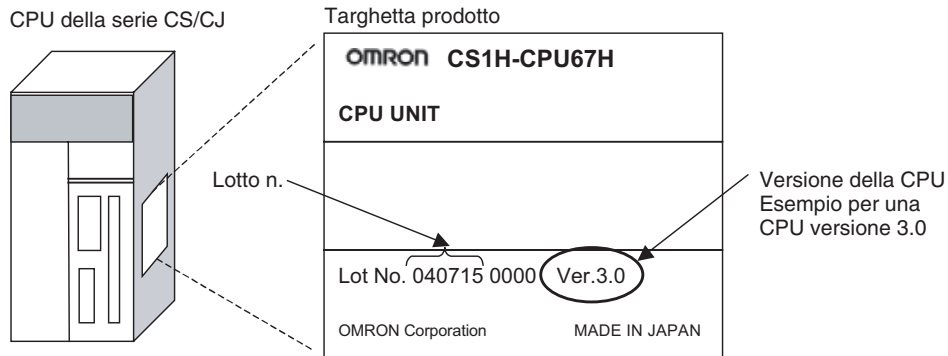
Versioni delle CPU della serie CS/CJ

Versioni delle CPU

Notazione delle versioni delle CPU sui prodotti

Per gestire le CPU della serie CS/CJ in base alle differenze di funzionalità presenti negli aggiornamenti, è stato introdotto il concetto di "versione della CPU". Questo concetto si applica alle CPU CS1-H, CJ1-H, CJ1M e CS1D.

La versione della CPU è riportata a destra del numero di lotto, sulla targhetta dei prodotti per i quali vengono gestite le versioni, come illustrato di seguito.



- Le CPU CS1-H, CJ1-H e CJ1M (eccetto i modelli di base) prodotti entro il 4 novembre del 2003 non riportano alcun numero di versione, ovvero la posizione della versione della CPU sopra illustrata risulta vuota.
- La versione delle CPU CS1-H, CJ1-H e CJ1M nonché delle CPU CS1D per sistemi a singola CPU inizia dalla versione 2.0.
- La versione delle CPU CS1D per sistemi a due CPU inizia dalla versione 1.1.
- Le CPU per le quali non viene fornita una versione sono definite *CPU precedenti alla versione* □. CPU □, ad esempio CPU precedenti alla versione 2.0 e CPU precedenti alla versione 1.1.

Verifica delle versioni delle CPU tramite software di supporto

È possibile verificare la versione della CPU tramite CX-Programmer versione 4.0 utilizzando uno dei due metodi riportati di seguito.

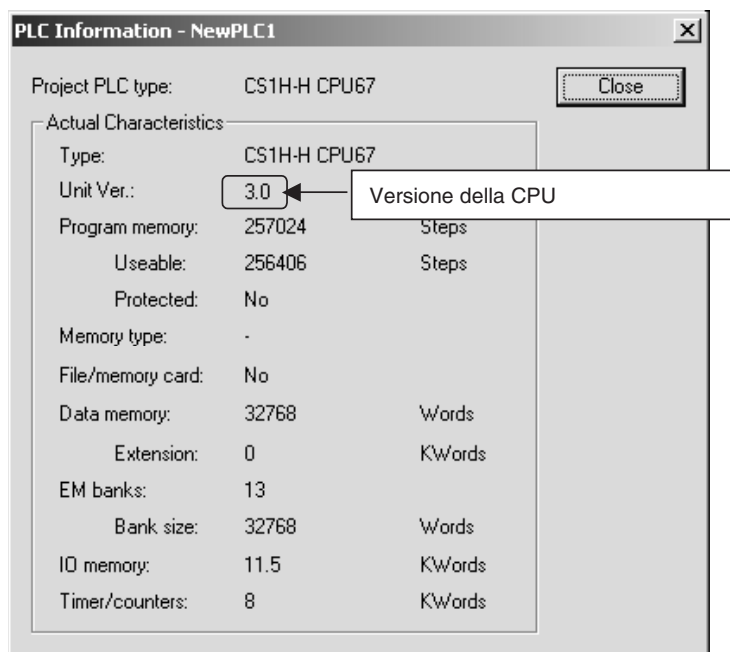
- Utilizzo della finestra **PLC Information** (*Informazioni sul PLC*)
- Utilizzo della finestra **Informazioni su produzione unità**. Questo metodo può essere utilizzato anche per i moduli di I/O speciali e le Unità bus CPU.

Nota Non è possibile verificare le versioni utilizzando CX-Programmer versione 3.3 o precedenti.

Informazioni sul PLC

- Se si conoscono il tipo di dispositivo e il tipo di CPU, selezionarli nella finestra di dialogo **Change PLC** (Cambia PLC), connettersi in linea e selezionare **PLC - Edit - Information** (PLC - Modifica - Informazioni) dai menu.
- Se non si conoscono il tipo di dispositivo e il tipo di CPU, ma si è connessi direttamente alla CPU tramite una linea seriale, selezionare **PLC - Auto Online** (PLC - Connessione in linea automatica) per stabilire una connessione in linea, quindi selezionare **PLC - Edit - Information** (PLC - Modifica - Informazioni).

In entrambi i casi, verrà visualizzata la finestra di dialogo **PLC Information** (Informazioni sul PLC) riportata di seguito.



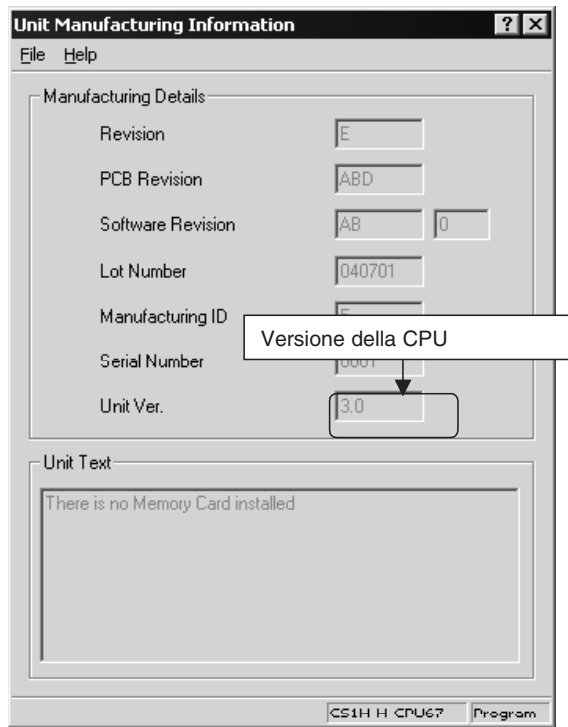
Utilizzare la finestra sopra illustrata per verificare la versione della CPU.

Informazioni su produzione unità

Nella finestra I/O Table (Tabella di I/O) fare clic con il pulsante destro del mouse e selezionare **Informazioni su produzione unità - Modulo CPU**.



Viene visualizzata la finestra di dialogo *Informazioni su produzione unità* mostrata di seguito.



Utilizzare la finestra sopra illustrata per verificare la versione della CPU connessa in linea.

Utilizzo delle etichette delle versioni delle CPU

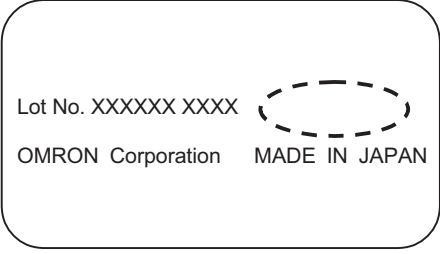
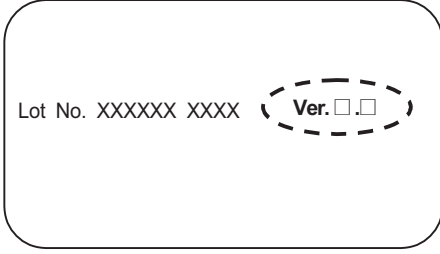
Con la CPU vengono fornite le etichette delle versioni riportate di seguito.



È possibile apporre queste etichette sulla parte frontale delle CPU precedenti per distinguerle da quelle di versioni differenti.

Notazione delle versioni delle CPU

Nel presente manuale la versione di una CPU viene fornita come illustrato nella seguente tabella.

| Targhetta prodotto Significato | CPU su cui non viene fornita alcuna versione  | CPU per le quali viene fornita una versione (Ver. □.□)  |
|---|--|---|
| Designazione di singole CPU (ad esempio CS1H-CPU67H) | CPU CS1-H precedenti alla versione 2.0 | CPU CS1H-CPU67H versione □.□ |
| Designazione dei gruppi delle CPU (ad esempio le CPU CS1-H) | CPU CS1-H precedenti alla versione 2.0 | CPU CS1-H versione □.□ |
| Designazione di un'intera serie di CPU (ad esempio le CPU della serie CS) | CPU della serie CS precedenti alla versione 2.0 | CPU della serie CS versione □.□ |

Versioni di CPU e numeri di lotto

| Serie | Modello | | Dati di produzione | | | | | | |
|----------------------------------|-------------------------------|----------------------------------|----------------------------------|----------------|---|---|--|--|--------------|
| | | | Precedente | Settembre 2003 | Ottobre 2003 | Novembre 2003 | Dicembre 2003 | Giugno 2004 | Successiva |
| Serie CS | CPU CS1 | CS1□-CPU□□ | Nessuna versione della CPU | | | | | | |
| | | | Nessuna versione della CPU | | | | | | |
| | CPU CS1-V1 | CS1□-CPU□□-V1 | Nessuna versione della CPU | | | | | | |
| | | | Nessuna versione della CPU | | | | | | |
| CPU CS1-H | CS1□-CPU□□H | CPU precedenti alla versione 2.0 | | | CPU versione 2.0 (N. lotto da 031105 in poi) | | CPU s versione 3.0 (N. lotto da 040622 in poi) | | |
| | | CPU precedenti alla versione 2.0 | | | CPU versione 2.0 (N. lotto da 031105 in poi) | | CPU s versione 3.0 (N. lotto da 040622 in poi) | | |
| CPU CS1D | CPU per sistemi a due CPU | CS1D-CPU□□H | CPU precedenti alla versione 1.1 | | | CPU versione 1.1 (N. lotto da 031120 in poi) | | | |
| | CPU per sistemi a singola CPU | CS1D-CPU□□S | CPU precedenti alla versione 1.1 | | | CPU versione 2.0 (N. lotto da 031215 in poi) | | | |
| Serie CJ | CPU CJ1 | CJ1G-CPU□□ | CPU precedenti alla versione 2.0 | | | | | | |
| | | | CPU precedenti alla versione 2.0 | | | | | | |
| | CPU CJ1-H | CJ1□-CPU□□H | CPU precedenti alla versione 2.0 | | | CPU versione 2.0 (N. lotto da 031105 in poi) | | CPU versione 3.0 (N. lotto da 040623 in poi) | |
| | | | CPU precedenti alla versione 2.0 | | | CPU versione 2.0 (N. lotto da 031105 in poi) | | CPU versione 3.0 (N. lotto da 040623 in poi) | |
| CPU CJ1M eccetto modelli di base | CJ1M-CPU□□ | CPU precedenti alla versione 2.0 | | | CPU versione 2.0 (N. lotto da 031105 in poi) | | CPU versione 3.0 (N. lotto da 040624 in poi) | | |
| | | CPU precedenti alla versione 2.0 | | | CPU versione 2.0 (N. lotto da 031105 in poi) | | CPU versione 3.0 (N. lotto da 040624 in poi) | | |
| CPU CJ1M, modelli di base | CJ1M-CPU11/21 | CPU precedenti alla versione 2.0 | | | CPU versione 2.0 (N. lotto da 031002 in poi) | | CPU versione 3.0 (N. lotto da 040629 in poi) | | |
| Software di supporto | CX-Programmer | WS02-CXPC1-EV□ | Versione 3.2 | | Versione 3.3 | | Versione 4.0 | | Versione 5.0 |
| | | | Versione 3.2 | | Versione 3.3 | | Versione 4.0 | | Versione 5.0 |

Funzioni supportate dalla versione della CPU

CPU CS1-H (CS1□-CPU□□H)

| Funzioni | | Versione della CPU | |
|---|---|---|--------------------------------|
| | | CPU precedenti alla versione 2.0 | CPU versione 2.0 |
| Scaricamento e caricamento di singoli task | | --- | OK |
| Protezione da lettura migliorata mediante password | | --- | OK |
| Protezione da scrittura dai comandi FINS inviati alle CPU tramite reti | | --- | OK |
| Connessioni di rete in linea senza tabelle degli I/O | | --- | OK |
| Comunicazioni tramite un massimo di 8 livelli di rete | | --- | OK |
| Connessione in linea ai PLC tramite PT della serie NS | | OK a partire dal numero di lotto 030201 | OK |
| Impostazione dei canali del primo slot | | OK per un massimo di 8 gruppi | OK per un massimo di 64 gruppi |
| Trasferimento automatico all'accensione del sistema senza un file dei parametri | | --- | OK |
| Rilevamento automatico del metodo di allocazione degli I/O per il trasferimento automatico all'accensione del sistema | | --- | --- |
| Ore di inizio e fine del funzionamento | | --- | OK |
| Nuove istruzioni di applicazione | MILH, MILR, MILC | --- | OK |
| | =DT, <>DT, <DT, <=DT, >DT, >=DT | --- | OK |
| | BCMP2 | --- | OK |
| | GRY | OK a partire dal numero di lotto 030201 | OK |
| | TPO | --- | OK |
| | DSW, TKY, HKY, MTR, 7SEG | --- | OK |
| | EXPLT, EGATR, ESATR, ECHRD, ECHWR | --- | OK |
| | Lettura/Scrittura da/su Unità bus CPU bus tramite IORD/IOWR | OK a partire dal numero di lotto 030418 | OK |
| | PRV2 | --- | --- |

CPU CS1D

| Funzioni | | CPU CS1D per sistemi a due CPU (CS1D-CPU□□H) | | CPU CS1D per sistemi a singola CPU (CS1D-CPU□□S) |
|---|--|---|------------------|---|
| | | CPU precedenti alla versione 1.1 | CPU versione 1.1 | CPU versione 2.0 |
| Funzione esclusiva delle CPU CS1D | CPU duplex | OK | OK | --- |
| | Sostituzione in linea | OK | OK | OK |
| | Moduli di alimentazione duplex | OK | OK | OK |
| | Due moduli Controller Link | OK | OK | OK |
| | Due moduli Ethernet | --- | OK | OK |
| Scaricamento e caricamento di singoli task | | --- | --- | OK |
| Protezione da lettura migliorata mediante password | | --- | --- | OK |
| Protezione da scrittura dai comandi FINS inviati alle CPU tramite reti | | --- | --- | OK |
| Connessioni di rete in linea senza tabelle degli I/O | | --- | --- | OK |
| Comunicazioni tramite un massimo di 8 livelli di rete | | --- | --- | OK |
| Connessione in linea ai PLC tramite PT della serie NS | | --- | --- | OK |
| Impostazione dei canali del primo slot | | --- | --- | OK per un massimo di 64 gruppi |
| Trasferimento automatico all'accensione del sistema senza un file dei parametri | | --- | --- | OK |
| Rilevamento automatico del metodo di allocazione degli I/O per il trasferimento automatico all'accensione del sistema | | --- | --- | --- |
| Ore di inizio e fine del funzionamento | | --- | OK | OK |
| Nuove istruzioni di applicazione | MILH, MILR, MILC | --- | --- | OK |
| | =DT, <>DT, <DT, <=DT, >DT, >=DT | --- | --- | OK |
| | BCMP2 | --- | --- | OK |
| | GRY | --- | --- | OK |
| | TPO | --- | --- | OK |
| | DSW, TKY, HKY, MTR, 7SEG | --- | --- | OK |
| | EXPLT, EGATR, ESATR, ECHRD, ECHWR | --- | --- | OK |
| | Lettura/Scrittura da/su moduli CPU bus tramite IORD/IOWR | --- | --- | OK |
| PRV2 | --- | --- | --- | |

CPU CJ1-H/CJ1M

| Funzioni | CPU CJ1-H (CJ1□-CPU□□H) | | CPU CJ1M, eccetto i modelli di base (CJ1M-CPU□□) | | CPU CJ1M, modelli di base (CJ1M- CPU11/21) |
|---|---|---|---|---|---|
| | CPU precedenti alla versione 2.0 | CPU versione 2.0 | CPU precedenti alla versione 2.0 | CPU versione 2.0 | CPU versione 2.0 |
| Scaricamento e caricamento di singoli task | --- | OK | --- | OK | OK |
| Protezione da lettura migliorata mediante password | --- | OK | --- | OK | OK |
| Protezione da scrittura dai comandi FINS inviati alle CPU tramite reti | --- | OK | --- | OK | OK |
| Connessioni di rete in linea senza tabelle degli I/O | OK, ma solo se all'accensione del sistema è impostata l'allocazione delle tabelle degli I/O | OK | OK, ma solo se all'accensione del sistema è impostata l'allocazione delle tabelle degli I/O | OK | OK |
| Comunicazioni tramite un massimo di 8 livelli di rete | OK per un massimo di 8 gruppi | OK per un massimo di 64 gruppi | OK per un massimo di 8 gruppi | OK per un massimo di 64 gruppi | OK per un massimo di 64 gruppi |
| Connessione in linea ai PLC tramite PT della serie NS | OK a partire dal numero di lotto 030201 | OK | OK a partire dal numero di lotto 030201 | OK | OK |
| Impostazione dei canali del primo slot | --- | OK | --- | OK | OK |
| Trasferimento automatico all'accensione del sistema senza un file dei parametri | --- | OK | --- | OK | OK |
| Rilevamento automatico del metodo di allocazione degli I/O per il trasferimento automatico all'accensione del sistema | --- | OK | --- | OK | OK |
| Ore di inizio e fine del funzionamento | --- | OK | --- | OK | OK |
| Nuove istruzioni di applicazione | MILH, MILR, MILC | --- | OK | --- | OK |
| | =DT, <>DT, <DT, <=>DT, >DT, >=DT | --- | OK | --- | OK |
| | BCMP2 | --- | OK | OK | OK |
| | GRY | OK a partire dal numero di lotto 030201 | OK | OK a partire dal numero di lotto 030201 | OK |
| | TPO | --- | OK | --- | OK |
| | DSW, TKY, HKY, MTR, 7SEG | --- | OK | --- | OK |
| | EXPLT, EGATR, ESATR, ECHRD, ECHWR | --- | OK | --- | OK |
| | Lettura/Scrittura da/su Unità bus CPU bus tramite IORD/IOWR | --- | OK | --- | OK |
| PRV2 | --- | --- | --- | OK, ma solo per i modelli con I/O integrati | OK, ma solo per i modelli con I/O integrati |

Funzioni supportate dalle CPU versione 3.0 o successiva

CPU CS1-H (CS1□-CPU□□H)

| Funzioni | | Versione della CPU | |
|---|---|--|--------------|
| | | CPU precedenti alla versione 2.0, versione 2.0 | Versione 3.0 |
| Blocchi funzione (supportati da CX-Programmer versione 5.0 o successiva) | | --- | OK |
| Gateway seriale (conversione dei comandi FINS in comandi CompoWay/F a livello di porta seriale integrata) | | --- | OK |
| Memoria dei commenti (nella memoria flash interna) | | --- | OK |
| Dati di backup semplice espansi | | --- | OK |
| Nuove istruzioni di applicazione | TXDU(256), RXDU(255) (supporto per la comunicazione senza protocollo tra moduli di comunicazione seriale e CPU della versione 1.2 o successiva) | --- | OK |
| | Istruzioni di conversione del modello: XFERC(565), DISTC(566), COLLC(567), MOVBC(568), BCNTC(621) | --- | OK |
| | Istruzioni speciali per blocchi funzione: GETID(286) | --- | OK |
| Funzioni di istruzione aggiuntive | Istruzioni TXD(235) e RXD(236) (supporto per la comunicazione senza protocollo tra schede di comunicazione seriale e CPU della versione 1.2 o successiva) | --- | OK |

CPU CS1D

Le CPU versione 3.0 non sono supportate.

CPU CJ1-H/CJ1M (CJ1□-CPU□□H, CJ1G-CPU□□P, CJ1M-CPU□□)

| Funzioni | | Versione della CPU | |
|---|---|--|--------------|
| | | CPU precedenti alla versione 2.0, versione 2.0 | Versione 3.0 |
| Blocchi funzione (supportati da CX-Programmer versione 5.0 o successiva) | | --- | OK |
| Gateway seriale (conversione dei comandi FINS in comandi CompoWay/F a livello di porta seriale integrata) | | --- | OK |
| Memoria dei commenti (nella memoria flash interna) | | --- | OK |
| Dati di backup semplice espansi | | --- | OK |
| Nuove istruzioni di applicazione | TXDU(256), RXDU(255) (supporto per la comunicazione senza protocollo tra moduli di comunicazione seriale e CPU della versione 1.2 o successiva) | --- | OK |
| | Istruzioni di conversione del modello: XFERC(565), DISTC(566), COLLC(567), MOVBC(568), BCNTC(621) | --- | OK |
| | Istruzioni speciali per blocchi funzione: GETID(286) | --- | OK |
| Funzioni di istruzione aggiuntive | Istruzioni PRV(881) e PRV2(883): aggiunti metodi ad alta frequenza per il calcolo della frequenza di impulsi (solo CPU CJ1M). | --- | OK |

Versioni delle CPU e dispositivi di programmazione

Per consentire l'utilizzo delle funzioni aggiunte alla CPU versione 2.0, è necessario utilizzare CX-Programmer versione 4.0 o successiva.

Per consentire l'utilizzo delle funzioni aggiunte ai blocchi funzione per le CPU versione 3.0, è necessario utilizzare CX-Programmer versione 5.0 o successiva.

Nelle seguenti tabelle è illustrata la relazione tra le versioni delle CPU e le versioni di CX-Programmer.

Versioni delle CPU e dispositivi di programmazione

| CPU | Funzioni | | CX-Programmer | | | | Console di programmazione |
|--|---|-------------------------------------|---------------------------|--------------|--------------|---------------------------|---------------------------|
| | | | Versione 3.2 o precedente | Versione 3.3 | Versione 4.0 | Versione 5.0 o successiva | |
| CPU CJ1M, modelli di base, versione della CPU 2.0 | Funzioni aggiuntive della versione 2.0 | Con utilizzo delle nuove funzioni | --- | --- | OK | OK | Nessuna restrizione |
| | | Senza utilizzo delle nuove funzioni | --- | OK | OK | OK | |
| CPU CS1-H, CJ1-H e CJ1M, eccetto modelli di base, versione della CPU 2.0 | Funzioni aggiuntive della versione 2.0 | Con utilizzo delle nuove funzioni | --- | --- | OK | OK | |
| | | Senza utilizzo delle nuove funzioni | OK | OK | OK | OK | |
| CPU CS1D per sistemi a singola CPU, versione della CPU 2.0 | Funzioni aggiuntive della versione 2.0 | Con utilizzo delle nuove funzioni | --- | --- | OK | OK | |
| | | Senza utilizzo delle nuove funzioni | | | | OK | |
| CPU CS1D per sistemi a due CPU, versione della CPU 1. | Funzioni aggiuntive della versione 1.1 | Con utilizzo delle nuove funzioni | --- | --- | OK | OK | |
| | | Senza utilizzo delle nuove funzioni | OK | OK | OK | OK | |
| CPU della serie CS/CJ versione 3.0 | Funzioni aggiuntive per blocchi funzione della versione 3.0 | Con utilizzo dei blocchi funzione | --- | --- | --- | OK | |
| | | Senza utilizzo dei blocchi funzione | OK | OK | OK | OK | |

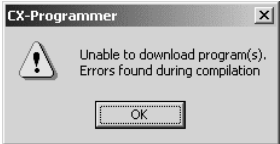

Nota Come illustrato sopra, se non si utilizzano le funzioni aggiunte alla versione 2.0 o alla versione 1.1, non è necessario eseguire l'aggiornamento a CX-Programmer versione 4.0.

Impostazione del tipo di dispositivo

La versione della CPU non influisce sull'impostazione effettuata per il tipo di dispositivo in CX-Programmer. Selezionare il tipo di dispositivo come illustrato nella seguente tabella, indipendentemente dalla versione della CPU.

| Serie | Gruppo CPU | Modello di CPU | Impostazione del tipo di dispositivo in CX-Programmer versione 4.0 o successiva |
|----------|------------------------------------|----------------|---|
| Serie CS | CPU CS1-H | CS1G-CPU□□H | CS1G-H |
| | | CS1H-CPU□□H | CS1H-H |
| | CPU CS1D per sistemi a due CPU | CS1D-CPU□□H | CS1D-H (o CS1H-H) |
| | CPU CS1D per sistemi a singola CPU | CS1D-CPU□□S | CS1D-S |
| Serie CJ | CPU CJ1-H | CJ1G-CPU□□H | CJ1G-H |
| | | CJ1H-CPU□□H | CJ1H-H |
| | CPU CJ1M | CJ1M-CPU□□ | CJ1M |

Risoluzione dei problemi relativi alle versioni delle CPU in CX-Programmer

| Problema | Causa | Soluzione |
|--|---|---|
|  <p>Dopo la visualizzazione del messaggio sopra riportato, verrà visualizzato un errore di compilazione nella scheda <i>Compile</i> (Compila) della finestra di output.</p> | <p>Utilizzando CX-Programmer versione 4.0 o successiva, si è tentato di scaricare un programma che contiene istruzioni supportate solo dalle CPU versione 2.0 o successiva in una CPU precedente alla versione 2.0.</p> | <p>Controllare il programma o cambiare la CPU in cui si desidera scaricare il programma con una CPU versione 2.0 o successiva.</p> |
|  | <p>Utilizzando CX-Programmer versione 4.0 o successiva, si è tentato di scaricare impostazioni del PLC che contengono impostazioni supportate solo dalle CPU versione 2.0 o successiva, ovvero non impostate sui valori predefiniti, in una CPU precedente alla versione 2.0.</p> | <p>Controllare le impostazioni delle impostazioni del PLC o cambiare la CPU in cui si desidera scaricare le impostazioni con una CPU versione 2.0 o successiva.</p> |
| <p>“????” viene visualizzato in un programma trasferito dal PLC a CX-Programmer.</p> | <p>Si è utilizzato CX-Programmer versione 3.3 o precedente per caricare un programma che contiene istruzioni supportate solo dalle CPU versione 2.0 o successiva da una CPU versione 2.0 o successiva.</p> | <p>Non è possibile caricare le nuove istruzioni utilizzando CX-Programmer versione 3.3 o precedente. Utilizzare CX-Programmer versione 4.0 o successiva.</p> |

SOMMARIO

| | |
|--|--------------|
| PRECAUZIONI | xxiii |
| 1 Destinatari del manuale..... | xxiv |
| 2 Precauzioni generali | xxiv |
| 3 Precauzioni per la sicurezza | xxiv |
| 4 Precauzioni relative all'ambiente operativo..... | xxvi |
| 5 Precauzioni per le applicazioni | xxvi |
| 6 Conformità alle direttive dell'Unione Europea | xxx |
| | |
| CAPITOLO 1 | |
| Introduzione | 1 |
| 1-1 Caratteristiche generali delle istruzioni..... | 2 |
| 1-2 Controlli dell'esecuzione di istruzioni..... | 13 |
| | |
| CAPITOLO 2 | |
| Sommario delle istruzioni | 15 |
| 2-1 Classificazioni in base alla funzione..... | 16 |
| 2-2 Istruzioni | 24 |
| 2-3 Elenco alfabetico delle istruzioni in base al codice mnemonico..... | 108 |
| 2-4 Elenco delle istruzioni in base al codice funzione | 125 |
| | |
| CAPITOLO 3 | |
| Istruzioni | 141 |
| 3-1 Descrizioni delle istruzioni: notazione e layout | 148 |
| 3-2 Aggiornamenti delle istruzioni e nuove istruzioni..... | 151 |
| 3-3 Istruzioni di ingresso sequenza | 153 |
| 3-4 Istruzioni di uscita sequenza | 177 |
| 3-5 Istruzioni di controllo sequenza | 197 |
| 3-6 Istruzioni di temporizzatore e contatore | 233 |
| 3-7 Istruzioni di confronto..... | 275 |
| 3-8 Istruzioni di spostamento dati | 315 |
| 3-9 Istruzioni di scorrimento dei dati | 344 |
| 3-10 Istruzioni di incremento e decremento | 393 |
| 3-11 Istruzioni matematiche con simboli | 409 |
| 3-12 Istruzioni di conversione..... | 465 |
| 3-13 Istruzioni logiche..... | 517 |
| 3-14 Istruzioni matematiche speciali..... | 534 |
| 3-15 Istruzioni matematiche a virgola mobile..... | 558 |
| 3-16 Istruzioni a virgola mobile in doppia precisione (solo CPU CS1-H, CJ1-H, CJ1M o CS1D) | 613 |
| 3-17 Istruzioni di elaborazione dei dati delle tabelle | 660 |
| 3-18 Istruzioni di controllo dei dati..... | 720 |
| 3-19 Subroutine..... | 773 |
| 3-20 Istruzioni di controllo degli interrupt | 798 |

SOMMARIO

| | |
|---|------|
| 3-21 Istruzioni per contatore veloce e uscita a impulsi | 823 |
| 3-22 Istruzioni di step | 867 |
| 3-23 Istruzioni per moduli di I/O di base | 885 |
| 3-24 Istruzioni per la comunicazione seriale | 926 |
| 3-25 Istruzioni di rete | 973 |
| 3-26 Istruzioni per la memoria dei file | 1042 |
| 3-27 Istruzioni di visualizzazione: DISPLAY MESSAGE: MSG(046) | 1058 |
| 3-28 Istruzioni per l'orologio | 1061 |
| 3-29 Istruzioni di debug | 1075 |
| 3-30 Istruzioni di diagnostica per malfunzionamento | 1079 |
| 3-31 Altre istruzioni | 1104 |
| 3-32 Istruzioni di programmazione a blocchi | 1124 |
| 3-33 Istruzioni di elaborazione delle stringhe di testo | 1158 |
| 3-34 Istruzioni di controllo dei task | 1192 |
| 3-35 Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva) | 1199 |

CAPITOLO 4

Tempi di esecuzione delle istruzioni e numero di step 1219

| | |
|---|------|
| 4-1 Tempi di esecuzione delle istruzioni e numero di step per la serie CS | 1221 |
| 4-2 Tempi di esecuzione delle istruzioni e numero di step per la serie CJ | 1250 |

Appendices

| | |
|----------------------------------|------|
| A Tabella dei codici ASCII | 1285 |
|----------------------------------|------|

Index..... 1287

Storico delle revisioni 1295

Informazioni sul manuale

Nel presente manuale viene fornita una descrizione delle istruzioni di programmazione per i diagrammi ladder dei moduli CPU supportate dai controllori programmabili (PLC) della serie CS/CJ. I prodotti della serie CS e della serie CJ sono suddivisi come illustrato nella seguente tabella.

| Modulo | Serie CS | Serie CJ |
|-------------------------|--|--|
| CPU | CPU CS1-H: CS1H-CPU□□H CS1G-CPU□□H | CPU CJ1-H: CJ1H-CPU□□H CJ1G-CPU□□H CJ1G-CPU□□P |
| | CPU CS1: CS1H-CPU□□-EV1 CS1G-CPU□□-EV1 | CPU CJ1: CJ1G-CPU□□-EV1 CPU CJ1M: CJ1M-CPU□□ |
| | CPU CS1D: CPU CS1D per sistemi a due CPU: CS1D-CPU□□H CPU CS1D per sistemi a singola CPU: CS1D-CPU□□S CPU con processo CS1D: CS1D-CPU□□P | |
| Moduli di I/O di base | Moduli di I/O di base della serie CS | Moduli di I/O di base della serie CJ |
| Moduli di I/O speciali | Moduli di I/O speciali della serie CS | Moduli di I/O speciali della serie CJ |
| Unità bus CPU | Unità bus CPU della serie CS | Unità bus CPU della serie CJ |
| Moduli di alimentazione | Moduli di alimentazione della serie CS | Moduli di alimentazione della serie CJ |

Leggere attentamente il presente manuale e tutti i manuali correlati elencati nella tabella riportata nella pagina seguente e accertarsi di avere ben compreso le informazioni fornite prima di procedere alla programmazione o all'utilizzo delle CPU della serie CS/CJ in un sistema PLC.

Capitolo 1 introduce i PLC della serie CS/CJ in base alla serie di istruzioni supportate.

Capitolo 2 fornisce vari elenchi di istruzioni che è possibile utilizzare come riferimento.

Capitolo 3 descrive le singole istruzioni della serie di istruzioni per la serie CS/CJ.

Capitolo 4 fornisce i tempi di esecuzione delle istruzioni e il numero di step necessari per ciascuna istruzione della serie CS/CJ.

Informazioni sul manuale (continua)

| Nome | N. cat. | Descrizione |
|---|---------|--|
| Serie SYSMAC CS/CJ CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1G-CPU□□, CJ1M-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H Controllori programmabili - Manuale di riferimento delle istruzioni | W340 | Fornisce una descrizione delle istruzioni di programmazione utilizzate nei diagrammi ladder supportate dai PLC della serie CS/CJ (il presente manuale). |
| Serie SYSMAC CS/CJ CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1G-CPU□□, CJ1M-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H Controllori programmabili - Manuale di programmazione | W394 | Fornisce informazioni sulla programmazione e altri metodi per l'utilizzo delle funzioni dei PLC della serie CS/CJ |
| Serie SYSMAC CS CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H Manuale dell'operatore dei controllori programmabili | W339 | Fornisce informazioni generali sui PLC della serie CS, ne descrive le caratteristiche di progettazione, le procedure di installazione e manutenzione nonché altre operazioni di base per l'utilizzo dei PLC. |
| Serie SYSMAC CJ CJ1G-CPU□□, CJ1M-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H Controllori programmabili - Manuale dell'operatore | W393 | Fornisce informazioni generali sui PLC della serie CJ, ne descrive le caratteristiche di progettazione, le procedure di installazione e manutenzione nonché altre operazioni di base per l'utilizzo dei PLC |
| CPU CJ1M-CPU21/22/23 della serie CJ con I/O integrati Manuale dell'operatore | W395 | Descrive le funzioni degli I/O integrati delle CPU CJ1M |
| SYSMAC CS Series CS1D-CPU□□H CPU Units CS1D-CPU□□S CPU Units CS1D-DPL1 Duplex Unit CS1D-PA207R Power Supply Unit Duplex System Operation Manual | W405 | Fornisce informazioni generali e descrive le caratteristiche di progettazione, le procedure di installazione e manutenzione nonché altre operazioni di base per un sistema duplex basato sulla CPU CS1D. |
| SYSMAC CS/CJ Series CQM1H-PRO01-E, C200H-PRO27-E, CQM1-PRO01-E Programming Consoles Operation Manual | W341 | Fornisce informazioni sui metodi di programmazione e utilizzo dei PLC della serie CS/CJ mediante una console di programmazione. |
| SYSMAC CS/CJ Series CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1M-CPU□□, CJ1G-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H, CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21-V1/41-V1 Communications Commands Reference Manual | W342 | Descrive i comandi di comunicazione FINS e dei PLC della serie C (Host Link) utilizzati con i PLC della serie CS/CJ. |
| SYSMAC WS02-CXP□□-E CX-Programmer Operation Manual Version 3.□ | W414 | Fornisce informazioni sull'utilizzo di CX-Programmer, un dispositivo di programmazione software che supporta i PLC della serie CS/CJ, e del programma CX-Net incluso in CX-Programmer. |
| SYSMAC WS02-CXP□□-E CX-Programmer Operation Manual Version 4.□ | W425 | |
| SYSMAC WS02-CXP□□-E CX-Programmer Operation Manual Version 5.□ | W437 | |

| Nome | N. cat. | Descrizione |
|--|---------|--|
| SYSMAC WS02-CXP□□-E CX-Programmer Operation Manual Function Blocks | W438 | Descrive specifiche e metodi operativi relativi ai blocchi funzione. Queste informazioni sono rilevanti solo se si utilizzano i blocchi funzione con CX-Programmer versione 5.0 e CPU CJ1-H/ CJ1M versione 3.0. Per informazioni dettagliate su altre funzionalità di CX-Programmer versione 5.0, fare riferimento al manuale <i>CX-Programmer Operation Manual Version 5.□ (W437). 5.0.</i> |
| SYSMAC CS/CJ Series CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21-V1/41-V1 Serial Communications Boards/Units Operation Manual | W336 | Descrive l'utilizzo dei moduli e delle schede di comunicazione seriale per la comunicazione seriale con i dispositivi esterni, incluso l'impiego dei protocolli di sistema standard per i prodotti OMRON. |
| SYSMAC WS02-PSTC1-E CX-Protocol Operation Manual | W344 | Fornisce informazioni sull'utilizzo di CX-Protocol per la creazione di protocol macro, impiegate come sequenze di comunicazione nella comunicazione con dispositivi esterni. |



AVVERTENZA La mancata lettura o comprensione delle informazioni fornite in questo manuale può fare insorgere condizioni di rischio e conseguenti lesioni personali, anche mortali, danneggiamento o funzionamento non corretto del prodotto. Leggere ogni sezione per intero e accertarsi di avere compreso le informazioni in essa contenute e quelle correlate prima di eseguire una delle procedure o operazioni descritte.

PRECAUZIONI

In questo capitolo sono riportate le precauzioni generali per l'uso dei controllori programmabili (PLC) della serie CS/CJ e dei dispositivi collegati.

Le informazioni contenute in questo capitolo sono importanti per garantire un utilizzo sicuro e affidabile dei controllori programmabili. È necessario leggere il capitolo e comprenderne il contenuto prima di configurare o utilizzare un PLC.

| | | |
|-----|---|------|
| 1 | Destinatari del manuale. | xxiv |
| 2 | Precauzioni generali | xxiv |
| 3 | Precauzioni per la sicurezza | xxiv |
| 4 | Precauzioni relative all'ambiente operativo. | xxvi |
| 5 | Precauzioni per le applicazioni | xxvi |
| 6 | Conformità alle direttive dell'Unione Europea | xxx |
| 6-1 | Direttive applicabili | xxx |
| 6-2 | Principi | xxxi |
| 6-3 | Conformità alle direttive dell'Unione Europea | xxxi |
| 6-4 | Metodi di riduzione dei disturbi nelle uscite a relè. | xxxi |

1 Destinatari del manuale

Questo manuale si rivolge al personale indicato di seguito, che deve avere anche conoscenze di sistemi elettrici, come un ingegnere elettronico.

- Responsabili dell'installazione di sistemi di automazione industriale.
- Responsabili della progettazione di sistemi di automazione industriale.
- Responsabili della gestione di sistemi di automazione industriale e delle relative infrastrutture.


2 Precauzioni generali

L'utente deve utilizzare il prodotto in base alle specifiche riportate nei manuali dell'operatore.


Prima di utilizzare il prodotto in condizioni non descritte nel manuale o di applicarlo a sistemi di controllo nucleare, sistemi ferroviari, sistemi per l'aviazione, veicoli, sistemi di combustione, apparecchiature medicali, macchine da luna park, apparecchiature di sicurezza e qualunque altro sistema, macchina o apparecchiatura, il cui utilizzo errato potrebbe avere un serio impatto sull'incolumità di persone o protezione di cose, rivolgersi al proprio rappresentante OMRON.

Accertarsi che i valori nominali e le specifiche del prodotto siano sufficienti per i sistemi, le macchine e le apparecchiature che verranno utilizzati e dotare sempre tali sistemi, macchine e apparecchiature di doppi meccanismi di sicurezza.


Nel presente manuale vengono fornite informazioni sulla programmazione e sul funzionamento del modulo. Si raccomanda di leggere il manuale prima di utilizzare il modulo per la prima volta e tenerlo sempre a portata di mano come riferimento durante le operazioni.


 **AVVERTENZA** È di fondamentale importanza che il PLC e tutti i relativi moduli vengano utilizzati per lo scopo specificato e nelle condizioni specificate, in particolare in applicazioni che implicano rischi diretti o indiretti per l'incolumità delle persone. Prima di utilizzare il PLC per tali applicazioni, rivolgersi al proprio rappresentante OMRON.

3 Precauzioni per la sicurezza

 **AVVERTENZA** La CPU aggiorna gli I/O anche quando il programma non è in esecuzione, ovvero anche in modalità PROGRAM. Verificare preliminarmente lo stato della sicurezza prima di modificare lo stato di qualunque parte della memoria assegnata ai moduli di I/O, ai moduli di I/O speciali o alle unità bus CPU. Qualunque modifica ai dati della memoria allocati a un modulo può provocare l'inattesa attivazione dei carichi collegati a tale modulo. Una qualunque delle seguenti operazioni può determinare la modifica dello stato della memoria.

- Trasferimento di dati della memoria I/O da un dispositivo di programmazione alla CPU.
- Modifica dei valori attuali in memoria da un dispositivo di programmazione.
- Impostazione/reset forzato di bit da un dispositivo di programmazione.
- Trasferimento di file della memoria I/O da una scheda di memoria o dall'area di memoria per i file nell'area EM alla CPU.
- Trasferimento di dati della memoria I/O da un computer host o da un altro PLC collegato in rete.

 **AVVERTENZA** Non tentare di aprire un modulo in presenza di alimentazione, in quanto ciò implica il rischio di scosse elettriche.

 **AVVERTENZA** Non toccare i terminali o le morsettiere quando il sistema è alimentato, in quanto ciò implica il rischio di scosse elettriche.

- ⚠ AVVERTENZA** Non tentare di smontare, riparare o modificare alcun modulo. Qualsiasi intervento in tal senso potrebbe provocare malfunzionamento, incendi o scosse elettriche.
- ⚠ AVVERTENZA** Applicare adeguate misure di sicurezza ai circuiti esterni (cioè, esterni al controllore programmabile), incluse quelle riportate di seguito, per garantire la massima sicurezza del sistema in caso di anomalie dovute al malfunzionamento del PLC o ad altri fattori esterni che influiscono sul funzionamento del PLC. Disattendere queste precauzioni potrebbe essere causa di gravi incidenti.
- I circuiti di controllo esterni devono essere dotati di circuiti di arresto di emergenza, circuiti di interblocco, circuiti di finecorsa e altre misure di sicurezza analoghe.
 - Il PLC disattiva tutte le uscite quando la funzione di autodiagnostica rileva un errore o viene eseguita un'istruzione FALS (allarme di guasto grave). Come contromisura in caso di tali errori, il sistema deve essere dotato di misure di sicurezza esterne.
 - Le uscite del PLC potrebbero restare attivate o disattivate in caso di deposizione elettrolitica, bruciatura dei relè di uscita o distruzione dei transistor di uscita. Come contromisura per questo problema, il sistema deve essere dotato di misure di sicurezza esterne.
 - In presenza di sovraccarico o cortocircuito sull'uscita a 24 Vc.c. (alimentazione di servizio del PLC), si potrebbe verificare un abbassamento di tensione e una conseguente disattivazione delle uscite. Come contromisura per questo problema, il sistema deve essere dotato di misure di sicurezza esterne.
- ⚠ Attenzione** Verificare lo stato di sicurezza prima di trasferire file di dati memorizzati nella memoria file (scheda di memoria o area di memoria file nell'area EM) all'area degli I/O (CIO) della CPU utilizzando un dispositivo periferico. In caso contrario, è possibile che i dispositivi collegati al modulo di uscita non funzionino in modo corretto, indipendentemente dalla modalità operativa della CPU.
- ⚠ Attenzione** Il cliente è tenuto a implementare meccanismi di sicurezza per guasti ed errori allo scopo di garantire la sicurezza in caso di segnali errati, mancanti o anormali provocati da guasti a carico delle linee di segnale, cadute di tensione temporanee o altre cause. Se non vengono adottate misure appropriate, il funzionamento anomalo del sistema potrebbe essere causa di gravi incidenti.
- ⚠ Attenzione** Eseguire modifiche in linea solo dopo aver verificato che l'estensione del tempo di ciclo non provoca effetti negativi. In caso contrario, i segnali di ingresso potrebbero risultare illeggibili.
- ⚠ Attenzione** Le CPU CS1-H, CJ1-H, CJ1M e CS1D eseguono automaticamente il backup del programma utente e dei dati di parametro nella memoria flash quando tali dati vengono scritti nella CPU. I dati della memoria I/O (comprese le aree DM, EM e HR) non vengono tuttavia scritti nella memoria flash. Un'apposita batteria consente di conservare i dati delle aree DM, EM e HR nel caso in cui si verificano cadute di tensione. Se si verificano errori relativi alla batteria, i dati contenuti in queste aree potrebbero risultare non corretti dopo la caduta di tensione. Se i dati delle aree DM, EM e HR vengono utilizzati per il controllo di uscite esterne, impedire l'invio di dati non corretti quando il flag di errore della batteria (A40204) è attivato.
- ⚠ Attenzione** Verificare lo stato di sicurezza sul nodo di destinazione prima di trasferire un programma o modificare il contenuto dell'area di memoria I/O. La mancata osservanza di questa precauzione prima di procedere a tali operazioni implica il rischio di lesioni.

- ⚠ **Attenzione** Serrare le viti sulla morsettiera del modulo di alimentazione c.a. applicando la coppia specificata nel manuale dell'operatore. La presenza di viti allentate può provocare bruciature o malfunzionamento.
- ⚠ **Attenzione** Non toccare il modulo di alimentazione mentre eroga corrente o immediatamente dopo lo spegnimento, poiché la temperatura del modulo di alimentazione potrebbe essere tale da provocare ustioni.
- ⚠ **Attenzione** Prestare particolare attenzione durante il collegamento di PC o altri dispositivi periferici a un PLC su cui sia installato un modulo non isolato (CS1W-CLK12/52(-V1) o CS1W-ETN01) collegato a un alimentatore esterno. Se l'estremità a 24 V dell'alimentatore esterno viene collegata a terra e l'estremità a 0 V della periferica viene collegata a terra, si creerà un corto circuito. Durante il collegamento di una periferica a questo tipo di PLC, eseguire la messa a terra dell'estremità a 0 V dell'alimentatore esterno oppure non eseguire affatto la messa a terra dell'alimentatore esterno.

4 Precauzioni relative all'ambiente operativo

- ⚠ **Attenzione** Non utilizzare il sistema di controllo nei seguenti luoghi:
 - Luoghi esposti alla luce solare diretta.
 - Luoghi con temperature o tassi di umidità al di fuori dell'intervallo di valori riportato nelle specifiche.
 - Luoghi soggetti a formazione di condensa a causa di considerevoli escursioni termiche.
 - Luoghi esposti a gas corrosivi o infiammabili.
 - Luoghi esposti a polvere (in particolare polvere metallica) o agenti salini.
 - Luoghi esposti ad acqua, oli o agenti chimici.
 - Luoghi soggetti a urti o vibrazioni.
- ⚠ **Attenzione** Applicare soluzioni di sicurezza adeguate e sufficienti quando si installano sistemi nei seguenti luoghi:
 - Luoghi soggetti a elettricità statica o altre forme di disturbi.
 - Luoghi in cui sono presenti forti campi elettromagnetici.
 - Luoghi potenzialmente esposti a radioattività.
 - Luoghi in prossimità di fonti di alimentazione.
- ⚠ **Attenzione** L'ambiente in cui opera il PLC può avere un grande impatto sulla durata e sull'affidabilità del sistema. L'utilizzo in ambienti operativi non appropriati può essere causa di malfunzionamento, guasti e altri problemi non prevedibili. Accertarsi che l'ambiente operativo rispetti le condizioni richieste per l'installazione e che tali condizioni siano mantenute per l'intera durata di esercizio del sistema.


5 Precauzioni per le applicazioni

Osservare le seguenti precauzioni nell'uso del PLC.

- Per la programmazione di più task, utilizzare CX-Programmer, il software di programmazione per Windows. La console di programmazione può essere utilizzata per programmare un solo task ciclico e task ad interrupt. È tuttavia possibile utilizzare una console di programmazione per modificare programmi composti da più task creati con CX-Programmer.

- ⚠ **AVVERTENZA** Attenersi sempre alle seguenti precauzioni. Il mancato rispetto di tali precauzioni può essere causa di lesioni gravi, anche mortali.

- Durante l'installazione dei moduli, effettuare sempre un collegamento a terra con una resistenza di 100 Ω o inferiore. Il mancato collegamento a terra a una resistenza di 100 Ω o inferiore potrebbe determinare scosse elettriche.
- Se si collegano in cortocircuito i terminali di messa a terra della linea (LG) e di messa a terra (GR) sul modulo di alimentazione, eseguire un collegamento a terra con una resistenza di 100 Ω o inferiore.
- Spegnerne sempre il PLC e scollegare l'alimentazione prima di eseguire una delle operazioni riportate di seguito. La mancata interruzione dell'alimentazione comporta il rischio di scosse elettriche o malfunzionamento.
 - Montaggio o smontaggio di moduli di alimentazione, moduli di I/O, CPU, schede interne o altri moduli.
 - Assemblaggio di moduli.
 - Impostazione di DIP switch o di selettori rotanti.
 - Collegamento di cavi o cablaggio del sistema.
 - Collegamento e scollegamento di connettori.

 **Attenzione** Il mancato rispetto delle seguenti precauzioni può causare un malfunzionamento del PLC o del sistema o danni al PLC e ai relativi moduli. Seguire sempre tali precauzioni.

- Una copia di backup del programma utente e dei dati dell'area dei parametri nelle CPU CS1-H, CS1D, CJ1-H e CJ1M viene salvata nella memoria flash integrata. Durante l'esecuzione del backup, l'indicatore BKUP posto sulla parte anteriore della CPU si accende. Non spegnere la CPU quando l'indicatore BKUP è acceso. Se si interrompe l'alimentazione, il backup dei dati non verrà eseguito.
- Quando si utilizza una CPU CS1 per la prima volta, prima di procedere alla programmazione installare la batteria CS1W-BAT1 fornita con la CPU e cancellare tutte le aree di memoria utilizzando un dispositivo di programmazione. Quando si utilizza l'orologio interno, accendere il sistema dopo aver installato la batteria e impostare l'orologio tramite un dispositivo di programmazione o utilizzando l'istruzione DATE(735). L'orologio non verrà avviato finché non viene impostata l'ora.
- Al momento dell'acquisto, il PLC è configurato in modo tale che la CPU venga avviata nella modalità operativa impostata tramite il selettore di modalità della console di programmazione. Se non si collega una console di programmazione, le CPU CS1 della serie CS verranno avviate in modalità PROGRAM, mentre le CPU CS1-H, CS1D, CJ1, CJ1-H o CJ1M verranno avviate in modalità RUN, entrando immediatamente in funzione. Accertarsi che i moduli non entrino in funzione senza avere prima verificato le condizioni di sicurezza.
- Quando si crea un file AUTOEXEC.IOM mediante un dispositivo di programmazione, quale una console di programmazione o il software CX-Programmer, per il trasferimento automatico dei dati all'avvio, impostare il primo indirizzo di scrittura su D20000 e verificare che le dimensioni dei dati non superino la capacità dell'area DM. Quando all'avvio viene letto il file di dati dalla scheda di memoria, i dati vengono scritti nella CPU a partire dall'indirizzo D20000, anche se al momento della creazione del file AUTOEXEC.IOM è stato specificato un indirizzo diverso. Inoltre, se si supera la capacità dell'area DM, e tale possibilità può verificarsi quando si utilizza CX-Programmer, i dati rimanenti verranno scritti nell'area EM.
- Accendere sempre il PLC prima di accendere il sistema di controllo. Se il PLC viene acceso dopo il sistema di controllo, potrebbero verificarsi errori di segnale temporanei a livello del sistema di controllo, in quanto i termi-

nali di uscita sui moduli di uscita c.c. e altri moduli vengono momentaneamente sollecitati all'accensione del PLC.

- Il cliente è tenuto a implementare meccanismi di sicurezza per guasti ed errori allo scopo di garantire la sicurezza nel caso in cui le uscite dei moduli di uscita rimangano attivate a seguito di guasti dei circuiti interni a carico di relè, transistor e altri elementi.
- Il cliente è tenuto a implementare meccanismi di sicurezza per guasti ed errori allo scopo di garantire la sicurezza in caso di segnali errati, mancanti o anomali provocati da guasti a carico delle linee di segnale, cadute di tensione temporanee o altre cause.
- È responsabilità del cliente dotare i circuiti esterni, ovvero esterni al controllore programmabile, di circuiti di interblocco, circuiti di finecorsa e altre misure di sicurezza analoghe.
- Non spegnere il PLC durante il trasferimento di dati. In particolare, non spegnere il PLC durante le operazioni di lettura o scrittura su una scheda di memoria e non rimuovere la scheda di memoria mentre l'indicatore BUSY è acceso. Per rimuovere una scheda di memoria, premere innanzitutto l'interruttore di alimentazione della scheda di memoria, quindi attendere che l'indicatore BUSY si spenga prima di rimuoverla.
- Se il bit di ritentività IOM è impostato su ON e si passa dalla modalità operativa RUN o MONITOR alla modalità PROGRAM, le uscite del PLC non vengono disattivate e mantengono lo stato precedente. Accertarsi che i carichi esterni non instaurino condizioni pericolose quando ciò accade. Se il funzionamento viene interrotto a causa di un errore fatale, inclusi gli errori generati dall'istruzione FALS(007), tutte le uscite del modulo di uscita vengono disattivate e viene mantenuto solo lo stato delle uscite interne.
- L'integrità del contenuto delle aree DM, EM e HR della CPU è assicurata da una batteria di backup. Se la batteria si scarica, i dati potrebbero andare persi. Adottare contromisure adeguate a livello di programma utilizzando il flag di errore della batteria (A40204) per reinizializzare i dati o intervenire in altro modo in caso di esaurimento della batteria.
- Quando si fornisce corrente a 200-240 Vc.a. con un PLC della serie CS, rimuovere sempre il ponticello di metallo dai terminali del selettore di tensione del modulo di alimentazione (tale operazione non è necessaria per moduli di alimentazione con gamme di tensione più ampie). Se il ponticello di metallo rimane collegato durante l'alimentazione a 200-240 Vc.a., il prodotto verrà distrutto.
- Utilizzare sempre le tensioni di alimentazione specificate nei manuali dell'operatore. Una tensione errata può provocare un malfunzionamento o bruciature.
- Adottare le misure necessarie per garantire che il sistema sia sempre alimentato nel rispetto delle specifiche di tensione e frequenza nominali. In particolare, fare molta attenzione in luoghi dove l'alimentazione è instabile. Un'alimentazione non adeguata può comportare un funzionamento incorretto.
- Installare interruttori esterni o altri dispositivi di sicurezza per evitare cortocircuiti nelle aree di cablaggio esterno. Misure insufficienti di protezione da cortocircuiti potrebbero causare bruciature.
- Non applicare ai moduli di ingresso tensioni superiori alla tensione di ingresso nominale. Tensioni eccessivamente alte potrebbero essere causa di bruciature.
- Non applicare tensioni o collegare carichi ai moduli di uscita superiori alla corrente di carico massima. Tensioni o carichi eccessivamente elevati potrebbero essere causa di bruciature.

- Separare il terminale di messa a terra della linea (LG) dal terminale di messa a terra funzionale (GR) sul modulo di alimentazione prima di eseguire test della tensione di resistenza o test della resistenza di isolamento. In caso contrario possono verificarsi bruciature.
- Installare i moduli in modo appropriato, seguendo le istruzioni riportate nei manuali dell'operatore. L'installazione errata dei moduli può comportare un malfunzionamento.
- Con i PLC della serie CS, accertarsi che tutte le viti di montaggio del modulo e del rack siano serrate rispettando la coppia specificata nei relativi manuali. Una coppia di serraggio non appropriata può provocare un malfunzionamento.
- Accertarsi che tutte le viti dei terminali e le viti dei connettori dei cavi siano serrate rispettando la coppia specificata nei relativi manuali. Una coppia di serraggio non appropriata può provocare un malfunzionamento.
- Durante il cablaggio, lasciare l'etichetta attaccata al modulo. La rimozione dell'etichetta può comportare la penetrazione di materiale estraneo nel modulo e il conseguente malfunzionamento.
- Una volta completato il cablaggio, rimuovere l'etichetta per garantire un'adeguata dissipazione del calore. Se non si rimuove l'etichetta, il modulo potrebbe non funzionare correttamente.
- Quando si procede al cablaggio, utilizzare terminali a crimpare. Non collegare direttamente ai terminali fili scoperti. Il collegamento diretto di fili scoperti può causare bruciature.
- Accertarsi di cablare correttamente tutti i collegamenti.
- Verificare a fondo l'intero cablaggio e le impostazioni degli interruttori prima di attivare l'alimentazione. Un cablaggio errato può essere causa di bruciature.
- Installare i moduli solo dopo avere verificato tutte le morsettiere e i connettori.
- Accertarsi che le morsettiere, i moduli di memoria, le prolunghe e altri componenti dotati di dispositivi di bloccaggio siano correttamente bloccati in posizione. L'errato bloccaggio di questi componenti può causare un malfunzionamento.
- Verificare le impostazioni degli interruttori, il contenuto dell'area DM e ogni altro prerequisito prima di mettere in funzione il sistema. L'avvio in presenza di impostazioni o dati non corretti può provocare un funzionamento imprevisto.
- Verificare la corretta esecuzione del programma utente prima di eseguirlo sul modulo. La mancata verifica del programma può provocare un funzionamento imprevisto.
- Prima di eseguire le operazioni riportate di seguito, accertarsi che non abbiano effetti negativi sul sistema. Disattendere questa precauzione potrebbe dare luogo a un funzionamento imprevisto.
 - Cambio della modalità operativa del PLC (compresa l'impostazione della modalità operativa all'avvio).
 - Impostazione/reset forzato di qualunque bit in memoria.
 - Modifica del valore attuale di qualsiasi canale o valore impostato in memoria.
- Non tirare o piegare i cavi oltre il limite di resistenza naturale. Ciò potrebbe provocarne la rottura.
- Non appoggiare alcun oggetto sui cavi. Ciò potrebbe provocarne la rottura.

- Non utilizzare cavi RS-232C per personal computer di terze parti. Utilizzare sempre i cavi speciali elencati in questo manuale o assemblare i cavi in conformità alle specifiche indicate. L'impiego di cavi di terze parti potrebbe causare danni ai dispositivi esterni o alla CPU.
- Non collegare mai il pin 6 (alimentazione a 5 V) della porta RS-232C di una CPU a un dispositivo diverso da un adattatore NT-AL001 o CJ1W-CIF11. Il dispositivo esterno o la CPU potrebbe subire danni.
- Quando si sostituiscono componenti, accertarsi sempre che le specifiche tecniche del nuovo componente siano appropriate. Disattendere questa precauzione può causare un malfunzionamento o bruciature.
- Prima di toccare un modulo, toccare un oggetto metallico con messa a terra per scaricare l'elettricità statica accumulata. Disattendere questa precauzione può causare un malfunzionamento o danneggiare il modulo.
- Quando si trasportano o immagazzinano schede di circuiti stampati, coprirle sempre con materiale antistatico per proteggerle dall'elettricità statica e mantenere la temperatura di stoccaggio appropriata.
- Non toccare le schede di circuiti stampati o i componenti montati sulle schede a mani nude. Sulle schede vi sono contatti appuntiti ed altre parti che, se toccate incautamente, potrebbero provocare lesioni.
- Non cortocircuitare i terminali della batteria né caricare, smontare, scaldare eccessivamente o incenerire la batteria. Non sottoporre la batteria a forti urti. Ciò potrebbe provocare dispersione, rottura, emissione di calore o combustione della batteria. Eliminare le batterie che hanno subito forti urti, ad esempio cadendo sul pavimento, in quanto potrebbero verificarsi perdite di elettroliti durante l'uso.
- Gli standard UL richiedono che le batterie vengano sostituite solo da tecnici esperti. Consentire la sostituzione delle batterie solo da parte di personale qualificato.
- Nei PLC della serie CJ, i dispositivi scorrevoli posti sulla parte superiore e inferiore del modulo di alimentazione, della CPU, dei moduli di I/O, dei moduli di I/O speciale e dei moduli CPU bus devono essere bloccati, facendoli scorrere finché non scattano in posizione. Se i dispositivi scorrevoli non vengono bloccati, il modulo non funzionerà in modo corretto.
- Nei PLC della serie CJ, collegare sempre la piastrina di blocco al modulo sul lato destro del PLC. Senza piastrina di blocco il PLC non funzionerà in modo corretto.
- L'impostazione di tabelle data link o parametri non corretti può dare luogo a un funzionamento imprevisto. Anche se le tabelle data link e i parametri sono stati impostati correttamente, verificare che il sistema controllato non ne sia influenzato negativamente prima di avviare o interrompere i data link.
- Quando le tabelle di routing vengono trasferite da un dispositivo di programmazione alla CPU, i moduli CPU bus vengono riavviati. Il riavvio di tali unità è necessario per la lettura e l'abilitazione delle nuove tabelle di routing. Verificare che il sistema non evidenzii problemi prima di consentire la reimpostazione delle unità bus CPU.

6 Conformità alle direttive dell'Unione Europea

6-1 Direttive applicabili

- Direttive sulla compatibilità elettromagnetica (EMC)
- Direttiva per le basse tensioni (LVD)

6-2 Principi

Direttive sulla compatibilità elettromagnetica (EMC)

I dispositivi OMRON conformi alle Direttive dell'Unione Europea sono altresì conformi agli standard EMC, in modo da poter essere facilmente incorporati in altri dispositivi o macchine complesse. La conformità dei singoli prodotti agli standard EMC è stata verificata (vedere nota). Tuttavia, la conformità del prodotto agli standard, una volta installato nel sistema del cliente, deve essere verificata dal cliente stesso.

Le prestazioni relative agli standard EMC dei dispositivi OMRON conformi alle Direttive dell'Unione Europea variano a seconda della configurazione, cablaggio e altre condizioni dell'apparecchiatura o pannello di controllo nel quale i dispositivi OMRON sono installati. Pertanto il cliente deve effettuare i controlli finali per accertarsi che tali dispositivi e la macchina nel suo complesso siano conformi agli standard EMC.

Nota Gli standard di compatibilità elettromagnetica applicabili sono i seguenti:

| | |
|--|--|
| EMS (susceptibilità elettromagnetica): | EN61131-2 (Serie CS)/ EN61000-6-2 (Serie CJ) |
| EMI (interferenza elettromagnetica): | EN61000-6-4 (Emissione irradiata: norme 10 m) |

Direttiva per le basse tensioni (LVD)

Accertarsi sempre che i dispositivi che operano nella gamma di tensioni 50-1.000 Vc.a. e 75-1.500 Vc.c. soddisfino gli standard di sicurezza per il PLC (EN 61131-2).

6-3 Conformità alle direttive dell'Unione Europea

I PLC della serie CS/CJ sono conformi alle Direttive dell'Unione Europea. Per garantire che la macchina o il dispositivo in cui viene utilizzato il PLC della serie CS/CJ sia conforme alle Direttive dell'Unione Europea, è necessario soddisfare i seguenti requisiti di installazione del PLC:

- 1,2,3...**
1. Il PLC della serie CS/CJ deve essere installato in un pannello di controllo.
 2. È necessario utilizzare un isolamento rinforzato o un doppio isolamento per gli alimentatori c.c. utilizzati per l'alimentazione dei moduli di comunicazione e di I/O.
 3. I PLC della serie CS/CJ conformi alle Direttive dell'Unione Europea sono altresì conformi agli standard generici sulle emissioni (EN61000-6-4). Le caratteristiche per le emissioni irradiate possono variare in base alla configurazione del quadro utilizzato, agli altri dispositivi collegati al quadro, al cablaggio e ad altre condizioni specifiche. È pertanto necessario verificare che la macchina nel suo complesso o l'apparecchiatura utilizzata sia conforme alle Direttive dell'Unione Europea.

6-4 Metodi di riduzione dei disturbi nelle uscite a relè

I PLC della serie CS/CJ sono conformi agli standard generici sulle emissioni (EN61000-6-4) delle Direttive EMC. Tuttavia, i disturbi generati dalla commutazione delle uscite a relè potrebbero non risultare conformi a questi standard. Qualora ciò si verificasse, è necessario applicare un filtro antidisturbo dal lato carico o adottare altre soluzioni equivalenti esterne al PLC.

Le soluzioni da adottare per soddisfare i requisiti posti dagli standard possono variare a seconda dei dispositivi collegati sul lato carico, del cablaggio, della configurazione delle macchine e così via. Di seguito sono riportati alcuni esempi di soluzioni per la riduzione dei disturbi generati.

Contromisure

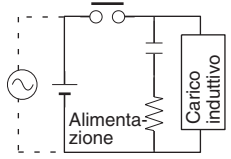
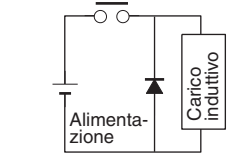
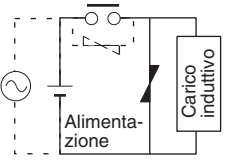
Per ulteriori informazioni, fare riferimento alla norma EN61000-6-4.

Non è necessaria alcuna contromisura per la riduzione dei disturbi se la frequenza di commutazione del carico per l'intero sistema, incluso il PLC, è inferiore a 5 volte al minuto.

È necessario adottare contromisure per la riduzione dei disturbi se la frequenza di commutazione del carico per l'intero sistema, incluso il PLC, è superiore a 5 volte al minuto.

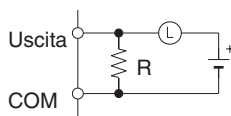
Esempi di contromisure

Quando si commuta un carico induttivo, collegare un dispositivo di protezione da sovracorrente, diodi e così via in parallelo con il carico o contatto come mostrato di seguito.

| Circuito | Corrente | | Caratteristiche | Requisiti |
|---|----------|------|--|---|
| | C.A. | C.C. | | |
| <p>Metodo CR</p>  | Si | Si | <p>Se il carico è un relè o solenoide, si determina un ritardo tra l'apertura del circuito e il ripristino del carico.</p> <p>Se la tensione di alimentazione è 24 o 48 V, collegare il dispositivo di protezione da sovracorrente in parallelo con il carico. Se la tensione di alimentazione è da 100 a 200V, inserire il dispositivo di protezione da sovracorrente tra i contatti.</p> | <p>La capacità del condensatore deve essere di 1-0,5 μF per una corrente di contatto pari a 1 A e la resistenza del resistore deve essere di 0,5-1 Ω per una tensione di contatto pari a 1 V. Tuttavia, questi valori possono variare in base al carico e alle caratteristiche del relè. Definire tali valori sulla base di test, tenendo in considerazione il fatto che il condensatore sopprime la scarica della scintilla quando i contatti vengono separati, mentre il resistore limita il flusso di corrente verso il carico quando il circuito viene richiuso.</p> <p>La rigidità dielettrica del condensatore deve essere di 200-300 V. Se il circuito è un circuito c.a., utilizzare un condensatore senza polarità.</p> |
| <p>Metodo con diodo</p>  | No | Si | <p>L'energia accumulata nella bobina viene trasformata in corrente dal diodo collegato in parallelo al carico, quindi la corrente che transita nella bobina viene assorbita e convertita in calore dalla resistenza del carico induttivo.</p> <p>Il ritardo tra l'apertura del circuito e il ripristino del carico indotto da questo metodo è più lungo di quello ottenuto con il metodo CR (condensatore-resistore).</p> | <p>La rigidità dielettrica inversa del diodo deve essere almeno 10 volte superiore alla tensione del circuito. La corrente diretta del diodo deve essere pari o superiore alla corrente di carico.</p> <p>La rigidità dielettrica inversa del diodo deve essere da due a tre volte superiore alla tensione di alimentazione se ai circuiti elettronici con basse tensioni è applicato un dispositivo di protezione da sovracorrente.</p> |
| <p>Metodo con varistore</p>  | Si | Si | <p>Il metodo con varistore impedisce l'imposizione di tensioni elevate tra i contatti sfruttando la tensione costante caratteristica del varistore. Si determina un ritardo tra l'apertura del circuito e il ripristino del carico.</p> <p>Se la tensione di alimentazione è 24 o 48 V, collegare il varistore in parallelo con il carico. Se la tensione di alimentazione è da 100 a 200 V, inserire il varistore tra i contatti.</p> | --- |

Quando si commuta un carico con una forte corrente di picco, come in una lampada a incandescenza, ridurre la corrente come illustrato di seguito.

Soluzione 1



Generando una corrente di riposo pari a circa un terzo del valore nominale della lampada a incandescenza

Soluzione 2



Utilizzando un resistore per limitare il flusso di corrente

CAPITOLO 1

Introduzione

In questa sezione sono riportate le informazioni sulle caratteristiche generali delle istruzioni e sugli errori che possono verificarsi durante l'esecuzione di un'istruzione.

| | | |
|-------|--|----|
| 1-1 | Caratteristiche generali delle istruzioni. | 2 |
| 1-1-1 | Capacità del programma. | 2 |
| 1-1-2 | Istruzioni differenziate | 3 |
| 1-1-3 | Variazioni di istruzione. | 4 |
| 1-1-4 | Posizione delle istruzioni e condizioni di esecuzione. | 5 |
| 1-1-5 | Inserimento di dati negli operandi | 5 |
| 1-1-6 | Formati dei dati. | 11 |
| 1-2 | Controlli dell'esecuzione di istruzioni. | 13 |
| 1-2-1 | Errori che possono verificarsi durante l'esecuzione di un'istruzione | 13 |
| 1-2-2 | Errori irreversibili (errori di programma). | 13 |

1-1 Caratteristiche generali delle istruzioni

1-1-1 Capacità del programma

La capacità del programma indica le dimensioni dell'area del programma utente nel modulo CPU e viene espressa come numero di step di un programma. Il numero di step necessari nell'area del programma utente per ciascuna delle istruzioni della serie CS/CJ varia da 1 a 7 a seconda dell'istruzione e degli operandi specificati.

Serie CS

Nelle tabelle seguenti è riportato il numero massimo di step programmabili in ciascun modulo CPU della serie CS.

- CPU CS1-H

| Modello | Capacità del programma | Punti di I/O |
|-------------|------------------------|--------------|
| CS1H-CPU67H | 250.000 step | 5.120 |
| CS1H-CPU66H | 120.000 step | |
| CS1H-CPU65H | 60.000 step | |
| CS1H-CPU64H | 30.000 step | |
| CS1H-CPU63H | 20.000 step | |
| CS1G-CPU45H | 60.000 step | |
| CS1G-CPU44H | 30.000 step | 1.280 |
| CS1G-CPU43H | 20.000 step | 960 |
| CJ1G-CPU42H | 10.000 step | |

- CPU CS1

| Modello | Capacità del programma | Punti di I/O |
|--------------|------------------------|--------------|
| CS1H-CPU67-E | 250.000 step | 5.120 |
| CS1H-CPU66-E | 120.000 step | |
| CS1H-CPU65-E | 60.000 step | |
| CS1H-CPU64-E | 30.000 step | |
| CS1H-CPU63-E | 20.000 step | |
| CS1G-CPU45-E | 60.000 step | |
| CS1G-CPU44-E | 30.000 step | 1.280 |
| CS1G-CPU43-E | 20.000 step | 960 |
| CS1G-CPU42-E | 10.000 step | |

- CPU CS1D per sistemi a singola CPU

| Modello | Capacità del programma | Punti di I/O |
|-------------|------------------------|--------------|
| CS1D-CPU67H | 250.000 step | 5.120 |
| CS1D-CPU65H | 60.000 step | |

CPU CS1D per sistemi a due CPU

| Modello | Capacità del programma | Punti di I/O |
|-------------|------------------------|--------------|
| CS1D-CPU42S | 10.000 step | 960 |
| CS1D-CPU44S | 30.000 step | 1.280 |
| CS1D-CPU65S | 60.000 step | 5.120 |
| CS1D-CPU67S | 250.000 step | |

Serie CJ

Nelle tabelle seguenti è riportato il numero massimo di step programmabili in ciascun modulo CPU della serie CJ.

- CPU CJ1-H

| Modello | Capacità del programma | Punti di I/O |
|-------------|------------------------|--------------|
| CJ1H-CPU67H | 250.000 step | 2.560 |
| CJ1H-CPU66H | 120.000 step | |
| CJ1H-CPU65H | 60.000 step | |
| CJ1G-CPU45H | 60.000 step | 1.280 |
| CJ1G-CPU44H | 30.000 step | |
| CJ1G-CPU43H | 20.000 step | 960 |
| CJ1G-CPU42H | 10.000 step | |

- CPU CJ1

| Modello | Capacità del programma | Punti di I/O |
|------------|------------------------|--------------|
| CJ1G-CPU45 | 60.000 step | 1.280 |
| CJ1G-CPU44 | 30.000 step | |

- CPU CJ1M

| Modello | Capacità del programma | Punti di I/O |
|------------|------------------------|--------------|
| CJ1M-CPU23 | 20.000 step | 640 |
| CJ1M-CPU22 | 10.000 step | 320 |
| CJ1M-CPU21 | 5.000 step | 160 |
| CJ1M-CPU13 | 20.000 step | 640 |
| CJ1M-CPU12 | 10.000 step | 320 |
| CJ1M-CPU11 | 5.000 step | 160 |

Nota La capacità dei programmi per i PLC della serie CS/CJ viene misurata in step, mentre la capacità dei programmi per i PLC OMRON precedenti, ad esempio i PLC delle serie C e CV, veniva misurata in canali. In altre parole, 1 step equivale a 1 canale. Tuttavia, poiché per alcune istruzioni della serie CS/CJ la quantità di memoria richiesta per l'esecuzione è diversa rispetto ai modelli precedenti, la conversione per un PLC della serie CJ della capacità di un programma utente sviluppato per un altro PLC in base al criterio secondo cui 1 canale equivale a 1 step non produrrà un valore accurato. Per indicazioni sulla conversione delle capacità di programmi sviluppati per PLC OMRON precedenti, fare riferimento alle informazioni riportate alla fine della *CAPITOLO 4 Tempi di esecuzione delle istruzioni e numero di step*.

Il numero di step in un programma non corrisponde al numero di istruzioni. Ad esempio, le istruzioni LD e OUT richiedono 1 step ciascuna, mentre l'istruzione MOV(021) richiede 3 step. Altre istruzioni richiedono fino a 7 step ciascuna. Inoltre, il numero di step richiesto da un'istruzione viene incrementato di uno step per ogni operando a doppia lunghezza utilizzato nell'istruzione. Ad esempio, l'istruzione MOVL(498) richiede normalmente 3 step, ma saranno necessari 4 step se è stata specificata una costante per l'operando del canale di origine, S. Per informazioni sul numero di step richiesti da ogni istruzione, consultare la *CAPITOLO 4 Tempi di esecuzione delle istruzioni e numero di step*.

1-1-2 Istruzioni differenziate

La maggior parte delle istruzioni nei PLC della serie CS/CJ vengono fornite con variazioni non differenziate e differenziate sul fronte di salita e alcune anche con una variazione differenziata sul fronte di discesa.

- Un'istruzione non differenziata viene eseguita ogni volta che viene analizzata.

- Un'istruzione con differenziazione sul fronte di salita viene eseguita solo una volta dopo che la condizione di esecuzione è passata dallo stato OFF allo stato ON.
- Un'istruzione con differenziazione sul fronte di discesa viene eseguita solo una volta dopo che la condizione di esecuzione è passata dallo stato ON allo stato OFF.

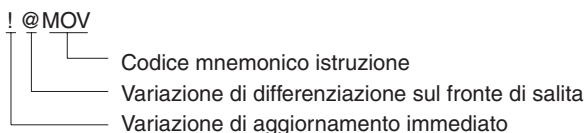
| Variazione | Tipo di istruzione | Operazione | Formato | Esempio |
|--|---|---|---------|--|
| Non differenziata | Istruzioni di uscita (istruzioni che necessitano di una condizione di esecuzione) | L'istruzione viene eseguita ad ogni ciclo quando la condizione di esecuzione è vera (ON). | | |
| | Istruzioni di ingresso (istruzioni utilizzate come condizioni di esecuzione) | L'elaborazione del bit (ad esempio di lettura, confronto o test) viene eseguita ad ogni ciclo. La condizione di esecuzione è vera finché il risultato è ON. | | |
| Differenziata sul fronte di salita (con prefisso @) | Istruzioni di uscita | L'istruzione viene eseguita solo una volta quando la condizione di esecuzione passa dallo stato OFF allo stato ON. | | Istruzione MOV(021) eseguita una volta per ogni transizione da OFF a ON in CIO 000102. |
| | Istruzioni di ingresso (istruzioni utilizzate come condizioni di esecuzione) | L'elaborazione del bit (ad esempio di lettura, confronto o test) viene eseguita ad ogni ciclo. La condizione di esecuzione è vera per ogni ciclo quando il risultato passa dallo stato OFF allo stato ON. | | Condizione di esecuzione con stato ON creata per un solo ciclo per ciascuna transizione da OFF a ON in CIO 000103. |
| Differenziata sul fronte di discesa (con prefisso %) | Istruzioni di uscita | L'istruzione viene eseguita solo una volta quando la condizione di esecuzione passa dallo stato ON allo stato OFF. | | Istruzione SET eseguita una volta per ogni transizione da ON a OFF in CIO 000102. |
| | Istruzioni di ingresso (istruzioni utilizzate come condizioni di esecuzione) | L'elaborazione del bit (ad esempio di lettura, confronto o test) viene eseguita ad ogni ciclo. La condizione di esecuzione è vera per ogni ciclo quando il risultato passa dallo stato ON allo stato OFF. | | Condizione di esecuzione con stato ON creata per un solo ciclo per ciascuna transizione da ON a OFF in CIO 000103. |

Nota L'opzione di differenziazione sul fronte di discesa (%) è disponibile solo per le istruzioni LD, AND, OR e RSET. Per creare variazioni con differenziazione sul fronte di discesa di altre istruzioni, è necessario controllare l'esecuzione dell'istruzione con bit di lavoro controllati con DIFD(014) o DOWN(522).

1-1-3 Variazioni di istruzione

È possibile aggiungere a un'istruzione un prefisso di variazione (@, % e !) per creare un'istruzione differenziata o fornire un aggiornamento immediato.

| Variazione | | Prefisso | Operazione |
|-------------------------|---------------------------------------|----------|--|
| Differenziazione | Differenziazione sul fronte di salita | @ | Crea un'istruzione con differenziazione sul fronte di salita. |
| | Differenziata sul fronte di discesa | % | Crea un'istruzione con differenziazione sul fronte di discesa. |
| Aggiornamento immediato | | ! | Quando l'istruzione viene eseguita, verranno aggiornati i dati dell'operando dell'istruzione nell'area di I/O. |



1-1-4 Posizione delle istruzioni e condizioni di esecuzione

La tabella seguente mostra le posizioni in cui è possibile programmare le istruzioni e indica inoltre quando un'istruzione necessita o non necessita di una condizione di esecuzione. Per ulteriori informazioni su istruzioni specifiche, consultare la *CAPITOLO 2 Sommario delle istruzioni*.

| Tipo di istruzione | | Posizione | Condizione di esecuzione | Formato | Esempi |
|--------------------|---|---|--------------------------|---------|--|
| Ingresso | Istruzioni che avviano condizioni logiche | Sul bus sinistro o all'avvio di un blocco di istruzioni | Non richiesta | | Istruzioni LD, LD TST e istruzioni di confronto di ingresso, come LD > |
| | Istruzioni di collegamento | Tra un'istruzione di avvio e un'istruzione di uscita | Richiesta | | Istruzioni AND, OR, AND TST, istruzioni di confronto di ingresso come AND >, UP, DOWN, NOT |
| Uscita | | Sul bus destro | Richiesta | | La maggior parte della istruzioni (come OUT e MOV) |
| | | | Non richiesta | | Istruzioni come END, JME, FOR e ILC |

Oltre a queste istruzioni, i PLC della serie CS/CJ dispongono di istruzioni di programmazione a blocchi. Per ulteriori informazioni, fare riferimento alle descrizioni delle istruzioni di programmazione a blocchi.

Nota Se una condizione di esecuzione non precede un'istruzione che ne richiede una, si verificherà un errore di programma quando il programma verrà controllato da un dispositivo periferico.

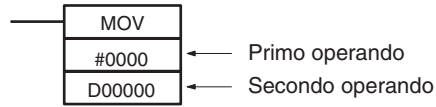
1-1-5 Inserimento di dati negli operandi

Gli operandi sono parametri impostati in precedenza con gli indirizzi di memoria I/O o con le costanti da utilizzare quando viene eseguita l'istruzione. Esistono fondamentalmente tre tipi di operandi: operandi di origine, operandi di destinazione e numeri.



| Operando | | Codice abituale | Descrizione | |
|--------------|--|-----------------|---------------------|--|
| Origine | Indirizzo contenente i dati o i dati stessi | S | Operando di origine | Dati di origine diversi dai dati di controllo |
| | | C | Dati di controllo | Dati di controllo con uno o più bit che controllano l'esecuzione dell'istruzione |
| Destinazione | Indirizzo in cui verranno memorizzati i dati | D | --- | |
| Numero | Contiene un numero, ad esempio un numero di salto o un numero di subroutine. | N | --- | |

Nota È possibile fare riferimento agli operandi di un'istruzione anche in base alla relativa posizione nell'istruzione (primo operando, secondo operando, ecc.). I codici utilizzati per l'operando variano in base alla funzione specifica dell'operando stesso.



Specifica degli indirizzi di bit

| Descrizione | Esempio | Esempio di istruzione |
|---|---|--|
| <p>Per specificare un indirizzo di bit, specificare direttamente l'indirizzo del canale e l'indirizzo di bit.</p> <p>□□□□ □□ └───┬───┘ Numero di bit Indirizzo del canale</p> <p>Nota Il formato dell'indirizzo del canale + indirizzo di bit non è usato per i flag di completamento del temporizzatore/contatore o per i flag di esecuzione dei task.</p> | <p>0001 02 └──┬──┘ Bit 02 Canale CIO 000</p> | <p>0001 02 └──┬──┘ ───┘</p> |

Specifica degli indirizzi di canale

| Descrizione | Esempio | Esempio di istruzione |
|--|--|------------------------|
| <p>Per specificare un indirizzo di canale, specificare direttamente l'indirizzo del canale.</p> <p>□□□□ └──┬──┘ Indirizzo del canale</p> | <p>0003 └──┬──┘ Canale CIO 000</p> <p>D00200 └──┬──┘ Canale D00200</p> | <p>MOV 0003 D00200</p> |

Specifica degli indirizzi DM/EM indiretti in modalità binaria

| Descrizione | Esempio | Esempio di istruzione |
|--|---|---|
| <p>Quando il prefisso @ viene immesso prima di un indirizzo DM o EM, il contenuto di tale canale specifica un altro canale che verrà usato come operando. Il contenuto può essere da 0000 a 7FFF (da 0 a 32.767), a seconda dell'indirizzo di canale desiderato nell'area DM o EM.</p> <p>@D□□□□□□ ↓ Contenuto □□□□□□ Da 00000 a 32767 (da 0000 a 7FFF) ↓ D □□□□□□</p> | <p>---</p> | <p>---</p> |
| <p>Quando il contenuto di @D□□□□□□ è compreso tra 0000 e 7FFF (da 00000 a 32.767), è specificato il canale corrispondente tra D00000 e D32767.</p> | <p>@D00300</p> <p>0 1 0 0 └──┬──┘ Decimale: 256 ↓ Specifica D00256.</p> <p>└──┬──┘ Aggiunge il prefisso @.</p> | <p>MOV #0001 @D00300</p> |

| Descrizione | Esempio | Esempio di istruzione |
|---|---|------------------------|
| Quando il contenuto di @D□□□□□□ è compreso tra 8000 e FFFF (da 32.768 a 65.535), è specificato il canale corrispondente tra E0_00000 e E0_32767 nel banco EM 0. | @D00300 <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 5px;">8 0 0 1</div> Decimale: 32.769 ↓ Specifica E0_00001. | --- |
| Quando il contenuto di @En□_□□□□□□ è compreso tra 0000 e 7FFF (da 00000 a 32.767), è specificato il canale corrispondente tra En□_00000 e En□_32767. | @E1_00200 <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 5px;">0 1 0 1</div> Decimale: 257 ↓ Specifica E1_00257. | MOV #0001 @E1_00200 |
| Quando il contenuto di @En□_□□□□□□ è compreso tra 8000 e 7FFF (da 32.768 a 65.535), è specificato il canale corrispondente tra E (□+1)_00000 e E (□+1)_32767 (nel banco EM successivo). | @E1_00200 <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 5px;">8 0 0 2</div> Decimale: 32770 ↓ Specifica E2_00002. | |

Nota Quando nella configurazione del PLC è selezionata la modalità binaria, l'area DM e gli indirizzi del banco EM corrente (da banco 0 a C) vengono considerati come indirizzi di memoria consecutivi. Se un canale DM indirizzato indirettamente contiene un valore maggiore di 32.767, nel banco EM 0 verrà specificato un canale. Ad esempio, E00000 nel banco 0 specificherà quando il canale DM indirizzato indirettamente contiene un valore esadecimale di 8000 (32.768).

Se un canale EM indirizzato indirettamente contiene un valore maggiore di 32.767, nel banco EM successivo verrà specificato un canale. Ad esempio, verrà specificato E3_00000 quando il canale EM indirizzato indirettamente nel banco 2 contiene un valore esadecimale di 8000 (32.768).

Specifica degli indirizzi DM/EM indiretti in modalità BCD

| Metodo | Descrizione | Esempio | Esempio di istruzione |
|---|--|---|-----------------------|
| Indirizzamento indiretto dell'area DM/EM (modalità BCD) | Quando il prefisso * viene immesso prima di un indirizzo DM o EM, il contenuto BCD di tale canale specifica un altro canale che verrà usato come operando. Il contenuto può essere da 0000 a 9999, a seconda dell'indirizzo di canale desiderato nell'area DM o EM. <div style="text-align: center;"> *D □□□□□□ ↓ Contenuto □ □ □ □ □ □ Da 0000 a 9999 (BCD) ↓ D □ □ □ □ □ □ </div> | *D00200 <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 5px;">0 1 0 0</div> ↓ Specifica D00100. — Aggiunge il prefisso *. | MOV #0001 *D00200 |

Indirizzamento dei registri indice

| Metodo | Descrizione | | Esempio | Esempio di istruzione |
|--|---|--|----------------------|---|
| Indirizzamento diretto dei registri indice | L'istruzione MOVR(560) sposta l'indirizzo di memoria del PLC di un canale o un bit in un registro indice (da IR0 a IR15). (L'istruzione MOVRW(561) sposta l'indirizzo di memoria del PLC di un valore attuale del temporizzatore o del contatore in un registro indice). | | IR0 IR2 | MOVR 0010 IR0 Memorizza l'indirizzo di memoria del PLC di CIO 0010 in IR0. MOVR 000102 IR2 Memorizza l'indirizzo di memoria del PLC di CIO 00102 in IR0. |
| Indirizzamento indiretto dei registri indice | Funzionamento di base (nessun offset) | Il canale o il bit nell'indirizzo di memoria I/O contenuto in IR□ viene utilizzato come operando. Inserire una virgola prima del registro indice per indicare l'indirizzamento indiretto. (L'identificazione del bit/canale può essere determinata dall'istruzione o dall'operando). | ,IR0 ,IR1 | LD ,IR0 Carica lo stato del bit all'indirizzo di memoria I/O contenuto in IR0. MOV #0001, IR1 Sposta #0001 nel canale all'indirizzo di memoria I/O contenuto in IR1. |
| | Offset con costante | Il valore di offset (da -2.047 a +2.047) viene aggiunto all'indirizzo di memoria I/O contenuto in IR□ e l'indirizzo risultante viene utilizzato come operando. (L'offset viene convertito in dato binario durante l'esecuzione dell'istruzione). | +5 ,IR0 +31 ,IR1 | LD +5 ,IR0 Aggiunge 5 all'indirizzo di memoria I/O contenuto in IR0, quindi carica lo stato del bit a tale indirizzo. MOV #0001 +31 ,IR1 Aggiunge 31 all'indirizzo di memoria I/O contenuto in IR1, quindi sposta #0001 nel canale a tale indirizzo. |
| | Offset DR | Il contenuto binario con segno del registro dati viene aggiunto all'indirizzo di memoria I/O contenuto in IR□ e l'indirizzo risultante viene utilizzato come operando. | DR0 ,IR0 DR0 ,IR1 | LD DR0 ,IR0 Aggiunge il contenuto di DR0 all'indirizzo di memoria I/O contenuto in IR0, quindi carica lo stato del bit a tale indirizzo. MOV #0001 DR0 ,IR1 Aggiunge il contenuto di DR0 all'indirizzo di memoria I/O contenuto in IR1, quindi sposta #0001 nel canale a tale indirizzo. |
| | Incremento automatico | Dopo la lettura dell'indirizzo di memoria I/O da IR□, il contenuto del registro indice viene incrementato di uno o di due. Incremento di 1: ,R□+ Incremento di 2: ,IR□++ Nota I registri indice verranno incrementati quando viene eseguita l'istruzione, perfino se si verifica un errore e se il flag di errore passa allo stato ON. | ,IR0 ++ ,IR1 ++ | LD ,IR0 ++ Carica lo stato del bit all'indirizzo di memoria I/O contenuto in IR0, quindi incrementa il registro di due. MOV #0001 ,IR1 ++ Sposta #0001 nel canale all'indirizzo di memoria I/O contenuto in IR1, quindi incrementa il registro di uno. |
| | Decremento automatico | Il contenuto di IR□ viene decrementato di uno o di due e come operando viene utilizzato l'indirizzo di memoria I/O. Decremento di 1: ,- IR□ Decremento di 2: ,- -IR□ Nota I registri indice verranno decrementati quando viene eseguita l'istruzione, perfino se si verifica un errore e se il flag di errore passa allo stato ON. | ,-- IR0 ,- IR1 | LD ,- - IR0 Decrementa il contenuto di IR0 di due, quindi carica lo stato del bit all'indirizzo di memoria I/O così ottenuto. MOV #0001 ,- IR1 Decrementa il contenuto di IR0 di uno, quindi sposta #0001 nel canale all'indirizzo di memoria I/O così ottenuto. |

Nota Accertarsi che nei contenuti dei registri indice siano indicati indirizzi di memoria I/O validi.

Specifica di costanti

| Metodo | Operandi applicabili | Formato dei dati | Codice | Intervallo | Esempio |
|------------------------|--|----------------------|--------|------------------------------------|---------|
| Costante dati a 16 bit | Tutti i dati binari e i dati binari inclusi in un intervallo | Binario senza segno | # | Da #0000 a #FFFF | --- |
| | | Decimale con segno | ± | Da -32.768 a +32.767 | --- |
| | | Decimale senza segno | & | Da &0 a &66.535 | --- |
| | Tutti i dati BCD e i dati BCD inclusi in un intervallo | BCD | # | Da #0000 a #9999 | --- |
| Costante dati a 32 bit | Tutti i dati binari e i dati binari inclusi in un intervallo | Binario senza segno | # | Da #0000 0000 a #FFFF FFFF | --- |
| | | Decimale con segno | + - | Da -2.147.483.648 a +2.147.483.647 | --- |
| | | Decimale senza segno | & | Da &0 a &4.294.967.295 | --- |
| | Tutti i dati BCD e i dati BCD inclusi in un intervallo | BCD | # | Da #0000 0000 a #9999 9999 | --- |

Specifica di stringhe di testo

| Metodo | Descrizione | Codice | Esempi | Esempio di istruzione | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|---|--------|--|-----------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|---|--------|----|----|--------|----|----|--------|----|----|--------|----|----|--------|----|----|--------|----|----|
| Stringhe di testo | <p>Il testo viene memorizzato in ASCII (1 byte/carattere, esclusi i caratteri speciali), cominciando con il byte inferiore del canale inferiore nell'intervallo.</p> <p>Nel caso in cui vi sia un numero dispari di caratteri, il valore 00 (NULL) viene memorizzato nel byte superiore dell'ultimo canale nell'intervallo.</p> <p>Nel caso in cui vi sia un numero pari di caratteri, il valore 0000 (due valori NULL) viene memorizzato nel canale successivo all'ultimo nell'intervallo.</p> | | <p>"ABCDE"</p> <table border="1"> <tr><td>"A"</td><td>"B"</td></tr> <tr><td>"C"</td><td>"D"</td></tr> <tr><td>"E"</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>45</td><td>00</td></tr> </table> <p>"ABCD"</p> <table border="1"> <tr><td>"A"</td><td>"B"</td></tr> <tr><td>"C"</td><td>"D"</td></tr> <tr><td>NUL</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>00</td><td>00</td></tr> </table> | "A" | "B" | "C" | "D" | "E" | NUL | 41 | 42 | 43 | 44 | 45 | 00 | "A" | "B" | "C" | "D" | NUL | NUL | 41 | 42 | 43 | 44 | 00 | 00 | <p>MOV\$ D00100 D00200</p> <table border="1"> <tr><td>D00100</td><td>41</td><td>42</td></tr> <tr><td>D00101</td><td>43</td><td>44</td></tr> <tr><td>D00102</td><td>45</td><td>00</td></tr> </table> <p style="text-align: center;">↓</p> <table border="1"> <tr><td>D00200</td><td>41</td><td>42</td></tr> <tr><td>D00201</td><td>43</td><td>44</td></tr> <tr><td>D00202</td><td>45</td><td>00</td></tr> </table> | D00100 | 41 | 42 | D00101 | 43 | 44 | D00102 | 45 | 00 | D00200 | 41 | 42 | D00201 | 43 | 44 | D00202 | 45 | 00 |
| "A" | "B" | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| "C" | "D" | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| "E" | NUL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 41 | 42 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 43 | 44 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 45 | 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| "A" | "B" | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| "C" | "D" | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NUL | NUL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 41 | 42 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 43 | 44 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D00100 | 41 | 42 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D00101 | 43 | 44 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D00102 | 45 | 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D00200 | 41 | 42 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D00201 | 43 | 44 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D00202 | 45 | 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Nel diagramma seguente sono riportati i caratteri che possono essere espressi in ASCII.

| | | Bit più a sinistra | | | | | | | | | | | | | | | | |
|------------------|---|--------------------|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
| Bit più a destra | 0 | | | SP | 0 | @ | P | ' | p | | | | 一 | タ | ミ | | | |
| | 1 | | | ! | 1 | A | Q | a | q | | | | 。 | ア | チ | ム | | |
| | 2 | | | " | 2 | B | R | b | r | | | | 「 | イ | ツ | メ | | |
| | 3 | | | # | 3 | C | S | c | s | | | | 」 | ウ | テ | モ | | |
| | 4 | | | \$ | 4 | D | T | d | t | | | | 、 | エ | ト | ヤ | | |
| | 5 | | | % | 5 | E | U | e | u | | | | ・ | オ | ナ | ユ | | |
| | 6 | | | & | 6 | F | V | f | v | | | | ヲ | カ | ニ | ヨ | | |
| | 7 | | | ' | 7 | G | W | g | w | | | | ア | キ | ヌ | ラ | | |
| | 8 | | | (| 8 | H | X | h | x | | | | イ | ク | ネ | リ | | |
| | 9 | | |) | 9 | I | Y | i | y | | | | ウ | ケ | ノ | ル | | |
| | A | | | * | : | J | Z | j | z | | | | エ | コ | ハ | レ | | |
| | B | | | + | ; | K | [| k | { | | | | オ | サ | ヒ | ロ | | |
| | C | | | , | < | L | ¥ | | | | | | ヤ | シ | フ | ワ | | |
| | D | | | - | = | M |] | m | } | | | | ユ | ス | ヘ | ン | | |
| | E | | | . | > | N | ^ | n | ~ | | | | ヨ | セ | ホ | | | |
| | F | | | / | ? | O | _ | o | | | | | ツ | ソ | マ | ° | | |

Nota Le istruzioni seguenti vengono eseguite anche quando le condizioni di ingresso sono OFF. Pertanto, quando gli indirizzi di memoria indiretti vengono specificati mediante incremento e decremento automatico (,IR+ o ,IR-) in un operando di una di queste istruzioni, il valore nel registro indice (IR) viene aggiornato ad ogni ciclo, indipendentemente dalla condizione di ingresso (incrementato o decrementato di uno ad ogni ciclo). Quando si scrive un programma, è necessario tenere presente questo fattore.

| Classificazione | Istruzioni |
|---|--|
| Istruzioni di ingresso sequenza | LD, LD NOT, AND, AND NOT, OR, OR NOT, LD TST(350), LD TSTN(351), AND TST(350), AND TSTN(351), OR TST(350), OR TSTN(351) |
| Istruzioni di uscita sequenza | OUT, OUT NOT, DIFU(013), DIFD(014) |
| Istruzioni di controllo sequenza | JMP(004), FOR(512) |
| Istruzioni di temporizzatore e contatore | TIM/TIMX(550), TIMH(015)/TIMHX(551), TMHH(540)/TMHHX(552), TTIM(087)/TTIMX(555), TIML(542)/TIMLX(553), MTIM(533)/MTIMX(554), CNT/CNTX(546), CNTR(012)/CNTRX(548) |
| Istruzioni di confronto | Istruzioni di confronto con simboli (LD, AND, OR =, ecc.(codici di funzione: 300, 305, 310, 320 e 325)) |
| Istruzioni matematiche a virgola mobile in singola precisione | Confronto di dati a virgola mobile in singola precisione (LD, AND, OR = F, ecc.(codice di funzione: da 329 a 334)) |
| Istruzioni matematiche a virgola mobile in doppia precisione | Confronto di dati a virgola mobile in doppia precisione (LD, AND, OR = D, ecc.(codice di funzione: da 335 a 340)) |
| Istruzioni di programmazione a blocchi | BPPS(811), BPRS(812), EXIT(806), EXIT(806) NOT, IF(802), IF(802) NOT, WAIT(805), WAIT(805) NOT, TIMW(813)/TIMWX(816), CNTW(814)/CNTWX(818), TMHW(815)/TMHWX(817), LEND(810), LEND(810) NOT |
| Istruzioni di elaborazione delle stringhe di testo | STRING COMPARISON (LD, AND, OR = \$, ecc. (codice di funzione: da 670 a 675)) |

Gli esempi seguenti sulla programmazione ladder mostrano come vengono considerati i registri indice.

Esempio 1

Programma ladder:

```
LD P_Off
OUT, IR0+
```

Operazione: quando l'indirizzo di memoria del PLC 000013 viene memorizzato in IR0.

La condizione di ingresso è OFF (P_Off è il flag di sempre OFF), di modo che l'istruzione OUT imposti 000013, indirizzato indirettamente da IR0, su OFF. L'istruzione OUT viene eseguita e quindi IR0 viene incrementato. Come risultato viene memorizzato l'indirizzo di memoria del PLC 000014, che è stato incrementato di +1 nel registro indice IR0. Pertanto, nel ciclo seguente l'istruzione OUT imposta 000014 nello stato OFF.

Esempio 2

Programma ladder:

```
LD P_Off
SET, IR0+
```

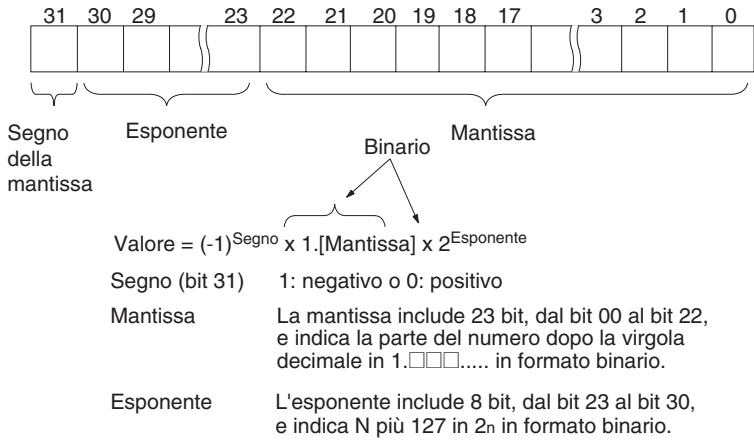
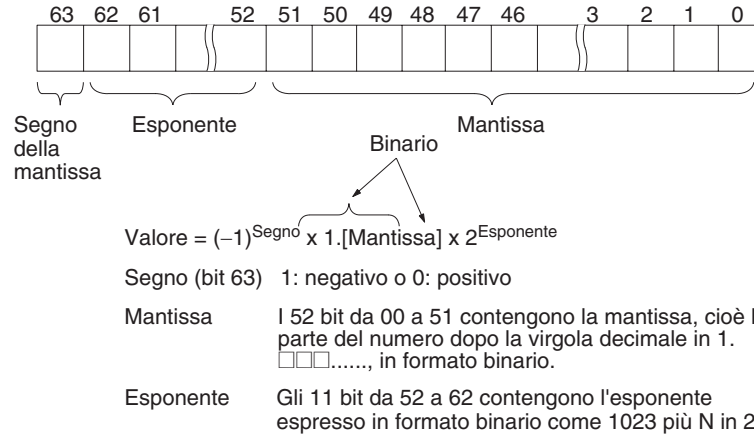
Operazione: quando l'indirizzo di memoria del PLC 000013 viene memorizzato in IR0.

La condizione di ingresso è OFF (P_Off è il flag di sempre OFF), quindi l'istruzione SET non viene eseguita. Pertanto IR0 non viene incrementato e il valore memorizzato in IR0 resta l'indirizzo di memoria del PLC 000013.

1-1-6 Formati dei dati

Nella tabella riportata di seguito sono illustrati i formati dei dati che possono essere utilizzati con i PLC delle serie CS e CJ.

| Nome | Formato | Intervallo decimale | Intervallo esadecimale |
|-------------------------|---------|----------------------|------------------------|
| Dati binari senza segno | | Da 0 a 65.535 | Da 0000 a FFFF |
| Dati binari con segno | | Da -32.768 a +32.767 | Da 8000 a 7FFF |
| Dati BCD | | Da 0 a 9.999 | Da 0000 a 9999 |

| Nome | Formato | Intervallo decimale | Intervallo esadecimale |
|--|---|---------------------|------------------------|
| Decimale a virgola mobile |  <p>Segno della mantissa</p> <p>Esponente</p> <p>Binario</p> <p>Mantissa</p> <p>Valore = $(-1)^{\text{Segno}} \times 1.[\text{Mantissa}] \times 2^{\text{Esponente}}$</p> <p>Segno (bit 31) 1: negativo o 0: positivo</p> <p>Mantissa La mantissa include 23 bit, dal bit 00 al bit 22, e indica la parte del numero dopo la virgola decimale in 1.□□□..... in formato binario.</p> <p>Esponente L'esponente include 8 bit, dal bit 23 al bit 30, e indica N più 127 in 2ⁿ in formato binario.</p> <p>Nota Questo formato dei dati a virgola mobile in singola precisione è conforme agli standard IEEE754 e viene utilizzato solo con istruzioni che convertono o utilizzano numeri a virgola mobile. Questo formato può essere utilizzato per eseguire impostazioni o monitoraggi dalla pagina di modifica e monitoraggio della memoria I/O di CX-Programmer, ma non è supportato dalle console di programmazione. Gli utenti non devono necessariamente conoscere i dettagli relativi a questo formato, eccetto per il fatto che occupa 2 canali.</p> | --- | --- |
| Decimale a virgola mobile in doppia precisione |  <p>Segno della mantissa</p> <p>Esponente</p> <p>Binario</p> <p>Mantissa</p> <p>Valore = $(-1)^{\text{Segno}} \times 1.[\text{Mantissa}] \times 2^{\text{Esponente}}$</p> <p>Segno (bit 63) 1: negativo o 0: positivo</p> <p>Mantissa I 52 bit da 00 a 51 contengono la mantissa, cioè la parte del numero dopo la virgola decimale in 1.□□□....., in formato binario.</p> <p>Esponente Gli 11 bit da 52 a 62 contengono l'esponente espresso in formato binario come 1023 più N in 2ⁿ.</p> <p>Nota Questo formato dei dati a virgola mobile in doppia precisione è conforme agli standard IEEE754 e viene utilizzato solo con istruzioni che convertono o utilizzano numeri a virgola mobile. Questo formato può essere utilizzato per eseguire impostazioni o monitoraggi dalla pagina di modifica e monitoraggio della memoria I/O di CX-Programmer, ma non è supportato dalle console di programmazione. Gli utenti non devono necessariamente conoscere i dettagli relativi a questo formato, eccetto per il fatto che occupa 4 canali.</p> | --- | --- |

Numeri binari con segno

I numeri negativi con segno vengono espressi come complemento a 2 del valore esadecimale assoluto. Per un valore decimale di -12.345, il valore assoluto è equivalente al valore 3039 esadecimale. Il complemento a 2 è 10000 - 3039 (entrambi esadecimali) o CFC7.

Per convertire un numero binario con segno negativo (CFC7) in decimale, è necessario calcolare il complemento a 2 di tale numero (10000 - CFC7 = 3039), convertirlo in decimale (3039 esadecimale = 12.345 decimale), quindi aggiungere un segno meno (-12.345).

1-2 Controlli dell'esecuzione di istruzioni

1-2-1 Errori che possono verificarsi durante l'esecuzione di un'istruzione

Quando un'istruzione viene immessa da un dispositivo periferico o quando viene eseguito un controllo del programma da un dispositivo periferico diverso da una console di programmazione, vengono controllati gli operandi e la posizione di un'istruzione. Tuttavia, non si tratta di controlli finali. Quando un'istruzione viene eseguita, possono verificarsi i quattro errori seguenti.

Errore di elaborazione istruzione (flag ER in stato ON)

Normalmente gli errori di elaborazione delle istruzioni non sono errori irreversibili, ma è possibile configurare il PLC in modo che tali errori vengano considerati fatali. Con questa impostazione, verrà attivato (ON) il flag di errore di elaborazione istruzione (A29508) e l'esecuzione del programma verrà interrotta quando si verifica un errore nell'elaborazione di un'istruzione.

Errore di accesso (flag AER ON)

Normalmente gli errori di accesso non sono errori irreversibili, ma è possibile configurare il PLC in modo che tali errori vengano considerati fatali. Con questa impostazione, verranno attivati (ON) il flag di errore di accesso illegale (A29510) e il flag di errore BCD DM/EM indiretto (A29509) e l'esecuzione del programma verrà interrotta quando si verifica un errore di accesso.

Errore di istruzione illegale

Verrà attivato (ON) il flag di errore di istruzione illegale (A29514) e l'esecuzione del programma verrà interrotta quando si verifica questo errore.

Errore di overflow UM (memoria del programma utente)

Verrà attivato (ON) il flag di errore di overflow UM (A29515) e l'esecuzione del programma verrà interrotta quando si verifica questo errore.

1-2-2 Errori irreversibili (errori di programma)

L'esecuzione del programma verrà interrotta quando si verifica uno degli errori di programma riportati di seguito. Quando si è verificato un errore di programma, il numero di task del task in corso di esecuzione quando si è interrotta l'esecuzione del programma viene scritto in A294 e l'indirizzo del programma viene scritto in A298 e A299.

Per individuare ed eventualmente correggere l'errore di programma, utilizzare il contenuto di questi canali.

| Indirizzo | Descrizione |
|-------------|---|
| A294 | In questo canale viene scritto il numero del task in esecuzione al momento dell'interruzione del programma a causa di un errore di programma. I task ciclici hanno numeri di task compresi tra 0000 e 001F (task ciclici da 0 a 31). I task ad interrupt hanno numeri di task compresi tra 8000 e 80FF (task di interrupt da 0 a 255). |
| A298 e A299 | In questi canali viene scritto l'indirizzo corrente del programma al momento dell'interruzione a causa di un errore di programma. A299 contiene le cifre all'estrema sinistra dell'indirizzo del programma, mentre A298 le cifre all'estrema destra dell'indirizzo del programma. |

Tutti gli errori che provocano l'attivazione del flag di errore o del flag di errore di accesso vengono considerati come errori di programma. Nella tabella seguente sono elencati gli errori di programma. È possibile configurare il PLC in modo che l'esecuzione del programma venga interrotta quando si verifica uno di questi errori.

| Tipo di errore | Descrizione | Flag correlati |
|------------------------------------|---|---|
| Istruzione END mancante | Nel programma è assente l'istruzione END(001). | Flag di errore di END mancante (A29511) |
| Flag di errore di esecuzione task | Per un errore di esecuzione task esistono tre possibili cause: 1) Non esiste un task ciclico eseguibile. 2) Non esiste un programma allocato al task. 3) È stato generato un interrupt ma il task ad interrupt corrispondente non esiste. | Flag di errore di esecuzione task (A29512) |
| Errore di elaborazione istruzione* | La CPU ha tentato di eseguire un'istruzione ma i dati forniti nell'operando dell'istruzione non erano corretti. *Se il PLC è stato configurato in modo da considerare gli errori di istruzione come errori irreversibili (errori di programma), verrà attivato il flag di errore di elaborazione istruzione (A29508) e l'esecuzione del programma verrà interrotta. | Flag di errore (ER), Flag di errore di elaborazione istruzione (A29508) |
| Errore di accesso* | Per un errore di accesso esistono cinque possibili cause: 1) Lettura e scrittura nell'area dei parametri. 2) Scrittura in una scheda di memoria non installata. 3) Lettura/scrittura in un banco EM formattato come memoria per i file EM. 4) Scrittura in un'area di sola lettura. 5) Il contenuto di un canale DM/EM non è in formato BCD, nonostante il PLC sia impostato per l'indirizzamento indiretto nel formato BCD. *Se il PLC è stato configurato in modo da considerare gli errori di istruzione come errori irreversibili (errori di programma), verrà attivato il flag di errore di accesso illegale (A29510) e l'esecuzione del programma verrà interrotta. | Flag di errore di accesso (AER) Flag di errore di accesso illegale (A29510) |
| Errore BCD DM/EM indiretto* | Il contenuto di un canale DM/EM non è in formato BCD, nonostante il PLC sia impostato per l'indirizzamento indiretto nel formato BCD. *Se il PLC è stato configurato in modo da considerare gli errori di istruzione come errori irreversibili (errori di programma), verrà attivato il flag di errore BCD DM/EM indiretto (A29509) e l'esecuzione del programma verrà interrotta. | Flag di errore di accesso (AER), Flag di errore BCD DM/EM indiretto (A29509) |
| Errore di overflow comparativo | Durante la modifica in linea sono state ripetutamente inserite ed eliminate istruzioni differenziate (oltre 31.072 volte). | Flag di errore di overflow comparativo (A29513) |
| Errore di overflow UM | È stato superato l'ultimo indirizzo nella UM (memoria del programma utente). | Flag di errore di overflow UM (A29515) |
| Errore di istruzione illegale | Il programma contiene un'istruzione che non è possibile eseguire. | Flag di errore di istruzione illegale (A29514) |

CAPITOLO 2

Sommario delle istruzioni

In questa sezione viene fornito un sommario delle istruzioni utilizzate con i PLC delle serie CS/CJ.

| | | |
|--------|---|-----|
| 2-1 | Classificazioni in base alla funzione | 16 |
| 2-2 | Istruzioni | 24 |
| 2-2-1 | Istruzioni di ingresso sequenza | 24 |
| 2-2-2 | Istruzioni di uscita sequenza | 26 |
| 2-2-3 | Istruzioni di controllo sequenza | 29 |
| 2-2-4 | Istruzioni di temporizzatore e contatore | 33 |
| 2-2-5 | Istruzioni di confronto | 37 |
| 2-2-6 | Istruzioni di spostamento dati | 41 |
| 2-2-7 | Istruzioni di scorrimento dei dati | 44 |
| 2-2-8 | Istruzioni di incremento e decremento | 48 |
| 2-2-9 | Istruzioni matematiche con simboli | 49 |
| 2-2-10 | Istruzioni di conversione | 54 |
| 2-2-11 | Istruzioni logiche | 60 |
| 2-2-12 | Istruzioni matematiche speciali | 62 |
| 2-2-13 | Istruzioni matematiche a virgola mobile | 63 |
| 2-2-14 | Istruzioni a virgola mobile in doppia precisione | 67 |
| 2-2-15 | Istruzioni di elaborazione dei dati delle tabelle | 71 |
| 2-2-16 | Istruzioni di controllo dei dati | 75 |
| 2-2-17 | Istruzioni di subroutine | 79 |
| 2-2-18 | Istruzioni di controllo degli interrupt | 80 |
| 2-2-19 | Istruzioni per contatore veloce e uscita a impulsi (solo CJ1M-CPU21/22/23) | 82 |
| 2-2-20 | Istruzioni di step | 84 |
| 2-2-21 | Istruzioni per moduli di I/O di base | 84 |
| 2-2-22 | Istruzioni per la comunicazione seriale | 87 |
| 2-2-23 | Istruzioni di rete | 88 |
| 2-2-24 | Istruzioni per la memoria dei file | 91 |
| 2-2-25 | Istruzioni di visualizzazione | 92 |
| 2-2-26 | Istruzioni per l'orologio | 92 |
| 2-2-27 | Istruzioni di debug | 93 |
| 2-2-28 | Istruzioni di diagnostica per malfunzionamento | 94 |
| 2-2-29 | Altre istruzioni | 95 |
| 2-2-30 | Istruzioni di programmazione a blocchi | 96 |
| 2-2-31 | Istruzioni di elaborazione delle stringhe di testo | 102 |
| 2-2-32 | Istruzioni di controllo dei task | 105 |
| 2-2-33 | Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva) | 106 |
| 2-2-34 | Istruzioni speciali per blocchi funzione | 107 |
| 2-3 | Elenco alfabetico delle istruzioni in base al codice mnemonico | 108 |
| 2-4 | Elenco delle istruzioni in base al codice funzione | 125 |

2-1 Classificazioni in base alla funzione

Nella tabella seguente sono riportate le istruzioni per la serie CS/CJ, elencate per funzione. Le istruzioni sono descritte in base alla rispettiva funzione nella *Sezione 3, Istruzioni*.

*Le istruzioni e i gruppi di istruzioni contrassegnati da un asterisco sono supportati solo dai moduli CPU CS1-H, CJ1-H, CJ1M e CS1D.

**Le istruzioni e i gruppi di istruzioni contrassegnati da due asterischi sono supportati solo dai moduli CPU CJ1M.

***Le istruzioni e i gruppi di istruzioni contrassegnati da tre asterischi non sono supportati dai moduli CPU CS1D per sistemi a due CPU.

- Nota**
1. Solo CPU della serie CS/CJ versione 2.0 o successiva.
 2. CPU CJ1M-CPU21/22/23 CPU versione 2.0 o successiva.
 3. Solo CPU della serie CS/CJ versione 2.0 o successiva
CPU CJ1M (precedenti alla versione 2.0 o versione 2.0 o successiva)

| Classificazione | Sottoclasse | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|----------------------------------|--------------------|--------------------------|---|--------------------------|-----------------------|------------------|--------------------------------------|
| Istruzioni di base | Ingresso | LD | LOAD | LD NOT | LOAD NOT | AND | AND |
| | | AND NOT | AND NOT | OR | OR | OR NOT | OR NOT |
| | | AND LD | AND LOAD | OR LD | OR LOAD | --- | --- |
| | Uscita | OUT | OUTPUT | OUT NOT | OUTPUT NOT | --- | --- |
| Istruzioni di ingresso sequenza | --- | NOT | NOT | UP | CONDITION ON | DOWN | CONDITION OFF |
| | Test bit | LD TST | LD BIT TEST | LD TSTN | LD BIT TEST NOT | AND TST | AND BIT TEST NOT |
| | | AND TSTN | AND BIT TEST NOT | OR TST | OR BIT TEST | OR TSTN | OR BIT TEST NOT |
| Istruzioni di uscita sequenza | --- | KEEP | KEEP | DIFU | DIFFERENTIATE UP | DIFD | DIFFERENTIATE DOWN |
| | | OUTB* | SINGLE BIT OUTPUT | --- | --- | --- | --- |
| | Impostazione/Reset | SET | SET | RSET | RESET | SETA | MULTIPLE BIT SET |
| | | RSTA | MULTIPLE BIT RESET | SETB* | SINGLE BIT SET | RSTB* | SINGLE BIT RESET |
| Istruzioni di controllo sequenza | --- | END | END | NOP | NO OPERATION | --- | --- |
| | Interblocco | IL | INTERLOCK | ILC | INTERLOCK CLEAR | MILH | MULTI-INTERLOCK DIFFERENTIATION HOLD |
| | | MILR (vedere la nota 1). | MULTI-INTERLOCK DIFFERENTIATION RELEASE | MILC (vedere la nota 1). | MULTI-INTERLOCK CLEAR | --- | --- |
| | Salto | JMP | JUMP | JME | JUMP END | CJP | CONDITIONAL JUMP |
| | | CJPN | CONDITIONAL JUMP | JMP0 | MULTIPLE JUMP | JME0 | MULTIPLE JUMP END |
| Ripetizione | FOR | FOR-NEXT LOOPS | BREAK | BREAK LOOP | NEXT | FOR-NEXT LOOPS | |

| Classificazione | Sottoclasse | | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|--|--|---|--|--|--|---|-----------------------------------|---------------------|
| Istruzioni di temporizzatore e contatore | BCD | Temporizzatore (con numeri di temporizzatore) | TIM | TIMER | TIMH | HIGH-SPEED TIMER | TMHH | ONE-MS TIMER |
| | | | TTIM | ACCUMULATIVE TIMER | --- | --- | --- | --- |
| | | Temporizzatore (senza numeri di temporizzatore) | TIML | LONG TIMER | MTIM | MULTI-OUTPUT TIMER | --- | --- |
| | | | CNT | COUNTER | CNTR | REVERSIBLE TIMER | CNR | RESET TIMER/COUNTER |
| | Binario* | Temporizzatore (con numeri di temporizzatore) | TIMX | TIMER | TIMHX | HIGH-SPEED TIMER | TMHHX | ONE-MS TIMER |
| | | | TTIMX | ACCUMULATIVE TIMER | --- | --- | --- | --- |
| | | Temporizzatore (senza numeri di temporizzatore) | TIMLX | LONG TIMER | MTIMX | MULTI-OUTPUT TIMER | --- | --- |
| | | | CNTX | COUNTER | CNTRX | REVERSIBLE TIMER | CNRX | RESET TIMER/COUNTER |
| Istruzioni di confronto | Confronto con simboli | LD, AND, OR + =, <>, <, <=, >, >= | Confronto con simboli (senza segno) | LD, AND, OR + =, <>, <, <=, >, >= + L | Confronto con simboli (doppio canale, senza segno) | LD, AND, OR + =, <>, <, <=, >, >= +S | Confronto con simboli (con segno) | |
| | | LD, AND, OR + =, <>, <, <=, >, >= + SL | Confronto con simboli (doppio canale, con segno) | LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT (vedere la nota 1). | Confronto di date e ore | --- | --- | |
| | Confronto di dati (flag di condizione) | CMP | UNSIGNED COMPARE | CMPL | DOUBLE UNSIGNED COMPARE | CPS | SIGNED BINARY COMPARE | |
| | | CPSL | DOUBLE SIGNED BINARY COMPARE | ZCP* | AREA RANGE COMPARE | ZCPL* | DOUBLE AREA RANGE COMPARE | |
| | Confronto di tabelle | MCMP | MULTIPLE COMPARE | TCMP | TABLE COMPARE | BCMP | UNSIGNED BLOCK COMPARE | |
| | | BCMP2 (vedere la nota 3). | EXPANDED BLOCK COMPARE | --- | --- | --- | --- | |

| Classificazione | Sottoclasse | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|-------------------------------------|----------------------------|---------------------------|----------------------------------|-----------------------------------|-----------------------------------|------------------|---------------------------|
| Istruzioni di spostamento dei dati | Singolo/doppio canale | MOV | MOVE | MOVL | DOUBLE MOVE | MVN | MOVE NOT |
| | | MVNL | DOUBLE MOVE NOT | --- | --- | --- | --- |
| | Bit/cifra | MOVB | MOVE BIT | MOVD | MOVE DIGIT | --- | --- |
| | Scambio | XCHG | DATA EXCHANGE | XCGL | DOUBLE DATA EXCHANGE | --- | --- |
| | Trasferimento blocco/bit | XFRB | MULTIPLE BIT TRANSFER | XFER | BLOCK TRANSFER | BSET | BLOCK SET |
| | Distribuzione/raccolta | DIST | SINGLE WORD DISTRIBUTE | COLL | DATA COLLECT | --- | --- |
| Registro indice | MOVR | MOVE TO REGISTER | MOVW | MOVE TIMER/COUNTER PV TO REGISTER | --- | --- | |
| Istruzioni di scorrimento dei dati | Scorrimento di 1 bit | SFT | SHIFT REGISTER | SFTR | REVERSIBLE SHIFT REGISTER | ASLL | DOUBLE SHIFT LEFT |
| | | ASL | ARITHMETIC SHIFT LEFT | ASR | ARITHMETIC SHIFT RIGHT | ASRL | DOUBLE SHIFT RIGHT |
| | 0000 esadecimale asincrono | ASFT | ASYNCHRONOUS SHIFT REGISTER | --- | --- | --- | --- |
| | Scorrimento del canale | WSFT | WORD SHIFT | --- | --- | --- | --- |
| | Rotazione di 1 bit | ROL | ROTATE LEFT | ROLL | DOUBLE ROTATE LEFT | RLNC | ROTATE LEFT WITHOUT CARRY |
| | | RLNL | DOUBLE ROTATE LEFT WITHOUT CARRY | ROR | ROTATE RIGHT | RORL | DOUBLE ROTATE RIGHT |
| | | RRNC | ROTATE RIGHT WITHOUT CARRY | RRNL | DOUBLE ROTATE RIGHT WITHOUT CARRY | --- | --- |
| | Scorrimento di 1 cifra | SLD | ONE DIGIT SHIFT LEFT | SRD | ONE DIGIT SHIFT RIGHT | --- | --- |
| | Scorrimento dati di n bit | NSFL | SHIFT N-BIT DATA LEFT | NSFR | SHIFT N-BIT DATA RIGHT | --- | --- |
| | Scorrimento di n bit | NASL | SHIFT N-BITS LEFT | NSLL | DOUBLE SHIFT N-BITS LEFT | NASR | SHIFT N-BITS RIGHT |
| NSRL | | DOUBLE SHIFT N-BITS RIGHT | --- | --- | --- | --- | |
| Istruzioni di incremento/decremento | BCD | ++B | INCREMENT BCD | ++BL | DOUBLE INCREMENT BCD | --B | DECREMENT BCD |
| | | --BL | DOUBLE DECREMENT BCD | --- | --- | --- | --- |
| | Binario | ++ | INCREMENT BINARY | ++L | DOUBLE INCREMENT BINARY | -- | DECREMENT BINARY |
| | | --L | DOUBLE DECREMENT BINARY | --- | --- | --- | --- |

| Classificazione | Sottoclasse | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|------------------------------------|-------------------------|------------------|--------------------------------------|------------------|---|------------------|-----------------------------------|
| Istruzioni matematiche con simboli | Addizione binario | + | SIGNED BINARY ADD WITHOUT CARRY | +L | DOUBLE SIGNED BINARY ADD WITHOUT CARRY | +C | SIGNED BINARY ADD WITH CARRY |
| | | +CL | DOUBLE SIGNED BINARY ADD WITH CARRY | --- | --- | --- | --- |
| | Addizione BCD | +B | BCD ADD WITHOUT CARRY | +BL | DOUBLE BCD ADD WITHOUT CARRY | +BC | BCD ADD WITH CARRY |
| | | +BCL | DOUBLE BCD ADD WITH CARRY | --- | --- | --- | --- |
| | Sottrazione binario | - | SIGNED BINARY SUBTRACT WITHOUT CARRY | -L | DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY | -C | SIGNED BINARY SUBTRACT WITH CARRY |
| | | -CL | DOUBLE SIGNED BINARY WITH CARRY | --- | --- | --- | --- |
| | Sottrazione BCD | -B | BCD SUBTRACT WITHOUT CARRY | -BL | DOUBLE BCD SUBTRACT WITHOUT CARRY | -BC | BCD SUBTRACT WITH CARRY |
| | | -BCL | DOUBLE BCD SUBTRACT WITH CARRY | --- | --- | --- | --- |
| | Moltiplicazione binario | * | SIGNED BINARY MULTIPLY | *L | DOUBLE SIGNED BINARY MULTIPLY | *U | UNSIGNED BINARY MULTIPLY |
| | | *UL | DOUBLE UNSIGNED BINARY MULTIPLY | --- | --- | --- | --- |
| | Moltiplicazione BCD | *B | BCD MULTIPLY | *BL | DOUBLE BCD MULTIPLY | --- | --- |
| | Divisione binario | / | SIGNED BINARY DIVIDE | /L | DOUBLE SIGNED BINARY DIVIDE | /U | UNSIGNED BINARY DIVIDE |
| | | /UL | DOUBLE UNSIGNED BINARY DIVIDE | --- | --- | --- | --- |
| | Divisione BCD | /B | BCD DIVIDE | /BL | DOUBLE BCD DIVIDE | --- | --- |

| Classificazione | Sottoclasse | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|---|---|---------------------------------------|--|----------------------|-----------------------------|------------------|-------------------------|
| Istruzioni di conversione | Conversione BCD/binario | BIN | BCD-TO-BINARY | BINL | DOUBLE BCD-TO-DOUBLE BINARY | BCD | BINARY-TO-BCD |
| | | BCDL | DOUBLE BINARY-TO-DOUBLE BCD | NEG | 2'S COMPLEMENT | NEGL | DOUBLE 2'S COMPLEMENT |
| | | SIGN | 16-BIT TO 32-BIT SIGNED BINARY | --- | --- | --- | --- |
| | Decoder/encoder | MLPX | DATA DECODER | DMPX | DATA ENCODER | --- | --- |
| | Conversione ASCII/esadecimale | ASC | ASCII CONVERT | HEX | ASCII TO HEX | --- | --- |
| | Conversione riga/colonna | LINE | COLUMN TO LINE | COLM | LINE TO COLUMN | --- | --- |
| | Conversione BCD-binario con segno | BINS | SIGNED BCD-TO-BINARY | BISL | DOUBLE SIGNED BCD-TO-BINARY | BCDS | SIGNED BINARY-TO-BCD |
| BDSL | | DOUBLE SIGNED BINARY-TO-BCD | GRY (vedere la nota 1). | GRAY CODE CONVERSION | --- | --- | |
| Istruzioni logiche | Istruzioni logiche AND/OR | ANDW | LOGICAL AND | ANDL | DOUBLE LOGICAL AND | ORW | LOGICAL OR |
| | | ORWL | DOUBLE LOGICAL OR | XORW | EXCLUSIVE OR | XORL | DOUBLE EXCLUSIVE OR |
| | | XNRW | EXCLUSIVE NOR | XNRL | DOUBLE EXCLUSIVE NOR | --- | --- |
| | Complemento | COM | COMPLEMENT | COML | DOUBLE COMPLEMENT | --- | --- |
| Istruzioni matematiche speciali | --- | ROTB | BINARY ROOT | ROOT | BCD SQUARE ROOT | APR | ARITHMETIC PROCESS |
| | | FDIV | FLOATING POINT DIVIDE | BCNT | BIT COUNTER | --- | --- |
| Istruzioni matematiche a virgola mobile | Conversione virgola mobile/binario | FIX | FLOATING TO 16-BIT | FIXL | FLOATING TO 32-BIT | FLT | 16-BIT TO FLOATING |
| | | FLTL | 32-BIT TO FLOATING | --- | --- | --- | --- |
| | Istruzioni matematiche di base a virgola mobile | +F | FLOATING-POINT ADD | -F | FLOATING-POINT SUBTRACT | /F | FLOATING POINT DIVIDE |
| | | *F | FLOATING-POINT MULTIPLY | --- | --- | --- | --- |
| | Istruzioni trigonometriche a virgola mobile | RAD | DEGREES TO RADIANS | DEG | RADIANS TO DEGREES | SIN | SINE |
| | | COS | COSINE | TAN | TANGENT | ASIN | ARC SINE |
| | | ACOS | ARC COSINE | ATAN | ARC TANGENT | --- | --- |
| | Istruzioni matematiche a virgola mobile | SQRT | SQUARE ROOT | EXP | EXPONENT | LOG | LOGARITHM |
| | | PWR | EXPONENTIAL POWER | --- | --- | --- | --- |
| | Confronto con simboli e conversione* | LD, AND, OR + =, <>, <, <=, >, >= + F | Confronto con simboli (virgola mobile in singola precisione) | FSTR* | FLOATING-POINT TO ASCII | FVAL* | ASCII TO FLOATING-POINT |

| Classificazione | Sottoclasse | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|---|---|--|---|------------------|--------------------------------|-------------------------|------------------------------|
| Istruzioni a virgola mobile in doppia precisione* | Conversione virgola mobile/binario | FIXD | DOUBLE FLOATING TO 16-BIT | FIXLD | DOUBLE FLOATING TO 32-BIT | DBL | 16-BIT TO DOUBLE FLOATING |
| | | DBLL | 32-BIT TO DOUBLE FLOATING | --- | --- | --- | --- |
| | Istruzioni matematiche di base a virgola mobile | +D | DOUBLE FLOATING-POINT ADD | -D | DOUBLE FLOATING-POINT SUBTRACT | /D | DOUBLE FLOATING-POINT DIVIDE |
| | | *D | DOUBLE FLOATING-POINT MULTIPLY | --- | --- | --- | --- |
| | Istruzioni trigonometriche a virgola mobile | RADD | DOUBLE DEGREES TO RADIANS | DEGD | DOUBLE RADIANS TO DEGREES | SIND | DOUBLE SINE |
| | | COSD | DOUBLE COSINE | TAND | DOUBLE TANGENT | ASIND | DOUBLE ARC SINE |
| | | ACOSD | DOUBLE ARC COSINE | ATAND | DOUBLE ARC TANGENT | --- | --- |
| | Istruzioni matematiche a virgola mobile | SQRTD | DOUBLE SQUARE ROOT | EXPD | DOUBLE EXPONENT | LOGD | DOUBLE LOGARITHM |
| | | PWRD | DOUBLE EXPONENTIAL POWER | --- | --- | --- | --- |
| | Confronto con simboli | LD, AND, OR + =, <>, <, <=, >, >= + D | Confronto con simboli (virgola mobile in doppia precisione) | --- | --- | --- | --- |
| | Istruzioni di elaborazione dei dati di tabelle | Elaborazione di stack | SSET | SET STACK | PUSH | PUSH ONTO STACK | LIFO |
| FIFO | | | FIRST IN FIRST OUT | SNUM* | STACK SIZE READ | SREAD* | STACK DATA READ |
| SWRIT* | | | STACK DATA OVERWRITE | SINS* | STACK DATA INSERT | SDEL* | STACK DATA DELETE |
| Elaborazione di 1 record/più canali | | DIM | DIMENSION RECORD TABLE | SETR | SET RECORD LOCATION | GETR | GET RECORD NUMBER |
| Elaborazione record-canale | | SRCH | DATA SEARCH | MAX | FIND MAXIMUM | MIN | FIND MINIMUM |
| | | SUM | SUM | FCS | FRAME CHECKSUM | --- | --- |
| Elaborazione di byte | | SWAP | SWAP BYTES | --- | --- | --- | --- |
| Istruzioni di controllo dati | --- | PID | PID CONTROL | PIDAT* | PID CONTROL WITH AUTOTUNING | LMT | LIMIT CONTROL |
| | | BAND | DEAD BAND CONTROL | ZONE | DEAD ZONE CONTROL | TPO (vedere la nota 1). | TIME-PROPORTIONAL OUTPUT |
| | | SCL | SCALING | SCL2 | SCALING 2 | SCL3 | SCALING 3 |
| | | AVG | AVERAGE | --- | --- | --- | --- |
| Istruzioni di subroutine | --- | SBS | SUBROUTINE CALL | MCRO | MACRO | SBN | SUBROUTINE ENTRY |
| | | RET | SUBROUTINE RETURN | GSBS* | GLOBAL SUBROUTINE CALL | GSBN* | GLOBAL SUBROUTINE ENTRY |
| | | GRET* | GLOBAL SUBROUTINE RETURN | --- | --- | --- | --- |

| Classificazione | Sottoclasse | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|---|-------------|---------------------------|---------------------------------|---------------------------|----------------------------|---------------------------|---------------------------|
| Istruzioni di controllo interrupt | --- | MSKS*** | SET INTERRUPT MASK | MSKR*** | READ INTERRUPT MASK | CLI*** | CLEAR INTERRUPT |
| | | DI | DISABLE INTERRUPTS | EI | ENABLE INTERRUPTS | --- | --- |
| Istruzioni per contatore veloce/uscita a impulsi** | --- | INI | MODE CONTROL | PRV | HIGH-SPEED COUNTER PV READ | PRV2 (vedere la nota 2). | COUNTER FREQUENCY CONVERT |
| | | CTBL | COMPARISON TABLE LOAD | SPED | SPEED OUTPUT | PULS | SET PULSES |
| | | PLS2 | PULSE OUTPUT | ACC | ACCELERATION CONTROL | ORG | ORIGIN SEARCH |
| Istruzioni di step | --- | PWM | PULSE WITH VARIABLE DUTY FACTOR | STEP | STEP DEFINE | SNXT | STEP START |
| Istruzioni per moduli di I/O di base | --- | IORF | I/O REFRESH | SDEC | 7-SEGMENT DECODER | DSW (vedere la nota 1). | DIGITAL SWITCH INPUT |
| | | TKY (vedere la nota 1). | TEN KEY INPUT | HKY (vedere la nota 1). | HEXADECIMAL KEY INPUT | MTR (vedere la nota 1). | MATRIX INPUT |
| | | 7SEG (vedere la nota 1). | 7-SEGMENT DISPLAY OUTPUT | IORF | INTELLIGENT I/O READ | IOWR | INTELLIGENT I/O WRITE |
| | | DLNK* | CPU BUS UNIT I/O REFRESH | --- | --- | --- | --- |
| Istruzioni per la comunicazione seriale | --- | PMCR | PROTOCOL MACRO | TXD | TRANSMIT | RXD | RECEIVE |
| | | STUP | CHANGE SERIAL PORT SETUP | --- | --- | --- | --- |
| Istruzioni di rete | --- | SEND | NETWORK SEND | RECV | NETWORK RECEIVE | CMND | DELIVER COMMAND |
| | | EXPLT (vedere la nota 1). | SEND GENERAL EXPLICIT | EGATR (vedere la nota 1). | EXPLICIT GET ATTRIBUTE | ESATR (vedere la nota 1). | EXPLICIT SET ATTRIBUTE |
| | | ECHRD (vedere la nota 1). | EXPLICIT WORD READ | ECHWR (vedere la nota 1). | EXPLICIT WORD WRITE | --- | --- |
| Istruzioni di visualizzazione | --- | MSG | DISPLAY MESSAGE | --- | --- | --- | --- |
| Istruzioni di memoria file | --- | FREAD | READ DATA FILE | FWRIT | WRITE DATA FILE | --- | --- |
| Istruzioni per l'orologio | --- | CADD | CALENDAR ADD | CSUB | CALENDAR SUBTRACT | SEC | HOURS TO SECONDS |
| | | HMS | SECONDS TO HOURS | DATE | CLOCK ADJUSTMENT | --- | --- |
| Istruzioni di debug | --- | TRSM | TRACE MEMORY SAMPLING | --- | --- | --- | --- |
| Istruzioni per la diagnostica di funzionamento incorretto | --- | FAL | FAILURE ALARM | FALS | SEVERE FAILURE ALARM | FPD | FAILURE POINT DETECTION |

| Classificazione | Sottoclasse | | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione | Codice mnemonico | Istruzione |
|--|--|----------|---|---------------------------------|------------------------|-----------------------------------|--------------------------|------------------------------------|
| Altre istruzioni | --- | | STC | SET CARRY | CLC | CLEAR CARRY | EMBC | SELECT EM BANK |
| | --- | | WDT | EXTEND MAXIMUM CYCLE TIME | CCS* | SAVE CONDITION FLAGS | CCL* | LOAD CONDITION FLAGS |
| | --- | | FRMCV* | CONVERT ADDRESS FROM CV | TOCV* | CONVERT ADDRESS TO CV | IOSP*** | DISABLE PERIPHERAL SERVICING |
| | --- | | IORS*** | ENABLE PERIPHERAL SERVICING | --- | --- | --- | --- |
| Istruzioni di programmazione a blocchi | Definizione di un'area di programmazione a blocchi | | BPRG | BLOCK PROGRAM BEGIN | BEND | BLOCK PROGRAM END | --- | --- |
| | Avvio/arresto del programma a blocchi | | BPPS | BLOCK PROGRAM PAUSE | BPRS | BLOCK PROGRAM RESTART | --- | --- |
| | EXIT | | EXIT indirizzo_bit | END condizionale | EXIT NOT indirizzo_bit | END NOT condizionale | EXIT condizione_ingresso | END condizionale |
| | Elaborazione condizionata | | IF indirizzo_bit | CONDITIONAL BLOCK BRANCHING | IF NOT indirizzo_bit | CONDITIONAL BLOCK BRANCHING (NOT) | ELSE | CONDITIONAL BLOCK BRANCHING (ELSE) |
| | | | IEND | CONDITIONAL BLOCK BRANCHING END | --- | --- | --- | --- |
| | WAIT | | WAIT indirizzo_bit | ONE CYCLE AND WAIT | WAIT NOT indirizzo_bit | ONE CYCLE AND WAIT NOT | WAIT condizione_ingresso | ONE CYCLE AND WAIT |
| | Temporizzatore/contatore | BCD | TIMW | TIMER WAIT | CNTW | COUNTER WAIT | TMHW | HIGH-SPEED TIMER WAIT |
| | | Binario* | TIMWX | TIMER WAIT | CNTWX | COUNTER WAIT | TMHWX | HIGH-SPEED TIMER WAIT |
| | Ripetizione | | LOOP | LOOP BLOCK | LEND indirizzo_bit | LOOP BLOCK END | LEND NOT indirizzo_bit | LOOP BLOCK END NOT |
| | | | LEND condizione_ingresso | LOOP BLOCK END | --- | --- | --- | --- |
| Istruzioni di elaborazione stringhe di testo | --- | | MOV\$ | MOV STRING | +\$ | CONCATE-NATE STRING | LEFT\$ | GET STRING LEFT |
| | --- | | RIGHT\$ | GET STRING RIGHT | MID\$ | GET STRING MIDDLE | FIND\$ | FIND IN STRING |
| | --- | | LEN\$ | STRING LENGTH | RPLC\$ | REPLACE IN STRING | DEL\$ | DELETE STRING |
| | --- | | XCHG\$ | EXCHANGE STRING | CLR\$ | CLEAR STRING | INS\$ | INSERT INTO STRING |
| | --- | | LD, AND, OR + =\$, <>\$, <\$, <=\$, >\$, >=\$ | STRING COMPARISON | --- | --- | --- | --- |
| Istruzioni di controllo dei task | --- | | TKON | TASK ON | TKOF | TASK OFF | --- | --- |

2-2 Istruzioni

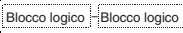
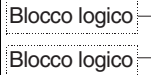
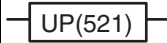

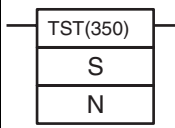
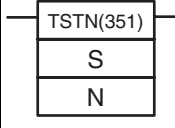
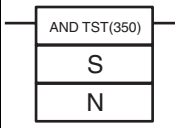
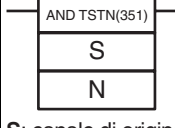
2-2-1 Istruzioni di ingresso sequenza

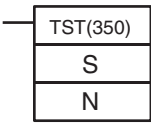
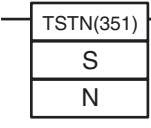
*1: non supportata dalle CPU CS1D per sistemi a due CPU.

*2: supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

*3: supportata solo dalle CPU CS1-H, CJ1-H e CJ1M.



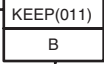
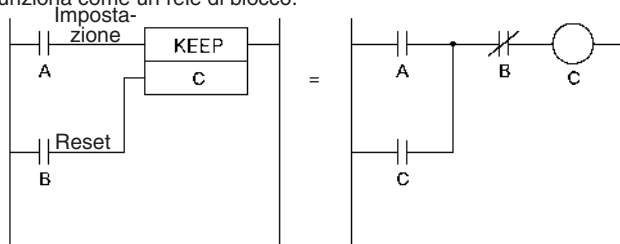
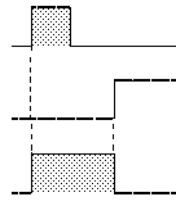
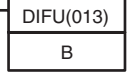
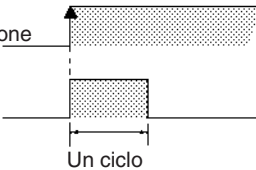
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|---|--|-------------|
| LOAD LD @LD %LD !LD* ¹ !@LD* ¹ !%LD* ¹ | <p>Barra di distribuzione</p> <p>Punto iniziale del blocco</p> | Indica un inizio logico e crea una condizione di esecuzione ON/OFF basata sullo stato ON/OFF del bit operando specificato. | Inizio della logica Non richiesta | 153 |
| LOAD NOT LD NOT @LD NOT* ² %LD NOT* ² !LD NOT* ¹ !@LD NOT* ³ !%LD NOT* ³ | <p>Barra di distribuzione</p> <p>Punto iniziale del blocco</p> | Indica un inizio logico e crea una condizione di esecuzione ON/OFF basata sull'inverso dello stato ON/OFF del bit operando specificato. | Inizio della logica Non richiesta | 155 |
| AND AND @AND %AND !AND* ¹ !@AND* ¹ !%AND* ¹ | | Esegue un AND logico dello stato del bit operando specificato e della condizione di esecuzione corrente. | Continua su rung Richiesta | 157 |
| AND NOT AND NOT @AND NOT* ² %AND NOT* ² !AND NOT* ¹ !@AND NOT* ³ !%AND NOT* ³ | | Inverte lo stato del bit operando specificato ed esegue un AND logico con la condizione di esecuzione corrente. | Continua su rung Richiesta | 159 |
| OR OR @OR %OR !OR* ¹ !@OR* ¹ !%OR* ¹ | <p>Barra di distribuzione</p> | Esegue un OR logico dello stato ON/OFF del bit operando specificato e della condizione di esecuzione corrente. | Continua su rung Richiesta | 161 |
| OR NOT OR NOT @OR NOT* ² %OR NOT* ² !OR NOT* ¹ !@OR NOT* ³ !%OR NOT* ³ | <p>Barra di distribuzione</p> | Inverte lo stato del bit specificato ed esegue un OR logico con la condizione di esecuzione corrente. | Continua su rung Richiesta | 163 |

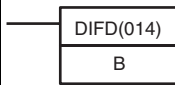


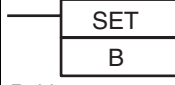
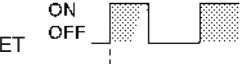
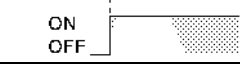
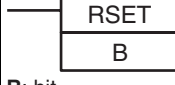
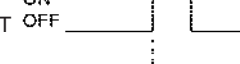
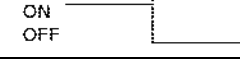
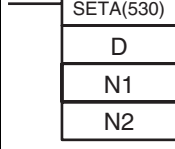
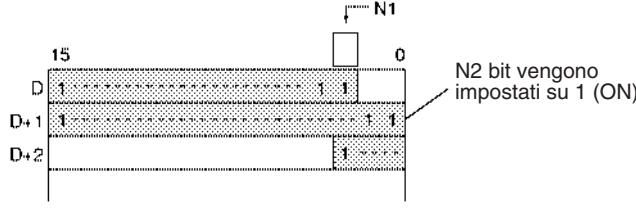
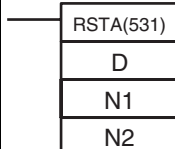
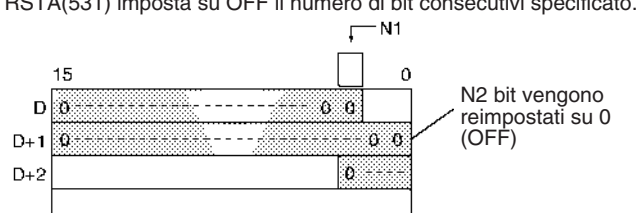
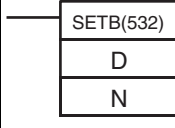
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|-------------------------------------|---|--|--|-------------|
| AND LOAD AND LD |  | Assume un AND logico tra blocchi logici. LD } Blocco logico A ... } LD } Blocco logico B ... } AND LD Collegamento seriale tra i blocchi logici A e B. | Continua su rung Richiesta | 164 |
| OR LOAD OR LD |  | Assume un OR logico tra blocchi logici. LD } Blocco logico A ... } LD } Blocco logico B ... } OR LD Collegamento parallelo tra i blocchi logici A e B. | Continua su rung Richiesta | 166 |
| NOT NOT 520 | --- | Inverte la condizione di esecuzione. | Continua su rung Richiesta | 172 |
| CONDITION ON UP 521 |  | UP(521) attiva la condizione di esecuzione per un ciclo quando lo stato di quest'ultima passa da OFF a ON. | Continua su rung Richiesta | 173 |
| CONDITION OFF DOWN 522 |  | DOWN(522) attiva la condizione di esecuzione per un ciclo quando lo stato di quest'ultima passa da ON a OFF. | Continua su rung Richiesta | 173 |
| BIT TEST LD TST 350 |  S: canale di origine N: numero di bit | LD TST(350), AND TST(350) e OR TST(350) vengono utilizzate nel programma analogamente a LD, AND e OR. La condizione di esecuzione viene attivata quando il bit specificato nel canale specificato viene impostato su ON e disattivata quando il bit viene impostato su OFF. | Continua su rung Non richiesta | 174 |
| BIT TEST LD TSTN 351 |  S: canale di origine N: numero di bit | LD TSTN(351), AND TSTN(351) e OR TSTN(351) vengono utilizzate nel programma analogamente a LD NOT, AND NOT e OR NOT. La condizione di esecuzione viene disattivata quando il bit specificato nel canale specificato viene impostato su ON e attivata quando il bit viene impostato su OFF. | Continua su rung Non richiesta | 174 |
| BIT TEST AND TST 350 |  S: canale di origine N: numero di bit | LD TST(350), AND TST(350) e OR TST(350) vengono utilizzate nel programma analogamente a LD, AND e OR. La condizione di esecuzione viene attivata quando il bit specificato nel canale specificato viene impostato su ON e disattivata quando il bit viene impostato su OFF. | Continua su rung Richiesta | 174 |
| BIT TEST AND TSTN 351 |  S: canale di origine N: numero di bit | LD TSTN(351), AND TSTN(351) e OR TSTN(351) vengono utilizzate nel programma analogamente a LD NOT, AND NOT e OR NOT. La condizione di esecuzione viene disattivata quando il bit specificato nel canale specificato viene impostato su ON e attivata quando il bit viene impostato su OFF. | Continua su rung Richiesta | 174 |

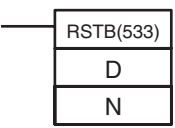
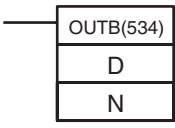
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|-----------------------------------|---|--|--|-------------|
| BIT TEST OR TST 350 |  S: canale di origine N: numero di bit | LD TST(350), AND TST(350) e OR TST(350) vengono utilizzate nel programma analogamente a LD, AND e OR. La condizione di esecuzione viene attivata quando il bit specificato nel canale specificato viene impostato su ON e disattivata quando il bit viene impostato su OFF. | Continua su rung Richiesta | 174 |
| BIT TEST OR TSTN 351 |  S: canale di origine N: numero di bit | LD TSTN(351), AND TSTN(351) e OR TSTN(351) vengono utilizzate nel programma analogamente a LD NOT, AND NOT e OR NOT. La condizione di esecuzione viene disattivata quando il bit specificato nel canale specificato viene impostato su ON e attivata quando il bit viene impostato su OFF. | Continua su rung Richiesta | 174 |

2-2-2 Istruzioni di uscita sequenza

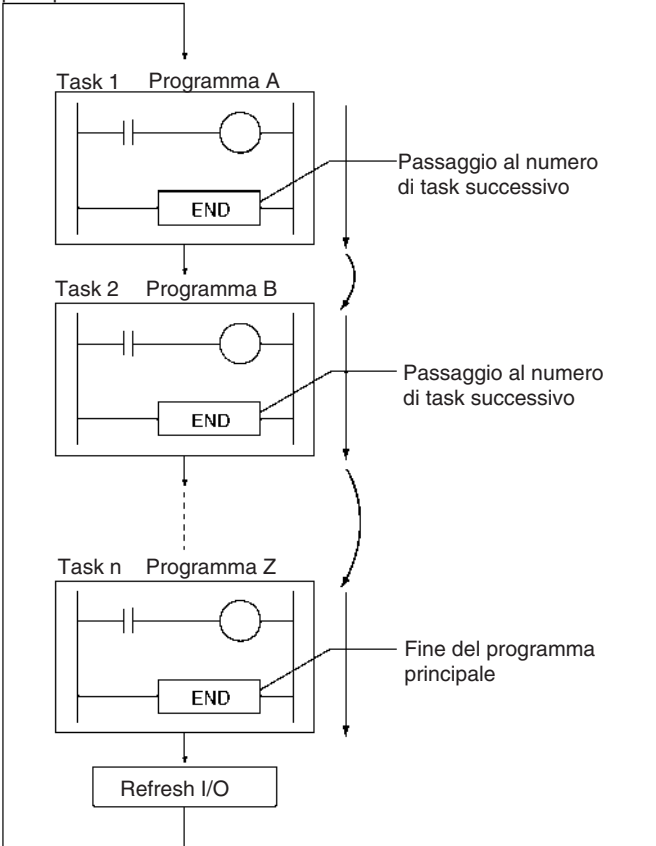
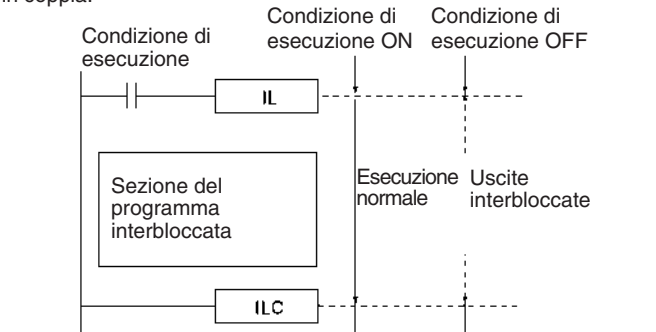
*1: non supportata dalle CPU CS1D per sistemi a due CPU.

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|--|--|-------------|
| OUTPUT OUT !OUT*1 |  | Invia in uscita il risultato (condizione di esecuzione) dell'elaborazione logica al bit specificato. | Uscita Richiesta | 177 |
| OUTPUT NOT OUT NOT !OUT NOT*1 |  | Inverte il risultato (condizione di esecuzione) dell'elaborazione logica e lo invia in uscita al bit specificato. | Uscita Richiesta | 178 |
| KEEP KEEP !KEEP*1 011 | S (impostazione)  R (reset) B: bit | Funziona come un relè di blocco.  Condizione di esecuzione S Condizione di esecuzione R Stato di B  | Uscita Richiesta | 180 |
| DIFFERENTIATE UP DIFU !DIFU*1 013 |  B: bit | DIFU(013) imposta su ON il bit specificato per un ciclo quando la condizione di esecuzione passa da OFF a ON (fronte di salita). Condizione di esecuzione Stato di B  Un ciclo | Uscita Richiesta | 184 |

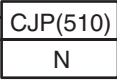
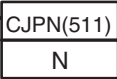

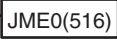
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| DIFFERENTIATE DOWN DIFD !DIFD ^{*1} 014 |  B: bit | DIFD(014) imposta su ON il bit specificato per un ciclo quando la condizione di esecuzione passa da ON a OFF (fronte di discesa). Condizione di esecuzione  Stato di B  Un ciclo | Uscita Richiesta | 184 |
| SET SET @SET %SET !SET ^{*1} !@SET ^{*1} !%SET ^{*1} |  B: bit | SET imposta il bit dell'operando su ON quando la condizione di esecuzione è ON. Condizione di esecuzione di SET  Stato di B  | Uscita Richiesta | 187 |
| RESET RSET @RSET %RSET !RSET ^{*1} !@RSET ^{*1} !%RSET ^{*1} |  B: bit | RSET imposta il bit dell'operando su OFF quando la condizione di esecuzione è ON. Condizione di esecuzione di RSET  Stato di B  | Uscita Richiesta | 187 |
| MULTIPLE BIT SET SETA @SETA 530 |  D: canale iniziale N1: bit iniziale N2: numero di bit | SETA(530) imposta su ON il numero di bit consecutivi specificato.  N2 bit vengono impostati su 1 (ON) | Uscita Richiesta | 189 |
| MULTIPLE BIT RESET RSTA @RSTA 531 |  D: canale iniziale N1: bit iniziale N2: numero di bit | RSTA(531) imposta su OFF il numero di bit consecutivi specificato.  N2 bit vengono reimposti su 0 (OFF) | Uscita Richiesta | 189 |
| SINGLE BIT SET (solo CS1-H, CJ1-H, CJ1M o CS1D) SETB @SETB !SETB ^{*1} !@SETB ^{*1} |  D: indirizzo del canale N: numero di bit | SETB(532) imposta su ON il bit specificato nel canale specificato quando lo stato della condizione di esecuzione è ON. A differenza dell'istruzione SET, è possibile utilizzare SETB(532) per impostare un bit in un canale DM o EM. | Uscita Richiesta | 192 |

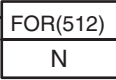
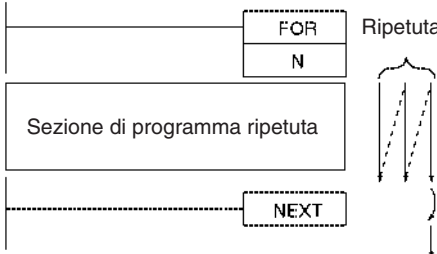

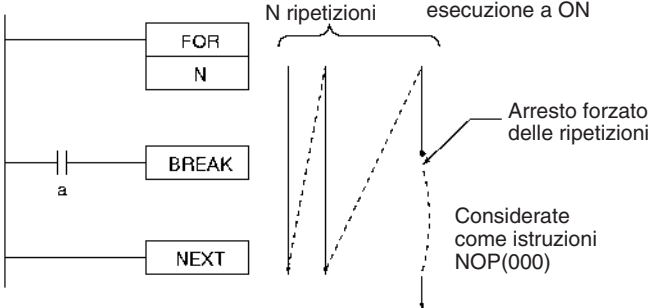
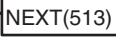
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| SINGLE BIT RESET (solo CS1-H, CJ1-H, CJ1M o CS1D) RSTB @RSTB !RSTB ^{*1} !@RSTB ^{*1} |  D: indirizzo del canale N: numero di bit | RSTB(533) imposta su OFF il bit specificato nel canale specificato quando lo stato della condizione di esecuzione è ON. A differenza dell'istruzione RSET, è possibile utilizzare RSTB(533) per reimpostare un bit in un canale DM o EM. | Uscita Richiesta | 192 |
| SINGLE BIT OUTPUT (solo CS1-H, CJ1-H, CJ1M o CS1D) OUTB @OUTB !OUTB ^{*1} |  D: indirizzo del canale N: numero di bit | OUTB(534) invia in uscita il risultato (condizione di esecuzione) dell'elaborazione logica al bit specificato. A differenza dell'istruzione OUT, è possibile utilizzare OUTB(534) per controllare un bit di un canale DM o EM. | Uscita Richiesta | 195 |

2-2-3 Istruzioni di controllo sequenza

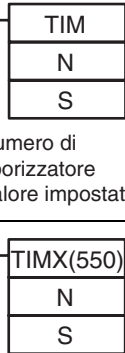
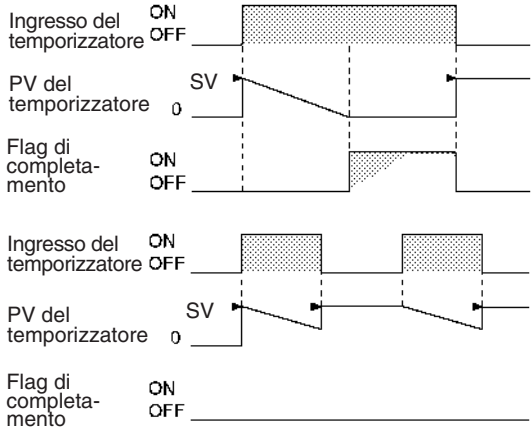
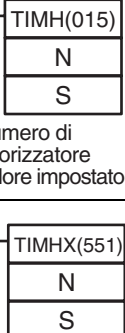
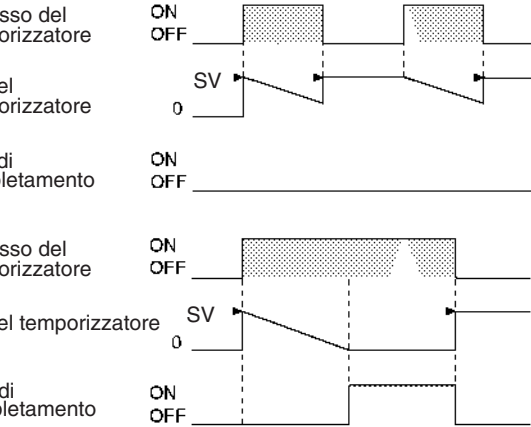
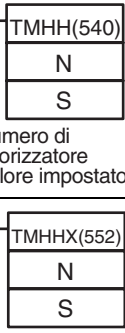
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|----------------------|---|--|-------------|
| <p>END</p> <p>END 001</p> | <p>END(001)</p> | <p>Indica la fine del programma. END(001) termina l'esecuzione di un programma in quel ciclo. Le istruzioni inserite dopo END(001) non vengono eseguite. Viene eseguito il programma con il numero di task successivo. Se numero di task del programma in esecuzione è il più alto all'interno del programma, l'istruzione END(001) marca la fine del programma principale.</p>  | <p>Uscita Non richiesta</p> | <p>197</p> |
| <p>NO OPERATION</p> <p>NOP 000</p> | | <p>Istruzione priva di funzione. Per NOP(000) non viene eseguita alcuna elaborazione.</p> | <p>Uscita Non richiesta</p> | <p>198</p> |
| <p>INTERLOCK</p> <p>IL 002</p> | <p>IL(002)</p> | <p>Determina l'interblocco per tutte le uscite comprese tra IL(002) e ILC(003) quando lo stato della condizione di esecuzione per IL(002) è OFF. IL(002) e ILC(003) vengono di norma utilizzate in coppia.</p>  | <p>Uscita Richiesta</p> | <p>201</p> |

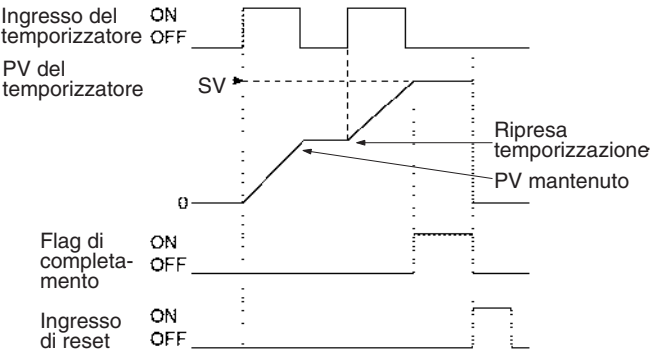
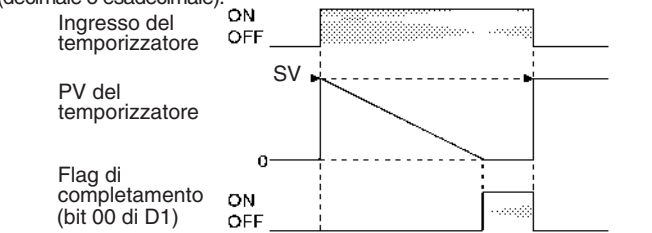
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|---|--|-------------|
| INTERLOCK CLEAR ILC 003 | | Interblocco di tutte le uscite comprese tra le istruzioni IL(002) e ILC(003) quando la condizione di esecuzione per IL(002) è OFF. ILC(002) e ILC(003) vengono di norma utilizzate in coppia. | Uscita Non richiesta | 201 |
| MULTI-INTER- LOCK DIFE- RENTIATION HOLD MILH 517 CPU della serie CS/CJ versione 2.0 o successiva | N: numero di interblocco D: bit di stato interblocco | Quando la condizione di esecuzione per MILH(517) è OFF, le uscite di tutte le istruzioni comprese tra tale istruzione MILH(517) e l'istruzione MILC(519) successiva vengono interbloccate. Le istruzioni MILH(517) e MILC(519) vengono utilizzate in coppia. Gli interblocchi MILH(517)/MILC(519) possono essere nidificati [ad esempio MILH(517)—MILH(517)—MILC(519)—MILC(519)]. Se tra l'istruzione MILH(517) e l'istruzione MILC(519) corrispondente è presente un'istruzione differenziata (DIFU, DIFD o un'istruzione con prefisso @ o %), tale istruzione verrà eseguita dopo la cancellazione dell'interblocco se viene soddisfatta la condizione di differenziazione dell'istruzione. | Uscita Richiesta | 205 |
| MULTI-INTER- LOCK DIFE- RENTIATION RELEASE MILR 518 CPU della serie CS/CJ versione 2.0 o successiva | N: numero di interblocco D: bit di stato interblocco | Quando la condizione di esecuzione per MILR(518) è OFF, le uscite di tutte le istruzioni comprese tra tale istruzione MILR(518) e l'istruzione MILC(519) successiva vengono interbloccate. Le istruzioni MILR(518) e MILC(519) vengono utilizzate in coppia. Gli interblocchi MILR(518)/MILC(519) possono essere nidificati [ad esempio MILR(518)—MILR(518)—MILC(519)—MILC(519)]. Se tra l'istruzione MILR(518) e l'istruzione MILC(519) corrispondente è presente un'istruzione differenziata (DIFU, DIFD o un'istruzione con prefisso @ o %), tale istruzione non verrà eseguita dopo la cancellazione dell'interblocco anche se viene soddisfatta la condizione di differenziazione dell'istruzione. | Uscita Richiesta | 205 |
| MULTI-INTER- LOCK CLEAR MILC 519 CPU della serie CS/CJ versione 2.0 o successiva | N: numero di interblocco | Cancella l'interblocco avviato da un'istruzione MILH(517) o MILR(518) con il medesimo numero di interblocco. Tutte le uscite comprese tra l'istruzione MILH(517)/MILR(518) e l'istruzione MILC(519) corrispondente con lo stesso numero di interblocco vengono interbloccate quando la condizione di esecuzione per MILH(517)/MILR(518) è OFF. | Uscita Non richiesta | 205 |
| JUMP JMP 004 | N: numero di salto | Quando la condizione di esecuzione per JMP(004) è OFF, l'esecuzione del programma passa direttamente alla prima istruzione JME(05) nel programma avente lo stesso numero di salto. Le istruzioni JMP(004) e JME(005) vengono utilizzate in coppia. | Uscita Richiesta | 219 |
| JUMP END JME 005 | N: numero di salto | Indica la fine di un salto il cui inizio è specificato dall'istruzione JMP(004) o CJP(510). | Uscita Non richiesta | 219 |

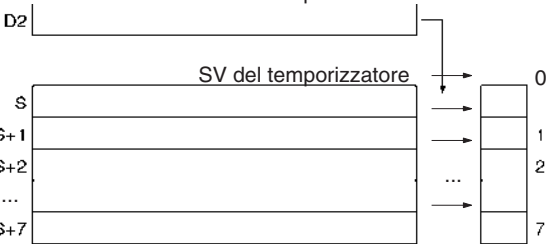
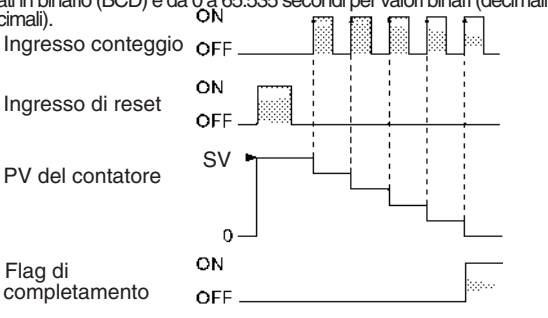
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| CONDITIONAL JUMP CJP 510 |  N: numero di salto | <p>Il funzionamento di CJP(510) è sostanzialmente opposto a quello di JMP(004). Quando la condizione di esecuzione per CJP(510) è ON, l'esecuzione del programma passa direttamente alla prima istruzione JME(05) nel programma avente lo stesso numero di salto. Le istruzioni CJP(510) e JME(005) vengono utilizzate in coppia.</p> | Uscita Richiesta | 223 |
| CONDITIONAL JUMP CJPN 511 |  N: numero di salto | <p>Il funzionamento di CJPN(511) è quasi uguale a quello di JMP(004). Quando la condizione di esecuzione per CJP(004) è OFF, l'esecuzione del programma passa direttamente alla prima istruzione JME(05) nel programma avente lo stesso numero di salto. Le istruzioni CJPN(511) e JME(005) vengono utilizzate in coppia.</p> | Uscita Non richiesta | 223 |
| MULTIPLE JUMP JMP0 515 |  | <p>Quando lo stato della condizione di esecuzione per JMP0(515) è OFF, tutte le istruzioni del programma comprese tra JMP0(515) e l'istruzione JME0(516) successiva vengono elaborate come NOP(000). Utilizzare JMP0(515) e JME0(516) in coppia. Nel programma è possibile utilizzare un numero illimitato di istruzioni JMP0(515)/JME0(516) in coppia.</p> | Uscita Richiesta | 227 |
| MULTIPLE JUMP END JME0 516 |  | <p>Quando lo stato della condizione di esecuzione per JMP0(515) è OFF, tutte le istruzioni del programma comprese tra JMP0(515) e l'istruzione JME0(516) successiva vengono elaborate come NOP(000). Utilizzare JMP0(515) e JME0(516) in coppia. Nel programma è possibile utilizzare un numero illimitato di istruzioni JMP0(515)/JME0(516) in coppia.</p> | Uscita Non richiesta | 227 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|--|--|-------------|
| FOR-NEXT LOOPS FOR 512 |  <p>N: numero di cicli</p> | <p>Le istruzioni comprese tra FOR(512) e NEXT(513) vengono ripetute per il numero di volte specificato. Le istruzioni FOR(512) e NEXT(513) devono essere utilizzate in coppia.</p>  | Uscita Non richiesta | 229 |
| BREAK LOOP BREAK 514 |  | <p>Viene inserita in un ciclo FOR-NEXT per interrompere l'esecuzione del ciclo quando si verifica una determinata condizione. Le rimanenti istruzioni all'interno del ciclo vengono considerate come istruzioni NOP(000)</p>  | Uscita Richiesta | 232 |
| FOR-NEXT LOOPS NEXT 513 |  | <p>Le istruzioni comprese tra FOR(512) e NEXT(513) vengono ripetute per il numero di volte specificato. Le istruzioni FOR(512) e NEXT(513) devono essere utilizzate in coppia.</p> | Uscita Non richiesta | 229 |

2-2-4 Istruzioni di temporizzatore e contatore

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|--|--|-------------|
| TIMER TIM (BCD) TIMX (binario) (solo CS1-H, CJ1- H, CJ1M o CS1D) |  <p>N: numero di temporizzatore S: valore impostato</p> <p>N: numero di temporizzatore S: valore impostato</p> | <p>TIM/TIMX(550) aziona un temporizzatore decrementale con unità di 0,1 s. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 999,9 s per il formato decimale codificato in binario e tra 0 e 6.553,5 s per il formato binario (decimale o esadecimale).</p>  | Uscita Richiesta | 235 |
| HIGH-SPEED TIMER TIMH 015 (BCD) TIMHX 551 (binario) (solo CS1-H, CJ1- H, CJ1M o CS1D) |  <p>N: numero di temporizzatore S: valore impostato</p> <p>N: numero di temporizzatore S: valore impostato</p> | <p>TIMH(015)/TIMHX(551) aziona un temporizzatore decrementale con unità di 10 ms. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 99,99 s per il formato decimale codificato in binario e tra 0 e 655,35 s per il formato binario (decimale o esadecimale).</p>  | Uscita Richiesta | 240 |
| ONE-MS TIMER TMHH 540 (BCD) TMHHX 552 (BCD) (solo CS1-H, CJ1- H, CJ1M o CS1D) |  <p>N: numero di temporizzatore S: valore impostato</p> <p>N: numero di temporizzatore S: valore impostato</p> | <p>TMHH(540)/TMHHX(552) aziona un temporizzatore decrementale con unità di 1 ms. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 9,999 s per il formato BCD e tra 0 e 65,535 s per il formato binario (decimale o esadecimale).</p> <p>I diagrammi di temporizzazione per TMHH(540) e TMHHX(552) sono uguali a quelli indicati in precedenza per TIMH(015).</p> | Uscita Richiesta | 244 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| ACCUMULATIVE TIMER TTIM 087 (BCD) TTIMX 555 (binario) (solo CS1-H, CJ1- H, CJ1M o CS1D) | <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> Ingresso del temporizzatore TTIM(087) N S </div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> Ingresso di reset N: numero di temporizzatore S: valore impostato </div> <div style="border: 1px solid black; padding: 2px;"> Ingresso del temporizzatore TTIMX(555) N S </div> <div style="border: 1px solid black; padding: 2px;"> Ingresso di reset N: numero di temporizzatore S: valore impostato </div> | <p>TTIM(087)/TTIMX(555) aziona un temporizzatore incrementale con unità di 0,1 s. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 999,9 s per il formato decimale codificato in binario e tra 0 e 6.553,5 s per il formato binario (decimale o esadecimale).</p>  <p>Ingresso del temporizzatore ON OFF</p> <p>PV del temporizzatore SV</p> <p>Flag di completamento ON OFF</p> <p>Ingresso di reset ON OFF</p> <p>Ripresa temporizzazione</p> <p>PV mantenuto</p> | Uscita Richiesta | 247 |
| LONG TIMER TIML 542 (BCD) TIMLX 553 (binario) (solo CS1-H, CJ1- H, CJ1M o CS1D) | <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> TIML(542) D1 D2 S </div> <div style="border: 1px solid black; padding: 2px;"> TIMLX(553) D1 D2 S </div> <p>D1: flag di completamento D2: canale PV S: canale SV</p> <p>D1: flag di completamento D2: canale PV S: canale SV</p> | <p>TIML(542)/TIMLX(553) aziona un temporizzatore decrementale con unità di 0,1 s che può raggiungere un periodo di circa 115 giorni per il formato decimale codificato in binario e 49.710 giorni per il formato binario (decimale o esadecimale).</p>  <p>Ingresso del temporizzatore ON OFF</p> <p>PV del temporizzatore SV</p> <p>Flag di completamento (bit 00 di D1) ON OFF</p> | Uscita Richiesta | 251 |

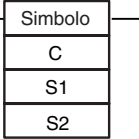
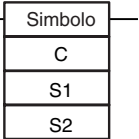
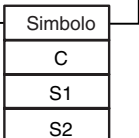
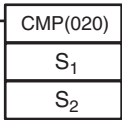
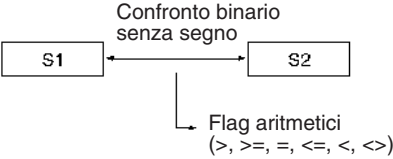
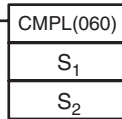
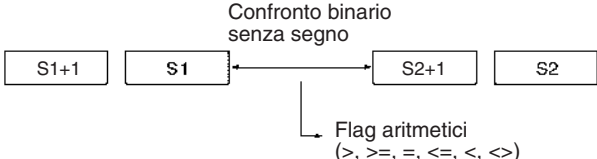
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|--|--|-------------|
| <p>MULTI-OUTPUT TIMER</p> <p>MTIM 543 (BCD)</p> <p>MTIMX 554 (binario) (solo CS1-H, CJ1- H, CJ1M o CS1D)</p> | <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> MTIM(543) D1 D2 S </div> <p>D1: flag di completamento D2: canale PV S: primo canale SV</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> MTIMX(554) D1 D2 S </div> <p>D1: flag di completamento D2: canale PV S: primo canale SV</p> | <p>MTIM(543)/MTIMX(554) aziona un temporizzatore incrementale con unità di 0,1 s che dispone di 8 flag di completamento e valori impostati indipendenti. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 999,9 secondi per valori decimali codificati in binario (BCD) e da 0 a 6.553,5 secondi per valori binari (decimali o esadecimali).</p> <p>PV del temporizzatore</p>  <p>Ingresso del temporizzatore</p> <p>ON OFF</p> <p>SV 7 ... SV 2 SV 1 PV del temporizzatore (D2) SV 0 0</p> <p>Flag di completamento (D1)</p> <p>Bit 7 ... Bit 2 Bit 1 Bit 0</p> | <p>Uscita Richiesta</p> | <p>254</p> |
| <p>COUNTER</p> <p>CNT (BCD)</p> <p>CNTX 546 (binario) (solo CS1-H, CJ1- H, CJ1M o CS1D)</p> | <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> CNT Ingresso conteggio N S </div> <p>Ingresso di reset</p> <p>N: numero di contatore S: valore impostato</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> CNTX(546) Ingresso conteggio N S Ingresso di reset </div> <p>N: numero di contatore S: valore impostato</p> | <p>CNT/CNTX(546) attiva un contatore decrementale. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 9.999 secondi per valori decimali codificati in binario (BCD) e da 0 a 65.535 secondi per valori binari (decimali o esadecimali).</p> <p>Ingresso conteggio</p> <p>ON OFF</p> <p>Ingresso di reset</p> <p>ON OFF</p> <p>PV del contatore</p> <p>SV 0</p> <p>Flag di completamento</p> <p>ON OFF</p>  | <p>Uscita Richiesta</p> | <p>260</p> |

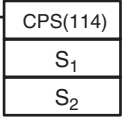
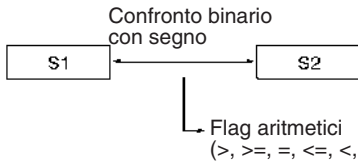
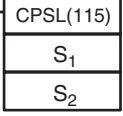
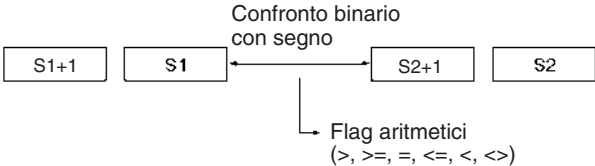
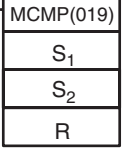
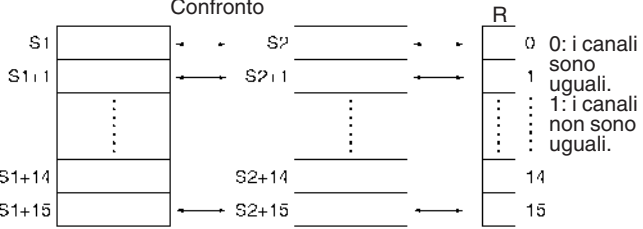
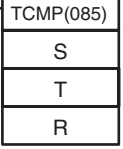
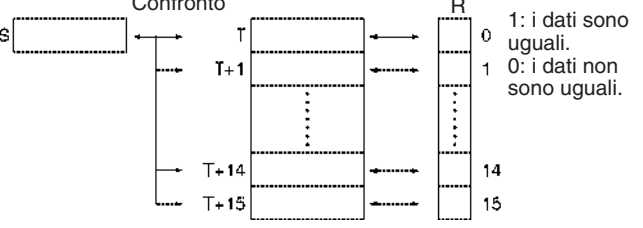
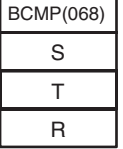
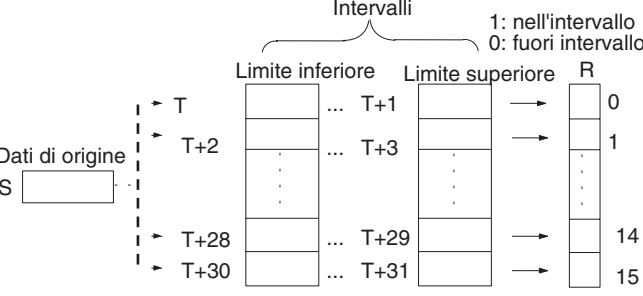
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|--|--|-------------|
| <p>REVERSIBLE COUNTER</p> <p>CNTR 012 (BCD)</p> <p>CNTRX 548 (binario) (solo CS1-H, CJ1-H, CJ1M o CS1D)</p> | <p>Ingresso di incremento Ingresso di decremento Ingresso di reset</p> <p>CNTR(012) N S</p> <p>N: numero di contatore S: valore impostato</p> <hr/> <p>Ingresso di incremento Ingresso di decremento Ingresso di reset</p> <p>CNTRX(548) N S</p> <p>N: numero di contatore S: valore impostato</p> | <p>CNTR(012)/CNTRX(548) attiva un contatore reversibile.</p> <p>Ingresso di incremento</p> <p>Ingresso di decremento</p> <p>PV del contatore</p> <p>0</p> <p>PV del contatore</p> <p>SV</p> <p>0</p> <p>+1</p> <p>Flag di completamento</p> <p>ON</p> <p>OFF</p> <p>PV del contatore</p> <p>SV</p> <p>0</p> <p>1</p> <p>Flag di completamento</p> <p>ON</p> <p>OFF</p> | <p>Uscita Richiesta</p> | <p>263</p> |
| <p>RESET TIMER/ COUNTER</p> <p>CNR @CNR 545 (BCD)</p> <p>CNRX @CNRX 547 (binario) (solo CS1-H, CJ1-H, CJ1M o CS1D)</p> | <p>CNR(545) N1 N2</p> <p>N₁: primo numero nell'intervallo N₂: ultimo numero nell'intervallo</p> <hr/> <p>CNRX(547) N1 N2</p> <p>N₁: primo numero nell'intervallo N₂: ultimo numero nell'intervallo</p> | <p>CNR(545)/CNRX(547) reimposta i temporizzatori o i contatori che rientrano nell'intervallo dei numeri di temporizzatore o contatore specificato. Imposta il valore impostato su 9999 (valore massimo).</p> | <p>Uscita Richiesta</p> | <p>267</p> |

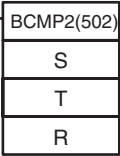
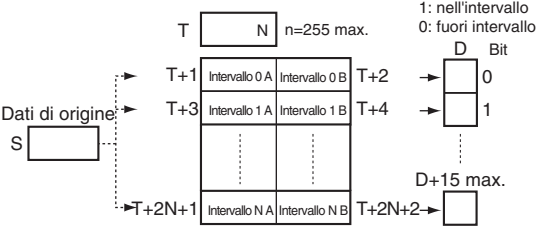
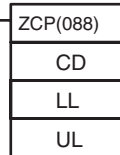

2-2-5 Istruzioni di confronto

*1: non supportata dalle CPU CS1D per sistemi a due CPU.

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|--|--|-------------|
| Confronto con simboli (senza segno) LD, AND, OR +=, <>, <, <=, >, >= 300 (=) 305 (<>) 310 (<) 315 (<=) 320 (>) 325 (>=) | <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Simbolo e opzioni</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px; text-align: center;">S₁</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px; text-align: center;">S₂</div> <p>S₁: dati di confronto 1 S₂: dati di confronto 2</p> | <p>Le istruzioni di confronto dei simboli (senza segno) confrontano due valori (costanti e/o il contenuto di canali specifici) in formato binario a 16 bit e creano una condizione di esecuzione ON quando la condizione di confronto è vera. Esistono tre tipi di istruzioni di confronto con simboli, ovvero LD (LOAD), AND e OR.</p> <p>LD Condizione di esecuzione ON quando il risultato del confronto è vero.</p> <p>AND Condizione di esecuzione ON quando il risultato del confronto è vero.</p> <p>OR Condizione di esecuzione ON quando il risultato del confronto è vero.</p> | LD: non richiesta AND, OR: richiesta | 275 |
| Confronto con simboli (doppio canale, senza segno) LD, AND, OR +=, <>, <, <=, >, >= + L 301 (=) 306 (<>) 311 (<) 316 (<=) 321 (>) 326 (>=) | <p>S₁: dati di confronto 1 S₂: dati di confronto 2</p> | <p>Le istruzioni di confronto con simboli (doppio canale, senza segno) confrontano due valori (costanti e/o il contenuto di dati specifici a doppio canale) in formato binario a 32 bit senza segno e creano una condizione di esecuzione ON quando la condizione di confronto è vera. Esistono tre tipi di istruzioni di confronto con simboli, ovvero LD (LOAD), AND e OR.</p> | LD: non richiesta AND, OR: richiesta | 275 |
| Confronto con simboli (con segno) LD, AND, OR +=, <>, <, <=, >, >= + +S 302 (=) 307 (<>) 312 (<) 317 (<=) 322 (>) 327 (>=) | <p>S₁: dati di confronto 1 S₂: dati di confronto 2</p> | <p>Le istruzioni di confronto con simboli (con segno) confrontano due valori (costanti e/o il contenuto di canali specifici) in formato binario a 16 bit con segno (esadecimale a 4 cifre) e creano una condizione di esecuzione ON quando la condizione di confronto è vera. Esistono tre tipi di istruzioni di confronto con simboli, ovvero LD (LOAD), AND e OR.</p> | LD: non richiesta AND, OR: richiesta | 275 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| Confronto con simboli (doppio canale, con segno) LD, AND, OR + =, <>, <, <=, >, >= +SL 303 (=) 308 (<>) 313 (<) 318 (<=) 323 (>) 328 (>=) | S₁ : dati di confronto 1 S₂ : dati di confronto 2 | Le istruzioni di confronto con simboli (doppio canale, con segno) confrontano due valori (costanti e/o il contenuto di dati specifici a doppio canale) in formato binario con segno a 32 bit (esadecimale a 8 cifre) e creano una condizione di esecuzione ON quando la condizione di confronto è vera. Esistono tre tipi di istruzioni di confronto con simboli, ovvero LD (LOAD), AND e OR. | LD: non richiesta AND, OR: richiesta | 275 |
| Confronto di date e ore LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT 341 (= DT) 342 (<> DT) 343 (< DT) 344 (<= DT) 345 (> DT) 346 (>= DT) (solo CPU della serie CS/CJ versione 2.0 o successiva). | LD (LOAD):  AND:  OR:  C : canale di controllo S₁ : primo canale del tempo attuale S₂ : primo canale del tempo di confronto | Le istruzioni di confronto di date e ore confrontano due valori temporali in formato BCD e creano una condizione di esecuzione ON quando la condizione di confronto è soddisfatta. Esistono tre tipi di istruzioni di confronto di date e ore, ovvero LD (LOAD), AND e OR. I valori (anno, mese, giorno, ora, minuto e secondo) possono essere mascherati/smascherati nel confronto per semplificare la creazione di funzioni di temporizzatore calendario. | LD: non richiesta AND, OR: richiesta | 281 |
| UNSIGNED COMPARE CMP !CMP*1 020 |  S₁ : dati di confronto 1 S₂ : dati di confronto 2 | Confronta due valori binari senza segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.  | Uscita Richiesta | 287 |
| DOUBLE UNSIGNED COMPARE CMPL 060 |  S₁ : dati di confronto 1 S₂ : dati di confronto 2 | Confronta due valori binari doppi senza segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.  | Uscita Richiesta | 290 |

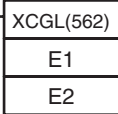
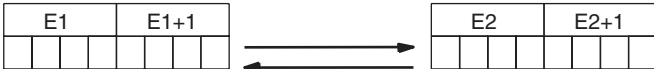
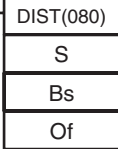
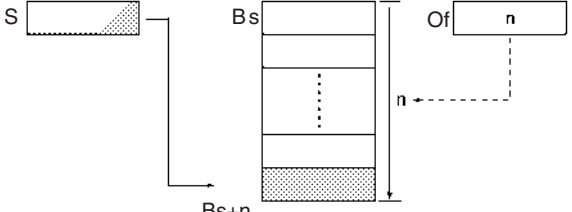
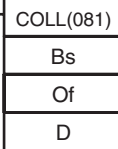
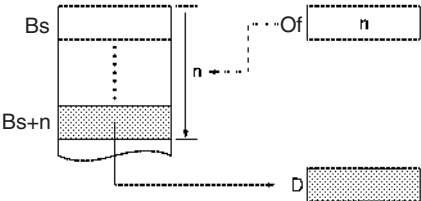
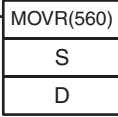
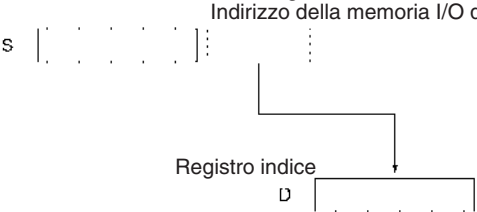
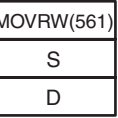
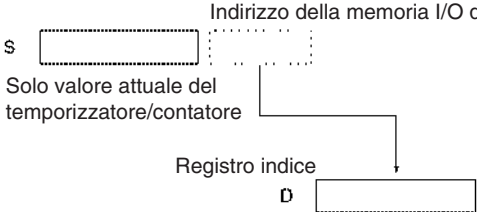
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|--|--|-------------|
| SIGNED BINARY COMPARE CPS !CPS*1 114 |  <p>S1: dati di confronto 1 S2: dati di confronto 2</p> | <p>Confronta due valori binari con segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.</p>  | Uscita Richiesta | 293 |
| DOUBLE SIGNED BINARY COMPARE CPSL 115 |  <p>S1: dati di confronto 1 S2: dati di confronto 2</p> | <p>Confronta due valori binari doppi con segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.</p>  | Uscita Richiesta | 296 |
| MULTIPLE COMPARE MCMP @MCMP 019 |  <p>S1: primo canale del gruppo 1 S2: primo canale del gruppo 2 R: canale del risultato</p> | <p>Confronta 16 canali consecutivi con altri 16 canali consecutivi e, se il contenuto di un canale non è uguale, imposta su ON il bit corrispondente nel canale del risultato.</p>  | Uscita Richiesta | 299 |
| TABLE COMPARE TCMP @TCMP 085 |  <p>S: dati di origine T: primo canale della tabella R: canale del risultato</p> | <p>Confronta i dati di origine con il contenuto di 16 canali e, se il contenuto è uguale, imposta su ON il bit corrispondente nel canale del risultato.</p>  | Uscita Richiesta | 301 |
| UNSIGNED BLOCK COMPARE BCMP @BCMP 068 |  <p>S: dati di origine T: primo canale della tabella R: canale del risultato</p> | <p>Confronta i dati di origine con 16 intervalli (definiti da 16 limiti inferiori e 16 limiti superiori) e, se i dati di origine rientrano nell'intervallo, imposta su ON il bit corrispondente nel canale del risultato.</p>  | Uscita Richiesta | 304 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|--|--|-------------|
| EXPANDED BLOCK COMPARE BCMP2 @BCMP2 502 (solo CPU CS1-H, CJ1-H o CS1D versione 2.0 o successiva). Solo CPU CJ1M (precedenti alla versione 2.0 o ver- sione 2.0 o suc- cessiva) |  <p>S: dati di origine T: primo canale del blocco R: canale del risultato</p> | <p>Confronta i dati di origine con intervalli definiti da un limite inferiore e superiore, fino a un massimo di 256, e quando i dati di origine rientrano in un intervallo, imposta su ON il bit corrispondente nel canale del risultato.</p>  <p>Note: A può essere inferiore, uguale o superiore a B.</p> | Uscita Richiesta | 306 |
| AREA RANGE COMPARE ZCP @ZCP 088 (solo CS1-H, CJ1- H, CJ1M o CS1D) |  <p>CD: dati da confrontare (1 canale) LL: limite inferiore dell'intervallo UL: limite superiore dell'intervallo</p> | <p>Confronta il valore binario a 16 bit senza segno in CD (contenuto del canale o costante) con l'intervallo definito da LL e UL e invia i risultati ai flag aritmetici dell'area ausiliaria.</p> | Uscita Richiesta | 310 |
| DOUBLE AREA RANGE COMPARE ZCPL @ZCPL 116 (solo CS1-H, CJ1- H, CJ1M o CS1D) |  <p>CD: dati da confrontare (2 canali) LL: limite inferiore dell'intervallo UL: limite superiore dell'intervallo</p> | <p>Confronta il valore binario a 32 bit senza segno in CD e CD+1 (contenuto del canale o costante) con l'intervallo definito da LL e UL e invia i risultati ai flag aritmetici dell'area ausiliaria.</p> | Uscita Richiesta | 313 |

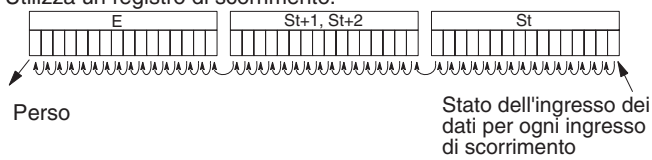
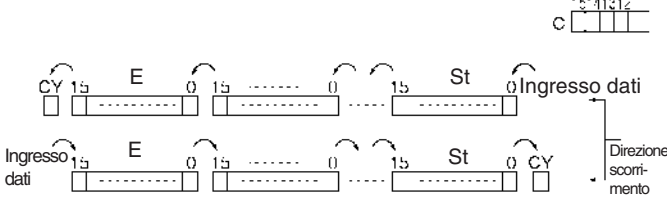
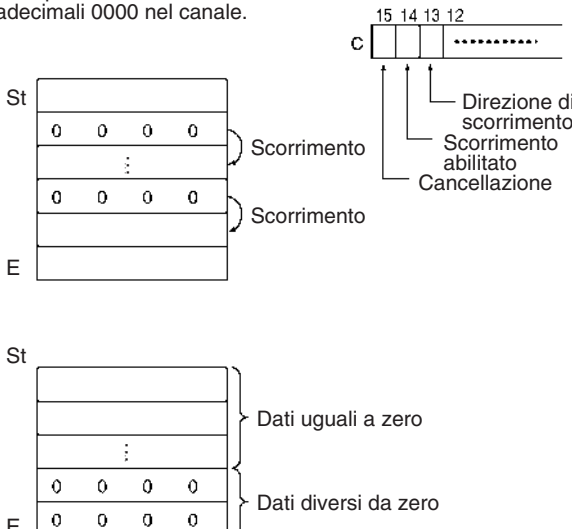
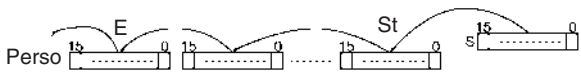
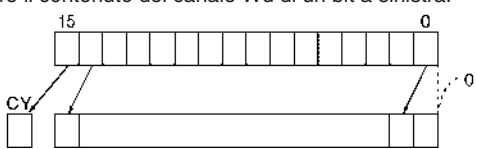
2-2-6 Istruzioni di spostamento dati

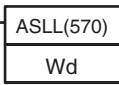
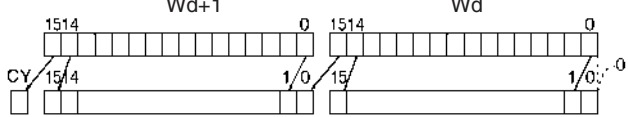
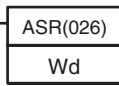
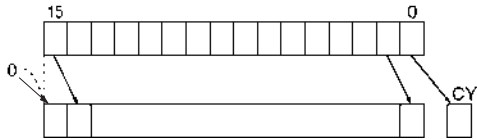
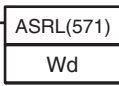
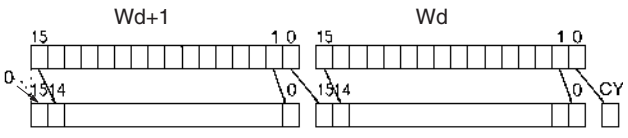
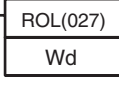
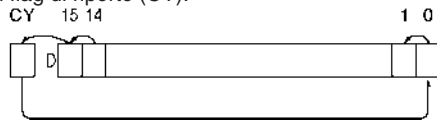
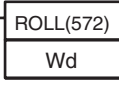
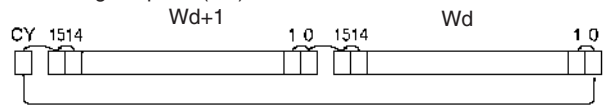
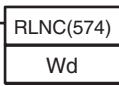
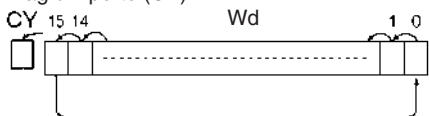
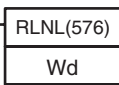
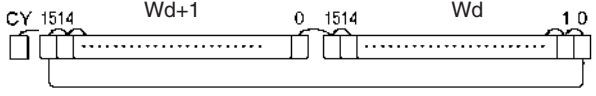
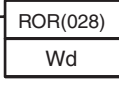
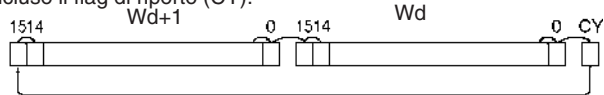
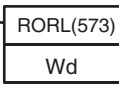
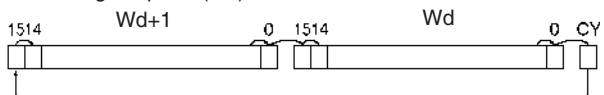
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| MOVE MOV @MOV !MOV !@MOV 021 | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> MOV(021) <hr/> S <hr/> D </div> <p>S: origine D: destinazione</p> | <p>Trasferisce un canale di dati nel canale specificato.</p> | Uscita Richiesta | 315 |
| DOUBLE MOVE MOVL @MOVL 498 | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> MOVL(498) <hr/> S <hr/> D </div> <p>S: primo canale di origine D: primo canale di destinazione</p> | <p>Trasferisce due canali di dati nei canali specificati.</p> | Uscita Richiesta | 318 |
| MOVE NOT MVN @MVN 022 | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> MVN(022) <hr/> S <hr/> D </div> <p>S: origine D: destinazione</p> | <p>Trasferisce il complemento di un canale di dati nel canale specificato.</p> | Uscita Richiesta | 317 |
| DOUBLE MOVE NOT MVNL @MVNL 499 | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> MVNL(499) <hr/> S <hr/> D </div> <p>S: primo canale di origine D: primo canale di destinazione</p> | <p>Trasferisce il complemento di due canali di dati nei canali specificati.</p> | Uscita Richiesta | 320 |
| MOVE BIT MOVB @MOVB 082 | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> MOVB(082) <hr/> S <hr/> C <hr/> D </div> <p>S: dati o canale di origine C: canale di controllo D: canale di destinazione</p> | <p>Trasferisce il bit specificato.</p> | Uscita Richiesta | 321 |

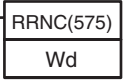

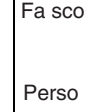
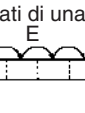
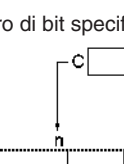

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | |
|--|--|-----------|--|-------------|---|--|---------------------|-----|
| MOVE DIGIT MOVD @MOVD 083 | <table border="1"> <tr><td>MOVD(083)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: dati o canale di origine C: canale di controllo D: canale di destinazione</p> | MOVD(083) | S | C | D | <p>Trasferisce la cifra o le cifre specificate (ogni cifra è composta da 4 bit).</p> | Uscita Richiesta | 323 |
| MOVD(083) | | | | | | | | |
| S | | | | | | | | |
| C | | | | | | | | |
| D | | | | | | | | |
| MULTIPLE BIT TRANSFER XFRB @XFRB 062 | <table border="1"> <tr><td>XFRB(062)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: canale di controllo S: primo canale di origine D: primo canale di destinazione</p> | XFRB(062) | C | S | D | <p>Trasferisce il numero di bit consecutivi specificato.</p> | Uscita Richiesta | 326 |
| XFRB(062) | | | | | | | | |
| C | | | | | | | | |
| S | | | | | | | | |
| D | | | | | | | | |
| BLOCK TRANSFER XFER @XFER 070 | <table border="1"> <tr><td>XFER(070)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>N: numero di canali S: primo canale di origine D: primo canale di destinazione</p> | XFER(070) | N | S | D | <p>Trasferisce il numero di canali consecutivi specificato.</p> | Uscita Richiesta | 328 |
| XFER(070) | | | | | | | | |
| N | | | | | | | | |
| S | | | | | | | | |
| D | | | | | | | | |
| BLOCK SET BSET @BSET 071 | <table border="1"> <tr><td>BSET(071)</td></tr> <tr><td>S</td></tr> <tr><td>St</td></tr> <tr><td>E</td></tr> </table> <p>S: canale di origine St: canale iniziale E: canale finale</p> | BSET(071) | S | St | E | <p>Copia lo stesso canale in un intervallo di canali consecutivi.</p> | Uscita Richiesta | 331 |
| BSET(071) | | | | | | | | |
| S | | | | | | | | |
| St | | | | | | | | |
| E | | | | | | | | |
| DATA EXCHANGE XCHG @XCHG 073 | <table border="1"> <tr><td>XCHG(073)</td></tr> <tr><td>E1</td></tr> <tr><td>E2</td></tr> </table> <p>E1: primo canale di scambio E2: secondo canale di scambio</p> | XCHG(073) | E1 | E2 | <p>Scambia il contenuto dei due canali specificati.</p> | Uscita Richiesta | 333 | |
| XCHG(073) | | | | | | | | |
| E1 | | | | | | | | |
| E2 | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| DOUBLE DATA EXCHANGE XCGL @XCGL 562 |  <p>E1: primo canale di scambio E2: secondo canale di scambio</p> | Scambia il contenuto di una coppia di canali consecutivi con il contenuto di un'altra coppia di canali consecutivi.  | Uscita Richiesta | 334 |
| SINGLE WORD DISTRIBUTE DIST @DIST 080 |  <p>S: canale di origine Bs: indirizzo base di destinazione Of: offset</p> | Trasferisce il canale di origine in un canale di destinazione calcolato aggiungendo il valore di offset all'indirizzo di base.  | Uscita Richiesta | 336 |
| DATA COLLECT COLL @COLL 081 |  <p>Bs: indirizzo base di origine Of: offset D: canale di destinazione</p> | Trasferisce il canale di origine, calcolato aggiungendo un valore di offset all'indirizzo di base, nel canale di destinazione.  | Uscita Richiesta | 338 |
| MOVE TO REGISTER MOVR @MOVR 560 |  <p>S: origine (canale o bit desiderato) D: destinazione (registro indice)</p> | Imposta l'indirizzo di memoria I/O interna del canale, del bit o del flag di completamento del temporizzatore/contatore specificato nel registro indice specificato. Usare l'istruzione MOVRW(561) per impostare l'indirizzo di memoria I/O interna relativo a un valore attuale del temporizzatore o del contatore in un registro indice. <p style="text-align: center;">Indirizzo della memoria I/O di S</p>  | Uscita Richiesta | 340 |
| MOVE TIMER/COUNTER PV TO REGISTER MOVRW @MOVRW 561 |  <p>S: origine (numero TC desiderato) D: destinazione (registro indice)</p> | Imposta l'indirizzo di memoria I/O interna del valore attuale del temporizzatore/contatore specificato nel registro indice specificato. Usare l'istruzione MOVR(560) per impostare l'indirizzo di memoria I/O interna relativo a un canale, a un bit o a un flag di completamento del temporizzatore/contatore in un registro indice. <p style="text-align: center;">Indirizzo della memoria I/O di S</p>  | Uscita Richiesta | 342 |

2-2-7 Istruzioni di scorrimento dei dati

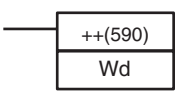
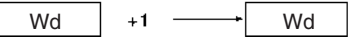
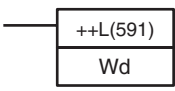

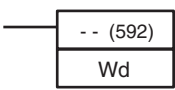
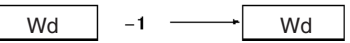
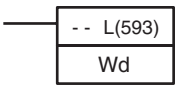
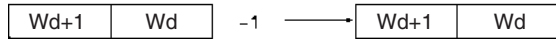
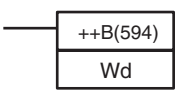
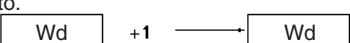
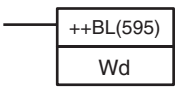

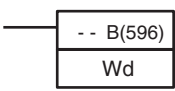
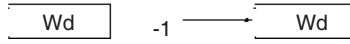


| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|--|--|-------------|
| SHIFT REGISTER SFT 010 | Ingresso dati SFT(010) Ingresso di scorrimento St Ingresso di reset E | Utilizza un registro di scorrimento.  | Uscita Richiesta | 345 |
| REVERSIBLE SHIFT REGISTER SFTR @SFTR 084 | SFTR(084) C St E | Crea un registro di scorrimento che fa scorrere i dati a destra o a sinistra.  | Uscita Richiesta | 346 |
| ASYNCHRONOUS SHIFT REGISTER ASFT @ASFT 017 | ASFT(017) C St E | Fa scorrere tutti i dati diversi da zero all'interno dell'intervallo di canali specificato verso St o verso E e sostituisce i dati esadecimali 0000 nel canale.  | Uscita Richiesta | 349 |
| WORD SHIFT WSFT @WSFT 016 | WSFT(016) S St E | Fa scorrere i dati tra St ed E, un canale alla volta.  | Uscita Richiesta | 352 |
| ARITHMETIC SHIFT LEFT ASL @ASL 025 | ASL(025) Wd | Fa scorrere il contenuto del canale Wd di un bit a sinistra.  | Uscita Richiesta | 354 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|--|--|-------------|
| DOUBLE SHIFT LEFT ASLL @ASLL 570 |  Wd: canale | Fa scorrere il contenuto dei canali Wd e Wd+1 di un bit a sinistra.  | Uscita Richiesta | 355 |
| ARITHMETIC SHIFT RIGHT ASR @ASR 026 |  Wd: canale | Fa scorrere il contenuto del canale Wd di un bit a destra.  | Uscita Richiesta | 357 |
| DOUBLE SHIFT RIGHT ASRL @ASRL 571 |  Wd: canale | Fa scorrere il contenuto dei canali Wd e Wd+1 di un bit a destra.  | Uscita Richiesta | 358 |
| ROTATE LEFT ROL @ROL 027 |  Wd: canale | Fa scorrere tutti i bit del canale Wd di un bit a sinistra, incluso il flag di riporto (CY).  | Uscita Richiesta | 360 |
| DOUBLE ROTATE LEFT ROLL @ROLL 572 |  Wd: canale | Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a sinistra, incluso il flag di riporto (CY).  | Uscita Richiesta | 362 |
| ROTATE LEFT WITHOUT CARRY RLNC @RLNC 574 |  Wd: canale | Fa scorrere tutti i bit del canale Wd di un bit a sinistra, escluso il flag di riporto (CY).  | Uscita Richiesta | 367 |
| DOUBLE ROTATE LEFT WITHOUT CARRY RLNL @RLNL 576 |  Wd: canale | Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a sinistra, escluso il flag di riporto (CY).  | Uscita Richiesta | 369 |
| ROTATE RIGHT ROR @ROR 028 |  Wd: canale | Fa scorrere tutti i bit del canale Wd di un bit a destra, incluso il flag di riporto (CY).  | Uscita Richiesta | 364 |
| DOUBLE ROTATE RIGHT RORL @RORL 573 |  Wd: canale | Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a destra, incluso il flag di riporto (CY).  | Uscita Richiesta | 365 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| ROTATE RIGHT WITHOUT CARRY RRNC @RRNC 575 |  Wd: canale | Fa scorrere tutti i bit del canale Wd di un bit a destra, escluso il flag di riporto (CY). Il contenuto del bit più a destra del canale Wd si sposta nel bit più a sinistra e nel flag di riporto (CY). | Uscita Richiesta | 371 |
| DOUBLE ROTATE RIGHT WITHOUT CARRY RRNL @RRNL 577 |  Wd: canale | Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a destra, escluso il flag di riporto (CY). Il contenuto del bit più a destra del canale Wd+1 si sposta nel bit più a sinistra del canale Wd e nel flag di riporto (CY). | Uscita Richiesta | 372 |
| ONE DIGIT SHIFT LEFT SLD @SLD 074 |  St: canale iniziale E: canale finale | Fa scorrere i dati di una cifra (4 bit) a sinistra. | Uscita Richiesta | 374 |
| ONE DIGIT SHIFT RIGHT SRD @SRD 075 |  St: canale iniziale E: canale finale | Fa scorrere i dati di una cifra (4 bit) a destra. | Uscita Richiesta | 376 |
| SHIFT N-BIT DATA LEFT NSFL @NSFL 578 |  D: canale iniziale dello scorrimento C: bit iniziale N: lunghezza dati di scorrimento | Fa scorrere a sinistra il numero di bit specificato. | Uscita Richiesta | 377 |
| SHIFT N-BIT DATA RIGHT NSFR @NSFR 579 |  D: canale iniziale dello scorrimento C: bit iniziale N: lunghezza dati di scorrimento | Fa scorrere a destra il numero di bit specificato. | Uscita Richiesta | 379 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | |
|--|---|-----------|--|---|---|------------------|-----|
| SHIFT N-BITS LEFT NASL @NASL 580 | <table border="1" style="width: 100%; text-align: center;"> <tr><td style="width: 50%;">D</td></tr> <tr><td style="width: 50%;">C</td></tr> </table> <p>D: canale di scorrimento C: canale di controllo</p> | D | C | <p>Fa scorrere i 16 bit specificati dei dati canale a sinistra del numero di bit specificato.</p> | Uscita Richiesta | 381 | |
| D | | | | | | | |
| C | | | | | | | |
| DOUBLE SHIFT N-BITS LEFT NSLL @NSLL 582 | <table border="1" style="width: 100%; text-align: center;"> <tr><td style="width: 50%;">NSLL(582)</td></tr> <tr><td style="width: 50%;">D</td></tr> <tr><td style="width: 50%;">C</td></tr> </table> <p>D: canale di scorrimento C: canale di controllo</p> | NSLL(582) | D | C | <p>Fa scorrere i 32 bit specificati dei dati canale a sinistra del numero di bit specificato.</p> | Uscita Richiesta | 384 |
| NSLL(582) | | | | | | | |
| D | | | | | | | |
| C | | | | | | | |
| SHIFT N-BITS RIGHT NASR @NASR 581 | <table border="1" style="width: 100%; text-align: center;"> <tr><td style="width: 50%;">NASR(581)</td></tr> <tr><td style="width: 50%;">D</td></tr> <tr><td style="width: 50%;">C</td></tr> </table> <p>D: canale di scorrimento C: canale di controllo</p> | NASR(581) | D | C | <p>Fa scorrere i 16 bit specificati dei dati canale a destra del numero di bit specificato.</p> | Uscita Richiesta | 387 |
| NASR(581) | | | | | | | |
| D | | | | | | | |
| C | | | | | | | |
| DOUBLE SHIFT N-BITS RIGHT NSRL @NSRL 583 | <table border="1" style="width: 100%; text-align: center;"> <tr><td style="width: 50%;">NSRL(583)</td></tr> <tr><td style="width: 50%;">D</td></tr> <tr><td style="width: 50%;">C</td></tr> </table> <p>D: canale di scorrimento C: canale di controllo</p> | NSRL(583) | D | C | <p>Fa scorrere i 32 bit specificati dei dati canale a destra del numero di bit specificato.</p> | Uscita Richiesta | 389 |
| NSRL(583) | | | | | | | |
| D | | | | | | | |
| C | | | | | | | |

2-2-8 Istruzioni di incremento e decremento

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| INCREMENT BINARY ++ @++ 590 |  Wd: canale | Incrementa di 1 il contenuto esadecimale a 4 cifre del canale specificato.  | Uscita Richiesta | 393 |
| DOUBLE INCREMENT BINARY ++L @++L 591 |  Wd: canale | Incrementa di 1 il contenuto esadecimale a 8 cifre dei canali specificati.  | Uscita Richiesta | 395 |
| DECREMENT BINARY -- @-- 592 |  Wd: canale | Decrementa di 1 il contenuto esadecimale a 4 cifre del canale specificato.  | Uscita Richiesta | 397 |
| DOUBLE DECREMENT BINARY --L @--L 593 |  Wd: primo canale | Decrementa di 1 il contenuto esadecimale a 8 cifre dei canali specificati.  | Uscita Richiesta | 399 |
| INCREMENT BCD ++B @++B 594 |  Wd: canale | Incrementa di 1 il contenuto in formato BCD a 4 cifre del canale specificato.  | Uscita Richiesta | 401 |
| DOUBLE INCREMENT BCD ++BL @++BL 595 |  Wd: primo canale | Incrementa di 1 il contenuto in formato BCD a 8 cifre dei canali specificati.  | Uscita Richiesta | 403 |
| DECREMENT BCD --B @--B 596 |  Wd: canale | Decrementa di 1 il contenuto in formato BCD a 4 cifre del canale specificato.  | Uscita Richiesta | 405 |
| DOUBLE DECREMENT BCD --BL @--BL 597 |  Wd: primo canale | Decrementa di 1 il contenuto in formato BCD a 8 cifre dei canali specificati.  | Uscita Richiesta | 407 |

2-2-9 Istruzioni matematiche con simboli

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|--|--|-------------|
| SIGNED BINARY ADD WITHOUT CARRY + @+ 400 | + (400) Au Ad R Au: canale augendo Ad: canale addendo R: canale del risultato | Somma dati e/o costanti esadecimali a 4 cifre (canale singolo). $\begin{array}{r} \boxed{\text{Au}} \\ + \quad \boxed{\text{Ad}} \\ \hline \end{array}$ Binario con segno Il flag CY viene attivato in presenza di riporto. $\begin{array}{r} \boxed{\text{CY}} \quad \boxed{\text{R}} \\ \hline \end{array}$ Binario con segno | Uscita Richiesta | 410 |
| DOUBLE SIGNED BINARY ADD WITHOUT CARRY +L @+L 401 | +L (401) Au Ad R Au: primo canale augendo Ad: primo canale addendo R: primo canale del risultato | Somma dati e/o costanti esadecimali a 8 cifre (doppio canale). $\begin{array}{r} \boxed{\text{Au}+1} \quad \boxed{\text{Au}} \\ + \quad \boxed{\text{Ad}+1} \quad \boxed{\text{Ad}} \\ \hline \end{array}$ Binario con segno Il flag CY viene attivato in presenza di riporto. $\begin{array}{r} \boxed{\text{CY}} \quad \boxed{\text{R}+1} \quad \boxed{\text{R}} \\ \hline \end{array}$ Binario con segno | Uscita Richiesta | 412 |
| SIGNED BINARY ADD WITH CARRY +C @+C 402 | +C (402) Au Ad R Au: canale augendo Ad: canale addendo R: canale del risultato | Somma dati e/o costanti esadecimali a 4 cifre (canale singolo), incluso il flag di riporto (CY). $\begin{array}{r} \boxed{\text{Au}} \\ + \quad \boxed{\text{Ad}} \\ \hline \end{array}$ Binario con segno Il flag CY viene attivato in presenza di riporto. $\begin{array}{r} \boxed{\text{CY}} \quad \boxed{\text{R}} \\ \hline \end{array}$ Binario con segno | Uscita Richiesta | 414 |
| DOUBLE SIGNED BINARY ADD WITH CARRY +CL @+CL 403 | +CL (403) Au Ad R Au: primo canale augendo Ad: primo canale addendo R: primo canale del risultato | Somma dati e/o costanti esadecimali a 8 cifre (doppio canale), incluso il flag di riporto (CY). $\begin{array}{r} \boxed{\text{Au}+1} \quad \boxed{\text{Au}} \\ + \quad \boxed{\text{Ad}+1} \quad \boxed{\text{Ad}} \\ \hline \end{array}$ Binario con segno Il flag CY viene attivato in presenza di riporto. $\begin{array}{r} \boxed{\text{CY}} \quad \boxed{\text{R}+1} \quad \boxed{\text{R}} \\ \hline \end{array}$ Binario con segno | Uscita Richiesta | 416 |
| BCD ADD WITHOUT CARRY +B @+B 404 | +B (404) Au Ad R Au: canale augendo Ad: canale addendo R: canale del risultato | Somma dati e/o costanti in formato BCD a 4 cifre (canale singolo). $\begin{array}{r} \boxed{\text{Au}} \text{ (BCD)} \\ + \quad \boxed{\text{Ad}} \text{ (BCD)} \\ \hline \end{array}$ Il flag CY viene attivato in presenza di riporto. $\begin{array}{r} \boxed{\text{CY}} \quad \boxed{\text{R}} \text{ (BCD)} \\ \hline \end{array}$ | Uscita Richiesta | 418 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---------------------|--|-------------|---|--|------|-------------------|-------------------|---|----|-------------------|-------|----|-------------------|---------------------|---------------------|-----|----|---------|---------------------|---------------------|-----|--|----|-----|---------|---------------------|-----|
| DOUBLE BCD ADD WITHOUT CARRY +BL @+BL 405 | <table border="1"> <tr><td>+BL(405)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: primo canale augendo Ad: primo canale addendo R: primo canale del risultato</p> | +BL(405) | Au | Ad | R | <p>Somma dati e/o costanti in formato BCD a 8 cifre (doppio canale).</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>Au+1</td><td>Au</td><td>(BCD)</td></tr> <tr><td colspan="3" style="text-align: center;">+</td></tr> <tr><td>Ad+1</td><td>Ad</td><td>(BCD)</td></tr> <tr><td colspan="3" style="text-align: center;">_____</td></tr> <tr><td>CY</td><td>R+1</td><td>R (BCD)</td></tr> </table> <p>Il flag CY viene attivato in presenza di riporto.</p> | Au+1 | Au | (BCD) | + | | | Ad+1 | Ad | (BCD) | _____ | | | CY | R+1 | R (BCD) | Uscita Richiesta | 419 | | | | | | |
| +BL(405) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Au | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ad | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Au+1 | Au | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ad+1 | Ad | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| _____ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CY | R+1 | R (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BCD ADD WITH CARRY +BC @+BC 406 | <table border="1"> <tr><td>+BC(406)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: canale augendo Ad: canale augendo R: canale del risultato</p> | +BC(406) | Au | Ad | R | <p>Somma dati e/o costanti in formato BCD a 4 cifre (canale singolo), incluso il flag di riporto (CY).</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>Au</td><td>(BCD)</td></tr> <tr><td colspan="2" style="text-align: center;">+</td></tr> <tr><td>Ad</td><td>(BCD)</td></tr> <tr><td colspan="2" style="text-align: center;">_____</td></tr> <tr><td>CY</td><td></td></tr> <tr><td colspan="2" style="text-align: center;">_____</td></tr> <tr><td>CY</td><td>R (BCD)</td></tr> </table> <p>Il flag CY viene attivato in presenza di riporto.</p> | Au | (BCD) | + | | Ad | (BCD) | _____ | | CY | | _____ | | CY | R (BCD) | Uscita Richiesta | 421 | | | | | | | |
| +BC(406) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Au | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ad | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Au | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ad | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| _____ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| _____ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CY | R (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DOUBLE BCD ADD WITH CARRY +BCL @+BCL 407 | <table border="1"> <tr><td>+BCL(407)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: primo canale augendo Ad: primo canale addendo R: primo canale del risultato</p> | +BCL(407) | Au | Ad | R | <p>Somma dati e/o costanti in formato BCD a 8 cifre (doppio canale), incluso il flag di riporto (CY).</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>Au+1</td><td>Au</td><td>(BCD)</td></tr> <tr><td colspan="3" style="text-align: center;">+</td></tr> <tr><td>Ad+1</td><td>Ad</td><td>(BCD)</td></tr> <tr><td colspan="3" style="text-align: center;">_____</td></tr> <tr><td colspan="3" style="text-align: center;">CY</td></tr> <tr><td colspan="3" style="text-align: center;">_____</td></tr> <tr><td>CY</td><td>R+1</td><td>R (BCD)</td></tr> </table> <p>Il flag CY viene attivato in presenza di riporto.</p> | Au+1 | Au | (BCD) | + | | | Ad+1 | Ad | (BCD) | _____ | | | CY | | | _____ | | | CY | R+1 | R (BCD) | Uscita Richiesta | 423 |
| +BCL(407) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Au | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ad | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Au+1 | Au | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ad+1 | Ad | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| _____ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| _____ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CY | R+1 | R (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SIGNED BINARY SUBTRACT WITHOUT CARRY - @- 410 | <table border="1"> <tr><td>-(410)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: canale minuendo Su: canale sottraendo R: canale del risultato</p> | -(410) | Mi | Su | R | <p>Esegue una sottrazione tra dati e/o costanti esadecimali a 4 cifre (canale singolo).</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>Mi</td><td>Binario con segno</td></tr> <tr><td colspan="2" style="text-align: center;">-</td></tr> <tr><td>Su</td><td>Binario con segno</td></tr> <tr><td colspan="2" style="text-align: center;">_____</td></tr> <tr><td>CY</td><td>R Binario con segno</td></tr> </table> <p>Il flag CY viene attivato in presenza di prestito.</p> | Mi | Binario con segno | - | | Su | Binario con segno | _____ | | CY | R Binario con segno | Uscita Richiesta | 424 | | | | | | | | | | | |
| -(410) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Su | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mi | Binario con segno | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Su | Binario con segno | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| _____ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CY | R Binario con segno | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY -L @-L 411 | <table border="1"> <tr><td>-L(411)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: canale minuendo Su: canale sottraendo R: canale del risultato</p> | -L(411) | Mi | Su | R | <p>Esegue una sottrazione tra dati e/o costanti esadecimali a 8 cifre (doppio canale).</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>Mi+1</td><td>Mi</td><td>Binario con segno</td></tr> <tr><td colspan="3" style="text-align: center;">-</td></tr> <tr><td>Su+1</td><td>Su</td><td>Binario con segno</td></tr> <tr><td colspan="3" style="text-align: center;">_____</td></tr> <tr><td>CY</td><td>R+1</td><td>R Binario con segno</td></tr> </table> <p>Il flag CY viene attivato in presenza di prestito.</p> | Mi+1 | Mi | Binario con segno | - | | | Su+1 | Su | Binario con segno | _____ | | | CY | R+1 | R Binario con segno | Uscita Richiesta | 426 | | | | | | |
| -L(411) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Su | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mi+1 | Mi | Binario con segno | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Su+1 | Su | Binario con segno | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| _____ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CY | R+1 | R Binario con segno | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|--|--|-------------|
| SIGNED BINARY SUBTRACT WITH CARRY -C @-C 412 | <div style="border: 1px solid black; padding: 2px; width: fit-content;">-C(412)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Mi</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Su</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">R</div> <p>Mi: canale minuendo Su: canale sottraendo R: canale del risultato</p> | <p>Esegue una sottrazione tra dati e/o costanti esadecimali a 4 cifre (canale singolo), incluso il flag di riporto (CY).</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Mi</div> Binario con segno <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Su</div> Binario con segno <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> </div> <p style="text-align: center;">-</p> <hr style="width: 50%; margin: 0 auto;"/> <p>Il flag CY viene attivato in presenza di prestito.</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">R</div> Binario con segno </div> | Uscita Richiesta | 430 |
| DOUBLE SIGNED BINARY WITH CARRY -CL @-CL 413 | <div style="border: 1px solid black; padding: 2px; width: fit-content;">-CL(413)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Mi</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Su</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">R</div> <p>Mi: canale minuendo Su: canale sottraendo R: canale del risultato</p> | <p>Esegue una sottrazione tra dati e/o costanti esadecimali a 8 cifre (doppio canale), incluso il flag di riporto (CY).</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Mi+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Mi</div> Binario con segno <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Su+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Su</div> Binario con segno <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> </div> <p style="text-align: center;">-</p> <hr style="width: 50%; margin: 0 auto;"/> <p>Il flag CY viene attivato in presenza di prestito.</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">R</div> Binario con segno </div> | Uscita Richiesta | 432 |
| BCD SUBTRACT WITHOUT CARRY -B @-B 414 | <div style="border: 1px solid black; padding: 2px; width: fit-content;">-B(414)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Mi</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Su</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">R</div> <p>Mi: canale minuendo Su: canale sottraendo R: canale del risultato</p> | <p>Esegue una sottrazione tra dati e/o costanti in formato BCD a 4 cifre (canale singolo).</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Mi</div> (BCD) <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Su</div> (BCD) <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> </div> <p style="text-align: center;">-</p> <hr style="width: 50%; margin: 0 auto;"/> <p>Il flag CY viene attivato in presenza di riporto.</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">R</div> (BCD) </div> | Uscita Richiesta | 435 |
| DOUBLE BCD SUBTRACT WITHOUT CARRY -BL @-BL 415 | <div style="border: 1px solid black; padding: 2px; width: fit-content;">-BL(415)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Mi</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Su</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">R</div> <p>Mi: primo canale minuendo Su: primo canale sottraendo R: primo canale del risultato</p> | <p>Esegue una sottrazione tra dati e/o costanti in formato BCD a 8 cifre (doppio canale).</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Mi +1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Mi</div> (BCD) <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Su+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Su</div> (BCD) <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> </div> <p style="text-align: center;">-</p> <hr style="width: 50%; margin: 0 auto;"/> <p>Il flag CY viene attivato in presenza di prestito.</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">R</div> (BCD) </div> | Uscita Richiesta | 436 |
| BCD SUBTRACT WITH CARRY -BC @-BC 416 | <div style="border: 1px solid black; padding: 2px; width: fit-content;">-BC(416)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Mi</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">Su</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">R</div> <p>Mi: canale minuendo Su: canale sottraendo R: canale del risultato</p> | <p>Esegue una sottrazione tra dati e/o costanti in formato BCD a 4 cifre (canale singolo), incluso il flag di riporto (CY).</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Mi</div> (BCD) <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Su</div> (BCD) <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> </div> <p style="text-align: center;">-</p> <hr style="width: 50%; margin: 0 auto;"/> <p>Il flag CY viene attivato in presenza di prestito.</p> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">R</div> (BCD) </div> | Uscita Richiesta | 440 |

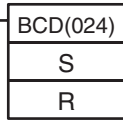

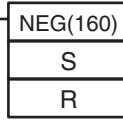
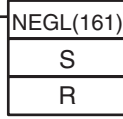
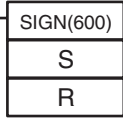
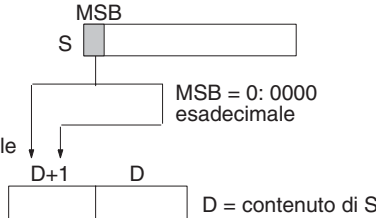
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|----------------------------|---|--|-------------|
| DOUBLE BCD SUBTRACT WITH CARRY -BCL @-BCL 417 | -BCL(417) Mi Su R | Esegue una sottrazione tra dati e/o costanti in formato BCD a 8 cifre (doppio canale), incluso il flag di riporto (CY). $\begin{array}{r} \text{Mi} + 1 \quad \text{Mi} \quad (\text{BCD}) \\ \text{Su} + 1 \quad \text{Su} \quad (\text{BCD}) \\ - \\ \text{CY} \\ \hline \text{CY} \quad \text{R} + 1 \quad \text{R} \quad (\text{BCD}) \end{array}$ <p>Il flag CY viene attivato in presenza di prestito.</p> | Uscita Richiesta | 441 |
| SIGNED BINARY MULTIPLY * @* 420 | *(420) Md Mr R | Moltiplica dati e/o costanti esadecimali a 4 cifre con segno. $\begin{array}{r} \text{Md} \quad \text{Binario con segno} \\ \times \\ \text{Mr} \quad \text{Binario con segno} \\ \hline \text{R} + 1 \quad \text{R} \quad \text{Binario con segno} \end{array}$ | Uscita Richiesta | 443 |
| DOUBLE SIGNED BINARY MULTIPLY *L @*L 421 | *L(421) Md Mr R | Moltiplica dati e/o costanti esadecimali a 8 cifre con segno. $\begin{array}{r} \text{Md} + 1 \quad \text{Md} \quad \text{Binario con segr} \\ \times \\ \text{Mr} + 1 \quad \text{Mr} \quad \text{Binario con segr} \\ \hline \text{R} + 3 \quad \text{R} + 2 \quad \text{R} + 1 \quad \text{R} \quad \text{Binario con segr} \end{array}$ | Uscita Richiesta | 445 |
| UNSIGNED BINARY MULTIPLY *U @*U 422 | *U(422) Md Mr R | Moltiplica dati e/o costanti esadecimali a 4 cifre senza segno. $\begin{array}{r} \text{Md} \quad \text{Binario senza segno} \\ \times \\ \text{Mr} \quad \text{Binario senza segno} \\ \hline \text{R} + 1 \quad \text{R} \quad \text{Binario senza segno} \end{array}$ | Uscita Richiesta | 447 |
| DOUBLE UNSI- GNEED BINARY MULTIPLY *UL @*UL 423 | *UL(423) Md Mr R | Moltiplica dati e/o costanti esadecimali a 8 cifre senza segno. $\begin{array}{r} \text{Md} + 1 \quad \text{Md} \quad \text{Binario senza segno} \\ \times \\ \text{Mr} + 1 \quad \text{Mr} \quad \text{Binario senza segno} \\ \hline \text{R} + 3 \quad \text{R} + 2 \quad \text{R} + 1 \quad \text{R} \quad \text{Binario senza segno} \end{array}$ | Uscita Richiesta | 449 |

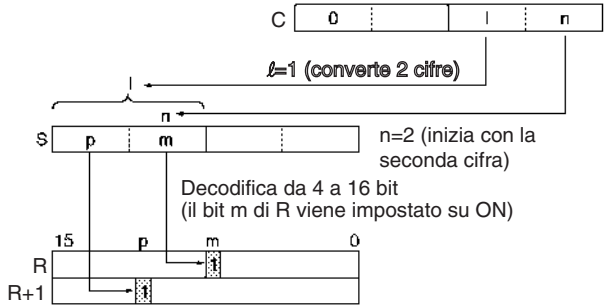
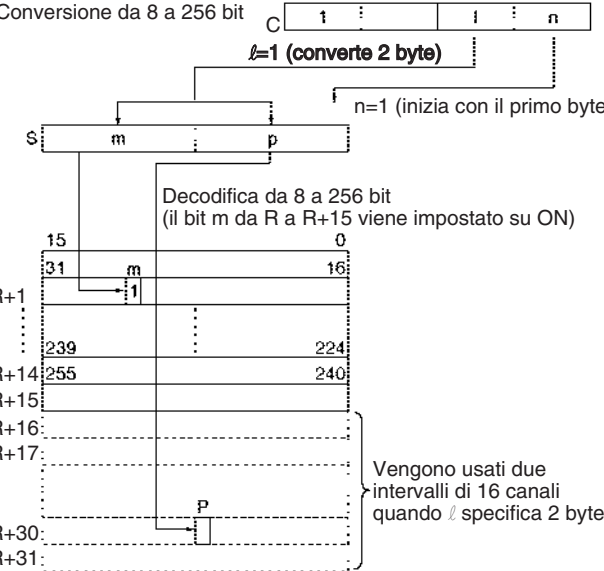
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---------------------------|---|--|-------------|
| BCD MULTIPLY *B @*B 424 | *B(424) Md Mr R | Moltiplica dati e/o costanti in formato BCD a 4 cifre (canale singolo). $\begin{array}{r} \boxed{\text{Md}} \text{ (BCD)} \\ \times \quad \boxed{\text{Mr}} \text{ (BCD)} \\ \hline \boxed{\text{R}+1} \quad \boxed{\text{R}} \text{ (BCD)} \end{array}$ | Uscita Richiesta | 450 |
| DOUBLE BCD MULTIPLY *BL @*BL 425 | *BL(425) Md Mr R | Moltiplica dati e/o costanti in formato BCD a 8 cifre (doppio canale). $\begin{array}{r} \boxed{\text{Md}+1} \quad \boxed{\text{Md}} \text{ (BCD)} \\ \times \quad \boxed{\text{Mr}+1} \quad \boxed{\text{Mr}} \text{ (BCD)} \\ \hline \boxed{\text{R}+3} \quad \boxed{\text{R}+2} \quad \boxed{\text{R}+1} \quad \boxed{\text{R}} \text{ (BCD)} \end{array}$ | Uscita Richiesta | 452 |
| SIGNED BINARY DIVIDE / @/ 430 | /(430) Dd Dr R | Esegue una divisione tra dati e/o costanti esadecimali a 4 cifre (canale singolo) con segno. $\begin{array}{r} \boxed{\text{Dd}} \text{ Binario con segno} \\ \div \quad \boxed{\text{Dr}} \text{ Binario con segno} \\ \hline \boxed{\text{R}+1} \quad \boxed{\text{R}} \text{ Binario con segno} \end{array}$ Resto Quoziente | Uscita Richiesta | 454 |
| DOUBLE SIGNED BINARY DIVIDE /L @/L 431 | /L(431) Dd Dr R | Esegue una divisione tra dati e/o costanti esadecimali a 8 cifre (doppio canale) con segno. $\begin{array}{r} \boxed{\text{Dd}+1} \quad \boxed{\text{Dd}} \text{ Binario con segno} \\ \div \quad \boxed{\text{Dr}+1} \quad \boxed{\text{Dr}} \text{ Binario con segno} \\ \hline \boxed{\text{R}+3} \quad \boxed{\text{R}+2} \quad \boxed{\text{R}+1} \quad \boxed{\text{R}} \text{ Binario con segno} \end{array}$ Resto Quoziente | Uscita Richiesta | 456 |
| UNSIGNED BINARY DIVIDE /U @/U 432 | /U(432) Dd Dr R | Esegue una divisione tra dati e/o costanti esadecimali a 4 cifre (canale singolo) senza segno. $\begin{array}{r} \boxed{\text{Dd}} \text{ Binario senza segno} \\ \div \quad \boxed{\text{Dr}} \text{ Binario senza segno} \\ \hline \boxed{\text{R}+1} \quad \boxed{\text{R}} \text{ Binario senza segno} \end{array}$ Resto Quoziente | Uscita Richiesta | 458 |

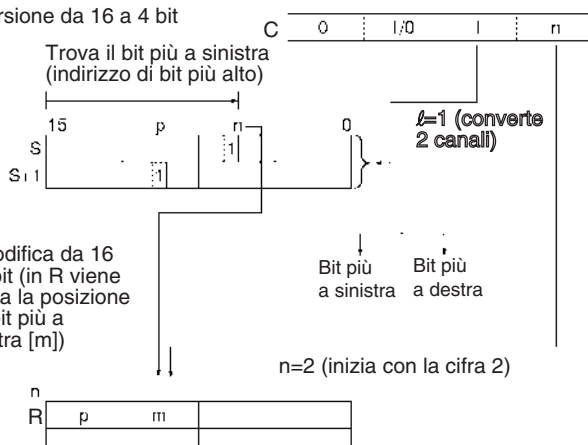
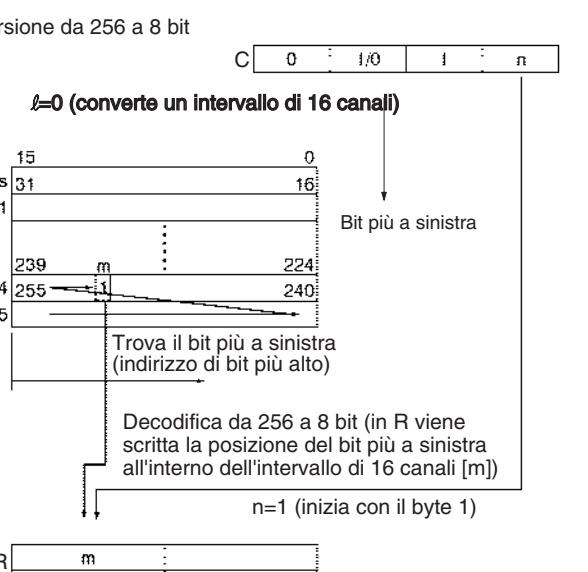
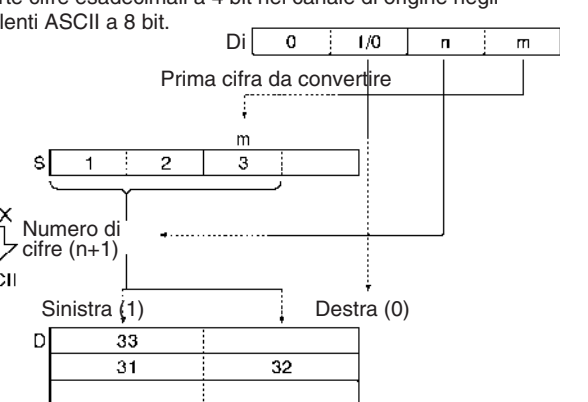
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---------------------|--|---------------------|---|--|--------|-------|---------------------|---|----|-------|--------|----|---------------------|-------|-------|-------|-----------|---------------------|-------|---|---------------------|-------|--|-----------|--|---------------------|---------------------|-----|
| DOUBLE UNSIGNED BINARY DIVIDE /UL @/UL 433 | <table border="1"> <tr><td>/UL(433)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: primo canale dividendo Dr: primo canale divisore R: primo canale del risultato</p> | /UL(433) | Dd | Dr | R | <p>Esegue una divisione tra dati e/o costanti esadecimali a 8 cifre (doppio canale) senza segno.</p> <table style="margin-left: 100px;"> <tr> <td style="border: 1px solid black; padding: 2px;">Dd + 1</td> <td style="border: 1px solid black; padding: 2px;">Dd</td> <td>Binario senza segno</td> </tr> <tr> <td colspan="3" style="text-align: center;">+</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Dr + 1</td> <td style="border: 1px solid black; padding: 2px;">Dr</td> <td>Binario senza segno</td> </tr> <tr> <td colspan="3" style="text-align: center;">-----</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">R + 3</td> <td style="border: 1px solid black; padding: 2px;">R + 2</td> <td style="border: 1px solid black; padding: 2px;">R + 1</td> <td style="border: 1px solid black; padding: 2px;">R</td> <td>Binario senza segno</td> </tr> <tr> <td colspan="2" style="text-align: center;">Resto</td> <td colspan="3" style="text-align: center;">Quoziente</td> </tr> </table> | Dd + 1 | Dd | Binario senza segno | + | | | Dr + 1 | Dr | Binario senza segno | ----- | | | R + 3 | R + 2 | R + 1 | R | Binario senza segno | Resto | | Quoziente | | | Uscita Richiesta | 460 |
| /UL(433) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dd | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dr | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dd + 1 | Dd | Binario senza segno | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dr + 1 | Dr | Binario senza segno | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R + 3 | R + 2 | R + 1 | R | Binario senza segno | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resto | | Quoziente | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BCD DIVIDE /B @/B 434 | <table border="1"> <tr><td>/B(434)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: canale dividendo Dr: canale divisore R: canale del risultato</p> | /B(434) | Dd | Dr | R | <p>Esegue una divisione tra dati e/o costanti in formato BCD a 4 cifre (canale singolo).</p> <table style="margin-left: 100px;"> <tr> <td style="border: 1px solid black; padding: 2px;">Dd</td> <td>(BCD)</td> </tr> <tr> <td colspan="2" style="text-align: center;">+</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Dr</td> <td>(BCD)</td> </tr> <tr> <td colspan="2" style="text-align: center;">-----</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">R + 1</td> <td style="border: 1px solid black; padding: 2px;">R</td> <td>(BCD)</td> </tr> <tr> <td style="text-align: center;">Resto</td> <td style="text-align: center;">Quoziente</td> </tr> </table> | Dd | (BCD) | + | | Dr | (BCD) | ----- | | R + 1 | R | (BCD) | Resto | Quoziente | Uscita Richiesta | 462 | | | | | | | | | |
| /B(434) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dd | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dr | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dd | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dr | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R + 1 | R | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resto | Quoziente | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DOUBLE BCD DIVIDE /BL @/BL 435 | <table border="1"> <tr><td>/BL(435)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: primo canale dividendo Dr: primo canale divisore R: primo canale del risultato</p> | /BL(435) | Dd | Dr | R | <p>Esegue una divisione tra dati e/o costanti in formato BCD a 8 cifre (doppio canale).</p> <table style="margin-left: 100px;"> <tr> <td style="border: 1px solid black; padding: 2px;">Dd + 1</td> <td style="border: 1px solid black; padding: 2px;">Dd</td> <td>(BCD)</td> </tr> <tr> <td colspan="3" style="text-align: center;">+</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Dr + 1</td> <td style="border: 1px solid black; padding: 2px;">Dr</td> <td>(BCD)</td> </tr> <tr> <td colspan="3" style="text-align: center;">-----</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">R + 3</td> <td style="border: 1px solid black; padding: 2px;">R + 2</td> <td style="border: 1px solid black; padding: 2px;">R + 1</td> <td style="border: 1px solid black; padding: 2px;">R</td> <td>(BCD)</td> </tr> <tr> <td colspan="2" style="text-align: center;">Resto</td> <td colspan="2" style="text-align: center;">Quoziente</td> </tr> </table> | Dd + 1 | Dd | (BCD) | + | | | Dr + 1 | Dr | (BCD) | ----- | | | R + 3 | R + 2 | R + 1 | R | (BCD) | Resto | | Quoziente | | Uscita Richiesta | 464 | |
| /BL(435) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dd | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dr | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dd + 1 | Dd | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dr + 1 | Dr | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R + 3 | R + 2 | R + 1 | R | (BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resto | | Quoziente | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

2-2-10 Istruzioni di conversione

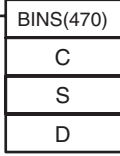
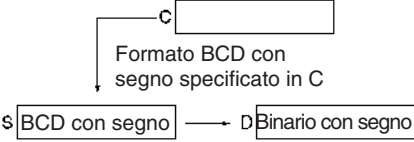
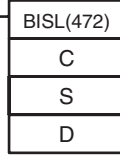
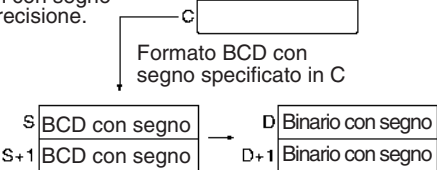
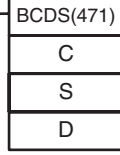
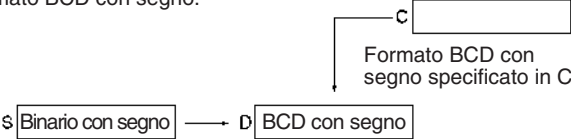
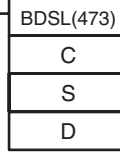
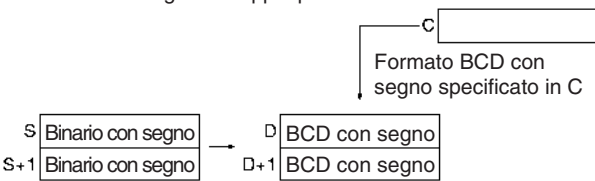
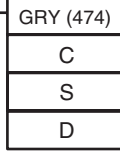
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | |
|--|---|-----------|--|-------------|---|---|-------|---|---|-----------|---------------------|-------|---|-----|-----------|---------------------|-----|
| BCD-TO-BINARY BIN @BIN 023 | <table border="1"> <tr><td>BIN(023)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: canale di origine R: canale del risultato</p> | BIN(023) | S | R | <p>Converte dati in formato BCD in dati binari.</p> <table style="margin-left: 100px;"> <tr> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">(BCD)</td> <td style="text-align: center;">→</td> <td style="border: 1px solid black; padding: 2px;">R</td> <td style="border: 1px solid black; padding: 2px;">(Binario)</td> </tr> </table> | S | (BCD) | → | R | (Binario) | Uscita Richiesta | 466 | | | | | |
| BIN(023) | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | |
| S | (BCD) | → | R | (Binario) | | | | | | | | | | | | | |
| DOUBLE BCD-TO-DOUBLE BINARY BINL @BINL 058 | <table border="1"> <tr><td>BINL(058)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: primo canale di origine R: primo canale del risultato</p> | BINL(058) | S | R | <p>Converte dati in formato BCD a 8 cifre in dati esadecimali a 8 cifre (binari a 32 bit).</p> <table style="margin-left: 100px;"> <tr> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">(BCD)</td> <td style="text-align: center;">→</td> <td style="border: 1px solid black; padding: 2px;">R</td> <td style="border: 1px solid black; padding: 2px;">(Binario)</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">S+1</td> <td style="border: 1px solid black; padding: 2px;">(BCD)</td> <td style="text-align: center;">→</td> <td style="border: 1px solid black; padding: 2px;">R+1</td> <td style="border: 1px solid black; padding: 2px;">(Binario)</td> </tr> </table> | S | (BCD) | → | R | (Binario) | S+1 | (BCD) | → | R+1 | (Binario) | Uscita Richiesta | 467 |
| BINL(058) | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | |
| S | (BCD) | → | R | (Binario) | | | | | | | | | | | | | |
| S+1 | (BCD) | → | R+1 | (Binario) | | | | | | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina |
|---|--|--|--|--------|
| BINARY-TO-BCD BCD @BCD 024 |  <p>S: canale di origine R: canale del risultato</p> | Convertire un canale di dati binari in un canale di dati in formato BCD. $s \text{ (Binario)} \longrightarrow R \text{ (BCD)}$ | Uscita Richiesta | 469 |
| DOUBLE BINARY-TO-DOUBLE BCD BCDL @BCDL 059 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Convertire dati esadecimali a 8 cifre (binari a 32 bit) in dati in formato BCD a 8 cifre. $s \text{ (Binario)} \longrightarrow R \text{ (BCD)}$ $s+1 \text{ (Binario)} \longrightarrow R+1 \text{ (BCD)}$ | Uscita Richiesta | 470 |
| 2'S COMPLEMENT NEG @NEG 160 |  <p>S: canale di origine R: canale del risultato</p> | Calcola il complemento a 2 di un canale di dati esadecimali. Complemento a 2 (complemento + 1) $\overline{(S)} \longrightarrow (R)$ | Uscita Richiesta | 472 |
| DOUBLE 2'S COMPLEMENT NEGL @NEGL 161 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola il complemento a 2 di due canali di dati esadecimali. Complemento a 2 (complemento + 1) $\overline{(S+1, S)} \longrightarrow (R+1, R)$ | Uscita Richiesta | 474 |
| 16-BIT TO 32-BIT SIGNED BINARY SIGN @SIGN 600 |  <p>S: canale di origine R: primo canale del risultato</p> | Convertire un valore binario con segno a 16 bit nell'equivalente a 32 bit.  <p>MSB = 1: FFFF esadecimale MSB = 0: 0000 esadecimale D = contenuto di S</p> | Uscita Richiesta | 476 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|--|--|-------------|
| DATA DECODER MLPX @MLPX 076 | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> MLPX(076) <hr/> S <hr/> C <hr/> R </div> <p>S: canale di origine C: canale di controllo R: primo canale del risultato</p> | <p>Legge il valore numerico nella cifra o nel byte specificato e lo copia nel canale di origine, imposta su ON il bit corrispondente e su OFF tutti gli altri bit nel canale del risultato o nell'intervallo di 16 canali.</p> <p>Conversione da 4 a 16 bit</p>  <p>Conversione da 8 a 256 bit</p>  | Uscita Richiesta | 477 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina | | | | |
|---|---|-----------|--|--------|---|--|-----------------------------|------------|
| <p>DATA ENCODER DMPX @DMPX 077</p> | <table border="1" style="margin-left: 20px;"> <tr><td>DMPX(077)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> <tr><td>C</td></tr> </table> <p>S: primo canale di origine R: canale del risultato C: canale di controllo</p> | DMPX(077) | S | R | C | <p>Determina la posizione del primo o dell'ultimo bit impostato su ON all'interno del canale di origine o dell'intervallo di 16 canali e scrive tale valore nella cifra o nel byte specificato nel canale del risultato.</p> <p>Conversione da 16 a 4 bit</p>  <p>Conversione da 256 a 8 bit</p>  | <p>Uscita Richiesta</p> | <p>482</p> |
| DMPX(077) | | | | | | | | |
| S | | | | | | | | |
| R | | | | | | | | |
| C | | | | | | | | |
| <p>ASCII CONVERT ASC @ASC 086</p> | <table border="1" style="margin-left: 20px;"> <tr><td>ASC(086)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p>S: canale di origine Di: identificativo della cifra D: primo canale di destinazione</p> | ASC(086) | S | Di | D | <p>Converte cifre esadecimali a 4 bit nel canale di origine negli equivalenti ASCII a 8 bit.</p>  | <p>Uscita Richiesta</p> | <p>486</p> |
| ASC(086) | | | | | | | | |
| S | | | | | | | | |
| Di | | | | | | | | |
| D | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | |
|---|---|-----------|--|-------------|---|--|---------------------|-----|
| ASCII TO HEX HEX @HEX 162 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>HEX(162)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p>S: primo canale di origine Di: identificativo della cifra D: canale di destinazione</p> | HEX(162) | S | Di | D | <p>Convertire fino a 4 byte di dati ASCII all'interno del canale di origine negli equivalenti esadecimali e scrive tali cifre nel canale di destinazione specificato.</p> <p style="text-align: center;">C: 0021</p> | Uscita Richiesta | 490 |
| HEX(162) | | | | | | | | |
| S | | | | | | | | |
| Di | | | | | | | | |
| D | | | | | | | | |
| COLUMN TO LINE LINE @LINE 063 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>LINE(063)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>D</td></tr> </table> <p>S: primo canale di origine N: numero di bit D: canale di destinazione</p> | LINE(063) | S | N | D | <p>Convertire una colonna di bit all'interno di un intervallo di 16 canali (lo stesso numero di bit in 16 canali consecutivi) nei 16 bit del canale di destinazione</p> | Uscita Richiesta | 494 |
| LINE(063) | | | | | | | | |
| S | | | | | | | | |
| N | | | | | | | | |
| D | | | | | | | | |
| LINE TO COLUMN COLM @COLM 064 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>COLM(064)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>N</td></tr> </table> <p>S: canale di origine D: primo canale di destinazione N: numero di bit</p> | COLM(064) | S | D | N | <p>Convertire i 16 bit del canale di origine in una colonna di bit all'interno di un intervallo di 16 canali di destinazione (lo stesso numero di bit in 16 canali consecutivi).</p> | Uscita Richiesta | 496 |
| COLM(064) | | | | | | | | |
| S | | | | | | | | |
| D | | | | | | | | |
| N | | | | | | | | |

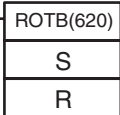
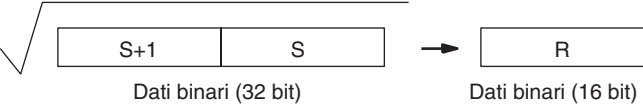
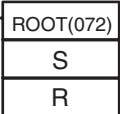

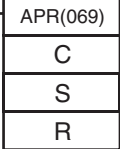
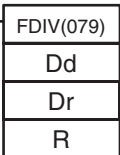
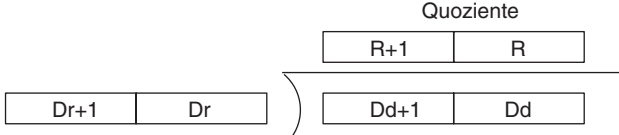
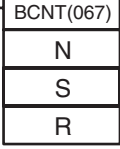
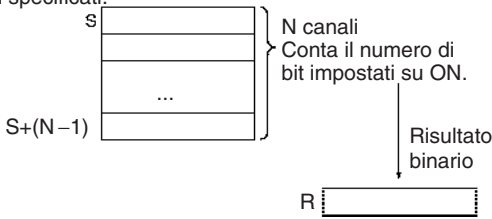
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| SIGNED BCD-TO-BINARY BINS @BINS 470 |  <p>C: canale di controllo S: canale di origine D: canale di destinazione</p> | Convertire un canale di dati in formato BCD con segno in un canale di dati binari con segno.  | Uscita Richiesta | 499 |
| DOUBLE SIGNED BCD-TO-BINARY BISL @BISL 472 |  <p>C: canale di controllo S: primo canale di origine D: primo canale di destinazione</p> | Convertire dati in formato BCD con segno in doppia precisione in dati binari con segno in doppia precisione.  | Uscita Richiesta | 502 |
| SIGNED BINARY-TO-BCD BCDS @BCDS 471 |  <p>C: canale di controllo S: canale di origine D: canale di destinazione</p> | Convertire un canale di dati binari con segno in un canale di dati in formato BCD con segno.  | Uscita Richiesta | 505 |
| DOUBLE SIGNED BINARY-TO-BCD BDSL @BDSL 473 |  <p>C: canale di controllo S: primo canale di origine D: primo canale di destinazione</p> | Convertire dati binari con segno in doppia precisione in dati in formato BCD con segno in doppia precisione.  | Uscita Richiesta | 507 |
| GRAY CODE CONVERSION GRY 474 (solo CPU della serie CS/CJ versione 2.0 o successiva, comprese le CPU CS1-H, CJ1-H e CJ1M con numero di lotto 030201 e successivo) |  <p>C: canale di controllo S: canale di origine D: primo canale di destinazione</p> | Convertire i dati in codice Gray nel canale specificato in dati in formato binario, BCD o sotto forma di angolo (°) alla risoluzione specificata. | Uscita Richiesta | 511 |

2-2-11 Istruzioni logiche

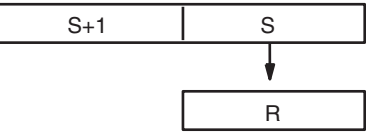
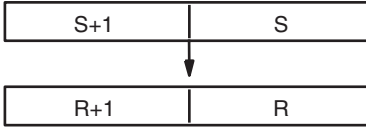
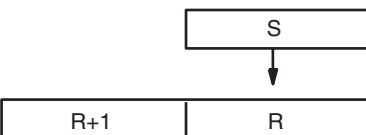
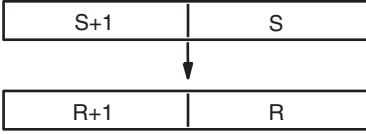
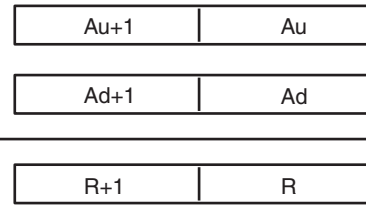
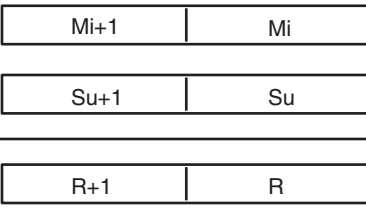
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | | | | | | | |
|---|--|-----------|--|----------------|---|--|------------------------------------|------------------------------------|--------|---|---|---|---|---|---|---|---|---|---|---|---|---------------------|-----|
| LOGICAL AND ANDW @ANDW 034 | <table border="1"> <tr><td>ANDW(034)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: canale del risultato</p> | ANDW(034) | I ₁ | I ₂ | R | <p>Assume un AND logico tra i bit corrispondenti nei canali singoli di dati canale e/o costanti.</p> <p>$I_1 \cdot I_2 \rightarrow R$</p> <table border="1"> <thead> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> | I ₁ | I ₂ | R | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Uscita Richiesta | 517 |
| ANDW(034) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | I ₂ | R | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| DOUBLE LOGICAL AND ANDL @ANDL 610 | <table border="1"> <tr><td>ANDL(610)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: canale del risultato</p> | ANDL(610) | I ₁ | I ₂ | R | <p>Assume un AND logico tra i bit corrispondenti nei canali doppi di dati canale e/o costanti.</p> <p>$(I_1, I_1+1) \cdot (I_2, I_2+1) \rightarrow (R, R+1)$</p> <table border="1"> <thead> <tr><th>I₁, I₁+1</th><th>I₂, I₂+1</th><th>R, R+1</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> | I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Uscita Richiesta | 519 |
| ANDL(610) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| LOGICAL OR ORW @ORW 035 | <table border="1"> <tr><td>ORW(035)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: canale del risultato</p> | ORW(035) | I ₁ | I ₂ | R | <p>Assume un OR logico tra i bit corrispondenti nei canali singoli di dati canale e/o costanti.</p> <p>$I_1 + I_2 \rightarrow R$</p> <table border="1"> <thead> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> | I ₁ | I ₂ | R | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | Uscita Richiesta | 520 |
| ORW(035) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | I ₂ | R | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| DOUBLE LOGICAL OR ORWL @ORWL 611 | <table border="1"> <tr><td>ORWL(611)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: canale del risultato</p> | ORWL(611) | I ₁ | I ₂ | R | <p>Assume un OR logico tra i bit corrispondenti nei canali doppi di dati canale e/o costanti.</p> <p>$(I_1, I_1+1) + (I_2, I_2+1) \rightarrow (R, R+1)$</p> <table border="1"> <thead> <tr><th>I₁, I₁+1</th><th>I₂, I₂+1</th><th>R, R+1</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> | I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | Uscita Richiesta | 522 |
| ORWL(611) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| EXCLUSIVE OR XORW @XORW 036 | <table border="1"> <tr><td>XORW(036)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: canale del risultato</p> | XORW(036) | I ₁ | I ₂ | R | <p>Assume un OR logico esclusivo tra i bit corrispondenti nei canali singoli di dati canale e/o costanti.</p> <p>$I_1 \cdot \bar{I}_2 + \bar{I}_1 \cdot I_2 \rightarrow R$</p> <table border="1"> <thead> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> | I ₁ | I ₂ | R | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | Uscita Richiesta | 524 |
| XORW(036) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | I ₂ | R | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |


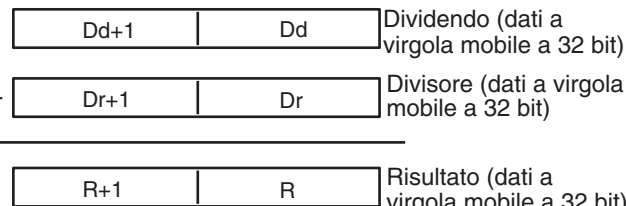

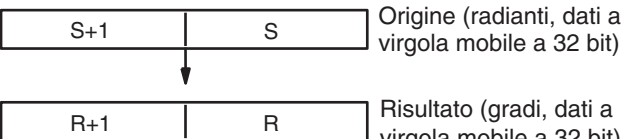

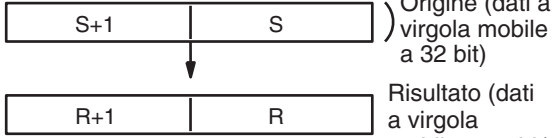
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | | | | | | | |
|---|--|-----------|--|--|---------------------|--|-----------------------------------|-----------------------------------|--------|---|---|---|---|---|---|---|---|---|---|---|---|---------------------|-----|
| DOUBLE EXCLUSIVE OR XORL @XORL 612 | <table border="1"> <tr><td>XORL(612)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: canale del risultato</p> | XORL(612) | I ₁ | I ₂ | R | <p>Assume un OR logico esclusivo tra i bit corrispondenti nei canali doppi di dati canale e/o costanti.</p> $(I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) + (I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) \rightarrow (R, R+1)$ <table border="1"> <thead> <tr> <th>I₁·I₁+1</th> <th>I₂·I₂+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> | I ₁ ·I ₁ +1 | I ₂ ·I ₂ +1 | R, R+1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | Uscita Richiesta | 526 |
| XORL(612) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ ·I ₁ +1 | I ₂ ·I ₂ +1 | R, R+1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| EXCLUSIVE NOR XNRW @XNRW 037 | <table border="1"> <tr><td>XNRW(037)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: canale del risultato</p> | XNRW(037) | I ₁ | I ₂ | R | <p>Assume un NOR logico esclusivo tra i canali singoli corrispondenti di dati canale e/o costanti.</p> $I_1 \cdot I_2 + \bar{I}_1 \cdot \bar{I}_2 \rightarrow R$ <table border="1"> <thead> <tr> <th>I₁</th> <th>I₂</th> <th>R</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table> | I ₁ | I ₂ | R | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Uscita Richiesta | 528 |
| XNRW(037) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | I ₂ | R | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | |
| DOUBLE EXCLUSIVE NOR XNRL @XNRL 613 | <table border="1"> <tr><td>XNRL(613)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: ingresso 1 I₂: ingresso 2 R: primo canale del risultato</p> | XNRL(613) | I ₁ | I ₂ | R | <p>Assume un NOR logico esclusivo tra i bit corrispondenti nei canali doppi di dati canale e/o costanti.</p> $(I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) + (\bar{I}_1 \cdot \bar{I}_1 + 1) \cdot (\bar{I}_2 \cdot \bar{I}_2 + 1) \rightarrow (R, R+1)$ <table border="1"> <thead> <tr> <th>I₁·I₁+1</th> <th>I₂·I₂+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table> | I ₁ ·I ₁ +1 | I ₂ ·I ₂ +1 | R, R+1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Uscita Richiesta | 529 |
| XNRL(613) | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ | | | | | | | | | | | | | | | | | | | | | | | |
| I ₂ | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | |
| I ₁ ·I ₁ +1 | I ₂ ·I ₂ +1 | R, R+1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | |
| COMPLEMENT COM @COM 029 | <table border="1"> <tr><td>COM(029)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: canale</p> | COM(029) | Wd | Wd → Wd: 1 → 0 e 0 → 1 | Uscita Richiesta | 531 | | | | | | | | | | | | | | | | | |
| COM(029) | | | | | | | | | | | | | | | | | | | | | | | |
| Wd | | | | | | | | | | | | | | | | | | | | | | | |
| DOUBLE COMPLEMENT COML @COML 614 | <table border="1"> <tr><td>COML(614)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: canale</p> | COML(614) | Wd | <p>Imposta su OFF tutti i bit il cui stato è ON e su ON tutti i bit il cui stato è OFF nei canali Wd e Wd+1.</p> $(Wd+1, Wd) \rightarrow (Wd+1, Wd)$ | Uscita Richiesta | 533 | | | | | | | | | | | | | | | | | |
| COML(614) | | | | | | | | | | | | | | | | | | | | | | | |
| Wd | | | | | | | | | | | | | | | | | | | | | | | |

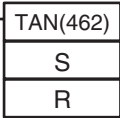
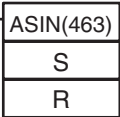
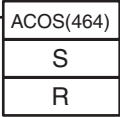
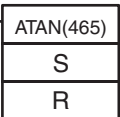
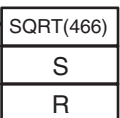
2-2-12 Istruzioni matematiche speciali

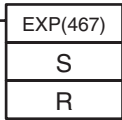
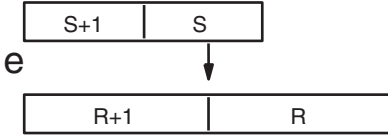
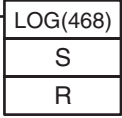
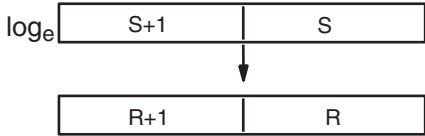
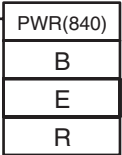
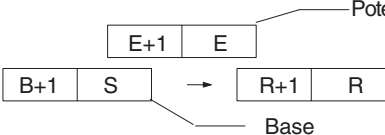



| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| BINARY ROOT ROTB @ROTB 620 |  <p>S: primo canale di origine R: canale del risultato</p> | <p>Calcola la radice quadrata del contenuto binario a 32 bit dei canali specificati e invia la parte intera del risultato al canale del risultato specificato.</p>  | Uscita Richiesta | 534 |
| BCD SQUARE ROOT ROOT @ROOT 072 |  <p>S: primo canale di origine R: canale del risultato</p> | <p>Calcola la radice quadrata di un numero in formato BCD a 8 cifre e invia la parte intera del risultato al canale del risultato specificato.</p>  | Uscita Richiesta | 536 |
| ARITHMETIC PROCESS APR @APR 069 |  <p>C: canale di controllo S: dati di origine R: canale del risultato</p> | <p>Calcola seno, coseno o un'extrapolazione lineare dei dati di origine. La funzione di extrapolazione lineare consente di approssimare relazioni tra X e Y con segmenti di linea.</p> | Uscita Richiesta | 540 |
| FLOATING POINT DIVIDE FDIV @FDIV 079 |  <p>Dd: primo canale dividendo Dr: primo canale divisore R: primo canale del risultato</p> | <p>Divide un numero a virgola mobile a 7 cifre per un altro numero. I numeri a virgola mobile sono espressi in notazione scientifica (7 cifre di mantissa e 1 cifra di esponente).</p>  | Uscita Richiesta | 552 |
| BIT COUNTER BCNT @BCNT 067 |  <p>N: numero di canali S: primo canale di origine R: canale del risultato</p> | <p>Conta il numero totale di bit impostati su ON presenti nei canali specificati.</p>  | Uscita Richiesta | 556 |

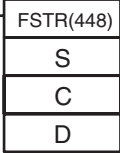
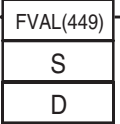
2-2-13 Istruzioni matematiche a virgola mobile

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | |
|--|---|-----------|--|-------------|--|---|---------------------|-----|
| FLOATING TO 16-BIT FIX @FIX 450 | <table border="1"> <tr><td>FIX(450)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: primo canale di origine R: canale del risultato</p> | FIX(450) | S | R | <p>Converte un valore a virgola mobile a 32 bit in dati binari a 16 bit con segno e invia il risultato al canale del risultato specificato.</p>  <p>Dati a virgola mobile (32 bit)</p> <p>Dati binari con segno (16 bit)</p> | Uscita Richiesta | 563 | |
| FIX(450) | | | | | | | | |
| S | | | | | | | | |
| R | | | | | | | | |
| FLOATING TO 32-BIT FIXL @FIXL 451 | <table border="1"> <tr><td>FIXL(451)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: primo canale di origine R: primo canale del risultato</p> | FIXL(451) | S | R | <p>Converte un valore a virgola mobile a 32 bit in dati binari a 32 bit con segno e invia il risultato ai canali del risultato specificati.</p>  <p>Dati a virgola mobile (32 bit)</p> <p>Dati binari con segno (32 bit)</p> | Uscita Richiesta | 565 | |
| FIXL(451) | | | | | | | | |
| S | | | | | | | | |
| R | | | | | | | | |
| 16-BIT TO FLOATING FLT @FLT 452 | <table border="1"> <tr><td>FLT(452)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: canale di origine R: primo canale del risultato</p> | FLT(452) | S | R | <p>Converte un valore binario a 16 bit con segno in dati a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.</p>  <p>Dati binari con segno (16 bit)</p> <p>Dati a virgola mobile (32 bit)</p> | Uscita Richiesta | 566 | |
| FLT(452) | | | | | | | | |
| S | | | | | | | | |
| R | | | | | | | | |
| 32-BIT TO FLOATING FLTL @FLTL 453 | <table border="1"> <tr><td>FLTL(453)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: primo canale di origine R: primo canale del risultato</p> | FLTL(453) | S | R | <p>Converte un valore binario a 32 bit con segno in dati a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.</p>  <p>Dati binari con segno (32 bit)</p> <p>Dati a virgola mobile (32 bit)</p> | Uscita Richiesta | 568 | |
| FLTL(453) | | | | | | | | |
| S | | | | | | | | |
| R | | | | | | | | |
| FLOATING-POINT ADD +F @+F 454 | <table border="1"> <tr><td>+F(454)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: primo canale augendo Ad: primo canale addendo R: primo canale del risultato</p> | +F(454) | Au | Ad | R | <p>Somma due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.</p>  <p>Augendo (dati a virgola mobile a 32 bit)</p> <p>Addendo (dati a virgola mobile a 32 bit)</p> <p>Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 570 |
| +F(454) | | | | | | | | |
| Au | | | | | | | | |
| Ad | | | | | | | | |
| R | | | | | | | | |
| FLOATING-POINT SUBTRACT -F @-F 455 | <table border="1"> <tr><td>F(455)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: primo canale minuendo Su: primo canale sottraendo R: primo canale del risultato</p> | F(455) | Mi | Su | R | <p>Esegue la sottrazione tra due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.</p>  <p>Minuendo (dati a virgola mobile a 32 bit)</p> <p>Sottraendo (dati a virgola mobile a 32 bit)</p> <p>Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 572 |
| F(455) | | | | | | | | |
| Mi | | | | | | | | |
| Su | | | | | | | | |
| R | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|---|--|-------------|
| FLOATING- POINT MULTIPLY *F @*F 456 | *F(456) Md Mr R Md: primo canale moltiplicando Mr: primo canale moltiplicatore R: primo canale del risultato | Moltiplica due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.  Moltiplicando (dati a virgola mobile a 32 bit) Moltiplicatore (dati a virgola mobile a 32 bit) Risultato (dati a virgola mobile a 32 bit) | Uscita Richiesta | 574 |
| FLOATING POINT DIVIDE /F @/F 457 | /F(457) Dd Dr R Dd: primo canale dividendo Dr: primo canale divisore R: primo canale del risultato | Esegue la divisione tra due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.  Dividendo (dati a virgola mobile a 32 bit) Divisore (dati a virgola mobile a 32 bit) Risultato (dati a virgola mobile a 32 bit) | Uscita Richiesta | 576 |
| DEGREES TO RADIANS RAD @RAD 458 | RAD(458) S R S: primo canale di origine R: primo canale del risultato | Converte un numero a virgola mobile a 32 bit da gradi a radianti e invia il risultato ai canali del risultato specificati.  Origine (gradi, dati a virgola mobile a 32 bit) Risultato (radianti, dati a virgola mobile a 32 bit) | Uscita Richiesta | 578 |
| RADIANS TO DEGREES DEG @DEG 459 | DEG(459) S R S: primo canale di origine R: primo canale del risultato | Converte un numero a virgola mobile a 32 bit da radianti a gradi e invia il risultato ai canali del risultato specificati.  Origine (radianti, dati a virgola mobile a 32 bit) Risultato (gradi, dati a virgola mobile a 32 bit) | Uscita Richiesta | 579 |
| SINE SIN @SIN 460 | SIN(460) S R S: primo canale di origine R: primo canale del risultato | Calcola il seno di un numero a virgola mobile a 32 bit in radianti e invia il risultato ai canali del risultato specificati.  Origine (dati a virgola mobile a 32 bit) Risultato (dati a virgola mobile a 32 bit) | Uscita Richiesta | 581 |
| COSINE COS @COS 461 | COS(461) S R S: primo canale di origine R: primo canale del risultato | Calcola il coseno di un numero a virgola mobile a 32 bit in radianti e invia il risultato ai canali del risultato specificati.  Origine (dati a virgola mobile a 32 bit) Risultato (dati a virgola mobile a 32 bit) | Uscita Richiesta | 583 |

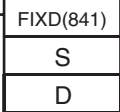
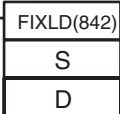
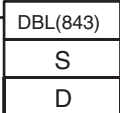
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| TANGENT TAN @TAN 462 |  <p>S: primo canale di origine R: primo canale del risultato</p> | <p>Calcola la tangente di un numero a virgola mobile a 32 bit in radianti e invia il risultato ai canali del risultato specificati.</p> $\text{TAN} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>Origine (dati a virgola mobile a 32 bit)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 585 |
| ARC SINE ASIN @ASIN 463 |  <p>S: primo canale di origine R: primo canale del risultato</p> | <p>Calcola l'arcoseno di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati. La funzione di arcoseno è l'inverso della funzione di seno e restituisce l'angolo che produce un dato valore di seno compreso tra -1 e 1.</p> $\text{SIN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>Origine (dati a virgola mobile a 32 bit)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 587 |
| ARC COSINE ACOS @ACOS 464 |  <p>S: primo canale di origine R: primo canale del risultato</p> | <p>Calcola l'arcocoseno di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati. La funzione di arcocoseno è l'inverso della funzione di coseno e restituisce l'angolo che produce un dato valore di coseno compreso tra -1 e 1.</p> $\text{COS}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>Origine (dati a virgola mobile a 32 bit)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 589 |
| ARC TANGENT ATAN @ATAN 465 |  <p>S: primo canale di origine R: primo canale del risultato</p> | <p>Calcola l'arcotangente di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati. La funzione di arcotangente è l'inverso della funzione di tangente e restituisce l'angolo che produce un dato valore di tangente.</p> $\text{TAN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>Origine (dati a virgola mobile a 32 bit)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 591 |
| SQUARE ROOT SQRT @SQRT 466 |  <p>S: primo canale di origine R: primo canale del risultato</p> | <p>Calcola la radice quadrata di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.</p> $\sqrt{\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array}}$ <p>Origine (dati a virgola mobile a 32 bit)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 593 |

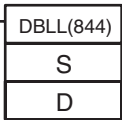
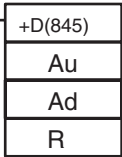
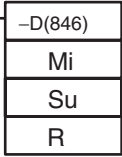
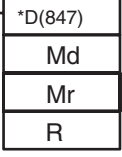
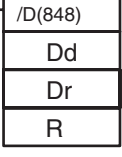
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|--|--|-------------|
| EXPONENT EXP @EXP 467 |  <p>S: primo canale di origine R: primo canale del risultato</p> | <p>Calcola il valore esponenziale naturale (in base e) di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.</p>  <p>Origine (dati a virgola mobile a 32 bit) Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 595 |
| LOGARITHM LOG @LOG 468 |  <p>S: primo canale di origine R: primo canale del risultato</p> | <p>Calcola il logaritmo naturale (in base e) di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.</p>  <p>Origine (dati a virgola mobile a 32 bit) Risultato (dati a virgola mobile a 32 bit)</p> | Uscita Richiesta | 597 |
| EXPONENTIAL POWER PWR @PWR 840 |  <p>B: primo canale base E: primo canale esponente R: primo canale del risultato</p> | <p>Eleva un numero a virgola mobile a 32 bit alla potenza di un altro numero a virgola mobile a 32 bit.</p>  <p>Potenza Base</p> | Uscita Richiesta | 599 |
| FLOATING SYMBOL COMPARISON (solo CS1-H, CJ1-H, CJ1M o CS1D) LD, AND o OR + =F (329), <>F (330), <F (331), <=F (332), >F (333), o >=F (334) | <p>Uso di LD:</p>  <p>Uso di AND:</p>  <p>Uso di OR:</p>  <p>S1: dati di confronto 1 S2: dati di confronto 2</p> | <p>Confronta i dati in singola precisione (32 bit) oppure le costanti specificate e crea una condizione di esecuzione ON se il risultato del confronto è vero.</p> <p>Con le istruzioni di confronto con simboli a virgola mobile è possibile utilizzare tre tipi di simboli: LD (LOAD), AND e OR.</p> | LD: non richiesta AND o OR: richiesta | 600 |

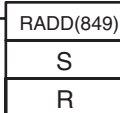
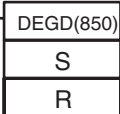
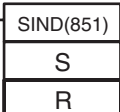
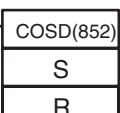
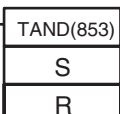
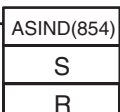
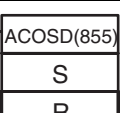
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| FLOATING- POINT TO ASCII (solo CS1-H, CJ1-H, CJ1M o CS1D) FSTR @FSTR 448 |  <p>S: primo canale di origine C: canale di controllo D: canale di destinazione</p> | Converte i dati a virgola mobile in singola precisione specificati (formato decimale o esponenziale a 32 bit) in stringhe di testo (ASCII) e invia il risultato al canale di destinazione. | Uscita richiesta | 604 |
| ASCII TO FLOA- TING-POINT (solo CS1-H, CJ1-H, CJ1M o CS1D) FVAL @FVAL 449 |  <p>S: canale di origine D: primo canale di destinazione</p> | Converte la rappresentazione in stringhe di testo specificata (ASCII) dei dati a virgola mobile in singola precisione (formato decimale o esponenziale) in dati a virgola mobile in singola precisione a 32 bit e invia il risultato ai canali di destinazione. | Uscita richiesta | 609 |

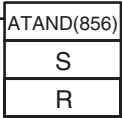
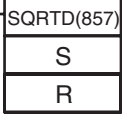
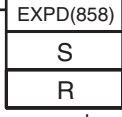
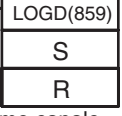
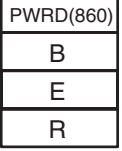
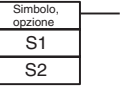

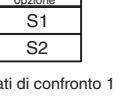
2-2-14 Istruzioni a virgola mobile in doppia precisione

Le istruzioni a virgola mobile in doppia precisione sono supportate solo dalle CPU CS1-H, CJ1-H, CJ1M o CS1D.

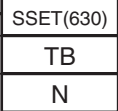
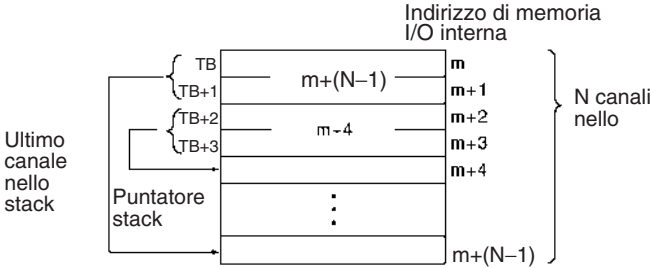
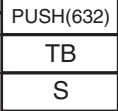
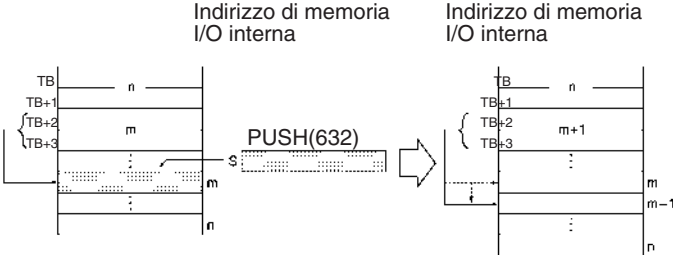
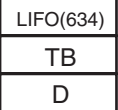
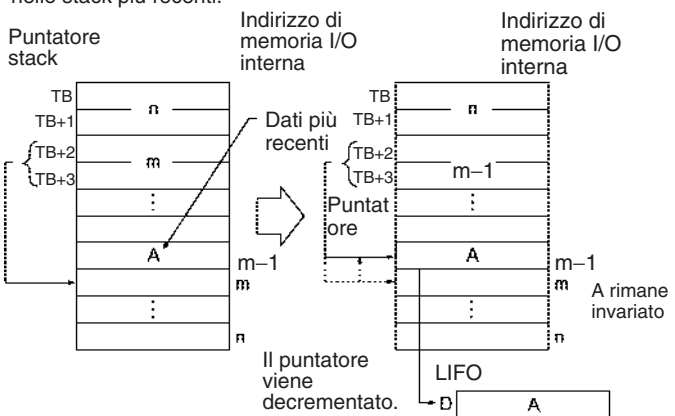
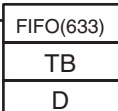
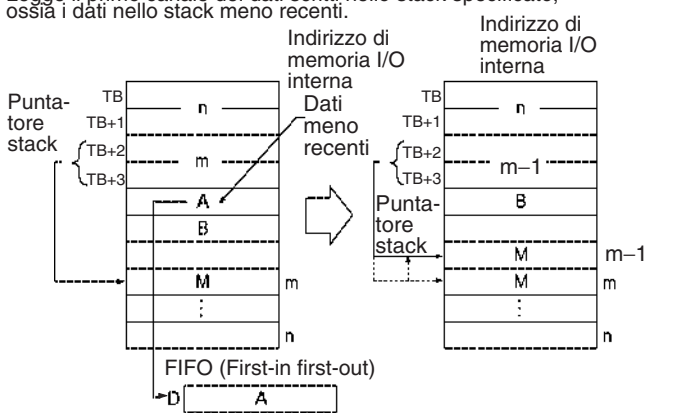
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|--|--|-------------|
| DOUBLE FLOA- TING TO 16-BIT BINARY FIXD @FIXD 841 |  <p>S: primo canale di origine D: canale di destinazione</p> | Converte i dati a virgola mobile in doppia precisione specificati (64 bit) in dati binari a 16 bit con segno e invia il risultato al canale di destinazione. | Uscita Richiesta | 620 |
| DOUBLE FLOA- TING TO 32-BIT BINARY FIXLD @FIXLD 842 |  <p>S: primo canale di origine D: primo canale di destinazione</p> | Converte i dati a virgola mobile in doppia precisione specificati (64 bit) in dati binari a 32 bit con segno e invia il risultato ai canali di destinazione. | Uscita Richiesta | 621 |
| 16-BIT BINARY TO DOUBLE FLOATING DBL @DBL 843 |  <p>S: canale di origine D: primo canale di destinazione</p> | Converte i dati binari a 16 bit con segno specificati in dati a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali di destinazione. | Uscita Richiesta | 623 |

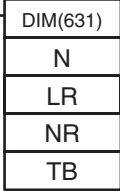
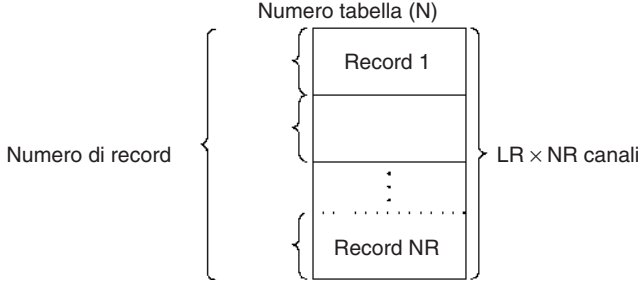
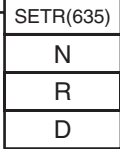
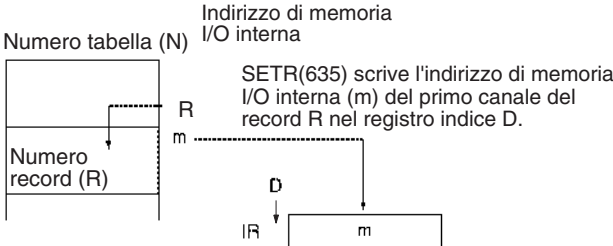
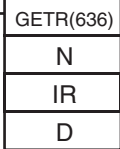
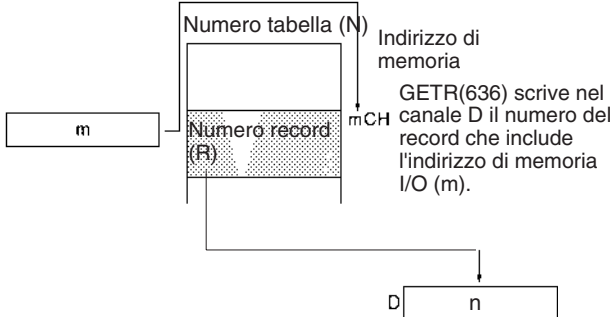
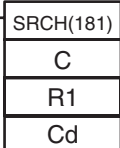
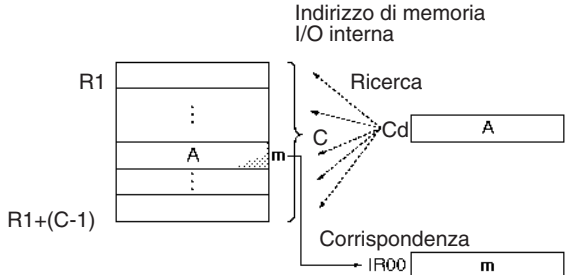
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|--|--|-------------|
| 32-BIT BINARY TO DOUBLE FLOATING DBLL @DBLL 844 |  <p>S: primo canale di origine D: primo canale di destinazione</p> | Converta i dati binari a 32 bit con segno specificati in dati a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali di destinazione. | Uscita Richiesta | 624 |
| DOUBLE FLOA- TING-POINT ADD +D @+D 845 |  <p>Au: primo canale augendo Ad: primo canale addendo R: primo canale del risultato</p> | Aggiunge i valori a virgola mobile in doppia precisione specificati (64 bit ciascuno) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 626 |
| DOUBLE FLOA- TING-POINT SUBTRACT -D @-D 846 |  <p>Mi: primo canale minuendo Su: primo canale sottraendo R: primo canale del risultato</p> | Sottrae i valori a virgola mobile in doppia precisione specificati (64 bit ciascuno) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 628 |
| DOUBLE FLOA- TING-POINT MULTIPLY *D @*D 847 |  <p>Md: primo canale moltiplicando Mr: primo canale moltiplicatore R: primo canale del risultato</p> | Moltiplica i valori a virgola mobile in doppia precisione specificati (64 bit ciascuno) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 630 |
| DOUBLE FLOA- TING-POINT DIVIDE /D @/D 848 |  <p>Dd: primo canale dividendo Dr: primo canale divisore R: primo canale del risultato</p> | Divide i valori a virgola mobile in doppia precisione specificati (64 bit ciascuno) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 632 |

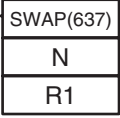
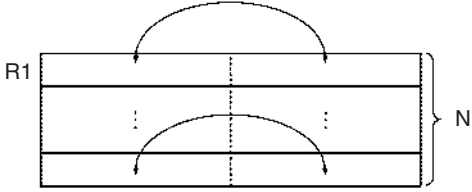
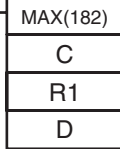
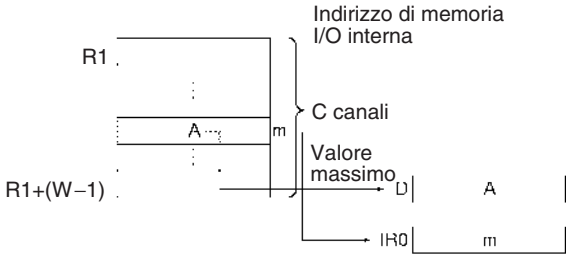
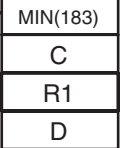
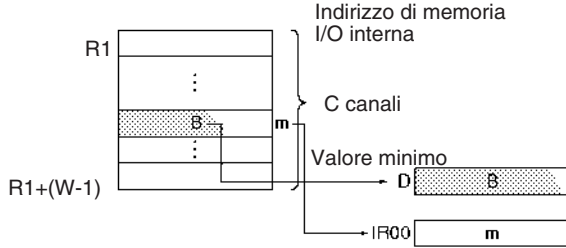
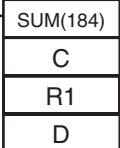
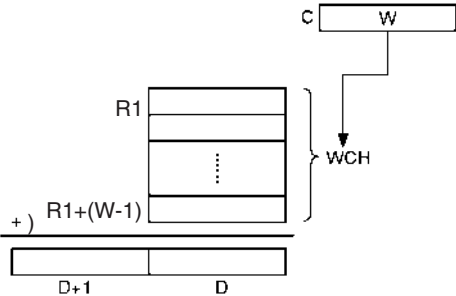
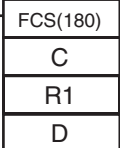
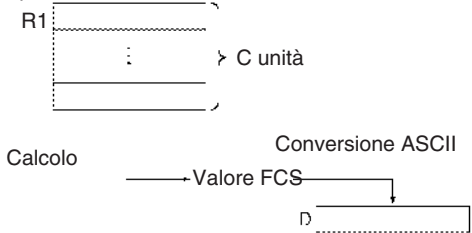
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| DOUBLE DEGREES TO RADIANS RADD @RADD 849 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Converte i dati a virgola mobile in doppia precisione specificati (64 bit) da gradi in radianti e invia il risultato ai canali dei risultati. | Uscita Richiesta | 634 |
| DOUBLE RADIANS TO DEGREES DEGD @DEGD 850 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Converte i dati a virgola mobile in doppia precisione specificati (64 bit) da radianti in gradi e invia il risultato ai canali dei risultati. | Uscita Richiesta | 636 |
| DOUBLE SINE SIND @SIND 851 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola il seno dell'angolo (in radianti) dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 637 |
| DOUBLE COSINE COSD @COSD 852 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola il coseno dell'angolo (in radianti) dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 639 |
| DOUBLE TAN- GENT TAND @TAND 853 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola la tangente dell'angolo (in radianti) dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 641 |
| DOUBLE ARC SINE ASIND @ASIND 854 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola l'angolo (in radianti) derivandolo dal valore di seno dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. La funzione di arcoseno è l'inverso della funzione di seno e restituisce l'angolo che produce un dato valore di seno compreso tra -1 e 1. | Uscita Richiesta | 643 |
| DOUBLE ARC COSINE ACOSD @ACOSD 855 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola l'angolo (in radianti) derivandolo dal valore di coseno dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. La funzione di arcocoseno è l'inverso della funzione di coseno e restituisce l'angolo che produce un dato valore di coseno compreso tra -1 e 1. | Uscita Richiesta | 645 |

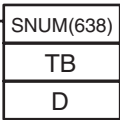
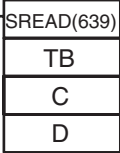
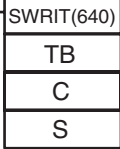
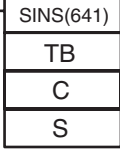
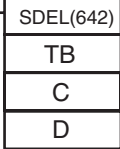
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| DOUBLE ARC TANGENT ATAND @ATAND 856 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola l'angolo (in radianti) derivandolo dal valore di tangente dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. La funzione di arcotangente è l'inverso della funzione di tangente e restituisce l'angolo che produce un dato valore di tangente. | Uscita Richiesta | 647 |
| DOUBLE SQUARE ROOT SQRTD @SQRTD 857 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola la radice quadrata dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 649 |
| DOUBLE EXPO- NENT EXPD @EXPD 858 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola il valore esponenziale naturale (in base e) dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 651 |
| DOUBLE LOGA- RITHM LOGD @LOGD 859 |  <p>S: primo canale di origine R: primo canale del risultato</p> | Calcola il logaritmo naturale (in base e) dei dati a virgola mobile in doppia precisione specificati (64 bit) e invia il risultato ai canali dei risultati. | Uscita Richiesta | 653 |
| DOUBLE EXPO- NENTIAL POWER PWRD @PWRD 860 |  <p>B: primo canale base E: primo canale esponente R: primo canale del risultato</p> | Eleva un numero a virgola mobile in doppia precisione (64 bit) alla potenza di un altro numero a virgola mobile in doppia precisione e invia il risultato ai canali dei risultati. | Uscita Richiesta | 655 |
| DOUBLE SYM- BOL COMPARI- SON LD, AND o OR + =D (335), <>D (336), <D (337), <=D (338), >D (339), o >=D (340) | <p>Uso di LD:</p>  <p>Uso di AND:</p>  <p>Uso di OR:</p>  <p>S1: dati di confronto 1 S2: dati di confronto 2</p> | Confronta i dati in doppia precisione specificati (64 bit) e crea una condizione di esecuzione ON se il risultato del confronto è vero. Con le istruzioni di confronto con simboli a virgola mobile è possibile utilizzare tre tipi di simboli: LD (LOAD), AND e OR. | LD: non richiesta AND o OR: richiesta | 657 |

2-2-15 Istruzioni di elaborazione dei dati delle tabelle

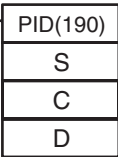
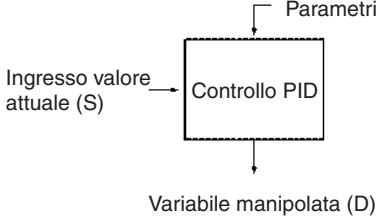
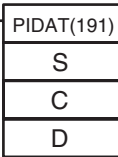
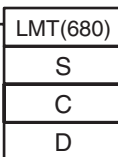
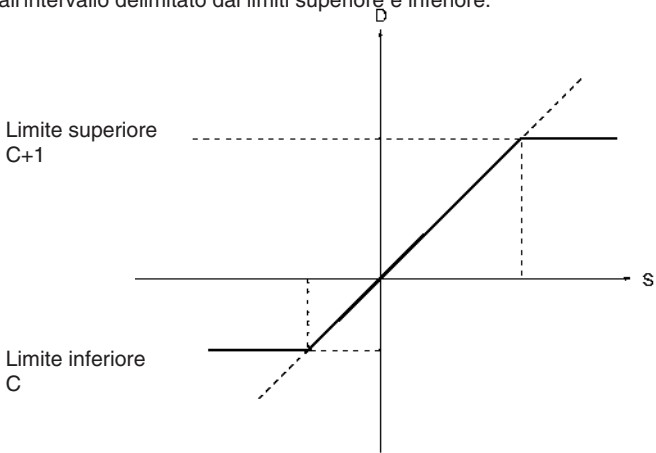
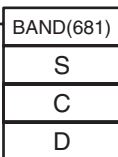
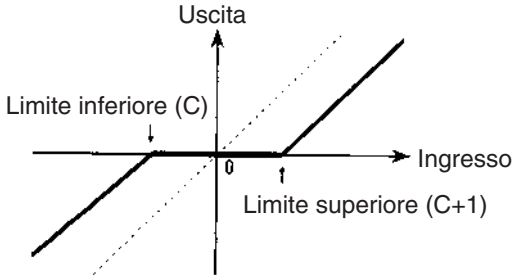
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|---|--|-------------|
| SET STACK SSET @SSET 630 |  <p>TB: primo indirizzo dello stack N: numero di canali</p> | <p>Definisce uno stack della lunghezza specificata che inizia dal canale specificato e inizializza i canali in tale area dati impostandoli su zero.</p> <p>Indirizzo di memoria I/O interna</p>  | Uscita Richiesta | 666 |
| PUSH ONTO STACK PUSH @PUSH 632 |  <p>TB: primo indirizzo dello stack S: canale di origine</p> | <p>Scrive un canale di dati nello stack specificato.</p> <p>Indirizzo di memoria I/O interna Indirizzo di memoria I/O interna</p>  | Uscita Richiesta | 669 |
| LAST IN FIRST OUT LIFO @LIFO 634 |  <p>TB: primo indirizzo dello stack D: canale di destinazione</p> | <p>Legge l'ultimo canale dei dati scritti nello stack specificato, ossia i dati nello stack più recenti.</p> <p>Puntatore stack Indirizzo di memoria I/O interna Indirizzo di memoria I/O interna</p>  <p>Dati più recenti</p> <p>Puntatore</p> <p>Il puntatore viene decrementato.</p> <p>LIFO</p> <p>A rimane invariato</p> | Uscita Richiesta | 675 |
| FIRST IN FIRST OUT FIFO @FIFO 633 |  <p>TB: primo indirizzo dello stack D: canale di destinazione</p> | <p>Legge il primo canale dei dati scritti nello stack specificato, ossia i dati nello stack meno recenti.</p> <p>Indirizzo di memoria I/O interna Indirizzo di memoria I/O interna</p>  <p>Dati meno recenti</p> <p>Puntatore stack</p> <p>FIFO (First-in first-out)</p> | Uscita Richiesta | 672 |

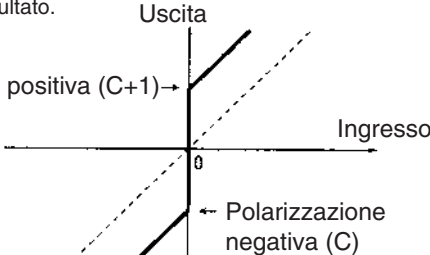
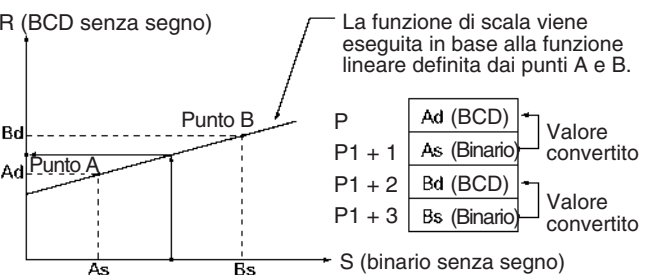
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|---|--|-------------|
| DIMENSION RECORD TABLE DIM @DIM 631 |  <p>N: numero tabella LR: lunghezza di ciascun record NR: numero di record TB: primo canale tabella</p> | Definisce una tabella di record dichiarando la lunghezza di ciascun record e il numero di record. È possibile definire un numero massimo di 16 tabelle di record.  | Uscita Richiesta | 678 |
| SET RECORD LOCATION SETR @SETR 635 |  <p>N: numero tabella R: numero di record D: registro indice di destinazione</p> | Scrive la posizione del record specificato (l'indirizzo di memoria I/O interna relativo all'inizio del record) nel registro indice specificato.  <p>SETR(635) scrive l'indirizzo di memoria I/O interna (m) del primo canale del record R nel registro indice D.</p> | Uscita Richiesta | 681 |
| GET RECORD NUMBER GETR @GETR 636 |  <p>N: numero tabella IR: registro indice D: canale di destinazione</p> | Restituisce il numero del record all'indirizzo di memoria I/O interna contenuto nel registro indice specificato.  <p>GETR(636) scrive nel canale D il numero del record che include l'indirizzo di memoria I/O (m).</p> | Uscita Richiesta | 683 |
| DATA SEARCH SRCH @SRCH 181 |  <p>C: primo canale di controllo R1: primo canale nell'intervallo Cd: dati di confronto</p> | Cerca un canale di dati all'interno di un intervallo di canali.  | Uscita Richiesta | 685 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|--|--|-------------|
| SWAP BYTES SWAP @SWAP 637 |  <p>N: numero di canali R1: primo canale nell'intervallo</p> | Scambia i byte più a sinistra e più a destra in tutti i canali dell'intervallo. La posizione del byte viene scambiata.  | Uscita Richiesta | 687 |
| FIND MAXIMUM MAX @MAX 182 |  <p>C: primo canale di controllo R1: primo canale nell'intervallo D: canale di destinazione</p> | Trova il valore massimo nell'intervallo.  | Uscita Richiesta | 689 |
| FIND MINIMUM MIN @MIN 183 |  <p>C: primo canale di controllo R1: primo canale nell'intervallo D: canale di destinazione</p> | Trova il valore minimo nell'intervallo.  | Uscita Richiesta | 693 |
| SUM SUM @SUM 184 |  <p>C: primo canale di controllo R1: primo canale nell'intervallo D: primo canale di destinazione</p> | Somma i byte o i canali nell'intervallo e invia il risultato a due canali.  | Uscita Richiesta | 697 |
| FRAME CHECK-SUM FCS @FCS 180 |  <p>C: primo canale di controllo R1: primo canale nell'intervallo D: primo canale di destinazione</p> | Calcola il valore FCS (checksum dei frame) ASCII per l'intervallo specificato.  | Uscita Richiesta | 700 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| STACK SIZE READ (solo CS1-H, CJ1-H, CJ1M o CS1D) SNUM @SNUM 638 |  <p>TB: primo indirizzo dello stack D: canale di destinazione</p> | Conta la quantità di dati di stack (numero di canali) nello stack specificato. | Uscita richiesta | 704 |
| STACK DATA READ (solo CS1-H, CJ1-H, CJ1M o CS1D) SREAD @SREAD 639 |  <p>TB: primo indirizzo dello stack C: valore di offset D: canale di destinazione</p> | Legge i dati dall'elemento specificato dello stack. Il valore di offset (numero di elementi prima della posizione corrente del puntatore) indica la posizione dell'elemento desiderato. | Uscita richiesta | 707 |
| STACK DATA OVERWRITE (solo CS1-H, CJ1-H, CJ1M o CS1D) SWRIT @SWRIT 640 |  <p>TB: primo indirizzo dello stack C: valore di offset S: dati di origine</p> | Scrive i dati di origine nell'elemento specificato dello stack, sovrascrivendo i dati già esistenti. Il valore di offset (numero di elementi prima della posizione corrente del puntatore) indica la posizione dell'elemento desiderato. | Uscita richiesta | 710 |
| STACK DATA INSERT (solo CS1-H, CJ1-H, CJ1M o CS1D) SINS @SINS 641 |  <p>TB: primo indirizzo dello stack C: valore di offset S: dati di origine</p> | Inserisce i dati di origine nella posizione specificata nello stack e fa scorrere verso il basso i dati rimanenti dello stack. Il valore di offset indica la posizione del punto d'inserimento (il numero di elementi di dati che precede la posizione corrente del puntatore). | Uscita richiesta | 713 |
| STACK DATA DELETE (solo CS1-H, CJ1-H, CJ1M o CS1D) SDEL @SDEL 642 |  <p>TB: primo indirizzo dello stack C: valore di offset D: canale di destinazione</p> | Elimina l'elemento di dati che si trova nella posizione specificata nello stack e fa scorrere verso l'alto i dati rimanenti dello stack. Il valore di offset (numero di elementi prima della posizione corrente del puntatore) indica la posizione del punto di eliminazione. | Uscita richiesta | 716 |

2-2-16 Istruzioni di controllo dei dati

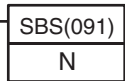
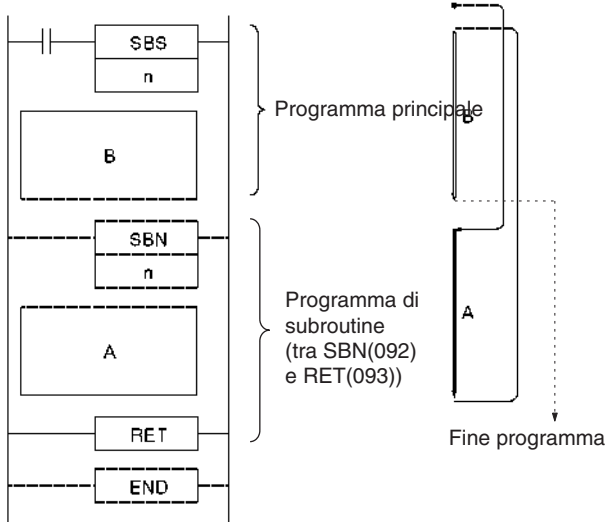
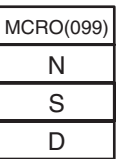
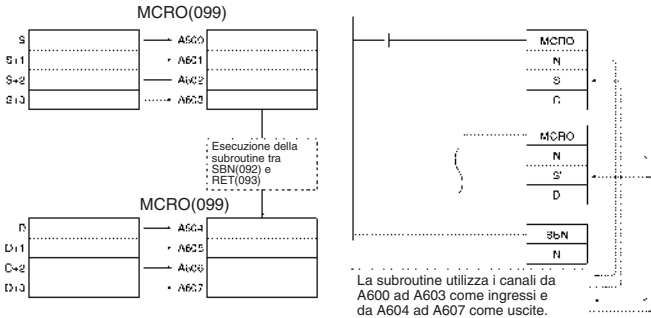
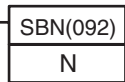
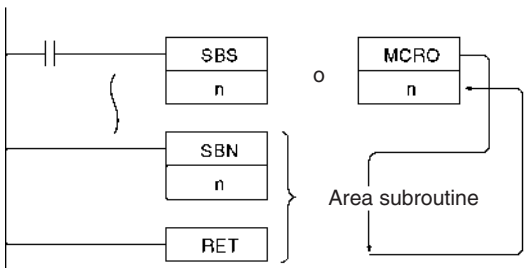
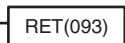
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| PID CONTROL PID 190 |  <p>S: canale di ingresso C: primo canale parametro D: canale di uscita</p> | <p>Esegue un controllo PID in base ai parametri specificati.</p> <p>Parametri (da C a C+8)</p>  | Uscita Richiesta | 720 |
| PID CONTROL WITH AUTOTUNING PIDAT 191 (solo CS1-H, CJ1-H o CJ1M) |  <p>S: canale di ingresso C: primo canale parametro D: canale di uscita</p> | <p>Esegue un controllo PID in base ai parametri specificati. PIDAT(191) consente di regolare automaticamente le costanti PID.</p> | Uscita richiesta | 731 |
| LIMIT CONTROL LMT @LMT 680 |  <p>S: canale di ingresso C: primo canale limite D: canale di uscita</p> | <p>Controlla i dati di uscita in base all'appartenenza dei dati di ingresso all'intervallo delimitato dai limiti superiore e inferiore.</p>  | Uscita Richiesta | 741 |
| DEAD BAND CONTROL BAND @BAND 681 |  <p>S: canale di ingresso C: primo canale limite D: canale di uscita</p> | <p>Controlla i dati di uscita in base all'appartenenza dei dati di ingresso all'intervallo della zona morta.</p>  | Uscita Richiesta | 743 |


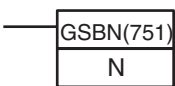

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | | |
|--|---|---------------------|--|-------------|---|--|---------------------|----------|---------------------|-------|--------------|-------|----------|---------------------|-------|--------------|---------------------|-----|
| DEAD ZONE CONTROL ZONE @ZONE 682 | <table border="1" style="margin-left: 20px;"> <tr><td style="padding: 2px;">ZONE(682)</td></tr> <tr><td style="padding: 2px;">S</td></tr> <tr><td style="padding: 2px;">C</td></tr> <tr><td style="padding: 2px;">D</td></tr> </table> <p>S: canale di ingresso C: primo canale limite D: canale di uscita</p> | ZONE(682) | S | C | D | <p>Aggiunge la polarizzazione specificata ai dati di ingresso e restituisce il risultato.</p> <p style="text-align: center;">Uscita</p>  <p style="text-align: center;">Ingresso</p> <p style="text-align: center;">0</p> <p style="text-align: right;">Polarizzazione negativa (C)</p> | Uscita Richiesta | 746 | | | | | | | | | | |
| ZONE(682) | | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | | |
| C | | | | | | | | | | | | | | | | | | |
| D | | | | | | | | | | | | | | | | | | |
| TIME-PROPORTIONAL OUTPUT TPO 685 (solo CPU della serie CS/CJ versione 2.0 o successiva). | <table border="1" style="margin-left: 20px;"> <tr><td style="padding: 2px;">TPO(685)</td></tr> <tr><td style="padding: 2px;">S</td></tr> <tr><td style="padding: 2px;">C</td></tr> <tr><td style="padding: 2px;">R</td></tr> </table> <p>S: canale di ingresso C: primo canale parametro R: bit di uscita a impulsi</p> | TPO(685) | S | C | R | <p>A partire dal rapporto di funzionamento o da una variabile manipolata contenuta nel canale specificato, converte il rapporto di funzionamento in un'uscita proporzionale al tempo in base ai parametri specificati ed emette il risultato dall'uscita specificata.</p> | Uscita Richiesta | 749 | | | | | | | | | | |
| TPO(685) | | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | | |
| C | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | |
| SCALING SCL @SCL 194 | <table border="1" style="margin-left: 20px;"> <tr><td style="padding: 2px;">SCL(194)</td></tr> <tr><td style="padding: 2px;">S</td></tr> <tr><td style="padding: 2px;">P1</td></tr> <tr><td style="padding: 2px;">R</td></tr> </table> <p>S: canale di origine P1: primo canale parametro R: canale del risultato</p> | SCL(194) | S | P1 | R | <p>Converte dati binari senza segno in dati in formato BCD senza segno in base alla funzione lineare specificata.</p> <p>R (BCD senza segno)</p>  <p>La funzione di scala viene eseguita in base alla funzione lineare definita dai punti A e B.</p> <table style="margin-left: 20px;"> <tr><td>P</td><td>Ad (BCD)</td><td rowspan="2">} Valore convertito</td></tr> <tr><td>P + 1</td><td>As (Binario)</td></tr> <tr><td>P + 2</td><td>Bd (BCD)</td><td rowspan="2">} Valore convertito</td></tr> <tr><td>P + 3</td><td>Bs (Binario)</td></tr> </table> <p>S (binario senza segno)</p> | P | Ad (BCD) | } Valore convertito | P + 1 | As (Binario) | P + 2 | Bd (BCD) | } Valore convertito | P + 3 | Bs (Binario) | Uscita Richiesta | 757 |
| SCL(194) | | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | | |
| P1 | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | |
| P | Ad (BCD) | } Valore convertito | | | | | | | | | | | | | | | | |
| P + 1 | As (Binario) | | | | | | | | | | | | | | | | | |
| P + 2 | Bd (BCD) | } Valore convertito | | | | | | | | | | | | | | | | |
| P + 3 | Bs (Binario) | | | | | | | | | | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | |
|---|--|---------------------|--|-------------|---|---|----|--------|---------------------|--------|----|---------------------|--------|----|-----------------|-----------------------------|------------|
| <p>SCALING 2</p> <p>SCL2 @SCL2 486</p> | <table border="1" style="margin-left: 20px;"> <tr><td>SCL2(486)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: canale di origine P1: primo canale parametro R: canale del risultato</p> | SCL2(486) | S | P1 | R | <p>Converte dati binari con segno in dati in formato BCD con segno in base alla funzione lineare specificata. È possibile specificare un offset nella definizione della funzione lineare.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Offset positivo</p> </div> <div style="text-align: center;"> <p>Offset negativo</p> </div> </div> <div style="margin-top: 20px;"> <p>Offset di 0000</p> <table border="1" style="margin-right: 20px;"> <tr><td>P1</td><td>Offset</td><td>(Binario con segno)</td></tr> <tr><td>P1 + 1</td><td>ΔY</td><td>(Binario con segno)</td></tr> <tr><td>P1 + 2</td><td>ΔX</td><td>(BCD con segno)</td></tr> </table> <p>Offset = 0000 esadecimale</p> </div> | P1 | Offset | (Binario con segno) | P1 + 1 | ΔY | (Binario con segno) | P1 + 2 | ΔX | (BCD con segno) | <p>Uscita Richiesta</p> | <p>762</p> |
| SCL2(486) | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | |
| P1 | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | |
| P1 | Offset | (Binario con segno) | | | | | | | | | | | | | | | |
| P1 + 1 | ΔY | (Binario con segno) | | | | | | | | | | | | | | | |
| P1 + 2 | ΔX | (BCD con segno) | | | | | | | | | | | | | | | |

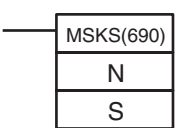
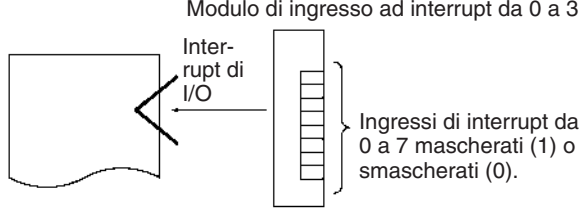
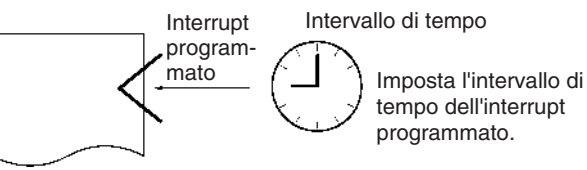
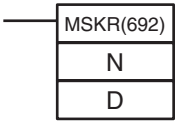
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | |
|---|--|-----------|--|-------------|---|--|-----------------------------|------------|
| <p>SCALING 3</p> <p>SCL3 @SCL3 487</p> | <table border="1" style="margin-left: 20px;"> <tr><td>SCL3(487)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: canale di origine P1: primo canale parametro R: canale del risultato</p> | SCL3(487) | S | P1 | R | <p>Converti dati in formato BCD con segno in dati binari con segno in base alla funzione lineare specificata. È possibile specificare un offset nella definizione della funzione lineare.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Offset positivo</p> </div> <div style="text-align: center;"> <p>Offset negativo</p> </div> </div> <div style="text-align: center; margin-top: 20px;"> <p>Offset di 0000</p> </div> | <p>Uscita Richiesta</p> | <p>766</p> |
| SCL3(487) | | | | | | | | |
| S | | | | | | | | |
| P1 | | | | | | | | |
| R | | | | | | | | |
| <p>AVERAGE</p> <p>AVG 195</p> | <table border="1" style="margin-left: 20px;"> <tr><td>AVG(195)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>R</td></tr> </table> <p>S: canale di origine N: numero di cicli R: canale del risultato</p> | AVG(195) | S | N | R | <p>Calcola il valore medio di un canale di ingresso per il numero di cicli specificato.</p> <p>S: canale di origine</p> <p>N: numero di cicli</p> <p>R</p> <p>R + 1 Puntatore</p> <p>Flag di media valida</p> <p>R + 2</p> <p>R + 3</p> <p>N valori</p> <p>R + N + 1</p> <p>Media</p> | <p>Uscita Richiesta</p> | <p>769</p> |
| AVG(195) | | | | | | | | |
| S | | | | | | | | |
| N | | | | | | | | |
| R | | | | | | | | |

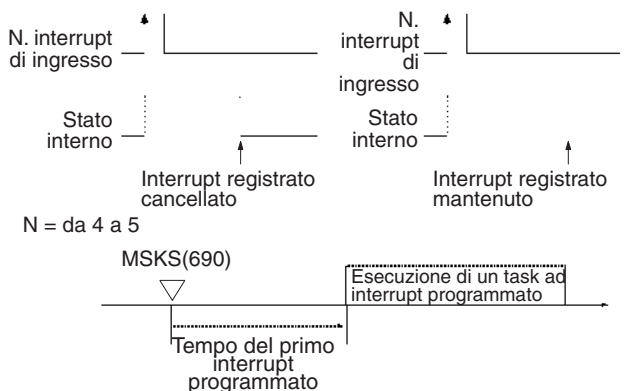
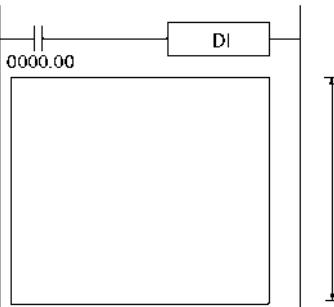
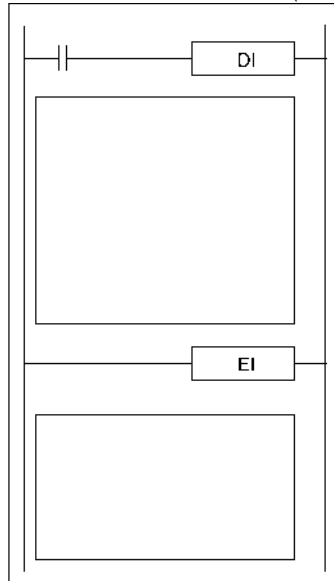
2-2-17 Istruzioni di subroutine

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|--|--|-------------|
| SUBROUTINE CALL SBS @SBS 091 |  <p>N: numero subroutine</p> | <p>Richiama la subroutine con il numero di subroutine specificato ed esegue il programma.</p> <p>Condizione di esecuzione ON</p>  | Uscita Richiesta | 773 |
| MACRO MCRO @MCRO 099 |  <p>N: numero subroutine S: primo canale parametro di ingresso D: primo canale parametro di uscita</p> | <p>Chiama la subroutine con il numero specificato ed esegue il programma utilizzando i parametri di ingresso specificati da S a S+3 e i parametri di uscita specificati da D a D+3.</p>  <p>La subroutine utilizza i canali da A600 ad A603 come ingressi e da A604 ad A607 come uscite.</p> | Uscita Richiesta | 779 |
| SUBROUTINE ENTRY SBN 092 |  <p>N: numero subroutine</p> | <p>Indica l'inizio del programma di subroutine con il numero di subroutine specificato.</p>  | Uscita Non richiesta | 783 |
| SUBROUTINE RETURN RET 093 |  | <p>Indica la fine del programma di subroutine.</p> | Uscita Non richiesta | 786 |

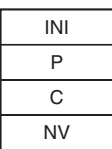
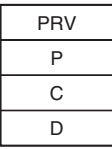
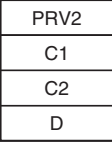
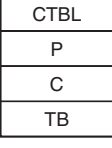
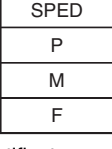
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|--|--|-------------|
| GLOBAL SUBROUTINE CALL (solo CS1-H, CJ1-H, CJ1M o CS1D) GSBS 750 |  N: numero subroutine | Richiama la subroutine con il numero di subroutine specificato ed esegue il programma. | Uscita Non richiesta | 786 |
| GLOBAL SUBROUTINE ENTRY (solo CS1-H, CJ1-H, CJ1M o CS1D) GSBN 751 |  N: numero subroutine | Indica l'inizio del programma di subroutine con il numero di subroutine specificato. | Uscita Non richiesta | 794 |
| GLOBAL SUBROUTINE RETURN (solo CS1-H, CJ1-H, CJ1M o CS1D) GRET 752 |  | Indica la fine del programma di subroutine. | Uscita Non richiesta | 797 |

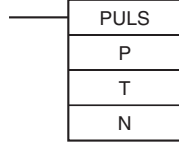
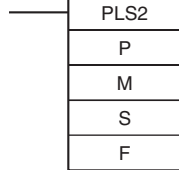
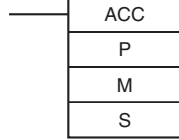
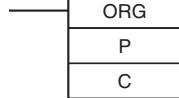
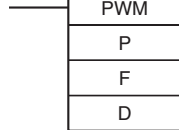
2-2-18 Istruzioni di controllo degli interrupt

| Istruzione Codice mnemonico | Simbolo/ope- rando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|--|--|-------------|
| SET INTERRUPT MASK (non supportata dalle CPU CS1D per sistemi a due CPU). MSKS @MSKS 690 |  N: identificatore di interrupt S: dati di interrupt | Imposta l'elaborazione interrupt degli interrupt I/O o programmati. Alla prima accensione del PLC, sia i task ad interrupt di I/O che programmati sono mascherati (disabilitati). Per smascherare e mascherare gli interrupt di I/O e per impostare gli intervalli di tempo per gli interrupt programmati, è possibile usare l'istruzione MSKS(690). Modulo di ingresso ad interrupt da 0 a 3  Interrupt programmato Intervallo di tempo  | Uscita Richiesta | 798 |
| READ INTERRUPT MASK (non supportata dalle CPU CS1D per sistemi a due CPU). MSKR @MSKR 692 |  N: identificatore di interrupt D: canale di destinazione | Legge le impostazioni di elaborazione dell'intervallo correnti definite tramite l'istruzione MSKS(690). | Uscita Richiesta | 805 |

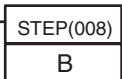
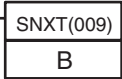
| Istruzione Codice mnemonico | Simbolo/ope- rando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|--|--|-------------|
| <p>CLEAR INTERRUPT (non supportata dalle CPU CS1D per sistemi a due CPU).</p> <p>CLI @CLI 691</p> | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> CLI(691) <hr/> N <hr/> S </div> <p>N: identificatore di interrupt S: dati di interrupt</p> | <p>Cancella o mantiene gli ingressi di interrupt registrati per gli interrupt di I/O o imposta il tempo del primo interrupt per gli interrupt programmati.</p> <p>N = da 0 a 3</p>  | Uscita Richiesta | 810 |
| <p>DISABLE INTERRUPTS</p> <p>DI @DI 693</p> | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> DI(693) </div> | <p>Disabilita l'esecuzione dei task ad interrupt eccetto quello di spegnimento.</p>  <p>Disabilita l'esecuzione dei task ad interrupt (eccetto quello di spegnimento).</p> | Uscita Richiesta | 814 |
| <p>ENABLE INTERRUPTS</p> <p>EI 694</p> | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> EI(694) </div> | <p>Abilita l'esecuzione di tutti i task ad interrupt che erano stati disabilitati tramite l'istruzione DI(693).</p>  <p>Disabilita l'esecuzione dei task ad interrupt (eccetto quello di spegnimento).</p> <p>Abilita l'esecuzione di tutti i task ad interrupt disabilitati.</p> | Uscita Non richiesta | 816 |

2-2-19 Istruzioni per contatore veloce e uscita a impulsi (solo CJ1M-CPU21/22/23)

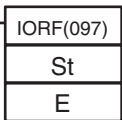
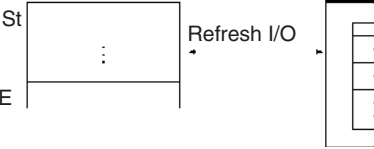
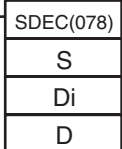
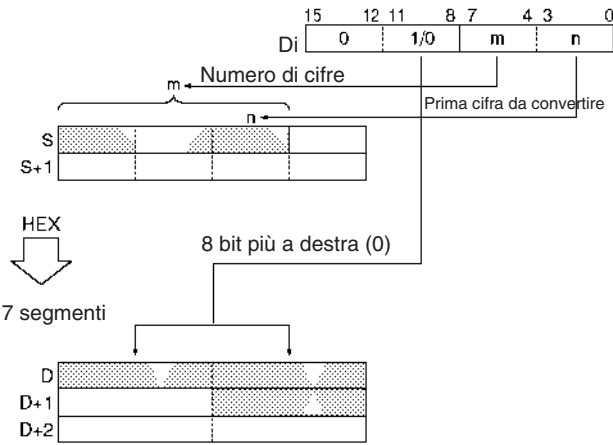
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|---|--|-------------|
| MODE CONTROL INI @INI 880 |  <p>P: identificatore della porta C: dati di controllo NV: primo canale con nuovo valore attuale</p> | L'istruzione INI(880) viene utilizzata per avviare e interrompere il confronto dei valori di riferimento, modificare il valore attuale di un contatore veloce, modificare il valore attuale di un interrupt di ingresso (modalità contatore), modificare il valore attuale di un'uscita a impulsi o arrestare l'uscita. | Uscita Richiesta | 823 |
| HIGH-SPEED COUNTER PV READ PRV @PRV 881 |  <p>P: identificatore della porta C: dati di controllo D: primo canale di destinazione</p> | L'istruzione PRV(881) viene utilizzata per leggere il valore attuale di un contatore veloce, di un'uscita a impulsi o di un interrupt di ingresso (modalità contatore). | Uscita Richiesta | 827 |
| COUNTER FREQUENCY CONVERT PRV2 883 (Solo CPU CJ1M versione 2.0 o successiva). |  <p>C1: dati di controllo C2: impulsi/rotazione D: primo canale di destinazione</p> | Legge l'ingresso della frequenza a impulsi di un contatore veloce e converte la frequenza in velocità di rotazione (numero di rotazioni) oppure converte il valore attuale del contatore nel numero totale di rotazioni. Il risultato viene inviato ai canali di destinazione in formato esadecimale a 8 cifre. Gli impulsi possono essere immessi esclusivamente dal contatore veloce 0. | Uscita Richiesta | 833 |
| COMPARISON TABLE LOAD CTBL @CTBL 882 |  <p>P: identificatore della porta C: dati di controllo TB: primo canale della tabella di confronto</p> | L'istruzione CTBL(882) viene utilizzata per eseguire confronti tra il valore attuale di un contatore veloce e valori o intervalli di riferimento. | Uscita Richiesta | 837 |
| SPEED OUTPUT SPED @SPED 885 |  <p>P: identificatore della porta M: modalità di uscita F: primo canale di frequenza degli impulsi</p> | L'istruzione SPED(885) viene utilizzata per specificare la frequenza e consentire l'emissione degli impulsi senza accelerazione o decelerazione. | Uscita Richiesta | 841 |

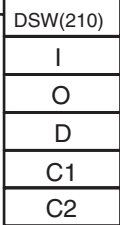


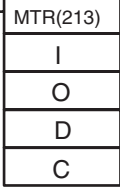
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|---|--|-------------|
| SET PULSES PULS @PULS 886 |  <p>P: identificatore della porta T: tipo di impulsi N: numero di impulsi</p> | L'istruzione PULS(886) viene utilizzata per impostare il numero di impulsi per l'uscita a impulsi. | Uscita Richiesta | 846 |
| PULSE OUTPUT PLS2 @PLS2 887 |  <p>P: identificatore della porta M: modalità di uscita S: primo canale della tabella impostazioni F: primo canale della frequenza iniziale</p> | L'istruzione PLS2(887) viene utilizzata per impostare la frequenza degli impulsi e i valori di accelerazione e decelerazione, nonché consentire l'emissione degli impulsi con accelerazione o decelerazione (con valori di accelerazione e decelerazione distinti). È previsto solo il posizionamento. | Uscita Richiesta | 849 |
| ACCELERATION CONTROL ACC @ACC 888 |  <p>P: identificatore della porta M: modalità di uscita S: primo canale della tabella impostazioni</p> | L'istruzione ACC(888) viene utilizzata per impostare la frequenza degli impulsi e i valori di accelerazione e decelerazione, nonché consentire l'emissione degli impulsi con accelerazione o decelerazione (con valori di accelerazione e decelerazione uguali). Sono previsti sia il posizionamento che il controllo della velocità. | Uscita Richiesta | 855 |
| ORIGIN SEARCH ORG @ORG 889 |  <p>P: identificatore della porta C: dati di controllo</p> | L'istruzione ORG(889) viene utilizzata per eseguire operazioni di ricerca dell'origine e ritorno all'origine. | Uscita Richiesta | 862 |
| PULSE WITH VARIABLE DUTY FACTOR PWM @ 891 |  <p>P: identificatore della porta F: frequenza D: duty-cycle</p> | L'istruzione PWM(891) viene utilizzata per l'emissione di impulsi con duty-cycle variabile. | Uscita Richiesta | 865 |

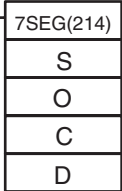
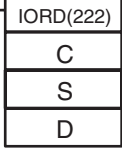
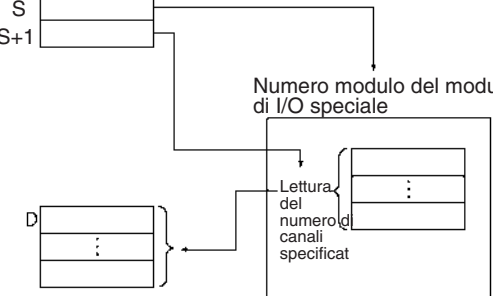
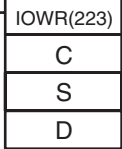
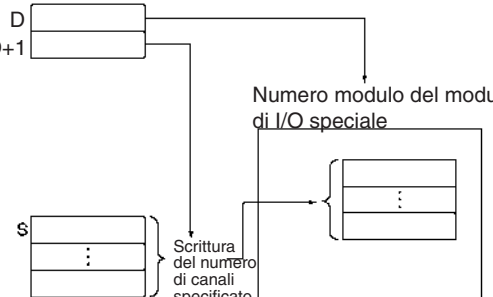
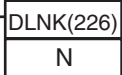
2-2-20 Istruzioni di step

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|-----------------------------------|---|--|--|-------------|
| STEP DEFINE STEP 008 |  B: bit | STEP(008) opera nei due modi indicati di seguito, a seconda della posizione e della presenza o meno di un bit di controllo specificato. (1) Avvia uno step specifico. (2) Termina l'area della programmazione step (ovvero l'esecuzione step). | Uscita Richiesta | 868 |
| STEP START SNXT 009 |  B: bit | L'istruzione SNXT(009) viene utilizzata nei seguenti modi: (1) Per avviare l'esecuzione della programmazione step. (2) Per passare al bit di controllo dello step successivo. (3) Per terminare l'esecuzione della programmazione step. | Uscita Richiesta | 868 |


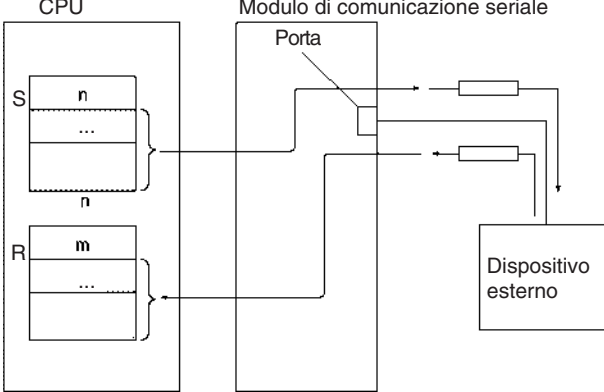
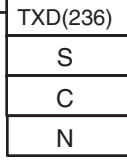
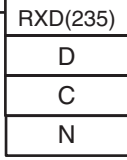
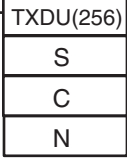
2-2-21 Istruzioni per moduli di I/O di base

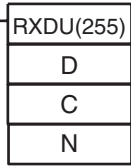
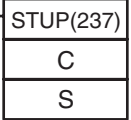
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|---|--|-------------|
| I/O REFRESH IORF @IORF 097 |  St: canale iniziale E: canale finale | Aggiorna i canali di I/O specificati. Area dei bit del modulo di I/O o del modulo di I/O speciale  Modulo di I/O o modulo di I/O speciale | Uscita Richiesta | 885 |
| 7-SEGMENT DECODER SDEC @SDEC 078 |  S: canale di origine Di: identificativo della cifra D: primo canale di destinazione | Converte il contenuto esadecimale delle cifre specificate in codici display a 7 segmenti e li memorizza negli 8 bit superiori o inferiori dei canali di destinazione specificati.  | Uscita Richiesta | 888 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|--|--|--|-------------|
| DIGITAL SWITCH INPUT DSW 210 (solo CPU della serie CS/CJ ver- sione 2.0 o suc- cessiva). |  <p>I: canale di ingresso dati (da D0 a D3) O: canale di uscita D: primo canale del risultato C1: numero di cifre C2: canale di sistema</p> | Legge il valore impostato su un selettore digitale esterno (o selettore rotativo) collegato a un modulo di ingresso o di uscita e memorizza i dati in formato BCD a 4 o 8 cifre nei canali specificati. | Uscita Richiesta | 890 |
| TEN KEY INPUT TKY 211 (solo CPU della serie CS/CJ ver- sione 2.0 o suc- cessiva). |  <p>I: canale di ingresso dati D1: primo canale registro D2: canale di immissione</p> | Legge i dati numerici da un tastierino a dieci tasti collegato a un modulo di ingresso e memorizza fino a 8 cifre di dati in formato BCD nei canali specificati. | Uscita Richiesta | 896 |
| HEXADECIMAL KEY INPUT HKY 212 (solo CPU della serie CS/CJ ver- sione 2.0 o suc- cessiva). |  <p>I: canale di ingresso dati O: canale di uscita D: primo canale registro</p> | Legge i dati numerici da un tastierino esadecimale collegato a un modulo di ingresso e a un modulo di uscita e memorizza fino a 8 cifre di dati in formato esadecimale nei canali specificati. | Uscita Richiesta | 899 |
| MATRIX INPUT MTR 213 (solo CPU della serie CS/CJ ver- sione 2.0 o suc- cessiva). |  <p>I: canale di ingresso dati O: canale di uscita D: primo canale di destinazione C: canale di sistema</p> | Accetta fino a 64 segnali da una matrice 8 x 8 collegata a un modulo di ingresso e a un modulo di uscita (utilizzando 8 punti di ingresso e 8 punti di uscita) e memorizza i dati a 64 bit nei 4 canali di destinazione. | Uscita Richiesta | 904 |

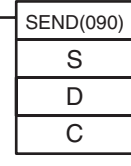
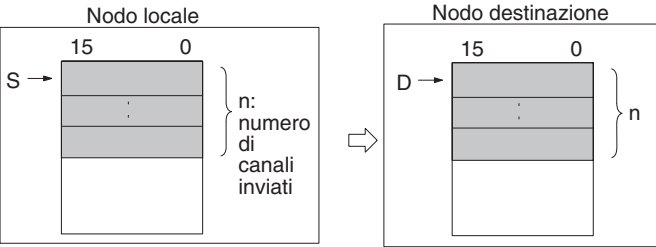
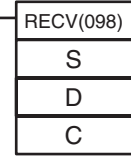
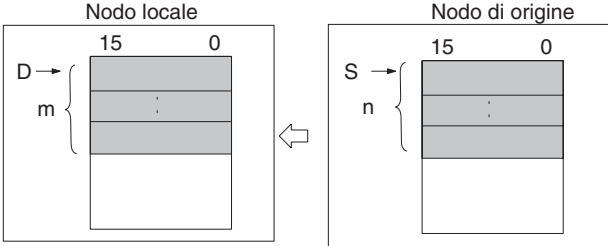
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|---|--|-------------|
| 7-SEGMENT DISPLAY OUTPUT 7SEG 214 (solo CPU della serie CS/CJ ver- sione 2.0 o suc- cessiva). |  <p>S: primo canale di origine O: canale di uscita C: dati di controllo D: canale di sistema</p> | Converte i dati di origine (a 4 o 8 cifre in formato BCD) nel formato per il display a 7 segmenti, quindi emette tali dati sul canale di uscita specificato. | Uscita Richiesta | 908 |
| INTELLIGENT I/O READ IORD @IORD 222 |  <p>C: dati di controllo S: origine trasferimento e numero di canali D: destinazione trasferimento e numero di canali</p> | Legge il contenuto dell'area di memoria del modulo di I/O speciale o dell'unità bus CPU (vedere nota).  <p>Nota: Solo CPU della serie CS/CJ versione 2.0 o successiva (comprese le CPU CS1-H, CJ1-H e CJ1M con numero di lotto 030418 o successivo) possono leggere dalle unità bus CPU.</p> | Uscita Richiesta | 913 |
| INTELLIGENT I/O WRITE IOWR @IOWR 223 |  <p>C: dati di controllo S: origine trasferimento e numero di canali D: destinazione trasferimento e numero di canali</p> | Invia il contenuto dell'area di memoria I/O della CPU al modulo di I/O speciale o all'unità bus CPU (vedere nota).  <p>Nota: Solo CPU della serie CS/CJ versione 2.0 o successiva (comprese le CPU CS1-H, CJ1-H e CJ1M con numero di lotto 030418 o successivo) possono scrivere nelle unità bus CPU.</p> | Uscita Richiesta | 917 |
| CPU BUS UNIT I/O REFRESH (solo CS1-H, CJ1-H, CJ1M o CS1D) DLNK @DLNK 226 |  <p>N: numero modulo</p> | Esegue immediatamente l'aggiornamento degli I/O dell'unità bus CPU con il numero di unità specificato. | Uscita richie- sta | 921 |

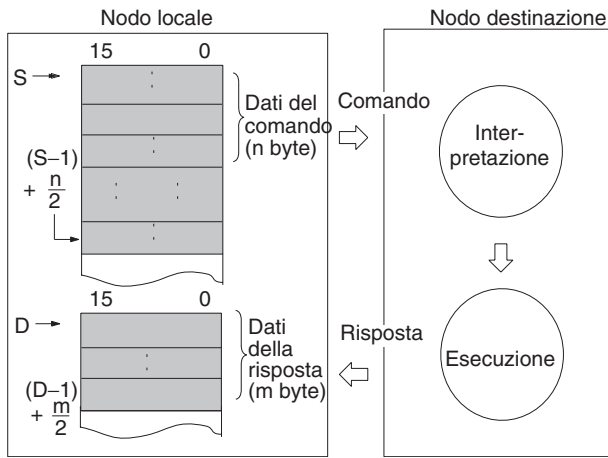
2-2-22 Istruzioni per la comunicazione seriale

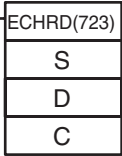
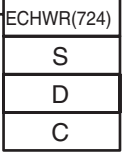
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|--|--|-------------|
| PROTOCOL MACRO PMCR @PMCR 260 |  <p>C1: canale di controllo 1 C2: canale di controllo 2 S: primo canale inviato R: primo canale ricevuto</p> | <p>Richiama ed esegue una sequenza di comunicazione registrata in un modulo di comunicazione seriale o in una scheda di comunicazione seriale (solo serie CS).</p>  | Uscita Richiesta | 928 |
| TRANSMIT TXD @TXD 236 |  <p>S: primo canale di origine C: canale di controllo N: numero di byte da 0000 a 0100 esadecimale (da 0 a 256 decimale)</p> | <p>Invia il numero di byte di dati specificato dalla porta RS-232C integrata nella CPU o dalla porta seriale di una scheda di comunicazione seriale (versione 1.2 o successiva).</p> | Uscita Richiesta | 937 |
| RECEIVE RXD @RXD 235 |  <p>D: primo canale di destinazione C: canale di controllo N: numero di byte da memorizzare da 0000 a 0100 esadecimale (da 0 a 256 decimale)</p> | <p>Legge il numero di byte di dati specificato dalla porta RS-232C integrata nella CPU o dalla porta seriale di una scheda di comunicazione seriale (versione 1.2 o successiva).</p> | Uscita Richiesta | 944 |
| TRANSMIT VIA SERIAL COMMU- NICATIONS UNIT TXDU @TXDU 256 |  <p>S: primo canale di origine C: primo canale di controllo N: numero di byte da 0000 a 0256 BCD</p> | <p>Invia il numero di byte di dati specificato dalla porta seriale di un modulo di comunicazione seriale (versione del modulo 1.2 o successiva). I dati vengono inviati in modalità senza protocollo con gli eventuali codici di inizio e di fine specificati nell'area di configurazione DM allocata.</p> | Uscita Richiesta | 952 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|---|--|-------------|
| RECEIVE VIA SERIAL COMMU- NICATIONS UNIT RXDU @RXDU 255 |  <p>D: primo canale di destinazione C: primo canale di controllo N: numero di byte da 0000 a 0256 BCD</p> | Legge il numero di byte di dati specificato dalla porta seriale di un modulo di comunicazione seriale (versione del modulo 1.2 o successiva). I dati vengono letti in modalità senza protocollo con gli eventuali codici di inizio e di fine specificati nell'area di configurazione DM allocata. | Uscita Richiesta | 960 |
| CHANGESERIAL PORT SETUP STUP @STUP 237 |  <p>C: canale di controllo (porta) S: primo canale di origine</p> | Modifica i parametri di comunicazione di una porta seriale sulla CPU, sul modulo di comunicazione seriale (unità bus CPU) o sulla scheda di comunicazione seriale. L'istruzione STUP(237) consente pertanto di modificare la modalità di protocollo durante il funzionamento del PLC. | Uscita Richiesta | 968 |

2-2-23 Istruzioni di rete

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|--|--|-------------|
| NETWORK SEND SEND @SEND 090 |  <p>S: primo canale di origine D: primo canale di destinazione C: primo canale di controllo</p> | Trasmette i dati a un nodo nella rete.  | Uscita Richiesta | 991 |
| NETWORK RECEIVE RECV @RECV 098 |  <p>S: primo canale di origine D: primo canale di destinazione C: primo canale di controllo</p> | Richiede la trasmissione di dati da un nodo nella rete e riceve tali dati.  | Uscita Richiesta | 997 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | |
|---|---|------------|--|-------------|---|--|---------------------|------|
| DELIVER COMMAND CMND @CMND 490 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>CMND(490)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>S: primo canale di comando D: primo canale di risposta C: primo canale di controllo</p> | CMND(490) | S | D | C | Invia comandi FINS e riceve la risposta.  <p style="text-align: center;">Nodo locale Nodo destinazione</p> <p>S → [15 0] [15 0] Comando → Interpretazione</p> <p>(S-1) [15 0] ↓</p> <p>+ $\frac{n}{2}$ [15 0] Esecuzione</p> <p>← Risposta ← [15 0]</p> <p>D → [15 0] [15 0]</p> <p>(D-1) [15 0]</p> <p>+ $\frac{m}{2}$ [15 0]</p> <p style="text-align: right;">Dati del comando (n byte)</p> <p style="text-align: right;">Dati della risposta (m byte)</p> | Uscita Richiesta | 1003 |
| CMND(490) | | | | | | | | |
| S | | | | | | | | |
| D | | | | | | | | |
| C | | | | | | | | |
| EXPLICIT MES- SAGE SEND EXPLT 720 (solo CPU della serie CS/CJ versione 2.0 o successiva). | <table border="1" style="width: 100%; text-align: center;"> <tr><td>EXPLT(720)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>S: primo canale del messaggio di invio D: primo canale del messaggio ricevuto C: primo canale di controllo</p> | EXPLT(720) | S | D | C | Invia un messaggio esplicito con qualsiasi codice di servizio. | Uscita Richiesta | 1013 |
| EXPLT(720) | | | | | | | | |
| S | | | | | | | | |
| D | | | | | | | | |
| C | | | | | | | | |
| EXPLICIT GET ATTRIBUTE EGATR 721 (solo CPU della serie CS/CJ versione 2.0 o successiva). | <table border="1" style="width: 100%; text-align: center;"> <tr><td>EGATR(721)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>S: primo canale del messaggio di invio D: primo canale del messaggio ricevuto C: primo canale di controllo del messaggio</p> | EGATR(721) | S | D | C | Legge le informazioni sullo stato con un messaggio esplicito (Get Attribute Single, codice di servizio: 0E esadecimale). | Uscita Richiesta | 1021 |
| EGATR(721) | | | | | | | | |
| S | | | | | | | | |
| D | | | | | | | | |
| C | | | | | | | | |
| EXPLICIT SET ATTRIBUTE ESATR 722 (solo CPU della serie CS/CJ versione 2.0 o successiva). | <table border="1" style="width: 100%; text-align: center;"> <tr><td>ESATR(722)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> </table> <p>S: primo canale del messaggio di invio C: primo canale di controllo</p> | ESATR(722) | S | C | Scrive le informazioni sullo stato con un messaggio esplicito (Set Attribute Single, codice di servizio: 0E esadecimale). | Uscita Richiesta | 1028 | |
| ESATR(722) | | | | | | | | |
| S | | | | | | | | |
| C | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|---|--|--|-------------|
| EXPLICIT WORD READ ECHRD 723 (solo CPU della serie CS/CJ ver- sione 2.0 o suc- cessiva). |  <p>S: primo canale di origine nella CPU remota D: primo canale di destinazione nella CPU locale C: primo canale di controllo</p> | Legge dati sulla CPU locale da una CPU remota collegata in rete (la CPU remota deve supportare i messaggi espliciti). | Uscita Richiesta | 1034 |
| EXPLICIT WORD WRITE ECHWR 724 (solo CPU della serie CS/CJ ver- sione 2.0 o suc- cessiva). |  <p>S: primo canale di origine nella CPU locale D: primo canale di destinazione nella CPU remota C: primo canale di controllo</p> | Scrive dati dalla CPU locale in una CPU remota collegata in rete (la CPU remota deve supportare i messaggi espliciti). | Uscita Richiesta | 1038 |

2-2-24 Istruzioni per la memoria dei file

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | |
|--|---|------------|--|-------------|----|---|---|-----------------------------|-------------|
| <p>READ DATA FILE FREAD @FREAD 700</p> | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="text-align: center;">FREAD(700)</td></tr> <tr><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;">S1</td></tr> <tr><td style="text-align: center;">S2</td></tr> <tr><td style="text-align: center;">D</td></tr> </table> <p>C: canale di controllo S1: primo canale di origine S2: nome file D: primo canale di destinazione</p> | FREAD(700) | C | S1 | S2 | D | <p>Legge la quantità di dati o i dati specificati dal file di dati indicato nella memoria dei file nell'area dati specificata della CPU.</p> <p>Scheda di memoria o memoria dei file nell'area EM (specificata nella quarta cifra di C).</p> <p>Scheda di memoria o memoria dei file nell'area EM (specificata nella quarta cifra di C).</p> | <p>Uscita Richiesta</p> | <p>1045</p> |
| FREAD(700) | | | | | | | | | |
| C | | | | | | | | | |
| S1 | | | | | | | | | |
| S2 | | | | | | | | | |
| D | | | | | | | | | |
| <p>WRITE DATA FILE FWRIT @FWRIT 701</p> | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="text-align: center;">FWRIT(701)</td></tr> <tr><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;">D1</td></tr> <tr><td style="text-align: center;">D2</td></tr> <tr><td style="text-align: center;">S</td></tr> </table> <p>C: canale di controllo D1: primo canale di destinazione D2: nome file S: primo canale di origine</p> | FWRIT(701) | C | D1 | D2 | S | <p>Sovrascrive o aggiunge i dati specificati ai dati contenuti nel file di dati indicato nella memoria dei file dall'area dati della CPU. Se il file indicato non esiste, viene creato un nuovo file con tale nome.</p> <p>Scheda di memoria o memoria dei file nell'area EM (specificata nella quarta cifra di C).</p> <p>Scheda di memoria o memoria dei file nell'area EM (specificata nella quarta cifra di C).</p> <p>Scheda di memoria o memoria dei file nell'area EM (specificata nella quarta cifra di C).</p> | <p>Uscita Richiesta</p> | <p>1052</p> |
| FWRIT(701) | | | | | | | | | |
| C | | | | | | | | | |
| D1 | | | | | | | | | |
| D2 | | | | | | | | | |
| S | | | | | | | | | |

2-2-25 Istruzioni di visualizzazione

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | |
|--|---|----------|--|-------------|--|---------------------|------|
| DISPLAY MESSAGE MSG @MSG 046 | <table border="1"> <tr><td>MSG(046)</td></tr> <tr><td>N</td></tr> <tr><td>M</td></tr> </table> <p>N: numero del messaggio M: primo canale messaggio</p> | MSG(046) | N | M | Legge i sedici canali di codice ASCII esteso specificati e visualizza il messaggio su un dispositivo periferico quale una console di programmazione. | Uscita Richiesta | 1058 |
| MSG(046) | | | | | | | |
| N | | | | | | | |
| M | | | | | | | |

2-2-26 Istruzioni per l'orologio

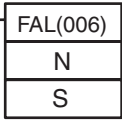
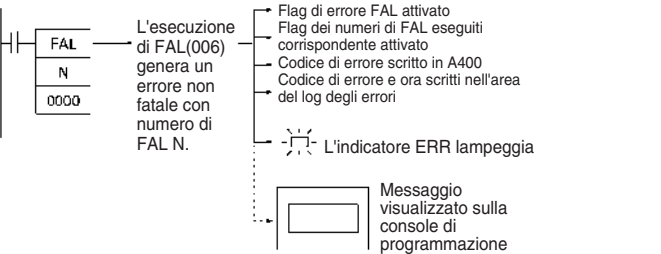
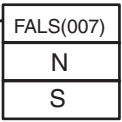
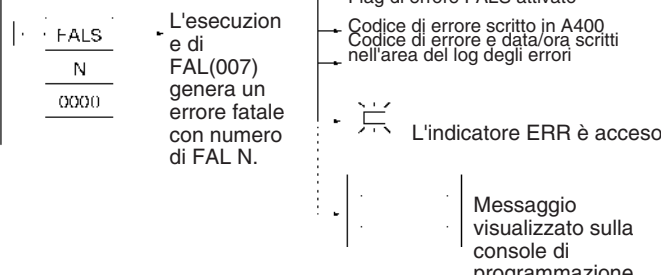
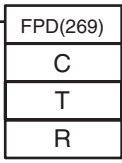
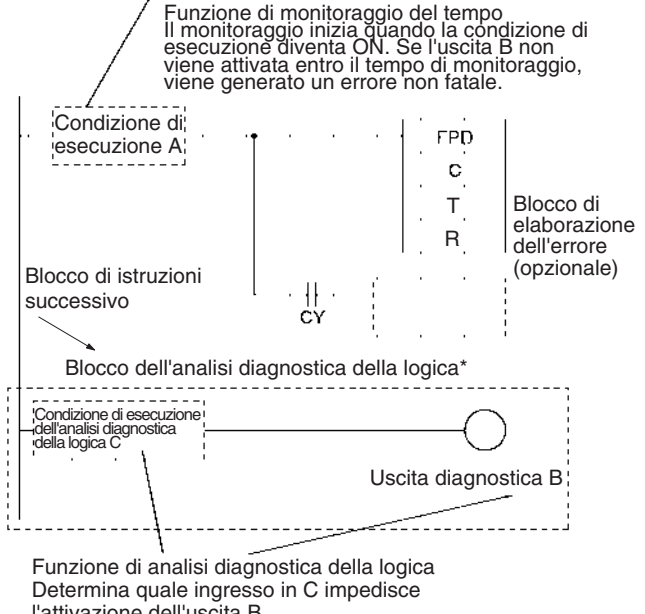
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|-----------|--|-------------|---|---|----|----|---|---|--------|---------|-----|--------|-----|-----|------|------|----|----|---|---|--------|---------|-----|-----|--|----|----|---|---|--------|---------|-----|--------|-----|-----|------|------|---------------------|------|
| CALENDAR ADD CADD @CADD 730 | <table border="1"> <tr><td>CADD(730)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C: primo canale del calendario T: primo canale dell'ora R: primo canale del risultato</p> | CADD(730) | C | T | R | <p>Aggiunge tempo ai dati del calendario nei canali specificati.</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>C</td><td>Minuti</td><td>Secondi</td></tr> <tr><td>C+1</td><td>Giorno</td><td>Ora</td></tr> <tr><td>C+2</td><td>Anno</td><td>Mese</td></tr> </table> <p style="text-align: center;">+</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>T</td><td>Minuti</td><td>Secondi</td></tr> <tr><td>T+1</td><td colspan="2">Ore</td></tr> </table> <p style="text-align: center;">↓</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>R</td><td>Minuti</td><td>Secondi</td></tr> <tr><td>R+1</td><td>Giorno</td><td>Ora</td></tr> <tr><td>R+2</td><td>Anno</td><td>Mese</td></tr> </table> | 15 | 87 | 0 | C | Minuti | Secondi | C+1 | Giorno | Ora | C+2 | Anno | Mese | 15 | 87 | 0 | T | Minuti | Secondi | T+1 | Ore | | 15 | 87 | 0 | R | Minuti | Secondi | R+1 | Giorno | Ora | R+2 | Anno | Mese | Uscita Richiesta | 1061 |
| CADD(730) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 87 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | Minuti | Secondi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C+1 | Giorno | Ora | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C+2 | Anno | Mese | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 87 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T | Minuti | Secondi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T+1 | Ore | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 87 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | Minuti | Secondi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R+1 | Giorno | Ora | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R+2 | Anno | Mese | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CALENDAR SUBTRACT CSUB @CSUB 731 | <table border="1"> <tr><td>CSUB(731)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C: primo canale del calendario T: primo canale dell'ora R: primo canale del risultato</p> | CSUB(731) | C | T | R | <p>Sottrae tempo dai dati del calendario nei canali specificati.</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>C</td><td>Minuti</td><td>Secondi</td></tr> <tr><td>C+1</td><td>Giorno</td><td>Ora</td></tr> <tr><td>C+2</td><td>Anno</td><td>Mese</td></tr> </table> <p style="text-align: center;">-</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>T</td><td>Minuti</td><td>Secondi</td></tr> <tr><td>T+1</td><td colspan="2">Ore</td></tr> </table> <p style="text-align: center;">↓</p> <table style="margin-left: 40px;"> <tr><td>15</td><td>87</td><td>0</td></tr> <tr><td>R</td><td>Minuti</td><td>Secondi</td></tr> <tr><td>R+1</td><td>Giorno</td><td>Ora</td></tr> <tr><td>R+2</td><td>Anno</td><td>Mese</td></tr> </table> | 15 | 87 | 0 | C | Minuti | Secondi | C+1 | Giorno | Ora | C+2 | Anno | Mese | 15 | 87 | 0 | T | Minuti | Secondi | T+1 | Ore | | 15 | 87 | 0 | R | Minuti | Secondi | R+1 | Giorno | Ora | R+2 | Anno | Mese | Uscita Richiesta | 1065 |
| CSUB(731) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 87 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | Minuti | Secondi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C+1 | Giorno | Ora | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C+2 | Anno | Mese | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 87 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T | Minuti | Secondi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T+1 | Ore | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 87 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | Minuti | Secondi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R+1 | Giorno | Ora | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R+2 | Anno | Mese | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | | | | | | |
|---|--|-----------------|--|---|--|---------------------|---------|-----|--------|-----|-----|------|------|-----|----|-----------------|---------------------|------|
| HOURS TO SECONDS SEC @SEC 065 | <table border="1"> <tr><td>SEC(065)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: primo canale di origine D: primo canale di destinazione</p> | SEC(065) | S | D | <p>Converte i dati temporali in formato ore/minuti/secondi in un tempo equivalente espresso solo in secondi.</p> | Uscita Richiesta | 1068 | | | | | | | | | | | |
| SEC(065) | | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | | |
| D | | | | | | | | | | | | | | | | | | |
| SECONDS TO HOURS HMS @HMS 066 | <table border="1"> <tr><td>HMS(066)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: primo canale di origine D: primo canale di destinazione</p> | HMS(066) | S | D | <p>Converte i dati temporali espressi in secondi in un tempo equivalente in formato ore/minuti/secondi.</p> | Uscita Richiesta | 1070 | | | | | | | | | | | |
| HMS(066) | | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | | |
| D | | | | | | | | | | | | | | | | | | |
| CLOCK ADJUSTMENT DATE @DATE 735 | <table border="1"> <tr><td>DATE(735)</td></tr> <tr><td>S</td></tr> </table> <p>S: primo canale di origine</p> | DATE(735) | S | <p>Modifica l'impostazione dell'orologio interno in base all'impostazione dei canali di origine specificati.</p> <p>CPU</p> <table border="1"> <tr><td>S1</td><td>Minuti</td><td>Secondi</td></tr> <tr><td>S+1</td><td>Giorno</td><td>Ora</td></tr> <tr><td>S+2</td><td>Anno</td><td>Mese</td></tr> <tr><td>S+3</td><td>00</td><td>Giorno d. sett.</td></tr> </table> <p>Nuova impostazione</p> | S1 | Minuti | Secondi | S+1 | Giorno | Ora | S+2 | Anno | Mese | S+3 | 00 | Giorno d. sett. | Uscita Richiesta | 1073 |
| DATE(735) | | | | | | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | | | | | | |
| S1 | Minuti | Secondi | | | | | | | | | | | | | | | | |
| S+1 | Giorno | Ora | | | | | | | | | | | | | | | | |
| S+2 | Anno | Mese | | | | | | | | | | | | | | | | |
| S+3 | 00 | Giorno d. sett. | | | | | | | | | | | | | | | | |

2-2-27 Istruzioni di debug

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | |
|---|---|-----------|---|-------------------------|------|
| TRACE MEMORY SAMPLING TRSM 045 | <table border="1"> <tr><td>TRSM(045)</td></tr> </table> | TRSM(045) | <p>Quando viene eseguita l'istruzione TRSM(045), lo stato del bit o del canale preselezionato viene campionato e memorizzato nella memoria di registrazione. È possibile utilizzare TRSM(045) in qualsiasi punto del programma e un numero di volte illimitato.</p> | Uscita Non richiesta | 1075 |
| TRSM(045) | | | | | |

2-2-28 Istruzioni di diagnostica per malfunzionamento

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina |
|--|---|--|--|--------|
| FAILURE ALARM FAL @ FAL 006 |  <p>N: numero FAL S: primo canale del messaggio o codice di errore da generare</p> | <p>Genera o cancella errori non fatali definiti dall'utente. Gli errori non fatali non causano l'arresto del PLC. Inoltre genera errori non fatali nel sistema.</p>  | Uscita Richiesta | 1079 |
| SEVERE FAILURE ALARM FALS 007 |  <p>N: numero di FALS S: primo canale del messaggio o codice di errore da generare</p> | <p>Genera errori fatali definiti dall'utente. Gli errori fatali causano l'arresto del PLC. Inoltre genera errori fatali nel sistema.</p>  | Uscita Richiesta | 1087 |
| FAILURE POINT DETECTION FPD 269 |  <p>C: canale di controllo T: tempo di monitoraggio R: primo canale registro</p> | <p>L'istruzione FPD(269) consente di rilevare un errore in un blocco di istruzioni monitorando il tempo che intercorre tra l'esecuzione dell'istruzione stessa e l'attivazione di un'uscita di diagnostica, individuando in tal modo l'ingresso che impedisce l'attivazione di un'uscita.</p> <p>Funzione di monitoraggio del tempo Il monitoraggio inizia quando la condizione di esecuzione diventa ON. Se l'uscita B non viene attivata entro il tempo di monitoraggio, viene generato un errore non fatale.</p>  | Uscita Richiesta | 1095 |


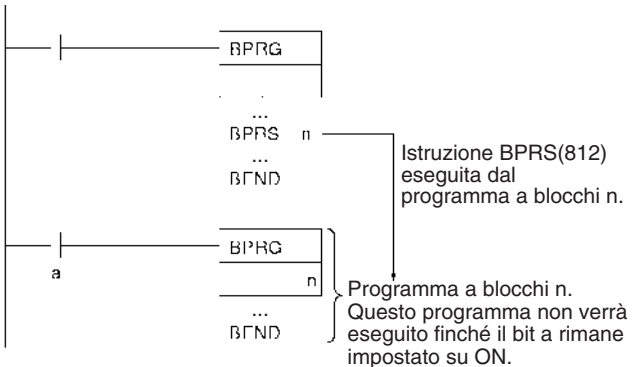
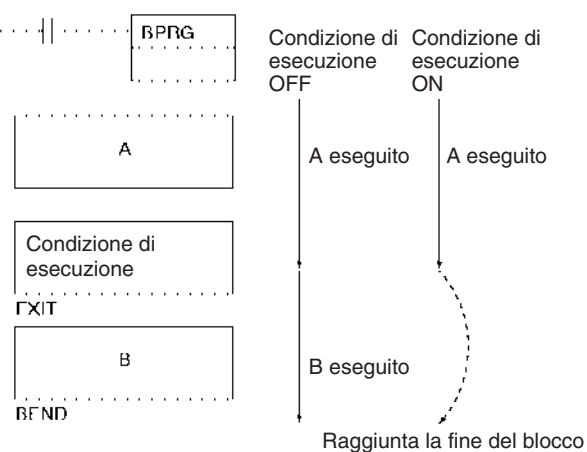
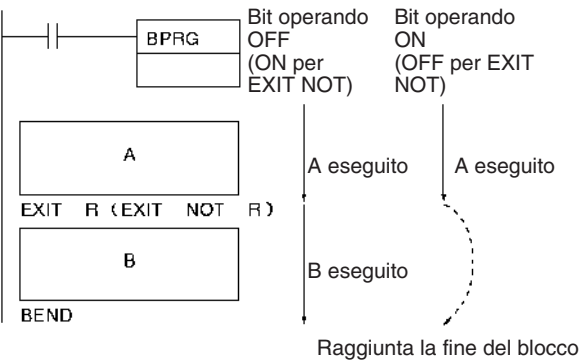
2-2-29 Altre istruzioni

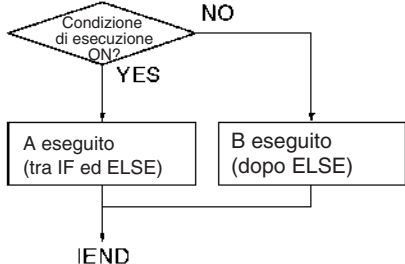
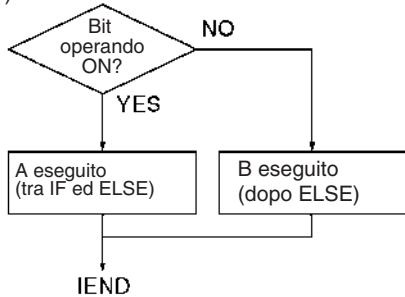
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | |
|---|---|------------|--|---|---|---------------------|------|
| SET CARRY STC @STC 040 | — <table border="1"><tr><td>STC(040)</td></tr></table> | STC(040) | Imposta il flag di riporto (CY). | Uscita Richiesta | 1104 | | |
| STC(040) | | | | | | | |
| CLEAR CARRY CLC @CLC 041 | — <table border="1"><tr><td>CLC(041)</td></tr></table> | CLC(041) | Disattiva il flag di riporto (CY). | Uscita Richiesta | 1105 | | |
| CLC(041) | | | | | | | |
| SELECT EM BANK EMBC @EMBC 281 | — <table border="1"><tr><td>EMBC(281)</td></tr><tr><td>N</td></tr></table> N: numero del banco EM | EMBC(281) | N | Modifica il banco EM corrente. | Uscita Richiesta | 1106 | |
| EMBC(281) | | | | | | | |
| N | | | | | | | |
| EXTEND MAXIMUM CYCLE TIME WDT @WDT 094 | — <table border="1"><tr><td>WDT(094)</td></tr><tr><td>T</td></tr></table> T: impostazione del temporizzatore | WDT(094) | T | Estende il tempo di ciclo massimo solo per il ciclo in cui viene eseguita l'istruzione. | Uscita Richiesta | 1108 | |
| WDT(094) | | | | | | | |
| T | | | | | | | |
| SAVE CONDI- TION FLAGS (solo CS1-H, CJ1-H, CJ1M o CS1D) CCS @CCS 282 | — <table border="1"><tr><td>CCS(282)</td></tr></table> | CCS(282) | Salva lo stato dei flag di condizione. | Uscita Richiesta | 1110 | | |
| CCS(282) | | | | | | | |
| LOAD CONDI- TION FLAGS (solo CS1-H, CJ1-H, CJ1M o CS1D) CCL @CCL 283 | — <table border="1"><tr><td>CCL(283)</td></tr></table> | CCL(283) | Legge lo stato dei flag di condizione precedentemente salvato. | Uscita Richiesta | 1112 | | |
| CCL(283) | | | | | | | |
| CONVERT ADDRESS FROM CV (solo CS1-H, CJ1-H, CJ1M o CS1D) FRMCV @FRMCV 284 | — <table border="1"><tr><td>FRMCV(284)</td></tr><tr><td>S</td></tr><tr><td>D</td></tr></table> S: canale contenente l'indirizzo di memoria della serie CV D: registro indice di destinazione | FRMCV(284) | S | D | Converte l'indirizzo di memoria di un PLC della serie CV nell'indirizzo di memoria equivalente di un PLC della serie CS/CJ. | Uscita Richiesta | 1113 |
| FRMCV(284) | | | | | | | |
| S | | | | | | | |
| D | | | | | | | |
| CONVERT ADDRESS TO CV (solo CS1-H, CJ1-H, CJ1M o CS1D) TOCV @TOCV 285 | — <table border="1"><tr><td>TOCV(285)</td></tr><tr><td>S</td></tr><tr><td>D</td></tr></table> S: registro indice contenente l'indirizzo di memoria della serie CS D: canale di destinazione | TOCV(285) | S | D | Converte l'indirizzo di memoria di un PLC della serie CS/CJ nell'indirizzo di memoria equivalente di un PLC della serie CV. | Uscita Richiesta | 1117 |
| TOCV(285) | | | | | | | |
| S | | | | | | | |
| D | | | | | | | |

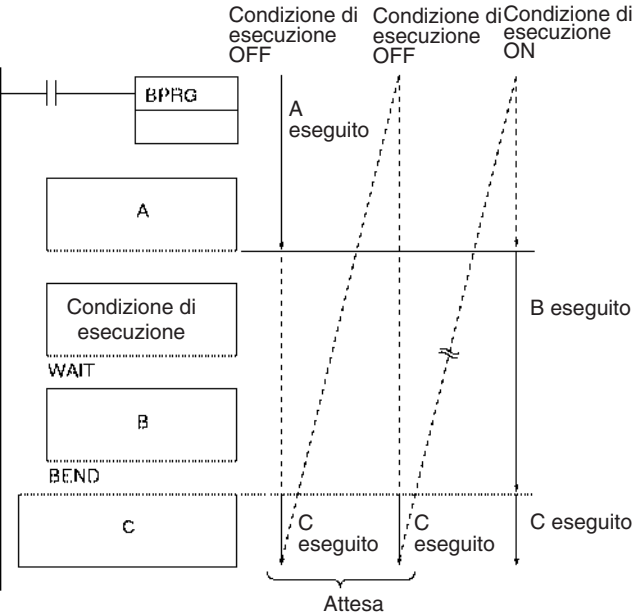
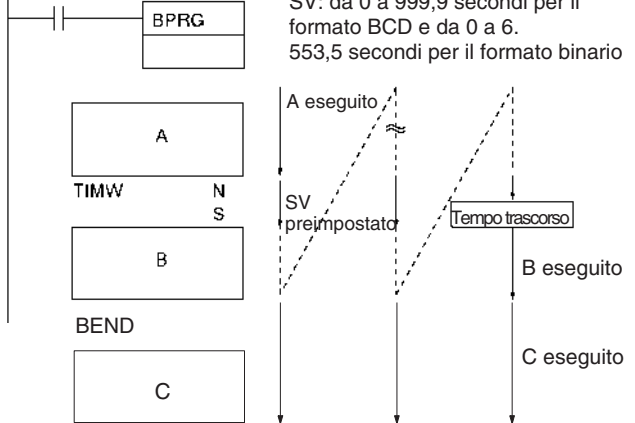
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|--|----------------------|--|--|-------------|
| DISABLE PERIPHERAL SERVICING (solo CPU CS1D per sistemi a singola CPU, CS1-H, CJ1-H o CJ1M) IOSP @IOSP 287 | | Disabilita la gestione delle periferiche durante l'esecuzione del programma in una delle modalità di elaborazione parallela o in modalità di gestione prioritaria delle periferiche. | Uscita Richiesta | 1121 |
| ENABLE PERIPHERAL SERVICING (solo CPU CS1D per sistemi a singola CPU, CS1-H, CJ1-H o CJ1M) IORS 288 | | Abilita la gestione delle periferiche disabilitata tramite l'istruzione IOS P(287) per l'esecuzione del programma in una delle modalità di elaborazione parallela o in modalità di gestione prioritaria delle periferiche. | Uscita Non richiesta | 1123 |

2-2-30 Istruzioni di programmazione a blocchi

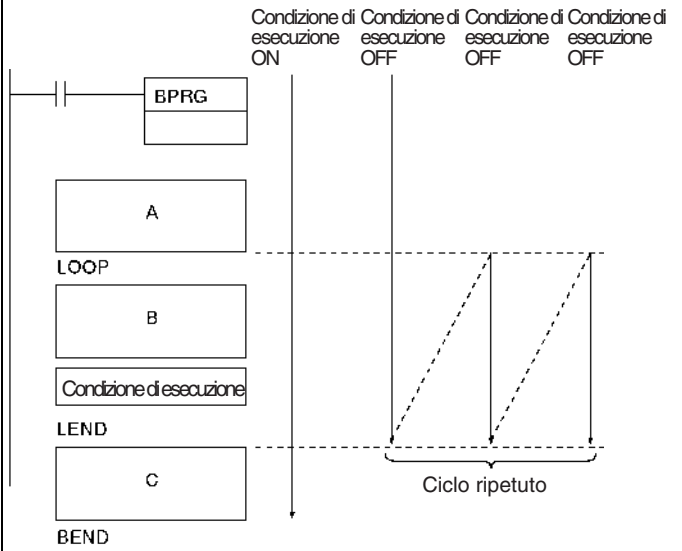
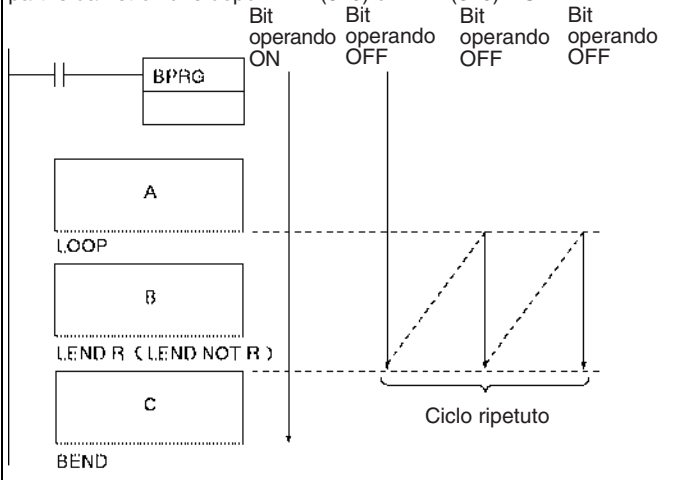
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|--|--|-------------|
| BLOCK PROGRAM BEGIN BPRG 096 | N: numero programma a blocchi | Definisce un'area di programmazione a blocchi. Per ogni istruzione BPRG(096) deve esistere un'istruzione BEND(801) corrispondente. | Uscita Richiesta | 1128 |
| BLOCK PROGRAM END BEND 801 | | Definisce un'area di programmazione a blocchi. Per ogni istruzione BPRG(096) deve esistere un'istruzione BEND(801) corrispondente. | Programma a blocchi Richiesto | 1128 |
| BLOCK PROGRAM PAUSE BPPS 811 | BPPS (811) N: numero programma a blocchi | Mette in pausa e riavvia il programma a blocchi specificato da un altro programma a blocchi. | Programma a blocchi Richiesto | 1131 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina |
|--|---|--|--|--------|
| BLOCK PROGRAM RESTART BPRS 812 | (812)  N: numero programma a blocchi | Mette in pausa e riavvia il programma a blocchi specificato da un altro programma a blocchi.  | Programma a blocchi Richiesto | 1131 |
| CONDITIONAL BLOCK EXIT EXIT 806 | EXIT(806) B: bit operando | EXIT(806) senza un bit operando esce dal programma se la condizione di esecuzione è ON.  | Programma a blocchi Richiesto | 1137 |
| CONDITIONAL BLOCK EXIT EXIT 806 | EXIT(806)B B: bit operando | EXIT(806) senza un bit operando esce dal programma se la condizione di esecuzione è ON.  | Programma a blocchi Richiesto | 1137 |
| CONDITIONAL BLOCK EXIT NOT EXIT NOT 806 | EXIT NOT(806) B B: bit operando | EXIT(806) senza un bit operando esce dal programma se la condizione di esecuzione è OFF. | Programma a blocchi Richiesto | 1137 |

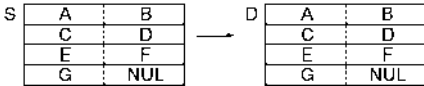
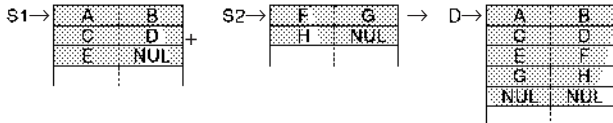
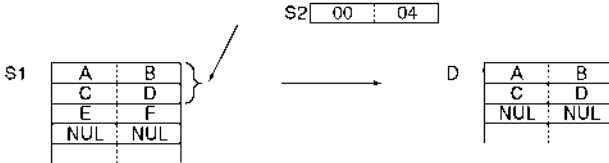
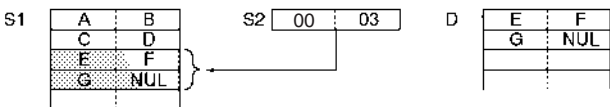
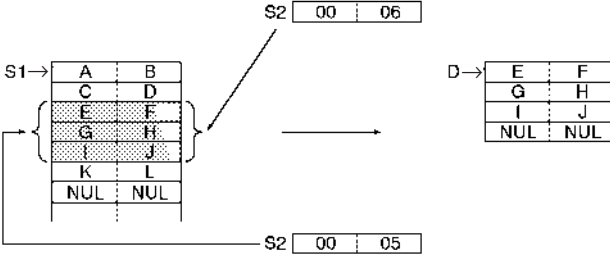
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--------------------------------------|---|--|-------------|
| CONDITIONAL BLOCK BRAN- CHING IF 802 | IF (802) | <p>Se la condizione di esecuzione è ON, vengono eseguite le istruzioni comprese tra IF(802) ed ELSE(803), mentre se la condizione di esecuzione è OFF vengono eseguite le istruzioni tra ELSE(803) e IEND(804).</p> <div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <p>Condizione di esecuzione</p> <p>IF</p> <div style="border: 1px solid black; width: 60px; height: 20px; margin: 5px auto; text-align: center;">A</div> <p>ELSE</p> <div style="border: 1px solid black; width: 60px; height: 20px; margin: 5px auto; text-align: center;">B</div> <p>IEND</p> </div> <div>  <pre> graph TD D{Condizione di esecuzione ON?} -- YES --> A[A eseguito (tra IF ed ELSE)] D -- NO --> B[B eseguito (dopo ELSE)] A --> IEND[IEND] B --> IEND </pre> </div> </div> | Programma a blocchi Richiesto | 1133 |
| CONDITIONAL BLOCK BRAN- CHING IF 802 | IF (802) B B: bit operando | <p>Se il bit operando è impostato su ON, vengono eseguite le istruzioni comprese tra IF(802) ed ELSE(803). Se il bit operando è impostato su OFF, vengono eseguite le istruzioni comprese tra ELSE(803) e IEND(804).</p> <div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <p>IF R (IF NOT R)</p> <div style="border: 1px solid black; width: 60px; height: 20px; margin: 5px auto; text-align: center;">A</div> <p>ELSE</p> <div style="border: 1px solid black; width: 60px; height: 20px; margin: 5px auto; text-align: center;">B</div> <p>IEND</p> </div> <div>  <pre> graph TD D{Bit operando ON?} -- YES --> A[A eseguito (tra IF ed ELSE)] D -- NO --> B[B eseguito (dopo ELSE)] A --> IEND[IEND] B --> IEND </pre> </div> </div> | Programma a blocchi Richiesto | 1133 |
| CONDITIONAL BLOCK BRAN- CHING (NOT) IF NOT 802 | IF (802) NOT B B: bit operando | <p>Le istruzioni comprese tra IF(802) ed ELSE(803) vengono eseguite se il bit operando è impostato su ON, mentre le istruzioni tra ELSE(803) e IEND(804) vengono eseguite se il bit operando è impostato su OFF.</p> | Programma a blocchi Richiesto | 1133 |
| CONDITIONAL BLOCK BRAN- CHING (ELSE) ELSE 803 | --- | <p>Se l'istruzione ELSE(803) viene omessa e il bit operando è impostato su ON, verranno eseguite le istruzioni comprese tra IF(802) e IEND(804).</p> | Programma a blocchi Richiesto | 1133 |
| CONDITIONAL BLOCK BRAN- CHING END IEND 804 | --- | <p>Se il bit operando è impostato su OFF, verranno eseguite solo le istruzioni inserite dopo IEND(804).</p> | Programma a blocchi Richiesto | 1133 |

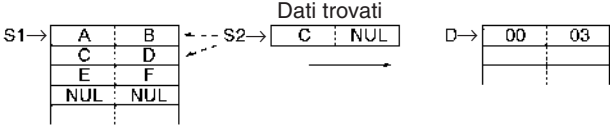
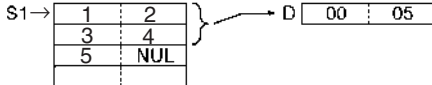
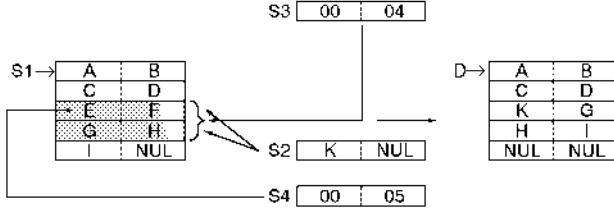
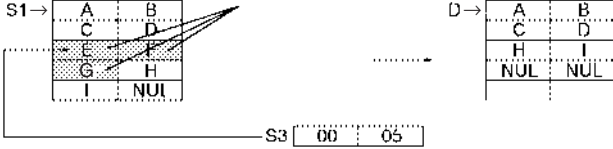
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina |
|--|---|---|--|--------|
| ONE CYCLE AND WAIT WAIT 805 | WAIT(805) | <p>Se la condizione di esecuzione è ON per WAIT(805), vengono ignorate le rimanenti istruzioni del programma a blocchi.</p>  <p>Condizione di esecuzione OFF Condizione di esecuzione OFF Condizione di esecuzione ON</p> <p>A eseguito B eseguito C eseguito</p> <p>Attesa</p> | Programma a blocchi Richiesto | 1140 |
| ONE CYCLE AND WAIT WAIT 805 B: bit operando | WAIT(805) B: bit operando | <p>Se il bit operando è impostato su OFF (ON per WAIT(805) NOT), le istruzioni rimanenti del programma a blocchi verranno ignorate. Nel ciclo successivo non verrà eseguita alcuna istruzione del programma a blocchi, ad eccezione della condizione di esecuzione per WAIT(805) o WAIT(805) NOT. Quando la condizione di esecuzione passa a ON (OFF per WAIT(805) NOT), verranno eseguite le istruzioni da WAIT(805) o WAIT(805) NOT fino alla fine del programma.</p> | Programma a blocchi Richiesto | 1140 |
| ONE CYCLE AND WAIT (NOT) WAIT NOT 805 B: bit operando | WAIT(805) NOT B: bit operando | <p>Se il bit operando è impostato su OFF (ON per WAIT(805) NOT), le istruzioni rimanenti del programma a blocchi verranno ignorate. Nel ciclo successivo non verrà eseguita alcuna istruzione del programma a blocchi, ad eccezione della condizione di esecuzione per WAIT(805) o WAIT(805) NOT. Quando la condizione di esecuzione passa a ON (OFF per WAIT(805) NOT), verranno eseguite le istruzioni da WAIT(805) o WAIT(805) NOT fino alla fine del programma.</p> | Programma a blocchi Richiesto | 1140 |
| TIMER WAIT TIMW 813 (BCD) TIMWX 816 (binario) (solo CS1-H, CJ1-H, CJ1M o CS1D) | TIMW(813) N SV N: numero del temporizzatore SV: valore impostato TIMWX(816) N SV N: numero del temporizzatore SV: valore impostato | <p>Ritarda l'esecuzione del programma a blocchi del periodo di tempo specificato. L'esecuzione riprende dall'istruzione dopo TIMW(813)/TIMWX(816) al completamento del conteggio.</p> <p>SV: da 0 a 999,9 secondi per il formato BCD e da 0 a 6.553,5 secondi per il formato binario</p>  <p>A eseguito B eseguito C eseguito</p> <p>SV preimpostato Tempo trascorso</p> | Programma a blocchi Richiesto | 1144 |

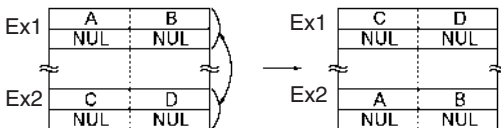
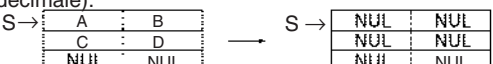
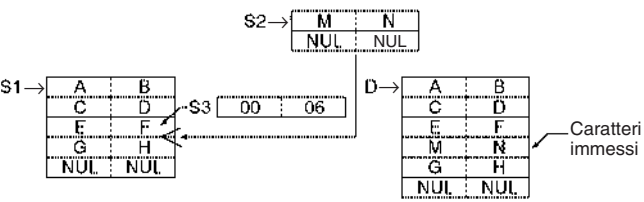
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina |
|---|---|--|--|--------|
| COUNTER WAIT CNTW 814 (BCD) CNTWX 818 (binario) (solo CS1-H, CJ1-H, CJ1M o CS1D) | CNTW(814) N SV N: numero di contatore SV: valore imposto I: ingresso conteggio CNTWX(818) N SV N: numero di contatore SV: valore imposto I: ingresso conteggio | Ritarda l'esecuzione delle rimanenti istruzioni del programma a blocchi fino al completamento del conteggio specificato. L'esecuzione riprende dall'istruzione dopo CNTW(814)/CNTWX(818) al completamento del conteggio. SV: da 0 a 9.999 volte per il formato BCD e da 0 a 65.535 volte per il formato binario | Programma a blocchi Richiesto | 1147 |
| HIGH-SPEED TIMER WAIT TMHW 815 (BCD) TMHWX 817 (binario) (solo CS1-H, CJ1-H, CJ1M o CS1D) | TMHW(815) N SV N: numero del temporizzatore SV: valore imposto TMHWX(817) N SV N: numero del temporizzatore SV: valore imposto | Ritarda l'esecuzione delle rimanenti istruzioni del programma a blocchi del periodo di tempo specificato. L'esecuzione riprende dall'istruzione dopo TMHW(815)/TMHWX(817) allo scadere del temporizzatore. SV: da 0 a 99,99 secondi per il formato BCD e da 0 a 655,35 secondi per il formato binario | Programma a blocchi Richiesto | 1150 |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina |
|--|---|---|--|--------|
| LOOP LOOP 809 | --- | <p>LOOP(809) specifica l'inizio del programma ciclico.</p>  <p>BEND</p> | Programma a blocchi Richiesto | 1153 |
| LEND LEND 810 | LEND(810) | <p>Le istruzioni LEND(810) e LEND(810) NOT specificano la fine del ciclo. Quando viene raggiunta l'istruzione LEND(810) o LEND(810) NOT, l'esecuzione del programma torna ciclicamente all'istruzione LOOP(809) immediatamente precedente finché il bit operando per LEND(810) o LEND(810) NOT non passa, rispettivamente, a ON o a OFF oppure finché lo stato della condizione di esecuzione per LEND(810) è ON.</p> | Programma a blocchi Richiesto | 1153 |
| LEND LEND 810 | LEND(810) B B: bit operando | <p>Se il bit operando è OFF per LEND(810) (o ON per LEND(810) NOT), l'esecuzione del ciclo viene ripetuta a partire dall'istruzione dopo LOOP(809). Se il bit operando è ON per LEND(810) (o OFF per LEND(810) NOT), il ciclo viene interrotto e l'esecuzione continua a partire dall'istruzione dopo LEND(810) o LEND(810) NOT.</p>  <p>Nota Lo stato del bit operando risulta invertito per LEND(810) NOT.</p> | Programma a blocchi Richiesto | 1153 |
| LEND NOT LEND NOT 810 | LEND(810) NOT B: bit operando | <p>Le istruzioni LEND(810) e LEND(810) NOT specificano la fine del ciclo. Quando viene raggiunta l'istruzione LEND(810) o LEND(810) NOT, l'esecuzione del programma torna ciclicamente all'istruzione LOOP(809) immediatamente precedente finché il bit operando per LEND(810) o LEND(810) NOT non passa, rispettivamente, a ON o a OFF oppure finché lo stato della condizione di esecuzione per LEND(810) è ON.</p> | Programma a blocchi Richiesto | 1153 |

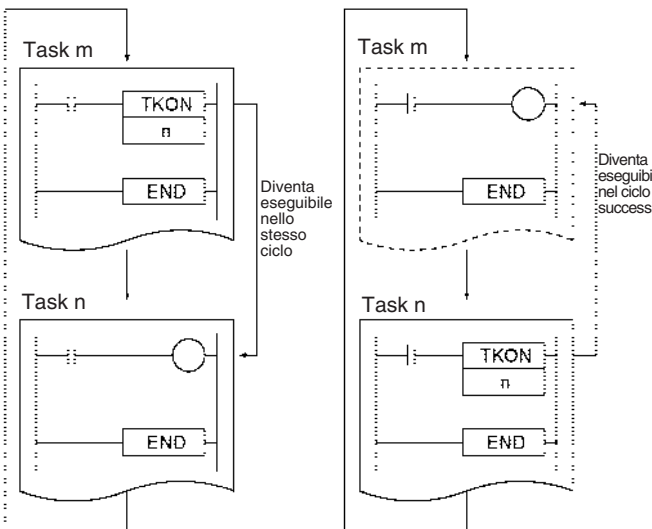
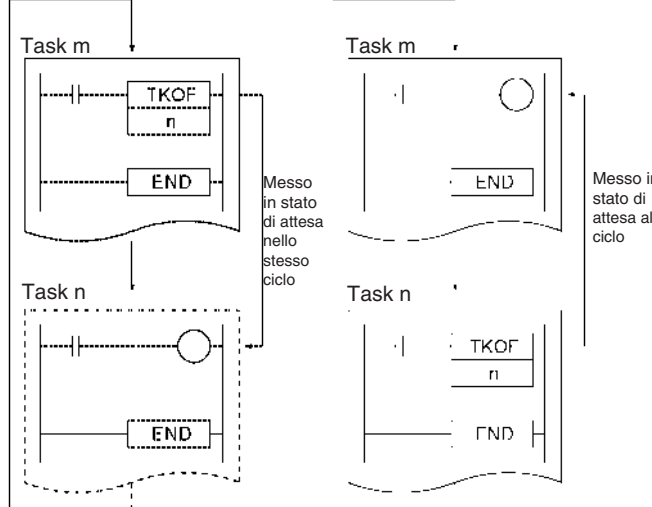
2-2-31 Istruzioni di elaborazione delle stringhe di testo

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|-----------------------------------|---|--|-------------|
| MOV STRING MOV\$ @MOV\$ 664 | MOV\$(664) S D | Trasferisce una stringa di testo.  | Uscita Richiesta | 1159 |
| CONCATENATE STRING +\$(@+\$(656 | +(656) S1 S2 D | Concatena due stringhe di testo.  | Uscita Richiesta | 1161 |
| GET STRING LEFT LEFT\$(@LEFT\$(652 | LEFT\$(652) S1 S2 D | Legge il numero di caratteri specificato a partire dall'inizio della stringa di testo.  | Uscita Richiesta | 1164 |
| GET STRING RIGHT RGHT\$(@RGHT\$(653 | RGHT\$(653) S1 S2 D | Legge il numero di caratteri specificato a partire dalla fine della stringa di testo.  | Uscita Richiesta | 1166 |
| GET STRING MIDDLE MID\$(@MID\$(654 | MID\$(654) S1 S2 S3 D | Legge il numero di caratteri specificato a partire da una determinata posizione all'interno della stringa di testo.  | Uscita Richiesta | 1168 |

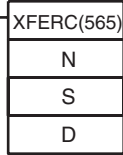
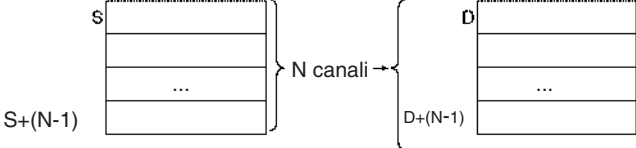
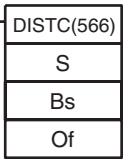
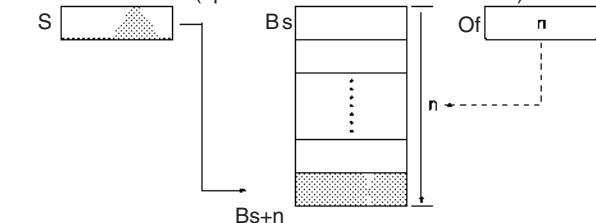
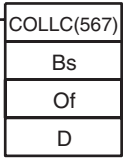
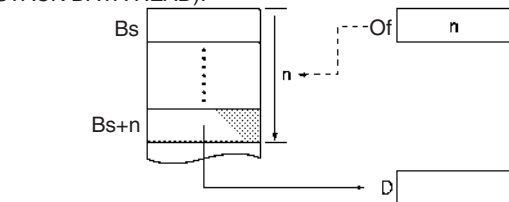
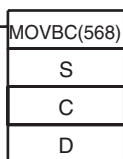
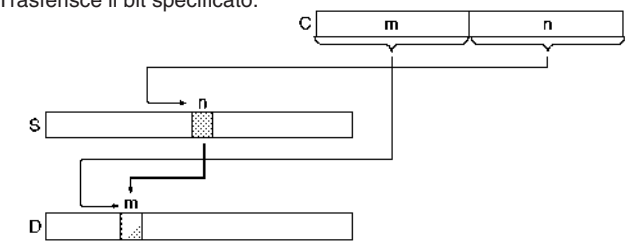
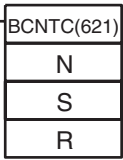
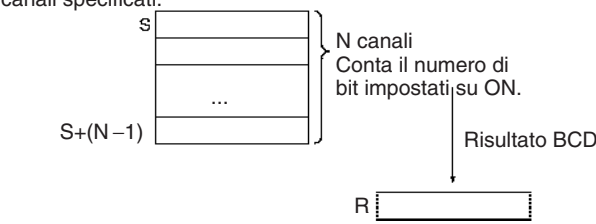
| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pagina | | | | | | |
|--|--|-------------|--|--------|---|--|---|--|---------------------|------|
| FIND IN STRING FIND @FIND\$ 660 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>FIND\$(660)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: primo canale della stringa di testo di origine S2: primo canale stringa di testo di ricerca D: primo canale di destinazione</p> | FIND\$(660) | S1 | S2 | D | Cerca la stringa di testo specificata all'interno di un'altra stringa. Dati trovati  | Uscita Richiesta | 1171 | | |
| FIND\$(660) | | | | | | | | | | |
| S1 | | | | | | | | | | |
| S2 | | | | | | | | | | |
| D | | | | | | | | | | |
| STRING LENGTH LEN\$ @LEN\$ 650 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>LEN\$(650)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: primo canale della stringa di testo D: primo canale di destinazione</p> | LEN\$(650) | S | D | Calcola la lunghezza di una stringa di testo.  | Uscita Richiesta | 1173 | | | |
| LEN\$(650) | | | | | | | | | | |
| S | | | | | | | | | | |
| D | | | | | | | | | | |
| REPLACE IN STRING RPLC\$ @RPLC\$ 661 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>RPLC\$(654)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>S4</td></tr> <tr><td>D</td></tr> </table> <p>S1: primo canale della stringa di testo S2: primo canale della stringa di testo di sostituzione S3: numero di caratteri S4: posizione iniziale D: primo canale di destinazione</p> | RPLC\$(654) | S1 | S2 | S3 | S4 | D | Sostituisce una stringa di testo con quella specificata a partire da una determinata posizione.  | Uscita Richiesta | 1175 |
| RPLC\$(654) | | | | | | | | | | |
| S1 | | | | | | | | | | |
| S2 | | | | | | | | | | |
| S3 | | | | | | | | | | |
| S4 | | | | | | | | | | |
| D | | | | | | | | | | |
| DELETE STRING DEL\$ @DEL\$ 658 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>DEL\$(658)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: primo canale della stringa di testo S2: numero di caratteri S3: posizione iniziale D: primo canale di destinazione</p> | DEL\$(658) | S1 | S2 | S3 | D | Elimina la stringa di testo specificata dall'interno di un'altra stringa. Numero di caratteri da eliminare (specificato da S2)  | Uscita Richiesta | 1178 | |
| DEL\$(658) | | | | | | | | | | |
| S1 | | | | | | | | | | |
| S2 | | | | | | | | | | |
| S3 | | | | | | | | | | |
| D | | | | | | | | | | |

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | | | | | | |
|---|---|-------------|--|--|---|---------------------|---|---------------------|------|----|--|---|------|
| EXCHANGE STRING XCHG\$ @XCHG\$ 665 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>XCHG\$(665)</td></tr> <tr><td>Ex1</td></tr> <tr><td>Ex2</td></tr> </table> <p>Ex1: primo canale di scambio 1 Ex2: primo canale di scambio 2</p> | XCHG\$(665) | Ex1 | Ex2 | Sostituisce la stringa di testo specificata con un'altra stringa specificata.  | Uscita Richiesta | 1180 | | | | | | |
| XCHG\$(665) | | | | | | | | | | | | | |
| Ex1 | | | | | | | | | | | | | |
| Ex2 | | | | | | | | | | | | | |
| CLEAR STRING CLR\$ @CLR\$ 666 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>CLR\$(666)</td></tr> <tr><td>S</td></tr> </table> <p>S: primo canale della stringa di testo</p> | CLR\$(666) | S | Sostituisce il contenuto di una stringa di testo con NUL (00 esadecimale).  | Uscita Richiesta | 1182 | | | | | | | |
| CLR\$(666) | | | | | | | | | | | | | |
| S | | | | | | | | | | | | | |
| INSERT INTO STRING INS\$ @INS\$ 657 | <table border="1" style="width: 100%; text-align: center;"> <tr><td>INS\$(657)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: primo canale della stringa di testo di base S2: primo canale della stringa di testo inserito S3: posizione iniziale D: primo canale di destinazione</p> | INS\$(657) | S1 | S2 | S3 | D | Elimina la stringa di testo specificata dall'interno di un'altra stringa.  | Uscita Richiesta | 1184 | | | | |
| INS\$(657) | | | | | | | | | | | | | |
| S1 | | | | | | | | | | | | | |
| S2 | | | | | | | | | | | | | |
| S3 | | | | | | | | | | | | | |
| D | | | | | | | | | | | | | |
| Confronto di stringhe LD, AND, OR + =\$, <>\$, <\$, <=\$, >\$, >=\$ 670 (=\$) 671 (<>\$) 672 (<\$) 673 (<=\$) 674 (>\$) 675 (>=\$) | <p>LD</p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>Simbolo</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> </table> <p>AND</p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>Simbolo</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> </table> <p>OR</p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>Simbolo</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> </table> <p>S1: stringa di testo 1 S2: stringa di testo 2</p> | Simbolo | S1 | S2 | Simbolo | S1 | S2 | Simbolo | S1 | S2 | Le istruzioni di confronto delle stringhe (=\$, <>\$, <\$, <=\$, >\$, >=\$) confrontano due stringhe di testo a partire dall'inizio, verificando i valori dei codici ASCII. Se il risultato del confronto è vero, viene creata una condizione di esecuzione ON per LOAD, AND o OR. | LD: non richiesta AND, OR: richiesta | 1187 |
| Simbolo | | | | | | | | | | | | | |
| S1 | | | | | | | | | | | | | |
| S2 | | | | | | | | | | | | | |
| Simbolo | | | | | | | | | | | | | |
| S1 | | | | | | | | | | | | | |
| S2 | | | | | | | | | | | | | |
| Simbolo | | | | | | | | | | | | | |
| S1 | | | | | | | | | | | | | |
| S2 | | | | | | | | | | | | | |

2-2-32 Istruzioni di controllo dei task

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|--|---|--|-------------|
| <p>TASK ON TKON @TKON 820</p> | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">TKON(820)</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">N</div> <p>N: numero di task</p> | <p>Rende il task specificato eseguibile.</p> <p>Il numero di task del task specificato è superiore a quello del task locale ($m < n$).</p> <p>Il numero di task del task specificato è inferiore a quello del task locale ($m > n$).</p>  | <p>Uscita Richiesta</p> | <p>1192</p> |
| <p>TASK OFF TKOF @TKOF 821</p> | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">TKOF(821)</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">N</div> <p>N: numero di task</p> | <p>Mette in stato di attesa il task specificato.</p> <p>Il numero di task del task specificato è superiore a quello del task locale ($m < n$).</p> <p>Il numero di task del task specificato è inferiore a quello del task locale ($m > n$).</p>  | <p>Uscita Richiesta</p> | <p>1196</p> |

2-2-33 Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva)

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina |
|---|---|---|--|-------------|
| BLOCK TRANSFER XFERC @XFERC 565 |  <p>N: numero di canali S: primo canale di origine D: primo canale di destinazione</p> | <p>Trasferisce il numero di canali consecutivi specificato.</p>  | Uscita Richiesta | 1201 |
| SINGLE WORD DISTRIBUTE DISTC @DISTC 566 |  <p>S: canale di origine Bs: indirizzo base di destinazione Of: offset</p> | <p>Trasferisce il canale di origine in un canale di destinazione calcolato aggiungendo il valore di offset all'indirizzo di base. È anche possibile scrivere in uno stack (operazione PUSH ONTO STACK).</p>  | Uscita Richiesta | 1203 |
| DATA COLLECT COLLC @COLLC 567 |  <p>Bs: indirizzo base di origine Of: offset D: canale di destinazione</p> | <p>Trasferisce il canale di origine, calcolato aggiungendo un valore di offset all'indirizzo di base, nel canale di destinazione. È anche possibile leggere da uno stack in ordine FIFO o LIFO (operazione STACK DATA READ).</p>  | Uscita Richiesta | 1206 |
| MOVE BIT MOVBC @MOVBC 568 |  <p>S: dati o canale di origine C: canale di controllo D: canale di destinazione</p> | <p>Trasferisce il bit specificato.</p>  | Uscita Richiesta | 1211 |
| BIT COUNTER BCNTC @BCNTC 621 |  <p>N: numero di canali (BCD) S: primo canale di origine R: canale del risultato</p> | <p>Conta il numero totale di bit impostati su ON presenti nei canali specificati.</p>  | Uscita Richiesta | 1212 |

2-2-34 Istruzioni speciali per blocchi funzione

| Istruzione Codice mnemonico | Simbolo/ operando | Funzione | Posizione Condizione di esecuzione | Pa- gina | | | | |
|--|---|------------|--|-------------|----|--|---------------------|------|
| GET VARIABLE ID GETID @GETID 286 | <table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td style="text-align: center;">GETID(286)</td></tr> <tr><td style="text-align: center;">S</td></tr> <tr><td style="text-align: center;">D1</td></tr> <tr><td style="text-align: center;">D2</td></tr> </table> <p>S: variabile o indirizzo D1: codice ID D2: canale di destinazione</p> | GETID(286) | S | D1 | D2 | <p>Invia il codice del tipo di variabile del comando FINS (area dati) e l'indirizzo di canale per la variabile o l'indirizzo specificato. Questa istruzione viene in genere utilizzata per ottenere l'indirizzo assegnato a una variabile in un blocco funzione.</p> | Uscita Richiesta | 1214 |
| GETID(286) | | | | | | | | |
| S | | | | | | | | |
| D1 | | | | | | | | |
| D2 | | | | | | | | |

2-3 Elenco alfabetico delle istruzioni in base al codice mnemonico

A

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|----------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| ACC | ACCELERATION CONTROL | 888 | @ACC | --- | --- | 855 |
| ACOS | ARC COSINE | 464 | @ACOS | --- | --- | 589 |
| ACOSD | DOUBLE ARC COSINE | 855 | @ACOSD | --- | --- | 645 |
| AND | AND | --- | @AND | %AND | !AND | 157 |
| AND < | AND LESS THAN | 310 | --- | --- | --- | 275 |
| AND <\$ | AND STRING LESS THAN | 672 | --- | --- | --- | 1187 |
| AND <> | AND NOT EQUAL | 305 | --- | --- | --- | 275 |
| AND <>\$ | AND STRING NOT EQUAL | 671 | --- | --- | --- | 1187 |
| AND <>D | AND DOUBLE FLOATING NOT EQUAL | 336 | --- | --- | --- | 657 |
| AND <>DT | AND TIME NOT EQUAL | 342 | --- | --- | --- | 281 |
| AND <>F | AND FLOATING NOT EQUAL | 330 | --- | --- | --- | 600 |
| AND <>L | AND DOUBLE NOT EQUAL | 306 | --- | --- | --- | 275 |
| AND <>S | AND SIGNED NOT EQUAL | 307 | --- | --- | --- | 275 |
| AND <>SL | AND DOUBLE SIGNED NOT EQUAL | 308 | --- | --- | --- | 275 |
| AND <D | AND DOUBLE FLOATING LESS THAN | 337 | --- | --- | --- | 657 |
| AND <DT | AND TIME LESS THAN | 343 | --- | --- | --- | 281 |
| AND <F | AND FLOATING LESS THAN | 331 | --- | --- | --- | 600 |
| AND <L | AND DOUBLE LESS THAN | 311 | --- | --- | --- | 275 |
| AND <S | AND SIGNED LESS THAN | 312 | --- | --- | --- | 275 |
| AND <SL | AND DOUBLE SIGNED LESS THAN | 313 | --- | --- | --- | 275 |
| AND = | AND EQUAL | 300 | --- | --- | --- | 275 |
| AND =\$ | AND STRING EQUALS | 670 | --- | --- | --- | 1187 |
| AND =D | AND DOUBLE FLOATING EQUAL | 335 | --- | --- | --- | 657 |
| AND =DT | AND TIME EQUAL | 341 | --- | --- | --- | 281 |
| AND =F | AND FLOATING EQUAL | 329 | --- | --- | --- | 600 |
| AND =L | AND DOUBLE EQUAL | 301 | --- | --- | --- | 275 |
| AND =S | AND SIGNED EQUAL | 302 | --- | --- | --- | 275 |
| AND =SL | AND DOUBLE SIGNED EQUAL | 303 | --- | --- | --- | 275 |
| AND > | AND GREATER THAN | 320 | --- | --- | --- | 275 |
| AND >\$ | AND STRING GREATER THAN | 674 | --- | --- | --- | 1187 |
| AND >D | AND DOUBLE FLOATING GREATER THAN | 339 | --- | --- | --- | 657 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---|-----------------|---------------------------------------|--|-------------------------|--------|
| AND >DT | AND TIME GREATER THAN | 345 | --- | --- | --- | 281 |
| AND >F | AND FLOATING GREATER THAN | 333 | --- | --- | --- | 600 |
| AND >L | AND DOUBLE GREATER THAN | 321 | --- | --- | --- | 275 |
| AND >S | AND SIGNED GREATER THAN | 322 | --- | --- | --- | 275 |
| AND >SL | AND DOUBLE SIGNED GREATER THAN | 323 | --- | --- | --- | 275 |
| AND LD | AND LOAD | --- | --- | --- | --- | 164 |
| AND NOT | AND NOT | --- | --- | --- | !AND NOT | 159 |
| AND TST | AND BIT TEST | 350 | --- | --- | --- | 174 |
| AND TSTN | AND BIT TEST | 351 | --- | --- | --- | 174 |
| AND <= | AND LESS THAN OR EQUAL | 315 | --- | --- | --- | 275 |
| AND <=\$ | AND STRING LESS THAN OR EQUAL | 673 | --- | --- | --- | 1187 |
| AND <=D | AND DOUBLE FLOATING LESS THAN OR EQUAL | 338 | --- | --- | --- | 657 |
| AND <=DT | AND TIME LESS THAN OR EQUAL | 344 | --- | --- | --- | 281 |
| AND <=F | AND FLOATING LESS THAN OR EQUAL | 332 | --- | --- | --- | 600 |
| AND <=L | AND DOUBLE LESS THAN OR EQUAL | 316 | --- | --- | --- | 275 |
| AND <=S | AND SIGNED LESS THAN OR EQUAL | 317 | --- | --- | --- | 275 |
| AND <=SL | AND DOUBLE SIGNED LESS THAN OR EQUAL | 318 | --- | --- | --- | 275 |
| AND >= | AND GREATER THAN OR EQUAL | 325 | --- | --- | --- | 275 |
| AND >=\$ | AND STRING GREATER THAN OR EQUALS | 675 | --- | --- | --- | 1187 |
| AND >=D | AND DOUBLE FLOATING GREATER THAN OR EQUAL | 340 | --- | --- | --- | 657 |
| AND >=DT | AND TIME GREATER THAN OR EQUAL | 346 | --- | --- | --- | 281 |
| AND >=F | AND FLOATING GREATER THAN OR EQUAL | 334 | --- | --- | --- | 600 |
| AND >=L | AND DOUBLE GREATER THAN OR EQUAL | 326 | --- | --- | --- | 275 |
| AND >=S | AND SIGNED GREATER THAN OR EQUAL | 327 | --- | --- | --- | 275 |
| AND >=SL | AND DOUBLE SIGNED GREATER THAN OR EQUAL | 328 | --- | --- | --- | 275 |
| ANDL | DOUBLE LOGICAL AND | 610 | @ANDL | --- | --- | 519 |
| ANDW | LOGICAL AND | 034 | @ANDW | --- | --- | 517 |
| APR | ARITHMETIC PROCESS | 069 | @APR | --- | --- | 540 |
| ASC | ASCII CONVERT | 086 | @ASC | --- | --- | 486 |
| ASFT | ASYNCHRONOUS SHIFT REGISTER | 017 | @ASFT | --- | --- | 349 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| ASIN | ARC SINE | 463 | @ASIN | --- | --- | 587 |
| ASIND | DOUBLE ARC SINE | 854 | @ASIND | --- | --- | 643 |
| ASL | ARITHMETIC SHIFT LEFT | 025 | @ASL | --- | --- | 354 |
| ASLL | DOUBLE SHIFT LEFT | 570 | @ASLL | --- | --- | 355 |
| ASR | ARITHMETIC SHIFT RIGHT | 026 | @ASR | --- | --- | 357 |
| ASRL | DOUBLE SHIFT RIGHT | 571 | @ASRL | --- | --- | 358 |
| ATAN | ARC TANGENT | 465 | @ATAN | --- | --- | 591 |
| ATAND | DOUBLE ARC TANGENT | 856 | @ATAND | --- | --- | 647 |
| AVG | AVERAGE | 195 | --- | --- | --- | 769 |

B

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|-----------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| BAND | DEAD BAND CONTROL | 681 | @BAND | --- | --- | 743 |
| BCD | BINARY-TO-BCD | 024 | @BCD | --- | --- | 469 |
| BCDL | DOUBLE BINARY-TO-BCD | 059 | @BCDL | --- | --- | 470 |
| BCDS | SIGNED BINARY-TO-BCD | 471 | @BCDS | --- | --- | 505 |
| BCMP | UNSIGNED BLOCK COMPARE | 068 | @BCMP | --- | --- | 304 |
| BCMP2 | EXPANDED BLOCK COMPARE | 502 | @BCMP2 | --- | --- | 306 |
| BCNT | BIT COUNTER | 067 | @BCNT | --- | --- | 556 |
| BCNTC | BIT COUNTER | 621 | @BCNTC | --- | --- | 1212 |
| BDSL | DOUBLE SIGNED BINARY-TO-BCD | 473 | @BDSL | --- | --- | 507 |
| BEND | BLOCK PROGRAM END | 801 | --- | --- | --- | 1128 |
| BIN | BCD-TO-BINARY | 023 | @BIN | --- | --- | 466 |
| BINL | DOUBLE BCD-TO-DOUBLE BINARY | 058 | @BINL | --- | --- | 467 |
| BINS | SIGNED BCD-TO-BINARY | 470 | @BINS | --- | --- | 499 |
| BISL | DOUBLE SIGNED BCD-TO-BINARY | 472 | @BISL | --- | --- | 502 |
| BPPS | BLOCK PROGRAM PAUSE | 811 | --- | --- | --- | 1131 |
| BPRG | BLOCK PROGRAM BEGIN | 096 | --- | --- | --- | 1128 |
| BPRS | BLOCK PROGRAM RESTART | 812 | --- | --- | --- | 1131 |
| BREAK | BREAK LOOP | 514 | --- | --- | --- | 232 |
| BSET | BLOCK SET | 071 | @BSET | --- | --- | 331 |

C

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|----------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| CADD | CALENDAR ADD | 730 | @CADD | --- | --- | 1061 |
| CCL | LOAD CONDITION FLAGS | 283 | @CCL | --- | --- | 1112 |
| CCS | SAVE CONDITION FLAGS | 282 | @CCS | --- | --- | 1110 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| CJP | CONDITIONAL JUMP | 510 | --- | --- | --- | 223 |
| CJPN | CONDITIONAL JUMP | 511 | --- | --- | --- | 223 |
| CLC | CLEAR CARRY | 041 | @CLC | --- | --- | 1105 |
| CLI | CLEAR INTERRUPT | 691 | @CLI | --- | --- | 810 |
| CLR\$ | CLEAR STRING | 666 | @CLR\$ | --- | --- | 1182 |
| CMND | DELIVER COMMAND | 490 | @CMND | --- | --- | 1003 |
| CMP | COMPARE | 020 | --- | --- | !CMP | 287 |
| CMPL | DOUBLE COMPARE | 060 | --- | --- | --- | 290 |
| CNR | RESET TIMER/ COUNTER | 545 | @CNR | --- | --- | 267 |
| CNRX | RESET TIMER/ COUNTER | 548 | @CNRX | --- | --- | 267 |
| CNT | COUNTER | --- | --- | --- | --- | 260 |
| CNTX | COUNTER | 546 | --- | --- | --- | 260 |
| CNTR | REVERSIBLE COUNTER | 012 | --- | --- | --- | 263 |
| CNTRX | REVERSIBLE COUNTER | 548 | --- | --- | --- | 263 |
| CNTW | COUNTER WAIT | 814 | --- | --- | --- | 1147 |
| CNTWX | COUNTER WAIT | 818 | --- | --- | --- | 1147 |
| COLL | DATA COLLECT | 081 | @COLL | --- | --- | 338 |
| COLLC | DATA COLLECT | 567 | @COLLC | --- | --- | 1206 |
| COLM | LINE TO COLUMN | 064 | @COLM | --- | --- | 496 |
| COM | COMPLEMENT | 029 | --- | --- | --- | 531 |
| COML | DOUBLE COMPLEMENT | 614 | @COML | --- | --- | 533 |
| COS | COSINE | 461 | @COS | --- | --- | 583 |
| COSD | DOUBLE COSINE | 852 | @COSD | --- | --- | 639 |
| CPS | SIGNED BINARY COMPARE | 114 | --- | --- | !CPS | 293 |
| CPSL | DOUBLE SIGNED BINARY COMPARE | 115 | --- | --- | --- | 296 |
| CSUB | CALENDAR SUBTRACT | 731 | @CSUB | --- | --- | 1065 |
| CTBL | COMPARISON TABLE LOAD | 882 | @CTBL | --- | --- | 837 |

D

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|-------------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| DATE | CLOCK ADJUSTMENT | 735 | @DATE | --- | --- | 1073 |
| DBL | 16-BIT BINARY TO DOUBLE FLOATING | 843 | @DBL | --- | --- | 623 |
| DBLL | 32-BIT BINARY TO DOUBLE FLOATING | 844 | @DBLL | --- | --- | 624 |
| DEG | RADIANS TO DEGREES | 459 | @DEG | --- | --- | 579 |
| DEGD | DOUBLE RADIANS TO DEGREES | 850 | @RADD | --- | --- | 634 |
| DEL\$ | DELETE STRING | 658 | @DEL\$ | --- | --- | 1178 |
| DI | DISABLE INTERRUPTS | 693 | @DI | --- | --- | 814 |
| DIFD | DIFFERENTIATE DOWN | 014 | --- | --- | !DIFD | 184 |
| DIFU | DIFFERENTIATE UP | 013 | --- | --- | !DIFU | 184 |
| DIM | DIMENSION RECORD TABLE | 631 | @DIM | --- | --- | 678 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| DIST | SINGLE WORD DISTRIBUTE | 080 | @DIST | --- | --- | 336 |
| DISTC | SINGLE WORD DISTRIBUTE | 566 | @DISTC | --- | --- | 1203 |
| DLNK | CPU BUS UNIT I/O REFRESH | 226 | @DLNK | --- | --- | 921 |
| DMPX | DATA ENCODER | 077 | @DMPX | --- | --- | 482 |
| DOWN | CONDITION OFF | 522 | --- | --- | --- | 173 |
| DSW | DIGITAL SWITCH INPUT | 210 | --- | --- | --- | 890 |

E

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-------------------------------|----------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| ECHRD | EXPLICIT WORD READ | 723 | @ECHRD | --- | --- | 1034 |
| ECHWR | EXPLICIT WORD WRITE | 724 | @ECHWR | --- | --- | 1038 |
| EGATR | EXPLICIT GET ATTRIBUTE | 721 | @EGATR | --- | --- | 1021 |
| EI | ENABLE INTERRUPTS | 694 | --- | --- | --- | 816 |
| ELSE | ELSE | 803 | --- | --- | --- | 1133 |
| EMBC | SELECT EM BANK | 281 | @EMBC | --- | --- | 1106 |
| END | END | 001 | --- | --- | --- | 197 |
| ESATR | EXPLICIT SET ATTRIBUTE | 722 | @ESATR | --- | --- | 1028 |
| EXIT NOT (operando) | CONDITIONAL BLOCK EXIT NOT | 806 | --- | --- | --- | 1137 |
| EXIT (condizione di ingresso) | CONDITIONAL BLOCK EXIT | 806 | --- | --- | --- | 1137 |
| EXIT (operando) | CONDITIONAL BLOCK EXIT | 806 | --- | --- | --- | 1137 |
| EXP | EXPONENT | 467 | @EXP | --- | --- | 595 |
| EXPD | DOUBLE EXPONENT | 858 | @EXPD | --- | --- | 651 |
| EXPLT | EXPLICIT MESSAGE SEND | 720 | @EXPLT | --- | --- | 1013 |

F

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|----------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| FAL | FAILURE ALARM | 006 | @FAL | --- | --- | 1079 |
| FALS | SEVERE FAILURE ALARM | 007 | --- | --- | --- | 1087 |
| FCS | FRAME CHECKSUM | 180 | @FCS | --- | --- | 700 |
| FDIV | FLOATING POINT DIVIDE | 079 | @FDIV | --- | --- | 552 |
| FIFO | FIRST IN FIRST OUT | 633 | @FIFO | --- | --- | 672 |
| FIND\$ | FIND IN STRING | 660 | @FIND\$ | --- | --- | 1171 |
| FIX | FLOATING TO 16-BIT | 450 | @FIX | --- | --- | 563 |
| FIXD | DOUBLE FLOATING TO 16-BIT BINARY | 841 | @FIXD | --- | --- | 620 |
| FIXL | FLOATING TO 32-BIT | 451 | @FIXL | --- | --- | 565 |
| FIXLD | DOUBLE FLOATING TO 32-BIT BINARY | 842 | @FIXLD | --- | --- | 621 |
| FLT | 16-BIT TO FLOATING | 452 | @FLT | --- | --- | 566 |
| FLTL | 32-BIT TO FLOATING | 453 | @FLTL | --- | --- | 568 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|-------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| FOR | FOR-NEXT LOOPS | 512 | --- | --- | --- | 229 |
| FPD | FAILURE POINT DETECTION | 269 | --- | --- | --- | 1095 |
| FREAD | READ DATA FILE | 700 | @FREAD | --- | --- | 1045 |
| FRMCV | CONVERT ADDRESS FROM CV | 284 | @FRMCV | --- | --- | 1113 |
| FSTR | FLOATING POINT TO ASCII | 448 | @FSTR | --- | --- | 604 |
| FWRIT | WRITE DATA FILE | 701 | @FWRIT | --- | --- | 1052 |
| FVAL | ASCII TO FLOATING POINT | 449 | @FVAL | --- | --- | 609 |

G

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| GETID | GET VARIABLE ID | 286 | @GETID | --- | --- | 1214 |
| GETR | GET RECORD NUMBER | 636 | @GETR | --- | --- | 683 |
| GRET | GLOBAL SUBROUTINE RETURN | 752 | --- | --- | --- | 797 |
| GRY | GRAY CODE CONVERSION | 474 | @GRY | --- | --- | 511 |
| GSBN | GLOBAL SUBROUTINE ENTRY | 751 | --- | --- | --- | 794 |
| GSBS | GLOBAL SUBROUTINE CALL | 750 | @GSBS | --- | --- | 786 |

H

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|-----------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| HEX | ASCII TO HEX | 162 | @HEX | --- | --- | 490 |
| HKY | HEXADECIMAL KEY INPUT | 212 | --- | --- | --- | 899 |
| HMS | SECONDS TO HOURS | 066 | @HMS | --- | --- | 1070 |

I

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------------------|-----------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| IEND | IF END | 804 | --- | --- | --- | 1133 |
| IF NOT (operando) | IF NOT | 802 | --- | --- | --- | 1133 |
| IF (condizione di ingresso) | IF | 802 | --- | --- | --- | 1133 |
| IF (operando) | IF | 802 | --- | --- | --- | 1133 |
| IL | INTERLOCK | 002 | --- | --- | --- | 201 |
| ILC | INTERLOCK CLEAR | 003 | --- | --- | --- | 201 |
| INI | MODE CONTROL | 880 | @INI | --- | --- | 823 |
| INS\$ | INS\$ | 657 | @INS\$ | --- | --- | 1184 |
| IORD | INTELLIGENT I/O READ | 222 | @IORD | --- | --- | 913 |
| IORF | I/O REFRESH | 097 | @IORF | --- | --- | 885 |
| IORS | ENABLE PERIPHERAL SERVICING | 288 | --- | --- | --- | 1123 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| IOSP | DISABLE PERIPHERAL SERVICING | 287 | @IOSP | --- | --- | 1121 |
| IOWR | INTELLIGENT I/O WRITE | 223 | @IOWR | --- | --- | 917 |

J

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|-------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| JME | JUMP END | 005 | --- | --- | --- | 219 |
| JME0 | MULTIPLE JUMP END | 516 | --- | --- | --- | 227 |
| JMP | JUMP | 004 | --- | --- | --- | 219 |
| JMP0 | MULTIPLE JUMP | 515 | --- | --- | --- | 227 |

K

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|------------|-----------------|---------------------------------------|--|-------------------------|--------|
| KEEP | KEEP | 011 | --- | --- | !KEEP | 180 |

L

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| LD | LOAD | --- | @LD | %LD | !LD | 153 |
| LD < | LOAD LESS THAN | 310 | --- | --- | --- | 275 |
| LD <\$ | LOAD STRING LESS THAN | 672 | --- | --- | --- | 1187 |
| LD <D | LOAD DOUBLE FLOATING LESS THAN | 337 | --- | --- | --- | 657 |
| LD <DT | LOAD TIME LESS THAN | 343 | --- | --- | --- | 281 |
| LD <F | LOAD FLOATING LESS THAN | 331 | --- | --- | --- | 600 |
| LD <> | LOAD NOT EQUAL | 305 | --- | --- | --- | 275 |
| LD <>\$ | LOAD STRING NOT EQUAL | 671 | --- | --- | --- | 1187 |
| LD <>D | LOAD DOUBLE FLOATING NOT EQUAL | 336 | --- | --- | --- | 657 |
| LD <>DT | LOAD TIME NOT EQUAL | 342 | --- | --- | --- | 281 |
| LD <>F | LOAD FLOATING NOT EQUAL | 330 | --- | --- | --- | 600 |
| LD <>L | LOAD DOUBLE NOT EQUAL | 306 | --- | --- | --- | 275 |
| LD <>S | LOAD SIGNED NOT EQUAL | 307 | --- | --- | --- | 275 |
| LD <>SL | LOAD DOUBLE SIGNED NOT EQUAL | 308 | --- | --- | --- | 275 |
| LD <L | LOAD DOUBLE LESS THAN | 311 | --- | --- | --- | 275 |
| LD <S | LOAD SIGNED LESS THAN | 312 | --- | --- | --- | 275 |
| LD <SL | LOAD DOUBLE SIGNED LESS THAN | 313 | --- | --- | --- | 275 |
| LD = | LOAD EQUAL | 300 | --- | --- | --- | 275 |
| LD =\$ | LOAD STRING EQUALS | 670 | --- | --- | --- | 1187 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--|-----------------|---------------------------------------|--|-------------------------|--------|
| LD =D | LOAD DOUBLE FLOATING EQUAL | 335 | --- | --- | --- | 657 |
| LD =DT | LOAD TIME EQUAL | 341 | --- | --- | --- | 281 |
| LD =F | LOAD FLOATING EQUAL | 329 | --- | --- | --- | 600 |
| LD =L | LOAD DOUBLE EQUAL | 301 | --- | --- | --- | 275 |
| LD =S | LOAD SIGNED EQUAL | 302 | --- | --- | --- | 275 |
| LD =SL | LOAD DOUBLE SIGNED EQUAL | 303 | --- | --- | --- | 275 |
| LD > | LOAD GREATER THAN | 320 | --- | --- | --- | 275 |
| LD >\$ | LOAD STRING GREATER THAN | 674 | --- | --- | --- | 1187 |
| LD >D | LOAD DOUBLE FLOATING GREATER THAN | 339 | --- | --- | --- | 657 |
| LD >DT | LOAD TIME GREATER THAN | 345 | --- | --- | --- | 281 |
| LD >F | LOAD FLOATING GREATER THAN | 333 | --- | --- | --- | 600 |
| LD >L | LOAD DOUBLE GREATER THAN | 321 | --- | --- | --- | 275 |
| LD >S | LOAD SIGNED GREATER THAN | 322 | --- | --- | --- | 275 |
| LD >SL | LOAD DOUBLE SIGNED GREATER THAN | 323 | --- | --- | --- | 275 |
| LD NOT | LOAD NOT | --- | --- | --- | !LD NOT | 155 |
| LD TST | LOAD BIT TEST | 350 | --- | --- | --- | 174 |
| LD TSTN | LOAD BIT TEST | 351 | --- | --- | --- | 174 |
| LD <= | LOAD LESS THAN OR EQUAL | 315 | --- | --- | --- | 275 |
| LD <=\$ | LOAD STRING LESS THAN OR EQUAL | 673 | --- | --- | --- | 1187 |
| LD <=D | LOAD DOUBLE FLOATING LESS THAN OR EQUAL | 338 | --- | --- | --- | 657 |
| LD <=DT | LOAD TIME LESS THAN OR EQUAL | 344 | --- | --- | --- | 281 |
| LD <=F | LOAD FLOATING LESS THAN OR EQUAL | 332 | --- | --- | --- | 600 |
| LD <=L | LOAD DOUBLE LESS THAN OR EQUAL | 316 | --- | --- | --- | 275 |
| LD <=S | LOAD SIGNED LESS THAN OR EQUAL | 317 | --- | --- | --- | 275 |
| LD <=SL | LOAD DOUBLE SIGNED LESS THAN OR EQUAL | 318 | --- | --- | --- | 275 |
| LD >= | LOAD GREATER THAN OR EQUAL | 325 | --- | --- | --- | 275 |
| LD >=\$ | LOAD STRING GREATER THAN OR EQUALS | 675 | --- | --- | --- | 1187 |
| LD >=D | LOAD DOUBLE FLOATING GREATER THAN OR EQUAL | 340 | --- | --- | --- | 657 |
| LD >=DT | LOAD TIME GREATER THAN OR EQUAL | 346 | --- | --- | --- | 281 |
| LD >=F | LOAD FLOATING GREATER THAN OR EQUAL | 334 | --- | --- | --- | 600 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-------------------------------|--|-----------------|---------------------------------------|--|-------------------------|--------|
| LD >=L | LOAD DOUBLE GREATER THAN OR EQUAL | 326 | --- | --- | --- | 275 |
| LD >=S | LOAD SIGNED GREATER THAN OR EQUAL | 327 | --- | --- | --- | 275 |
| LD >=SL | LOAD DOUBLE SIGNED GREATER THAN OR EQUAL | 328 | --- | --- | --- | 275 |
| LEFT\$ | GET STRING LEFT | 652 | @LEFT\$ | --- | --- | 1164 |
| LEN\$ | STRING LENGTH | 650 | @LEN\$ | --- | --- | 1173 |
| LEND NOT (operando) | LOOP END NOT | 810 | --- | --- | --- | 1153 |
| LEND (condizione di ingresso) | LOOP END | 810 | --- | --- | --- | 1153 |
| LEND (operando) | LOOP END | 810 | --- | --- | --- | 1153 |
| LIFO | LAST IN FIRST OUT | 634 | @LIFO | --- | --- | 675 |
| LINE | COLUMN TO LINE | 063 | @LINE | --- | --- | 494 |
| LMT | LIMIT CONTROL | 680 | @LMT | --- | --- | 741 |
| LOG | LOGARITHM | 468 | @LOG | --- | --- | 597 |
| LOGD | DOUBLE LOGARITHM | 859 | @LOGD | --- | --- | 653 |
| LOOP | LOOP | 809 | --- | --- | --- | 1153 |

M

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---|-----------------|---------------------------------------|--|-------------------------|--------|
| MAX | FIND MAXIMUM | 182 | @MAX | --- | --- | 689 |
| MCMP | MULTIPLE COMPARE | 019 | @MCMP | --- | --- | 299 |
| MCRO | MACRO | 099 | @MCRO | --- | --- | 779 |
| MID\$ | GET STRING MIDDLE | 654 | @MID\$ | --- | --- | 1168 |
| MILC | MULTI-INTERLOCK CLEAR | 519 | --- | --- | --- | 205 |
| MILH | MULTI-INTERLOCK DIFFERENTIATION HOLD | 517 | --- | --- | --- | 205 |
| MILR | MULTI-INTERLOCK DIFFERENTIATION RELEASE | 518 | --- | --- | --- | 205 |
| MIN | FIND MINIMUM | 183 | @MIN | --- | --- | 693 |
| MLPX | DATA DECODER | 076 | @MLPX | --- | --- | 477 |
| MOV | MOVE | 021 | @MOV | --- | !MOV | 315 |
| MOV\$ | MOVE STRING | 664 | @MOV\$ | --- | --- | 1159 |
| MOVB | MOVE BIT | 082 | @MOVB | --- | --- | 321 |
| MOVBC | MOVE BIT | 568 | @MOVBC | --- | --- | 1211 |
| MOVD | MOVE DIGIT | 083 | @MOVD | --- | --- | 323 |
| MOVL | DOUBLE MOVE | 498 | @MOVL | --- | --- | 318 |
| MOVR | MOVE TO REGISTER | 560 | @MOVR | --- | --- | 340 |
| MOVRW | MOVE TIMER/ COUNTER PV TO REGISTER | 561 | --- | --- | --- | 342 |
| MSG | DISPLAY MESSAGE | 046 | @MSG | --- | --- | 1058 |
| MSKR | READ INTERRUPT MASK | 692 | @MSKR | --- | --- | 805 |
| MSKS | SET INTERRUPT MASK | 690 | @MSKS | --- | --- | 798 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| MTIM | MULTI-OUTPUT TIMER | 543 | --- | --- | --- | 254 |
| MTIMX | MULTI-OUTPUT TIMER | 554 | --- | --- | --- | 254 |
| MTR | MATRIX INPUT | 213 | --- | --- | --- | 904 |
| MVN | MOVE NOT | 022 | @MVN | --- | --- | 317 |
| MVNL | DOUBLE MOVE NOT | 499 | @MVNL | --- | --- | 320 |

N

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| NASL | SHIFT N-BITS LEFT | 580 | @NASL | --- | --- | 381 |
| NASR | SHIFT N-BITS RIGHT | 581 | @NASR | --- | --- | 387 |
| NEG | 2'S COMPLEMENT | 160 | @NEG | --- | --- | 472 |
| NEGL | DOUBLE 2'S COMPLEMENT | 161 | @NEGL | --- | --- | 474 |
| NEXT | FOR-NEXT LOOPS | 513 | --- | --- | --- | 229 |
| NOP | NO OPERATION | 000 | --- | --- | --- | 198 |
| NOT | NOT | 520 | --- | --- | --- | 172 |
| NSFL | SHIFT N-BIT DATA LEFT | 578 | @NSFL | --- | --- | 377 |
| NSFR | SHIFT N-BIT DATA RIGHT | 579 | @NSFR | --- | --- | 379 |
| NSLL | DOUBLE SHIFT N-BITS LEFT | 582 | @NSLL | --- | --- | 384 |
| NSRL | DOUBLE SHIFT N-BITS RIGHT | 583 | @NSRL | --- | --- | 389 |

O

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| OR | OR | --- | @OR | %OR | !OR | 161 |
| OR < | OR LESS THAN | 310 | --- | --- | --- | 275 |
| OR <\$ | OR STRING LESS THAN | 672 | --- | --- | --- | 1187 |
| OR <> | OR NOT EQUAL | 305 | --- | --- | --- | 275 |
| OR <>\$ | OR STRING NOT EQUAL | 671 | --- | --- | --- | 1187 |
| OR <>D | OR DOUBLE FLOATING NOT EQUAL | 336 | --- | --- | --- | 657 |
| OR <>DT | OR TIME NOT EQUAL | 342 | --- | --- | --- | 281 |
| OR <>F | OR FLOATING NOT EQUAL | 330 | --- | --- | --- | 600 |
| OR <>L | OR DOUBLE NOT EQUAL | 306 | --- | --- | --- | 275 |
| OR <>S | OR SIGNED NOT EQUAL | 307 | --- | --- | --- | 275 |
| OR <>SL | OR DOUBLE SIGNED NOT EQUAL | 308 | --- | --- | --- | 275 |
| OR <D | OR DOUBLE FLOATING LESS THAN | 337 | --- | --- | --- | 657 |
| OR <DT | OR TIME LESS THAN | 343 | --- | --- | --- | 281 |
| OR <F | OR FLOATING LESS THAN | 331 | --- | --- | --- | 600 |
| OR <L | OR DOUBLE LESS THAN | 311 | --- | --- | --- | 275 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--|-----------------|---------------------------------------|--|-------------------------|--------|
| OR <S | OR SIGNED LESS THAN | 312 | --- | --- | --- | 275 |
| OR <SL | OR DOUBLE SIGNED LESS THAN | 313 | --- | --- | --- | 275 |
| OR = | OR EQUAL | 300 | --- | --- | --- | 275 |
| OR =\$ | OR STRING EQUALS | 670 | --- | --- | --- | 1187 |
| OR =D | OR DOUBLE FLOATING EQUAL | 335 | --- | --- | --- | 657 |
| OR =DT | OR TIME EQUAL | 341 | --- | --- | --- | 281 |
| OR =F | OR FLOATING EQUAL | 329 | --- | --- | --- | 600 |
| OR =L | OR DOUBLE EQUAL | 301 | --- | --- | --- | 275 |
| OR =S | OR SIGNED EQUAL | 302 | --- | --- | --- | 275 |
| OR =SL | OR DOUBLE SIGNED EQUAL | 303 | --- | --- | --- | 275 |
| OR > | OR GREATER THAN | 320 | --- | --- | --- | 275 |
| OR >\$ | OR STRING GREATER THAN | 674 | --- | --- | --- | 1187 |
| OR >D | OR DOUBLE FLOATING GREATER THAN | 339 | --- | --- | --- | 657 |
| OR >DT | OR TIME GREATER THAN | 345 | --- | --- | --- | 281 |
| OR >F | OR FLOATING GREATER THAN | 333 | --- | --- | --- | 600 |
| OR >L | OR DOUBLE GREATER THAN | 321 | --- | --- | --- | 275 |
| OR >S | OR SIGNED GREATER THAN | 322 | --- | --- | --- | 275 |
| OR >SL | OR DOUBLE SIGNED GREATER THAN | 323 | --- | --- | --- | 275 |
| OR LD | OR LOAD | --- | --- | --- | --- | 166 |
| OR NOT | OR NOT | --- | --- | --- | IOR NOT | 163 |
| OR TST | OR BIT TEST | 350 | --- | --- | --- | 174 |
| OR TSTN | OR BIT TEST | 351 | --- | --- | --- | 174 |
| OR <= | OR LESS THAN OR EQUAL | 315 | --- | --- | --- | 275 |
| OR <=\$ | OR STRING LESS THAN OR EQUALS | 673 | --- | --- | --- | 1187 |
| OR <=D | OR DOUBLE FLOATING LESS THAN OR EQUAL | 338 | --- | --- | --- | 657 |
| OR <=DT | OR TIME LESS THAN OR EQUAL | 344 | --- | --- | --- | 281 |
| OR <=F | OR FLOATING LESS THAN OR EQUAL | 332 | --- | --- | --- | 600 |
| OR <=L | OR DOUBLE LESS THAN OR EQUAL | 316 | --- | --- | --- | 275 |
| OR <=S | OR SIGNED LESS THAN OR EQUAL | 317 | --- | --- | --- | 275 |
| OR <=SL | OR DOUBLE SIGNED LESS THAN OR EQUAL | 318 | --- | --- | --- | 275 |
| OR >= | OR GREATER THAN OR EQUAL | 325 | --- | --- | --- | 275 |
| OR >=\$ | OR STRING GREATER THAN OR EQUALS | 675 | --- | --- | --- | 1187 |
| OR >=D | OR DOUBLE FLOATING GREATER THAN OR EQUAL | 340 | --- | --- | --- | 657 |
| OR >=DT | OR TIME GREATER THAN OR EQUAL | 346 | --- | --- | --- | 281 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--|-----------------|---------------------------------------|--|-------------------------|--------|
| OR >=F | OR FLOATING GREATER THAN OR EQUAL | 334 | --- | --- | --- | 600 |
| OR >=L | OR DOUBLE GREATER THAN OR EQUAL | 326 | --- | --- | --- | 275 |
| OR >=S | OR SIGNED GREATER THAN OR EQUAL | 327 | --- | --- | --- | 275 |
| OR >=SL | OR DOUBLE SIGNED GREATER THAN OR EQUAL | 328 | --- | --- | --- | 275 |
| ORG | ORIGIN SEARCH | 889 | @ORG | --- | --- | 862 |
| ORW | LOGICAL OR | 035 | @ORW | --- | --- | 520 |
| ORWL | DOUBLE LOGICAL OR | 611 | @ORWL | --- | --- | 522 |
| OUT | OUTPUT | --- | --- | --- | !OUT | 177 |
| OUTB | SINGLE BIT OUTPUT | 534 | @OUTB | --- | !OUTB | 195 |
| OUT NOT | OUTPUT NOT | --- | --- | --- | !OUT NOT | 178 |

P

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| PID | PID CONTROL | 190 | --- | --- | --- | 720 |
| PIDAT | PID CONTROL WITH AUTOTUNING | 191 | --- | --- | --- | 731 |
| PMCR | PROTOCOL MACRO | 260 | @PMCR | --- | --- | 928 |
| PRV | HIGH-SPEED COUNTER PV READ | 881 | @PRV | --- | --- | 827 |
| PRV2 | COUNTER FREQUENCY CONVERT | 883 | @PRV2 | --- | --- | 833 |
| PULS | SET PULSES | 886 | @PULS | --- | --- | 846 |
| PLS2 | PULSE OUTPUT | 887 | @PLS2 | --- | --- | 849 |
| PUSH | PUSH ONTO STACK | 632 | @PUSH | --- | --- | 669 |
| PWM | PULSE WITH VARIABLE DUTY FACTOR | 891 | @PWM | --- | --- | 865 |
| PWR | EXPONENTIAL POWER | 840 | @PWR | --- | --- | 599 |
| PWRD | DOUBLE EXPONENTIAL POWER | 860 | @PWRD | --- | --- | 655 |

R

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|----------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| RAD | DEGREES TO RADIANS | 458 | @RAD | --- | --- | 597 |
| RADD | DOUBLE DEGREES TO RADIANS | 849 | @RADD | --- | --- | 634 |
| RECV | NETWORK RECEIVE | 098 | @RECV | --- | --- | 997 |
| RET | SUBROUTINE RETURN | 093 | --- | --- | --- | 786 |
| RGHT\$ | GET STRING RIGHT | 653 | @RGHT\$ | --- | --- | 1166 |
| RLNC | ROTATE LEFT WITHOUT CARRY | 574 | @RLNC | --- | --- | 367 |
| RLNL | DOUBLE ROTATE LEFT WITHOUT CARRY | 576 | @RLNL | --- | --- | 369 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--|-----------------|---------------------------------------|--|-------------------------|--------|
| ROL | ROTATE LEFT | 027 | @ROL | --- | --- | 360 |
| ROLL | DOUBLE ROTATE LEFT | 572 | @ROLL | --- | --- | 362 |
| ROOT | BCD SQUARE ROOT | 072 | @ROOT | --- | --- | 536 |
| ROR | ROTATE RIGHT | 028 | @ROR | --- | --- | 364 |
| RORL | DOUBLE ROTATE RIGHT | 573 | @RORL | --- | --- | 365 |
| ROTB | BINARY ROOT | 620 | @ROTB | --- | --- | 534 |
| RPLC\$ | REPLACE IN STRING | 661 | @RPLC\$ | --- | --- | 1175 |
| RRNC | ROTATE RIGHT WITHOUT CARRY | 575 | @RRNC | --- | --- | 371 |
| RRNL | DOUBLE ROTATE RIGHT WITHOUT CARRY | 577 | @RRNL | --- | --- | 372 |
| RSET | RESET | --- | @RSET | %RSET | IRSET | 187 |
| RSTA | MULTIPLE BIT RESET | 531 | @RSTA | --- | --- | 189 |
| RSTB | SINGLE BIT RESET | 533 | @RSTB | --- | IRSTB | 192 |
| RXD | RECEIVE | 235 | @RXD | --- | --- | 944 |
| RXDU | RECEIVE VIA SERIAL COMMUNICATIONS UNIT | 255 | @RXDU | --- | --- | 960 |

S

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| SBN | SUBROUTINE ENTRY | 092 | --- | --- | --- | 783 |
| SBS | SUBROUTINE CALL | 091 | @SBS | --- | --- | 773 |
| SCL | SCALING | 194 | @SCL | --- | --- | 757 |
| SCL2 | SCALING 2 | 486 | @SCL2 | --- | --- | 762 |
| SCL3 | SCALING 3 | 487 | @SCL3 | --- | --- | 766 |
| SDEC | 7-SEGMENT DECODER | 078 | @SDEC | --- | --- | 928 |
| SDEL | STACK DATA DELETE | 642 | @SDEL | --- | --- | 716 |
| SEC | HOURS TO SECONDS | 065 | @SEC | --- | --- | 1068 |
| SEND | NETWORK SEND | 090 | @SEND | --- | --- | 991 |
| SET | SET | --- | @SET | %SET | ISET | 187 |
| SETA | MULTIPLE BIT SET | 530 | @SETA | --- | --- | 189 |
| SETB | SINGLE BIT SET | 532 | @SETB | --- | ISETB | 192 |
| SETR | SET RECORD LOCATION | 635 | @SETR | --- | --- | 681 |
| SFT | SHIFT REGISTER | 010 | --- | --- | --- | 345 |
| SFTR | REVERSIBLE SHIFT REGISTER | 084 | @SFTR | --- | --- | 346 |
| SIGN | 16-BIT TO 32-BIT SIGNED BINARY | 600 | @SIGN | --- | --- | 476 |
| SIN | SINE | 460 | @SIN | --- | --- | 581 |
| SIND | DOUBLE SINE | 851 | @SIND | --- | --- | 637 |
| SINS | STACK DATA INSERT | 641 | @SINS | --- | --- | 713 |
| SLD | ONE DIGIT SHIFT LEFT | 074 | @SLD | --- | --- | 374 |
| SNUM | STACK SIZE READ | 638 | @SNUM | --- | --- | 704 |
| SNXT | STEP START | 009 | --- | --- | --- | 868 |
| SPED | SPEED OUTPUT | 885 | @SPED | --- | --- | 841 |
| SQRT | SQUARE ROOT | 466 | @SQRT | --- | --- | 593 |
| SQRTD | DOUBLE SQUARE ROOT | 857 | @SQRTD | --- | --- | 649 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| SRCH | DATA SEARCH | 181 | @SRCH | --- | --- | 685 |
| SRD | ONE DIGIT SHIFT RIGHT | 075 | @SRD | --- | --- | 376 |
| SREAD | STACK DATA READ | 639 | @SREAD | --- | --- | 707 |
| SSET | SET STACK | 630 | @SSET | --- | --- | 666 |
| STC | SET CARRY | 040 | @STC | --- | --- | 1104 |
| STEP | STEP DEFINE | 008 | --- | --- | --- | 868 |
| STUP | CHANGE SERIAL PORT SETUP | 237 | @STUP | --- | --- | 968 |
| SUM | SUM | 184 | @SUM | --- | --- | 697 |
| SWAP | SWAP BYTES | 637 | @SWAP | --- | --- | 687 |
| SWRIT | STACK DATA WRITE | 640 | @SWRIT | --- | --- | 710 |

T

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---|-----------------|---------------------------------------|--|-------------------------|--------|
| TAN | TANGENT | 462 | @TAN | --- | --- | 585 |
| TAND | DOUBLE TANGENT | 853 | @TAND | --- | --- | 641 |
| TCMP | TABLE COMPARE | 085 | @TCMP | --- | --- | 301 |
| TIM | TIMER | --- | --- | --- | --- | 235 |
| TIMH | HIGH-SPEED TIMER | 015 | --- | --- | --- | 240 |
| TIMHX | HIGH-SPEED TIMER | 551 | --- | --- | --- | 240 |
| TIML | LONG TIMER | 542 | --- | --- | --- | 251 |
| TIMLX | LONG TIMER | 553 | --- | --- | --- | 251 |
| TIMW | TIMER WAIT | 813 | --- | --- | --- | 1144 |
| TIMWX | TIMER WAIT | 816 | --- | --- | --- | 1144 |
| TIMX | TIMER | 505 | --- | --- | --- | 235 |
| TKOF | TASK OFF | 821 | @TKOF | --- | --- | 1196 |
| TKON | TASK ON | 820 | @TKON | --- | --- | 1192 |
| TKY | TEN KEY INPUT | 211 | @TKY | --- | --- | 896 |
| TMHH | ONE-MS TIMER | 540 | --- | --- | --- | 244 |
| TMHHX | ONE-MS TIMER | 552 | --- | --- | --- | 244 |
| TMHW | HIGH-SPEED TIMER WAIT | 815 | --- | --- | --- | 1150 |
| TMHWX | HIGH-SPEED TIMER WAIT | 817 | --- | --- | --- | 1150 |
| TOCV | CONVERT ADDRESS TO CV | 285 | @TOCV | --- | --- | 1117 |
| TPO | TIME-PROPORTIONAL OUTPUT | 685 | --- | --- | --- | 749 |
| TRSM | TRACE MEMORY SAMPLING | 045 | --- | --- | --- | 1075 |
| TTIM | ACCUMULATIVE TIMER | 087 | --- | --- | --- | 247 |
| TTIMX | ACCUMULATIVE TIMER | 555 | --- | --- | --- | 247 |
| TXD | TRANSMIT | 236 | @TXD | --- | --- | 937 |
| TXDU | TRANSMIT VIA SERIAL COMMUNICATIONS UNIT | 256 | @TXDU | --- | --- | 952 |

U

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--------------|-----------------|---------------------------------------|--|-------------------------|--------|
| UP | CONDITION ON | 521 | --- | --- | --- | 173 |

W

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-------------------------------|---------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| WAIT NOT (operando) | ONE CYCLE AND WAIT NOT | 805 | --- | --- | --- | 1140 |
| WAIT (condizione di ingresso) | ONE CYCLE AND WAIT | 805 | --- | --- | --- | 1140 |
| WAIT (operando) | ONE CYCLE AND WAIT | 805 | --- | --- | --- | 1140 |
| WDT | EXTEND MAXIMUM CYCLE TIME | 094 | @WDT | --- | --- | 1108 |
| WSFT | WORD SHIFT | 016 | @WSFT | --- | --- | 352 |

X

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|-----------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| XCGL | DOUBLE DATA EXCHANGE | 562 | @XCGL | --- | --- | 334 |
| XCHG | DATA EXCHANGE | 073 | @XCHG | --- | --- | 333 |
| XCHG\$ | EXCHANGE STRING | 665 | @XCHG\$ | --- | --- | 1180 |
| XFER | BLOCK TRANSFER | 070 | @XFER | --- | --- | 328 |
| XFERC | BLOCK TRANSFER | 565 | @XFERC | --- | --- | 1201 |
| XFRB | MULTIPLE BIT TRANSFER | 062 | @XFRB | --- | --- | 326 |
| XNRL | DOUBLE EXCLUSIVE NOR | 613 | @XNRL | --- | --- | 529 |
| XNRW | EXCLUSIVE NOR | 037 | @XNRW | --- | --- | 528 |
| XORL | DOUBLE EXCLUSIVE OR | 612 | @XORL | --- | --- | 526 |
| XORW | EXCLUSIVE OR | 036 | @XORW | --- | --- | 524 |

Z

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| ZCP | AREA RANGE COMPARE | 088 | --- | --- | --- | 310 |
| ZCPL | DOUBLE AREA RANGE COMPARE | 116 | --- | --- | --- | 313 |
| ZONE | DEAD ZONE CONTROL | 682 | @ZONE | --- | --- | 746 |

Simboli

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---------------------------------|-----------------|---------------------------------------|--|-------------------------|--------|
| 7SEG | 7-SEGMENT DISPLAY OUTPUT | 214 | --- | --- | --- | 908 |
| + | SIGNED BINARY ADD WITHOUT CARRY | 400 | @+ | --- | --- | 410 |
| +\$ | CONCATENATE STRING | 656 | @+\$ | --- | --- | 1161 |
| ++ | INCREMENT BINARY | 590 | @++ | --- | --- | 393 |
| ++B | INCREMENT BCD | 594 | @++B | --- | --- | 401 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|--|-----------------|---------------------------------------|--|-------------------------|--------|
| ++BL | DOUBLE INCREMENT BCD | 595 | @++BL | --- | --- | 403 |
| ++L | DOUBLE INCREMENT BINARY | 591 | @++L | --- | --- | 395 |
| +B | BCD ADD WITHOUT CARRY | 404 | @+B | --- | --- | 418 |
| +BC | BCD ADD WITH CARRY | 406 | @+BC | --- | --- | 421 |
| +BCL | DOUBLE BCD ADD WITH CARRY | 407 | @+BCL | --- | --- | 423 |
| +BL | DOUBLE BCD ADD WITHOUT CARRY | 405 | @+BL | --- | --- | 419 |
| +C | SIGNED BINARY ADD WITH CARRY | 402 | @+C | --- | --- | 414 |
| +CL | DOUBLE SIGNED BINARY ADD WITH CARRY | 403 | @+CL | --- | --- | 416 |
| +D | DOUBLE FLOATING-POINT ADD | 845 | @+D | --- | --- | 626 |
| +F | FLOATING-POINT ADD | 454 | @+F | --- | --- | 570 |
| +L | DOUBLE SIGNED BINARY ADD WITHOUT CARRY | 401 | @+L | --- | --- | 412 |
| - | SIGNED BINARY SUBTRACT WITHOUT CARRY | 410 | @- | --- | --- | 424 |
| -- | DECREMENT BINARY | 592 | @-- | --- | --- | 397 |
| --B | DECREMENT BCD | 596 | @--B | --- | --- | 405 |
| --BL | DOUBLE DECREMENT BCD | 597 | @--BL | --- | --- | 407 |
| --L | DOUBLE DECREMENT BINARY | 593 | @--L | --- | --- | 399 |
| -B | BCD SUBTRACT WITHOUT CARRY | 414 | @-B | --- | --- | 435 |
| -BC | BCD SUBTRACT WITH CARRY | 416 | @-BC | --- | --- | 440 |
| -BCL | DOUBLE BCD SUBTRACT WITH CARRY | 417 | @-BCL | --- | --- | 441 |
| -BL | DOUBLE BCD SUBTRACT WITHOUT CARRY | 415 | @-BL | --- | --- | 436 |
| -C | SIGNED BINARY SUBTRACT WITH CARRY | 412 | @-C | --- | --- | 430 |
| -CL | DOUBLE SIGNED BINARY SUBTRACT WITH CARRY | 413 | @-CL | --- | --- | 432 |
| -D | DOUBLE FLOATING-POINT SUBTRACT | 846 | @-D | --- | --- | 628 |
| -F | FLOATING-POINT SUBTRACT | 455 | @-F | --- | --- | 572 |
| * | SIGNED BINARY MULTIPLY | 420 | @* | --- | --- | 443 |
| *B | BCD MULTIPLY | 424 | @*B | --- | --- | 450 |
| *BL | DOUBLE BCD MULTIPLY | 425 | @*BL | --- | --- | 452 |
| *D | DOUBLE FLOATING-POINT MULTIPLY | 847 | @*D | --- | --- | 630 |
| *F | FLOATING-POINT MULTIPLY | 456 | @*F | --- | --- | 574 |

| Codice mnemonico | Istruzione | Codice funzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|------------------|---|-----------------|---------------------------------------|--|-------------------------|--------|
| *L | DOUBLE SIGNED BINARY MULTIPLY | 421 | @*L | --- | --- | 445 |
| *U | UNSIGNED BINARY MULTIPLY | 422 | @*U | --- | --- | 447 |
| *UL | DOUBLE UNSIGNED BINARY MULTIPLY | 423 | @*UL | --- | --- | 449 |
| -L | DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY | 411 | @-L | --- | --- | 426 |
| / | SIGNED BINARY DIVIDE | 430 | @/ | --- | --- | 454 |
| /B | BCD DIVIDE | 434 | @/B | --- | --- | 462 |
| /BL | DOUBLE BCD DIVIDE | 435 | @/BL | --- | --- | 464 |
| /D | DOUBLE FLOATING-POINT DIVIDE | 848 | @/D | --- | --- | 632 |
| /F | FLOATING POINT DIVIDE | 457 | @/F | --- | --- | 576 |
| /L | DOUBLE SIGNED BINARY DIVIDE | 431 | @/L | --- | --- | 456 |
| /U | UNSIGNED BINARY DIVIDE | 432 | @/U | --- | --- | 458 |
| /UL | DOUBLE UNSIGNED BINARY DIVIDE | 433 | @/UL | --- | --- | 460 |

2-4 Elenco delle istruzioni in base al codice funzione

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|-----------------------------|---------------------------------------|--|-------------------------|--------|
| --- | LD | LOAD | @LD | %LD | !LD | 153 |
| --- | LD NOT | LOAD NOT | --- | --- | !LD NOT | 155 |
| --- | AND | AND | @AND | %AND | !AND | 157 |
| --- | AND NOT | AND NOT | --- | --- | !AND NOT | 159 |
| --- | OR | OR | @OR | %OR | !OR | 161 |
| --- | OR NOT | OR NOT | --- | --- | !OR NOT | 163 |
| --- | AND LD | AND LOAD | --- | --- | --- | 164 |
| --- | OR LD | OR LOAD | --- | --- | --- | 166 |
| --- | OUT | OUTPUT | --- | --- | !OUT | 177 |
| --- | OUT NOT | OUTPUT NOT | --- | --- | !OUT NOT | 178 |
| --- | SET | SET | @SET | %SET | !SET | 187 |
| --- | RSET | RESET | @RSET | %RSET | !RSET | 187 |
| --- | TIM | TIMER | --- | --- | --- | 235 |
| --- | TIMX | TIMER | --- | --- | --- | 235 |
| --- | CNT | COUNTER | --- | --- | --- | 260 |
| 000 | NOP | NO OPERATION | --- | --- | --- | 198 |
| 001 | END | END | --- | --- | --- | 197 |
| 002 | IL | INTERLOCK | --- | --- | --- | 201 |
| 003 | ILC | INTERLOCK CLEAR | --- | --- | --- | 201 |
| 004 | JMP | JUMP | --- | --- | --- | 219 |
| 005 | JME | JUMP END | --- | --- | --- | 219 |
| 006 | FAL | FAILURE ALARM | @FAL | --- | --- | 1079 |
| 007 | FALS | SEVERE FAILURE ALARM | --- | --- | --- | 1087 |
| 008 | STEP | STEP DEFINE | --- | --- | --- | 868 |
| 009 | SNXT | STEP START | --- | --- | --- | 868 |
| 010 | SFT | SHIFT REGISTER | --- | --- | --- | 345 |
| 011 | KEEP | KEEP | --- | --- | !KEEP | 180 |
| 012 | CNTR | REVERSIBLE COUNTER | --- | --- | --- | 263 |
| 013 | DIFU | DIFFERENTIATE UP | --- | --- | !DIFU | 184 |
| 014 | DIFD | DIFFERENTIATE DOWN | --- | --- | !DIFD | 184 |
| 015 | TIMH | HIGH-SPEED TIMER | --- | --- | --- | 240 |
| 016 | WSFT | WORD SHIFT | @WSFT | --- | --- | 352 |
| 017 | ASFT | ASYNCHRONOUS SHIFT REGISTER | @ASFT | --- | --- | 349 |
| 019 | MCMP | MULTIPLE COMPARE | @MCMP | --- | --- | 299 |
| 020 | CMP | UNSIGNED COMPARE | --- | --- | !CMP | 287 |
| 021 | MOV | MOVE | @MOV | --- | !MOV | 315 |
| 022 | MVN | MOVE NOT | @MVN | --- | --- | 317 |
| 023 | BIN | BCD-TO-BINARY | @BIN | --- | --- | 466 |
| 024 | BCD | BINARY-TO-BCD | @BCD | --- | --- | 469 |
| 025 | ASL | ARITHMETIC SHIFT LEFT | @ASL | --- | --- | 354 |
| 026 | ASR | ARITHMETIC SHIFT RIGHT | @ASR | --- | --- | 357 |
| 027 | ROL | ROTATE LEFT | @ROL | --- | --- | 360 |
| 028 | ROR | ROTATE RIGHT | @ROR | --- | --- | 364 |
| 029 | COM | COMPLEMENT | @COM | --- | --- | 531 |
| 034 | ANDW | LOGICAL AND | @ANDW | --- | --- | 517 |
| 035 | ORW | LOGICAL OR | @ORW | --- | --- | 520 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|-----------------------------|---------------------------------------|--|-------------------------|--------|
| 036 | XORW | EXCLUSIVE OR | @XORW | --- | --- | 524 |
| 037 | XNRW | EXCLUSIVE NOR | @XNRW | --- | --- | 528 |
| 040 | STC | SET CARRY | @STC | --- | --- | 1104 |
| 041 | CLC | CLEAR CARRY | @CLC | --- | --- | 1105 |
| 045 | TRSM | TRACE MEMORY SAMPLING | --- | --- | --- | 1075 |
| 046 | MSG | DISPLAY MESSAGE | @MSG | --- | --- | 1058 |
| 058 | BINL | DOUBLE BCD-TO-DOUBLE BINARY | @BINL | --- | --- | 467 |
| 059 | BCDL | DOUBLE BINARY-TO-BCD | @BCDL | --- | --- | 470 |
| 060 | CMPL | DOUBLE UNSIGNED COMPARE | --- | --- | --- | 290 |
| 062 | XFRB | MULTIPLE BIT TRANSFER | @XFRB | --- | --- | 326 |
| 063 | LINE | COLUMN TO LINE | @LINE | --- | --- | 494 |
| 064 | COLM | LINE TO COLUMN | @COLM | --- | --- | 496 |
| 065 | SEC | HOURS TO SECONDS | @SEC | --- | --- | 1068 |
| 066 | HMS | SECONDS TO HOURS | @HMS | --- | --- | 1070 |
| 067 | BCNT | BIT COUNTER | @BCNT | --- | --- | 556 |
| 068 | BCMP | UNSIGNED BLOCK COMPARE | @BCMP | --- | --- | 304 |
| 069 | APR | ARITHMETIC PROCESS | @APR | --- | --- | 540 |
| 070 | XFER | BLOCK TRANSFER | @XFER | --- | --- | 328 |
| 071 | BSET | BLOCK SET | @BSET | --- | --- | 331 |
| 072 | ROOT | BCD SQUARE ROOT | @ROOT | --- | --- | 536 |
| 073 | XCHG | DATA EXCHANGE | @XCHG | --- | --- | 333 |
| 074 | SLD | ONE DIGIT SHIFT LEFT | @SLD | --- | --- | 374 |
| 075 | SRD | ONE DIGIT SHIFT RIGHT | @SRD | --- | --- | 376 |
| 076 | MLPX | DATA DECODER | @MLPX | --- | --- | 477 |
| 077 | DMPX | DATA ENCODER | @DMPX | --- | --- | 482 |
| 078 | SDEC | 7-SEGMENT DECODER | @SDEC | --- | --- | 928 |
| 079 | FDIV | FLOATING POINT DIVIDE | @FDIV | --- | --- | 552 |
| 080 | DIST | SINGLE WORD DISTRIBUTE | @DIST | --- | --- | 336 |
| 081 | COLL | DATA COLLECT | @COLL | --- | --- | 338 |
| 082 | MOVB | MOVE BIT | @MOVB | --- | --- | 321 |
| 083 | MOVD | MOVE DIGIT | @MOVD | --- | --- | 323 |
| 084 | SFTR | REVERSIBLE SHIFT REGISTER | @SFTR | --- | --- | 346 |
| 085 | TCMP | TABLE COMPARE | @TCMP | --- | --- | 301 |
| 086 | ASC | ASCII CONVERT | @ASC | --- | --- | 486 |
| 087 | TTIM | ACCUMULATIVE TIMER | --- | --- | --- | 247 |
| 088 | ZCP | AREA RANGE COMPARE | --- | --- | --- | 310 |
| 090 | SEND | NETWORK SEND | @SEND | --- | --- | 991 |
| 091 | SBS | SUBROUTINE CALL | @SBS | --- | --- | 773 |
| 092 | SBN | SUBROUTINE ENTRY | --- | --- | --- | 783 |
| 093 | RET | SUBROUTINE RETURN | --- | --- | --- | 786 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|---|---------------------------------------|--|-------------------------|--------|
| 094 | WDT | EXTEND MAXIMUM CYCLE TIME | @WDT | --- | --- | 1108 |
| 096 | BPRG | BLOCK PROGRAM BEGIN | --- | --- | --- | 1128 |
| 097 | IORF | I/O REFRESH | @IORF | --- | --- | 885 |
| 098 | RECV | NETWORK RECEIVE | @RECV | --- | --- | 997 |
| 099 | MCRO | MACRO | @MCRO | --- | --- | 779 |
| 114 | CPS | SIGNED BINARY COMPARE | --- | --- | !CPS | 293 |
| 115 | CPSL | DOUBLE SIGNED BINARY COMPARE | --- | --- | --- | 296 |
| 116 | ZCPL | DOUBLE AREA RANGE COMPARE | --- | --- | --- | 313 |
| 160 | NEG | 2'S COMPLEMENT | @NEG | --- | --- | 472 |
| 161 | NEGL | DOUBLE 2'S COMPLEMENT | @NEGL | --- | --- | 474 |
| 162 | HEX | ASCII TO HEX | @HEX | --- | --- | 490 |
| 180 | FCS | FRAME CHECKSUM | @FCS | --- | --- | 700 |
| 181 | SRCH | DATA SEARCH | @SRCH | --- | --- | 685 |
| 182 | MAX | FIND MAXIMUM | @MAX | --- | --- | 689 |
| 183 | MIN | FIND MINIMUM | @MIN | --- | --- | 693 |
| 184 | SUM | SUM | @SUM | --- | --- | 697 |
| 190 | PID | PID CONTROL | --- | --- | --- | 720 |
| 191 | PIDAT | PID CONTROL WITH AUTOTUNING | --- | --- | --- | 731 |
| 194 | SCL | SCALING | @SCL | --- | --- | 757 |
| 195 | AVG | AVERAGE | --- | --- | --- | 769 |
| 210 | DSW | DIGITAL SWITCH INPUT | --- | --- | --- | 890 |
| 211 | TKY | TEN KEY INPUT | @TKY | --- | --- | 896 |
| 212 | HKY | HEXADECIMAL KEY INPUT | --- | --- | --- | 899 |
| 213 | MTR | MATRIX INPUT | --- | --- | --- | 904 |
| 214 | 7SEG | 7-SEGMENT DISPLAY OUTPUT | --- | --- | --- | 908 |
| 222 | IORD | INTELLIGENT I/O READ | @IORD | --- | --- | 913 |
| 223 | IOWR | INTELLIGENT I/O WRITE | @IOWR | --- | --- | 917 |
| 226 | DLNK | CPU BUS UNIT I/O REFRESH | @DLNK | --- | --- | 921 |
| 235 | RXD | RECEIVE | @RXD | --- | --- | 944 |
| 236 | TXD | TRANSMIT | @TXD | --- | --- | 937 |
| 255 | RXDU | RECEIVE VIA SERIAL COMMUNICATIONS UNIT | @RXDU | --- | --- | 960 |
| 256 | TXDU | TRANSMIT VIA SERIAL COMMUNICATIONS UNIT | @TXDU | --- | --- | 952 |
| 237 | STUP | CHANGE SERIAL PORT SETUP | @STUP | --- | --- | 968 |
| 260 | PMCR | PROTOCOL MACRO | @PMCR | --- | --- | 928 |
| 269 | FPD | FAILURE POINT DETECTION | --- | --- | --- | 1095 |
| 281 | EMBC | SELECT EM BANK | @EMBC | --- | --- | 1106 |
| 282 | CCS | SAVE CONDITION FLAGS | @CCS | --- | --- | 1110 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|------------------------------|---------------------------------------|--|-------------------------|--------|
| 283 | CCL | LOAD CONDITION FLAGS | @CCL | --- | --- | 1112 |
| 284 | FRMCV | CONVERT ADDRESS FROM CV | @FRMCV | --- | --- | 1113 |
| 285 | TOCV | CONVERT ADDRESS TO CV | @TOCV | --- | --- | 1117 |
| 286 | GETID | GET VARIABLE ID | @GETID | --- | --- | 1214 |
| 287 | IOSP | DISABLE PERIPHERAL SERVICING | @IOSP | --- | --- | 1121 |
| 288 | IOSR | ENABLE PERIPHERAL SERVICING | --- | --- | --- | 1123 |
| 300 | AND = | AND EQUAL | --- | --- | --- | 275 |
| 300 | LD = | LOAD EQUAL | --- | --- | --- | 275 |
| 300 | OR = | OR EQUAL | --- | --- | --- | 275 |
| 301 | AND =L | AND DOUBLE EQUAL | --- | --- | --- | 275 |
| 301 | LD =L | LOAD DOUBLE EQUAL | --- | --- | --- | 275 |
| 301 | OR =L | OR DOUBLE EQUAL | --- | --- | --- | 275 |
| 302 | AND =S | AND SIGNED EQUAL | --- | --- | --- | 275 |
| 302 | LD =S | LOAD SIGNED EQUAL | --- | --- | --- | 275 |
| 302 | OR =S | OR SIGNED EQUAL | --- | --- | --- | 275 |
| 303 | AND =SL | AND DOUBLE SIGNED EQUAL | --- | --- | --- | 275 |
| 303 | LD =SL | LOAD DOUBLE SIGNED EQUAL | --- | --- | --- | 275 |
| 303 | OR =SL | OR DOUBLE SIGNED EQUAL | --- | --- | --- | 275 |
| 305 | AND <> | AND NOT EQUAL | --- | --- | --- | 275 |
| 305 | LD <> | LOAD NOT EQUAL | --- | --- | --- | 275 |
| 305 | OR <> | OR NOT EQUAL | --- | --- | --- | 275 |
| 306 | AND <>L | AND DOUBLE NOT EQUAL | --- | --- | --- | 275 |
| 306 | LD <>L | LOAD DOUBLE NOT EQUAL | --- | --- | --- | 275 |
| 306 | OR <>L | OR DOUBLE NOT EQUAL | --- | --- | --- | 275 |
| 307 | AND <>S | AND SIGNED NOT EQUAL | --- | --- | --- | 275 |
| 307 | LD <>S | LOAD SIGNED NOT EQUAL | --- | --- | --- | 275 |
| 307 | OR <>S | OR SIGNED NOT EQUAL | --- | --- | --- | 275 |
| 308 | AND <>SL | AND DOUBLE SIGNED NOT EQUAL | --- | --- | --- | 275 |
| 308 | LD <>SL | LOAD DOUBLE SIGNED NOT EQUAL | --- | --- | --- | 275 |
| 308 | OR <>SL | OR DOUBLE SIGNED NOT EQUAL | --- | --- | --- | 275 |
| 310 | AND < | AND LESS THAN | --- | --- | --- | 275 |
| 310 | LD < | LOAD LESS THAN | --- | --- | --- | 275 |
| 310 | OR < | OR LESS THAN | --- | --- | --- | 275 |
| 311 | AND <L | AND DOUBLE LESS THAN | --- | --- | --- | 275 |
| 311 | LD <L | LOAD DOUBLE LESS THAN | --- | --- | --- | 275 |
| 311 | OR <L | OR DOUBLE LESS THAN | --- | --- | --- | 275 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|---------------------------------------|---------------------------------------|--|-------------------------|--------|
| 312 | AND <S | AND SIGNED LESS THAN | --- | --- | --- | 275 |
| 312 | LD <S | LOAD SIGNED LESS THAN | --- | --- | --- | 275 |
| 312 | OR <S | OR SIGNED LESS THAN | --- | --- | --- | 275 |
| 313 | AND <SL | AND DOUBLE SIGNED LESS THAN | --- | --- | --- | 275 |
| 313 | LD <SL | LOAD DOUBLE SIGNED LESS THAN | --- | --- | --- | 275 |
| 313 | OR <SL | OR DOUBLE SIGNED LESS THAN | --- | --- | --- | 275 |
| 315 | AND <= | AND LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 315 | LD <= | LOAD LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 315 | OR <= | OR LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 316 | AND <=L | AND DOUBLE LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 316 | LD <=L | LOAD DOUBLE LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 316 | OR <=L | OR DOUBLE LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 317 | AND <=S | AND SIGNED LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 317 | LD <=S | LOAD SIGNED LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 317 | OR <=S | OR SIGNED LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 318 | AND <=SL | AND DOUBLE SIGNED LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 318 | LD <=SL | LOAD DOUBLE SIGNED LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 318 | OR <=SL | OR DOUBLE SIGNED LESS THAN OR EQUAL | --- | --- | --- | 275 |
| 320 | AND > | AND GREATER THAN | --- | --- | --- | 275 |
| 320 | LD > | LOAD GREATER THAN | --- | --- | --- | 275 |
| 320 | OR > | OR GREATER THAN | --- | --- | --- | 275 |
| 321 | AND >L | AND DOUBLE GREATER THAN | --- | --- | --- | 275 |
| 321 | LD >L | LOAD DOUBLE GREATER THAN | --- | --- | --- | 275 |
| 321 | OR >L | OR DOUBLE GREATER THAN | --- | --- | --- | 275 |
| 322 | AND >S | AND SIGNED GREATER THAN | --- | --- | --- | 275 |
| 322 | LD >S | LOAD SIGNED GREATER THAN | --- | --- | --- | 275 |
| 322 | OR >S | OR SIGNED GREATER THAN | --- | --- | --- | 275 |
| 323 | AND >SL | AND DOUBLE SIGNED GREATER THAN | --- | --- | --- | 275 |
| 323 | LD >SL | LOAD DOUBLE SIGNED GREATER THAN | --- | --- | --- | 275 |
| 323 | OR >SL | OR DOUBLE SIGNED GREATER THAN | --- | --- | --- | 275 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|--|---------------------------------------|--|-------------------------|--------|
| 325 | AND >= | AND GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 325 | LD >= | LOAD GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 325 | OR >= | OR GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 326 | AND >=L | AND DOUBLE GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 326 | LD >=L | LOAD DOUBLE GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 326 | OR >=L | OR DOUBLE GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 327 | AND >=S | AND SIGNED GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 327 | LD >=S | LOAD SIGNED GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 327 | OR >=S | OR SIGNED GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 328 | AND >=SL | AND DOUBLE SIGNED GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 328 | LD >=SL | LOAD DOUBLE SIGNED GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 328 | OR >=SL | OR DOUBLE SIGNED GREATER THAN OR EQUAL | --- | --- | --- | 275 |
| 329 | AND =F | AND FLOATING EQUAL | --- | --- | --- | 600 |
| 329 | LD =F | LOAD FLOATING EQUAL | --- | --- | --- | 600 |
| 329 | OR =F | OR FLOATING EQUAL | --- | --- | --- | 600 |
| 330 | AND <>F | AND FLOATING NOT EQUAL | --- | --- | --- | 600 |
| 330 | LD <>F | LOAD FLOATING NOT EQUAL | --- | --- | --- | 600 |
| 330 | OR <>F | OR FLOATING NOT EQUAL | --- | --- | --- | 600 |
| 331 | AND <F | AND FLOATING LESS THAN | --- | --- | --- | 600 |
| 331 | LD <F | LOAD FLOATING LESS THAN | --- | --- | --- | 600 |
| 331 | OR <F | OR FLOATING LESS THAN | --- | --- | --- | 600 |
| 332 | AND <=F | AND FLOATING LESS THAN OR EQUAL | --- | --- | --- | 600 |
| 332 | LD <=F | LOAD FLOATING LESS THAN OR EQUAL | --- | --- | --- | 600 |
| 332 | OR <=F | OR FLOATING LESS THAN OR EQUAL | --- | --- | --- | 600 |
| 333 | AND >F | AND FLOATING GREATER THAN | --- | --- | --- | 600 |
| 333 | LD >F | LOAD FLOATING GREATER THAN | --- | --- | --- | 600 |
| 333 | OR >F | OR FLOATING GREATER THAN | --- | --- | --- | 600 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|--|---------------------------------------|--|-------------------------|--------|
| 334 | AND >=F | AND FLOATING GREATER THAN OR EQUAL | --- | --- | --- | 600 |
| 334 | LD >=F | LOAD FLOATING GREATER THAN OR EQUAL | --- | --- | --- | 600 |
| 334 | OR >=F | OR FLOATING GREATER THAN OR EQUAL | --- | --- | --- | 600 |
| 335 | AND =D | AND DOUBLE FLOATING EQUAL | --- | --- | --- | 657 |
| 335 | LD =D | LOAD DOUBLE FLOATING EQUAL | --- | --- | --- | 657 |
| 335 | OR =D | OR DOUBLE FLOATING EQUAL | --- | --- | --- | 657 |
| 336 | AND <>D | AND DOUBLE FLOATING NOT EQUAL | --- | --- | --- | 657 |
| 336 | LD <>D | LOAD DOUBLE FLOATING NOT EQUAL | --- | --- | --- | 657 |
| 336 | OR <>D | OR DOUBLE FLOATING NOT EQUAL | --- | --- | --- | 657 |
| 337 | AND <D | AND DOUBLE FLOATING LESS THAN | --- | --- | --- | 657 |
| 337 | LD <D | LOAD DOUBLE FLOATING LESS THAN | --- | --- | --- | 657 |
| 337 | OR <D | OR DOUBLE FLOATING LESS THAN | --- | --- | --- | 657 |
| 338 | AND <=D | AND DOUBLE FLOATING LESS THAN OR EQUAL | --- | --- | --- | 657 |
| 338 | LD <=D | LOAD DOUBLE FLOATING LESS THAN OR EQUAL | --- | --- | --- | 657 |
| 338 | OR <=D | OR DOUBLE FLOATING LESS THAN OR EQUAL | --- | --- | --- | 657 |
| 339 | AND >D | AND DOUBLE FLOATING GREATER THAN | --- | --- | --- | 657 |
| 339 | LD >D | LOAD DOUBLE FLOATING GREATER THAN | --- | --- | --- | 657 |
| 339 | OR >D | OR DOUBLE FLOATING GREATER THAN | --- | --- | --- | 657 |
| 340 | AND >=D | AND DOUBLE FLOATING GREATER THAN OR EQUAL | --- | --- | --- | 657 |
| 340 | LD >=D | LOAD DOUBLE FLOATING GREATER THAN OR EQUAL | --- | --- | --- | 657 |
| 340 | OR >=D | OR DOUBLE FLOATING GREATER THAN OR EQUAL | --- | --- | --- | 657 |
| 341 | AND =DT | AND TIME EQUAL | --- | --- | --- | 281 |
| 341 | LD =DT | LOAD TIME EQUAL | --- | --- | --- | 281 |
| 341 | OR =DT | OR TIME EQUAL | --- | --- | --- | 281 |
| 342 | AND <> DT | AND TIME NOT EQUAL | --- | --- | --- | 281 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|---|---------------------------------------|--|-------------------------|--------|
| 342 | LD <> DT | LOAD TIME NOT EQUAL | --- | --- | --- | 281 |
| 342 | OR <> DT | OR TIME NOT EQUAL | --- | --- | --- | 281 |
| 343 | AND < DT | AND TIME LESS THAN | --- | --- | --- | 281 |
| 343 | LD < DT | LOAD TIME LESS THAN | --- | --- | --- | 281 |
| 343 | OR < DT | OR TIME LESS THAN | --- | --- | --- | 281 |
| 344 | AND <= DT | AND TIME LESS THAN OR EQUAL | --- | --- | --- | 281 |
| 344 | LD <= DT | LD TIME LESS THAN OR EQUAL | --- | --- | --- | 281 |
| 344 | OR <= DT | OR TIME LESS THAN OR EQUAL | --- | --- | --- | 281 |
| 345 | AND > DT | AND TIME GREATER THAN | --- | --- | --- | 281 |
| 345 | LD > DT | LOAD TIME GREATER THAN | --- | --- | --- | 281 |
| 345 | OR > DT | OR TIME GREATER THAN | --- | --- | --- | 281 |
| 346 | AND >= DT | AND TIME GREATER THAN OR EQUAL | --- | --- | --- | 281 |
| 346 | LD >= DT | LOAD TIME GREATER THAN OR EQUAL | --- | --- | --- | 281 |
| 346 | OR >= DT | OR TIME GREATER THAN OR EQUAL | --- | --- | --- | 281 |
| 350 | AND TST | AND BIT TEST | --- | --- | --- | 174 |
| 350 | LD TST | LOAD BIT TEST | --- | --- | --- | 174 |
| 350 | OR TST | OR BIT TEST | --- | --- | --- | 174 |
| 351 | AND TSTN | AND BIT TEST NOT | --- | --- | --- | 174 |
| 351 | LD TSTN | LOAD BIT TEST NOT | --- | --- | --- | 174 |
| 351 | OR TSTN | OR BIT TEST NOT | --- | --- | --- | 174 |
| 400 | + | SIGNED BINARY ADD WITHOUT CARRY | @+ | --- | --- | 410 |
| 401 | +L | DOUBLE SIGNED BINARY ADD WITHOUT CARRY | @+L | --- | --- | 412 |
| 402 | +C | SIGNED BINARY ADD WITH CARRY | @+C | --- | --- | 414 |
| 403 | +CL | DOUBLE SIGNED BINARY ADD WITH CARRY | @+CL | --- | --- | 416 |
| 404 | +B | BCD ADD WITHOUT CARRY | @+B | --- | --- | 421 |
| 405 | +BL | DOUBLE BCD ADD WITHOUT CARRY | @+BL | --- | --- | 419 |
| 406 | +BC | BCD ADD WITH CARRY | @+BC | --- | --- | 421 |
| 407 | +BCL | DOUBLE BCD ADD WITH CARRY | @+BCL | --- | --- | 423 |
| 410 | - | SIGNED BINARY SUBTRACT WITHOUT CARRY | @- | --- | --- | 424 |
| 411 | -L | DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY | @-L | --- | --- | 426 |
| 412 | -C | SIGNED BINARY SUBTRACT WITH CARRY | @-C | --- | --- | 430 |
| 413 | -CL | DOUBLE SIGNED BINARY SUBTRACT WITH CARRY | @-CL | --- | --- | 432 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|-----------------------------------|---------------------------------------|--|-------------------------|--------|
| 414 | -B | BCD SUBTRACT WITHOUT CARRY | @-B | --- | --- | 435 |
| 415 | -BL | DOUBLE BCD SUBTRACT WITHOUT CARRY | @-BL | --- | --- | 436 |
| 416 | -BC | BCD SUBTRACT WITH CARRY | @-BC | --- | --- | 440 |
| 417 | -BCL | DOUBLE BCD SUBTRACT WITH CARRY | @-BCL | --- | --- | 441 |
| 420 | * | SIGNED BINARY MULTIPLY | @* | --- | --- | 443 |
| 421 | *L | DOUBLE SIGNED BINARY MULTIPLY | @*L | --- | --- | 445 |
| 422 | *U | UNSIGNED BINARY MULTIPLY | @*U | --- | --- | 447 |
| 423 | *UL | DOUBLE UNSIGNED BINARY MULTIPLY | @*UL | --- | --- | 449 |
| 424 | *B | BCD MULTIPLY | @*B | --- | --- | 450 |
| 425 | *BL | DOUBLE BCD MULTIPLY | @*BL | --- | --- | 452 |
| 430 | / | SIGNED BINARY DIVIDE | @/ | --- | --- | 454 |
| 431 | /L | DOUBLE SIGNED BINARY DIVIDE | @/L | --- | --- | 456 |
| 432 | /U | UNSIGNED BINARY DIVIDE | @/U | --- | --- | 458 |
| 433 | /UL | DOUBLE UNSIGNED BINARY DIVIDE | @/UL | --- | --- | 460 |
| 434 | /B | BCD DIVIDE | @/B | --- | --- | 462 |
| 435 | /BL | DOUBLE BCD DIVIDE | @/BL | --- | --- | 464 |
| 448 | FSTR | FLOATING POINT TO ASCII | @FSTR | --- | --- | 604 |
| 449 | FVAL | ASCII TO FLOATING POINT | @FVAL | --- | --- | 609 |
| 450 | FIX | FLOATING TO 16-BIT | @FIX | --- | --- | 563 |
| 451 | FIXL | FLOATING TO 32-BIT | @FIXL | --- | --- | 565 |
| 452 | FLT | 16-BIT TO FLOATING | @FLT | --- | --- | 566 |
| 453 | FLTL | 32-BIT TO FLOATING | @FLTL | --- | --- | 568 |
| 454 | +F | FLOATING-POINT ADD | @+F | --- | --- | 570 |
| 455 | -F | FLOATING-POINT SUBTRACT | @-F | --- | --- | 572 |
| 456 | *F | FLOATING-POINT MULTIPLY | @*F | --- | --- | 574 |
| 457 | /F | FLOATING POINT DIVIDE | @/F | --- | --- | 576 |
| 458 | RAD | DEGREES TO RADIANS | @RAD | --- | --- | 597 |
| 459 | DEG | RADIANS TO DEGREES | @DEG | --- | --- | 579 |
| 460 | SIN | SINE | @SIN | --- | --- | 581 |
| 461 | COS | COSINE | @COS | --- | --- | 583 |
| 462 | TAN | TANGENT | @TAN | --- | --- | 585 |
| 463 | ASIN | ARC SINE | @ASIN | --- | --- | 587 |
| 464 | ACOS | ARC COSINE | @ACOS | --- | --- | 589 |
| 465 | ATAN | ARC TANGENT | @ATAN | --- | --- | 591 |
| 466 | SQRT | SQUARE ROOT | @SQRT | --- | --- | 593 |
| 467 | EXP | EXPONENT | @EXP | --- | --- | 595 |
| 468 | LOG | LOGARITHM | @LOG | --- | --- | 597 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|---|---------------------------------------|--|-------------------------|--------|
| 470 | BINS | SIGNED BCD-TO-BINARY | @BINS | --- | --- | 499 |
| 471 | BCDS | SIGNED BINARY-TO-BCD | @BCDS | --- | --- | 505 |
| 472 | BISL | DOUBLE SIGNED BCD-TO-BINARY | @BISL | --- | --- | 502 |
| 473 | BDSL | DOUBLE SIGNED BINARY-TO-BCD | @BDSL | --- | --- | 507 |
| 474 | GRY | GRAY CODE CONVERSION | @GRY | --- | --- | 511 |
| 486 | SCL2 | SCALING 2 | @SCL2 | --- | --- | 762 |
| 487 | SCL3 | SCALING 3 | @SCL3 | --- | --- | 766 |
| 490 | CMND | DELIVER COMMAND | @CMND | --- | --- | 1003 |
| 498 | MOVL | DOUBLE MOVE | @MOVL | --- | --- | 318 |
| 499 | MVNL | DOUBLE MOVE NOT | @MVNL | --- | --- | 320 |
| 502 | BCMP2 | EXPANDED BLOCK COMPARE | @BCMP2 | --- | --- | 306 |
| 510 | CJP | CONDITIONAL JUMP | --- | --- | --- | 223 |
| 511 | CJPN | CONDITIONAL JUMP | --- | --- | --- | 223 |
| 512 | FOR | FOR-NEXT LOOPS | --- | --- | --- | 229 |
| 513 | NEXT | FOR-NEXT LOOPS | --- | --- | --- | 229 |
| 514 | BREAK | BREAK LOOP | --- | --- | --- | 232 |
| 515 | JMP0 | MULTIPLE JUMP | --- | --- | --- | 227 |
| 516 | JME0 | MULTIPLE JUMP END | --- | --- | --- | 227 |
| 517 | MILH | MULTI-INTERLOCK DIFFERENTIATION HOLD | --- | --- | --- | 205 |
| 518 | MILR | MULTI-INTERLOCK DIFFERENTIATION RELEASE | --- | --- | --- | 205 |
| 519 | MILC | MULTI-INTERLOCK CLEAR | --- | --- | --- | 205 |
| 520 | NOT | NOT | --- | --- | --- | 172 |
| 521 | UP | CONDITION ON | --- | --- | --- | 173 |
| 522 | DOWN | CONDITION OFF | --- | --- | --- | 173 |
| 530 | SETA | MULTIPLE BIT SET | @SETA | --- | --- | 189 |
| 531 | RSTA | MULTIPLE BIT RESET | @RSTA | --- | --- | 189 |
| 532 | SETB | SINGLE BIT SET | @SETB | --- | ISETB | 192 |
| 533 | RSTB | SINGLE BIT RESET | @RSTB | --- | IRSTB | 192 |
| 534 | OUTB | SINGLE BIT OUTPUT | @OUTB | --- | IOUTB | 195 |
| 540 | TMHH | ONE-MS TIMER | --- | --- | --- | 244 |
| 542 | TIML | LONG TIMER | --- | --- | --- | 251 |
| 543 | MTIM | MULTI-OUTPUT TIMER | --- | --- | --- | 254 |
| 545 | CNR | RESET TIMER/COUNTER | @CNR | --- | --- | 267 |
| 546 | CNTX | COUNTER | --- | --- | --- | 260 |
| 547 | CNRX | RESET TIMER/COUNTER | --- | --- | --- | 267 |
| 548 | CNTRX | REVERSIBLE COUNTER | --- | --- | --- | 263 |
| 550 | TIMX | TIMER | --- | --- | --- | 235 |
| 551 | TIMHX | HIGH-SPEED TIMER | --- | --- | --- | 240 |
| 552 | TMHHX | ONE-MS TIMER | --- | --- | --- | 244 |
| 553 | TIMLX | LONG TIMER | --- | --- | --- | 251 |
| 554 | MTIMX | MULTI-OUTPUT TIMER | --- | --- | --- | 254 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|---------------------------------------|---------------------------------------|--|-------------------------|--------|
| 555 | TTIMX | ACCUMULATIVE TIMER | --- | --- | --- | 247 |
| 560 | MOVR | MOVE TO REGISTER | @MOVR | --- | --- | 340 |
| 561 | MOVRW | MOVE TIMER/ COUNTER PV TO REGISTER | @MOVRW | --- | --- | 342 |
| 562 | XCGL | DOUBLE DATA EXCHANGE | @XCGL | --- | --- | 334 |
| 565 | XFERC | BLOCK TRANSFER | @XFERC | --- | --- | 1201 |
| 566 | DISTC | SINGLE WORD DISTRIBUTE | @DISTC | --- | --- | 1203 |
| 567 | COLLC | DATA COLLECT | @COLLC | --- | --- | 1206 |
| 568 | MOVBC | MOVE BIT | @MOVBC | --- | --- | 1211 |
| 570 | ASLL | DOUBLE SHIFT LEFT | @ASLL | --- | --- | 355 |
| 571 | ASRL | DOUBLE SHIFT RIGHT | @ASRL | --- | --- | 358 |
| 572 | ROLL | DOUBLE ROTATE LEFT | @ROLL | --- | --- | 362 |
| 573 | RORL | DOUBLE ROTATE RIGHT | @RORL | --- | --- | 365 |
| 574 | RLNC | ROTATE LEFT WITHOUT CARRY | @RLNC | --- | --- | 367 |
| 575 | RRNC | ROTATE RIGHT WITHOUT CARRY | @RRNC | --- | --- | 371 |
| 576 | RLNL | DOUBLE ROTATE LEFT WITHOUT CARRY | @RLNL | --- | --- | 369 |
| 577 | RRNL | DOUBLE ROTATE RIGHT WITHOUT CARRY | @RRNL | --- | --- | 372 |
| 578 | NSFL | SHIFT N-BIT DATA LEFT | @NSFL | --- | --- | 377 |
| 579 | NSFR | SHIFT N-BIT DATA RIGHT | @NSFR | --- | --- | 379 |
| 580 | NASL | SHIFT N-BITS LEFT | @NASL | --- | --- | 381 |
| 581 | NASR | SHIFT N-BITS RIGHT | @NASR | --- | --- | 387 |
| 582 | NSLL | DOUBLE SHIFT N-BITS LEFT | @NSLL | --- | --- | 384 |
| 583 | NSRL | DOUBLE SHIFT N-BITS RIGHT | @NSRL | --- | --- | 389 |
| 590 | ++ | INCREMENT BINARY | @++ | --- | --- | 393 |
| 591 | ++L | DOUBLE INCREMENT BINARY | @++L | --- | --- | 395 |
| 592 | -- | DECREMENT BINARY | @-- | --- | --- | 397 |
| 593 | --L | DOUBLE DECREMENT BINARY | @--L | --- | --- | 399 |
| 594 | ++B | INCREMENT BCD | @++B | --- | --- | 401 |
| 595 | ++BL | DOUBLE INCREMENT BCD | @++BL | --- | --- | 403 |
| 596 | --B | DECREMENT BCD | @--B | --- | --- | 405 |
| 597 | --BL | DOUBLE DECREMENT BCD | @--BL | --- | --- | 407 |
| 600 | SIGN | 16-BIT TO 32-BIT SIGNED BINARY | @SIGN | --- | --- | 476 |
| 610 | ANDL | DOUBLE LOGICAL AND | @ANDL | --- | --- | 519 |
| 611 | ORWL | DOUBLE LOGICAL OR | @ORWL | --- | --- | 522 |
| 612 | XORL | DOUBLE EXCLUSIVE OR | @XORL | --- | --- | 526 |
| 613 | XNRL | DOUBLE EXCLUSIVE NOR | @XNRL | --- | --- | 529 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|--------------------------------|---------------------------------------|--|-------------------------|--------|
| 614 | COML | DOUBLE COMPLEMENT | @COML | --- | --- | 533 |
| 620 | ROTB | BINARY ROOT | @ROTB | --- | --- | 534 |
| 621 | BCNTC | BIT COUNTER | @BCNTC | --- | --- | 1212 |
| 630 | SSET | SET STACK | @SSET | --- | --- | 666 |
| 631 | DIM | DIMENSION RECORD TABLE | @DIM | --- | --- | 678 |
| 632 | PUSH | PUSH ONTO STACK | @PUSH | --- | --- | 669 |
| 633 | FIFO | FIRST IN FIRST OUT | @FIFO | --- | --- | 672 |
| 634 | LIFO | LAST IN FIRST OUT | @LIFO | --- | --- | 675 |
| 635 | SETR | SET RECORD LOCATION | @SETR | --- | --- | 681 |
| 636 | GETR | GET RECORD NUMBER | @GETR | --- | --- | 683 |
| 637 | SWAP | SWAP BYTES | @SWAP | --- | --- | 687 |
| 638 | SNUM | STACK SIZE READ | @SNUM | --- | --- | 704 |
| 639 | SREAD | STACK DATA READ | @SREAD | --- | --- | 707 |
| 640 | SWRIT | STACK DATA WRITE | @SWRIT | --- | --- | 710 |
| 641 | SINS | STACK DATA INSERT | @SINS | --- | --- | 713 |
| 642 | SDEL | STACK DATA DELETE | @SDEL | --- | --- | 716 |
| 650 | LEN\$ | STRING LENGTH | @LEN\$ | --- | --- | 1173 |
| 652 | LEFT\$ | GET STRING LEFT | @LEFT\$ | --- | --- | 1164 |
| 653 | RGHT\$ | GET STRING RIGHT | @RGHT\$ | --- | --- | 1166 |
| 654 | MID\$ | GET STRING MIDDLE | @MID\$ | --- | --- | 1168 |
| 656 | +\$ | CONCATENATE STRING | @+\$ | --- | --- | 1161 |
| 657 | INS\$ | INS\$ | @INS\$ | --- | --- | 1184 |
| 658 | DEL\$ | DELETE STRING | @DEL\$ | --- | --- | 1178 |
| 660 | FIND\$ | FIND IN STRING | @FIND\$ | --- | --- | 1171 |
| 661 | RPLC\$ | REPLACE IN STRING | @RPLC\$ | --- | --- | 1175 |
| 664 | MOV\$ | MOV STRING | @MOV\$ | --- | --- | 1159 |
| 665 | XCHG\$ | EXCHANGE STRING | @XCHG\$ | --- | --- | 1180 |
| 666 | CLR\$ | CLEAR STRING | @CLR\$ | --- | --- | 1182 |
| 670 | AND =\$ | AND STRING EQUALS | --- | --- | --- | 1187 |
| 670 | LD =\$ | LOAD STRING EQUALS | --- | --- | --- | 1187 |
| 670 | OR =\$ | OR STRING EQUALS | --- | --- | --- | 1187 |
| 671 | AND <>\$ | AND STRING NOT EQUAL | --- | --- | --- | 1187 |
| 671 | LD <>\$ | LOAD STRING NOT EQUAL | --- | --- | --- | 1187 |
| 671 | OR <>\$ | OR STRING NOT EQUAL | --- | --- | --- | 1187 |
| 672 | AND <\$ | AND STRING LESS THAN | --- | --- | --- | 1187 |
| 672 | LD <\$ | LOAD STRING LESS THAN | --- | --- | --- | 1187 |
| 672 | OR <\$ | OR STRING LESS THAN | --- | --- | --- | 1187 |
| 673 | AND <=\$ | AND STRING LESS THAN OR EQUALS | --- | --- | --- | 1187 |
| 673 | LD <=\$ | LOAD STRING LESS THAN OR EQUAL | --- | --- | --- | 1187 |
| 673 | OR <=\$ | OR STRING LESS THAN OR EQUALS | --- | --- | --- | 1187 |
| 674 | AND >\$ | AND STRING GREATER THAN | --- | --- | --- | 1187 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|------------------------------------|---------------------------------------|--|-------------------------|--------|
| 674 | LD >\$ | LOAD STRING GREATER THAN | --- | --- | --- | 1187 |
| 674 | OR >\$ | OR STRING GREATER THAN | --- | --- | --- | 1187 |
| 675 | AND >=\$ | AND STRING GREATER THAN OR EQUALS | --- | --- | --- | 1187 |
| 675 | LD >=\$ | LOAD STRING GREATER THAN OR EQUALS | --- | --- | --- | 1187 |
| 675 | OR >=\$ | OR STRING GREATER THAN OR EQUALS | --- | --- | --- | 1187 |
| 680 | LMT | LIMIT CONTROL | @LMT | --- | --- | 741 |
| 681 | BAND | DEAD BAND CONTROL | @BAND | --- | --- | 743 |
| 682 | ZONE | DEAD ZONE CONTROL | @ZONE | --- | --- | 746 |
| 685 | TPO | TIME-PROPORTIONAL OUTPUT | --- | --- | --- | 749 |
| 690 | MSKS | SET INTERRUPT MASK | @MSKS | --- | --- | 798 |
| 691 | CLI | CLEAR INTERRUPT | @CLI | --- | --- | 810 |
| 692 | MSKR | READ INTERRUPT MASK | @MSKR | --- | --- | 805 |
| 693 | DI | DISABLE INTERRUPTS | @DI | --- | --- | 814 |
| 694 | EI | ENABLE INTERRUPTS | --- | --- | --- | 816 |
| 700 | FREAD | READ DATA FILE | @FREAD | --- | --- | 1045 |
| 701 | FWRIT | WRITE DATA FILE | @FWRIT | --- | --- | 1052 |
| 720 | EXPLT | EXPLICIT MESSAGE SEND | @EXPLT | --- | --- | 1013 |
| 721 | EGATR | EXPLICIT GET ATTRIBUTE | @EGATR | --- | --- | 1021 |
| 722 | ESATR | EXPLICIT SET ATTRIBUTE | @ESATR | --- | --- | 1028 |
| 723 | ECHRD | EXPLICIT WORD READ | @ECHRD | --- | --- | 1034 |
| 724 | ECHWR | EXPLICIT WORD CLEAR | @ECHWR | --- | --- | 1038 |
| 730 | CADD | CALENDAR ADD | @CADD | --- | --- | 1061 |
| 731 | CSUB | CALENDAR SUBTRACT | @CSUB | --- | --- | 1065 |
| 735 | DATE | CLOCK ADJUSTMENT | @DATE | --- | --- | 1073 |
| 750 | GSBS | GLOBAL SUBROUTINE CALL | @GSBS | --- | --- | 786 |
| 751 | GSBN | GLOBAL SUBROUTINE ENTRY | --- | --- | --- | 794 |
| 752 | GRET | GLOBAL SUBROUTINE RETURN | --- | --- | --- | 797 |
| 801 | BEND | BLOCK PROGRAM END | --- | --- | --- | 1128 |
| 802 | IF | CONDITIONAL BRANCHING BLOCK | --- | --- | --- | 1133 |
| 802 | IF | CONDITIONAL BRANCHING BLOCK | --- | --- | --- | 1133 |
| 802 | IF NOT | CONDITIONAL BRANCHING BLOCK NOT | --- | --- | --- | 1133 |
| 803 | ELSE | ELSE | --- | --- | --- | 1133 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|----------------------------------|---------------------------------------|--|-------------------------|--------|
| 804 | IEND | IF END | --- | --- | --- | 1133 |
| 805 | WAIT | ONE CYCLE AND WAIT | --- | --- | --- | 1140 |
| 805 | WAIT | ONE CYCLE AND WAIT | --- | --- | --- | 1140 |
| 805 | WAIT NOT | ONE CYCLE AND WAIT NOT | --- | --- | --- | 1140 |
| 806 | EXIT | CONDITIONAL BLOCK EXIT | --- | --- | --- | 1137 |
| 806 | EXIT | CONDITIONAL BLOCK EXIT | --- | --- | --- | 1137 |
| 806 | EXIT NOT | CONDITIONAL BLOCK EXIT NOT | --- | --- | --- | 1137 |
| 809 | LOOP | LOOP | --- | --- | --- | 1153 |
| 810 | LEND | LOOP END | --- | --- | --- | 1153 |
| 810 | LEND | LOOP END | --- | --- | --- | 1153 |
| 810 | LEND NOT | LOOP END NOT | --- | --- | --- | 1153 |
| 811 | BPPS | BLOCK PROGRAM PAUSE | --- | --- | --- | 1131 |
| 812 | BPRS | BLOCK PROGRAM RESTART | --- | --- | --- | 1131 |
| 813 | TIMW | TIMER WAIT | --- | --- | --- | 1144 |
| 814 | CNTW | COUNTER WAIT | --- | --- | --- | 1147 |
| 815 | TMHW | HIGH-SPEED TIMER WAIT | --- | --- | --- | 1150 |
| 816 | TIMWX | TIMER WAIT | --- | --- | --- | 1144 |
| 817 | TMHWX | HIGH-SPEED TIMER WAIT | --- | --- | --- | 1150 |
| 818 | CNTWX | COUNTER WAIT | --- | --- | --- | 1147 |
| 820 | TKON | TASK ON | @TKON | --- | --- | 1192 |
| 821 | TKOF | TASK OFF | @TKOF | --- | --- | 1196 |
| 840 | PWR | EXPONENTIAL POWER | @PWR | --- | --- | 599 |
| 841 | FIXD | DOUBLE FLOATING TO 16-BIT BINARY | @FIXD | --- | --- | 620 |
| 842 | FIXLD | DOUBLE FLOATING TO 32-BIT BINARY | @FIXLD | --- | --- | 621 |
| 843 | DBL | 16-BIT BINARY TO DOUBLE FLOATING | @DBL | --- | --- | 623 |
| 844 | DBLL | 32-BIT BINARY TO DOUBLE FLOATING | @DBLL | --- | --- | 624 |
| 845 | +D | DOUBLE FLOATING-POINT ADD | @+D | --- | --- | 626 |
| 846 | -D | DOUBLE FLOATING-POINT SUBTRACT | @-D | --- | --- | 628 |
| 847 | *D | DOUBLE FLOATING-POINT MULTIPLY | @*D | --- | --- | 630 |
| 848 | /D | DOUBLE FLOATING-POINT DIVIDE | @/D | --- | --- | 632 |
| 849 | RADD | DOUBLE DEGREES TO RADIANS | @RADD | --- | --- | 634 |
| 850 | DEGD | DOUBLE RADIANS TO DEGREES | @RADD | --- | --- | 636 |
| 851 | SIND | DOUBLE SINE | @SIND | --- | --- | 637 |
| 852 | COSD | DOUBLE COSINE | @COSD | --- | --- | 639 |
| 853 | TAND | DOUBLE TANGENT | @TAND | --- | --- | 641 |
| 854 | ASIND | DOUBLE ARC SINE | @ASIND | --- | --- | 643 |
| 855 | ACOSD | DOUBLE ARC COSINE | @ACOSD | --- | --- | 645 |

| Codice funzione | Codice mnemonico | Istruzione | Differenziazione sul fronte di salita | Differenziazione sul fronte di discesa | Aggiornamento immediato | Pagina |
|-----------------|------------------|---------------------------------|---------------------------------------|--|-------------------------|--------|
| 856 | ATAND | DOUBLE ARC TANGENT | @ATAND | --- | --- | 647 |
| 857 | SQRTD | DOUBLE SQUARE ROOT | @SQRTD | --- | --- | 649 |
| 858 | EXPD | DOUBLE EXPONENT | @EXPD | --- | --- | 651 |
| 859 | LOGD | DOUBLE LOGARITHM | @LOGD | --- | --- | 653 |
| 860 | PWRD | DOUBLE EXPONENTIAL POWER | @PWRD | --- | --- | 655 |
| 880 | INI | MODE CONTROL | @INI | --- | --- | 823 |
| 881 | PRV | HIGH-SPEED COUNTER PV READ | @PRV | --- | --- | 827 |
| 882 | CTBL | COMPARISON TABLE LOAD | @CTBL | --- | --- | 837 |
| 883 | PRV2 | COUNTER FREQUENCY CONVERT | @PRV2 | --- | --- | 833 |
| 885 | SPED | SPEED OUTPUT | @SPED | --- | --- | 841 |
| 886 | PULS | SET PULSES | @PULS | --- | --- | 846 |
| 887 | PLS2 | PULSE OUTPUT | @PLS2 | --- | --- | 849 |
| 888 | ACC | ACCELERATION CONTROL | @ACC | --- | --- | 855 |
| 889 | ORG | ORIGIN SEARCH | @ORG | --- | --- | 862 |
| 891 | PWN | PULSE WITH VARIABLE DUTY FACTOR | @PWN | --- | --- | 865 |

CAPITOLO 3

Istruzioni

In questo capitolo sono illustrate tutte le istruzioni utili alla programmazione dei PLC della serie CS/CJ. Tali istruzioni sono raggruppate per funzionalità, in base alla classificazione riportata nel *Capitolo 2 Sommario delle istruzioni*.

| | | |
|--------|---|-----|
| 3-1 | Descrizioni delle istruzioni: notazione e layout | 148 |
| 3-2 | Aggiornamenti delle istruzioni e nuove istruzioni | 151 |
| 3-2-1 | Aggiornamenti per CPU della serie CS versione 1 | 151 |
| 3-2-2 | Aggiornamenti per CPU CS1/H e CJ1/H | 151 |
| 3-3 | Istruzioni di ingresso sequenza | 153 |
| 3-3-1 | LOAD: LD | 153 |
| 3-3-2 | LOAD NOT: LD NOT | 155 |
| 3-3-3 | AND: AND | 157 |
| 3-3-4 | AND NOT: AND NOT | 159 |
| 3-3-5 | OR: OR | 161 |
| 3-3-6 | OR NOT: OR NOT | 163 |
| 3-3-7 | AND LOAD: AND LD | 164 |
| 3-3-8 | OR LOAD: OR LD | 166 |
| 3-3-9 | Istruzioni di aggiornamento differenziato e immediato | 169 |
| 3-3-10 | Tempistica operativa delle istruzioni di I/O | 170 |
| 3-3-11 | Bit TR | 170 |
| 3-3-12 | NOT: NOT(520) | 172 |
| 3-3-13 | CONDITION ON/OFF UP(521) e DOWN(522) | 173 |
| 3-3-14 | BIT TEST: TST(350) e TSTN(351) | 174 |
| 3-4 | Istruzioni di uscita sequenza | 177 |
| 3-4-1 | OUTPUT: OUT | 177 |
| 3-4-2 | OUTPUT NOT: OUT NOT | 178 |
| 3-4-3 | KEEP: KEEP(011) | 180 |
| 3-4-4 | DIFFERENTIATE UP/DOWN: DIFU(013) e DIFD(014) | 184 |
| 3-4-5 | SET e RESET SET e RSET | 187 |
| 3-4-6 | MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531) | 189 |
| 3-4-7 | SINGLE BIT SET/RESET: SETB(532)/RSTB(533) | 192 |
| 3-4-8 | SINGLE BIT OUTPUT: OUTB(534) | 195 |
| 3-5 | Istruzioni di controllo sequenza | 197 |
| 3-5-1 | END: END(001) | 197 |
| 3-5-2 | NO OPERATION: NOP(000) | 198 |
| 3-5-3 | Panoramica sulle istruzioni di interblocco | 199 |
| 3-5-4 | INTERLOCK e INTERLOCK CLEAR: IL(002) e ILC(003) | 201 |
| 3-5-5 | MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE e MULTI-INTERLOCK CLEAR: MILH(517), MILR(518) e MILC(519) | 205 |
| 3-5-6 | JUMP e JUMP END: JMP(004) e JME(005) | 219 |
| 3-5-7 | CONDITIONAL JUMP: CJP(510)/CJPN(511) | 223 |
| 3-5-8 | MULTIPLE JUMP e JUMP END: JMP0(515) e JME0(516) | 227 |
| 3-5-9 | FOR-NEXT LOOPS: FOR(512)/NEXT(513) | 229 |
| 3-5-10 | BREAK LOOP: BREAK(514) | 232 |
| 3-6 | Istruzioni di temporizzatore e contatore | 233 |
| 3-6-1 | TIMER: TIM/TIMX(550) | 235 |
| 3-6-2 | HIGH-SPEED TIMER: TIMH(015)/TIMHX(551) | 240 |
| 3-6-3 | ONE-MS TIMER: TMHH(540)/TMHHX(552) | 244 |
| 3-6-4 | ACCUMULATIVE TIMER: TTIM(087)/TTIMX(555) | 247 |
| 3-6-5 | LONG TIMER: TIML(542)/TIMLX(553) | 251 |
| 3-6-6 | MULTI-OUTPUT TIMER: MTIM(543)/MTIMX(554) | 254 |

| | | |
|--------|--|-----|
| 3-6-7 | COUNTER: CNT/CNTX(546)..... | 260 |
| 3-6-8 | REVERSIBLE COUNTER: CNTR(012)/CNTRX(548)..... | 263 |
| 3-6-9 | RESET TIMER/COUNTER: CNR(545)/CNRX(547)..... | 267 |
| 3-6-10 | Applicazioni di esempio per temporizzatori e contatori..... | 269 |
| 3-6-11 | Indirizzamento indiretto dei numeri di temporizzatore/contatore..... | 272 |
| 3-7 | Istruzioni di confronto..... | 275 |
| 3-7-1 | Istruzioni di confronto di ingresso (da 300 a 328)..... | 275 |
| 3-7-2 | Istruzioni di confronto di date e ore (da 341 a 346)..... | 281 |
| 3-7-3 | COMPARE: CMP(020)..... | 287 |
| 3-7-4 | DOUBLE COMPARE: CMPL(060)..... | 290 |
| 3-7-5 | SIGNED BINARY COMPARE: CPS(114)..... | 293 |
| 3-7-6 | DOUBLE SIGNED BINARY COMPARE: CPSL(115)..... | 296 |
| 3-7-7 | MULTIPLE COMPARE: MCMP(019)..... | 299 |
| 3-7-8 | TABLE COMPARE: TCMP(085)..... | 301 |
| 3-7-9 | BLOCK COMPARE: BCMP(068)..... | 304 |
| 3-7-10 | EXPANDED BLOCK COMPARE: BCMP2(502)..... | 306 |
| 3-7-11 | AREA RANGE COMPARE: ZCP(088)..... | 310 |
| 3-7-12 | DOUBLE AREA RANGE COMPARE: ZCPL(116)..... | 313 |
| 3-8 | Istruzioni di spostamento dati..... | 315 |
| 3-8-1 | MOVE: MOV(021)..... | 315 |
| 3-8-2 | MOVE NOT: MVN(022)..... | 317 |
| 3-8-3 | DOUBLE MOVE: MOVL(498)..... | 318 |
| 3-8-4 | DOUBLE MOVE NOT: MVNL(499)..... | 320 |
| 3-8-5 | MOVE BIT: MOVB(082)..... | 321 |
| 3-8-6 | MOVE DIGIT: MOVD(083)..... | 323 |
| 3-8-7 | MULTIPLE BIT TRANSFER: XFRB(062)..... | 326 |
| 3-8-8 | BLOCK TRANSFER: XFER(070)..... | 328 |
| 3-8-9 | BLOCK SET: BSET(071)..... | 331 |
| 3-8-10 | DATA EXCHANGE: XCHG(073)..... | 333 |
| 3-8-11 | DOUBLE DATA EXCHANGE: XCGL(562)..... | 334 |
| 3-8-12 | SINGLE WORD DISTRIBUTE: DIST(080)..... | 336 |
| 3-8-13 | DATA COLLECT: COLL(081)..... | 338 |
| 3-8-14 | MOVE TO REGISTER: MOVR(560)..... | 340 |
| 3-8-15 | MOVE TIMER/COUNTER PV TO REGISTER: MOVRW(561)..... | 342 |
| 3-9 | Istruzioni di scorrimento dei dati..... | 344 |
| 3-9-1 | SHIFT REGISTER: SFT(010)..... | 345 |
| 3-9-2 | REVERSIBLE SHIFT REGISTER: SFTR(084)..... | 346 |
| 3-9-3 | ASYNCHRONOUS SHIFT REGISTER: ASFT(017)..... | 349 |
| 3-9-4 | WORD SHIFT: WSFT(016)..... | 352 |
| 3-9-5 | ARITHMETIC SHIFT LEFT: ASL(025)..... | 354 |
| 3-9-6 | DOUBLE SHIFT LEFT: ASLL(570)..... | 355 |
| 3-9-7 | ARITHMETIC SHIFT RIGHT: ASR(026)..... | 357 |
| 3-9-8 | DOUBLE SHIFT RIGHT: ASRL(571)..... | 358 |
| 3-9-9 | ROTATE LEFT: ROL(027)..... | 360 |
| 3-9-10 | DOUBLE ROTATE LEFT: ROLL(572)..... | 362 |
| 3-9-11 | ROTATE RIGHT: ROR(028)..... | 364 |
| 3-9-12 | DOUBLE ROTATE RIGHT: RORL(573)..... | 365 |
| 3-9-13 | ROTATE LEFT WITHOUT CARRY: RLNC(574)..... | 367 |
| 3-9-14 | DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576)..... | 369 |
| 3-9-15 | ROTATE RIGHT WITHOUT CARRY: RRNC(575)..... | 371 |
| 3-9-16 | DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577)..... | 372 |
| 3-9-17 | ONE DIGIT SHIFT LEFT: SLD(074)..... | 374 |
| 3-9-18 | ONE DIGIT SHIFT RIGHT: SRD(075)..... | 376 |
| 3-9-19 | SHIFT N-BIT DATA LEFT: NSFL(578)..... | 377 |
| 3-9-20 | SHIFT N-BIT DATA RIGHT: NSFR(579)..... | 379 |
| 3-9-21 | SHIFT N-BITS LEFT: NASL(580)..... | 381 |
| 3-9-22 | DOUBLE SHIFT N-BITS LEFT: NSLL(582)..... | 384 |

| | | |
|---------|--|-----|
| 3-9-23 | SHIFT N-BITS RIGHT: NASR(581) | 387 |
| 3-9-24 | DOUBLE SHIFT N-BITS RIGHT: NSRL(583) | 389 |
| 3-10 | Istruzioni di incremento e decremento | 393 |
| 3-10-1 | INCREMENT BINARY: ++(590) | 393 |
| 3-10-2 | DOUBLE INCREMENT BINARY: ++L(591) | 395 |
| 3-10-3 | DECREMENT BINARY: --(592) | 397 |
| 3-10-4 | DOUBLE DECREMENT BINARY: --L(593) | 399 |
| 3-10-5 | INCREMENT BCD: ++B(594) | 401 |
| 3-10-6 | DOUBLE INCREMENT BCD: ++BL(595) | 403 |
| 3-10-7 | DECREMENT BCD: --B(596) | 405 |
| 3-10-8 | DOUBLE DECREMENT BCD: --BL(597) | 407 |
| 3-11 | Istruzioni matematiche con simboli | 409 |
| 3-11-1 | SIGNED BINARY ADD WITHOUT CARRY: +(400) | 410 |
| 3-11-2 | DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401) | 412 |
| 3-11-3 | SIGNED BINARY ADD WITH CARRY: +C(402) | 414 |
| 3-11-4 | DOUBLE SIGNED BINARY ADD WITH CARRY: +CL(403) | 416 |
| 3-11-5 | BCD ADD WITHOUT CARRY: +B(404) | 418 |
| 3-11-6 | DOUBLE BCD ADD WITHOUT CARRY: +BL(405) | 419 |
| 3-11-7 | BCD ADD WITH CARRY: +BC(406) | 421 |
| 3-11-8 | DOUBLE BCD ADD WITH CARRY: +BCL(407) | 423 |
| 3-11-9 | SIGNED BINARY SUBTRACT WITHOUT CARRY: -(410) | 424 |
| 3-11-10 | DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: -L(411) | 426 |
| 3-11-11 | SIGNED BINARY SUBTRACT WITH CARRY: -C(412) | 430 |
| 3-11-12 | DOUBLE SIGNED BINARY SUBTRACT WITH CARRY: -CL(413) | 432 |
| 3-11-13 | BCD SUBTRACT WITHOUT CARRY: -B(414) | 435 |
| 3-11-14 | DOUBLE BCD SUBTRACT WITHOUT CARRY: -BL(415) | 436 |
| 3-11-15 | BCD SUBTRACT WITH CARRY: -BC(416) | 440 |
| 3-11-16 | DOUBLE BCD SUBTRACT WITH CARRY: -BCL(417) | 441 |
| 3-11-17 | SIGNED BINARY MULTIPLY: *(420) | 443 |
| 3-11-18 | DOUBLE SIGNED BINARY MULTIPLY: *L(421) | 445 |
| 3-11-19 | UNSIGNED BINARY MULTIPLY: *U(422) | 447 |
| 3-11-20 | DOUBLE UNSIGNED BINARY MULTIPLY: *UL(423) | 449 |
| 3-11-21 | BCD MULTIPLY: *B(424) | 450 |
| 3-11-22 | DOUBLE BCD MULTIPLY: *BL(425) | 452 |
| 3-11-23 | SIGNED BINARY DIVIDE: /(430) | 454 |
| 3-11-24 | DOUBLE SIGNED BINARY DIVIDE: /L(431) | 456 |
| 3-11-25 | UNSIGNED BINARY DIVIDE: /U(432) | 458 |
| 3-11-26 | DOUBLE UNSIGNED BINARY DIVIDE: /UL(433) | 460 |
| 3-11-27 | BCD DIVIDE: /B(434) | 462 |
| 3-11-28 | DOUBLE BCD DIVIDE: /BL(435) | 464 |
| 3-12 | Istruzioni di conversione | 465 |
| 3-12-1 | BCD-TO-BINARY: BIN(023) | 466 |
| 3-12-2 | DOUBLE BCD-TO-DOUBLE BINARY: BINL(058) | 467 |
| 3-12-3 | BINARY-TO-BCD: BCD(024) | 469 |
| 3-12-4 | DOUBLE BINARY-TO-DOUBLE BCD: BCDL(059) | 470 |
| 3-12-5 | 2'S COMPLEMENT: NEG(160) | 472 |
| 3-12-6 | DOUBLE 2'S COMPLEMENT: NEGL(161) | 474 |
| 3-12-7 | 16-BIT TO 32-BIT SIGNED BINARY: SIGN(600) | 476 |
| 3-12-8 | DATA DECODER: MLPX(076) | 477 |
| 3-12-9 | DATA ENCODER: DMPX(077) | 482 |
| 3-12-10 | ASCII CONVERT: ASC(086) | 486 |
| 3-12-11 | ASCII TO HEX: HEX(162) | 490 |
| 3-12-12 | COLUMN TO LINE: LINE(063) | 494 |
| 3-12-13 | LINE TO COLUMN: COLM(064) | 496 |
| 3-12-14 | SIGNED BCD-TO-BINARY: BINS(470) | 499 |
| 3-12-15 | DOUBLE SIGNED BCD-TO-BINARY: BISL(472) | 502 |
| 3-12-16 | SIGNED BINARY-TO-BCD: BCDS(471) | 505 |

| | | |
|---------|---|-----|
| 3-12-17 | DOUBLE SIGNED BINARY-TO-BCD: BDSL(473) | 507 |
| 3-12-18 | GRAY CODE CONVERT: GRY(474) | 511 |
| 3-13 | Istruzioni logiche | 517 |
| 3-13-1 | LOGICAL AND: ANDW(034) | 517 |
| 3-13-2 | DOUBLE LOGICAL AND: ANDL(610) | 519 |
| 3-13-3 | LOGICAL OR: ORW(035) | 520 |
| 3-13-4 | DOUBLE LOGICAL OR: ORWL(611) | 522 |
| 3-13-5 | EXCLUSIVE OR: XORW(036) | 524 |
| 3-13-6 | DOUBLE EXCLUSIVE OR: XORL(612) | 526 |
| 3-13-7 | EXCLUSIVE NOR: XNRW(037) | 528 |
| 3-13-8 | DOUBLE EXCLUSIVE NOR: XNRL(613) | 529 |
| 3-13-9 | COMPLEMENT: COM(029) | 531 |
| 3-13-10 | DOUBLE COMPLEMENT: COML(614) | 533 |
| 3-14 | Istruzioni matematiche speciali | 534 |
| 3-14-1 | BINARY ROOT: ROTB(620) | 534 |
| 3-14-2 | BCD SQUARE ROOT: ROOT(072) | 536 |
| 3-14-3 | ARITHMETIC PROCESS: APR(069) | 540 |
| 3-14-4 | FLOATING POINT DIVIDE: FDIV(079) | 552 |
| 3-14-5 | BIT COUNTER: BCNT(067) | 556 |
| 3-15 | Istruzioni matematiche a virgola mobile | 558 |
| 3-15-1 | FLOATING TO 16-BIT: FIX(450) | 563 |
| 3-15-2 | FLOATING TO 32-BIT: FIXL(451) | 565 |
| 3-15-3 | 16-BIT TO FLOATING: FLT(452) | 566 |
| 3-15-4 | 32-BIT TO FLOATING: FLTL(453) | 568 |
| 3-15-5 | FLOATING-POINT ADD: +F(454) | 570 |
| 3-15-6 | FLOATING-POINT SUBTRACT: -F(455) | 572 |
| 3-15-7 | FLOATING-POINT MULTIPLY: *F(456) | 574 |
| 3-15-8 | FLOATING-POINT DIVIDE: /F(457) | 576 |
| 3-15-9 | DEGREES TO RADIANS: RAD(458) | 578 |
| 3-15-10 | RADIANS TO DEGREES: DEG(459) | 579 |
| 3-15-11 | SINE: SIN(460) | 581 |
| 3-15-12 | COSINE: COS(461) | 583 |
| 3-15-13 | TANGENT: TAN(462) | 585 |
| 3-15-14 | ARC SINE: ASIN(463) | 587 |
| 3-15-15 | ARC COSINE: ACOS(464) | 589 |
| 3-15-16 | ARC TANGENT: ATAN(465) | 591 |
| 3-15-17 | SQUARE ROOT: SQRT(466) | 593 |
| 3-15-18 | EXPONENT: EXP(467) | 595 |
| 3-15-19 | LOGARITHM: LOG(468) | 597 |
| 3-15-20 | EXPONENTIAL POWER: PWR(840) | 599 |
| 3-15-21 | Istruzioni di confronto a virgola mobile in singola precisione | 600 |
| 3-15-22 | FLOATING-POINT TO ASCII: FSTR(448) | 604 |
| 3-15-23 | ASCII TO FLOATING-POINT: FVAL(449) | 609 |
| 3-16 | Istruzioni a virgola mobile in doppia precisione (solo CPU CS1-H, CJ1-H, CJ1M o CS1D) | 613 |
| 3-16-1 | DOUBLE FLOATING TO 16-BIT: FIXD(841) | 620 |
| 3-16-2 | DOUBLE FLOATING TO 32-BIT: FIXLD(842) | 621 |
| 3-16-3 | 16-BIT TO DOUBLE FLOATING: DBL(843) | 623 |
| 3-16-4 | 32-BIT TO DOUBLE FLOATING: DBLL(844) | 624 |
| 3-16-5 | DOUBLE FLOATING-POINT ADD: +D(845) | 626 |
| 3-16-6 | DOUBLE FLOATING-POINT SUBTRACT: -D(846) | 628 |
| 3-16-7 | DOUBLE FLOATING-POINT MULTIPLY: *D(847) | 630 |
| 3-16-8 | DOUBLE FLOATING-POINT DIVIDE: /D(848) | 632 |
| 3-16-9 | DOUBLE DEGREES TO RADIANS: RADD(849) | 634 |
| 3-16-10 | DOUBLE RADIANS TO DEGREES: DEGD(850) | 636 |
| 3-16-11 | DOUBLE SINE: SIND(851) | 637 |
| 3-16-12 | DOUBLE COSINE COSD(852) | 639 |
| 3-16-13 | DOUBLE TANGENT TAND(853) | 641 |

| | | |
|---------|--|-----|
| 3-16-14 | DOUBLE ARC SINE: ASIND(854) | 643 |
| 3-16-15 | DOUBLE ARC COSINE: ACOSD(855) | 645 |
| 3-16-16 | DOUBLE ARC TANGENT: ATAND(856) | 647 |
| 3-16-17 | DOUBLE SQUARE ROOT: SQRTD(857) | 649 |
| 3-16-18 | DOUBLE EXPONENT: EXPD(858) | 651 |
| 3-16-19 | DOUBLE LOGARITHM: LOGD(859) | 653 |
| 3-16-20 | DOUBLE EXPONENTIAL POWER: PWRD(860) | 655 |
| 3-16-21 | Istruzioni di ingresso a virgola mobile in doppia precisione | 657 |
| 3-17 | Istruzioni di elaborazione dei dati delle tabelle | 660 |
| 3-17-1 | SET STACK: SSET(630) | 666 |
| 3-17-2 | PUSH ONTO STACK: PUSH(632) | 669 |
| 3-17-3 | FIRST IN FIRST OUT: FIFO(633) | 672 |
| 3-17-4 | LAST IN FIRST OUT: LIFO(634) | 675 |
| 3-17-5 | DIMENSION RECORD TABLE: DIM(631) | 678 |
| 3-17-6 | SET RECORD LOCATION: SETR(635) | 681 |
| 3-17-7 | GET RECORD NUMBER: GETR(636) | 683 |
| 3-17-8 | DATA SEARCH: SRCH(181) | 685 |
| 3-17-9 | SWAP BYTES: SWAP(637) | 687 |
| 3-17-10 | FIND MAXIMUM: MAX(182) | 689 |
| 3-17-11 | FIND MINIMUM: MIN(183) | 693 |
| 3-17-12 | SUM: SUM(184) | 697 |
| 3-17-13 | FRAME CHECKSUM: FCS(180) | 700 |
| 3-17-14 | STACK SIZE READ: SNUM(638) | 704 |
| 3-17-15 | STACK DATA READ: SREAD(639) | 707 |
| 3-17-16 | STACK DATA OVERWRITE: SWRIT(640) | 710 |
| 3-17-17 | STACK DATA INSERT: SINS(641) | 713 |
| 3-17-18 | STACK DATA DELETE: SDEL(642) | 716 |
| 3-18 | Istruzioni di controllo dei dati | 720 |
| 3-18-1 | PID CONTROL: PID(190) | 720 |
| 3-18-2 | PID CONTROL WITH AUTOTUNING: PIDAT(191) | 731 |
| 3-18-3 | LIMIT CONTROL: LMT(680) | 741 |
| 3-18-4 | DEAD BAND CONTROL: BAND(681) | 743 |
| 3-18-5 | DEAD ZONE CONTROL: ZONE(682) | 746 |
| 3-18-6 | TIME-PROPORTIONAL OUTPUT: TPO(685) | 749 |
| 3-18-7 | SCALING: SCL(194) | 757 |
| 3-18-8 | SCALING 2: SCL2(486) | 762 |
| 3-18-9 | SCALING 3: SCL3(487) | 766 |
| 3-18-10 | AVERAGE: AVG(195) | 769 |
| 3-19 | Subroutine | 773 |
| 3-19-1 | SUBROUTINE CALL: SBS(091) | 773 |
| 3-19-2 | MACRO: MCRO(099) | 779 |
| 3-19-3 | SUBROUTINE ENTRY: SBN(092) | 783 |
| 3-19-4 | SUBROUTINE RETURN: RET(093) | 786 |
| 3-19-5 | GLOBAL SUBROUTINE CALL: GSBS(750) | 786 |
| 3-19-6 | GLOBAL SUBROUTINE ENTRY GSBN(751) | 794 |
| 3-19-7 | GLOBAL SUBROUTINE RETURN: GRET(752) | 797 |
| 3-20 | Istruzioni di controllo degli interrupt | 798 |
| 3-20-1 | SET INTERRUPT MASK: MSKS(690) | 798 |
| 3-20-2 | READ INTERRUPT MASK: MSKR(692) | 805 |
| 3-20-3 | CLEAR INTERRUPT: CLI(691) | 810 |
| 3-20-4 | DISABLE INTERRUPTS: DI(693) | 814 |
| 3-20-5 | ENABLE INTERRUPTS: EI(694) | 816 |
| 3-20-6 | Riepilogo del controllo degli interrupt | 818 |
| 3-21 | Istruzioni per contatore veloce e uscita a impulsi | 823 |
| 3-21-1 | MODE CONTROL: INI(880) (solo CJ1M-CPU21/22/23) | 823 |
| 3-21-2 | HIGH-SPEED COUNTER PV READ: PRV(881) (solo CJ1M-CPU21/22/23) | 827 |
| 3-21-3 | COUNTER FREQUENCY CONVERT: PRV2(883) | 833 |

| | | |
|---------|--|------|
| 3-21-4 | REGISTER COMPARISON TABLE: CTBL(882) (solo CJ1M-CPU21/22/23) | 837 |
| 3-21-5 | SPEED OUTPUT: SPED(885) (solo CJ1M-CPU21/22/23) | 841 |
| 3-21-6 | SET PULSES: PULS(886) (solo CJ1M-CPU21/22/23) | 846 |
| 3-21-7 | PULSE OUTPUT: PLS2(887) (solo CPU CJ1M 21/22/23) | 849 |
| 3-21-8 | ACCELERATION CONTROL: ACC(888) (solo CPU CJ1M 21/22/23) | 855 |
| 3-21-9 | ORIGIN SEARCH: ORG(889) (solo CPU CJ1M 21/22/23) | 862 |
| 3-21-10 | PULSE WITH VARIABLE DUTY FACTOR: PWM(891) (solo CPU CJ1M 21/22/23) | 865 |
| 3-22 | Istruzioni di step | 867 |
| 3-22-1 | STEP DEFINE e STEP START: STEP(008)/SNXT(009) | 868 |
| 3-23 | Istruzioni per moduli di I/O di base | 885 |
| 3-23-1 | I/O REFRESH: IORF(097) | 885 |
| 3-23-2 | 7-SEGMENT DECODER SDEC(078) | 888 |
| 3-23-3 | DIGITAL SWITCH INPUT – DSW(210) | 890 |
| 3-23-4 | TEN KEY INPUT – TKY(211) | 896 |
| 3-23-5 | HEXADECIMAL KEY INPUT – HKY(212) | 899 |
| 3-23-6 | MATRIX INPUT: MTR(213) | 904 |
| 3-23-7 | 7-SEGMENT DISPLAY OUTPUT – 7SEG(214) | 908 |
| 3-23-8 | INTELLIGENT I/O READ: IORD(222) | 913 |
| 3-23-9 | INTELLIGENT I/O WRITE: IOWR(223) | 917 |
| 3-23-10 | CPU BUS UNIT I/O REFRESH: DLNK(226) | 921 |
| 3-24 | Istruzioni per la comunicazione seriale. | 926 |
| 3-24-1 | Comunicazione seriale | 926 |
| 3-24-2 | PROTOCOL MACRO: PMCR(260) | 928 |
| 3-24-3 | TRANSMIT: TXD(236) | 937 |
| 3-24-4 | RECEIVE: RXD(235) | 944 |
| 3-24-5 | TRANSMIT VIA SERIAL COMMUNICATIONS UNIT: TXDU(256) | 952 |
| 3-24-6 | RECEIVE VIA SERIAL COMMUNICATIONS UNIT: RXDU(255) | 960 |
| 3-24-7 | CHANGE SERIAL PORT SETUP: STUP(237) | 968 |
| 3-25 | Istruzioni di rete | 973 |
| 3-25-1 | Informazioni sulle operazioni con SYSMAC NET Link/SYSMAC LINK | 973 |
| 3-25-2 | Informazioni sulle istruzioni per messaggi espliciti | 986 |
| 3-25-3 | NETWORK SEND: SEND(090) | 991 |
| 3-25-4 | NETWORK RECEIVE: RECV(098) | 997 |
| 3-25-5 | DELIVER COMMAND: CMND(490) | 1003 |
| 3-25-6 | EXPLICIT MESSAGE SEND: EXPLT(720) | 1013 |
| 3-25-7 | EXPLICIT GET ATTRIBUTE: EGATR(721) | 1021 |
| 3-25-8 | EXPLICIT SET ATTRIBUTE: ESATR(722) | 1028 |
| 3-25-9 | EXPLICIT WORD READ: ECHR(723) | 1034 |
| 3-25-10 | EXPLICIT WORD WRITE: ECHWR(724) | 1038 |
| 3-26 | Istruzioni per la memoria dei file | 1042 |
| 3-26-1 | Avvertenze per l'utilizzo delle schede di memoria | 1042 |
| 3-26-2 | READ DATA FILE: FREAD(700) | 1045 |
| 3-26-3 | WRITE DATA FILE: FWRT(701) | 1052 |
| 3-27 | Istruzioni di visualizzazione: DISPLAY MESSAGE: MSG(046) | 1058 |
| 3-28 | Istruzioni per l'orologio. | 1061 |
| 3-28-1 | CALENDAR ADD: CADD(730) | 1061 |
| 3-28-2 | CALENDAR SUBTRACT: CSUB(731) | 1065 |
| 3-28-3 | HOURS TO SECONDS: SEC(065) | 1068 |
| 3-28-4 | SECONDS TO HOURS: HMS(066) | 1070 |
| 3-28-5 | CLOCK ADJUSTMENT: DATE(735) | 1073 |
| 3-29 | Istruzioni di debug. | 1075 |
| 3-29-1 | Campionamento Trace Memory: TRSM(045) | 1075 |
| 3-30 | Istruzioni di diagnostica per malfunzionamento. | 1079 |
| 3-30-1 | FAILURE ALARM: FAL(006) | 1079 |
| 3-30-2 | SEVERE FAILURE ALARM: FALS(007) | 1087 |
| 3-30-3 | FAILURE POINT DETECTION: FPD(269) | 1095 |

| | | |
|---------|--|------|
| 3-31 | Altre istruzioni | 1104 |
| 3-31-1 | SET CARRY: STC(040) | 1104 |
| 3-31-2 | CLEAR CARRY: CLC(041) | 1105 |
| 3-31-3 | SELECT EM BANK: EMBC(281) | 1106 |
| 3-31-4 | EXTEND MAXIMUM CYCLE TIME: WDT(094) | 1108 |
| 3-31-5 | SAVE CONDITION FLAGS: CCS(282) | 1110 |
| 3-31-6 | LOAD CONDITION FLAGS: CCL(283) | 1112 |
| 3-31-7 | CONVERT ADDRESS FROM CV: FRMCV(284) | 1113 |
| 3-31-8 | CONVERT ADDRESS TO CV: TOCV(285) | 1117 |
| 3-31-9 | DISABLE PERIPHERAL SERVICING: IOSP(287) (solo CS1-H, CJ1-H e CJ1M) .. | 1121 |
| 3-31-10 | ENABLE PERIPHERAL SERVICING: IORS(288) (solo CS1-H, CJ1-H e CJ1M) .. | 1123 |
| 3-32 | Istruzioni di programmazione a blocchi | 1124 |
| 3-32-1 | Introduzione | 1124 |
| 3-32-2 | BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801) | 1128 |
| 3-32-3 | BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812) | 1131 |
| 3-32-4 | Diramazione: IF(802), ELSE(803) e IEND(804) | 1133 |
| 3-32-5 | CONDITIONAL BLOCK EXIT (NOT): EXIT (NOT)(806) | 1137 |
| 3-32-6 | ONE CYCLE AND WAIT (NOT): WAIT(805)/WAIT(805) NOT | 1140 |
| 3-32-7 | TIMER WAIT: TIMW(813) and TIMWX(816) | 1144 |
| 3-32-8 | COUNTER WAIT: CNTW(814) e CNTWX(818) | 1147 |
| 3-32-9 | HIGH-SPEED TIMER WAIT: TMHW(815) e TMHWX(817) | 1150 |
| 3-32-10 | Loop Control: LOOP(809)/LEND(810)/LEND(810) NOT | 1153 |
| 3-33 | Istruzioni di elaborazione delle stringhe di testo | 1158 |
| 3-33-1 | Panoramica sull'elaborazione delle stringhe di testo | 1158 |
| 3-33-2 | MOV STRING: MOV\$(664) | 1159 |
| 3-33-3 | CONCATENATE STRING: +\$(656) | 1161 |
| 3-33-4 | GET STRING LEFT: LEFT\$(652) | 1164 |
| 3-33-5 | GET STRING RIGHT: RGHT\$(653) | 1166 |
| 3-33-6 | GET STRING MIDDLE: MID\$(654) | 1168 |
| 3-33-7 | FIND IN STRING: FIND\$(660) | 1171 |
| 3-33-8 | STRING LENGTH: LEN\$(650) | 1173 |
| 3-33-9 | REPLACE IN STRING: RPLC\$(661) | 1175 |
| 3-33-10 | DELETE STRING: DEL\$(658) | 1178 |
| 3-33-11 | EXCHANGE STRING: XCHG\$(665) | 1180 |
| 3-33-12 | CLEAR STRING: CLR\$(666) | 1182 |
| 3-33-13 | INSERT INTO STRING: INSS\$(657) | 1184 |
| 3-33-14 | Istruzioni di confronto di stringhe (da 670 a 675) | 1187 |
| 3-34 | Istruzioni di controllo dei task | 1192 |
| 3-34-1 | TASK ON: TKON(820) | 1192 |
| 3-34-2 | TASK OFF: TKOF(821) | 1196 |
| 3-35 | Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva) | 1199 |
| 3-35-1 | BLOCK TRANSFER: XFERC(565) | 1201 |
| 3-35-2 | SINGLE WORD DISTRIBUTE: DISTC(566) | 1203 |
| 3-35-3 | DATA COLLECT: COLLC(567) | 1206 |
| 3-35-4 | MOVE BIT: MOVBC(568) | 1211 |
| 3-35-5 | BIT COUNTER: BCNTC(621) | 1212 |
| 3-35-6 | GET VARIABLE ID: GETID(286) | 1214 |

3-1 Descrizioni delle istruzioni: notazione e layout

Le istruzioni sono raggruppate e descritte per funzionalità. Per un elenco delle istruzioni per mnemonico con l'indicazione del numero di pagina per ciascuna istruzione illustrata nel presente capitolo, fare riferimento a *2-3 Elenco alfabetico delle istruzioni in base al codice mnemonico*.

La descrizione di ogni istruzione è organizzata come indicato nella tabella qui di seguito.

| Elemento | | Descrizione | | | | | | | | | | | | |
|---|--|--|-----------------------------|---|------------|-------------------|---|----------------------------------|----|--|-------------------------------|--|---|----------------------------------|
| Nome e mnemonico | | L'intestazione di ciascuna sezione è costituita dal nome dell'istruzione seguito dal mnemonico e dal codice funzione indicato tra parentesi. Ad esempio, MOVE BIT MOVB(082) | | | | | | | | | | | | |
| Scopo | | Lo scopo principale dell'istruzione è riportato dopo l'intestazione della sezione. | | | | | | | | | | | | |
| Simbolo programmazione ladder e nomi operando | | Indicazione del simbolo programmazione ladder utilizzato per rappresentare l'istruzione in CX-Programmer, come nell'esempio dell'istruzione MOVE BIT fornito qui di seguito. Oltre al simbolo di programmazione ladder, viene indicato il nome di ciascun operando. <table border="1" style="margin-left: 40px;"> <tr> <td style="border: none;">—</td> <td style="border: 1px solid black; padding: 2px;">MOVB(082)</td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: none;">S: Dati o canale sorgente</td> </tr> <tr> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px;">C</td> <td style="border: none;">C: Canale di controllo</td> </tr> <tr> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px;">D</td> <td style="border: none;">D: Canale di destinazione</td> </tr> </table> | — | MOVB(082) | | | S | S: Dati o canale sorgente | | C | C: Canale di controllo | | D | D: Canale di destinazione |
| — | MOVB(082) | | | | | | | | | | | | | |
| | S | S: Dati o canale sorgente | | | | | | | | | | | | |
| | C | C: Canale di controllo | | | | | | | | | | | | |
| | D | D: Canale di destinazione | | | | | | | | | | | | |
| Variazioni | Variazioni | <p>Le variazioni utili per controllare l'esecuzione dell'istruzione in circostanze particolari sono fornite in forma mnemonica. Ogni variazione non supportata da un'istruzione viene indicata con "Non supportata".</p> <ul style="list-style-type: none"> Eseguita a ogni ciclo con condizione ON: l'istruzione viene eseguita se riceve una condizione di esecuzione ON. Eseguita una sola volta per differenziazione sul fronte di salita: l'istruzione viene eseguita durante il ciclo successivo soltanto quando la condizione di esecuzione è passata da OFF a ON. Eseguita una sola volta per differenziazione sul fronte di discesa: l'istruzione viene eseguita durante il ciclo successivo soltanto quando la condizione di esecuzione è passata da ON a OFF. Eseguita sempre: l'istruzione non necessita di alcuna condizione di esecuzione e viene eseguita a ogni ciclo. Crea condizione ON....: l'istruzione viene eseguita a ogni ciclo per creare una condizione di esecuzione per l'istruzione successiva. <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Variazioni</th> <th>Eseguita a ogni ciclo con condizione ON</th> <th>MOVB(082)</th> </tr> </thead> <tbody> <tr> <td></td> <td>Eseguita una sola volta per differenziazione sul fronte di salita</td> <td>@MOVB(082)</td> </tr> <tr> <td></td> <td>Eseguita una sola volta per differenziazione sul fronte di discesa</td> <td>Non supportata</td> </tr> </tbody> </table> | Variazioni | Eseguita a ogni ciclo con condizione ON | MOVB(082) | | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVB(082) | | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata | | | |
| Variazioni | Eseguita a ogni ciclo con condizione ON | MOVB(082) | | | | | | | | | | | | |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVB(082) | | | | | | | | | | | | |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata | | | | | | | | | | | | |
| Variazioni | Variazioni | | | | | | | | | | | | | |
| | Aggiornamento immediato | <p>È possibile specificare l'aggiornamento immediato per alcune istruzioni al fine di aggiornare gli I/O durante l'esecuzione dell'istruzione. Se tale aggiornamento è supportato, l'indicazione viene fornita in forma mnemonica. Altrimenti risulta "Non supportato".</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="border: 1px solid black; padding: 2px;">Aggiornamento immediato</td> <td style="border: 1px solid black; padding: 2px;">Non supportata</td> </tr> </table> | Aggiornamento immediato | Non supportata | | | | | | | | | | |
| Aggiornamento immediato | Non supportata | | | | | | | | | | | | | |
| Aree di programma applicabili | | <p>Indicazione delle aree di programma in cui è possibile utilizzare l'istruzione. "OK" indica le aree in cui è possibile utilizzare l'istruzione.</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Aree di programma a blocchi</th> <th>Aree di programma step</th> <th>Subroutine</th> <th>Task ad interrupt</th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; padding: 2px;">OK</td> <td style="border: 1px solid black; padding: 2px;">OK</td> <td style="border: 1px solid black; padding: 2px;">OK</td> <td style="border: 1px solid black; padding: 2px;">OK</td> </tr> </tbody> </table> | Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt | OK | OK | OK | OK | | | | |
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt | | | | | | | | | | | |
| OK | OK | OK | OK | | | | | | | | | | | |

| Elemento | Descrizione | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------|---|---|-----------------|---------------|----------------|----------|---|---------------------|---|----------------|----------------|---|-----|--------------------|----------------|--|--|---------------------|-----------------|--|-----------------|-------------------------|------------------|--|--|--------------------|------------------|--|--|---------|--------------------|--|--|---------------------|--------------------|--|--|
| Operandi | <p>Quando è necessario viene indicato il significato dei canali e dei bit utilizzati in specifici operandi, come i canali di controllo.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Caratteristiche operando | <p>Gli indirizzi delle aree di memoria utilizzabili da ciascun operando sono elencati in una tabella simile alla seguente. Le lettere che compaiono a sinistra nell'intestazione delle colonne sono identiche a quelle utilizzate nel simbolo programmazione ladder. "..." indica quando non è possibile che un'area sia specifica per un operando.</p> <table border="1"> <thead> <tr> <th>Area</th> <th>S</th> <th>C</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>Area CIO</td> <td colspan="3">Da CIO 0000 a CIO 6143</td> </tr> <tr> <td>Area di lavoro</td> <td colspan="3">Da W000 a W511</td> </tr> <tr> <td>Area bit ritentivi</td> <td colspan="3">Da H000 a H511</td> </tr> <tr> <td>Area bit ausiliaria</td> <td>Da A000 ad A959</td> <td></td> <td>Da A448 ad A959</td> </tr> <tr> <td>Area del temporizzatore</td> <td colspan="3">Da T0000 a T4095</td> </tr> <tr> <td>Area del contatore</td> <td colspan="3">Da C0000 a C4095</td> </tr> <tr> <td>Area DM</td> <td colspan="3">Da D00000 a D32767</td> </tr> <tr> <td>Area EM senza banco</td> <td colspan="3">Da E00000 a E32767</td> </tr> </tbody> </table> | Area | S | C | D | Area CIO | Da CIO 0000 a CIO 6143 | | | Area di lavoro | Da W000 a W511 | | | Area bit ritentivi | Da H000 a H511 | | | Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 | Area del temporizzatore | Da T0000 a T4095 | | | Area del contatore | Da C0000 a C4095 | | | Area DM | Da D00000 a D32767 | | | Area EM senza banco | Da E00000 a E32767 | | |
| Area | S | C | D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area CIO | Da CIO 0000 a CIO 6143 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area di lavoro | Da W000 a W511 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area bit ritentivi | Da H000 a H511 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area del temporizzatore | Da T0000 a T4095 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area del contatore | Da C0000 a C4095 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area DM | Da D00000 a D32767 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Area EM senza banco | Da E00000 a E32767 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Descrizione | Informazioni sulla funzione dell'istruzione e sugli operandi utilizzati dall'istruzione stessa. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flag | <p>La tabella dei flag indica lo stato dei flag di condizione immediatamente dopo l'esecuzione dell'istruzione. I flag che non compaiono nell'elenco sono estranei all'istruzione. "OFF" indica che un flag viene impostato su OFF immediatamente dopo l'esecuzione dell'istruzione indipendentemente dai risultati dell'esecuzione stessa.</p> <table border="1"> <thead> <tr> <th>Nome</th> <th>Etichetta</th> <th>Funzionamento</th> </tr> </thead> <tbody> <tr> <td>Flag di errore</td> <td>ER</td> <td>ON se i dati di controllo rientrano negli intervalli. OFF in tutti gli altri casi.</td> </tr> <tr> <td>Flag di uguaglianza</td> <td>=</td> <td>OFF</td> </tr> <tr> <td>Flag negativo</td> <td>N</td> <td>OFF</td> </tr> </tbody> </table> | Nome | Etichetta | Funzionamento | Flag di errore | ER | ON se i dati di controllo rientrano negli intervalli. OFF in tutti gli altri casi. | Flag di uguaglianza | = | OFF | Flag negativo | N | OFF | | | | | | | | | | | | | | | | | | | | | | | | |
| Nome | Etichetta | Funzionamento | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flag di errore | ER | ON se i dati di controllo rientrano negli intervalli. OFF in tutti gli altri casi. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flag di uguaglianza | = | OFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flag negativo | N | OFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Avvertenze | Avvertenze particolari necessarie nell'utilizzo dell'istruzione. Assicurarsi di leggere e di rispettare tali avvertenze. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Esempio | Un esempio di utilizzo dell'istruzione con operandi specifici consente di osservare più dettagliatamente la funzione dell'istruzione. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Costanti

L'inserimento delle costanti per gli operandi viene indicato nel seguente elenco.

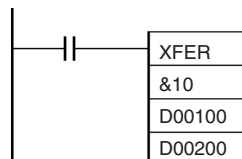
Descrizioni e caratteristiche degli operandi

- Operandi con specifica di stringhe di bit (di norma inserimento in formato esadecimale):
Per gli operandi con specifica di stringhe di bit viene fornito soltanto il formato esadecimale, ad esempio come operando S per l'istruzione MOV(021) viene indicato esclusivamente "Da #0000 a #FFFF". In CX-Programmer è comunque possibile inserire stringhe di bit in formato decimale utilizzando il prefisso &.
- Operandi con specifica di valori numerici (di norma inserimento in formato decimale, compresi i numeri di salto):
Per gli operandi con specifica di valori numerici vengono indicati sia il formato decimale che quello esadecimale, ad esempio "Da #0000 a #FFFF" e "Da &0 a &65535" vengono forniti nel caso dell'operando N per l'istruzione XFER(070).

- Operandi con specifica di numeri di controllo (tranne i numeri di salto):
Per i numeri di controllo viene fornito il formato decimale; ad esempio “Da 0 a 1023” è l’indicazione dell’operando N per l’istruzione SBS(091).

Esempi

Negli esempi le costanti sono indicate utilizzando la notazione di CX-Programmer; ad esempio gli operandi con specifica di valori numerici vengono indicati in formato decimale con un prefisso &, come illustrato qui di seguito.



La tabella seguente riporta i metodi di inserimento delle costanti per i dispositivi di programmazione.

| Operando | CX-Programmer | Console di programmazione |
|---|---|---|
| Operandi con specifica di stringhe di bit (di norma inserimento in formato esadecimale) | Inserimento in formato decimale con un prefisso & o in formato esadecimale con un prefisso # (vedere nota). | È possibile premere il tasto Cont/# al fine di inserire valori esadecimale con un prefisso # per impostazione predefinita. Per passare dai valori esadecimale (con prefisso #) ai decimali con segno (con +/-) e ai decimali senza segno (con prefisso &) è possibile premere il tasto CHG. |
| Operandi con specifica di valori numerici (di norma inserimento in formato decimale) | | |
| Operandi con specifica di numeri di controllo (tranne i numeri di salto) | Inserimento in formato decimale con un prefisso # (vedere nota). | Inserimento diretto in formato decimale. Se il prefisso & è aggiunto in modo automatico è possibile premere il tasto CHG per passare dai valori decimali senza segno (con prefisso &), a quelli esadecimale (con prefisso #) e ai decimali con segno (con +/-). Se non è visualizzato alcun prefisso, è necessario immettere il valore in formato decimale. |

Nota Quando gli operandi vengono immessi in CX-Programmer, gli intervalli di inserimento sono visualizzati con i prefissi corretti.

Flag di condizione

In questo capitolo le etichette della Console di programmazione sono utilizzate per i flag di condizione. Con CX-Programmer, i flag di condizione sono registrati in precedenza come simboli globali con “P_” anteposto al nome del simbolo.

| del temporizzatore | Etichetta della Console di programmazione | Etichetta di CX-Programmer |
|---------------------------|---|----------------------------|
| Flag di errore | ER | P_ER |
| Flag di errore accesso | AER | P_AER |
| Flag di riporto | CY | P_CY |
| Flag di maggiore | > | P_GT |
| Flag di uguaglianza | = | P_EQ |
| Flag di minore | < | P_LT |
| Flag negativo | N | P_N |
| Flag di overflow | OF | P_OF |
| Flag di underflow | UF | P_UF |
| Flag di maggiore o uguale | >= | P_GE |
| Flag di non uguaglianza | <> | P_NE |

| del temporizzatore | Etichetta della Console di programmazione | Etichetta di CX-Programmer |
|-------------------------|---|----------------------------|
| Flag di minore o uguale | <= | P_LE |
| Flag di sempre ON | ON | P_On |
| Flag di sempre OFF | OFF | P_Off |

3-2 Aggiornamenti delle istruzioni e nuove istruzioni

In questo capitolo sono elencati gli aggiornamenti delle istruzioni per le CPU CS1 dotate di suffisso -EV1 nonché per CPU CS1/H e CJ1/H.

3-2-1 Aggiornamenti per CPU della serie CS versione 1

Le istruzioni che seguono sono state aggiornate per moduli CPU versione 1. Per ulteriori informazioni, fare riferimento al numero di pagina specificato.

Le funzioni supportate esclusivamente da CPU versione 1 sono indicate con “(solo -EV1)”

| Nome | Codice mnemonico | Codice funzione | Funzione | Aggiornamento | Pagina |
|-----------------|------------------|-----------------|--|--|--------|
| READ DATA FILE | FREAD | 700 | Attualmente sono supportati entrambi i formati dati CSV e testo (.txt) (in precedenza erano supportati solo dati binari). | Il contenuto dei dati di controllo ha subito alcune modifiche grazie all'aggiunta del formato dei dati, alla presenza dei ritorni a capo e alle specifiche di posizione dei ritorni a capo stessi. | 1045 |
| WRITE DATA FILE | FWRIT | 701 | | | 1052 |
| DELIVER COMMAND | CMND | 490 | Il modulo CPU è ora in grado di inviare un comando FINS a se stesso (in precedenza ciò non era possibile). | È stata implementata la possibilità di inviare un comando FINS al modulo CPU eseguendo CMND(490). | 1003 |

3-2-2 Aggiornamenti per CPU CS1/H e CJ1/H

Nuove istruzioni

Le istruzioni che seguono sono state aggiunte ai moduli CPU CS1/H e CJ1/H.

Istruzioni di uscita sequenza

SINGLE BIT SET, SETB(532)
SINGLE BIT RESET, RSTB(533)
SINGLE BIT OUTPUT, OUTB(534)

Istruzioni di confronto dei dati

AREA RANGE COMPARE, ZCP(088)
DOUBLE AREA RANGE COMPARE, ZCPL(116)

Istruzioni di calcolo e conversione della virgola mobile

Istruzioni di confronto dei dati per la virgola mobile =F, <>F, <F, <=F, >F e >=F (da 329 a 334)

FLOATING POINT TO ASCII, FSTR(448)
ASCII TO FLOATING POINT, VAL(449)

Istruzioni di calcolo e conversione a virgola mobile in doppia precisione

Istruzioni di confronto in doppia precisione =D, <>D, <D, <=D, >D e >=D (da 335 a 340)

DOUBLE FLOATING TO 16-BIT BINARY, FIXD(841)
DOUBLE FLOATING TO 32-BIT BINARY, FIXLD(8420)
16-BIT BINARY TO DOUBLE FLOATING, DBL(843)
32-BIT BINARY TO DOUBLE FLOATING, DBLL(844)
DOUBLE FLOATING-POINT ADD, +D(845)
DOUBLE FLOATING-POINT SUBTRACT, -D(846)
DOUBLE FLOATING-POINT MULTIPLY, *D(847)

DOUBLE FLOATING-POINT DIVIDE, /D(848)
DOUBLE DEGREES TO RADIANS, RADD(849)
DOUBLE RADIANS TO DEGREES, DEGD(850)
DOUBLE SINE, SIND(851)
DOUBLE COSINE, COSD(852)
DOUBLE TANGENT, TAND(853)
DOUBLE ARC SINE, ASIND(854)
DOUBLE ARC COSINE, ACOSD(855)
DOUBLE ARC TANGENT, ATAND(856)
DOUBLE SQUARE ROOT, SQRTD(857)
DOUBLE EXPONENT, EXPD(858)
DOUBLE LOGARITHM, LOGD(859)
DOUBLE EXPONENTIAL POWER, PWRD(860)

Istruzioni di elaborazione dei dati delle tabelle

STACK SIZE READ, SNUM(638)
STACK DATA READ, SREAD(639)
STACK DATA WRITE, SWRIT(640)
STACK DATA INSERT, SINS(641)
STACK DATA DELETE, SDEL(642)

Istruzioni di controllo dei dati

PID CONTROL WITH AUTOTUNING, PIDAT(191)

Istruzioni di subroutine

GLOBAL SUBROUTINE CALL, GSBS(750)
GLOBAL SUBROUTINE ENTRY, GSBN(751)
GLOBAL SUBROUTINE RETURN, GRET(752)

Istruzioni per moduli di I/O

CPU BUS UNIT I/O REFRESH, DLNK(226)

Altre istruzioni

SAVE CONDITION FLAGS, CCS(282)
LOAD CONDITION FLAGS, CCL(283)
CONVERT ADDRESS FROM CV, FRMCV(284)
CONVERT ADDRESS TO CV, TOCV(285)
DISABLE PERIPHERAL SERVICING, IOSP(287)
ENABLE PERIPHERAL SERVICING, IORS(288)

Nuove istruzioni

Le istruzioni che seguono sono state aggiornate per CPU CS1/H e CJ1/H.

Istruzioni matematiche speciali

ARITHMETIC PROCESS, APR(069)

Istruzioni per la diagnostica di funzionamento incorretto

FAILURE ALARM, FAL(006)
SEVERE FAILURE ALARM, FALS(007)

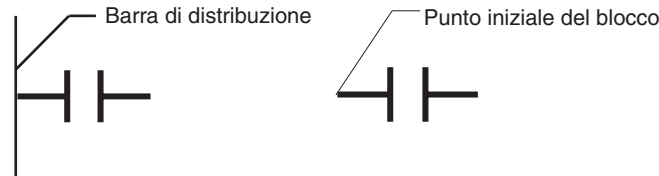
3-3 Istruzioni di ingresso sequenza

3-3-1 LOAD: LD

Scopo

Indica un inizio logico e crea una condizione di esecuzione impostata su ON/OFF in base allo stato ON/OFF del bit operando specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|---------------------------------------|---|------|
| Variazioni | Riavvia la logica e crea una condizione di esecuzione con stato ON a ogni ciclo se il bit operando è impostato su ON | LD |
| | Riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita | @LD |
| | Riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa | %LD |
| Aggiornamento immediato (vedere nota) | | !LD |
| Variazioni combinate | Aggiorna il bit di ingresso, riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota) | !@LD |
| | Aggiorna il bit di ingresso, riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota) | !%LD |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D per sistemi a due CPU.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando LD |
|-------------------------|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area dei flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, N, OF, UF, >, =, <, >=, <>, <=, A1, A0 |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area TR | Da TR0 a TR15 |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |

| Area | Bit operando LD |
|--|--|
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

LD viene utilizzato per il primo bit normalmente aperto dalla barra di distribuzione oppure per il primo bit normalmente aperto di un blocco logico. In assenza di aggiornamento immediato, viene letto il bit specificato nella memoria I/O. Nel caso di aggiornamento immediato, viene letto e utilizzato lo stato del terminal di ingresso del modulo di ingresso di base.

Nelle seguenti circostanze LD è utilizzato come istruzione per indicare un inizio logico.

- Nel collegamento diretto alla barra di distribuzione.
- Quando i blocchi logici sono collegati da AND LD oppure OR LD, ossia, all'inizio di un blocco logico.

Le istruzioni AND LOAD e OR LOAD vengono utilizzate per collegare in serie o in parallelo i blocchi logici che iniziano con LD oppure con LD NOT.

Quando non è possibile collegare le istruzioni relative all'uscita direttamente alla barra di distribuzione, è necessaria almeno un'istruzione LOAD o LOAD NOT per la condizione di esecuzione. In assenza di un'istruzione LOAD o LOAD NOT, si verificherà un errore di programmazione con il controllo del programma da parte della periferica.

Quando i blocchi logici sono collegati da istruzioni AND LOAD o OR LOAD, il numero complessivo di istruzioni AND LOAD/OR LOAD deve corrispondere al numero totale di LOAD/LOAD NOT meno uno. Altrimenti si verificherà un errore di programmazione. Per informazioni dettagliate, fare riferimento alle sezioni 3-3-7 AND LOAD: AND LD e 3-3-8 OR LOAD: OR LD.

Flag

Questa istruzione non influisce su alcun flag.

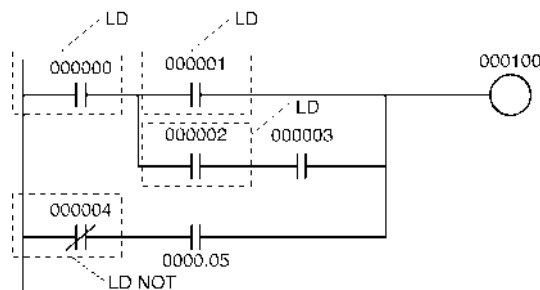
Avvertenze

Per LD è possibile scegliere tra differenziazione ascendente (@) oppure differenziazione discendente (%). Se viene specificata la prima, la condizione di esecuzione si imposta su ON per un ciclo, una volta che lo stato del bit operando è stato attivato. Se viene specificata la differenziazione discendente (%), la condizione di esecuzione si imposta su ON per un ciclo, una volta che lo stato del bit operando è stato disattivato.

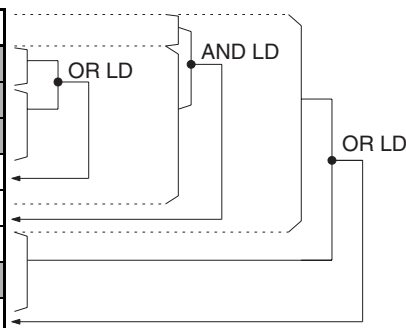
Per LD può essere specificato l'aggiornamento immediato (!). Un'istruzione con refresh immediato aggiorna lo stato del bit d'ingresso appena prima della propria esecuzione per moduli di ingresso di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H multipunto gruppo 2).

È inoltre possibile combinare l'aggiornamento immediato e la differenziazione ascendente/discendente (!@ or !%). Specificando una di queste opzioni, l'ingresso viene aggiornato dal modulo di ingresso di base appena prima dell'esecuzione dell'istruzione e la condizione di esecuzione viene impostata su ON per un ciclo, una volta che lo stato sia passato da OFF a ON o da ON a OFF.

Esempio



| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| LD | 000001 |
| LD | 000002 |
| AND | 000003 |
| OR LD | --- |
| AND LD | --- |
| LD NOT | 000004 |
| AND | 000005 |
| OR LD | --- |
| OUT | 000100 |



3-3-2 LOAD NOT: LD NOT

Scopo

Indica un inizio logico e crea una condizione di esecuzione ON/OFF in base all'inverso dello stato ON/OFF del bit operando specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|---|---|----------|
| Variazioni | Riavvia la logica e crea una condizione di esecuzione con stato ON a ogni ciclo se il bit operando è impostato su OFF | LD NOT |
| | Riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota 1) | @LD NOT |
| | Riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota 1) | %LD NOT |
| Aggiornamento immediato (vedere nota 2) | | !LD NOT |
| Variazioni combinate | Aggiorna il bit di ingresso, riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota 3) | !@LD NOT |
| | Aggiorna il bit di ingresso, riavvia la logica e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota 3) | !%LD NOT |

Nota

1. Le seguenti variazioni sono supportate soltanto da CPU CS1/H, CJ1/H, CJ1M e CS1D. @LD NOT, %LD NOT, !@LD NOT e !%LD NOT.
2. L'aggiornamento immediato non è supportato dalle CPU CS1D per sistemi a due CPU.
3. Variazioni combinate sono supportate da CPU CS1D per sistemi a singola CPU e CPU CS1/H, CJ1/H e CJ1M.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando LD NOT |
|--|---|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area dei flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area TR | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

LD NOT viene utilizzato per il primo bit normalmente chiuso dalla barra di distribuzione oppure per il primo bit normalmente chiuso di un blocco logico. In assenza di aggiornamento immediato, viene letto e invertito il bit specificato nella memoria I/O. Nel caso di aggiornamento immediato, viene letto, invertito e utilizzato lo stato del terminal di ingresso del modulo di ingresso di base.

Nelle seguenti circostanze LD NOT è utilizzato come istruzione per indicare un inizio logico.

- Nel collegamento diretto alla barra di distribuzione.
- Quando i blocchi logici sono collegati da AND LD oppure OR LD (utilizzati all'inizio di un blocco logico).

Le istruzioni AND LOAD e OR LOAD vengono utilizzate per collegare in serie o in parallelo i blocchi logici che iniziano con LD oppure con LD NOT.

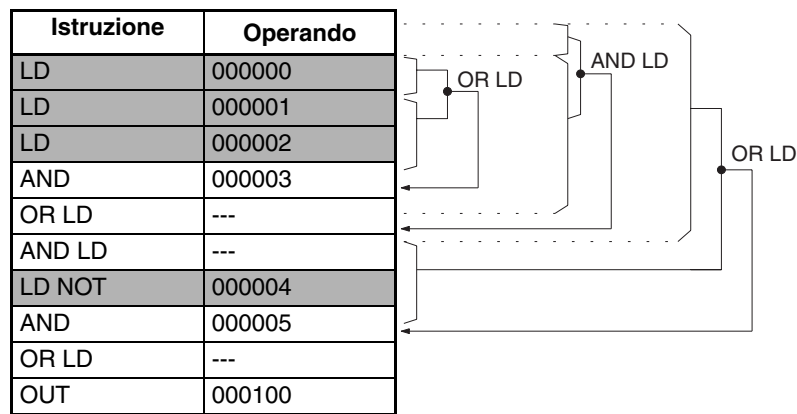
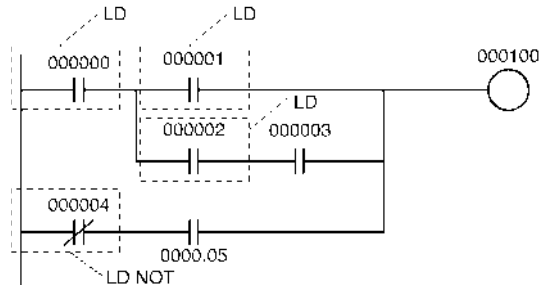
Quando non è possibile collegare le istruzioni relative all'uscita direttamente alla barra di distribuzione, è necessaria almeno un'istruzione LOAD o LOAD NOT per la condizione di esecuzione. In assenza di un'istruzione LOAD o LOAD NOT, si verificherà un errore di programma con il controllo del programma da parte della periferica.

Quando i blocchi logici sono collegati da istruzioni AND LOAD o OR LOAD, il numero complessivo di istruzioni AND LOAD/OR LOAD deve corrispondere al numero totale di LOAD/LOAD NOT meno uno. Altrimenti si verificherà un errore di programmazione.

Flag Questa istruzione non influisce su alcun flag.

Avvertenze Per LD NOT è possibile specificare l'aggiornamento immediato (!). Un'istruzione con refresh immediato aggiorna lo stato del bit d'ingresso appena prima della propria esecuzione per moduli di ingresso di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H multipunto gruppo 2).

Esempio



3-3-3 AND: AND

Scopo Assume un AND dello stato del bit operando specificato e la condizione di esecuzione corrente.

Simbolo programmazione ladder
Variazioni



| | | |
|--|--|-------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il risultato della funzione AND è ON | AND |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita | @AND |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa | %AND |
| Aggiornamento immediato (vedere nota) | | !AND |
| Variazioni combinate | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota) | !@AND |
| | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota) | !%AND |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D per sistemi a due CPU.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando AND |
|--|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area dei flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area TR | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

La funzione AND è utilizzata per un bit normalmente aperto collegato in serie. Non è possibile collegarla direttamente alla barra di distribuzione o utilizzarla all'inizio di un blocco logico. In assenza di aggiornamento immediato, viene letto il bit specificato nella memoria I/O. Nel caso di aggiornamento immediato, viene letto lo stato del terminal di ingresso del modulo di ingresso di base.

Flag

Questa istruzione non influisce su alcun flag.

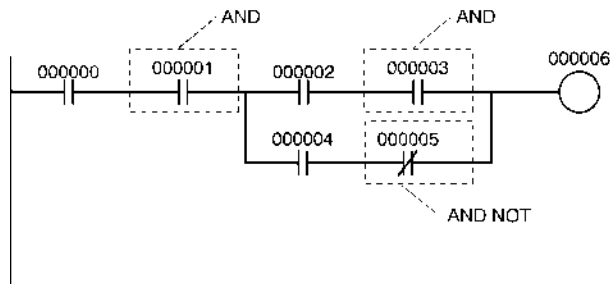
Avvertenze

Per AND è possibile scegliere tra differenziazione ascendente (@) o discendente (%). Se viene specificata la prima (@), la condizione di esecuzione si imposta su ON per un ciclo, una volta che lo stato del bit operando è stato attivato. Se viene specificata la differenziazione discendente (%), la condizione di esecuzione si imposta su ON per un ciclo, una volta che lo stato del bit operando è stato disattivato.

Per AND è possibile specificare l'aggiornamento immediato (!). Un'istruzione con refresh immediato aggiorna lo stato del bit d'ingresso appena prima dell'esecuzione dal modulo di ingresso di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H multipunto gruppo 2).

È inoltre possibile combinare l'aggiornamento immediato e la differenziazione ascendente/discendente (!@ o !%). Specificando una di queste opzioni, l'ingresso viene aggiornato dal modulo di ingresso di base appena prima dell'esecuzione dell'istruzione e la condizione di esecuzione viene impostata su ON per un ciclo, una volta che lo stato sia passato da OFF a ON o da ON a OFF.

Esempio



| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| AND | 000001 |
| LD | 000002 |
| AND | 000003 |
| LD | 000004 |
| AND NOT | 000005 |
| OR LD | --- |
| AND LD | --- |
| OUT | 000006 |

3-3-4 AND NOT: AND NOT

Scopo

Inverte lo stato del bit operando specificato e assume un AND logico con la condizione di esecuzione corrente.

Simbolo programmazione ladder



Variazioni

| | | |
|---|--|-----------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il risultato della funzione AND NOT è ON | AND NOT |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota 1) | @AND NOT |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota 1) | %AND NOT |
| Aggiornamento immediato (vedere nota 2) | | !AND NOT |
| Variazioni combinate | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota 3) | !@AND NOT |
| | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota 3) | !%AND NOT |

- Nota**
- Le seguenti variazioni sono supportate soltanto da CPU CS1/H, CJ1/H, CJ1M e CS1D. @AND NOT, %AND NOT, !@AND NOT e !%AND NOT.
 - L'aggiornamento immediato non è supportato dalle CPU CS1D per sistemi a due CPU.
 - Variazioni combinate sono supportate da CPU CS1D per sistemi a singola CPU e CPU CS1/H, CJ1/H e CJ1M.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando AND NOT |
|----------------|----------------------------|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |

| Area | Bit operando AND NOT |
|--|--|
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area dei flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area TR | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

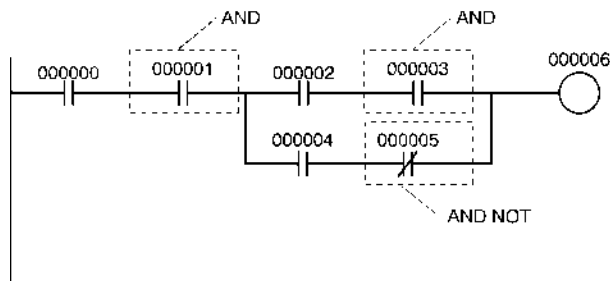
La funzione AND NOT è utilizzata per un bit normalmente chiuso collegato in serie. Non è possibile collegarla direttamente alla barra di distribuzione o utilizzarla all'inizio di un blocco logico. In assenza di aggiornamento immediato, viene letto il bit specificato nella memoria I/O. Nel caso di aggiornamento immediato, viene letto lo stato dei terminal di ingresso del modulo di ingresso di base.

Flag

Questa istruzione non influisce su alcun flag.

Avvertenze

Per AND NOT è possibile specificare l'aggiornamento immediato (!). Un'istruzione con refresh immediato aggiorna lo stato del bit d'ingresso appena prima dell'esecuzione da moduli di ingresso di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H multipunto gruppo 2).

Esempio

| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| AND | 000001 |
| LD | 000002 |
| AND | 000003 |
| LD | 000004 |
| AND NOT | 000005 |

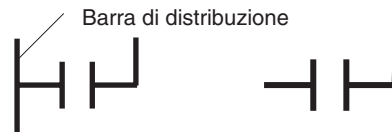
| Istruzione | Operando |
|------------|----------|
| OR LD | --- |
| AND LD | --- |
| OUT | 000006 |

3-3-5 OR: OR

Scopo

Assume un OR logico dello stato ON/OFF del bit operando specificato e la condizione di esecuzione corrente.

Simbolo programmazione ladder



Variazioni

| | | |
|---------------------------------------|--|------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il risultato della funzione OR è ON | OR |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita | @OR |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa | %OR |
| Aggiornamento immediato (vedere nota) | | !OR |
| Variazioni combinate | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota) | !@OR |
| | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota) | !%OR |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D per sistemi a due CPU.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando OR |
|--------------------------------------|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area dei flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |

| Area | Bit operando OR |
|--|--|
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

La funzione OR è utilizzata per un bit normalmente aperto collegato in parallelo. Un bit aperto normalmente viene configurato per formare un OR logico con un blocco logico che inizia con un'istruzione LOAD o LOAD NOT collegata alla barra di distribuzione o all'inizio del blocco logico. In assenza di aggiornamento immediato, viene letto il bit specificato nella memoria I/O. Nel caso di aggiornamento immediato, viene letto lo stato del terminal di ingresso del modulo di ingresso di base.

Flag

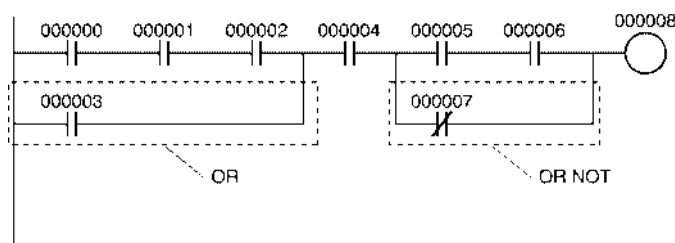
Questa istruzione non influisce su alcun flag.

Avvertenze

Per OR è possibile scegliere tra differenziazione ascendente (@) o discendente (%). Se viene specificata la prima (@), la condizione di esecuzione si imposta su ON per un ciclo, una volta che lo stato del bit operando è stato attivato. Se viene specificata la differenziazione discendente (%), la condizione di esecuzione si imposta su ON per un ciclo, una volta che lo stato del bit operando è stato disattivato.

Per OR è possibile specificare l'aggiornamento immediato (!). Un'istruzione con refresh immediato aggiorna lo stato del bit d'ingresso appena prima dell'esecuzione dal modulo di ingresso di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H multipunto gruppo 2).

È inoltre possibile combinare l'aggiornamento immediato e la differenziazione ascendente/discendente (!@ o !%). Specificando una di queste opzioni, l'ingresso viene aggiornato dal modulo di ingresso di base appena prima dell'esecuzione dell'istruzione e la condizione di esecuzione viene impostata su ON per un ciclo, una volta che lo stato del bit operando sia passato da OFF a ON o da ON a OFF.

Esempio

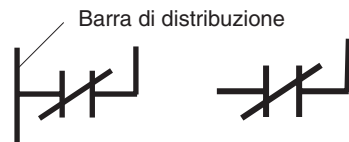
| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| AND | 000001 |
| AND | 000002 |
| OR | 000003 |
| AND | 000004 |
| LD | 000005 |
| AND | 000006 |
| OR NOT | 000007 |
| AND LD | --- |
| OUT | 000008 |

3-3-6 OR NOT: OR NOT

Scopo

Inverte lo stato del bit specificato e assume un OR logico con la condizione di esecuzione corrente.

Simbolo programmazione ladder



Variazioni

| | | |
|---|--|----------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il risultato della funzione OR NOT è ON | OR NOT |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota 1) | @OR NOT |
| | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota 1) | %OR NOT |
| Aggiornamento immediato (vedere nota 2) | | !OR NOT |
| Variazioni combinate | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita (vedere nota 3) | !@OR NOT |
| | Aggiorna il bit di ingresso e crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa (vedere nota 3) | !%OR NOT |

Nota

1. Le seguenti variazioni sono supportate soltanto da CPU CS1/H, CJ1/H, CJ1M e CS1D. @OR NOT, %OR NOT, !@OR NOT e !%OR NOT.
2. L'aggiornamento immediato non è supportato dalle CPU CS1D per sistemi a due CPU.
3. Variazioni combinate sono supportate da CPU CS1D per sistemi a singola CPU e CPU CS1/H, CJ1/H e CJ1M.

Aree di programma applicabili

| Arete di programma a blocchi | Arete di programma step | Subroutine | Task ad interrupt |
|------------------------------|-------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando OR NOT |
|--------------------------------------|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area dei flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, N, OF, UF, >, =, <, >=, <=, A1, A0 |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area TR | --- |
| Area DM | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |

| Area | Bit operando OR NOT |
|--|--|
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

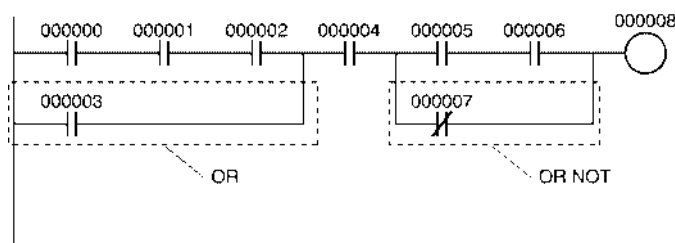
La funzione OR NOT è utilizzata per un bit normalmente chiuso collegato in parallelo. Un bit chiuso normalmente viene configurato per formare un OR logico con un blocco logico che inizia con un'istruzione LOAD o LOAD NOT collegata alla barra di distribuzione o all'inizio del blocco logico. In assenza di aggiornamento immediato, viene letto il bit specificato nella memoria I/O. Nel caso di aggiornamento immediato, viene letto lo stato del terminal di ingresso del modulo di ingresso di base.

Flag

Questa istruzione non influisce su alcun flag.

Avvertenze

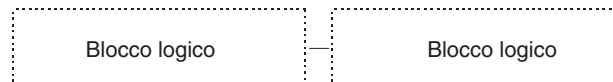
Per OR NOT è possibile specificare l'aggiornamento immediato (!). Un'istruzione con refresh immediato aggiorna lo stato del bit d'ingresso appena prima dell'esecuzione dal modulo di ingresso di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H multipunto gruppo 2).

Esempio

| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| AND | 000001 |
| AND | 000002 |
| OR | 000003 |
| AND | 000004 |
| LD | 000005 |
| AND | 000006 |
| OR NOT | 000007 |
| AND LD | --- |
| OUT | 000008 |

3-3-7 AND LOAD: AND LD**Scopo**

Assume un AND logico tra blocchi logici.

Simbolo programmazione ladder**Variazioni**

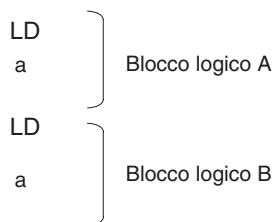
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il risultato della funzione AND è ON | AND LD |
|-------------------------|---|----------------|
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Descrizione

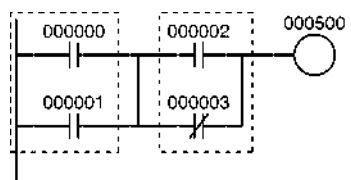
AND LD collega in serie il blocco logico che si trova immediatamente prima dell'istruzione a un altro blocco logico.



AND LD Collegamento seriale tra i blocchi logici A e B.

Il blocco logico è l'insieme di tutte le istruzioni di un'istruzione LOAD o LOAD NOT fino ad appena prima della successiva istruzione LOAD o LOAD NOT sugli stessi rung.

Nel seguente diagramma i due blocchi logici sono indicati da linee tratteggiate. L'esempio mostra che una condizione di esecuzione con stato ON viene prodotta quando una delle condizioni di esecuzione nel blocco logico sinistro è impostata su ON (cioè quando CIO 000000 o CIO 000001 sono ON) e quando una delle condizioni di esecuzione nel blocco logico destro è impostata su ON (ossia quando CIO 000002 è ON o CIO 000003 è OFF).



Flag

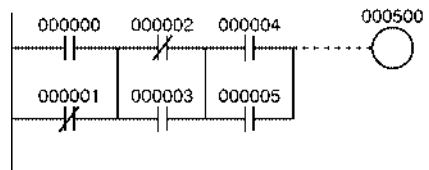
Questa istruzione non influisce su alcun flag.

Avvertenze

È possibile collegare tre o più blocchi logici in serie utilizzando questa istruzione in primo luogo per collegare due blocchi logici, quindi il terzo e in ordine i successivi. È inoltre possibile ripetere questa istruzione dopo tre o più blocchi logici e collegarli in serie.

Quando un blocco logico è collegato da istruzioni AND LOAD o OR LOAD, il numero complessivo di istruzioni AND LOAD/OR LOAD deve corrispondere al numero totale di istruzioni LOAD/LOAD NOT meno uno. Altrimenti si verificherà un errore di programma.

Esempio



Esempio di codice (1)

| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| OR NOT | 000001 |
| LD NOT | 000002 |
| OR | 000003 |
| AND LD | --- |
| LD | 000004 |
| OR | 000005 |

| Istruzione | Operando |
|------------|----------|
| AND LD | --- |
| . | . |
| . | . |
| OUT | 000500 |

Esempio di codice (2)

| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| OR NOT | 000001 |
| LD NOT | 000002 |
| OR | 000003 |
| LD | 000004 |
| OR | 000005 |
| . | . |
| . | . |
| AND LD | --- |
| AND LD | --- |
| . | . |
| . | . |
| OUT | 000500 |

L'istruzione AND LOAD può essere usata in modo ripetuto. Nel secondo metodo di programmazione (2), tuttavia, il numero delle istruzioni AND LOAD raggiunge il numero delle precedenti istruzioni LOAD e LOAD NOT meno 1.

Con il metodo (2), è opportuno assicurarsi che il numero complessivo di istruzioni LOAD e LOAD NOT prima di AND LOAD non sia superiore a otto. Per disporre di un numero superiore di istruzioni, utilizzare il metodo (1). In presenza di nove o più istruzioni con il metodo (2), si verificherà un errore di programma durante il controllo del programma da parte della periferica.

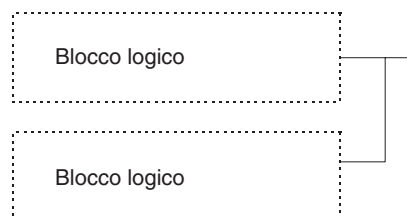
Codice

| Indirizzo | Istruzione | Operando |
|-----------|------------|----------|
| 000000 | LD | 000000 |
| 000001 | OR | 000001 |
| 000002 | LD | 000002 |
| 000003 | OR NOT | 000003 |
| 000004 | AND LD | --- |
| 000005 | OUT | 000500 |

Secondo LD: Utilizzato per il primo bit del blocco successivo collegato in serie al blocco precedente.

3-3-8 OR LOAD: OR LD**Scopo**

Assume un OR logico tra blocchi logici.

Simbolo programmazione ladder**Variazioni**

| | | |
|--------------------------------|--|----------------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il risultato della funzione AND è ON | OR LD |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Descrizione

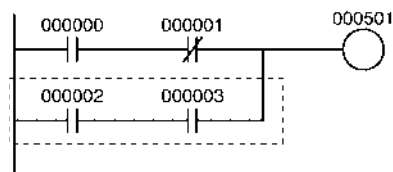
AND LD collega in parallelo il blocco logico appena prima che l'istruzione venga collegata a un altro blocco logico.



OR LD Collegamento parallelo tra i blocchi logici A e B.

Il blocco logico è l'insieme di tutte le istruzioni di un'istruzione LOAD o LOAD NOT fino ad appena prima della successiva istruzione LOAD o LOAD NOT sugli stessi rung.

Il seguente diagramma richiede un'istruzione OR LOAD tra il primo e l'ultimo blocco logico. Una condizione di esecuzione con stato ON viene prodotta quando CIO 000000 è impostato su ON e CIO 000001 su OFF oppure quando CIO 000002 e CIO 000003 sono entrambi impostati su ON. Il codice operazione e il codice mnemonico dell'istruzione OR LOAD corrispondono esattamente a quelli di un'istruzione AND LOAD, con la differenza che la condizione di esecuzione corrente è posto in OR all'ultima condizione di esecuzione non utilizzata.



Flag

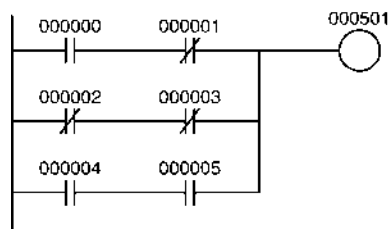
Questa istruzione non influisce su alcun flag.

Avvertenze

È possibile collegare tre o più blocchi logici in parallelo utilizzando questa istruzione in primo luogo per collegare due blocchi logici, quindi il terzo e in ordine i successivi. È inoltre possibile ripetere questa istruzione dopo tre o più blocchi logici e collegarli in parallelo.

Quando un blocco logico è collegato da istruzioni AND LOAD o OR LOAD, il numero complessivo di istruzioni AND LOAD/OR LOAD deve corrispondere al numero totale di LOAD/LOAD NOT meno uno. Altrimenti si verificherà un errore di programma.

Esempio



Esempio di codice (1)

| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| AND NOT | 000001 |
| LD NOT | 000002 |
| AND NOT | 000003 |
| OR LD | --- |
| LD | 000004 |
| AND | 000005 |
| OR LD | --- |
| . | . |
| . | . |
| OUT | 000501 |

Esempio di codice (2)

| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| AND NOT | 000001 |
| LD NOT | 000002 |
| AND NOT | 000003 |
| LD | 000004 |
| AND | 000005 |
| . | . |
| . | . |
| OR LD | --- |
| OR LD | --- |
| . | . |
| . | . |
| OUT | 000501 |

L'istruzione OR LOAD può essere usata in modo ripetuto. Nel secondo metodo di programmazione (2), tuttavia, il numero delle istruzioni OR LOAD raggiunge il numero delle precedenti istruzioni LOAD e LOAD NOT meno 1.

Con il metodo (2), è opportuno assicurarsi che il numero complessivo di istruzioni LOAD e LOAD NOT prima di OR LOAD non sia superiore a otto. Per disporre di un numero superiore di istruzioni, utilizzare il metodo (1). In presenza di nove o più istruzioni con il metodo (2), si verificherà un errore di programma durante il controllo del programma da parte della periferica.

Codice

| Indirizzo | Istruzione | Operando |
|-----------|------------|----------|
| 000100 | LD | 000000 |
| 000101 | AND NOT | 000001 |
| 000102 | LD | 000002 |
| 000103 | AND | 000003 |
| 000104 | OR LD | --- |
| 000105 | OUT | 000501 |

Secondo LD: Utilizzato per il primo bit del blocco successivo collegato in serie al blocco precedente.

3-3-9 Istruzioni di aggiornamento differenziato e immediato

Oltre alle forme consuete, le istruzioni LOAD, AND e OR presentano variazioni di aggiornamento differenziato e immediato nonché due combinazioni possibili.

Oltre alle forme consuete, le istruzioni LOAD NOT, AND NOT, OR NOT, OUT e OUT NOT presentano variazioni di aggiornamento immediato.

Le istruzioni normali e differenziate, le istruzioni di aggiornamento immediato e le istruzioni differenziate di aggiornamento immediato hanno una diversa temporizzazione I/O per i dati gestiti dalle istruzioni.

Le istruzioni normali e differenziate sono eseguite utilizzando l'ingresso dei dati della precedente elaborazione di I/O refresh e i risultati vengono restituiti con l'elaborazione I/O successiva. In questo caso "I/O refresh" si riferisce ai dati scambiati tra la memoria interna della CPU e il modulo di I/O.

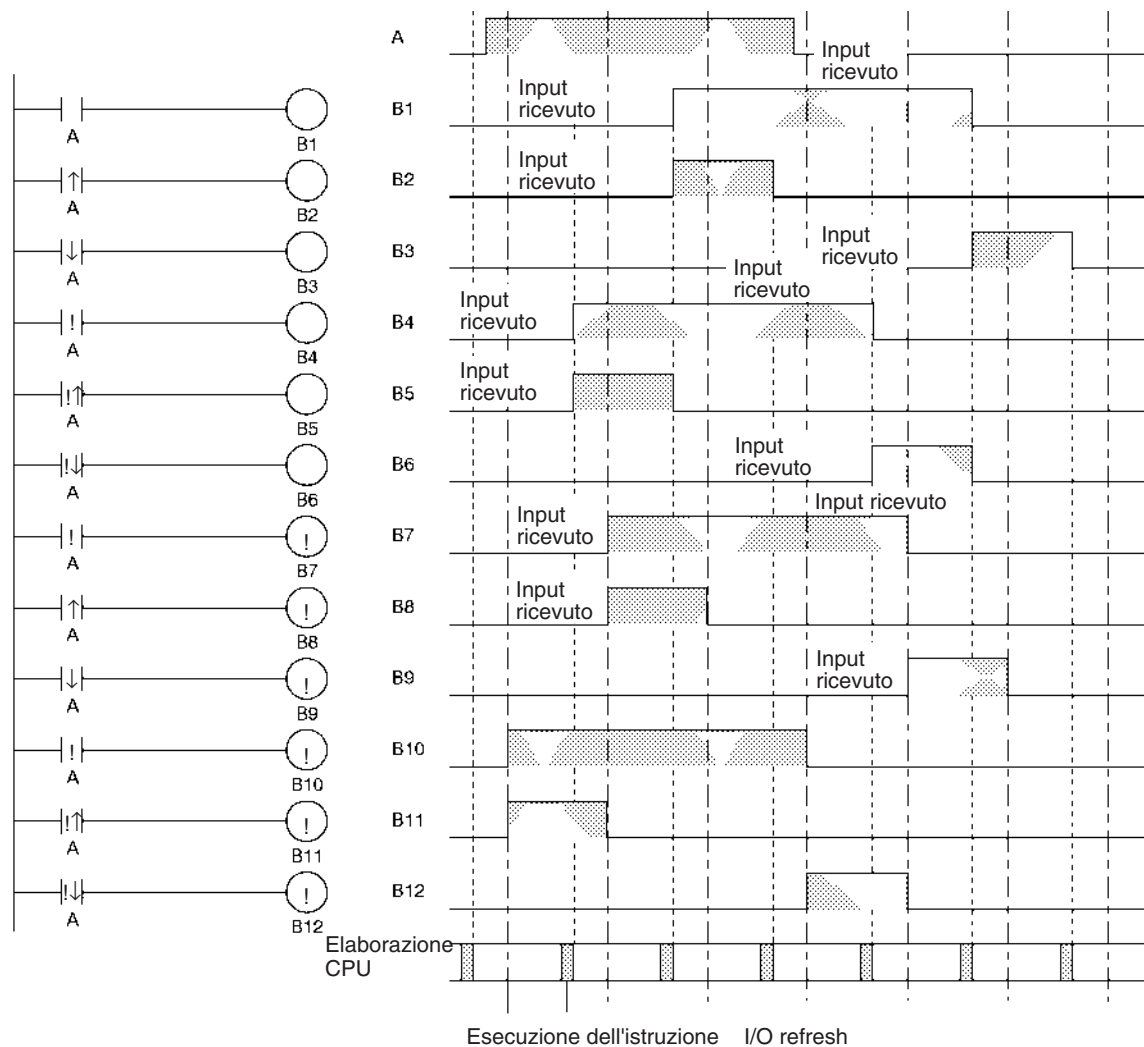
Oltre a questo I/O refresh, un'istruzione di aggiornamento immediato scambia i dati con il modulo di I/O per i canali utilizzati dall'istruzione. Oltre al bit specificato, un'istruzione con refresh immediato aggiorna contemporaneamente otto bit (gli otto bit più a sinistra o più a destra).

Non è possibile utilizzare le istruzioni di aggiornamento immediato per moduli su sistemi slave.

| Variazione di istruzione | Codice mnemonico | Funzione | I/O refresh |
|---|--|---|---|
| Normale | LD, AND, OR, LD NOT, AND NOT, OR NOT | Lo stato ON/OFF del bit specificato è assunto dal modulo con aggiornamento ciclico e viene ripreso nella successiva esecuzione dell'istruzione. | Aggiornamento ciclico |
| | OUT, OUT NOT | Dopo l'esecuzione dell'istruzione, lo stato ON/OFF del bit specificato viene inviato in uscita al successivo aggiornamento ciclico. | |
| Differenziata sul fronte di salita | @LD, @AND, @OR | L'istruzione viene eseguita una sola volta quando lo stato del bit specificato passa da OFF a ON e quest'ultima impostazione viene mantenuta per un ciclo. | |
| Differenziata sul fronte di discesa | %LD, %AND, %OR | L'istruzione viene eseguita una sola volta quando lo stato del bit specificato passa da ON a OFF e quest'ultima impostazione viene mantenuta per un ciclo. | |
| Aggiornamento immediato | !LD, !AND, !OR, !LD NOT, !AND NOT, !OR NOT | I dati di ingresso per il bit specificato sono assunti dal modulo e l'istruzione viene eseguita. | Prima dell'esecuzione dell'istruzione |
| | !OUT, !OUT NOT | In seguito all'esecuzione dell'istruzione, vengono estratti i dati per il bit specificato. | In seguito all'esecuzione dell'istruzione |
| Differenziata sul fronte di salita / aggiornamento immediato | !@LD, !@AND, !@OR | I dati di ingresso per il bit specificato sono aggiornati dal modulo e l'istruzione viene eseguita una sola volta quando lo stato del bit passa da OFF a ON e quest'ultima impostazione viene mantenuta per un ciclo. | Prima dell'esecuzione dell'istruzione |
| Differenziata sul fronte di discesa / aggiornamento immediato | !%LD, !%AND, !%OR | I dati di ingresso per il bit specificato sono aggiornati dal modulo e l'istruzione viene eseguita una sola volta quando lo stato del bit passa da ON a OFF e quest'ultima impostazione viene mantenuta per un ciclo. | |

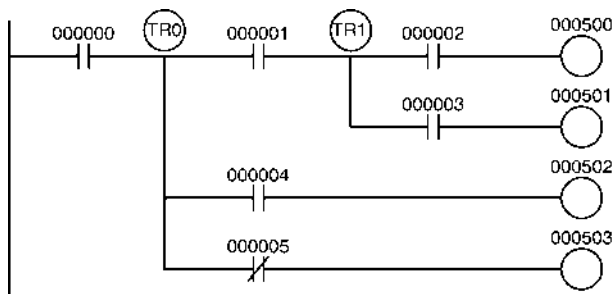
3-3-10 Tempistica operativa delle istruzioni di I/O

Il grafico qui di seguito mostra le differenze nella temporizzazione del funzionamento delle istruzioni per un programma configurato da LD e OUT.



3-3-11 Bit TR

I bit TR servono a mantenere temporaneamente lo stato ON/OFF di condizioni di esecuzione in un programma scritto utilizzando codici mnemonici. Non vengono utilizzati se si programma direttamente in linguaggio ladder perché l'elaborazione è eseguita in modo automatico dal dispositivo periferico. Nel diagramma seguente è riportata una semplice applicazione che utilizza due bit TR.

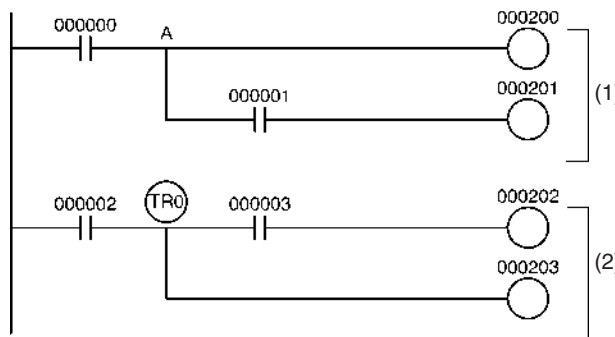


| Indirizzo | Istruzione | Operandi |
|-----------|------------|----------|
| 000000 | LD | 000000 |
| 000001 | OUT | TR0 |
| 000002 | AND | 000001 |
| 000003 | OUT | TR1 |
| 000004 | AND | 000002 |
| 000005 | OUT | 000500 |
| 000006 | LD | TR1 |
| 000007 | AND | 000003 |
| 000008 | OUT | 000501 |
| 000009 | LD | TR0 |
| 000010 | AND | 000004 |
| 000011 | OUT | 000502 |
| 000012 | LD | TR0 |
| 000013 | AND NOT | 000005 |
| 000014 | OUT | 000503 |

Utilizzo da TR0 a TR15

I bit da TR0 a TR15 vengono utilizzati soltanto con le istruzioni LOAD e OUTPUT. Non vi sono restrizioni per quanto riguarda l'ordine di utilizzo degli indirizzi di bit.

Quando è possibile, è più semplice riscrivere un programma in modo tale che i bit TR non siano più necessari. Nel diagramma seguente è riportato un caso in cui un bit TR non è necessario e un altro in cui lo è.



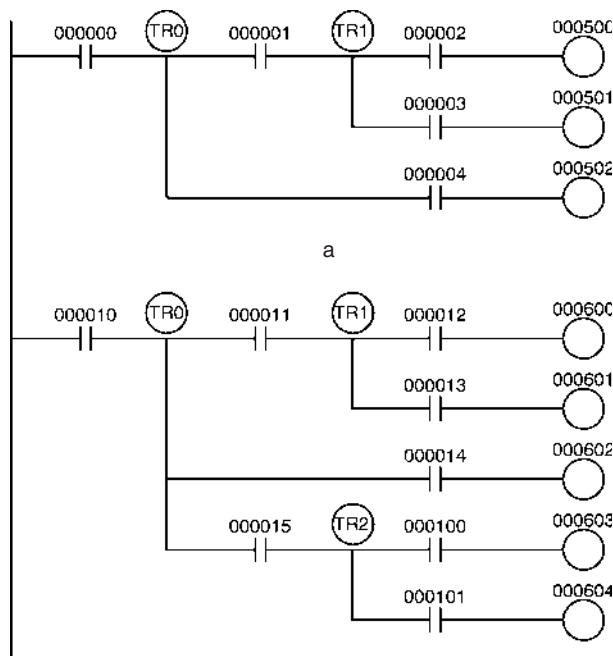
Nel blocco di istruzioni (1), lo stato ON/OFF al punto A è lo stesso dell'uscita CIO 00200, per cui è consentito codificare AND 000001 e OUT 000201 senza un bit TR. Nel blocco di istruzioni (2), lo stato del punto di diramazione e quello dell'uscita CIO 000202 non sono necessariamente uguali e quindi è necessario utilizzare un bit TR. In questo caso, si potrebbe ridurre il numero di step del programma utilizzando il blocco di istruzioni (1) al posto del blocco di istruzioni (2).

Da TR0 a TR15: considerazioni

I bit TR sono utilizzati soltanto per mantenere (da OUT TR0 a TR15) e ripristinare (da LD TR0 a TR15) lo stato ON/OFF dei punti di diramazione nei programmi con numerose diramazioni di uscita. Pertanto sono diversi dagli altri bit e non possono essere utilizzati con istruzioni AND od OR oppure, ancora, con istruzioni che includono NOT.

Duplicazione delle uscite da TR0 a TR15

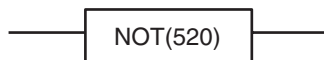
Come illustrato nel diagramma seguente, non è possibile ripetere un indirizzo di bit TR all'interno di uno stesso blocco in un programma con numerose diramazioni di uscita. È tuttavia possibile utilizzarlo nuovamente in un altro blocco.



3-3-12 NOT: NOT(520)

Scopo Inverte la condizione di esecuzione.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Inverte la condizione di esecuzione a ogni ciclo | NOT(520) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

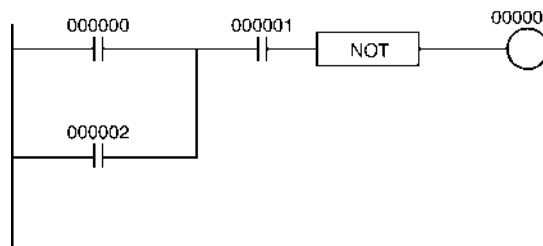
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| OK | OK | OK | OK |

Descrizione Per invertire la condizione di esecuzione, NOT(520) è collocata tra una condizione di esecuzione e un'altra istruzione.

Flag NOT(520) non influisce su alcun flag.

Avvertenze NOT(520) è un'istruzione intermedia, ossia non è possibile utilizzarla come istruzione sul lato destro. Dopo NOT(520) assicurarsi di programmare un'istruzione sul lato destro.

Esempio Nell'esempio seguente NOT(520) inverte la condizione di esecuzione.



La tabella che segue illustra il funzionamento di questa sezione del programma.

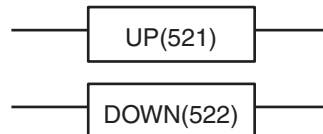
| Stato del bit di ingresso | | | Stato del bit di uscita |
|---------------------------|------------|------------|-------------------------|
| CIO 000000 | CIO 000001 | CIO 000002 | CIO 000003 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 |

3-3-13 CONDITION ON/OFF UP(521) e DOWN(522)

Scopo

UP(521) imposta la condizione di esecuzione dell'istruzione successiva su ON per un ciclo quando lo stato della condizione di esecuzione che riceve passa da OFF a ON. DOWN(522) imposta la condizione di esecuzione dell'istruzione successiva su ON per un ciclo quando lo stato della condizione di esecuzione che riceve passa da ON a OFF.

Simboli programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di salita | UP(521) |
| Aggiornamento immediato | | Non supportata |

| | | |
|--------------------------------|---|----------------|
| Variazioni | Crea una condizione di esecuzione con stato ON una sola volta per differenziazione sul fronte di discesa | UP(522) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Descrizione

Per trasformare la condizione di esecuzione in una condizione con differenziazione sul fronte di salita, UP(521) è collocata tra una condizione di esecuzione e un'altra istruzione. UP(521) fa in modo che l'istruzione di collegamento venga eseguita solo una volta quando la condizione di esecuzione passa dallo stato OFF allo stato ON.

Per trasformare la condizione di esecuzione in una condizione con differenziazione sul fronte di discesa, DOWN(522) è collocata tra una condizione di esecuzione e un'altra istruzione. DOWN(522) fa in modo che l'istruzione di collegamento venga eseguita solo una volta quando la condizione di esecuzione passa dallo stato ON allo stato OFF.

Allo stesso scopo è possibile utilizzare anche le istruzioni DIFU(013) e DIFD(014), che tuttavia necessitano di bit di lavoro. UP(521) e DOWN(522) semplificano la programmazione riducendo il numero dei bit di lavoro e degli indirizzi del programma necessari.

Flag

UP(521) e DOWN(522) non influiscono su alcun flag.

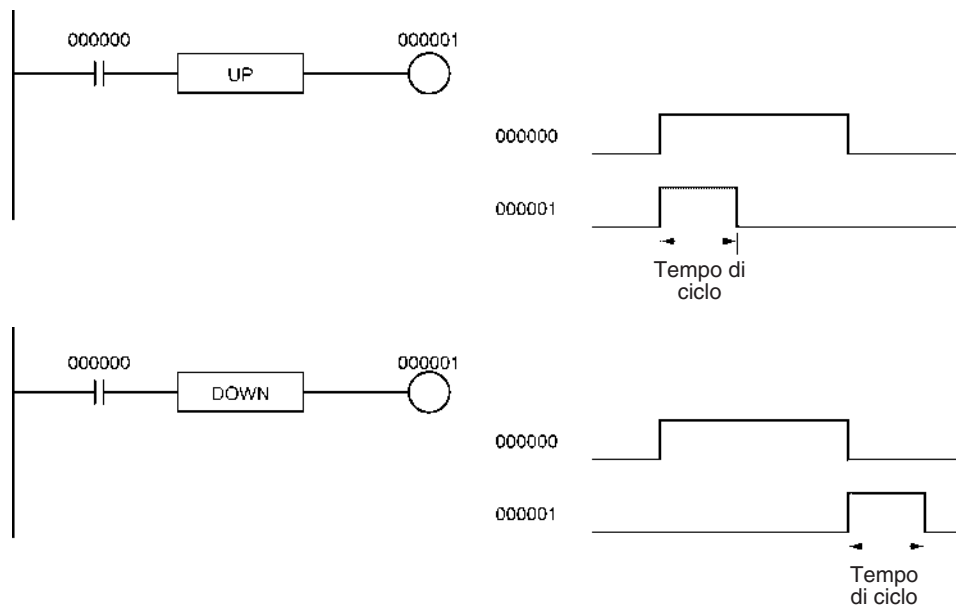
Avvertenze

UP(521) e DOWN(522) sono istruzioni intermedie, ossia non è possibile utilizzarle come istruzioni sul lato destro. Dopo UP(521) o DOWN(522) assicurarsi di programmare un'istruzione sul lato destro.

Il funzionamento di UP(521) e di DOWN(522) dipende dalla condizione di esecuzione dell'istruzione e dalla condizione di esecuzione della sezione del programma quando è programmato in una sezione del programma interbloccata o saltata oppure in una subroutine. Per ulteriori informazioni, fare riferimento alle sezioni 3-5-4 INTERLOCK e INTERLOCK CLEAR: IL(002) e ILC(003), 3-5-6 JUMP e JUMP END: JMP(004) e JME(005) e 3-20 Istruzioni di controllo degli interrupt.

Esempi

Nell'esempio che segue, quando lo stato di CIO 000000 passa da OFF a ON, lo stato di CIO 000001 viene impostato su ON per un solo ciclo.



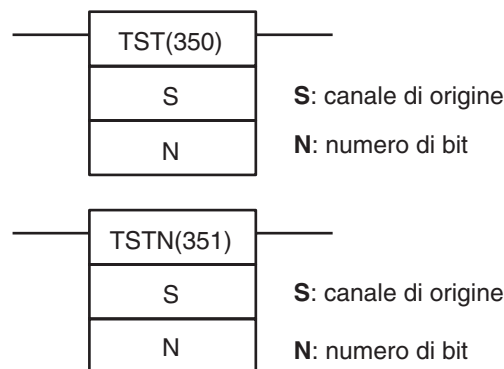
3-3-14 BIT TEST: TST(350) e TSTN(351)

Scopo

LD TST(350), AND TST(350) e OR TST(350) vengono utilizzate nel programma analogamente a LD, AND e OR. La condizione di esecuzione viene attivata quando il bit specificato nel canale specificato viene impostato su ON e disattivata quando il bit è impostato su OFF.

LD TSTN(351), AND TSTN(351) e OR TSTN(351) vengono utilizzate nel programma analogamente a LD NOT, AND NOT e OR NOT. La condizione di esecuzione viene disattivata quando il bit specificato nel canale specificato viene impostato su ON e attivata quando il bit è impostato su OFF.

Simboli programmazione ladder



Variazioni

| | | |
|-------------------|--------------------------------|----------------|
| Variazioni | Eseguita a ogni ciclo | TST(350) |
| | Aggiornamento immediato | Non supportata |

| | | |
|-------------------|--------------------------------|----------------|
| Variazioni | Eseguita a ogni ciclo | TSTN(351) |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero di bit

Il numero di bit deve essere compreso tra 0000 e 000F in formato esadecimale o tra &0000 e &0015 in formato decimale. Quando viene specificato un indirizzo di canale, l'unico bit valido tra quelli del canale è il bit più a destra (da 0 a F esadecimale).

Caratteristiche operando

| Area | S | N |
|--|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit ritentivi | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Da #0000 a #000F (in formato binario) o da &0 a &15 |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

LD TST(350), AND TST(350) e OR TST(350) possono essere utilizzate nel programma analogamente a LD, AND e OR. La condizione di esecuzione viene attivata quando il bit specificato nel canale specificato è impostato su ON e disattivata quando il bit è impostato su OFF. A differenza di LD, AND e OR, nelle aree DM ed EM è possibile utilizzare i bit come operandi in TST(350).

LD TSTN(351), AND TSTN(351) e OR TSTN(351) possono essere utilizzate nel programma analogamente a LD NOT, AND NOT e OR NOT. La condizione di esecuzione viene disattivata quando il bit specificato nel canale specificato è impostato su ON e attivata quando il bit è impostato su OFF. A differenza di LD NOT, AND NOT e OR NOT, nelle aree DM ed EM è possibile utilizzare i bit come operandi in TSTN(351).

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF o non modificato (vedere nota) |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

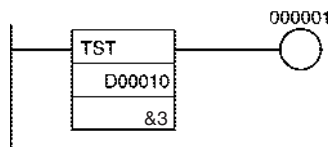
Avvertenze

TST(350) e TSTN(351) sono istruzioni intermedie, ossia non è possibile utilizzarle come istruzioni sul lato destro. Dopo TST(350) o TSTN(351) assicurarsi di programmare un'istruzione sul lato destro.

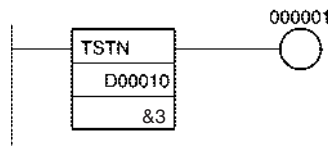
Esempi

LD TST(350) e LD TSTN(351)

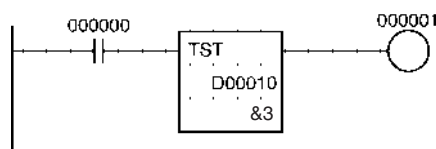
Nell'esempio che segue, CIO 000001 viene impostato su ON quando il bit 3 di D00010 è ON.



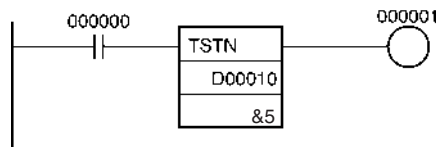
Nell'esempio che segue, CIO 000001 viene impostato su ON quando il bit 3 di D00010 è OFF.

**AND TST(350) e AND TSTN(351)**

Nell'esempio che segue, CIO 000001 viene impostato su ON quando CIO 000000 e il bit 3 di D00010 sono entrambi ON.

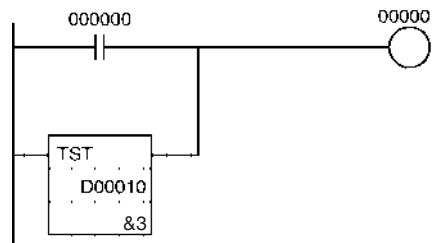


Nell'esempio che segue, CIO 000001 viene impostato su ON quando CIO 000000 è ON e il bit 5 di D00010 è OFF.

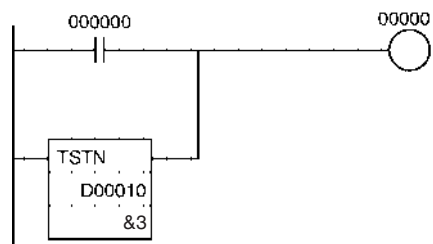


OR TST(350) e OR TSTN(351)

Nell'esempio che segue, CIO 000001 viene impostato su ON quando CIO 000000 o il bit 3 di D00010 sono ON.



Nell'esempio che segue, CIO 000001 viene impostato su ON quando CIO 000000 è ON o il bit 3 di D00010 è OFF.



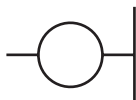
3-4 Istruzioni di uscita sequenza

3-4-1 OUTPUT: OUT

Scopo

Invia in uscita il risultato (condizione di esecuzione) dell'elaborazione logica al bit specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | OUT |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota) | | !OUT |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Non consentita | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando OUT |
|-------------------------|----------------------------|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A44800 ad A95915 |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area TR | Da TR0 a TR15 |

| Area | Bit operando OUT |
|--|---|
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a ,IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

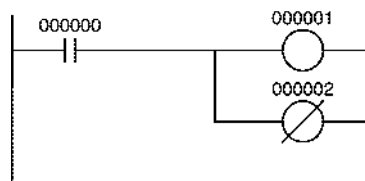
In assenza di aggiornamento immediato, lo stato della condizione di esecuzione (flusso logico di alimentazione) viene scritto nel bit specificato nella memoria I/O. In assenza di aggiornamento immediato, lo stato della condizione di esecuzione (flusso logico di alimentazione) viene scritto sia nel terminale di uscita del modulo di uscita di base che nel bit di uscita nella memoria I/O.

Flag

Questa istruzione non influisce su alcun flag.

Avvertenze

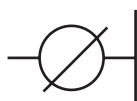
Per OUT e OUT NOT è possibile specificare l'aggiornamento immediato (!). Un'istruzione con refresh immediato aggiorna lo stato del terminal di uscita subito dopo l'esecuzione dell'istruzione per moduli di uscita di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H multipunto gruppo 2) e contemporaneamente scrive lo stato della condizione di esecuzione (flusso logico di alimentazione) nel bit di uscita specificato nella memoria I/O.

Esempio

| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| OUT | 000001 |
| OUT NOT | 000002 |

3-4-2 OUTPUT NOT: OUT NOT**Scopo**

Inverte il risultato (condizione di esecuzione) dell'elaborazione logica e lo invia in uscita al bit specificato.

Simbolo programmazione ladder

Variazioni

| | | |
|---------------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | OUT NOT |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota) | | !OUT NOT |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Caratteristiche operando

| Area | Bit operando OUT |
|--|---|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A44800 ad A95915 |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area TR | Da TR0 a TR15 |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a ,IR15 Da ,IR0(++) a ,IR15(++) Da ,-(-)IR0 a ,-(-)IR15 |

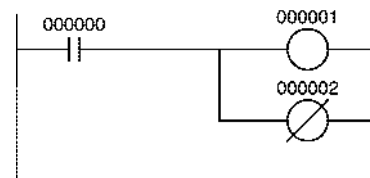
Descrizione

In assenza di aggiornamento immediato, lo stato della condizione di esecuzione (flusso logico di alimentazione) viene invertito e scritto in un bit specificato nella memoria I/O. In assenza di aggiornamento immediato, lo stato della condizione di esecuzione (flusso logico di alimentazione) viene invertito e inoltre scritto nel terminale di uscita del modulo di uscita di base e nel bit di uscita nella memoria I/O.

Flag

Questa istruzione non influisce su alcun flag.

Esempio



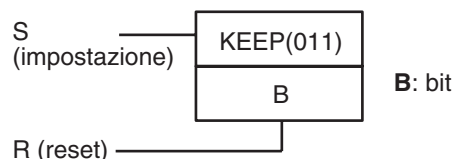
| Istruzione | Operando |
|------------|----------|
| LD | 000000 |
| OUT | 000001 |
| OUT NOT | 000002 |

3-4-3 KEEP: KEEP(011)

Scopo

Funziona come un relé di blocco.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | KEEP(011) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato (vedere nota) | !KEEP(011) |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

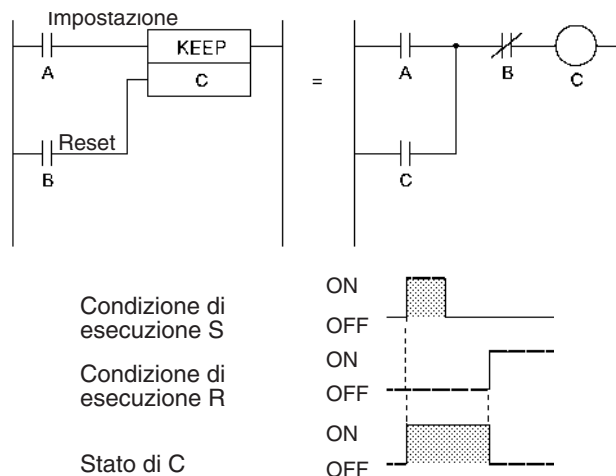
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Caratteristiche operando

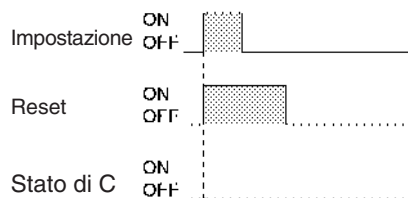
| Area | B |
|--|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A44800 ad A95915 |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

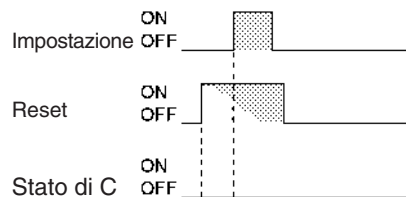
Quando S è attivato, il bit specificato passa a ON e mantiene questo stato fino al reset, indipendentemente dal fatto che S rimanga attivo o venga disattivato. Quando R viene attivato, il bit specificato passerà allo stato OFF. La relazione tra le condizioni di esecuzione e lo stato del bit KEEP(011) viene illustrata qui di seguito.



Se S ed R sono ON contemporaneamente, l'ingresso di reset prende la precedenza.

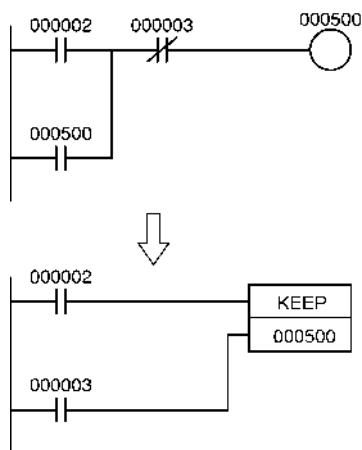


L'ingresso di impostazione (S) non può essere ricevuto se R è ON.

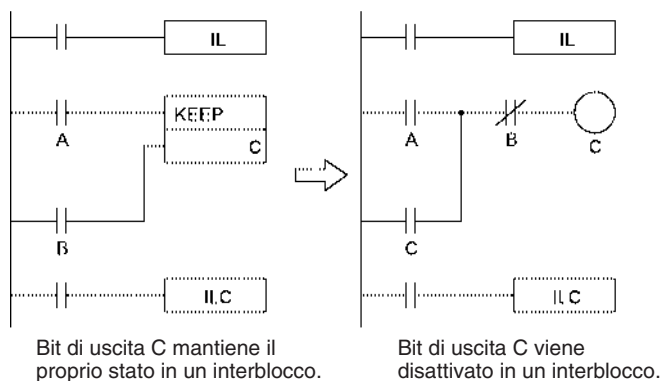


KEEP(011) presenta una variazione di aggiornamento immediato (!KEEP(011)). Nel caso in cui in un'istruzione !KEEP(011) sia stata specificata un'uscita esterna per B, qualsiasi modifica di B verrà aggiornata all'esecuzione di !KEEP(011) e riprodotta immediatamente nel bit di uscita. (Le modifiche non sono aggiornate immediatamente se il bit è allocato in un modulo di I/O ad alta densità gruppo 2, in un modulo di I/O speciale ad alta densità o in un modulo installato su un sistema slave di I/O remoti SYSMAC BUS).

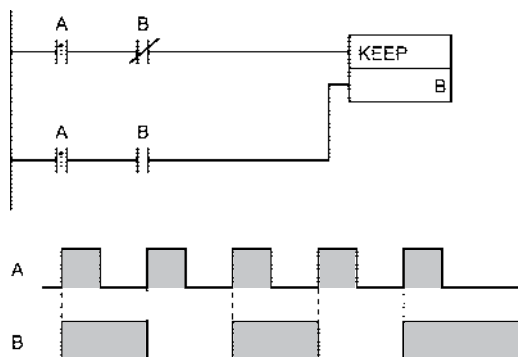
KEEP(011) funziona come il bit a autoritenuta ma richiede un'istruzione in più rispetto a un bit a autoritenuta programmato con KEEP(011).



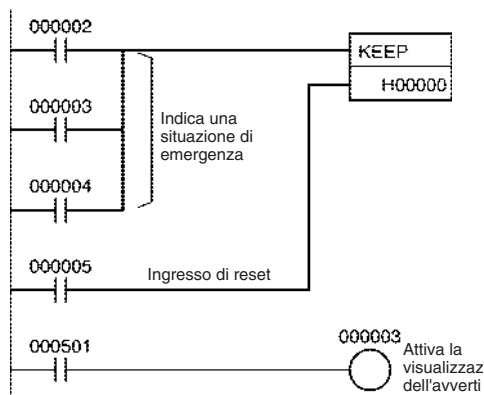
A differenza dei bit a autoritenuta programmati senza KEEP(011), i bit a autoritenuta programmati con KEEP(011) manterranno lo stato anche in una sezione del programma interbloccata.



Come illustrato qui di seguito, è possibile utilizzare KEEP(011) per creare flip-flop.



Se un bit di ritenività viene utilizzato per B, lo stato del bit sarà mantenuto anche durante una caduta di tensione. È quindi possibile utilizzare KEEP(011) per programmare bit che mantengano il proprio stato al riavvio del PLC in seguito a una caduta di tensione. Qui di seguito è riportato un esempio utile per visualizzare un messaggio di avvertimento in seguito a un arresto di sistema per un'emergenza.



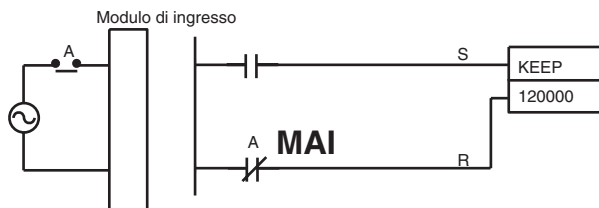
Nel caso di caduta di tensione è possibile mantenere lo stato dei bit dell'area di I/O attivando il bit di ritenività IOM e scegliendo l'opzione IOM Hold Bit Hold nell'impostazione del PLC. In questa ipotesi, i bit dell'area di I/O utilizzati in KEEP(011) manterranno il proprio stato a un riavvio del PLC in seguito a una caduta di tensione, esattamente come i bit di ritenività. Dopo avere apportato le modifiche al PLC assicurarsi di riavviarlo, altrimenti le nuove impostazioni resteranno inutilizzate.

Flag

KEEP(011) non influisce su alcun flag.

Avvertenze

Se l'alimentazione del dispositivo di ingresso è in corrente alternata, con l'istruzione KEEP(011) è opportuno non specificare mai un bit di ingresso in una condizione normalmente chiusa per il ripristino (R). Il ritardo nell'arresto dell'alimentatore c.c. del PLC (dovuto all'alimentazione c.a. del dispositivo di ingresso) può provocare il ripristino del bit operando di KEEP(011). Questo caso è illustrato qui di seguito.



Nei diagrammi ladder e nel codice mnemonico gli operandi per KEEP(011) vengono immessi in ordine differente.

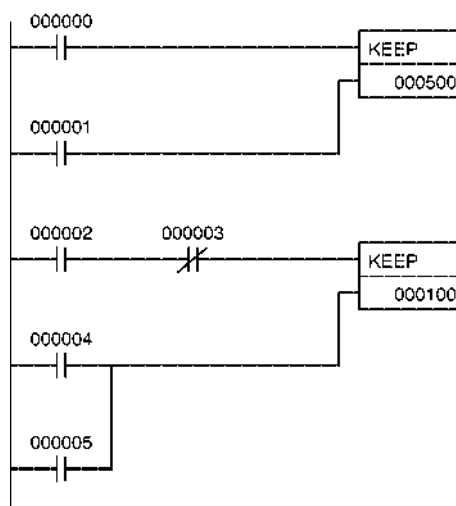
Ordine del diagramma ladder: Ingresso di impostazione → KEEP(011) → Ingresso di reset

Ordine del codice mnemonico: Ingresso di impostazione → Ingresso di reset → KEEP(011)

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, CIO 00500 viene attivato. CIO 00500 resta impostato su ON fino all'attivazione di CIO 000001.

Nell'esempio seguente, quando CIO 000002 è impostato su ON e CIO 000003 su OFF, CIO 00100 viene attivato. CIO 00100 resta impostato su ON fino all'attivazione di CIO 000004 o di CIO 000005.

**Codice**

| Indirizzo | Istruzione | Operando |
|-----------|------------|----------|
| 000100 | LD | 000000 |
| 000101 | LD | 000001 |
| 000102 | KEEP(011) | 000500 |
| 000103 | LD | 000002 |
| 000104 | AND NOT | 000003 |
| 000105 | LD | 000004 |
| 000106 | OR | 000005 |
| 000107 | KEEP(011) | 000100 |

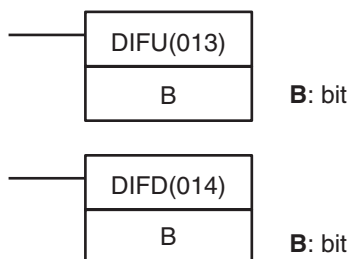
Nota L'ordine di immissione di KEEP(011) dipende dal tipo di criterio prescelto: ladder o forma mnemonica. Nel primo caso, specificare l'ingresso di impostazione, KEEP(011) e infine l'ingresso di reset. Nel secondo, specificare l'ingresso di impostazione, l'ingresso di reset e infine KEEP(011).

3-4-4 DIFFERENTIATE UP/DOWN: DIFU(013) e DIFD(014)

Scopo

DIFU(013) imposta su ON il bit specificato per un ciclo quando la condizione di esecuzione passa da OFF a ON (fronte di salita).

DIFD(014) imposta su ON il bit specificato per un ciclo quando la condizione di esecuzione passa da ON a OFF (fronte di discesa).

Simboli programmazione ladder
**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | Non supportata |
|---------------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | DIFU(013) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota) | | !DIFU(013) |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

| | | |
|---------------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di salita | DIFD(014) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota) | | !DIFD(014) |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

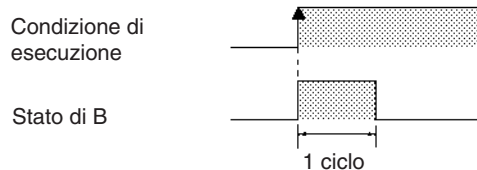
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Caratteristiche operando

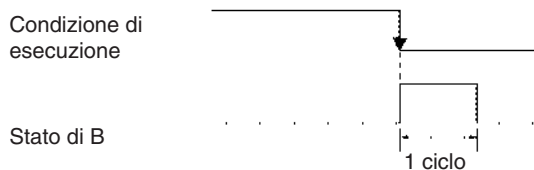
| Area | B |
|--|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A44800 ad A95915 |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

Quando la condizione di esecuzione passa da OFF a ON, DIFU(013) imposta B su ON. Quando DIFU(013) viene raggiunta al ciclo successivo, B passa a OFF.



Quando la condizione di esecuzione passa da ON a OFF, DIFU(014) imposta B su ON. Quando DIFU(014) viene raggiunta al ciclo successivo, B passa a OFF.



DIFU(013) e DIFD(014) presentano variazioni di aggiornamento immediato (!DIFU(013) e !DIFD(014)). Nel caso in cui in una di queste istruzioni sia stata specificata un'uscita esterna per B, qualsiasi modifica di B verrà aggiornata all'esecuzione dell'istruzione e riprodotta immediatamente nel bit di uscita. (Le modifiche non sono aggiornate immediatamente se il bit è allocato in un modulo di I/O ad alta densità gruppo 2, in un modulo di I/O speciale ad alta densità o in un modulo installato su un sistema slave di I/O remoti SYSMAC BUS).

È possibile utilizzare UP(521) e DOWN(522) per eseguire un'istruzione per un solo ciclo quando la condizione di esecuzione passa da OFF → ON o da ON → OFF. Per ulteriori informazioni, fare riferimento alla sezione 3-3-13 *CONDITION ON/OFF UP(521) e DOWN(522)*.

Flag

DIFU(013) e DIFD(014) non influiscono su alcun flag.

Avvertenze

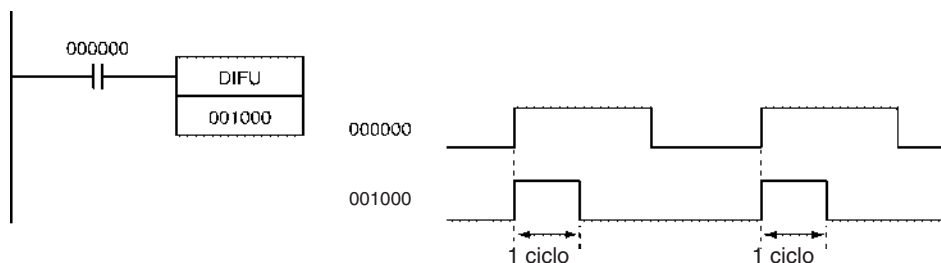
Il funzionamento di DIFU(013) e DIFD(014) dipende dalla condizione di esecuzione dell'istruzione stessa e dalla condizione di esecuzione della sezione del programma quando è programmato in una sezione del programma interbloccata o saltata oppure in una subroutine. Per ulteriori informazioni, fare riferimento alle sezioni 3-5-4 *INTERLOCK e INTERLOCK CLEAR: IL(002) e ILC(003)*, 3-5-6 *JUMP e JUMP END: JMP(004) e JME(005)* e 3-20 *Istruzioni di controllo degli interrupt*.

Se DIFU(013) è utilizzata in un ciclo FOR-NEXT che si ripete a sua volta in un ciclo, il bit controllato sarà sempre impostato su ON o sempre su OFF all'interno di quel ciclo.

Esempi

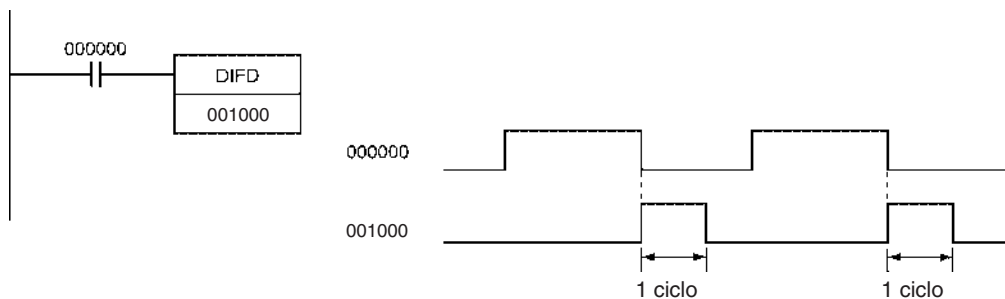
Funzionamento di DIFU(013)

Nell'esempio che segue, quando CIO 000000 passa da OFF a ON, CIO 001000 viene impostato su ON per un solo ciclo.



Funzionamento di DIFD(014)

Nell'esempio che segue, quando CIO 000000 passa da ON a OFF, CIO 001000 viene impostato su ON per un solo ciclo.



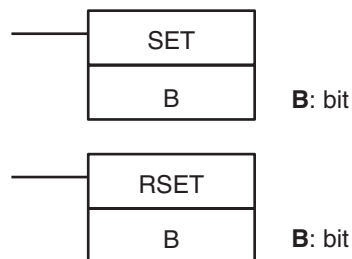
3-4-5 SET e RESET SET e RSET

Scopo

SET imposta il bit dell'operando su ON quando la condizione di esecuzione è ON.

RSET imposta il bit dell'operando su OFF quando la condizione di esecuzione è ON.

Simboli programmazione ladder



Variazioni

| | | |
|---------------------------------------|--|-------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SET |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SET |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | %SET |
| Aggiornamento immediato (vedere nota) | | !SET |
| Variazioni combinate | Eseguita una sola volta con immediato aggiornamento del bit per differenziazione sul fronte di salita (vedere nota) | !@SET |
| | Eseguita una sola volta con immediato aggiornamento del bit per differenziazione sul fronte di discesa (vedere nota) | !%SET |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

| | | |
|---------------------------------------|---|--------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RSET |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RSET |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | %RSET |
| Aggiornamento immediato (vedere nota) | | !RSET |
| Variazioni combinate | Aggiornamento immediato una sola volta per differenziazione sul fronte di salita (vedere nota) | !@RSET |
| | Aggiornamento immediato una sola volta per differenziazione sul fronte di discesa (vedere nota) | !%RSET |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

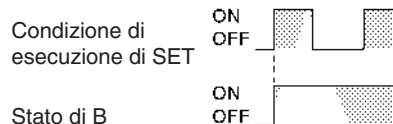
Caratteristiche operando

| Area | B |
|-------------------------|----------------------------|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit ritentivi | Da H00000 a H51115 |
| Area bit ausiliaria | Da A44800 ad A95915 |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |

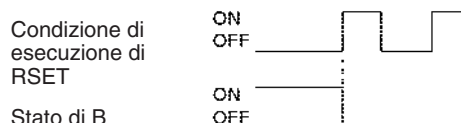
| Area | B |
|--|--|
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

SET imposta il bit operando su ON quando la condizione di esecuzione è ON; quando la condizione è OFF, SET non influisce sullo stato del bit. Utilizzare RSET per impostare su OFF un bit precedentemente impostato su ON con SET.



RSET imposta il bit operando su OFF quando la condizione di esecuzione è ON; quando la condizione è OFF, RSET non influisce sullo stato del bit. Utilizzare SET per impostare su ON un bit precedentemente impostato su OFF con RSET.



SET e RSET presentano variazioni di aggiornamento immediato (!SET e !RSET). Nel caso in cui in una di queste istruzioni sia stata specificata un'uscita esterna per B, qualsiasi modifica di B verrà aggiornata all'esecuzione dell'istruzione e riprodotta immediatamente nel bit di uscita. (Le modifiche non sono aggiornate immediatamente se il bit è allocato in un modulo di I/O ad alta densità gruppo 2, in un modulo di I/O speciale ad alta densità o in un modulo installato su un sistema slave di I/O remoti SYSMAC BUS).

È necessario programmare gli ingressi di impostazione e di reset per un'istruzione KEEP(011) insieme all'istruzione stessa; le istruzioni SET e RSET, tuttavia, possono essere programmate in modo del tutto indipendente. Inoltre, è possibile utilizzare lo stesso bit come operando in tutte le istruzioni SET o RSET desiderate.

Flag

SET e RSET non influiscono su alcun flag.

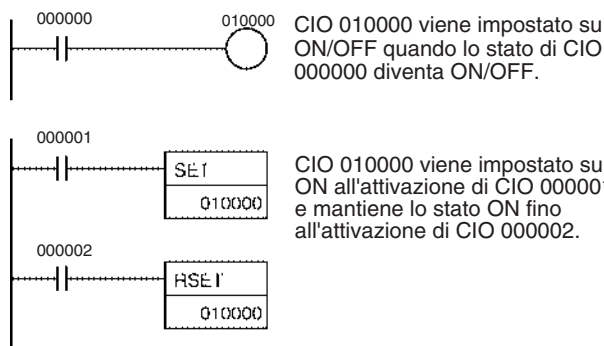
Avvertenze

Non è possibile utilizzare SET e RSET per impostare e ripristinare temporizzatori o contatori.

Quando SET o RSET sono programmate tra IL(002) e ILC(003) o tra JMP(004) e JME(005), lo stato del bit specificato non subirà modifiche se la sezione del programma è interbloccata o di salto.

Esempio**Differenze tra OUT/OUT NOT e SET/RSET**

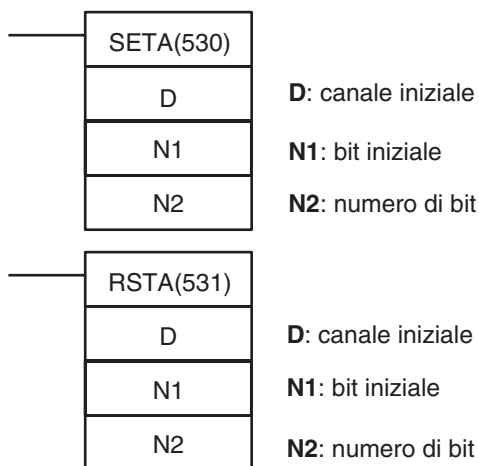
Il funzionamento di SET differisce da quello di OUT in quanto l'istruzione OUT imposta il bit operando su OFF quando la propria condizione di esecuzione è OFF. Analogamente RSET si differenzia da OUT NOT poiché quest'ultima imposta il bit operando su ON quando la propria condizione di esecuzione è OFF.



3-4-6 MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531)

Scopo SETA(530) imposta su ON il numero di bit consecutivi specificato.
RSTA(531) imposta su OFF il numero di bit consecutivi specificato.

Simboli programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SETA(530) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SETA(530) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RSTA(531) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RSTA(531) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

D: canale iniziale

Specifica il primo canale in cui i bit saranno impostati su ON o su OFF.

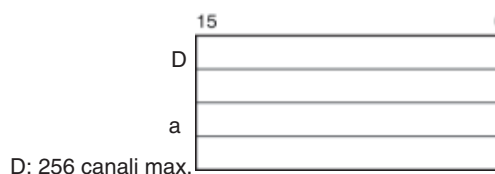
N1: bit iniziale

Specifica il primo bit che sarà impostato su ON o su OFF. N1 deve essere compreso tra #0000 e #000F o da &0 a &15.

N2: numero di bit

Specifica il numero di bit che saranno impostati su ON o su OFF. N2 deve essere compreso tra #0000 e #FFFF o da &0 a &65535.

Nota I bit che vengono attivati o disattivati devono appartenere alla stessa area dati. (L'intervallo di canali si attesta approssimativamente tra D a D+N2÷16.)



Caratteristiche operando

| Area | D | N1 | N2 |
|--|--|---|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit ritentivi | Da H000 a H511 | | |
| Area bit ausiliaria | Da A448 ad A959 | Da A000 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a #000F (in formato binario) o da &0 a &15 | Da #0000 a #FFFF (in formato binario) o da &0 a &65535 |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

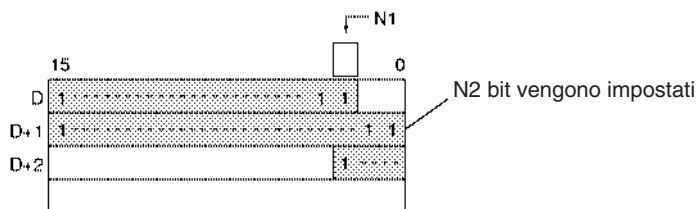
Descrizione

Il funzionamento di SETA(530) e quello di RSTA(531) sono illustrati singolarmente qui di seguito.

Funzionamento di SETA(530)

SETA(530) imposta su ON gli N2 bit, iniziando da bit N1 di D e proseguendo a sinistra, ossia verso i bit più significativi. Tutti gli altri bit vengono lasciati inalterati. (Se N2 è impostato su 0, non viene apportata alcuna modifica).

È possibile disattivare i bit impostati su ON da SETA(530) utilizzando RSTA(531) o qualsiasi altra istruzione.

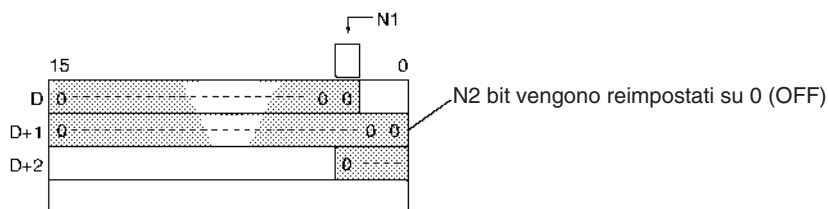


SETA(530) può essere utilizzata per impostare su ON i bit di aree dati in cui di norma accedono soltanto i canali, come le aree DM ed EM.

Funzionamento di RSTA(531)

RSTA(531) imposta su OFF gli N2 bit, iniziando da bit N1 di D e proseguendo a sinistra, ossia verso i bit più significativi. Tutti gli altri bit vengono lasciati inalterati. (Se N2 è impostato su 0, non viene apportata alcuna modifica).

È possibile attivare i bit impostati su OFF da RSTA(531) utilizzando SETA(530) o qualsiasi altra istruzione.



RSTA(531) può essere utilizzata per impostare su OFF i bit di aree dati in cui di norma accedono soltanto i canali, come le aree DM ed EM.

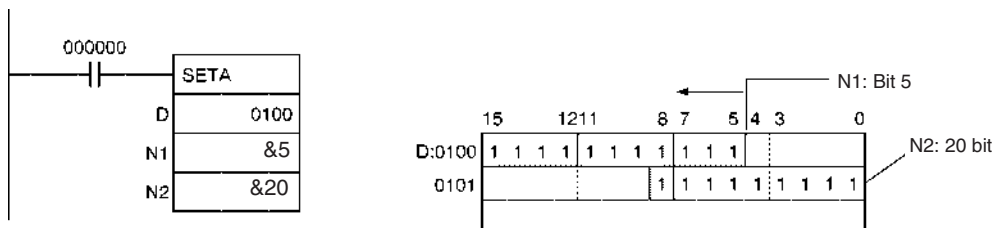
Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se N1 non rientra nell'intervallo specificato 0000 - 000F. OFF in tutti gli altri casi. |

Esempi

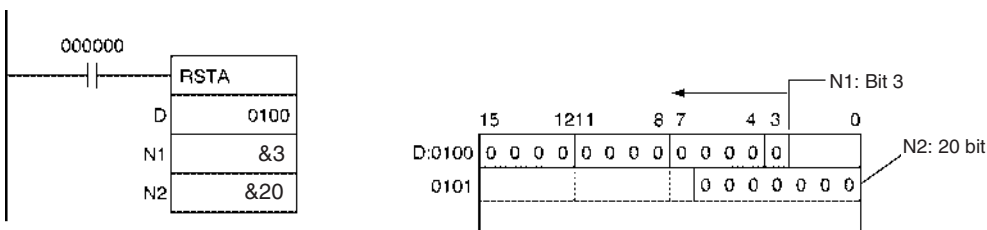
Esempio per SETA(530)

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i 20 bit (0014 esadecimale) a partire da bit 5 di CIO 0100 vengono a loro volta impostati su ON.



Esempio per RSTA(531)

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i 20 bit (0014 esadecimale) a partire da bit 3 di CIO 0100 vengono a loro volta impostati su OFF.



3-4-7 SINGLE BIT SET/RESET: SETB(532)/RSTB(533)

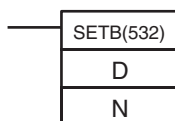
Scopo

SETA(532) imposta su ON il bit specificato.

RSTA(533) imposta su OFF il bit specificato.

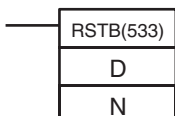
Queste istruzioni sono supportate solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simboli programmazione ladder



D: indirizzo del canale

N: numero di bit



D: indirizzo del canale

N: numero di bit

Variazioni

| | | |
|---------------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SETB(532) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SETB(532) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota) | | !SETB(532) |
| Variazioni combinate | Eseguita una sola volta con immediato aggiornamento del bit per differenziazione sul fronte di salita (vedere nota) | !@SETB(532) |
| | Eseguita una sola volta con immediato aggiornamento del bit per differenziazione sul fronte di discesa | Non supportata |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

| | | |
|---------------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RSTB(533) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RSTB(533) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota) | | !RSTB(533) |
| Variazioni combinate | Eseguita una sola volta con immediato aggiornamento del bit per differenziazione sul fronte di salita (vedere nota) | !@RSTB(533) |
| | Eseguita una sola volta con immediato aggiornamento del bit per differenziazione sul fronte di discesa | Non supportata |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

D: indirizzo del canale

Specifica il canale in cui il bit sarà impostato su ON o su OFF.

N: bit iniziale

Specifica il bit che sarà impostato su ON o su OFF. N deve essere compreso tra #0000 e #000F o da &0 a &15.

Caratteristiche operando

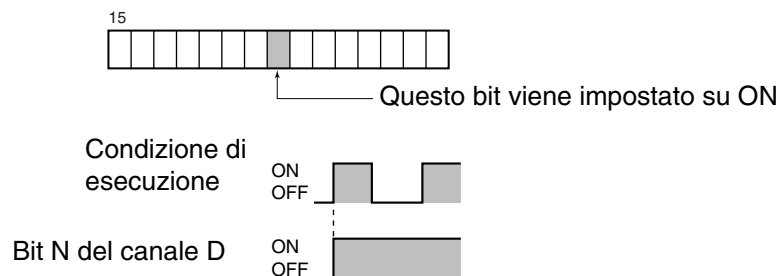
| Area | D | N |
|--|---|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 ad A959 | Da A000 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Da #0000 a #000F (in formato binario) o da &0 a &15 |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

Il funzionamento di SETB(532) e quello di RSTB(533) sono illustrati singolarmente qui di seguito.

Funzionamento di SETB(532)

SETB(532) imposta su ON il bit N del canale D quando lo stato della condizione di esecuzione è ON. Lo stato del bit non subisce modifiche se la condizione di esecuzione è impostata su OFF. A differenza di SET, SETB(532) può attivare un bit nell'area DM o in quella EM.

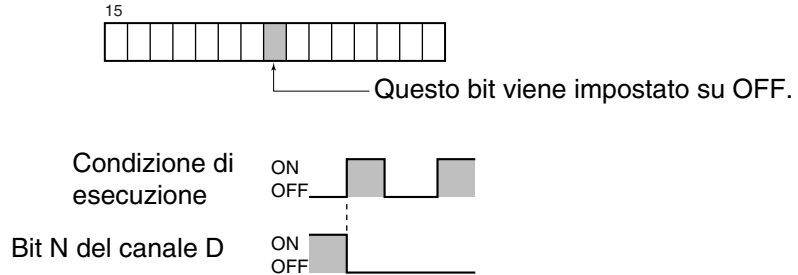


È possibile disattivare i bit impostati su ON da SETB(532) utilizzando RSTB(533) o qualsiasi altra istruzione.

SETB(532) è supportata solo dalle CPU CS1/H, CJ1/H e CJ1M.

Funzionamento di RSTB(533)

RSTB(533) imposta su OFF il bit N del canale D quando lo stato della condizione di esecuzione è ON. Lo stato del bit non subisce modifiche se la condizione di esecuzione è impostata su OFF. Specificare SETB(532) per attivare il bit specificato. A differenza di RST, RSTB(533) può disattivare un bit nell'area DM o in quella EM.



È possibile attivare i bit impostati su OFF da RSTB(533) utilizzando SETB(532) o qualsiasi altra istruzione.

RSTB(533) è supportata solo dalle CPU CS1/H, CJ1/H e CJ1M.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato 0000 - 000F (da &0 a &15). OFF in tutti gli altri casi. |

Avvertenze

Non è possibile utilizzare SETB(532) e RSTB(533) per impostare e reimpostare temporizzatori o contatori.

Quando SETB(532) o RSTB(533) sono programmate tra IL(002) e ILC(003) o tra JMP(004) e JME(005), lo stato del bit specificato non subirà modifiche se la sezione del programma è interbloccata o di salto, ossia se la condizione di interblocco o di salto sono impostate su OFF.

SETB(532) e RSTB(533) presentano variazioni di aggiornamento immediato (!SETB(532) e !RSTB(533)). Nel caso in cui un'uscita esterna sia stata specificata in una di queste istruzioni, qualsiasi modifica del bit specificato verrà aggiornata all'esecuzione dell'istruzione e riprodotta immediatamente nel bit di uscita. (Le modifiche non sono aggiornate immediatamente se il bit è allocato in un modulo di I/O ad alta densità gruppo 2, in un modulo di I/O speciale ad alta densità o in un modulo installato su un sistema slave di I/O remoti SYSMAC BUS).

Differenze tra SET/RSET e SETB(532)/RSTB(533)

Il funzionamento delle istruzioni SET e RSET è leggermente diverso da quello di SETB(532) e RSTB(533).

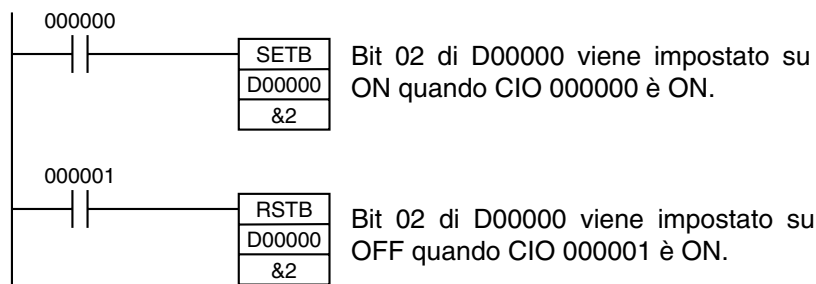
1. Queste istruzioni procedono nello stesso modo quando il bit specificato si trova nelle aree CIO, W, H o A.
2. A differenza di SET e RSET, le istruzioni SETB(532) e RSTB(533) non sono in grado di controllare i bit delle aree DM ed EM.

Differenze tra OUTB(534) e SETB(532)/RSTB(533)

Il funzionamento dell'istruzione OUTB(534) è leggermente diverso da quello di SETB(532) e RSTB(533).

1. Le istruzioni SETB(532) e RSTB(533) modificano lo stato del bit specificato soltanto se la loro condizione di esecuzione è impostata su ON. Con condizione di esecuzione impostata su OFF, queste istruzioni non influiscono sullo stato del bit specificato.

- Quando la condizione di esecuzione di OUTB(534) è ON, l'istruzione imposta su ON il bit specificato; il bit specificato viene disattivato se OUTB(534) è OFF.
- È necessario programmare gli ingressi di impostazione e di reset per un'istruzione KEEP(011) insieme all'istruzione stessa; le istruzioni SETB(532) RSTB(533), tuttavia, possono essere programmate in modo del tutto indipendente. Inoltre, è possibile utilizzare lo stesso bit come operando in tutte le istruzioni SETB(532) e RSTB(533) desiderate.



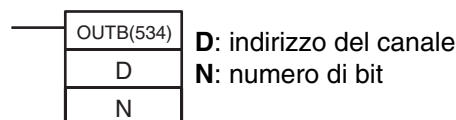
3-4-8 SINGLE BIT OUTPUT: OUTB(534)

Scopo

OUTB(534) invia in uscita lo stato della condizione di esecuzione dell'istruzione al bit specificato. A differenza di OUT, OUTB(534) è in grado di controllare un bit nell'area DM o EM.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simboli programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | OUTB(534) |
|--|--|-------------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @OUTB(534) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota) | | !OUTB(534) |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Operandi

D: indirizzo del canale

Specifica il canale che contiene il bit da controllare.

N: bit iniziale

Specifica il bit da controllare. N deve essere compreso tra #0000 e #000F o da &0 a &15.

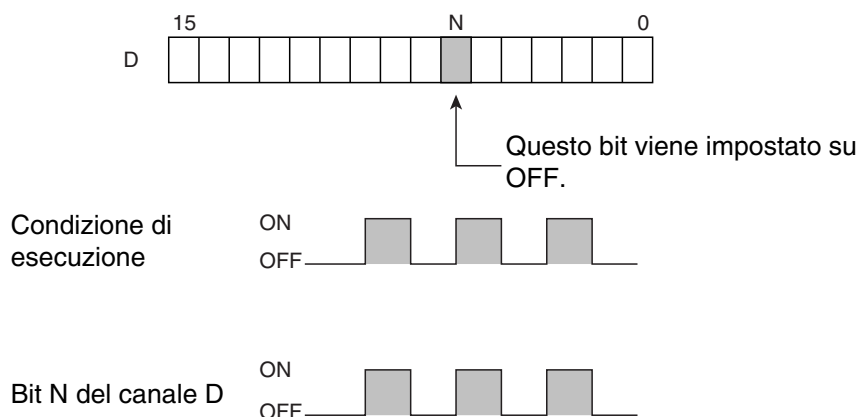
Caratteristiche operando

| Area | D | N |
|------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 ad A959 | Da A000 ad A959 |

| Area | D | N |
|--|---|---|
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Da #0000 a #000F (in formato binario) o da &0 a &15 |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0(++), a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

Quando la condizione di esecuzione è ON, OUTB(534) imposta su ON il bit N del canale D. Quando la condizione di esecuzione è OFF, OUTB(534) imposta su OFF il bit N del canale D.



Se non viene utilizzata la versione di aggiornamento immediato, lo stato della condizione di esecuzione (flusso logico di alimentazione) viene scritto nel bit specificato nella memoria I/O. Se viene utilizzata la versione di aggiornamento immediato, lo stato della condizione di esecuzione (flusso logico di alimentazione) viene scritto sia nel terminale di uscita del modulo di uscita di base che nel bit di uscita nella memoria I/O.

OUTB(534) è supportata solo dalle CPU CS1/H, CJ1/H e CJ1M.

Flag

Questa istruzione non influisce su alcun flag.

Avvertenze

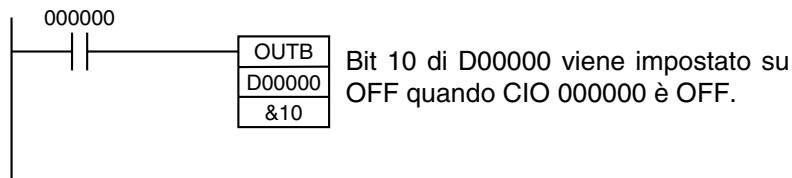
È possibile specificare l'aggiornamento immediato (!OUTB(534)). Un'istruzione con refresh immediato aggiorna lo stato del terminal di uscita subito dopo l'esecuzione dell'istruzione in un bit di uscita allocato in un modulo di uscita di base (tranne quelli sui sistemi slave o per moduli di ingresso C200H

multipunto gruppo 2) e contemporaneamente scrive lo stato della condizione di esecuzione (flusso logico di alimentazione) nel bit di uscita specificato nella memoria I/O.

Quando OUTB(534) viene programmata tra IL(002) e ILC(003), il bit specificato sarà impostato su OFF se la sezione del programma è interbloccata, come avviene nel caso di un'istruzione OUT.

Quando viene specificato un canale per il numero di bit (N), vengono utilizzati soltanto i bit da 00 a 03 di N. Ad esempio, se N include FFFA esadecimale, OUTB(534) controllerà il bit 10 del canale D.

Esempio

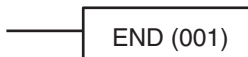


3-5 Istruzioni di controllo sequenza

3-5-1 END: END(001)

Scopo Indica la fine del programma.

Simbolo programmazione ladder



Variazioni

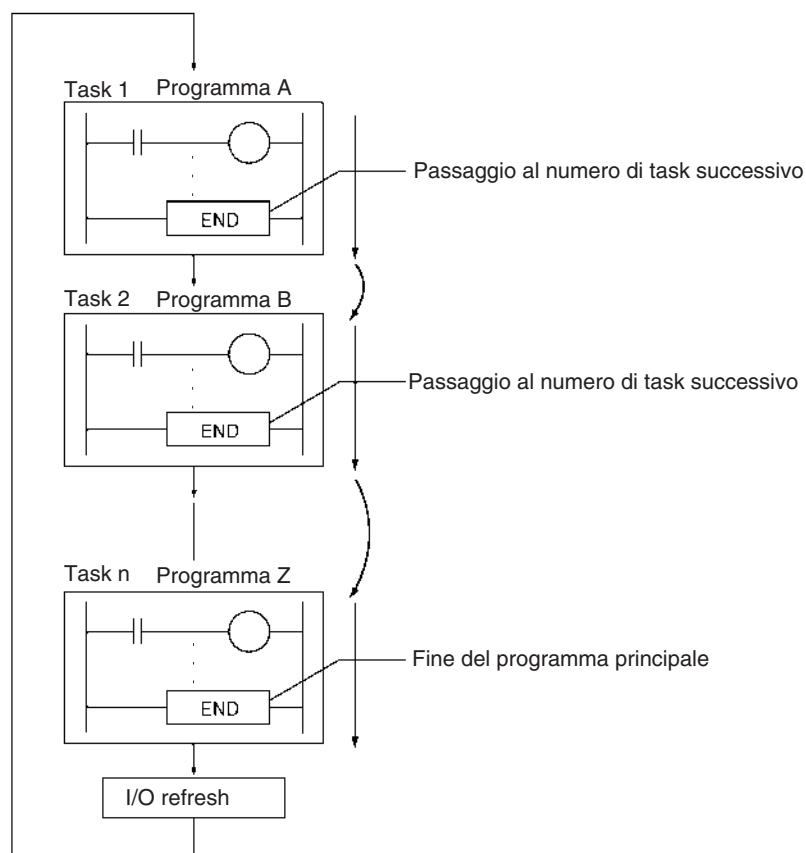
| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | END (001) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Non consentita | Non consentita | Non consentita | OK |

Descrizione END(001) termina l'esecuzione di un programma in quel ciclo. Le istruzioni inserite dopo END(001) non sono eseguite.

Viene eseguito il programma con il numero successivo di task. Se il numero di task del programma in esecuzione è il più alto all'interno del programma, l'istruzione END(001) marca la fine del programma principale.

**Avvertenze**

Specificare sempre END(001) alla fine di ciascun programma. Se un'istruzione END(001) non è inserita nel programma, si verificherà un errore di programma.

3-5-2 NO OPERATION: NOP(000)**Scopo**

Istruzione priva di funzione. Per NOP(000) non viene eseguita alcuna elaborazione.

Simbolo programmazione ladder

NOP(000) non prevede alcuna associazione con un simbolo programmazione ladder.

Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | NOP(000) |
| Aggiornamento immediato | | Non supportata |

Arete di programma applicabili

| Arete di programma a blocchi | Arete di programma step | Subroutine | Task ad interrupt |
|-------------------------------------|--------------------------------|-------------------|--------------------------|
| OK | OK | OK | OK |

Descrizione

Per NOP(000) non viene eseguita alcuna elaborazione, ma è possibile utilizzare l'istruzione per aggiungere al programma alcune righe per l'inserimento futuro di istruzioni. Quando le istruzioni vengono inserite in un secondo tempo, non avviene alcuna modifica negli indirizzi del programma.

Flag

NOP(000) non influisce su alcun flag.

Avvertenze

È possibile utilizzare NOP(000) soltanto con display mnemonici e non con programmi ladder.

3-5-3 Panoramica sulle istruzioni di interblocco

istruzioni di interblocco

È possibile utilizzare le seguenti combinazioni di istruzioni per bloccare le uscite in una sezione di programma.

- INTERLOCK e INTERLOCK CLEAR (IL(002) e IL(003))
- MULTI-INTERLOCK DIFFERENTIATION HOLD e MULTI-INTERLOCK CLEAR (MILH(517) e MILC(519))*

Nota MILH(517) mantiene lo stato del flag comparativo, affinché le istruzioni differenziate precedentemente bloccate vengano eseguite quando l'interblocco è stato cancellato.

- MULTI-INTERLOCK DIFFERENTIATION RELEASE e MULTI-INTERLOCK CLEAR (MILR(518) e MILC(519))*

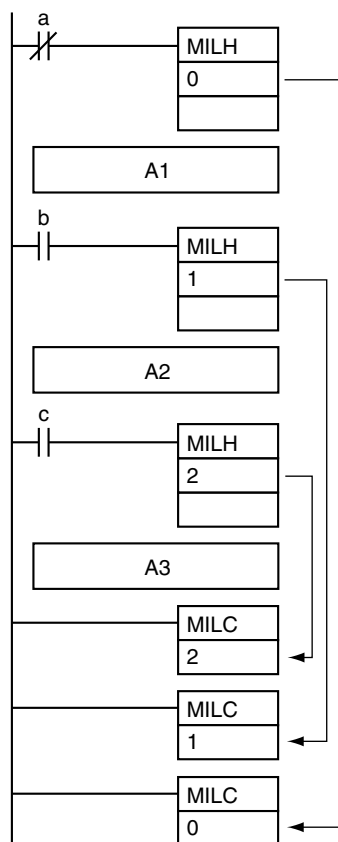
Nota MILR(518) non mantiene lo stato del flag comparativo, per cui le istruzioni differenziate precedentemente bloccate non vengono eseguite una volta che l'interblocco è stato cancellato.

* Queste istruzioni sono supportate solo dalle CPU della serie CS/CJ versione 2.0 o successiva.

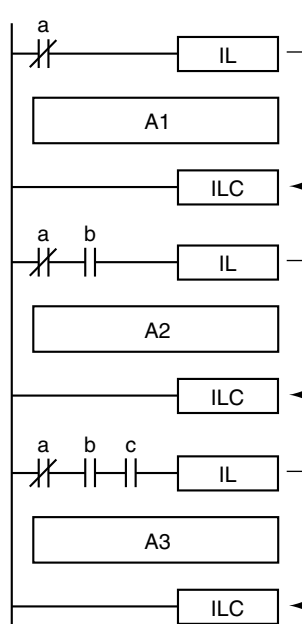
Differenze tra interblocchi e interblocchi multipli

Non è possibile nidificare gli interblocchi normali (IL(002) and IL(003)), ma solo quelli multipli come (MILH(517), MILR(518) e MILC(519)). Come illustrato nel diagramma qui di seguito, la nidificazione degli interblocchi multipli consente di semplificare la programmazione ladder.

Interblocchi con MILH e MILC



Interblocchi con IL e ILC



Differenze tra MILH(517) e MILR(518)

Negli interblocchi creati con MILH(517) e MILR(518), le istruzioni differenziate (DIFU, DIFD oppure quelle con prefisso @ o %) hanno un funzionamento diverso.

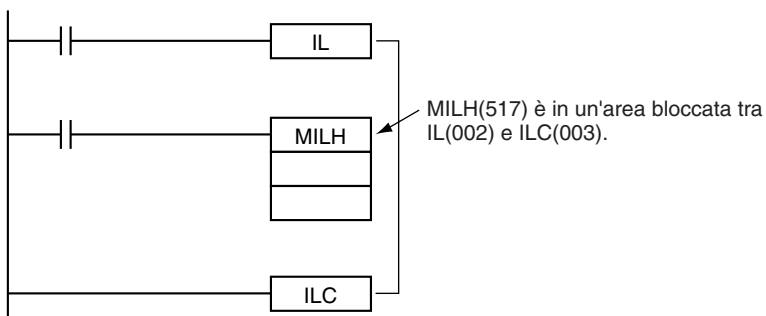
Il funzionamento delle istruzioni differenziate in un interblocco creato con MILH(517) è identico al funzionamento in un interblocco creato con IL(002).

Per informazioni dettagliate, fare riferimento alla sezione 3-5-5 *MULTI-INTERLOCK DIFFERENTIATION HOLD*, *MULTI-INTERLOCK DIFFERENTIATION RELEASE* e *MULTI-INTERLOCK CLEAR*: MILH(517), MILR(518) e MILC(519).

Avvertenze

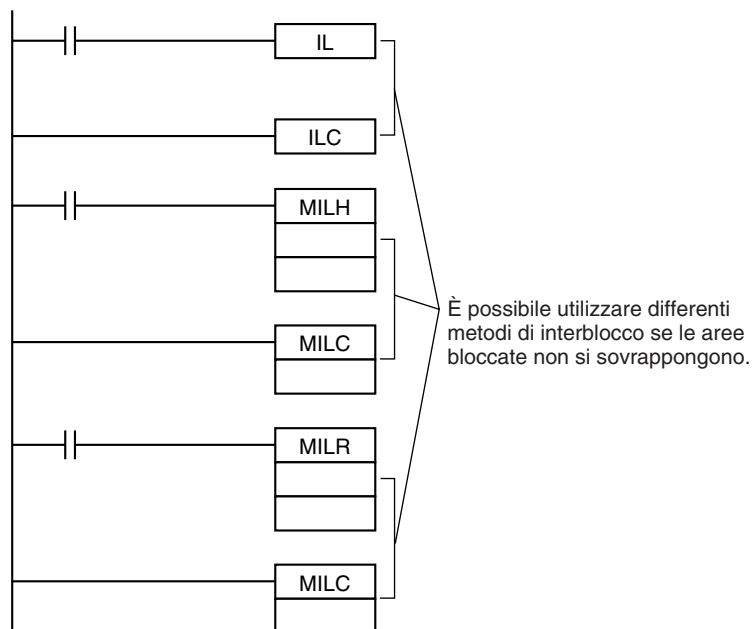
È opportuno non combinare interblocchi creati con istruzioni di interblocco differenti (IL-ILC, MILH-MILC e MILR-MILC). Se differenti metodi di interblocco vengono utilizzati congiuntamente, è possibile che gli interblocchi non funzionino in modo corretto. Per informazioni dettagliate sulle istruzioni combinate, fare riferimento alla sezione 3-5-5 *MULTI-INTERLOCK DIFFERENTIATION HOLD*, *MULTI-INTERLOCK DIFFERENTIATION RELEASE* e *MULTI-INTERLOCK CLEAR*: MILH(517), MILR(518) e MILC(519).

Ad esempio, non è possibile inserire un'istruzione MILH(517) tra IL(002) e ILC(003).



Nota È possibile utilizzare congiuntamente i differenti interblocchi (IL-ILC, MILH-MILC e MILR-MILC) se le sezioni del programma di interblocco non si sovrappongono.

Ad esempio, è possibile utilizzare i tre metodi di interblocco senza sovrapposizione come indicato nel diagramma qui di seguito.



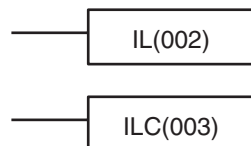
Differenze tra interblocchi e salti

Nella tabella seguente sono riportate le differenze tra gli interblocchi (creati con IL(002)/ILC(003), MILH(517)/MILC(519) o MILR(518)/MILC(519)) e i salti creati con JMP(004)/JME(005).

| Elemento | Funzionamento con IL(002)/ILC(003), MILH(517)/MILC(519) o MILR(518)/MILC(519) | Funzionamento con JMP(004)/JME(005) |
|--|---|---|
| Esecuzione dell'istruzione | Ad eccezione di OUT, OUT NOT e OUTB(534), tutte le istruzioni, comprese quelle del temporizzatore, non vengono eseguite. | Non viene eseguita alcuna istruzione |
| Stato dell'uscita nelle istruzioni | Ad eccezione delle uscite in OUT, OUT NOT, OUTB(534) e nelle istruzioni del temporizzatore, tutte le uscite mantengono lo stato precedente. | Tutte le uscite mantengono lo stato precedente. |
| Bit in OUT, OUT NOT, OUTB(534) | OFF | Tutte le uscite mantengono lo stato precedente. |
| Stato delle istruzioni del temporizzatore (tranne TTIM(087), TTIMX(555), MTIM(543) e MTIMX(554)) | Reset | I temporizzatori in funzione (soltanto TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540) e TMHHX(552)) continuano a funzionare poiché i valori attuali (PV) vengono aggiornati anche quando l'istruzione del temporizzatore non è in fase di esecuzione. |

3-5-4 INTERLOCK e INTERLOCK CLEAR: IL(002) e ILC(003)**Scopo**

Determina l'interblocco per tutte le uscite comprese tra IL(002) e ILC(003) quando lo stato della condizione di esecuzione per IL(002) è OFF. IL(002) e ILC(003) vengono di norma utilizzate in coppia.

Simboli programmazione ladder**Variazioni**

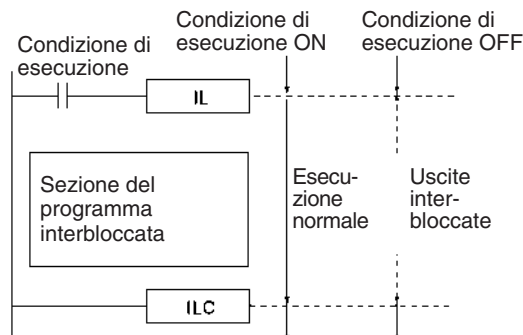
| Variazioni | Procede all'interblocco se impostata su OFF. Non procede all'interblocco se impostata su ON | IL(002) |
|-------------------------|---|----------------|
| Aggiornamento immediato | | Non supportata |
| Variazioni | Eseguita a ogni ciclo con condizione ON | ILC(003) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | Non consentita | OK | OK |

Descrizione

Se la condizione di esecuzione per IL(002) è impostata su OFF, le uscite per tutte le istruzioni comprese tra IL(002) e ILC(003) vengono bloccate. Se la condizione di esecuzione per IL(002) è impostata su ON, le istruzioni comprese tra IL(002) e ILC(003) vengono eseguite regolarmente.



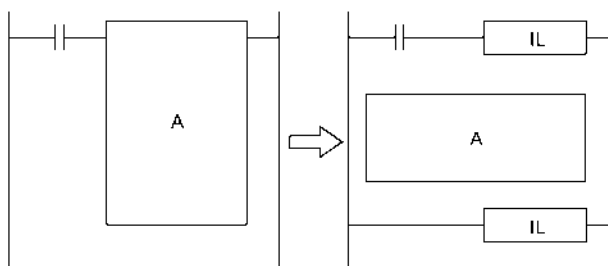
Nella tabella seguente viene illustrato il funzionamento di alcune uscite in una sezione interbloccata compresa tra IL(002) e ILC(003).

| Istruzione | | Funzionamento |
|--|-----------------------|---------------------------------------|
| Bit specificati in OUT, OUT NOT o OUTB(534) | | OFF |
| TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542) e TIMXL(553) | Flag di completamento | OFF (reset) |
| | PV | Valore impostato per il tempo (reset) |
| Bit e canali specificati in tutte le altre istruzioni (vedere nota) | | Mantengono lo stato precedente |

Nota I bit e i canali in tutte le altre istruzioni, comprese TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT e KEEP(011), mantengono lo stato precedente.

Se si desidera che alcuni bit mantengano lo stato ON in una sezione del programma interbloccata, utilizzare SET per impostare tali bit su ON appena prima di IL(002).

Spesso è più efficiente cambiare una sezione del programma con IL(002) e ILC(003). Quando numerosi processi sono controllati con la stessa condizione di esecuzione, è necessario un numero inferiore di step di programma per posizionare tali processi tra IL(002) e ILC(003).



Nella tabella seguente sono riportate le differenze tra IL(002)/ILC(003) e JMP(004)/JME(005).

| Elemento | Funzionamento con IL(002)/ILC(003) | Funzionamento con JMP(004)/JME(005) |
|--|---|---|
| Esecuzione dell'istruzione | Ad eccezione di OUT, OUT NOT e OUTB(534), tutte le istruzioni, comprese quelle del temporizzatore, non vengono eseguite. | Non viene eseguita alcuna istruzione |
| Stato dell'uscita nelle istruzioni | Ad eccezione delle uscite in OUT, OUT NOT, OUTB(534) e nelle istruzioni del temporizzatore, tutte le uscite mantengono lo stato precedente. | Tutte le uscite mantengono lo stato precedente. |
| Bit in OUT, OUT NOT, OUTB(534) | OFF | Tutte le uscite mantengono lo stato precedente. |
| Stato delle istruzioni del temporizzatore (tranne TTIM(087), TTIMX(555), MTIM(543) e MTIMX(554)) | Reset | I temporizzatori in funzione (soltanto TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540) e TMHHX(552)) continuano a funzionare poiché i valori attuali (PV) vengono aggiornati anche quando l'istruzione del temporizzatore non è in fase di esecuzione. |

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 i flag di uguaglianza e i flag negativi sono impostati su OFF.

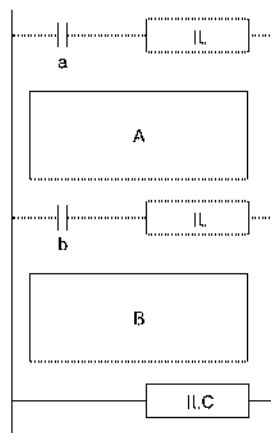
Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

Quando una sezione del programma è interbloccata a causa dell'esecuzione interna delle istruzioni bloccate, il tempo di ciclo non viene abbreviato.

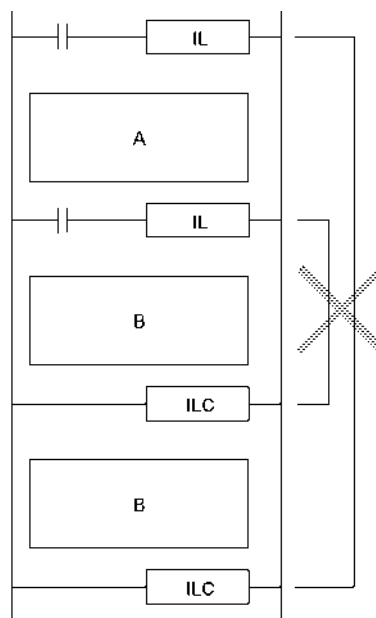
Quando DIFU(013), DIFD(014) e le istruzioni differenziate vengono programmate tra IL(002) e ILC(003), il loro funzionamento non dipende soltanto dallo stato della condizione di esecuzione. Se DIFU(013) o DIFD(014) si trovano in una sezione interbloccata e la condizione di esecuzione di IL(002) è impostata su OFF, non vengono registrate modifiche della condizione di esecuzione di DIFU(013), DIFD(014) o di un'istruzione differenziata.

In generale, IL(002) e ILC(003) vengono utilizzate in coppia, ma è possibile specificare più di un'istruzione IL(002) con una ILC(003) singola, come illustrato nel diagramma seguente. Se IL(002) e ILC(003) non sono in coppia, viene visualizzato un messaggio di errore durante il controllo del programma, ma il programma viene comunque eseguito correttamente.



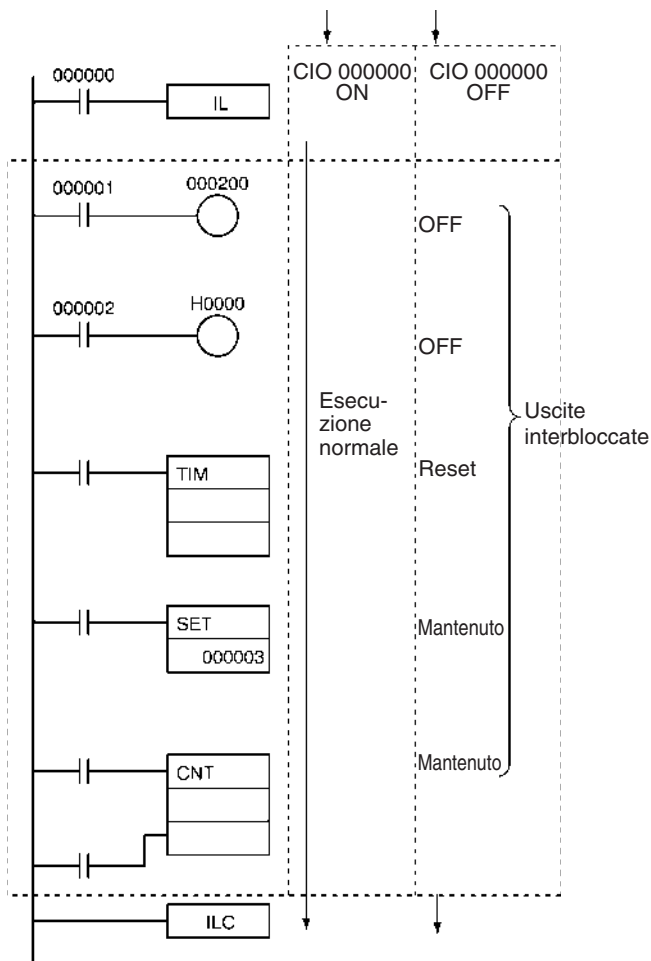
| Condizione di esecuzione | | Sezione di programma | |
|--------------------------|-----|----------------------|--------------------|
| a | b | A | B |
| OFF | ON | Interblocco | Interblocco |
| OFF | OFF | Interblocco | Interblocco |
| ON | OFF | Nessun interblocco | Interblocco |
| ON | ON | Nessun interblocco | Nessun interblocco |

Come illustrato nel diagramma seguente, non è possibile nidificare IL(002) e ILC(003). A tal fine è necessario utilizzare MILH(517)/MILR(518) e MILC(519).



Esempi

Nell'esempio che segue quando CIO 000000 è impostato su OFF, tutte le uscite comprese tra IL(002) e ILC(003) vengono rese interdipendenti. Nell'esempio che segue quando CIO 000000 è impostato su ON, le istruzioni comprese tra IL(002) e ILC(003) vengono eseguite normalmente.



3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE e MULTI-INTERLOCK CLEAR: MILH(517), MILR(518) e MILC(519)

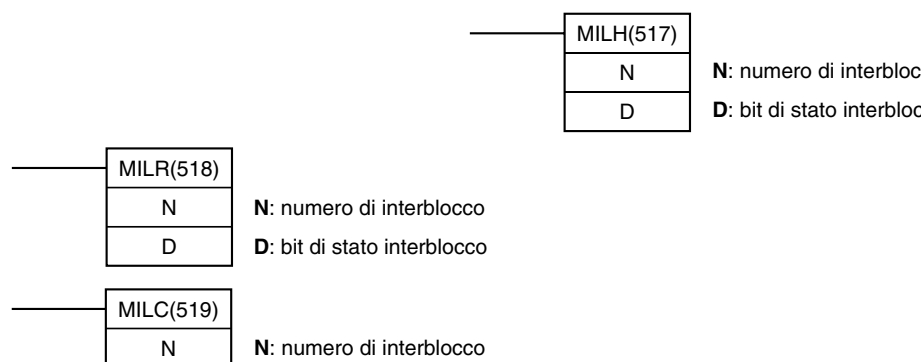
Scopo

Blocca tutte le uscite comprese tra MILH(517) (o MILR(518)) e MILC(519) quando la condizione di esecuzione per MILH(517) (o MILR(518)) è OFF. MILH(517) (o MILR(518)) e MILC(519) vengono normalmente utilizzate in coppia.

A differenza di quanto avviene per gli interblocchi di IL(002)/ILC(003), gli interblocchi di MILH(517)/MILC(519) e di MILR(518)/MILC(519) possono essere nidificati. Il funzionamento delle istruzioni differenziate è diverso per quanto riguarda gli interblocchi creati con MILH(517) e MILR(518).

Queste istruzioni sono supportate solo dalle CPU della serie CS/CJ versione 2.0 o successiva.

Simboli programmazione ladder



Operandi

N: numero di interblocco

Il numero di interblocco deve essere compreso tra 0 e 15. È necessario che il numero di interblocco dell'istruzione MILH(517) (o MILR(518)) corrisponda allo stesso numero nell'istruzione MILC(519) relativa.

I numeri di interblocco possono essere utilizzati in qualsiasi ordine.

D: bit di stato interblocco

- ON quando la sezione del programma non è interbloccata.
- OFF quando la sezione del programma è interbloccata.

Quando l'interblocco è impegnato, è possibile procedere a un'impostazione forzata del bit di stato interblocco per ottenere il rilascio. Viceversa, quando l'interblocco non è impegnato, è possibile procedere a un'impostazione forzata del bit di stato interblocco per ottenere l'impegno.

Caratteristiche operando

| Area | N | D |
|-------------------------|-----|----------------------------|
| Area CIO | --- | Da CIO 000000 a CIO 614315 |
| Area di lavoro | --- | Da W00000 a W51115 |
| Area bit di ritenività | --- | Da H00000 a H51115 |
| Area bit ausiliaria | --- | Da A00000 ad A95915 |
| Area del temporizzatore | --- | --- |
| Area del contatore | --- | --- |
| Area DM | --- | --- |
| Area EM senza banco | --- | --- |

| Area | N | D |
|--|-----------|--|
| Area EM con banco | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- |
| Indirizzi DM/EM indiretti in BCD | --- | --- |
| Costanti | Da 0 a 15 | --- |
| Registri dati | --- | --- |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 |

Variazioni

| Variazioni | Procede all'interblocco se impostata su OFF. Non procede all'interblocco se impostata su ON | MILH(517) e MILR(518) |
|-------------------------|---|-----------------------|
| Aggiornamento immediato | | Non supportata |

| Variazioni | Eseguita a ogni ciclo con condizione ON | MILC(519) |
|-------------------------|---|----------------|
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

Nella tabella seguente sono riportate le aree di programma applicabili per MILH(517), MILR(518) e MILC(519)

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | Non consentita | OK | OK |

Descrizione

Quando la condizione di esecuzione per MILH(517) (o MILR(518)) con numero di interblocco N è OFF, le uscite di tutte le istruzioni comprese tra tale MILH(517)/MILR(518) e la successiva istruzione MILC(519) con numero di interblocco N vengono bloccate.

Quando la condizione di esecuzione per MILH(517) (o MILR(518)) con numero di interblocco N è ON, le istruzioni comprese tra tale MILH(517)/MILR(518) e la successiva istruzione MILC(519) con numero di interblocco N vengono eseguite regolarmente.

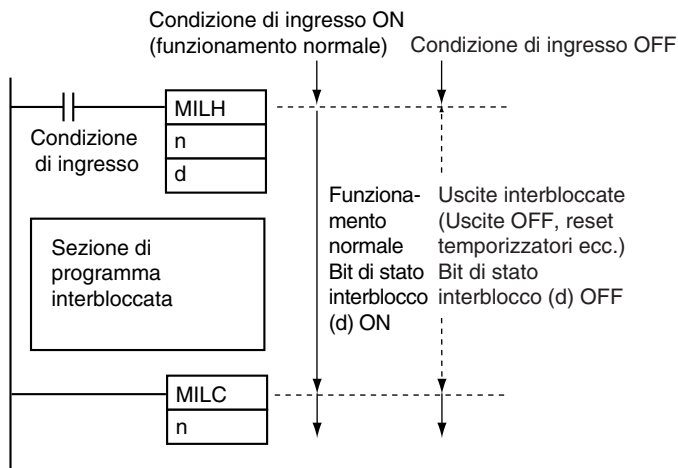
Stato di interblocco

Nella tabella seguente viene illustrato il funzionamento di alcune uscite in una sezione interbloccata compresa tra l'istruzione MILH(517)/MILR(518) e la successiva MILC(519).

| Istruzione | Funzionamento |
|--|--|
| Bit specificati in OUT, OUT NOT o OUTB(534) | OFF |
| TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542) e TIMXL(553) | Flag di completamento PV Valore impostato per il tempo (reset) |
| Bit e canali specificati in tutte le altre istruzioni (vedere nota) | Mantengono lo stato precedente |

Nota I bit e i canali in tutte le altre istruzioni, comprese TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT e KEEP(011), mantengono lo stato precedente.

Quando l'interblocco è impegnato, MILH(517)/MILR(518) imposta su OFF il bit di stato interblocco (operando D); quando l'interblocco non è impegnato, l'istruzione imposta il bit su ON. È quindi possibile controllare il bit di stato interblocco per verificare se l'interblocco per un determinato numero di interblocco sia impegnato o meno.



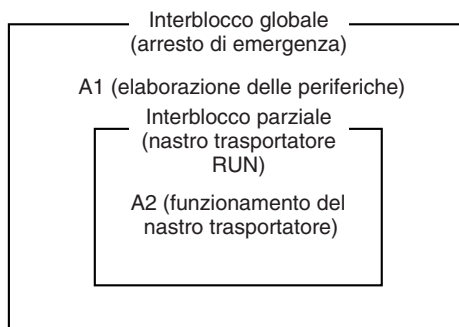
Nidificazione

Gli interblocchi vengono nidificati quando una sezione del programma interbloccata (combinazione di MILH(517)/MILR(518) e MILC(519)) viene collocata in un'altra sezione interbloccata del programma (combinazione MILH(517)/MILR(518) e MILC(519)). Gli interblocchi possono essere nidificati fino a un massimo di 16 livelli.

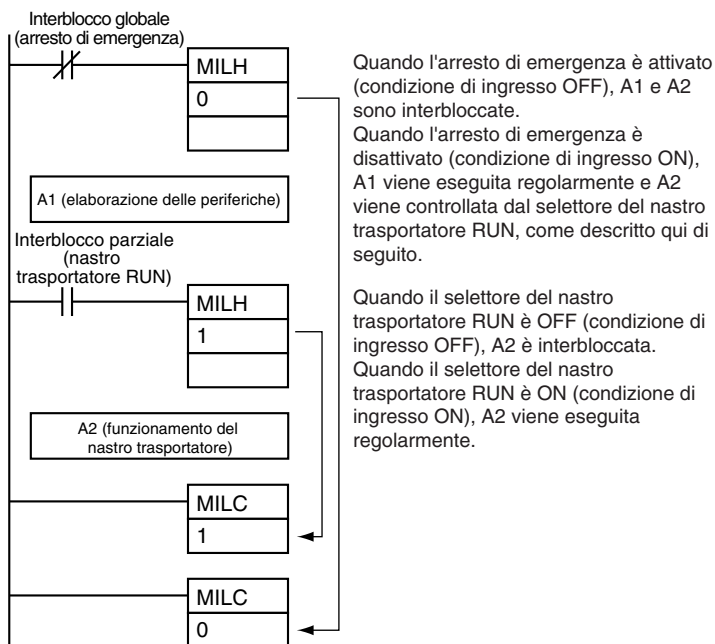
È possibile utilizzare la nidificazione per i seguenti tipi di applicazioni.

- Esempio 1

Interblocco dell'intero programma con una condizione e interblocco di una parte del programma con un'altra condizione (livello di nidificazione 1)

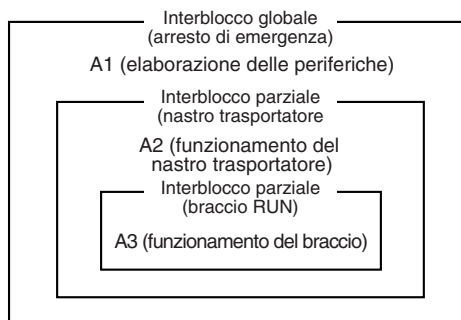


- A1 e A2 sono interbloccate quando il pulsante Arresto di emergenza è selezionato.
- A2 è interbloccata quando il nastro trasportatore RUN è OFF.

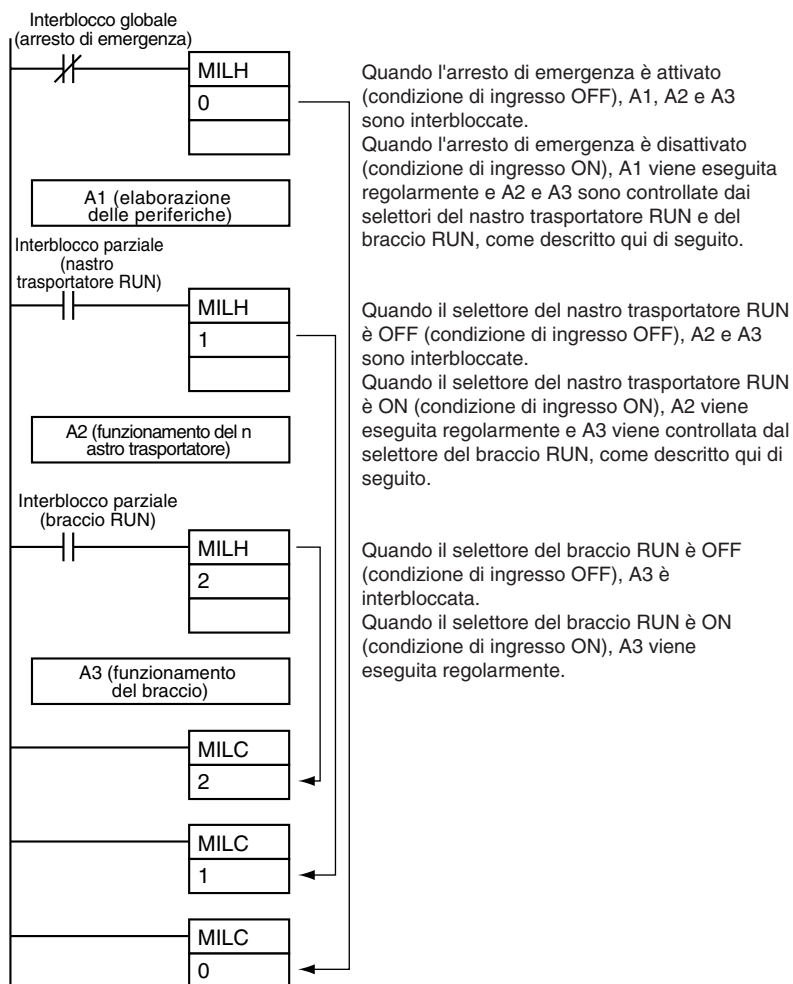


• Esempio 2

Interblocco dell'intero programma con una condizione e interblocco di due parti sovrapposte del programma con altre condizioni (livello di nidificazione 2)



- A1, A2 e A3 sono interbloccate quando il pulsante Arresto di emergenza è selezionato.
- A2 e A3 sono interbloccate quando il nastro trasportatore RUN è OFF.
- A3 è interbloccata quando il braccio RUN è OFF.



Differenze tra MILH(517) e MILR(518)

Negli interblocchi creati con MILH(517) e MILR(518), le istruzioni differenziate (DIFU, DIFD oppure quelle con prefisso @ o %) hanno un funzionamento diverso.

Quando una sezione del programma è interbloccata con MILR(518), un'istruzione differenziata **non** viene eseguita una volta cancellato l'interblocco, anche se durante l'interblocco era stata attivata la condizione di differenziazione (confronto tra lo stato della condizione di esecuzione all'inizio e all'eliminazione dell'interblocco).

Quando una sezione del programma è interbloccata con MILH(517), un'istruzione differenziata viene **eseguita** una volta cancellato l'interblocco se durante l'interblocco era stata attivata la condizione di differenziazione (confronto tra lo stato della condizione di esecuzione all'inizio e all'eliminazione dell'interblocco).

| Istruzione | Funzionamento delle istruzioni differenziate |
|--|--|
| MILH(517) MULTI-INTERLOCK DIFFERENTIATION HOLD | Un'istruzione differenziata (DIFU, DIFD oppure un'istruzione con prefisso @ o %) viene eseguita dopo la cancellazione dell'interblocco, se la condizione di differenziazione dell'istruzione era stata specificata al momento del blocco dell'istruzione stessa. (Lo stato della condizione di esecuzione all'inizio dell'interblocco viene messo a confronto con lo stato dell'istruzione dopo l'eliminazione dell'interblocco). |
| MILR(518) MULTI-INTERLOCK DIFFERENTIATION RELEASE | Un'istruzione differenziata (DIFU, DIFD oppure un'istruzione con prefisso @ o %) non viene eseguita dopo la cancellazione dell'interblocco, anche se la condizione di differenziazione dell'istruzione è stata specificata mentre l'istruzione era bloccata. |

- Funzionamento delle istruzioni differenziate in un interblocco MILH(517)

Se tra MILH(517) e l'istruzione MILC(519) corrispondente è presente un'istruzione differenziata (DIFU, DIFD o un'istruzione con prefisso @ o %), tale istruzione viene **eseguita** dopo la cancellazione dell'interblocco se è stata precedentemente specificata la condizione di differenziazione dell'istruzione (il sistema mette a confronto lo stato della condizione di esecuzione all'inizio dell'interblocco con lo stato dell'istruzione dopo la cancellazione dell'interblocco).

Analogamente, un'istruzione differenziata sarà eseguita se la sua condizione di esecuzione è stabilita nel momento stesso in cui l'interblocco ha inizio o viene cancellato.

Numerose altre condizioni del programma possono provocare la reimpostazione della condizione di differenziazione anche se tale condizione è stata assunta durante l'interblocco. In questo caso l'istruzione differenziata non viene eseguita una volta eliminato l'interblocco.

- Esempio

Se un'istruzione differenziata sul fronte di salita (DIFU(013)) è in uso e la condizione di ingresso è impostata su OFF all'inizio dell'interblocco e su ON alla cancellazione dell'interblocco, DIFU(013) viene **eseguita** quando l'interblocco è stato cancellato (il funzionamento delle istruzioni differenziate nell'interblocco MILH(517) è analogo a quello che si verificherebbe in un interblocco IL(002)).

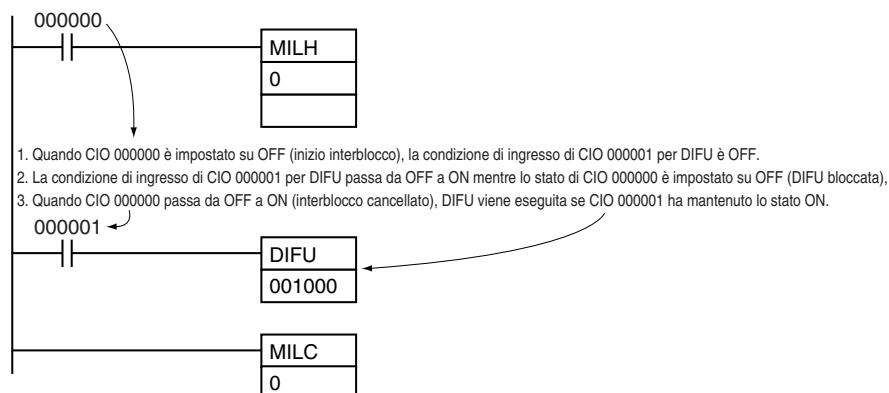
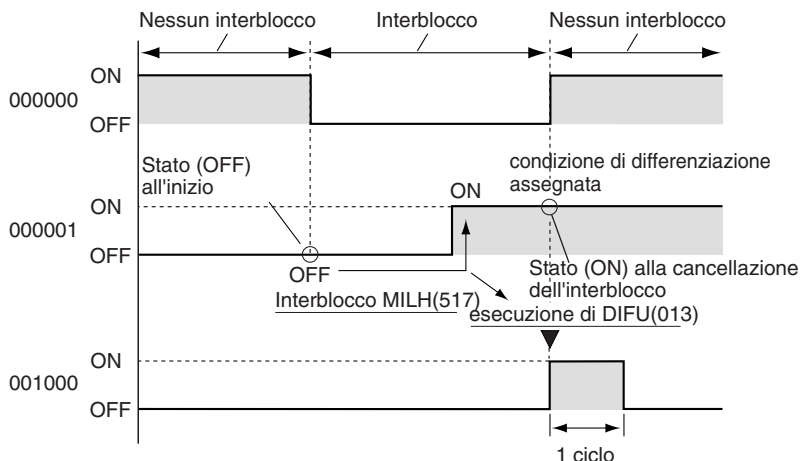


Diagramma di funzionamento



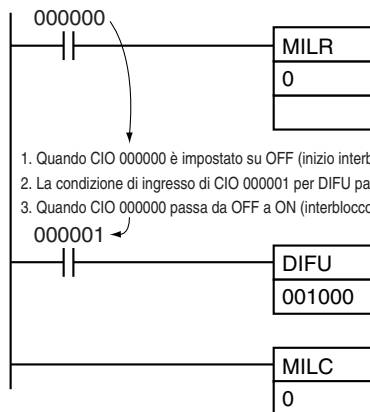
• Funzionamento delle istruzioni differenziate in un interblocco MILR(518)

Se tra MILR(518) e l'istruzione MILC(519) corrispondente è presente un'istruzione differenziata (DIFU, DIFD oppure un'istruzione con prefisso @ o %), tale istruzione **non** viene eseguita dopo la cancellazione dell'interblocco anche se aveva precedentemente assunto la condizione di differenziazione (il sistema mette a confronto lo stato della condizione di esecuzione nel ciclo all'inizio dell'interblocco con lo stato dell'istruzione nel ciclo dopo la cancellazione dell'interblocco).

Analogamente, un'istruzione differenziata non viene eseguita se la sua condizione di esecuzione è assegnata nel momento stesso in cui l'interblocco ha inizio o viene cancellato.

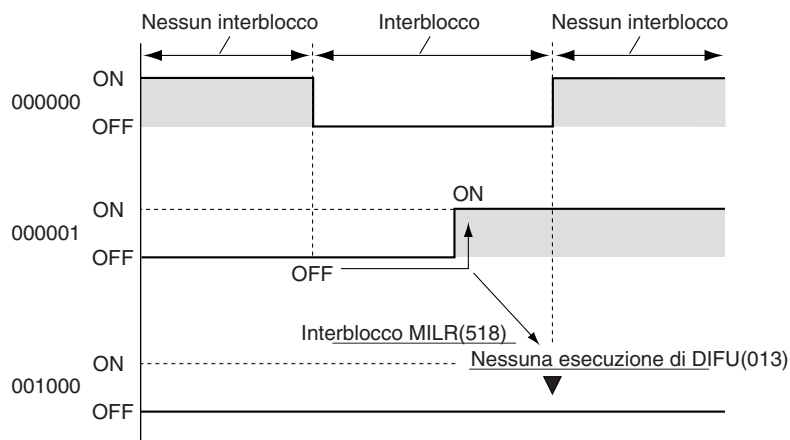
• Esempio

Se un'istruzione differenziata sul fronte di salita (DIFU(013)) è in uso e la condizione di ingresso è impostata su OFF all'inizio dell'interblocco e su ON alla cancellazione dell'interblocco, DIFU(013) **non** viene eseguita quando l'interblocco è stato cancellato.



1. Quando CIO 000000 è impostato su OFF (inizio interblocco), la condizione di ingresso di CIO 000001 per DIFU è OFF.
2. La condizione di ingresso di CIO 000001 per DIFU passa da OFF a ON mentre lo stato di CIO 000000 è impostato su OFF (DIFU bloccata),
3. Quando CIO 000000 passa da OFF a ON (interblocco cancellato), DIFU non viene eseguita anche se CIO 000001 ha mantenuto lo stato ON.

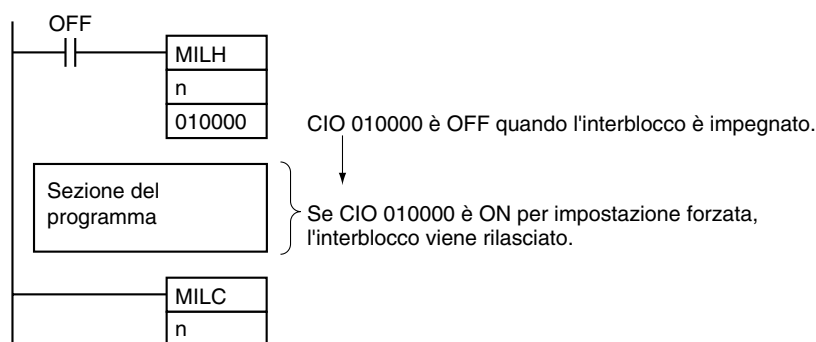
Diagramma di funzionamento



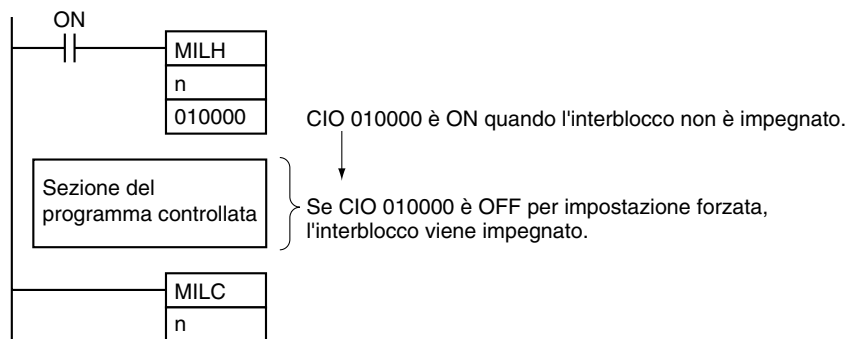
Controllo dello stato di interblocco da un dispositivo di programmazione

È possibile impegnare o rilasciare un interblocco manualmente da un dispositivo di programmazione eseguendo un ripristino forzato o un'impostazione forzata del bit di stato interblocco (specificato con l'operando D di MILH(517) e MILR(518)). Lo stato forzato del bit di stato interblocco ha la priorità e sovrascrive lo stato dell'interblocco calcolato dall'esecuzione del programma.

Impostazione forzata: rilascia l'interblocco.

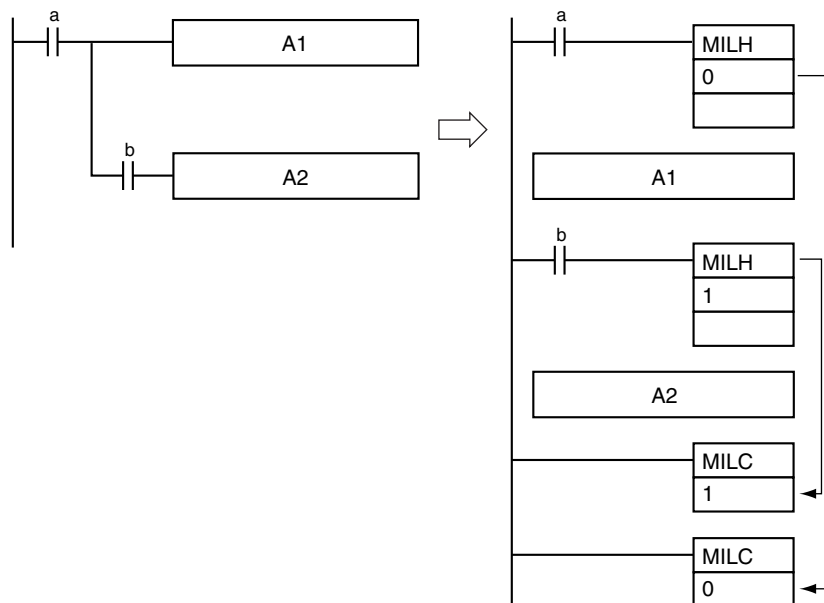


Ripristino forzato: impegna l'interblocco.



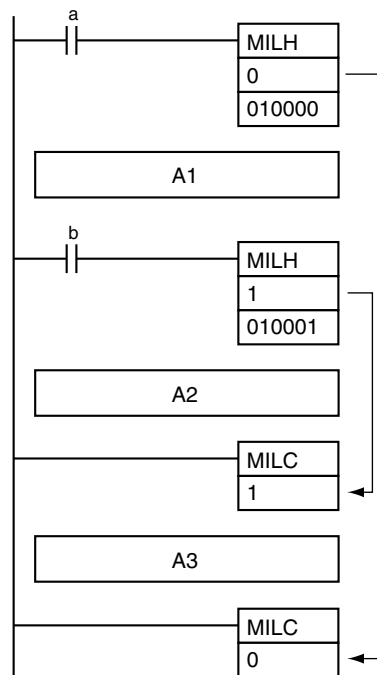
Nota L'utilizzo di interblocchi con MILH(517) o MILR(518) può migliorare l'efficienza del programma.

Invece di passare da un'elaborazione all'altra con molteplici condizioni, è opportuno specificare un'istruzione MILH(517) o MILR(518) prima di ogni processo e un'istruzione MILC(519) al termine.



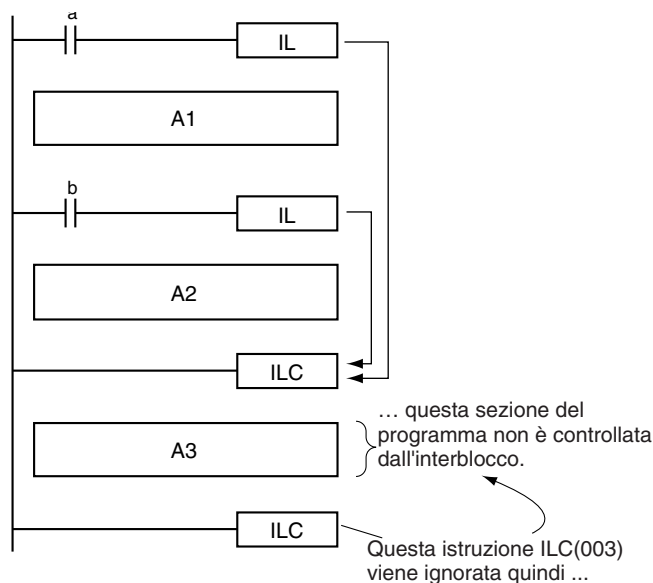
A differenza di quanto avviene per gli interblocchi di IL(002), è possibile modificare quelli di MILH(517) e MILR(518); di conseguenza il funzionamento di programmi simili sarà differente se MILH(517) o MILR(518) vengono utilizzate al posto di ILC(002).

Programma con interblocchi MILH(517)/MILC(519)



| Condizione di esecuzione | | Sezione di programma | | |
|--------------------------|-----|----------------------|--------------------|--------------------|
| a | b | A1 | A2 | A3 |
| OFF | ON | Interblocco | Interblocco | Nessun interblocco |
| | OFF | | | |
| ON | OFF | Nessun interblocco | Interblocco | Nessun interblocco |
| ON | ON | Nessun interblocco | Nessun interblocco | Nessun interblocco |

Programmi con interblocchi IL(002)/ILC(003)



| Condizione di esecuzione | | Sezione di programma | | |
|--------------------------|-----|----------------------|--------------------|---|
| a | b | A1 | A2 | A3 |
| OFF | ON | Interblocco | Interblocco | Nessun interblocco (nessuno controllo da interblocco IL(002)/ILC(003)) |
| | OFF | | | |
| ON | OFF | Nessun interblocco | Interblocco | |
| ON | ON | Nessun interblocco | Nessun interblocco | |

Se si desidera che alcuni bit mantengano lo stato ON in una sezione del programma interbloccata da MILH(517) o MILR(518), utilizzare SET per impostare tali bit su ON appena prima dell'istruzione MILH(517) o MILR(518).

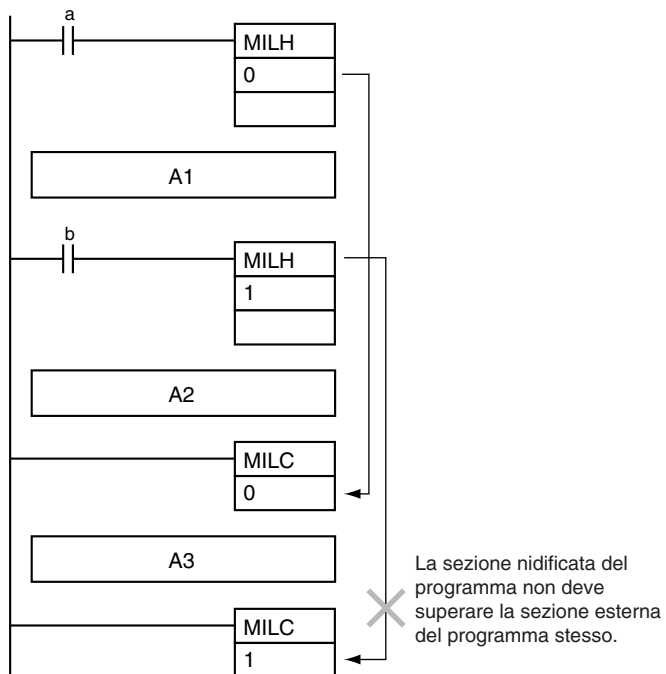
Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---------------|
| Flag di errore | ER | OFF |

Avvertenze

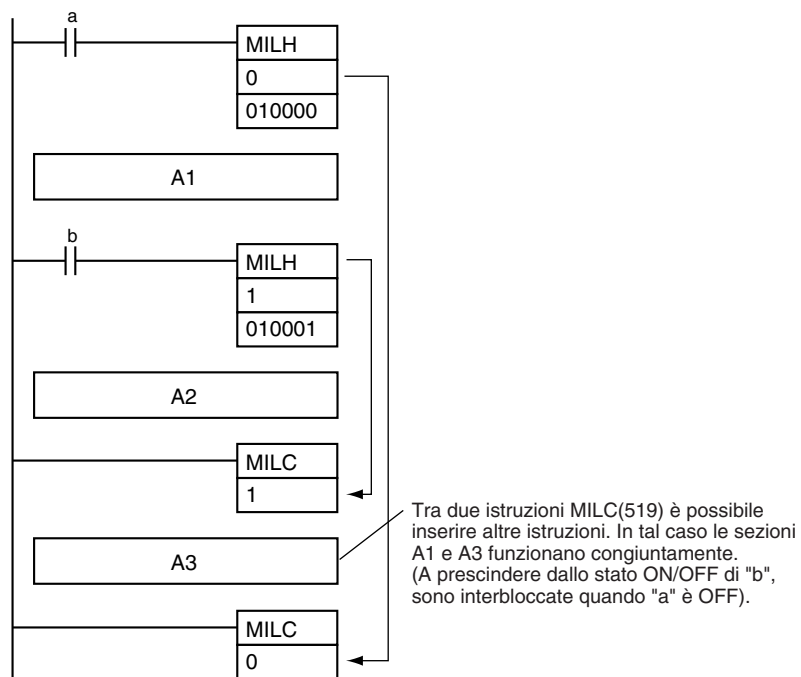
Quando una sezione del programma è interbloccata da MILH(517) o MILR(518) a causa dell'esecuzione interna delle istruzioni bloccate, il tempo di ciclo non viene abbreviato.

Nella nidificazione degli interblocchi è necessario assegnare numeri di interblocco in modo tale che la sezione nidificata del programma non superi la sezione esterna del programma stesso.

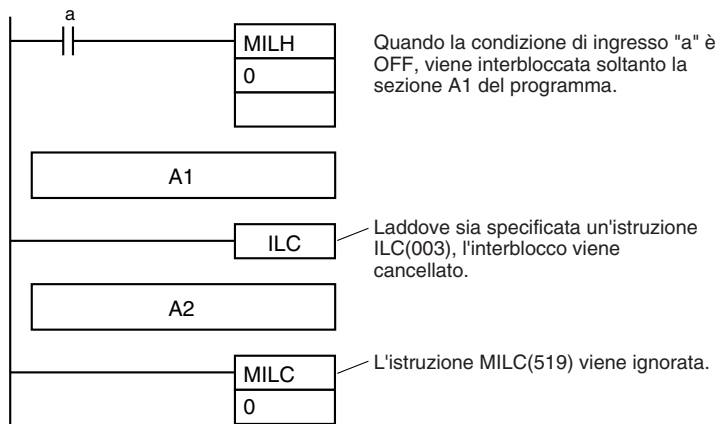


| Condizione di esecuzione | | Sezione di programma | | |
|--------------------------|-----|----------------------|--------------------|--------------------|
| a | b | A1 | A2 | A3 |
| OFF | ON | Interblocco | Interblocco | Nessun interblocco |
| | OFF | | | |
| ON | OFF | Nessun interblocco | Interblocco | Interblocco |
| | ON | Nessun interblocco | Nessun interblocco | Nessun interblocco |

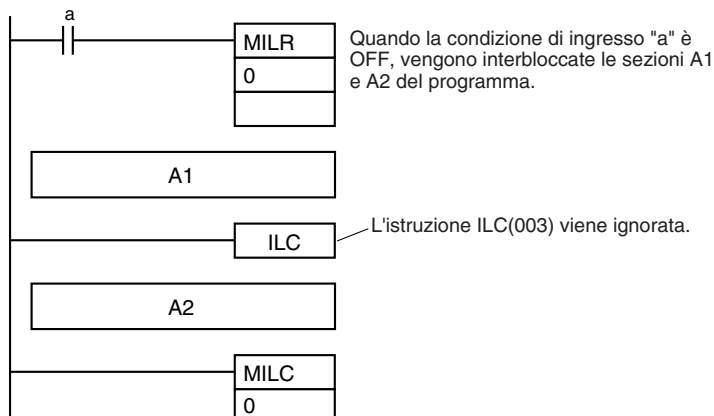
Tra le istruzioni MILC(519) è possibile inserirne altre, come illustrato nel diagramma che segue.



Se un'istruzione ILC(003) si trova tra una coppia MILH(517) - MILC(519), la sezione del programma tra MILH(517) e ILC(003) viene interbloccata.

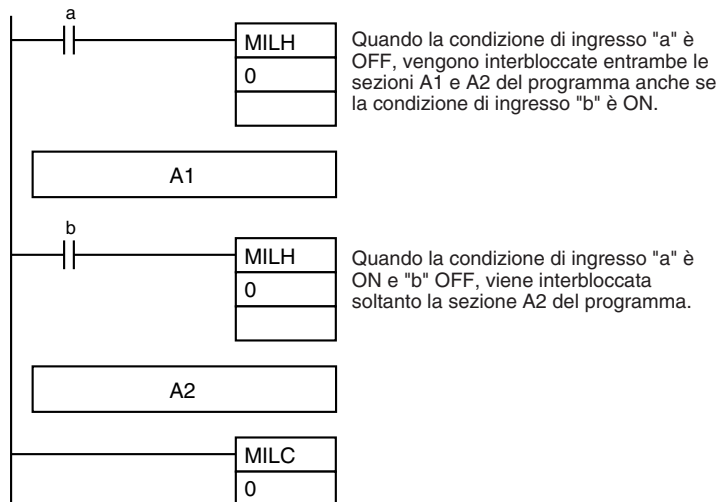


Se un'istruzione ILC(003) si trova tra una coppia MILR(518) - MILC(519), l'istruzione ILC(003) viene ignorata e tutta la sezione del programma tra MILR(518) e MILC(519) verrà interbloccata.



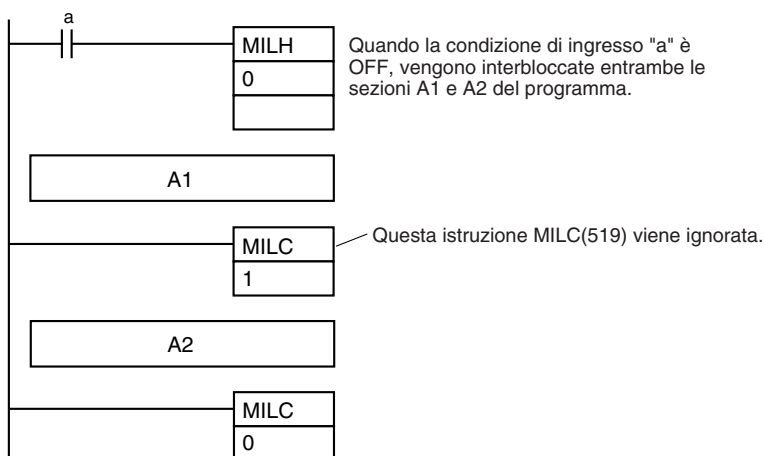
Se un'altra istruzione MILH(517) o MILR(518) con lo stesso numero di interblocco si trova tra una coppia MILH(517) - MILC(519) e il primo interblocco dell'istruzione MILH(517) è impegnato, la seconda MILH(517)/MILR(518) non viene eseguita.

Se un'altra istruzione MILH(517) o MILR(518) con lo stesso numero di interblocco si trova tra una coppia MILH(517) - MILC(519) e il primo interblocco dell'istruzione MILH(517) non è impegnato, la seconda MILH(517)/MILR(518) funziona correttamente.



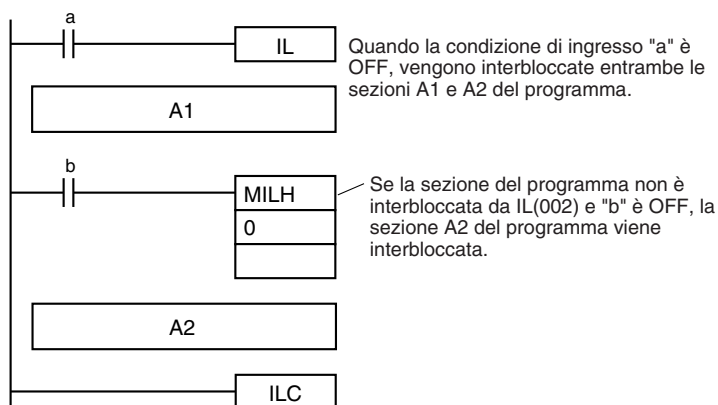
Nota Gli interblocchi MILR(518) funzionano in modo uguale se un'altra istruzione MILH(517) o MILR(518) con lo stesso numero di interblocco si trova tra una coppia MILR(518) - MILC(519).

Se tra una coppia MILH(517)/MILR(518) - MILC(519) si trova un'istruzione MILC(519) con un numero diverso di interblocco, tale istruzione verrà ignorata.

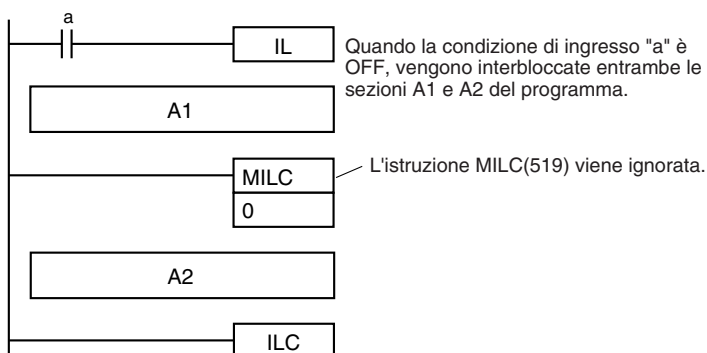


Se un'istruzione MILH(517) si trova tra una coppia IL(002) - ILC(003) e l'interblocco IL(002) è impegnato, la MILH(517) resta priva di effetti. In questo caso, la sezione del programma tra IL(002) e ILC(003) viene interbloccata.

Se l'interblocco IL(002) non è impegnato e la condizione di esecuzione dell'istruzione MILH(517) (in questo caso: b) è impostata su OFF, la sezione del programma tra MILH(517) e ILC(003) viene interbloccata.



Se un'istruzione MILC(519) si trova tra una coppia IL(002) - ILC(003), MILC(519) viene ignorata e tutta la sezione del programma tra IL(002) e ILC(003) verrà interbloccata.

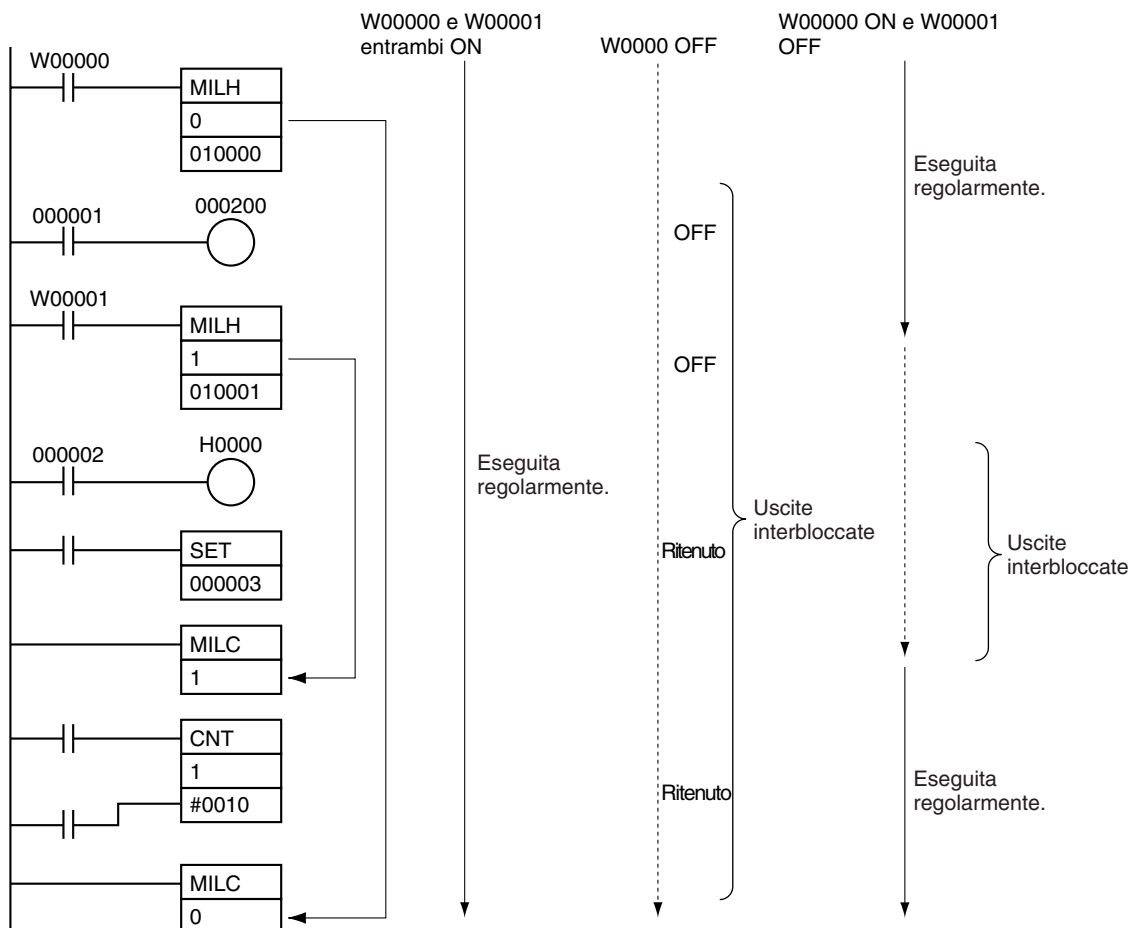


Esempi

Quando W00000 e W00001 sono entrambi impostati su ON, le istruzioni comprese tra MILH(517) con numero di interblocco 0 e MILC(519) con numero di interblocco 0 vengono eseguite regolarmente.

Quando W00000 è impostato su OFF, le istruzioni comprese tra MILH(517) con numero di interblocco 0 e MILC(519) con numero di interblocco 0 vengono bloccate.

Quando W00000 è impostato su ON e W00001 su OFF, le istruzioni comprese tra MILH(517) con numero di interblocco 1 e MILC(519) con numero di interblocco 1 vengono bloccate. Le altre istruzioni vengono eseguite regolarmente.

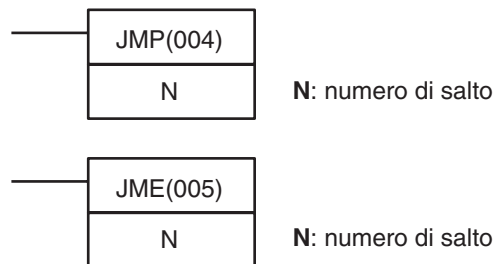


3-5-6 JUMP e JUMP END: JMP(004) e JME(005)

Scopo

Quando la condizione di esecuzione per JMP(004) è OFF, l'esecuzione del programma passa direttamente alla prima istruzione JME(005) nel programma che presenta lo stesso numero di salto. Le istruzioni JMP(004) e JME(005) vengono utilizzate in coppia.

Simboli programmazione ladder



Variazioni

| | | |
|-------------------------|---|----------------|
| Variazioni | Procede al salto se impostata su OFF. Non procede al salto se impostata su ON | JMP(004) |
| Aggiornamento immediato | | Non supportata |

| | | |
|-------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | JME(005) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | Non consentita | OK | OK |

Operandi

N: numero di salto

Il numero di salto deve essere compreso tra 0000 e 03FF o da &0 a &1.023 decimale.

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di salto deve essere compreso tra 0000 e 00FF esadecimale o tra &0 e &255 decimale.

Caratteristiche operando

| Area | N | |
|--|---|---|
| | JMP(004) | JME(005) |
| Area CIO | Da CIO 0000 a CIO 6143 | --- |
| Area di lavoro | Da W000 a W511 | --- |
| Area bit di ritentività | Da H000 a H511 | --- |
| Area bit ausiliaria | Da A000 ad A959 | --- |
| Area del temporizzatore | Da T0000 a T4095 | --- |
| Area del contatore | Da C0000 a C4095 | --- |
| Area DM | Da D00000 a D32767 | --- |
| Area EM senza banco | Da E00000 a E32767 | --- |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | --- |
| Costanti | Da #0000 a #03FF (in formato binario) o da &0 a &1023 (vedere nota) | Da #0000 a #03FF (in formato binario) o da &0 a &1023 (vedere nota) |
| Registri dati | Da DR0 a DR15 | --- |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | --- |

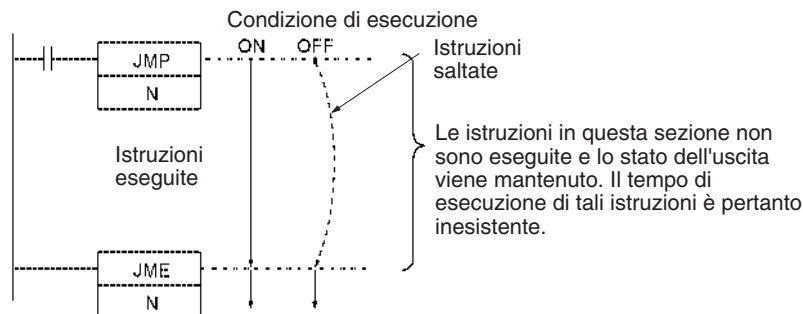
Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, l'intervallo è compreso tra #0000 e #00FF (formato binario) o tra &0 e &1023 (decimale).

Descrizione

Quando la condizione di esecuzione per JMP(004) è ON, non avviene alcun salto e il programma viene eseguito come è stato scritto senza interruzioni.

Quando la condizione di esecuzione per JMP(004) è OFF, l'esecuzione del programma passa direttamente alla prima istruzione JME(005) nel programma che presenta lo stesso numero di salto. Le istruzioni comprese tra JMP(004) e JME(005) non sono eseguite e quindi viene mantenuto lo stato

delle uscite comprese tra JMP(004) e JME(005). Nei programmi a blocchi, le istruzioni comprese tra JMP(004) e JME(005) sono ignorate indipendentemente dallo stato della condizione di esecuzione.



Poiché ogni istruzione compresa tra JMP(004) e JME(005) viene ignorata quando la condizione di esecuzione della prima è OFF, il tempo di ciclo si riduce del tempo di esecuzione complessivo delle istruzioni ignorate. L'elaborazione di NOP(000), invece, viene eseguita per le istruzioni comprese tra JMP0(515) e JME0(516); di conseguenza il tempo di ciclo non viene ridotto come nel caso delle istruzioni saltate.

Nella tabella seguente è riportato il confronto tra le varie istruzioni saltate.

| Elemento | JMP(004) JME(005) | CJP(510) JME(005) | CJPN(511) JME(005) | JMP0(515) JME0(516) |
|---|--|----------------------|-----------------------|-----------------------------|
| Condizione di esecuzione per salto | OFF | ON | OFF | OFF |
| Numero consentito | 1.024 in totale | | | Illimitato |
| Elaborazione dell'istruzione saltata | Non eseguita. | | | elaborazione di NOP(000) |
| Tempo di esecuzione dell'istruzione saltata | Nessuno | | | Come le istruzioni NOP(000) |
| Stato delle uscite (bit e canali) saltate | Bit e canali mantengono lo stato precedente | | | |
| Stato dei temporizzatori operativi saltati | I temporizzatori operativi continuano a funzionare | | | |
| Elaborazione nei programma a blocchi | Ignora sempre | Ignora quando ON | Ignora quando OFF | Non consentita |

Flag (JMP)

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se N non rientra nell'intervallo specificato 0000 - 03FF (vedere nota).</p> <p>ON se nel programma esiste una JMP(004) ma non una JME(005) con lo stesso numero di salto.</p> <p>ON se nel task esiste una JMP(004) ma non una JME(005) con lo stesso numero di salto.</p> <p>OFF in tutti gli altri casi.</p> |

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, l'intervallo è compreso tra 0 e 255 o tra 0000 e 00FF esadecimale.

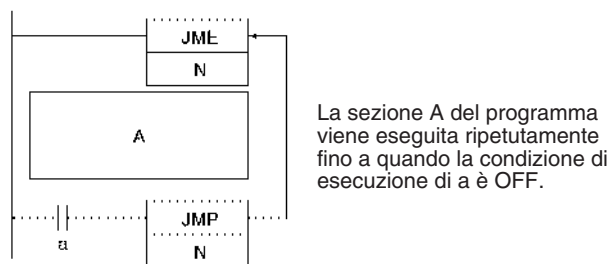
Avvertenze

Tutte le uscite (bit e canali) nelle istruzioni saltate mantengono lo stato precedente. I temporizzatori in funzione (TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540) e TMHHX(552)) continuano a funzionare poiché i valori attuali (PV) vengono aggiornati anche quando l'istruzione del temporizzatore non è in fase di esecuzione.

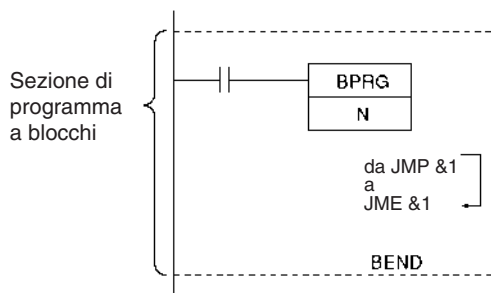
Se esistono due o più istruzioni JME(005) con lo stesso numero di salto, sarà valida soltanto l'istruzione con l'indirizzo minore. La JME(005) con l'indirizzo di programma maggiore sarà ignorata.

Se JME(005) precede JMP(004) nel programma, l'istruzione che si trova in mezzo viene eseguita ripetutamente fino a quando lo stato di JMP(004) resta impostato su OFF. Se la condizione di esecuzione non passa a ON oppure se

END(001) non viene eseguita entro il tempo di ciclo massimo, si verificherà un errore di superamento del tempo di ciclo massimo.



Nei programmi a blocchi, le istruzioni tra JMP(004) e JME(005) vengono comunque saltate indipendentemente dallo stato della condizione di esecuzione di JMP(004).



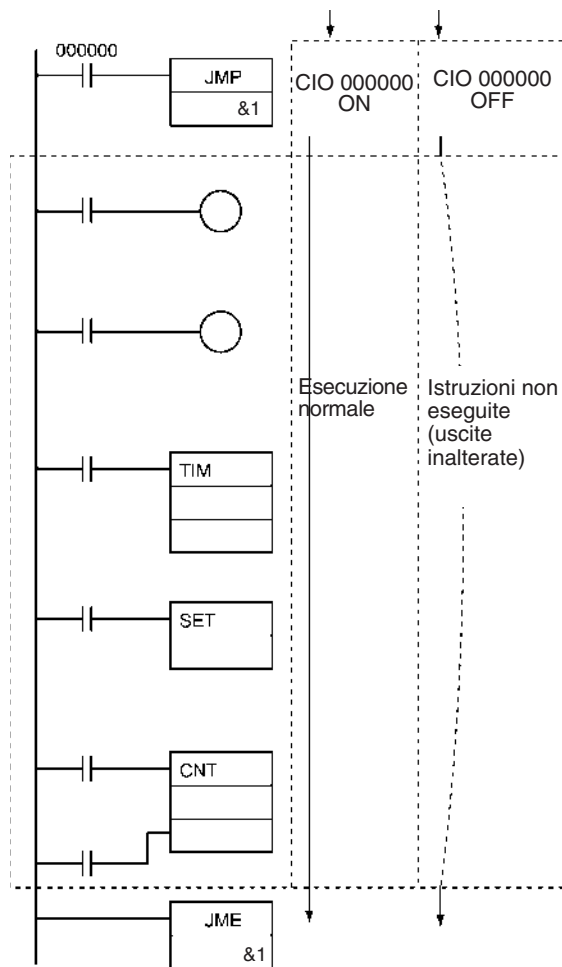
La coppia JMP(004) - JME(005) deve trovarsi nello stesso task poiché non sono consentiti salti tra task. Se un'istruzione JME(005) non è programmata nello stesso task della corrispondente istruzione JMP(004), si verifica un errore.

Quando DIFU(013), DIFD(014) e le istruzioni differenziate vengono programmate tra JMP(004) e JME(005), il loro funzionamento non dipende soltanto dallo stato della condizione di esecuzione. Quando DIFU(013), DIFD(014) o un'istruzione differenziata vengono eseguite in una sezione saltata subito dopo l'attivazione della condizione di esecuzione per JMP(004), la condizione di esecuzione per DIFU(013), DIFD(014) o per l'istruzione differenziata verrà confrontata con la condizione di esecuzione esistente prima che il salto diventasse effettivo, ossia prima che venisse disattivata la condizione di esecuzione per JMP(004).

Esempi

Funzionamento di base

Nell'esempio che segue quando CIO 000000 è impostato su OFF, le istruzioni comprese tra JMP(004) e JME(005) non vengono eseguite e le uscite mantengono lo stato precedente. Nell'esempio che segue quando CIO 000000 è impostato su ON, le istruzioni comprese tra JMP(004) e JME(005) vengono eseguite regolarmente.



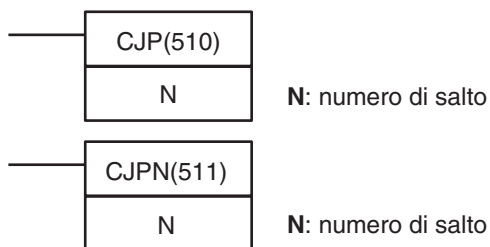
3-5-7 CONDITIONAL JUMP: CJP(510)/CJPN(511)

Scopo

Il funzionamento di CJP(510) è sostanzialmente opposto a quello di JMP(004). Quando la condizione di esecuzione per CJP(510) è ON, l'esecuzione del programma passa direttamente alla prima istruzione JME(05) nel programma avente lo stesso numero di salto. Le istruzioni CJP(510) e JME(005) vengono utilizzate in coppia.

Il funzionamento di CJPN(511) è quasi uguale a quello di JMP(004). Quando la condizione di esecuzione per CJP(004) è OFF, l'esecuzione del programma passa direttamente alla prima istruzione JME(005) nel programma avente lo stesso numero di salto. Le istruzioni CJPN(511) e JME(005) vengono utilizzate in coppia.

Simboli programmazione ladder



Variazioni

| | | |
|-------------------------|---|----------------|
| Variazioni | Procede al salto se impostata su ON. Non procede al salto se impostata su OFF | CJP(510) |
| Aggiornamento immediato | | Non supportata |

| | | |
|-------------------------|---|----------------|
| Variazioni | Procede al salto se impostata su OFF. Non procede al salto se impostata su ON | CJPN(511) |
| Aggiornamento immediato | | Non supportata |

| | | |
|-------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | JME(005) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | Non consentita | OK | OK |

Operandi

N: numero di salto

Il numero di salto deve essere compreso tra 0000 e 03FF o da 0 a 1.023 decimale.

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di salto deve essere compreso tra 0000 e 00FF esadecimale o tra &0 e &255 decimale.

Caratteristiche operando

| Area | N | | |
|--|--|-----------|---|
| | CJP(510) | CJPN(511) | JME(005) |
| Area CIO | Da CIO 0000 a CIO 6143 | | --- |
| Area di lavoro | Da W000 a W511 | | --- |
| Area bit di ritentività | Da H000 a H511 | | --- |
| Area bit ausiliaria | Da A000 ad A959 | | --- |
| Area del temporizzatore | Da T0000 a T4095 | | --- |
| Area del contatore | Da C0000 a C4095 | | --- |
| Area DM | Da D00000 a D32767 | | --- |
| Area EM senza banco | Da E00000 a E32767 | | --- |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | --- |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | --- |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | --- |
| Costanti | Da #0000 a #03FF (in formato binario) o da &0 a &1023 (vedere nota) | | Da #0000 a #03FF (in formato binario) o da &0 a &1023 (vedere nota) |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | | --- |

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, l'intervallo è compreso tra #0000 e #00FF (formato binario) o tra &0 e &1023 (decimale).

Descrizione

Il funzionamento di CJP(510) e CJPN(511) differisce soltanto nella condizione di esecuzione. CJP(510) passa alla prima JME(005) quando la condizione di

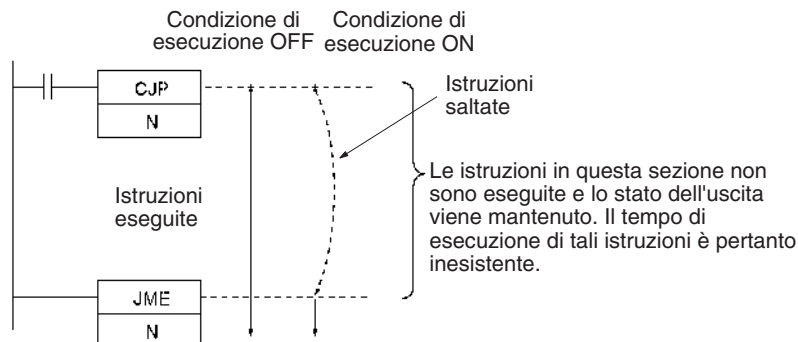
esecuzione è impostata su ON e CJP(511) passa alla prima JME(005) quando la condizione di esecuzione è impostata su OFF.

Poiché le istruzioni saltate non vengono eseguite, il tempo di ciclo si riduce del tempo di esecuzione complessivo di tali istruzioni.

Funzionamento di CJP(510)

Quando la condizione di esecuzione per CJP(510) è OFF, non avviene alcun salto e il programma viene eseguito come è stato scritto senza interruzioni.

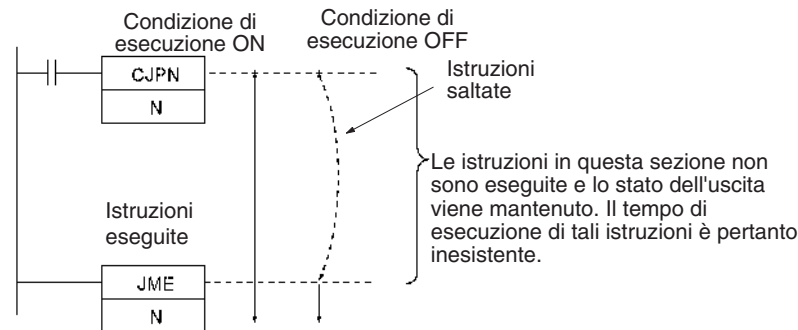
Quando la condizione di esecuzione per CJP(510) è ON, l'esecuzione del programma passa direttamente alla prima istruzione JME(05) nel programma avente lo stesso numero di salto.



Funzionamento di CJPN(511)

Quando la condizione di esecuzione per CJPN(511) è ON, non avviene alcun salto e il programma viene eseguito come è stato scritto senza interruzioni.

Quando la condizione di esecuzione per CJPN(511) è OFF, l'esecuzione del programma passa direttamente alla prima istruzione JME(005) nel programma avente lo stesso numero di salto.



Flag

Nella tabella seguente sono riportati i flag su cui influiscono CJP(510) e CJPN(511).

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON in assenza di un'istruzione JME(005) con lo stesso numero di salto di CJP(510) o di CJPN(511) (vedere nota). ON se N non rientra nell'intervallo specificato 0000 - 03FF ON in presenza di un'istruzione CJP(510) o CJPN(511) in un task senza che vi sia una JME(005) con lo stesso numero di salto. OFF in tutti gli altri casi. |

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di salto deve essere compreso tra 0 e 255 o tra 0000 e 00FF esadecimale.

Avvertenze

Tutte le uscite (bit e canali) nelle istruzioni saltate mantengono lo stato precedente. I temporizzatori in funzione (TIM, TIMX(550), TIMH(015), TIMHX(551),

TMHH(540) e TMHHX(552)) continuano a funzionare poiché i valori attuali (PV) vengono aggiornati anche quando l'istruzione del temporizzatore non è in fase di esecuzione.

Se esistono due o più istruzioni JME(005) con lo stesso numero di salto, sarà valida soltanto l'istruzione con l'indirizzo minore. La JME(005) con l'indirizzo di programma maggiore sarà ignorata.

Se JME(005) precede l'istruzione CJP(510) o CJPN(511) nel programma, l'istruzione che si trova nel mezzo viene eseguita ripetutamente fino a quando la condizione di esecuzione resta impostata su OFF (CJP(510)) o su ON (CJPN(511)). Se il salto non viene completato modificando la condizione di esecuzione durante l'esecuzione di END(001), si verificherà un errore di superamento del tempo di ciclo massimo.

Nei programmi a blocchi le istruzioni CJP(510) o CJPN(511) funzionano regolarmente.

Quando la condizione di esecuzione per CJP(510) è ON o quando la condizione di esecuzione per CJPN(511) è OFF, l'esecuzione del programma passerà direttamente all'istruzione JME senza eseguire le istruzioni comprese tra CJP(510)/CJPN(511) e JME. Il tempo di esecuzione per queste istruzioni è nullo e quindi il tempo di ciclo viene ridotto.

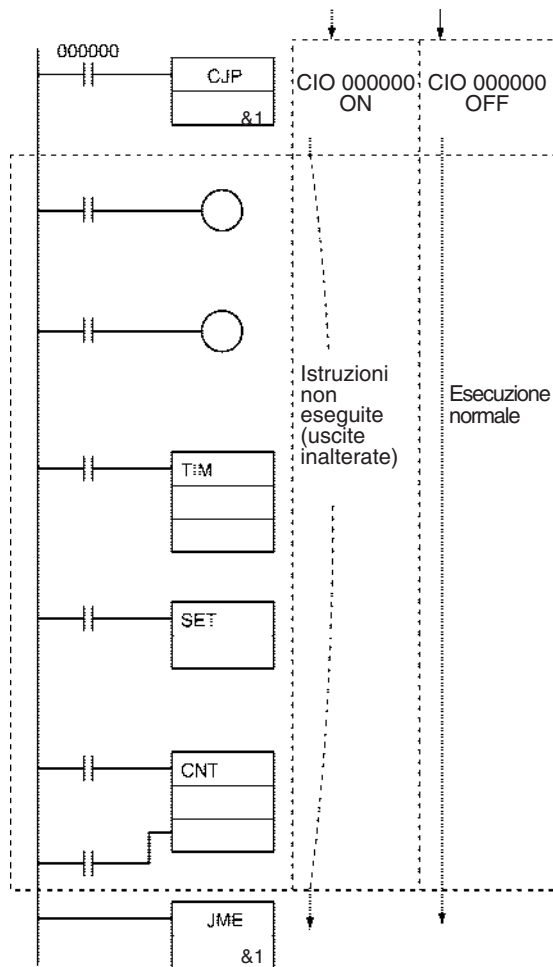
Quando la condizione di esecuzione per JMP0 è OFF, l'elaborazione NOP viene eseguita tra le istruzioni JMP0 e JME0 e necessita di tempo di esecuzione. Di conseguenza, il tempo di ciclo non verrà ridotto.

Quando un'istruzione CJP(510) o CJPN(511) sono programmate in un task, è necessario che esista una JME(005) con lo stesso numero di salto, poiché non sono consentiti salti tra i task. Se un'istruzione JME(005) corrispondente non è programmata nello stesso task, si verificherà un errore.

Quando DIFU(013), DIFD(014) e le istruzioni differenziate vengono programmate in una sezione del programma saltata, il loro funzionamento di non dipende soltanto dallo stato della condizione di esecuzione. Quando DIFU(013), DIFD(014) o un'istruzione differenziata vengono eseguite in una sezione saltata subito dopo la disattivazione della condizione di esecuzione per CJP(510) (ON per CJPN(511)), la condizione di esecuzione per DIFU(013), DIFD(014) o per l'istruzione differenziata verrà confrontata con la condizione di esecuzione esistente prima che il salto diventasse effettivo.

Esempio

Nell'esempio che segue quando CIO 000000 è impostato su ON, le istruzioni comprese tra CJP(510) e JME(005) non vengono eseguite e le uscite mantengono lo stato precedente. Nell'esempio che segue quando CIO 000000 è impostato su OFF, le istruzioni comprese tra CJP(510) e JME(005) vengono eseguite regolarmente.



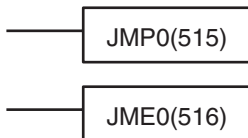
Nota Per CJPN(511) lo stato ON/OFF di CIO 000000 sarebbe invertito.

3-5-8 MULTIPLE JUMP e JUMP END: JMP0(515) e JME0(516)

Scopo

Quando la condizione di esecuzione per JMP0(515) è impostata su OFF, tutte le istruzioni del programma comprese tra JMP0(515) e l'istruzione JME0(516) successiva vengono elaborate come NOP(000). Utilizzare JMP0(515) e JME0(516) in coppia. Nel programma è possibile utilizzare un numero illimitato di istruzioni JMP0(515)/JME0(516) in coppia.

Simboli programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Procede al salto se impostata su OFF. Non procede al salto se impostata su ON | JMP0(515) |
| Aggiornamento immediato | | Non supportata |
| Variazioni | Eseguita a ogni ciclo con condizione ON | JME0(516) |
| Aggiornamento immediato | | Non supportata |

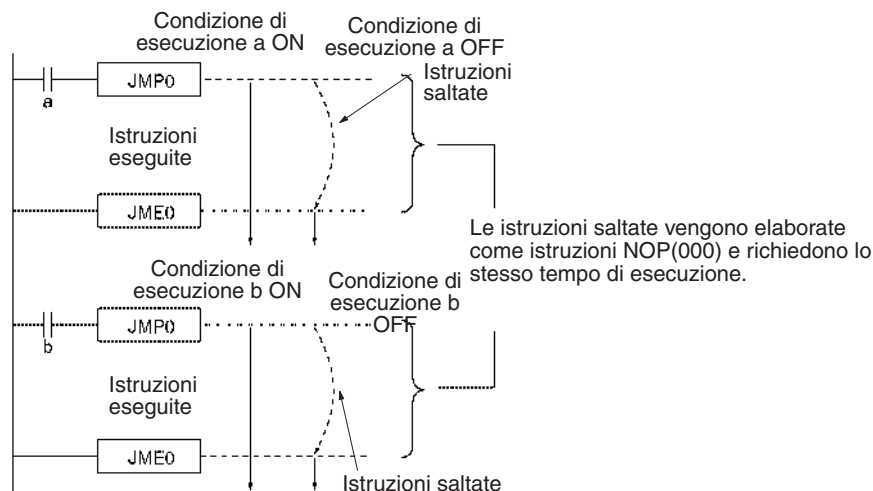
Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| Non consentita | Non consentita | OK | OK |

Descrizione

Quando la condizione di esecuzione per JMP0(515) è ON, non avviene alcun salto e il programma viene eseguito come è stato scritto senza interruzioni.

Quando la condizione di esecuzione per JMP0(515) è impostata su OFF, tutte le istruzioni del programma comprese tra JMP0(515) e l'istruzione JME0(516) successiva vengono elaborate come NOP(000). A differenza di JMP(004), CJP(510) e CJPN(511), JMP0(515) non utilizza numeri di salto, per cui queste istruzioni possono essere collocate in qualsiasi punto del programma.



A differenza di JMP(004), CJP(510) e CJPN(511), che passano direttamente alla prima istruzione JME(005), tutte le istruzioni del programma comprese tra JMP0(515) e JME0(516) vengono eseguite come NOP(000). Il tempo di esecuzione delle istruzioni saltate si riduce, ma non viene annullato. Le istruzioni saltate stesse non vengono eseguite e le loro uscite (bit e canali) mantengono lo stato precedente.

Avvertenze

Nel programma è possibile utilizzare coppie multiple di istruzioni JMP0(515) e JME0(516), che tuttavia non possono essere nidificate.

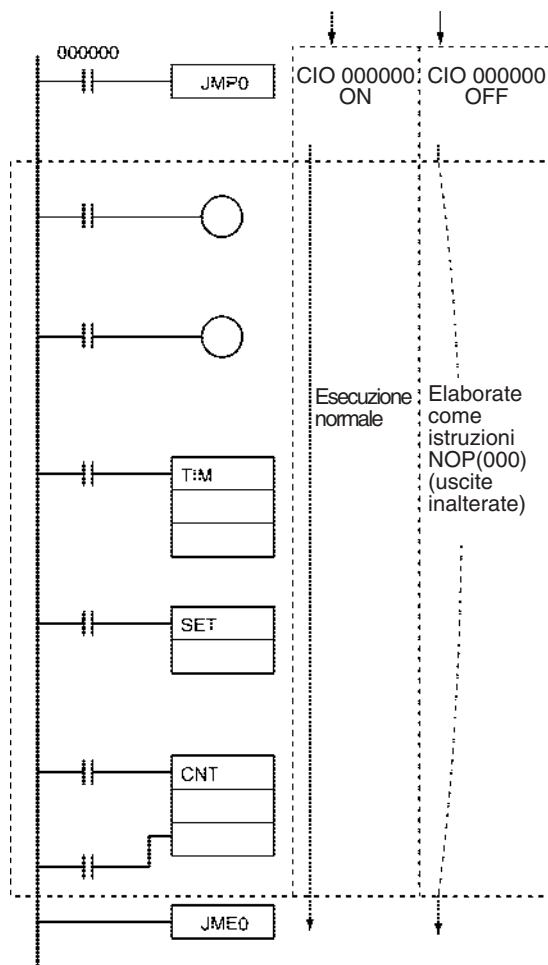
JMP0(515) e JME0(516) non possono essere utilizzate in programmi a blocchi.

È necessario che le coppie di JMP0(515) e JME0(516) siano negli stessi task poiché non sono consentiti salti tra i task.

Quando DIFU(013), DIFD(014) e le istruzioni differenziate vengono programmate tra JMP(515) e JME(516), il loro funzionamento non dipende soltanto dallo stato della condizione di esecuzione. Quando DIFU(013), DIFD(014) o un'istruzione differenziata vengono eseguite in una sezione saltata subito dopo l'attivazione della condizione di esecuzione per JMP(515), la condizione di esecuzione per DIFU(013), DIFD(014) o per l'istruzione differenziata viene confrontata con la condizione di esecuzione esistente prima che il salto diventasse effettivo, ossia prima che venisse disattivata la condizione di esecuzione per JMP(515).

Esempio

Nell'esempio che segue quando CIO 000000 è impostato su OFF, le istruzioni comprese tra JMP0(515) e JME0(516) vengono processate come istruzioni NOP(000) e le uscite mantengono lo stato precedente. Nell'esempio che segue quando CIO 000000 è impostato su ON, le istruzioni comprese tra JMP0(515) e JME0(516) vengono eseguite regolarmente.

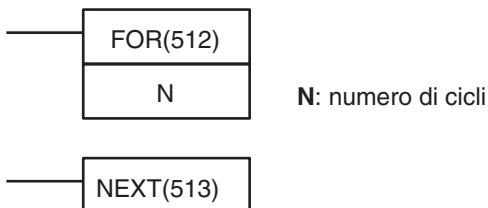


3-5-9 FOR-NEXT LOOPS: FOR(512)/NEXT(513)

Scopo

Le istruzioni comprese tra FOR(512) e NEXT(513) vengono ripetute per un numero di volte specificato. Le istruzioni FOR(512) e NEXT(513) devono essere utilizzate in coppia.

Simboli programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FOR(512) |
| | Eseguita a ogni ciclo con condizione OFF | NEXT(513) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Operandi

N: numero di cicli

Il numero di cicli deve essere compreso tra 0000 e FFFF o da 0 a 65.535 decimale.

Caratteristiche operando

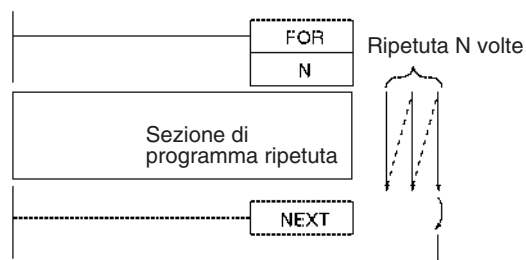
| Area | N |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Da #0000 a #FFFF (in formato binario) o da &0 a &65.535 |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

Le istruzioni comprese tra FOR(512) e NEXT(513) vengono eseguite N volte e quindi l'esecuzione del programma prosegue con l'istruzione successiva a NEXT(513). È possibile utilizzare l'istruzione BREAK(514) per annullare il ciclo.

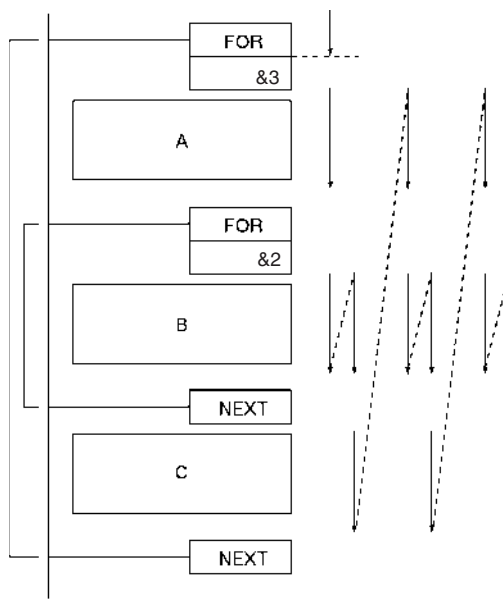
Se N è impostato su 0, le istruzioni comprese tra FOR(512) e NEXT(513) vengono elaborate come istruzioni NOP(000).

È possibile utilizzare i cicli per elaborare tabelle di dati ricorrendo soltanto a un minimo di programmazione.

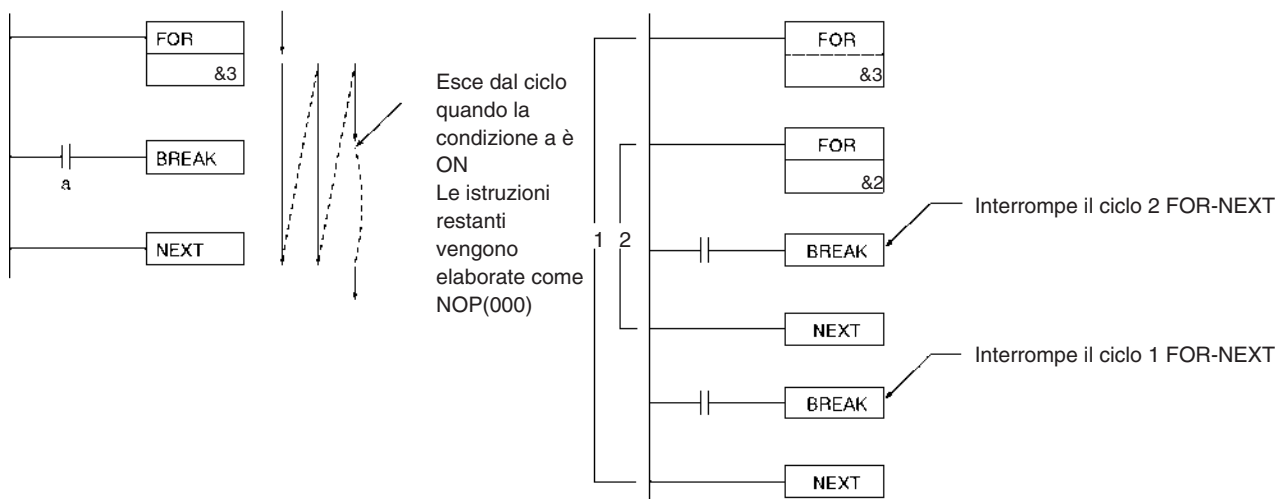


I cicli FOR-NEXT possono essere nidificati fino a un massimo di 15 livelli. Nell'esempio che segue le sezioni A, B e C del programma vengono eseguite come indicato:

A → B → B → C, A → B → B → C e A → B → B → C



Utilizzare BREAK(514) per uscire da un ciclo FOR-NEXT. Per uscire da cicli nidificati è necessario un numero di istruzioni BREAK(514) pari a quello dei livelli nidificati. Le altre istruzioni successive a BREAK(514) all'interno del ciclo vengono elaborate come istruzioni NOP(000).



Metodi alternativi per l'esecuzione di cicli

Per ripetere una sezione di programma fino all'indicazione di una determinata condizione di esecuzione, è possibile procedere in due modi.

1,2,3...

1. Ciclo FOR-NEXT con BREAK

Avviare un ciclo FOR-NEXT con un massimo di N ripetizioni. Programmare BREAK(514) all'interno del ciclo con la condizione di esecuzione prescelta. Se si immette la condizione di esecuzione, il ciclo terminerà prima di N ripetizioni.

2. Ciclo JME(005)-JMP(004)

Programmare un ciclo con JME(005) prima di JMP(004). Le istruzioni comprese tra JME(005) e JMP(004) vengono eseguite ripetutamente fino a quando la condizione di esecuzione per JMP(004) resta impostata su OFF. Se la condizione di esecuzione non passa a ON oppure se END(001) non viene eseguita entro il tempo di ciclo massimo, si verificherà un errore di superamento del tempo di ciclo massimo.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se i cicli nidificati sono più di 15. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF |
| Flag negativo | N | OFF |

Avvertenze

Programmare FOR(512) e NEXT(513) nello stesso task. Se queste istruzioni non sono nello stesso task, l'esecuzione non verrà ripetuta.

È possibile che un'istruzione come JMP(004) venga eseguita all'interno di un ciclo FOR-NEXT, ma non potrebbe passare oltre.

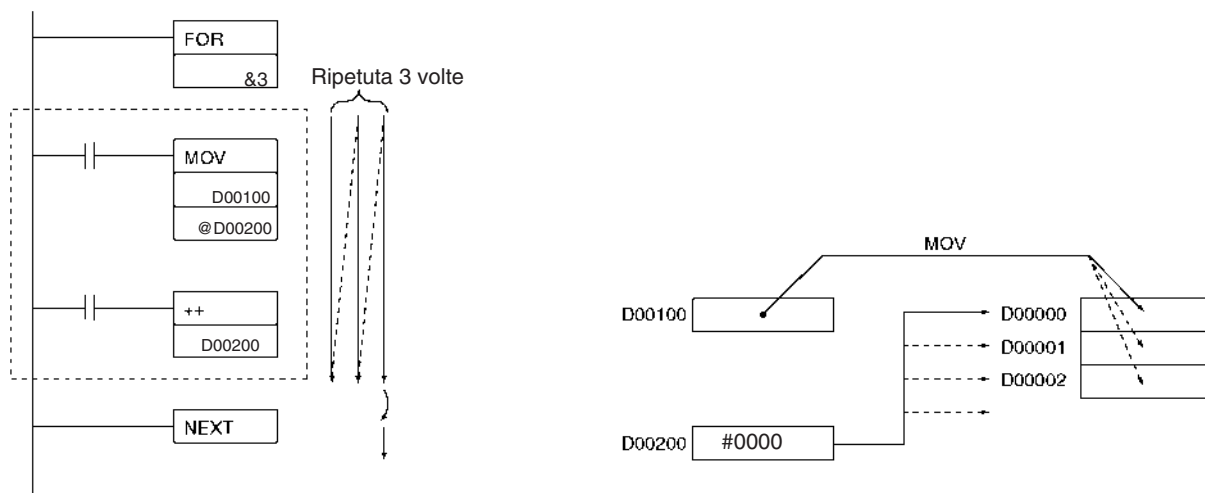
All'interno dei cicli FOR-NEXT non è possibile utilizzare le seguenti istruzioni:

- Istruzioni di programmazione a blocchi
- MULTIPLE JUMP e JUMP END: JMP(515) e JME(516)
- STEP DEFINE e STEP START: STEP(008) e SNXT(009)

Nota Se un loop si ripete in un ciclo e un bit differenziato viene utilizzato nel ciclo FOR-NEXT, tale bit sarà sempre impostato su ON o sempre su OFF all'interno di quel ciclo.

Esempio

Nell'esempio che segue la sezione di programma ciclica trasferisce il contenuto di D00100 all'indirizzo specificato in D00200 e quindi incrementa il contenuto di 1.



3-5-10 BREAK LOOP: BREAK(514)

Scopo

Viene programmata in un ciclo FOR-NEXT per interrompere l'esecuzione del ciclo quando si verifica una determinata condizione. Le rimanenti istruzioni all'interno del ciclo vengono elaborate come istruzioni NOP(000)

Simbolo programmazione ladder



Variazioni

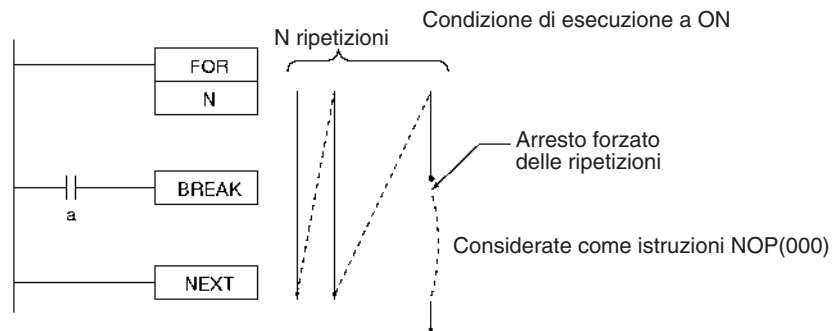
| Variazioni | Eseguita a ogni ciclo con condizione ON | BREAK(514) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Descrizione

Per interrompere il ciclo FOR-NEXT all'esecuzione di BREAK(514), programmare quest'ultima istruzione tra FOR(512) e NEXT(513). Quando BREAK(514) viene eseguita, le restanti istruzioni fino a NEXT(513) vengono elaborate come NOP(000).



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---------------|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | OFF |
| Flag negativo | N | OFF |

Avvertenze

Un'istruzione BREAK(514) interrompe un solo ciclo e quindi per uscire da cicli nidificati è necessario un numero di istruzioni BREAK(514) pari a quello dei livelli nidificati.

BREAK(514) può essere utilizzata esclusivamente in un ciclo FOR-NEXT.

3-6 Istruzioni di temporizzatore e contatore

In questa sezione vengono illustrate le istruzioni utili a definire e gestire temporizzatori e contatori.

| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|---------------------|------------------|-----------------|--------|
| TIMER | TIM/TIMX | ---/551 | 235 |
| HIGH-SPEED TIMER | TIMH/TIMHX | 015/551 | 240 |
| ONE-MS TIMER | TMHH/TIMHHX | 540/552 | 244 |
| ACCUMULATIVE TIMER | TTIM/TTIMX | 087/555 | 247 |
| LONG TIMER | TIML/TIMLX | 542/553 | 251 |
| MULTI-OUTPUT TIMER | MTIM/MTIMX | 543/554 | 254 |
| COUNTER | CNT/CNTX | ---/546 | 260 |
| REVERSIBLE COUNTER | CNTR/CNTRX | 012/548 | 263 |
| RESET TIMER/COUNTER | CNR/CNRX | 545/547 | 267 |

Metodi di aggiornamento del valore attuale (PV) di temporizzatori e contatori

■ Panoramica

Tutte le istruzioni per temporizzatori e contatori supportati da CPU CS1 e CJ1 utilizzano dati BCD e tramite BCD vengono inoltre specificati tutti i valori impostati. Per altre CPU serie CS e serie CJ (ossia CPU CS1/H, CJ1/H, CJ1M e CS1D) è possibile impostare il metodo di aggiornamento sia in BCD che in formato binario (vedere note 1 e 2).

L'utilizzo di dati binari invece di BCD consente di potenziare l'intervallo del valore impostato (SV) di temporizzatori e contatori da 0-9999 a 0-65535, nonché di ricorrere a dati binari calcolati con altre istruzioni direttamente come un valore impostato (SV) del temporizzatore/contatore. Il metodo di aggiornamento è valido anche quando si procede a un'impostazione indiretta di un valore impostato (SV), ossia quando si utilizza il contenuto del canale della memoria. In altri termini, il contenuto del canale indirizzato viene assunto come valore BCD o binario in base al metodo di aggiornamento impostato.

Per ulteriori informazioni sui metodi di aggiornamento, fare riferimento al *Manuale di programmazione dei PLC della serie CS/CJ*.

- Nota**
1. Nelle CPU CS1/H e CJ1/H prodotte entro il 31 maggio 2002, le istruzioni binarie vengono visualizzate sulla console di programmazione con il codice mnemonico dell'istruzione equivalente dell'operazione BCD (ad esempio, TIMX0 &16 viene visualizzata come TIM0 &16). L'istruzione, tuttavia, funziona utilizzando la modalità binaria.
 2. È possibile selezionare il metodo di aggiornamento solo con CX-Programmer versione 3.0 o successiva e non con la versione 2.1 o precedente, né tramite la console di programmazione.
 3. Con CX-Programmer versione 2.1 o precedente non è possibile leggere i programmi utente che utilizzano la modalità binaria di aggiornamento. Per leggere tali programmi è necessario passare alla modalità BCD.

■ Istruzioni applicabili

| Classificazione | Istruzione | Codice mnemonico | |
|--|-----------------------|------------------|------------|
| | | BCD | Binario |
| Istruzioni del temporizzatore/ contatore | TIMER | TIM | TIMX(550) |
| | HIGH-SPEED TIMER | TIMH(015) | TIMHX(551) |
| | ONE-MS TIMER | TMHH(540) | TMHHX(552) |
| | ACCUMULATIVE TIMER | TTIM(087) | TTIMX(555) |
| | LONG TIMER | TIML(542) | TIMLX(553) |
| | MULTI-OUTPUT TIMER | MTIM(543) | MTIMX(554) |
| | COUNTER | CNT | CNTX(546) |
| | REVERSIBLE COUNTER | CNTR(012) | CNTRX(548) |
| | RESET TIMER/COUNTER | CNR(545) | CNRX(547) |
| Istruzioni di programmazione a blocchi | TIMER WAIT | TIMW(813) | TIMWX(816) |
| | HIGH-SPEED TIMER WAIT | TMHW(815) | TMHWX(817) |
| | COUNTER WAIT | CNTW(814) | CNTWX(818) |

Caratteristiche di base del temporizzatore

Nella tabella che segue vengono riportate le caratteristiche di base dei temporizzatori.

| Elemento | TIM/TIMX(550) | TIMH(015)/ TIMHX(551) | TMHH(540)/ TMHHX(552) | TTIM(087)/ TTIMX(555) | TIML(542)/ TIMLX(553) | MTIM(543)/ MTIMX(554) |
|------------------------------|--------------------------------|----------------------------------|----------------------------------|-----------------------------------|--|-----------------------------------|
| Metodo di temporizzazione | Decremento | Decremento | Decremento | Incremento | Decremento | Incremento |
| Unità di temporizzazione | 0,1 s | 0,01 s | 0,001 s | 0,1 s | 0,1 s | 0,1 s |
| SV massimo | TIM 999,9 s TIMX: 6.553,5 s | TIMH: 99,99 s TIMHX: 655.35 s | TMHH: 9.999 s TMHHX: 65.535 s | TTIM: 999,9 s TTIMX: 6,553.5 s | TIML: 115 giorni TIMLX: 49,710 giorni | MTIM: 999,9 s MTIMX: 6,553.5 s |
| Istruzione/uscite | 1 | 1 | 1 | 1 | 1 | 8 |
| Numeri temporizzatore | Utilizzati | Utilizzati | Utilizzati | Utilizzati | Non utilizzati | Non utilizzati |
| Aggiornamento flag di compl. | All'esecuzione | All'esecuzione | Ad interrupt a ciascun ms | All'esecuzione | All'esecuzione | All'esecuzione |

| Elemento | TIM/TIMX(550) | TIMH(015)/ TIMHX(551) | TMHH(540)/ TMHHX(552) | TTIM(087)/ TTIMX(555) | TIML(542)/ TIMLX(553) | MTIM(543)/ MTIMX(554) |
|-------------------------------------|----------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Aggiornamento PV del temporizzatore | Vedere nota 1 | Vedere nota 2 | A ciascun ms | All'esecuzione | All'esecuzione | All'esecuzione |
| Valore dopo reset | Flag di compl. | OFF | OFF | OFF | OFF | OFF |
| | PV | SV | SV | SV | 0 | 0 |

- Nota**
1. I valori attuali (PV) di TIM sono aggiornati all'esecuzione, al termine dell'esecuzione del programma a ogni ciclo oppure ogni 80 ms ad interrupt se il tempo di ciclo supera 80 ms.
 2. I PV di TIMH(015)/TIMHX(551) vengono aggiornati all'esecuzione, al termine dell'esecuzione del programma a ogni ciclo e ogni 10 ms ad interrupt.

Funzionamento del temporizzatore

Nella tabella seguente sono indicati gli effetti delle condizioni operative e di programmazione sul funzionamento dei temporizzatori.

| Elemento | TIM/ TIMX(550) | TIMH(015)/ TIMHX(551) | TMHH(540)/ TMHHX(552) | TTIM(087)/ TTIMX(555) | TIML(542)/ TIMLX(553) | MTIM(543)/ MTIMX(554) | |
|---|---|--------------------------|--------------------------|---|---------------------------------|--------------------------------|-----|
| Cambiamento modalità operativa | PV = 0 Flag di completamento = OFF | | | | --- | --- | |
| Reset/caduta tensione | PV = 0 Flag di completamento = OFF | | | | --- | --- | |
| Esecuzione di CNR(545)/ CNRX(547) | Binario: PV = FFFF, flag di completamento = OFF BCD: PV = FFFF o 9999, flag di completamento = OFF | | | | Non applicabile | Non applicabile | |
| Funzionamento in una sezione del programma saltata (JMP(004)-JME(005)) | I temporizzatori operativi continuano a funzionare | | | Lo stato del temporizzatore viene mantenuto | | | |
| Funzionamento in una sezione del programma interbloccata (IL(002)-ILC(003)) | PV = SV Flag di completamento = OFF | | | Stato temporizzatore mantenuto | PV = SV Flag di compl. = OFF | Stato temporizzatore mantenuto | |
| Impostazione forzata | Flag di compl. | ON | | | | --- | --- |
| | PV | Impostati su 0 | | | | --- | --- |
| Ripristino forzato | Flag di compl. | OFF | | | | --- | --- |
| | PV | Reset su SV | | Impostati su 0 | --- | --- | |

3-6-1 TIMER: TIM/TIMX(550)

Scopo

TIM o TIMX(550) azionano un temporizzatore decrementale con unità di 0,1 s. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 999,9 s per TIM e tra 0 e 6.553,5 s per TIMX(550). La precisione del temporizzatore è compresa tra 0 e 0,01 s.

Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

Simbolo programmazione ladder

| Metodo di aggiornamento PV | Simbolo | Operandi | | | |
|----------------------------|--|-----------|---|---|---|
| BCD | <table border="1"> <tr><td>TIM</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: numero di temporizzatore S: valore impostato</p> | TIM | N | S | N: da 0000 a 4095 (decimale) S: da #0000 a #9999 (BCD) |
| TIM | | | | | |
| N | | | | | |
| S | | | | | |
| Binario | <table border="1"> <tr><td>TIMX(550)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> </table> <p>N: numero di temporizzatore S: valore impostato</p> | TIMX(550) | N | S | N: da 00000 a 4095 (decimale) S: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| TIMX(550) | | | | | |
| N | | | | | |
| S | | | | | |

Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TIM/TIMX(550) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

N: numero del temporizzatore

Il numero del temporizzatore deve essere compreso tra 0000 e 4095 (decimale).

S: valore impostato

Il valore impostato deve essere compreso tra #0000 e 9999 (BCD).

Se viene impostato su #0000, il flag di completamento passerà a ON all'esecuzione di TIM/TIMX(550).

Caratteristiche operando

| Area | N | S |
|--------------------------------------|---------------------------|---|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritentività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | Da 0000 a 4095 (decimale) | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |

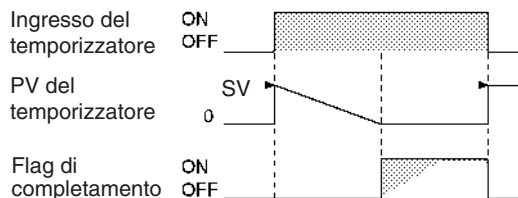
| Area | N | S |
|--|--|--|
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_032767 (n = da 0 a C) |
| Costanti | --- | BCD: Da #0000 a 9999 (BCD) Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

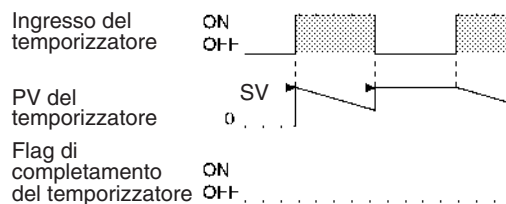
Quando l'ingresso del temporizzatore è impostato su OFF, viene ripristinato il temporizzatore specificato da N, ossia il valore attuale (PV) del temporizzatore è ripristinato in base al valore impostato (SV) e il flag di completamento passa a OFF.

Quando l'ingresso del temporizzatore passa da OFF a ON, TIM/TIMX(550) inizia a decrementare il valore attuale (PV). Il PV continua a diminuire fino a quando l'ingresso del temporizzatore resta impostato su ON; il flag di completamento del temporizzatore viene attivato una volta che il PV ha raggiunto 0000.

Lo stato del PV e del flag di completamento del temporizzatore viene mantenuto dopo il timeout del temporizzatore. Per avviare nuovamente il temporizzatore, è necessario impostarne l'ingresso su OFF e quindi ancora su ON oppure sostituire il valore attuale (PV) del temporizzatore con un valore diverso da zero (ad esempio con MOV(021)).



Nel seguente diagramma di funzionamento viene illustrato il comportamento del valore attuale (PV) e del flag di completamento del temporizzatore quando l'ingresso del temporizzatore stesso è impostato su OFF prima del timeout.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se N è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un temporizzatore. ON se in modalità BCD e se S non contiene dati BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

I numeri di temporizzatore vengono condivisi dalle istruzioni TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) e TMHWX(817). Se due temporizzatori condividono lo stesso numero di temporizzatore, ma non sono utilizzati contemporaneamente, viene generato un errore di duplicazione durante il controllo del programma; il funzionamento dei temporizzatori sarà comunque regolare. I temporizzatori che condividono lo stesso numero di temporizzatore non funzionano in modo corretto se vengono utilizzati contemporaneamente.

I temporizzatori creati con numeri di temporizzatore compresi tra 2048 e 4095 non funzionano in modo regolare quando il tempo di ciclo della CPU supera 80 ms. In tal caso è opportuno utilizzare numeri di temporizzatore compresi tra 0000 e 2047.

Anche quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 0000 a 2047 viene aggiornato. Quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 2048 a 4095 viene mantenuto.

I temporizzatori vengono reimpostati o messi in pausa nei casi indicati nella tabella qui di seguito. Quando un temporizzatore è reimpostato il valore attuale corrispondente viene a sua volta ripristinato in base al valore impostato e il flag di completamento passa a OFF.

| Condizione | PV | Flag di completamento |
|--|-----------------------------------|------------------------------|
| Modifica della modalità operativa da RUN o MONITOR a PROGRAM o viceversa. ¹ | 0000 | OFF |
| Interruzione e reset dell'alimentazione ² | 0000 | OFF |
| Esecuzione di CNR(545)/CNRX(547), le istruzioni di reimpostazione RESET TIMER/COUNTER ³ | BCD: 9999 Binario: FFFF | OFF |
| Funzionamento in una sezione del programma interbloccata (IL(002)-ILC(003)) | Reset su SV | OFF |
| Funzionamento in una sezione del programma saltata (JMP(004)-JME(005)) | PV continua a essere decrementato | Mantiene lo stato precedente |

- Nota**
1. Se il bit di ritentività IOM (A50012) è stato impostato su ON, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti alla modifica della modalità operativa.
 2. Se il bit di ritentività IOM (A50012) è stato impostato su ON e lo stato dello stesso bit è protetto nella configurazione del PLC, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti anche quando l'alimentazione è interrotta.
 3. Quando TIM/TIMX(550) viene eseguita, il valore attuale (PV) è impostato in base al valore impostato (SV).

Quando TIM/TIMX(550) si trova in una sezione del programma compresa tra IL(002) e ILC(003) e tale sezione è interbloccata, il valore attuale (PV) viene ripristinato in base al valore impostato (SV) e il flag di completamento passa a OFF.

Quando un temporizzatore in funzione per TIM/TIMX(550) è stato creato con un numero di temporizzatore compreso tra 0000 e 2047 e si trova in una sezione di programma saltata (JMP(004), CJMP(510), CJPN(511), JME(005)), il valore attuale (PV) del temporizzatore continua a funzionare (vedere nota). L'istruzione saltata TIM/TIMX(550) non viene eseguita, ma il valore attuale (PV) è aggiornato a ogni ciclo al termine dell'esecuzione di tutti i task.

Nota Nelle CPU CS1D il valore attuale (PV) non viene in questo caso aggiornato.

Quando un temporizzatore per TIM/TIMX(550) è caratterizzato da impostazione forzata, il flag di completamento corrispondente viene impostato su ON e il valore attuale (PV) su 0000. Quando un temporizzatore per TIM/TIMX(550) è caratterizzato da ripristino forzato, il flag di completamento corrispondente viene impostato su OFF e il valore attuale (PV) viene ripristinato in base al valore impostato (SV).

Il funzionamento dei flag = ed N dipende dal modello della CPU. Per ulteriori informazioni, fare riferimento a *Flag* poco sopra.

Il flag di completamento del temporizzatore viene aggiornato soltanto all'esecuzione di TIM/TIMX(550); allo scadere del temporizzatore, quindi, può rendersi necessario un ritardo di un intero ciclo per consentire l'attivazione del flag di completamento.

Se si utilizza la modifica in linea per convertire un temporizzatore in un altro tipo di temporizzatore dotato dello stesso numero di temporizzatore (ad esempio TIM/TIMX(550) ↔ TIMH(015)/TIMHX(551) oppure TIM/TIMX(550) ↔ TMHH(540)/TMHHX(552)), assicurarsi di reimpostare il flag di completamento. Se tale reset non viene eseguito, il temporizzatore non funzionerà in modo corretto.

In base al numero di temporizzatore utilizzato, è possibile aggiornare il valore attuale (PV) e il flag di completamento di un'istruzione TIM/TIMX(550) come indicato nelle tabelle seguenti.

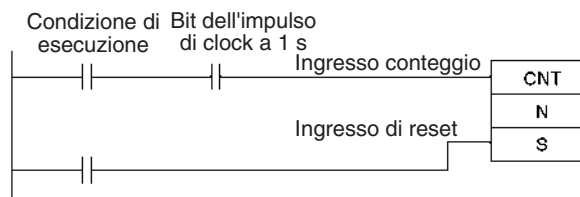
Temporizzatori creati con numero di temporizzatore compreso tra 0000 e 2047

| | |
|-----------------------------------|---|
| Esecuzione di TIM/TIMX(550) | Il PV viene aggiornato a ogni esecuzione di TIM/TIMX(550). Se il PV è 0000, il flag di completamento viene impostato su ON. Se il PV non è 0000, il flag di completamento viene impostato su OFF. |
| Dopo l'esecuzione di tutti i task | Il PV viene inoltre aggiornato a ogni ciclo al termine dell'esecuzione del programma. |
| Aggiornamento ogni 80 ms | Se il tempo di ciclo supera 80 ms, il PV del temporizzatore viene aggiornato ogni 80 ms. |

Temporizzatori creati con numero di temporizzatore compreso tra 2048 e 4095

| | |
|-------------------|---|
| Esecuzione di TIM | Il PV viene aggiornato a ogni esecuzione di TIM. Se il PV è 0000, il flag di completamento viene impostato su ON. Se il PV non è 0000, il flag di completamento viene impostato su OFF. |
|-------------------|---|

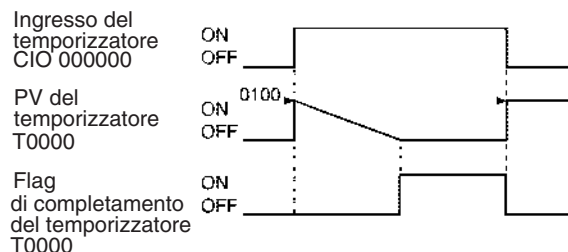
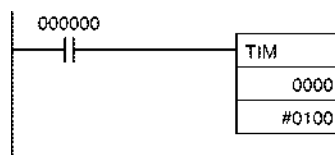
Nel caso di caduta di tensione i temporizzatori vengono ripristinati (PV = SV, flag di completamento OFF) a meno che il bit di ritentività IOM (A50012) sia impostato su ON e il bit sia protetto nella configurazione del PLC. Inoltre è possibile utilizzare un bit dell'impulso di clock e un'istruzione di contatore per programmare un temporizzatore che mantenga il proprio valore attuale (PV) nel caso di caduta di tensione. Osservare il diagramma seguente.



Esempio

Nell'esempio che segue, quando lo stato di CIO 000000 dell'ingresso del temporizzatore passa da OFF a ON, il valore attuale (PV) del temporizzatore comincia a diminuire partendo dal valore impostato (SV). Quando il valore attuale (PV) raggiunge 0000, il flag di completamento T0000 del temporizzatore viene impostato su ON.

Quando CIO 000000 passa a OFF, il valore attuale (PV) del temporizzatore è ripristinato in base al valore impostato (SV) e il flag di completamento viene impostato su OFF.



3-6-2 HIGH-SPEED TIMER: TIMH(015)/TIMHX(551)

Scopo

TIMH(015)/TIMHX(551) aziona un temporizzatore decrementale con unità di 10 ms. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 99,99 s per TIMH(015) e tra 0 e 655,35 s per TIMHX(551). La precisione del temporizzatore è compresa tra 0 e 0,01 s.

Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

Simbolo programmazione

ladder

| Metodo di aggiornamento PV | Simbolo | Operandi |
|----------------------------|--|--|
| BCD | <p>N: numero di temporizzatore S: valore impostato</p> | <p>N: da 0000 a 4095 (decimale) S: da #0000 a #9999 (BCD)</p> |
| Binario | <p>N: numero di temporizzatore S: valore impostato</p> | <p>N: da 00000 a 4095 (decimale) S: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale)</p> |

Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TIMH(015)/TIMHX(551) |
|------------|--|----------------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

N: numero del temporizzatore

Il numero del temporizzatore deve essere compreso tra 0000 e 4095 (decimale).

S: valore impostato

In modalità BCD il valore impostato deve essere compreso tra #0000 e 9999.

Caratteristiche operando

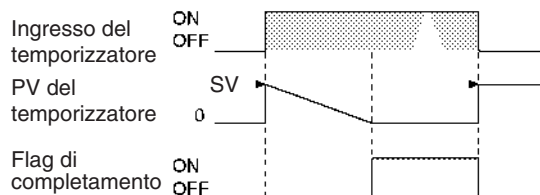
| Area | N | S |
|--|--|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | Da 0000 a 4095 (decimale) | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 @ Da E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- | BCD: Da #0000 a 9999 (BCD) Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

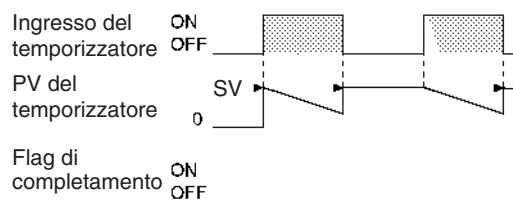
Quando l'ingresso del temporizzatore è impostato su OFF, viene ripristinato il temporizzatore specificato da N, ossia il valore attuale (PV) del temporizzatore è ripristinato in base al valore impostato (SV) e il flag di completamento passa a OFF.

Quando l'ingresso del temporizzatore passa da OFF a ON, TIMH(015)/TIMHX(551) inizia a decrementare il valore attuale (PV). Il PV continua a diminuire fino a quando l'ingresso del temporizzatore resta impostato su ON; il flag di completamento del temporizzatore viene attivato una volta che il PV ha raggiunto 0000.

Lo stato del PV e del flag di completamento del temporizzatore viene mantenuto dopo il timeout del temporizzatore. Per avviare nuovamente il temporizzatore, è necessario impostarne l'ingresso su OFF e quindi ancora su ON oppure sostituire il valore attuale (PV) del temporizzatore con un valore diverso da zero (ad esempio con MOV(021)).



Nel seguente diagramma di funzionamento viene illustrato il comportamento del valore attuale (PV) e del flag di completamento del temporizzatore quando l'ingresso del temporizzatore stesso è impostato su OFF prima del timeout.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se N è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un temporizzatore. ON se in modalità BCD e se S non contiene dati BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF. Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

I numeri di temporizzatore vengono condivisi dalle istruzioni TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) e TMHWX(817). Se due temporizzatori condividono lo stesso numero di temporizzatore, ma non sono utilizzati contemporaneamente, viene generato un errore di duplicazione durante il controllo del programma; il funzionamento dei temporizzatori sarà comunque regolare. I temporizzatori che condividono lo stesso numero di temporizzatore non funzionano in modo corretto se vengono utilizzati contemporaneamente.

I temporizzatori creati con numeri di temporizzatore compresi tra 2048 e 4095 non funzionano in modo regolare quando il tempo di ciclo della CPU supera 80 ms. In tal caso è opportuno utilizzare numeri di temporizzatore compresi tra 0000 e 2047.

I temporizzatori per TIMH(015)/TIMHX(551) creati con numeri di temporizzatore compresi tra 0000 e 0255 vengono aggiornati ogni 10 ms. Utilizzare tali numeri quando il programma utente fa riferimento al valore attuale (PV).

Anche quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 0000 a 2047 viene aggiornato. Quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 2048 a 4095 viene mantenuto.

Il funzionamento dei flag = ed N dipende dal modello della CPU. Per ulteriori informazioni, fare riferimento a *Flag* poco sopra.

All'esecuzione dell'istruzione vengono aggiornati i flag di completamento per i temporizzatori per TIMH(015)/TIMHX(551), a differenza di quanto avviene per i PLC CVM1 e della serie CV.

I temporizzatori vengono reimpostati o messi in pausa nei casi indicati nella tabella qui di seguito. Quando un temporizzatore è reimpostato il valore attuale corrispondente viene a sua volta ripristinato in base al valore impostato e il flag di completamento passa a OFF.

| Condizione | PV | Flag di completamento |
|--|----------------------------|-----------------------|
| Modifica della modalità operativa da RUN o MONITOR a PROGRAM o viceversa. ¹ | 0000 | OFF |
| Interruzione e reset dell'alimentazione ² | 0000 | OFF |
| Esecuzione di CNR(545)/CNRX(547), le istruzioni di reimpostazione RESET TIMER/COUNTER ³ | BCD: 9999 Binario: FFFF | OFF |

| Condizione | PV | Flag di completamento |
|---|-----------------------------------|------------------------------|
| Funzionamento in una sezione del programma interbloccata (IL(002)-ILC(003)) | Reset su SV | OFF |
| Funzionamento in una sezione del programma saltata (JMP(004)-JME(005)) | PV continua a essere decrementato | Mantiene lo stato precedente |

- Nota**
1. Se il bit di ritentività IOM (A50012) è stato impostato su ON, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti alla modifica della modalità operativa.
 2. Se il bit di ritentività IOM (A50012) è stato impostato su ON e lo stato dello stesso bit è protetto nella configurazione del PLC, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti anche quando l'alimentazione è interrotta.
 3. Quando TIMH(015)/TIMHX(551) viene eseguita, il valore attuale (PV) è impostato in base al valore impostato (SV).

Quando un temporizzatore in funzione per TIMH(015)/TIMHX(551) è stato creato con un numero di temporizzatore compreso tra 0000 e 2047 e si trova in una sezione di programma saltata (JMP(004), CJMP(510), CJPN(511), JME(005)), il valore attuale (PV) del temporizzatore continua a funzionare (vedere nota). L'istruzione saltata TIMH(015)/TIMHX(551) non viene eseguita, ma il valore attuale (PV) è aggiornato ogni 10 ms a ciascun ciclo al termine dell'esecuzione di tutti i task.

Nota Nelle CPU CS1D il valore attuale (PV) non viene in questo caso aggiornato.

Quando TIMH(015)/TIMHX(551) si trova in una sezione del programma compresa tra IL(002) e ILC(003) e tale sezione è interbloccata, il valore attuale (PV) viene reimpostato in base al valore impostato (SV) e il flag di completamento passa a OFF.

Quando un temporizzatore per TIMH(015)/TIMHX(551) è caratterizzato da impostazione forzata, il flag di completamento corrispondente viene impostato su ON e il valore attuale (PV) su 0000. Quando tale temporizzatore è caratterizzato da ripristino forzato, il flag di completamento corrispondente viene impostato su OFF e il valore attuale (PV) viene reimpostato in base al valore impostato (SV).

Il funzionamento dei flag = ed N dipende dal modello di CPU. Per ulteriori informazioni, fare riferimento a *Flag*.

Il flag di completamento del temporizzatore viene aggiornato soltanto all'esecuzione di TIMH(015)/TIMHX(551); allo scadere del temporizzatore, quindi, può rendersi necessario un ritardo di un intero ciclo per consentire l'attivazione del flag di completamento.

Se si utilizza la modifica in linea per convertire un temporizzatore in un altro tipo di temporizzatore dotato dello stesso numero di temporizzatore (ad esempio TIMH(015)/TIMHX(551) ↔ TIM/TIMX(550) oppure TIMH(015)/TIMHX(551) ↔ TMHH(540)/TMHHX(552)), assicurarsi di reimpostare il flag di completamento. Se tale reset non viene eseguito, il temporizzatore non funzionerà in modo corretto.

In base al numero di temporizzatore utilizzato, è possibile aggiornare il valore attuale (PV) e il flag di completamento di un'istruzione TIMH(015)/TIMHX(551) come indicato nelle tabelle seguenti.

Temporizzatori creati con numero di temporizzatore compreso tra 0000 e 0255

| | |
|------------------------------------|---|
| Esecuzione di TIMH(015)/TIMHX(551) | Se il PV è 0000, il flag di completamento viene impostato su ON. Se il PV non è 0000, il flag di completamento viene impostato su OFF. |
| Aggiornamento ogni 10 ms | Il PV del temporizzatore viene aggiornato ogni 10 ms. |

Temporizzatori creati con numero di temporizzatore compreso tra 0256 e 2047

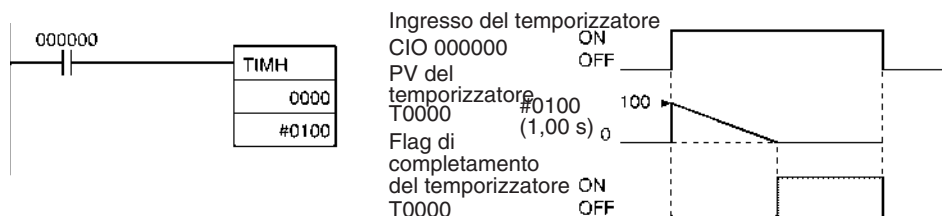
| | |
|------------------------------------|--|
| Esecuzione di TIMH(015)/TIMHX(551) | Il PV viene aggiornato a ogni esecuzione di TIMH(015)/TIMHX(551). Se il PV è 0000, il flag di completamento viene impostato su ON. Se il PV non è 0000, il flag di completamento viene impostato su OFF. |
| Dopo l'esecuzione di tutti i task | Il PV viene inoltre aggiornato a ogni ciclo al termine dell'esecuzione del programma. |
| Aggiornamento ogni 80 ms | Se il tempo di ciclo supera 80 ms, il PV del temporizzatore viene aggiornato ogni 80 ms. |

Temporizzatori creati con numero di temporizzatore compreso tra 2048 e 4095

| | |
|------------------------------------|---|
| Esecuzione di TIMH(015)/TIMHX(551) | Il PV viene aggiornato a ogni esecuzione di TIMH(015). Se il PV è 0000, il flag di completamento viene impostato su ON. Se il PV non è 0000, il flag di completamento viene impostato su OFF. |
|------------------------------------|---|

Esempio

Nell'esempio che segue, quando lo stato di CIO 000000 dell'ingresso del temporizzatore passa da OFF a ON, il valore attuale (PV) del temporizzatore comincia a diminuire partendo dal valore impostato (#0064 = 100 = 1,00 s). Quando il valore attuale (PV) raggiunge 0000, il flag di completamento T0000 del temporizzatore viene impostato su ON. Quando CIO 000000 passa a OFF, il valore attuale (PV) del temporizzatore è ripristinato in base al valore impostato (SV) e il flag di completamento viene impostato su OFF.

**3-6-3 ONE-MS TIMER: TMHH(540)/TMHHX(552)****Scopo**

TMHH(540)/TMHHX(552) aziona un temporizzatore decrementale con unità di 1 ms. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 9,999 s per TMHH(540) e tra 0 e 65,535 s per TMHHX(552). La precisione del temporizzatore è compresa tra -0,001 e 0 s.

Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

Simbolo programmazione ladder

| Metodo di aggiornamento PV | Simbolo | Operandi | | | | | |
|----------------------------|---|-------------|-----------------------------|---|---|---------------------|---|
| BCD | <table border="1"> <tr> <td>TIMHH(540)</td> <td rowspan="2">N: numero di temporizzatore</td> </tr> <tr> <td>N</td> </tr> <tr> <td>S</td> <td>S: valore impostato</td> </tr> </table> | TIMHH(540) | N: numero di temporizzatore | N | S | S: valore impostato | N: da 0000 a 15 (decimale) S: da #0000 a #9999 (BCD) |
| TIMHH(540) | N: numero di temporizzatore | | | | | | |
| N | | | | | | | |
| S | S: valore impostato | | | | | | |
| Binario | <table border="1"> <tr> <td>TIMHHX(552)</td> <td rowspan="2">N: numero di temporizzatore</td> </tr> <tr> <td>N</td> </tr> <tr> <td>S</td> <td>S: valore impostato</td> </tr> </table> | TIMHHX(552) | N: numero di temporizzatore | N | S | S: valore impostato | N: da 00000 a 15 (decimale) S: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| TIMHHX(552) | N: numero di temporizzatore | | | | | | |
| N | | | | | | | |
| S | S: valore impostato | | | | | | |

Variazioni

| | | |
|-------------------------|--|--------------------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | TMHH(540)/ TMHHX(552) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

N: numero del temporizzatore

Il numero del temporizzatore deve essere compreso tra 0000 e 0015 (decimale).

S: valore impostato

Il valore impostato deve essere compreso tra #0000 e 9999 (BCD).

Caratteristiche operando

| Area | N | S |
|--|--|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | Da 0000 a 0015 (decimale) | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- | BCD: Da #0000 a 9999 (BCD) Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

Quando l'ingresso del temporizzatore è impostato su OFF, viene ripristinato il temporizzatore specificato da N, ossia il valore attuale (PV) del temporizzatore è ripristinato in base al valore impostato (SV) e il flag di completamento passa a OFF.

Quando l'ingresso del temporizzatore passa da OFF a ON, TMHH(540)/TMHHX(552) inizia a decrementare il valore attuale (PV). Il PV continua a diminuire fino a quando l'ingresso del temporizzatore resta impostato su ON;

il flag di completamento del temporizzatore viene attivato una volta che il PV ha raggiunto 0000.

Lo stato del PV e del flag di completamento del temporizzatore viene mantenuto dopo il timeout del temporizzatore. Per avviare nuovamente il temporizzatore, è necessario impostarne l'ingresso su OFF e quindi ancora su ON oppure sostituire il valore attuale (PV) del temporizzatore con un valore diverso da zero (ad esempio con MOV(021)).

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se N è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un temporizzatore. ON se in modalità BCD e se S non contiene dati BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

I numeri di temporizzatore vengono condivisi dalle istruzioni TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) e TMHWX(817). Se due temporizzatori condividono lo stesso numero di temporizzatore, ma non sono utilizzati contemporaneamente, viene generato un errore di duplicazione durante il controllo del programma; il funzionamento dei temporizzatori sarà comunque regolare. I temporizzatori che condividono lo stesso numero di temporizzatore non funzionano in modo corretto se vengono utilizzati contemporaneamente.

Il flag di completamento viene aggiornato soltanto all'esecuzione di TMHH(540)/TMHHX(552) e può quindi subire un ritardo massimo di un intero tempo di ciclo a partire dal valore impostato effettivo.

Anche quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 0000 a 2047 viene aggiornato. Quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 2048 a 4095 viene mantenuto.

I temporizzatori vengono reimpostati o messi in pausa nei casi indicati nella tabella qui di seguito. Quando un temporizzatore è reimpostato il valore attuale corrispondente viene a sua volta ripristinato in base al valore impostato e il flag di completamento passa a OFF.

| Condizione | PV | Flag di completamento |
|--|-----------------------------------|------------------------------|
| Modifica della modalità operativa da RUN o MONITOR a PROGRAM o viceversa. ¹ | 0000 | OFF |
| Interruzione e reset dell'alimentazione ² | 0000 | OFF |
| Esecuzione di CNR(545)/CNRX(547), le istruzioni di reimpostazione RESET TIMER/COUNTER ³ | BCD: 9999 Binario: FFFF | OFF |
| Funzionamento in una sezione del programma interbloccata (IL(002)-ILC(003)) | Reset su SV | OFF |
| Funzionamento in una sezione del programma saltata (JMP(004)-JME(005)) | PV continua a essere decrementato | Mantiene lo stato precedente |

- Nota**
1. Se il bit di ritenività IOM (A50012) è stato impostato su ON, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti alla modifica della modalità operativa.
 2. Se il bit di ritenività IOM (A50012) è stato impostato su ON e lo stato dello stesso bit è protetto nella configurazione del PLC, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti anche quando l'alimentazione è interrotta.

3. Quando TMHH(540)/TMHHX(552) viene eseguita, il valore attuale (PV) è impostato in base al valore impostato (SV).

Quando un temporizzatore in funzione per TMHH(540)/TMHHX(552) è stato creato in una sezione del programma saltata (JMP(004), CJMP(510), CJPN(511), JME(005)), il valore attuale (PV) del temporizzatore continua a funzionare (vedere nota). L'istruzione saltata TMHH(540)/TMHHX(552) non viene eseguita, ma il valore attuale (PV) è aggiornato a ciascun ms.

Nota Nelle CPU CS1D il valore attuale (PV) non viene in questo caso aggiornato.

Quando TMHH(540)/TMHHX(552) si trova in una sezione del programma compresa tra IL(002) e ILC(003) e tale sezione è interbloccata, il valore attuale (PV) viene reimpostato in base al valore impostato (SV) e il flag di completamento passa a OFF.

Quando un temporizzatore per TMHH(540)/TMHHX(552) è caratterizzato da impostazione forzata, il flag di completamento corrispondente viene impostato su ON e il valore attuale (PV) su 0000. Quando tale temporizzatore è caratterizzato da ripristino forzato, il flag di completamento corrispondente viene impostato su OFF e il valore attuale (PV) viene reimpostato in base al valore impostato (SV).

Il funzionamento dei flag = ed N dipende dal modello della CPU. Per ulteriori informazioni, fare riferimento a *Flag* poco sopra.

Se si utilizza la modifica in linea per convertire un temporizzatore in un altro tipo di temporizzatore dotato dello stesso numero di temporizzatore (ad esempio TMHH(540)/TMHHX(552) ↔ TIM/TIMX(550) oppure TMHH(540)/TMHHX(552) ↔ TIMH(015)/TIMHX(551)), assicurarsi di reimpostare il flag di completamento. Se tale reset non viene eseguito, il temporizzatore non funzionerà in modo corretto.

Il valore attuale (PV) e il flag di completamento di un'istruzione TMHH(540)/TMHHX(552) vengono aggiornati come indicato nella tabella seguente.

| | |
|------------------------------------|---|
| Esecuzione di TMHH(540)/TMHHX(552) | Se il PV è 0000, il flag di completamento viene impostato su ON. Se il PV non è 0000, il flag di completamento viene impostato su OFF. |
| Aggiornamento ogni 1 ms | Il PV del temporizzatore viene aggiornato ogni 1 ms. |


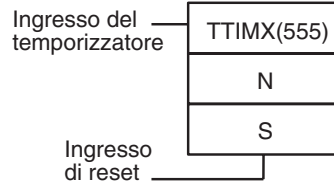
3-6-4 ACCUMULATIVE TIMER: TTIM(087)/TTIMX(555)

Scopo

TTIM(087)/TTIMX(555) aziona un temporizzatore incrementale con unità di 0,1 s. L'intervallo di impostazione per il valore impostato (SV) è compreso tra 0 e 999,9 s per TTIM(087) e tra 0 e 6.553,5 s per TTIMX(555). La precisione del temporizzatore è compresa tra -0,01 e 0 s.

Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

Simbolo programmazione ladder

| Metodo di aggiornamento PV | Simbolo | Operandi |
|----------------------------|---|---|
| BCD |  <p>N: numero di temporizzatore S: valore impostato</p> | N: da 0000 a 15 (decimale) S: da #0000 a #9999 (BCD) |
| Binario |  <p>N: numero di temporizzatore S: valore impostato</p> | N: da 00000 a 15 (decimale) S: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |

Variazioni

| | | |
|-------------------------|--|--------------------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | TTIM(087)/ TTIMX(555) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

N: numero del temporizzatore

Il numero del temporizzatore deve essere compreso tra 0000 e 4095 (decimale).

S: valore impostato

Il valore impostato deve essere compreso tra #0000 e 9999 (BCD).

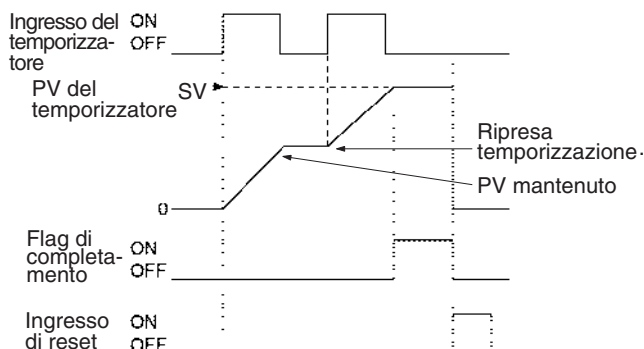
Caratteristiche operando

| Area | N | S |
|--|--|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritentività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | Da 0000 a 4095 (decimale) | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- | BCD: Da #0000 a 9999 (BCD) Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

Quando l'ingresso del temporizzatore è impostato su ON, TTIM(087)/TTIMX(555) incrementa il valore attuale (PV). Quando l'ingresso del temporizzatore passa a OFF, il temporizzatore interrompe l'incremento del valore attuale (PV), che comunque mantiene il proprio valore. Quando l'ingresso del temporizzatore passa nuovamente a ON, il PV riprende la temporizzazione. Quando il valore attuale (PV) raggiunge il valore impostato (SV), il flag di completamento del temporizzatore viene impostato su ON.

Lo stato del PV e del flag di completamento del temporizzatore viene mantenuto dopo il timeout del temporizzatore. Esistono tre modalità di riavvio del temporizzatore: è possibile modificare il valore attuale PV del temporizzatore con un valore diverso da zero (ad esempio con MOV(021)), attivare l'ingresso di reset oppure eseguire CNR(545)/CNRX(547).



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se N è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un temporizzatore. ON se in modalità BCD e se S non contiene dati BCD. OFF in tutti gli altri casi. |

Avvertenze

I numeri di temporizzatore vengono condivisi dalle istruzioni TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) e TMHWX(817). Se due temporizzatori condividono lo stesso numero di temporizzatore, ma non sono utilizzati contemporaneamente, viene generato un errore di duplicazione durante il controllo del programma; il funzionamento dei temporizzatori sarà comunque regolare. I temporizzatori che condividono lo stesso numero di temporizzatore non funzionano in modo corretto se vengono utilizzati contemporaneamente.

I temporizzatori vengono reimpostati o messi in pausa nei casi indicati nella tabella qui di seguito. Quando si reimposta un temporizzatore per TTIM(087)/TTIMX(555), il valore attuale corrispondente viene a sua volta reimpostato su 0000 e il flag di completamento passa a OFF.

| Condizione | PV | Flag di completamento |
|--|------------------------------|------------------------------|
| Modifica della modalità operativa da RUN o MONITOR a PROGRAM o viceversa. ¹ | 0000 | OFF |
| Interruzione e reset dell'alimentazione ² | 0000 | OFF |
| Esecuzione di CNR(545)/CNRX(547), le istruzioni di reimpostazione RESET TIMER/COUNTER ³ | BCD: 9999 Binario: FFFF | OFF |
| Funzionamento in una sezione del programma interbloccata (IL(002)-ILC(003)) | Mantiene lo stato precedente | Mantiene lo stato precedente |
| Funzionamento in una sezione del programma saltata (JMP(004)-JME(005)) | Mantiene lo stato precedente | Mantiene lo stato precedente |

Nota

- Se il bit di ritenività IOM (A50012) è stato impostato su ON, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti alla modifica della modalità operativa.
- Se il bit di ritenività IOM (A50012) è stato impostato su ON e lo stato dello stesso bit è protetto nella configurazione del PLC, lo stato dei flag di completamento e i valori attuali del temporizzatore vengono mantenuti anche quando l'alimentazione è interrotta.
- Quando TTIM(087)/TTIMX(555) viene eseguita, il valore attuale (PV) è impostato in base al valore impostato (SV).

Quando TTIM(087)/TTIMX(555) si trova in una sezione del programma compresa tra IL(002) e ILC(003) e tale sezione è interbloccata, il valore

attuale (PV) mantiene il proprio valore precedente, ossia non viene ripristinato. Se TTIM(087)/TTIMX(555) è programmata tra IL(002) e ILC(003) è importante tenere in opportuna considerazione questo fatto.

Quando un temporizzatore in funzione per TTIM(087)/TTIMX(555) si trova in una sezione del programma compresa tra JMP(004) e JME(005) e tale sezione è saltata, il valore attuale (PV) mantiene il proprio valore precedente. Se TTIM(087)/TTIMX(555) è programmata tra JMP(004) e JME(005) è importante tenere in opportuna considerazione questo fatto.

Quando un temporizzatore per TTIM(087)/TTIMX(555) è caratterizzato da impostazione forzata, il flag di completamento corrispondente è impostato su ON e il valore attuale (PV) viene reimpostato su 0000. Quando tale temporizzatore è caratterizzato da ripristino forzato, il flag di completamento corrispondente è impostato su OFF e il valore attuale (PV) viene reimpostato su 0000. Le operazioni di forzatura dell'impostazione e del ripristino hanno la precedenza sullo stato del temporizzatore e degli ingressi di reset.

Il valore attuale (PV) del temporizzatore viene aggiornato all'esecuzione di TTIM(087)/TTIMX(555); di conseguenza il temporizzatore non funziona in modo corretto quando il tempo di ciclo supera 100 ms a causa dell'incremento del temporizzatore in unità di 100 ms.

Il flag di completamento del temporizzatore viene aggiornato soltanto all'esecuzione di TTIM(087)/TTIMX(555); allo scadere del temporizzatore, quindi, può rendersi necessario un ritardo di un intero ciclo per consentire l'attivazione del flag di completamento.

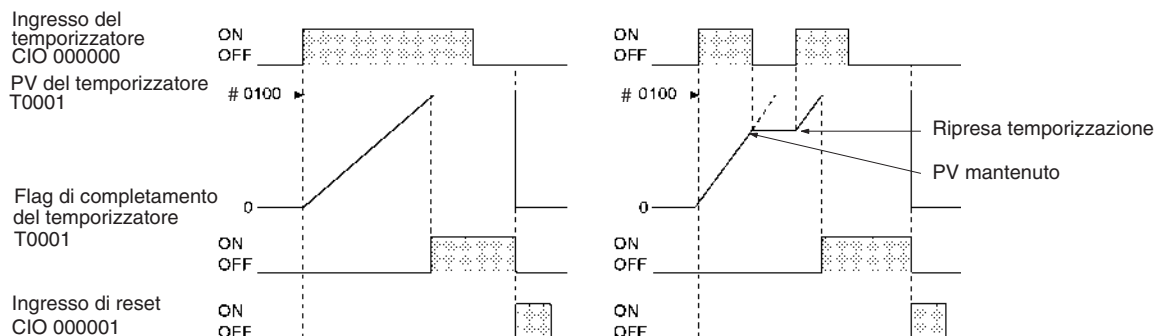
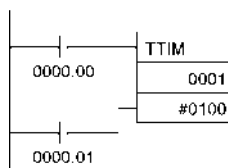
Temporizzatori specifici come TIM/TIMX(550) sono contatori decrementali e il valore attuale (PV) segnala il tempo restante al timeout del temporizzatore. Il valore attuale (PV) di TTIM(087)/TTIMX(555) segnala quanto tempo è trascorso e quindi può essere utilizzato senza alcuna modifica in numerosi calcoli e uscite display.

Esempio

Nell'esempio che segue, quando lo stato di CIO 000000 dell'ingresso del temporizzatore è impostato su ON, il valore attuale (PV) del temporizzatore comincia a incrementare da 0 e quando raggiunge il valore impostato (SV) il flag di completamento T0001 viene attivato.

Se l'ingresso di reset passa a ON, il valore attuale (PV) del temporizzatore è reimpostato su 0000 e il flag di completamento (T0001) viene impostato su OFF (in genere l'ingresso di reset viene attivato per reimpostare il temporizzatore e successivamente l'ingresso del temporizzatore stesso viene attivato per iniziare il monitoraggio).

Se l'ingresso del temporizzatore viene disattivato prima del raggiungimento del valore impostato (SV), il temporizzatore interrompe il monitoraggio ma il valore attuale (PV) viene mantenuto. Alla successiva attivazione del proprio ingresso, il temporizzatore riprende dal valore attuale (PV) precedente.



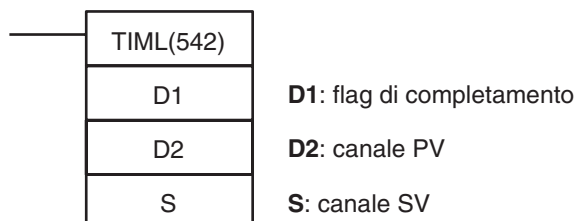
3-6-5 LONG TIMER: TIML(542)/TIMLX(553)

Scopo TIML(542)/TIMLX(553) aziona un temporizzatore decrementale con unità di 1 s che può raggiungere fino a 115 giorni per TIML(542) e 49.710 giorni per TIMLX(543). La precisione del temporizzatore è compresa tra 0 e 0,01 s.

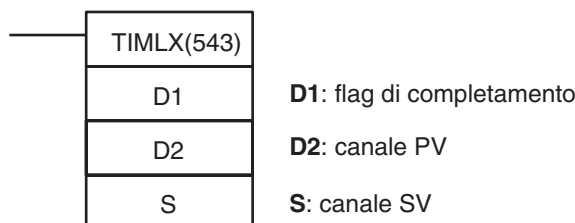
Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

Simbolo programmazione ladder

BCD



Binario



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TIML(542)/TIMLX(553) |
|--------------------------------|--|----------------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

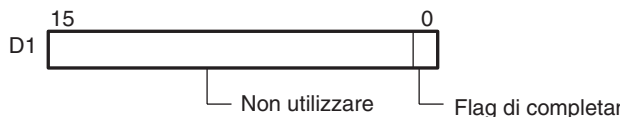
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

D1: Flag di completamento

Il bit 0 di D1 funge da flag di completamento per TIML(542)/TIMLX(553).



D2: canale PV

D2+1 e D2 contengono il valore attuale PV binario a 8 cifre o BCD. D2 e D2+1 devono trovarsi nella stessa area dati. Il valore attuale (PV) può essere compreso tra #00000000 e #99999999 per TIML(542) e tra &00000000 e &4294967294 (decimale) oppure tra #00000000 e #FFFFFFF (esadecimale) per TIMLX(553).



S: canale SV

S+1 e S contengono il valore impostato SV binario a 8 cifre o BCD. S e S+1 devono trovarsi nella stessa area dati. Il valore impostato (SV) deve essere compreso tra #00000000 e #99999999 per TIML(542) e tra &00000000 e &4294967294 (decimale) oppure tra #00000000 e #FFFFFFF (esadecimale) per TIMLX(553).



Caratteristiche operando

| Area | D1 | D2 | S |
|--|--|--|------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H510 | |
| Area bit ausiliaria | Da A448 ad A959 | Da A448 ad A958 | Da A000 ad A958 |
| Area del temporizzatore | --- | --- | Da T0000 a T4094 |
| Area del contatore | --- | --- | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | En_00000 ... En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | BCD: Da #00000000 a 99999999 (BCD) Impossibile utilizzare "&". Binario: Da &00000000 a &4294967294 (decimale) o da #00000000 a #FFFFFFFF (esadecimale) | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | | |

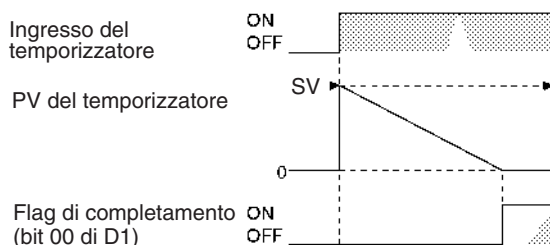
Descrizione

TIML(542)/TIMLX(553) è un temporizzatore in ritardo all'eccitazione con decremento di unità pari a 0,1 s che utilizza un valore impostato (SV) e un valore attuale (PV) a 8 cifre.

Quando l'ingresso del temporizzatore è disattivato, il temporizzatore viene reimpostato, ossia il valore attuale (PV) del temporizzatore è reimpostato in base al valore impostato (SV) e il flag di completamento passa a OFF.

Quando l'ingresso del temporizzatore passa da OFF a ON, TIML(542)/TIMLX(553) inizia a decrementare il valore attuale (PV) in D2+1 e D2. Il PV continua a diminuire fino a quando l'ingresso del temporizzatore resta impostato su ON; il flag di completamento del temporizzatore viene attivato una volta che il PV ha raggiunto 0000 0000.

Lo stato del PV e del flag di completamento del temporizzatore viene mantenuto dopo il timeout del temporizzatore. Per avviare nuovamente il temporizzatore, è necessario impostarne l'ingresso su OFF e quindi ancora su ON oppure sostituire il valore attuale (PV) del temporizzatore con un valore diverso da zero (ad esempio con MOV(021)).



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se il PV contenuto in D2+1 e D2 non è BCD. ON se l'SV contenuto in S+1 ed S non è BCD. OFF in tutti gli altri casi. |

Avvertenze

A differenza della maggior parte dei temporizzatori, TIML(542)/TIMLX(553) non utilizza un numero di temporizzatore (per TIML(542)/TIMLX(553) non viene eseguito l'aggiornamento del valore attuale (PV) dell'area del temporizzatore).

Poiché il flag di completamento per TIML(542)/TIMLX(553) è in un'area dati, è possibile procedere a un'impostazione o a un ripristino forzati come per gli altri bit, ma il valore attuale (PV) non verrà modificato.

Il valore attuale (PV) del temporizzatore viene aggiornato all'esecuzione di TIML(542)/TIMLX(553); di conseguenza il temporizzatore non funziona in modo corretto quando il tempo di ciclo supera 100 ms a causa dell'incremento del temporizzatore in unità di 100 ms.

Il flag di completamento del temporizzatore viene aggiornato soltanto all'esecuzione di TIML(542)/TIMLX(553); allo scadere del temporizzatore, quindi, può rendersi necessario un ritardo di un intero ciclo per consentire l'attivazione del flag di completamento.

Quando TIML(542)/TIMLX(553) si trova in una sezione del programma compresa tra IL(002) e ILC(003) e tale sezione è interbloccata, il valore attuale (PV) viene reimpostato in base al valore impostato (SV) e il flag di completamento passa a OFF.

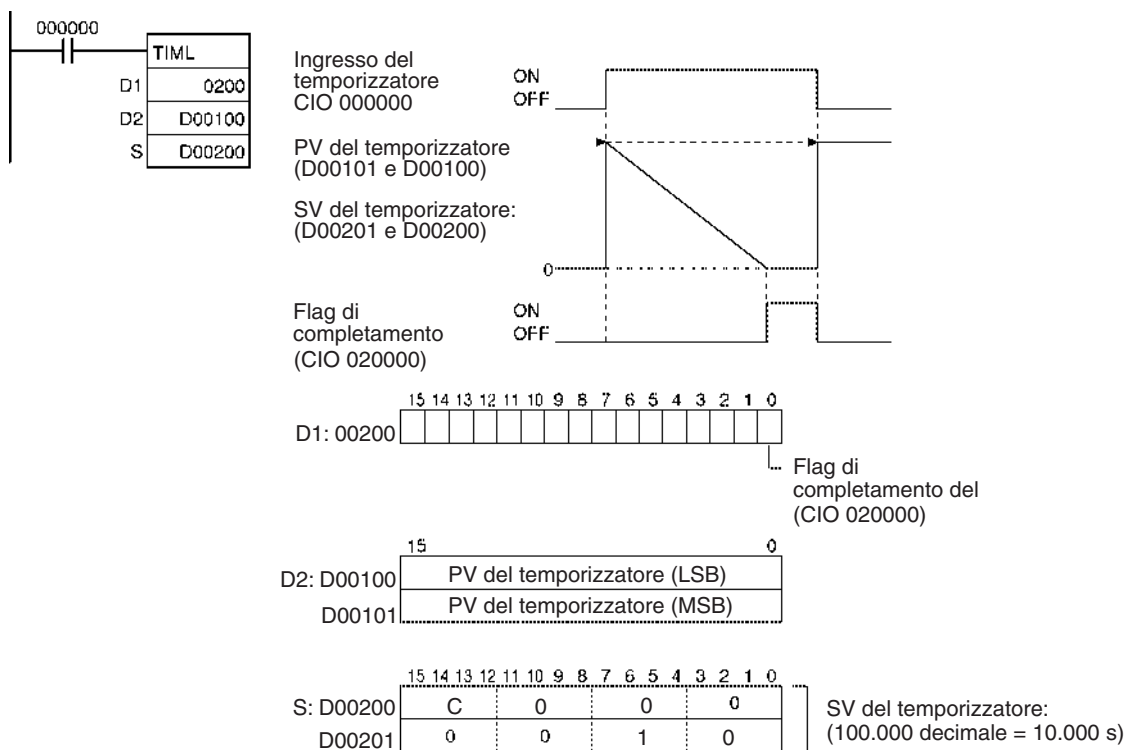
Quando un temporizzatore in funzione per TIML(542)/TIMLX(553) si trova in una sezione del programma compresa tra JMP(004) e JME(005) e tale sezione è saltata, il valore attuale (PV) mantiene il proprio valore precedente. Se TIML(542)/TIMLX(553) è programmata tra JMP(004) e JME(005) è importante tenere in opportuna considerazione questo fatto.

Assicurarsi che i canali specificati per il flag di completamento e il valore attuale (D1, D2 e D2+1) non siano utilizzati in altre istruzioni. Se altre istruzioni influiscono su questi canali, è possibile che il temporizzatore non funzioni correttamente.

Esempio

Nell'esempio che segue, quando lo stato di CIO 000000 dell'ingresso del temporizzatore è impostato su ON, il valore attuale del temporizzatore (in D00101 e D00100) viene impostato in base al valore impostato (in D00101 e D00100) e inizia a decrementare. Quando il valore attuale (PV) raggiunge 0000 0000, il flag di completamento (CIO 020000) del temporizzatore viene impostato su ON.

Quando CIO 000000 passa a OFF, il valore attuale (PV) del temporizzatore è ripristinato in base al valore impostato (SV) e il flag di completamento viene impostato su OFF.



3-6-6 MULTI-OUTPUT TIMER: MTIM(543)/MTIMX(554)

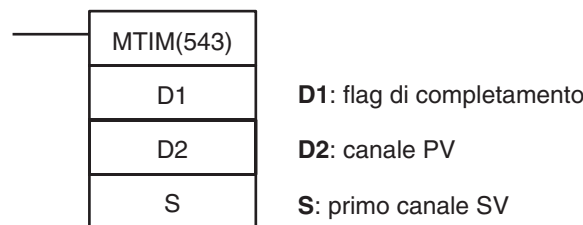
Scopo

MTIM(543)/MTIMX(554) aziona un temporizzatore incrementale con unità di 0,1 s che dispone di otto valori impostati e flag di completamento indipendenti. Il valore impostato è compreso tra 0 e 999,9 s per MTIM(543) e da 0 a 6.553,5 s per MTIMX(554); la precisione del temporizzatore, inoltre, è compresa tra 0 e 0,01 s.

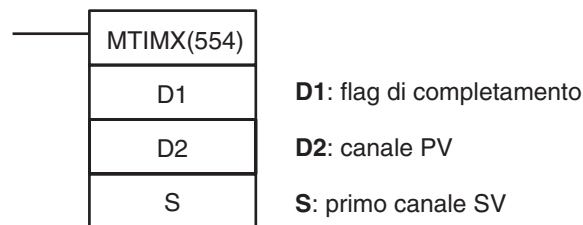
Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

Simbolo programmazione ladder

BCD



Binario



Variazioni

| | | |
|-------------------------|--|----------------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MTIM(543)/MTIMX(554) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

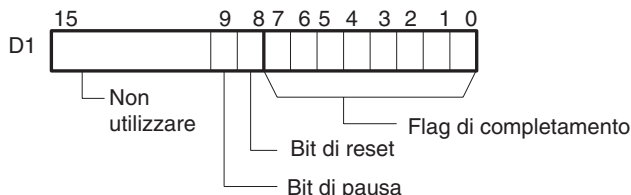
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

D1: Flag di completamento

D1 contiene gli otto flag di completamento nonché i bit di pausa e di reset.



D2: canale PV

D2 contiene il valore attuale PV binario a 4 cifre o BCD.

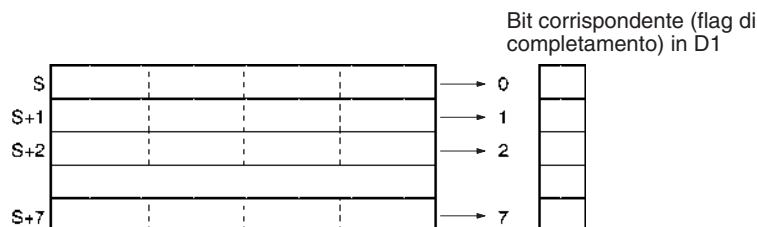
| Dati | Intervallo |
|---------|---|
| BCD | Da #0000 a #9999 |
| Binario | Da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |

S: primo canale SV

Otto SV indipendenti da S a S+7.

Ogni SV deve presentarsi nel modo seguente:

| Dati | Intervallo |
|---------|---|
| BCD | Da #0000 a #9999 |
| Binario | Da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |



| Dati | Intervallo |
|---------|---|
| BCD | Un canale per ciascuno degli otto SV del temporizzatore: Da #0000 a #9999 |
| Binario | Un canale per ciascuno degli otto SV del temporizzatore: Da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |

Nota I canali compresi tra S ed S+7 devono trovarsi nella stessa area dati.

Caratteristiche operando

| Area | D1 | D2 | S |
|-------------------------|------------------------|----|------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6136 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W504 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H504 |
| Area bit ausiliaria | Da A448 ad A959 | | Da A000 ad A952 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4088 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4088 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32760 |

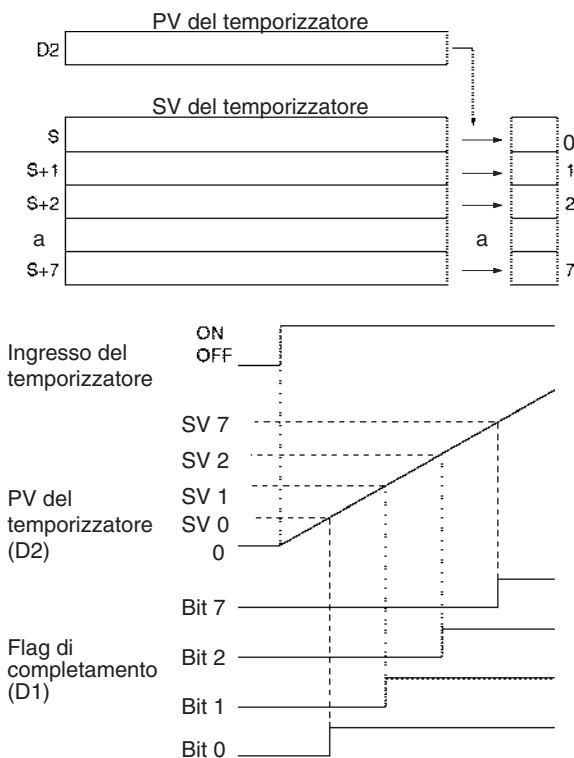
| Area | D1 | D2 | S |
|--|--|---------------|---|
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32760 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | En_00000 ... En_32760 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | Da DR0 a DR15 | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Quando la condizione di esecuzione per MTIM(543)/MTIMX(554) è impostata su ON e i bit di reset e di temporizzazione sono entrambi disattivati, MTIM(543)/MTIMX(554) incrementa il valore attuale (PV) in D2. Se il bit di pausa è impostato su ON, il temporizzatore interrompe l'incremento del valore attuale (PV), che comunque mantiene il proprio valore. Quando il bit di pausa passa nuovamente a OFF, MTIM(543)/MTIMX(554) riprende la temporizzazione.

A ogni esecuzione di MTIM(543)/MTIMX(554) il valore attuale (contenuto in D2) viene messo a confronto con gli otto valori di SV tra S e S+7; se un SV qualsiasi è inferiore o uguale al valore attuale (PV), il flag di completamento corrispondente (bit di D1 tra 00 e 07) viene impostato su ON.

Quando raggiunge 9999, il valore attuale (PV) viene reimpostato su 0000 e tutti i flag di completamento saranno disattivati. Se il bit di reset passa a ON mentre il temporizzatore è in funzione o in pausa, il valore attuale (PV) viene reimpostato su 0000 e tutti i flag di completamento saranno disattivati.



Nella tabella seguente viene riportato il funzionamento dell'istruzione MTIM(543)/MTIMX(554) nelle quattro possibili combinazioni dei bit di reset e di pausa.

| Bit di reset (bit 08) | Bit di pausa (bit 09) | Funzionamento |
|-----------------------|-----------------------|---|
| OFF | OFF | Il valore attuale (PV) viene aggiornato e il flag di completamento relativo passa a ON quando $SV \leq PV$. |
| | ON | Il valore attuale (PV) non viene aggiornato e MTIM(543)/MTIMX(554) è considerata come NOP(000). |
| ON | OFF | Il valore attuale (PV) viene reimpostato a 0000 e i flag di completamento passano a OFF. Il valore attuale (PV) non viene aggiornato. |
| | ON | |

I bit di reset e di pausa diventano effettivi soltanto quando la condizione di esecuzione per MTIM(543)/MTIMX(554) è impostata su ON.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se il PV contenuto in D2 non è BCD. OFF in tutti gli altri casi. |

Avvertenze

A differenza della maggior parte dei temporizzatori, MTIM(543)/MTIMX(554) non utilizza un numero di temporizzatore (per MTIM(543)/MTIMX(554) non viene eseguito l'aggiornamento del valore attuale (PV) dell'area del temporizzatore).

Quando raggiunge 9999, il valore attuale (PV) viene reimpostato su 0000 e tutti i flag di completamento saranno disattivati.

In modalità BCD viene ignorato un eventuale un valore SV impostato tra S ed S+7 che non contenga dati in formato BCD. Non si verifica alcun errore e il flag di errore non viene attivato.

Poiché il flag di completamento per MTIM(543)/MTIMX(554) è in un'area dati, è possibile procedere a un'impostazione o a un ripristino forzati come per gli altri bit, ma il valore attuale (PV) non verrà modificato.

Quando non sono richiesti più di otto valori impostati (SV), è opportuno che il canale successivo all'ultimo SV sia impostato su 0000. MTIM(543)/MTIMX(554) ignora l'SV impostato su 0000 nonché tutti i restanti SV.

| | | | | | |
|-------|--------|---|---|---|---|
| SCH | 0002CH | 1 | 0 | 2 | 9 |
| | 0003CH | 2 | 5 | 0 | 6 |
| | 0004CH | 6 | 0 | 4 | 0 |
| | 0005CH | 0 | 0 | 0 | 0 |
| | a | a | | | |
| S+7CH | 0009CH | | | | |

Questi SV vengono ignorati

Il valore attuale (PV) del temporizzatore viene aggiornato all'esecuzione di MTIM(543)/MTIMX(554); di conseguenza il temporizzatore non funziona in modo corretto quando il tempo di ciclo supera 100 ms a causa dell'incremento del temporizzatore in unità di 100 ms. Per garantire una precisa temporizzazione ed evitare problemi causati da tempi lunghi di ciclo, specificare la stessa istruzione MTIM(543)/MTIMX(554) in più punti all'interno del programma.

Il flag di completamento del temporizzatore viene aggiornato soltanto all'esecuzione di MTIM(543)/MTIMX(554); allo scadere del temporizzatore, quindi, può rendersi necessario un ritardo di un intero ciclo per consentire l'attivazione del flag di completamento.

Quando MTIM(543)/MTIMX(554) si trova in una sezione del programma compresa tra IL(002) e ILC(003) e tale sezione è interbloccata, il valore attuale (PV) mantiene il proprio valore precedente, ossia non viene reimpostato. Se MTIM(543)/MTIMX(554) è programmata tra IL(002) e ILC(003) è importante tenere in opportuna considerazione questo fatto.

Quando un temporizzatore in funzione per MTIM(543)/MTIMX(554) si trova in una sezione del programma compresa tra JMP(004) e JME(005) e tale sezione è saltata, il valore attuale (PV) mantiene il proprio valore precedente. Se MTIM(543)/MTIMX(554) è programmata tra JMP(004) e JME(005) è importante tenere in opportuna considerazione questo fatto.

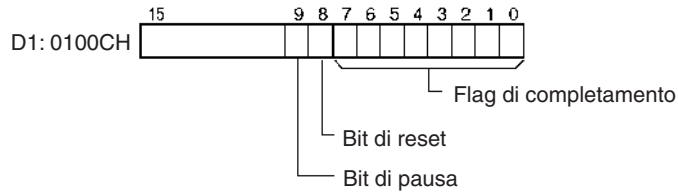
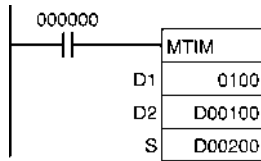
Assicurarsi che i canali specificati per i flag di completamento e il valore attuale (D1 e D2) non siano utilizzati in altre istruzioni. Se altre istruzioni influiscono su questi canali, è possibile che il temporizzatore non funzioni correttamente.

Se un canale dell'area CIO è specificato per D1, è possibile utilizzare le istruzioni SET e RSET per controllare i bit di pausa e di reset.

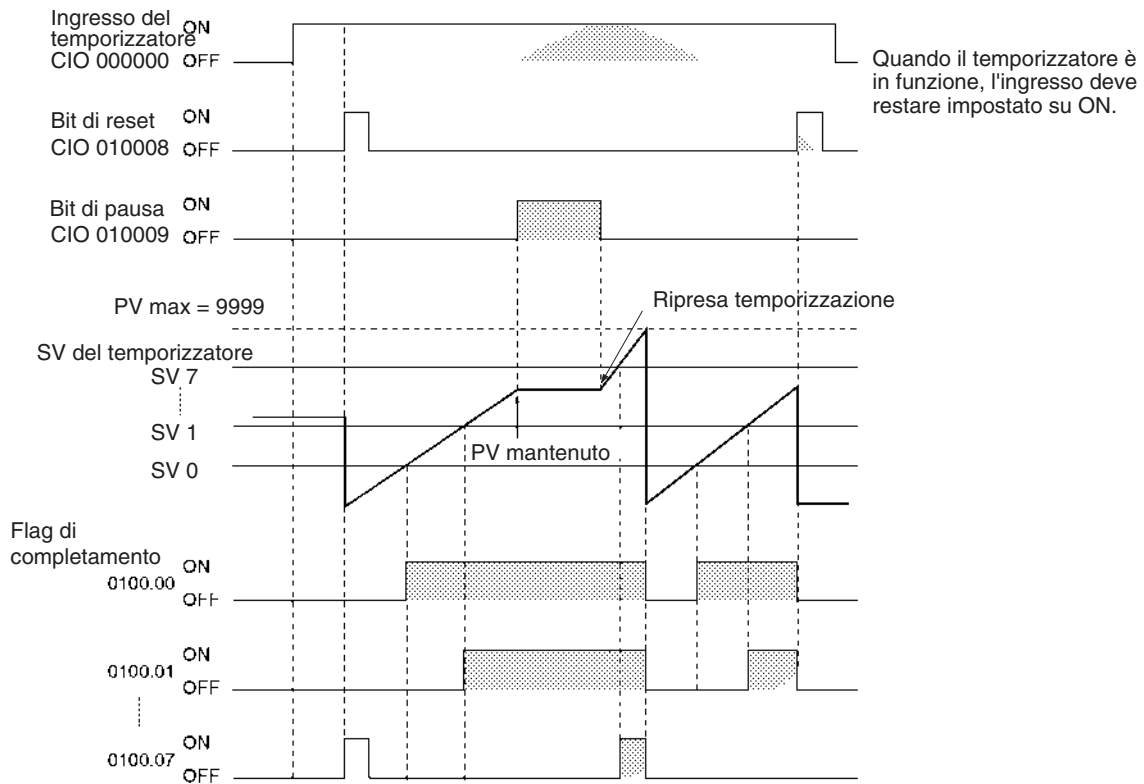
Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON e il bit di pausa (CIO 010009) è OFF, il temporizzatore comincia a funzionare quando il bit di reset (CIO 010009) passa da ON a OFF. Il valore attuale (PV) del temporizzatore comincia a incrementare a partire da 0000.

Gli otto valori impostati (SV) compresi tra D00200 e D00207 vengono messi a confronto con il valore attuale (PV); i flag di completamento relativi (da CIO 010000 a CIO 010007) vengono attivati quando $SV \leq PV$.



| | | | | | | | | | | | | | | | | | | |
|-----------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--------------|---|
| PV del temporizzatore | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (incremento) | |
| D2: D00100 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Flag di completamento corrispondente ON quando SV = PV. |
| SV del temporizzatore | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| S: D00200 | 0 | 0 | 8 | 0 | | | | | | | | | | | | | 1 0 | |
| S+1: D00201 | 0 | 0 | 9 | 0 | | | | | | | | | | | | | 1 1 | |
| S+2: D00202 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | 1 2 | |
| S+3: D00203 | 0 | 1 | 1 | 0 | | | | | | | | | | | | | 0 3 | |
| S+4: D00204 | 0 | 1 | 2 | 0 | | | | | | | | | | | | | 0 4 | |
| S+5: D00205 | 0 | 1 | 3 | 0 | | | | | | | | | | | | | 0 5 | |
| S+6: D00206 | 0 | 1 | 5 | 0 | | | | | | | | | | | | | 0 6 | |
| S+7: D00207 | 1 | 0 | 0 | 0 | | | | | | | | | | | | | 0 7 | |

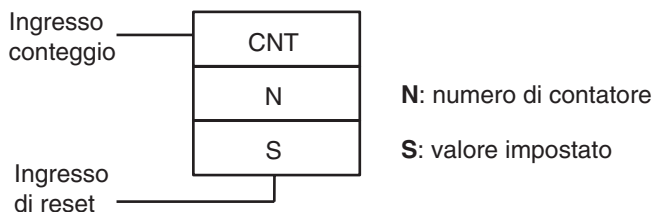


3-6-7 COUNTER: CNT/CNTX(546)

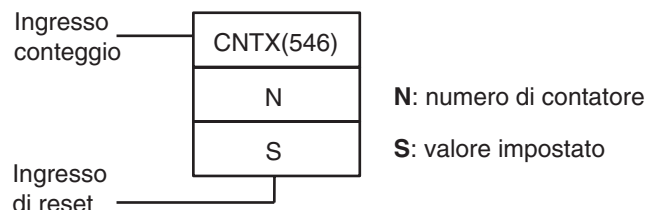
Scopo CNT/CNTX(546) attiva un contatore decrementale. L'intervallo di impostazione è compreso tra 0 e 9.999 per CNT e tra 0 e 65.535 per CNTX(546).

Simbolo programmazione ladder

BCD



Binario



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | CNT/ CNTX(546) |
|-------------------------|--|-------------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Operandi

N: numero contatore

Il numero del contatore deve essere compreso tra 0000 e 4095 (decimale).

S: valore impostato

| Dati | Intervallo |
|---------|---|
| BCD | Da #0000 a #9999 |
| Binario | Da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |

Caratteristiche operando

| Area | N | S |
|-------------------------|---------------------------|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritentività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | Da 0000 a 4095 (decimale) | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |

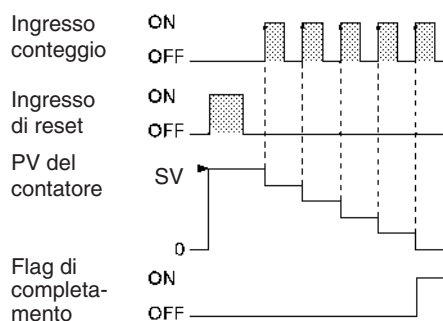
| Area | N | S |
|--|--|---|
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- | BCD: Da #0000 a 9999 (BCD) Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

Il valore attuale (PV) del contatore viene decrementato di 1 ogni volta che l'ingresso conteggio passa da OFF a ON. Quando il valore attuale PV raggiunge 0, il flag di completamento viene impostato su ON.

Una volta attivato il flag di completamento, reimpostare il contatore attivando l'ingresso di reset o utilizzando l'istruzione CNR(545)/CNRX(547). Altrimenti non è possibile riavviare il contatore.

Quando l'ingresso di reset è ON, il contatore è reimpostato e l'ingresso conteggio viene ignorato. Quando un contatore è reimpostato, il valore attuale corrispondente viene a sua volta reimpostato in base al valore impostato e il flag di completamento passa a OFF.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se N è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un contatore. ON se in modalità BCD e se S non contiene dati BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF. Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

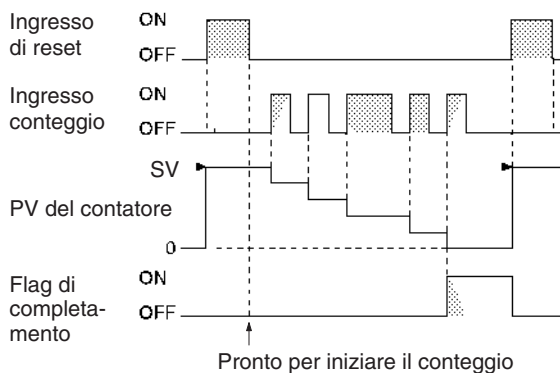
Avvertenze

I numeri di contatore sono condivisi dalle istruzioni CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) e CNTWX(818). Se due contatori condividono lo stesso numero di contatore, ma non sono utilizzati contemporaneamente, viene generato un errore di duplicazione durante il controllo del programma; il funzionamento dei contatori sarà comunque regolare. I contatori che condividono lo stesso numero di contatore non funzionano in modo corretto se vengono utilizzati contemporaneamente.

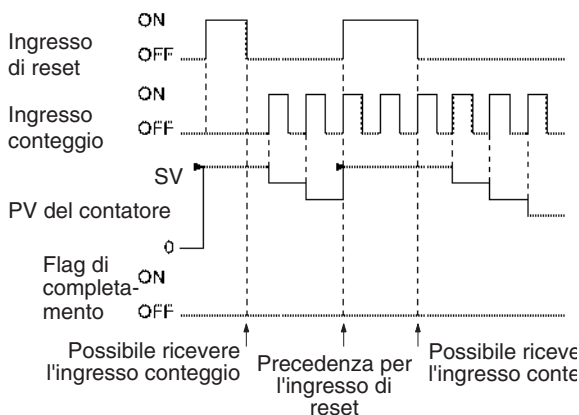
Il valore attuale (PV) di un contatore viene aggiornato quando l'ingresso conteggio passa da OFF a ON; il flag di completamento si aggiorna a ogni esecuzione di CNT/CNTX(546). Se il valore attuale (PV) è 0, il flag di completamento viene attivato, altrimenti è impostato su OFF.

Quando un contatore per CNT/CNTX(546) è caratterizzato da impostazione forzata, il flag di completamento corrispondente viene impostato su ON e il valore attuale (PV) è reimpostato su 0000. Quando tale contatore è caratterizzato da ripristino forzato, il flag di completamento corrispondente viene impostato su OFF e il valore attuale (PV) viene impostato in base al valore impostato (SV).

Prima di iniziare il conteggio con l'ingresso conteggio, assicurarsi di reimpostare il contatore modificando l'impostazione dell'ingresso di reset in questo modo: OFF → ON → . Osservare il diagramma seguente. Se l'ingresso di reset è impostato su ON, l'ingresso conteggio non viene ricevuto.



L'ingresso di reset ha la precedenza e il contatore viene reimpostato se l'ingresso di reset e l'ingresso conteggio sono entrambi e contemporaneamente impostati su ON. Il valore attuale (PV) viene reimpostato in base al valore impostato (SV) e il flag di completamento passa a OFF.



Il funzionamento dei flag = ed N dipende dal modello della CPU. Per ulteriori informazioni, fare riferimento a *Flag* poco sopra.

Nota Se si utilizza la modifica in linea per aggiungere un contatore, è necessario procedere innanzitutto alla reimpostazione del contatore stesso per assicurare un regolare funzionamento. Se il contatore non è reimpostato, come valore attuale (PV) del contatore viene utilizzato il valore precedente e il contatore potrebbe non funzionare una volta scritto.

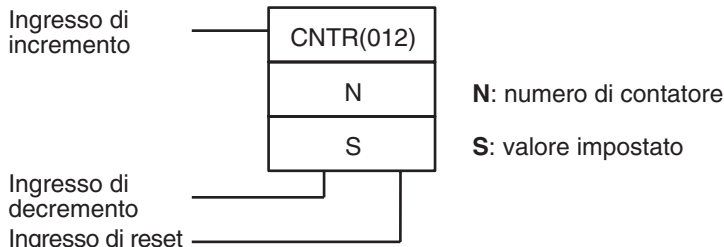
I valori attuali (PV) del contatore vengono mantenuti anche in seguito a una caduta di tensione. Se si desidera riavviare il conteggio dal valore impostato (SV) invece di riprenderlo dal valore attuale (PV) mantenuto, aggiungere il flag di primo ciclo (A20011) come ingresso di reset al contatore.



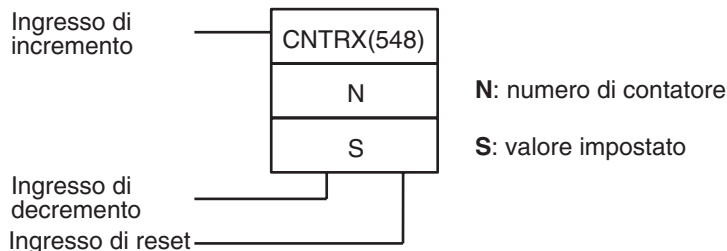
3-6-8 REVERSIBLE COUNTER: CNTR(012)/CNTRX(548)

Scopo CNTR(012)/CNTRX(548) attiva un contatore reversibile.

Simbolo programmazione ladder **BCD**



Binario



Variazioni

| | | |
|--------------------------------|---|--------------------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CNTR(012)/ CNTRX(548) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Operandi

N: Numero contatore

Il numero del contatore deve essere compreso tra 0000 e 4095 (decimale).

S: valore impostato

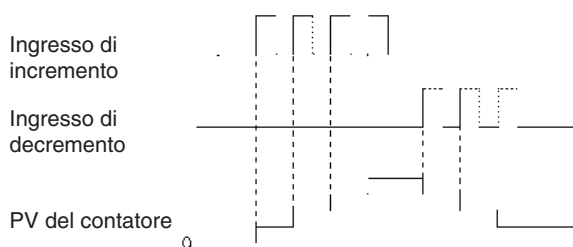
| Dati | Intervallo |
|---------|---|
| BCD | Da #0000 a #9999 |
| Binario | Da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |

Caratteristiche operando

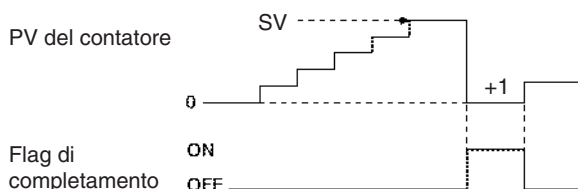
| Area | N | S |
|--|--|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | Da 0000 a 4095 (decimale) | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- | BCD: Da #0000 a 9999 (BCD) Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

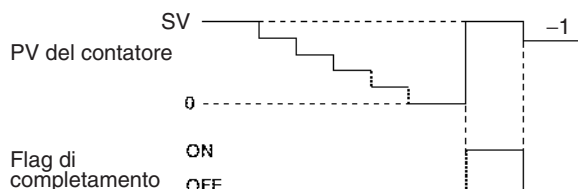
Il valore attuale (PV) del contatore viene incrementato di 1 ogni volta che l'ingresso di incremento passa da OFF a ON e decrementato di 1 ogni volta che l'ingresso di decremento passa da OFF a ON. Il valore attuale può variare tra 0 e il valore impostato.



Durante l'incremento il flag di completamento è impostato su ON quando il valore attuale (PV) viene incrementato a partire dal valore impostato (SV) e torna a 0: il flag di completamento quindi passa di nuovo a OFF quando il valore attuale (PV) viene incrementato da 0 a 1.



Durante il decremento il flag di completamento viene impostato su ON quando il valore attuale (PV) viene decrementato da 0 fino al valore impostato (SV) e passa di nuovo a OFF quando il valore attuale (PV) viene decrementato dal valore impostato (SV) a SV-1.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se N è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un contatore. ON se in modalità BCD e se S non contiene dati BCD. OFF in tutti gli altri casi. |

Avvertenze

I numeri di contatore sono condivisi dalle istruzioni CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) e CNTWX(818). Se due contatori condividono lo stesso numero di contatore, ma non sono utilizzati contemporaneamente, viene generato un errore di duplicazione durante il controllo del programma; il funzionamento dei contatori sarà comunque regolare. I contatori che condividono lo stesso numero di contatore non funzionano in modo corretto se vengono utilizzati contemporaneamente.

Il valore attuale (PV) non viene modificato se entrambi gli ingressi di incremento e decremento passano da OFF a ON contemporaneamente. Se l'ingresso di reset è impostato su ON, il valore attuale (PV) viene reimpostato su 0 ed entrambi gli ingressi conteggio verranno ignorati.

Il flag di completamento viene attivato soltanto quando il valore attuale (PV) è stato incrementato dal valore impostato (SV) a 0 o decrementato da 0 al valore impostato (SV); in tutti gli altri casi è disattivato.

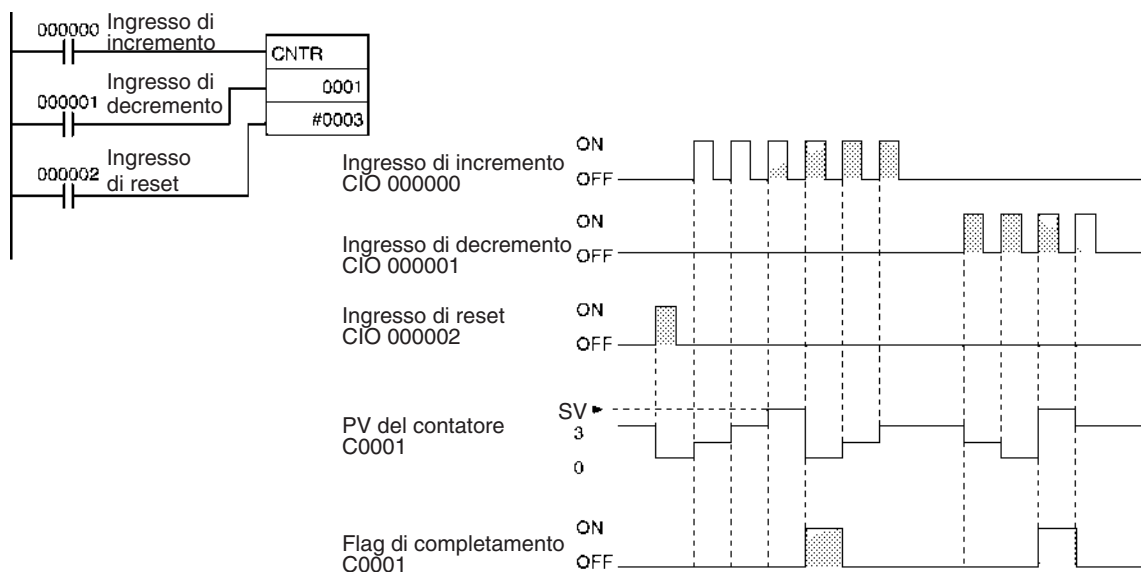
Quando si specifica l'istruzione CNTR(012)/CNTRX(548) con codici mnemonici, indicare in primo luogo l'ingresso di incremento (II), quindi l'ingresso di decremento (DI), l'ingresso di reset (R) e infine l'istruzione CNTR(012)/CNTRX(548). Se si utilizzano diagrammi ladder, indicare in primo luogo l'ingresso di incremento (II), quindi l'istruzione CNTR(012)/CNTRX(548), l'ingresso di decremento (DI) e infine l'ingresso di reset (R).

Esempi

Funzionamento di base di CNTR(012)/CNTRX(548)

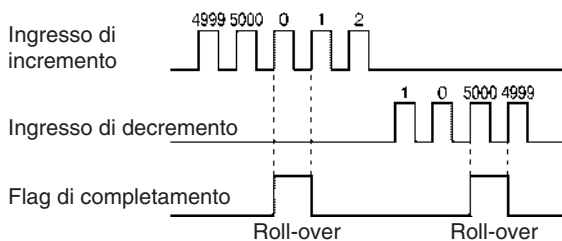
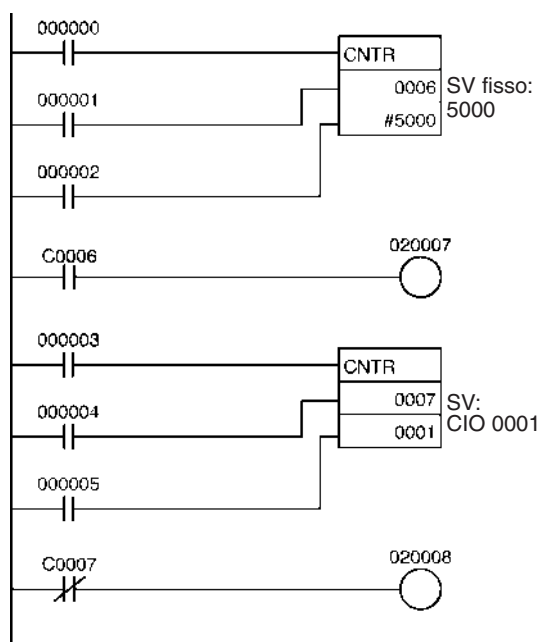
Il valore attuale (PV) viene reimpostato a 0 attivando e disattivando l'ingresso di reset (CIO 000002). Il valore attuale (PV) viene incrementato di 1 ogni volta che l'ingresso di incremento (CIO 000000) passa da OFF a ON. Quando il valore attuale (PV) è incrementato dal valore impostato SV (03), viene reimpostato automaticamente su 0 e il flag di completamento passa a ON.

In modo analogo, il valore attuale (PV) viene decrementato di 1 ogni volta che l'ingresso di decremento (CIO 000001) passa da OFF a ON. Quando il valore attuale (PV) è decrementato da 0, viene impostato in modo automatico sul valore impostato SV (03) e il flag di completamento passa a ON.



Specifica di un valore impostato (SV) in un canale

Nell'esempio seguente, il valore impostato (SV) per CNTR(012) 0007 viene determinato dal contenuto di CIO 0001. Se tale contenuto è controllato da un selettore esterno, è possibile modificare il valore impostato manualmente dal selettore.

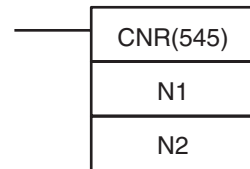


3-6-9 RESET TIMER/COUNTER: CNR(545)/CNRX(547)

Scopo Reimposta i temporizzatori o i contatori che rientrano nell'intervallo dei numeri di temporizzatore o contatore specificato.

Simbolo programmazione ladder

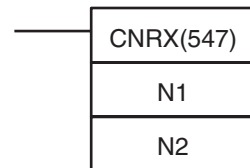
BCD



N₁: primo numero nell'intervallo

N₂: ultimo numero nell'intervallo

Binario



N₁: primo numero nell'intervallo

N₂: ultimo numero nell'intervallo

Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | CNR(545)/ CNRX(547) |
|--------------------------------|--|--------------------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ CNR(545)/ CNRX(547) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N₁: primo numero nell'intervallo

N₁ deve essere un numero di temporizzatore compreso tra T0000 e T4095 oppure un numero di contatore compreso tra C0000 e C4095.

N₂: ultimo numero nell'intervallo

N₂ deve essere un numero di temporizzatore compreso tra T0000 e T4095 oppure un numero di contatore compreso tra C0000 e C4095.

Nota N₁ ed N₂ devono trovarsi nella stessa area dati, cioè N₁ ed N₂ devono essere numeri temporizzatore o contatore.

Caratteristiche operando

| Area | N ₁ | N ₂ |
|--------------------------------------|------------------|------------------|
| Area CIO | --- | --- |
| Area di lavoro | --- | --- |
| Area bit di ritenività | --- | --- |
| Area bit ausiliaria | --- | --- |
| Area del temporizzatore | Da C0000 a C4095 | Da C0000 a C4095 |
| Area del contatore | Da T0000 a T4095 | Da T0000 a T4095 |
| Area DM | --- | --- |
| Area EM senza banco | --- | --- |
| Area EM con banco | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- |
| Indirizzi DM/EM indiretti in BCD | --- | --- |

| Area | N ₁ | N ₂ |
|--|--|----------------|
| Costanti | --- | --- |
| Registri dati | --- | --- |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

CNR(545)/CNRX(547) reimposta i flag di completamento di tutti i temporizzatori o contatori da N₁ a N₂. Allo stesso tempo tutti i valori attuali (PV) vengono impostati sul valore massimo (9999 per BCD ed FFFF per il formato binario). Il valore attuale (PV) viene impostato sul valore impostato (SV) la volta successiva all'esecuzione dell'istruzione del temporizzatore o del contatore.

Temporizzatori reimposti da CNR(545)/CNRX(547)

Nei temporizzatori indicati qui di seguito si verifica la reimpostazione se i numeri di temporizzatore rientrano nell'intervallo specificato: TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) e TMHWX(817). Quando un temporizzatore viene reimpostato, il flag di completamento corrispondente è disattivato e il valore attuale (PV) viene impostato sul valore massimo di 9999.

Nota I temporizzatori TIML(542), TIMXL(553), MTIM(543) e MTIMX(554) non vengono reimposti dall'istruzione CNR(545)/CNRX(547) poiché non utilizzano numeri di temporizzatore.

Contatori reimposti da CNR(545)/CNRX(547)

Nei contatori indicati qui di seguito si verifica la reimpostazione se i numeri di contatore rientrano nell'intervallo specificato: CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) e CNTWX(818). Quando un contatore viene reimpostato, il flag di completamento corrispondente è disattivato e il valore attuale (PV) viene impostato sul valore massimo di 9999.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se N ₁ è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un temporizzatore o di un contatore. ON se N ₂ è indirizzato indirettamente tramite un registro indice; l'indirizzo nel registro indice, tuttavia, non corrisponde all'indirizzo PV di un temporizzatore o di un contatore. ON se N ₁ ed N ₂ non si trovano nella stessa area dati. OFF in tutti gli altri casi. |

Avvertenze

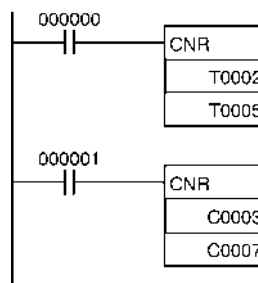
CNR(545)/CNRX(547) non procede alla reimpostazione delle istruzioni stesse di temporizzatore o contatore, bensì dei valori attuali (PV) e dei flag di completamento assegnati a tali istruzioni. Nella maggior parte dei casi l'effetto di CNR(545)/CNRX(547) è diverso da quello della reimpostazione diretta delle istruzioni. Ad esempio, quando un'istruzione per TIM/TIMX(550) viene reimpostata direttamente, il valore attuale (PV) viene impostato sul valore impostato (SV); quando tuttavia lo stesso temporizzatore viene reimpostato da CNR(545)/CNRX(547), il PV corrispondente viene impostato sul valore massimo di 9999.

Quando N1 ed N2 sono indicati con $N1 > N2$, si verifica soltanto la reimpostazione del flag di completamento per il numero di contatore/temporizzatore.

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i flag di completamento per i temporizzatori da T0002 a T0005 vengono disattivati e i valori attuali (PV) dei temporizzatori sono impostati sul valore massimo di 9999.

Quando CIO 000001 è impostato su ON, i flag di completamento per i contatori da C0003 a C0007 vengono disattivati e i valori attuali (PV) dei contatori sono impostati sul valore massimo di 9999.



3-6-10 Applicazioni di esempio per temporizzatori e contatori

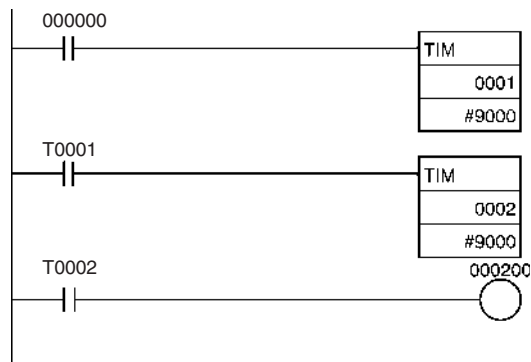
Negli esempi che seguono sono illustrate alcune applicazioni di istruzioni per temporizzatori e contatori: temporizzatori a lungo termine, un contatore a due stadi, ritardo di attivazione/disattivazione, bit a singola esecuzione e bit di lampeggiamento.

Esempio 1: Temporizzatori a lungo termine

Gli esempi di programmi seguenti descrivono tre procedure per creare temporizzatori a lungo termine con istruzioni standard TIM e CNT.

Due istruzioni TIM

In questo esempio due istruzioni TIM vengono combinate per costituire un temporizzatore di 30 minuti.

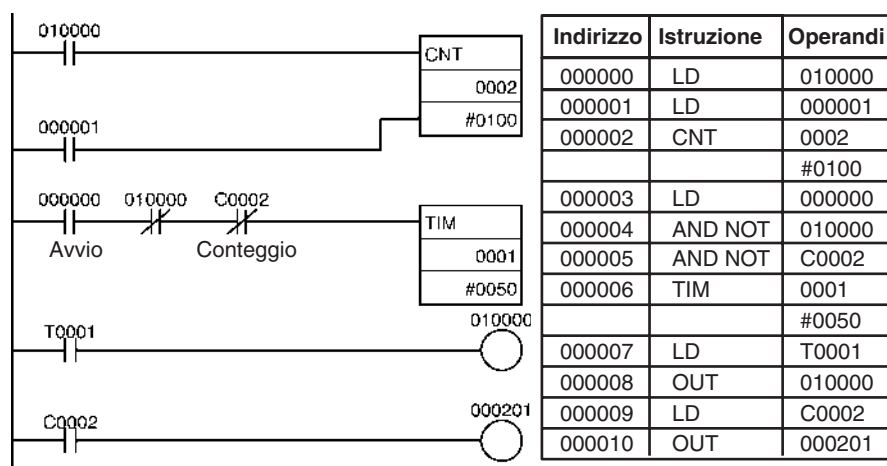


| Indirizzo | Istruzione | Operandi |
|-----------|------------|---------------|
| 000000 | LD | 000000 |
| 000001 | TIM | 0001 #9000 |
| 000002 | LD | T0001 |
| 000003 | TIM | 0002 #9000 |
| 000004 | LD | T0002 |
| 000005 | OUT | 000200 |

Istruzioni TIM e CNT

In questo esempio un'istruzione TIM e un'istruzione CNT vengono combinate per costituire un temporizzatore di 500 secondi.

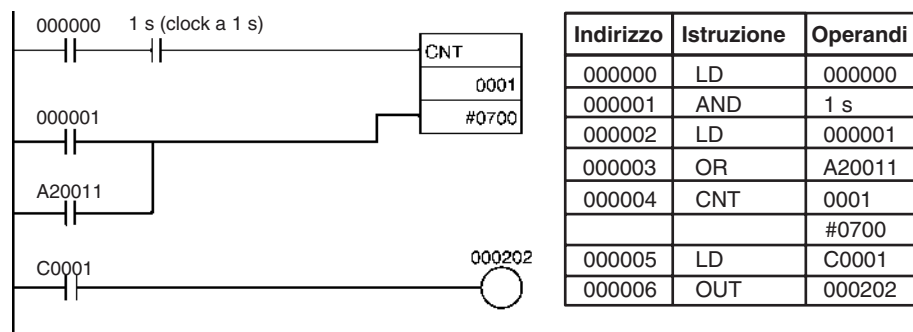
TIM 0001 genera un impulso ogni 5 s e CNT 0002 conta tali impulsi. Il valore impostato per questa combinazione è l'intervallo del temporizzatore \times il valore impostato (SV) del contatore. In questo caso, il valore impostato (SV) del temporizzatore sarebbe $5 \text{ s} \times 100 = 500 \text{ s}$. Con questa combinazione il valore attuale (PV) del temporizzatore è in effetti il valore attuale (PV) di un contatore che viene mantenuto anche in seguito a cadute di tensione.



Istruzioni dell'impulso di clock e CNT

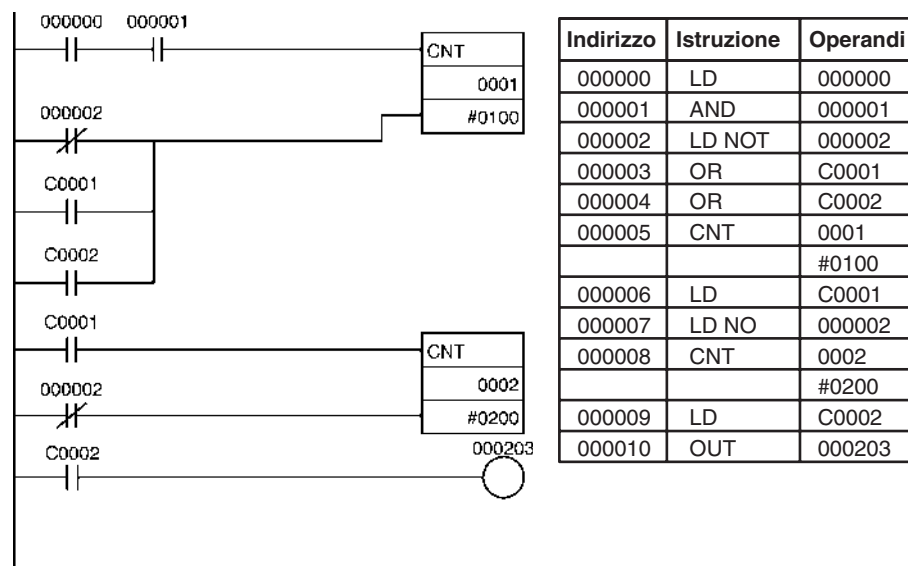
In questo esempio un'istruzione CNT conteggia gli impulsi a partire dall'impulso di clock a 1 s al fine di costituire un temporizzatore di 700 secondi.

Se il flag di primo ciclo (A20011) è posto in OR con l'ingresso di reset del contatore (CIO 000001), all'inizio dell'esecuzione del programma il valore attuale (PV) del contatore verrà reimpostato sull'SV (0700) piuttosto che riprendere il conteggio dal valore attuale (PV) precedente.



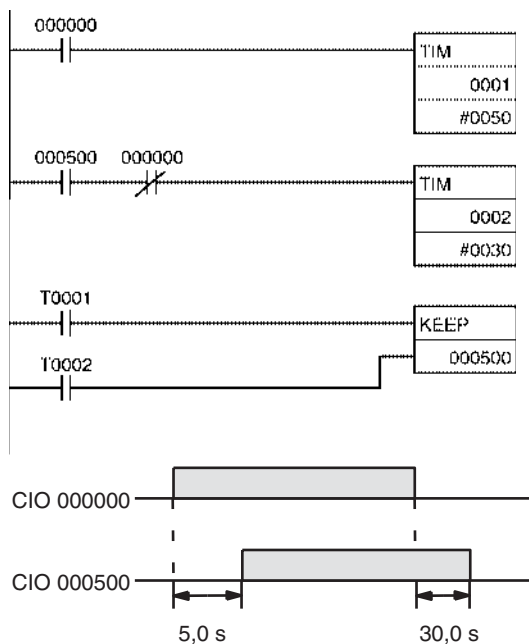
Esempio 2: Contatore a due stadi

Quando è necessario un valore impostato (SV) maggiore di 9999, è possibile combinare due contatori. Osservare l'esempio qui di seguito. In questo caso, due istruzioni CNT vengono combinate per costituire un contatore BCD con un valore impostato (SV) pari a 20.000.



**Esempio 3:
Ritardo di attivazione/
disattivazione**

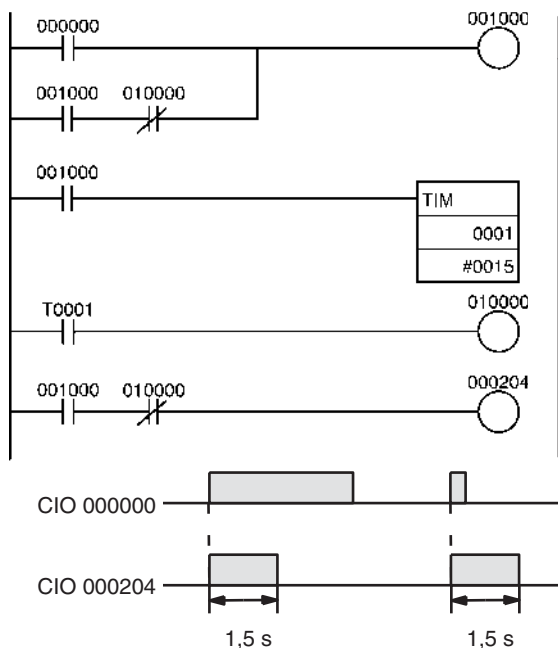
In questo esempio due temporizzatori TIM vengono combinati con l'istruzione KEEP(011) per costituire un ritardo di attivazione e un ritardo di disattivazione. CIO 000500 viene impostato su ON 5,0 secondi dopo l'attivazione di CIO 000000 e passa a OFF 3,0 secondi dopo la disattivazione di CIO 000000.



| Indirizzo | Istruzione | Operandi |
|-----------|------------|---------------|
| 000000 | LD | 000000 |
| 000001 | TIM | 0001 #0050 |
| 000002 | LD | 000500 |
| 000003 | AND NOT | 000000 |
| 000004 | TIM | 0002 #0030 |
| 000005 | LD | T0001 |
| 000006 | LD | T0002 |
| 000007 | KEEP(011) | 000500 |

**Esempio 4:
Bit a singola esecuzione**

È possibile combinare un temporizzatore TIM con le istruzioni OUT oppure OUT NOT al fine di controllare il tempo esatto durante il quale un determinato bit resta impostato su ON o su OFF. In questo esempio, CIO 000204 viene impostato su ON per 1,5 secondi (l'SV di T0001) dopo l'attivazione di CIO 000000.



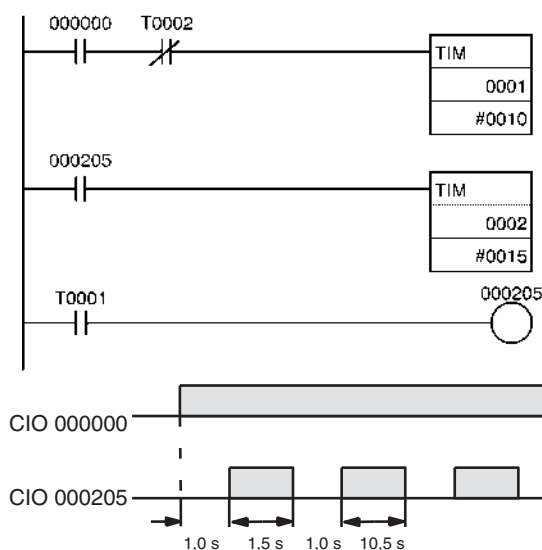
| Indirizzo | Istruzione | Operandi |
|-----------|------------|---------------|
| 000000 | LD | 000000 |
| 000001 | LD | 001000 |
| 000002 | AND NOT | 010000 |
| 000003 | OR | 000000 |
| 000004 | OUT | 001000 |
| 000005 | LD | 001000 |
| 000006 | TIM | 0001 #0015 |
| 000007 | LD | T0001 |
| 000008 | OUT | 010000 |
| 000009 | LD | 001000 |
| 000010 | AND NOT | 010000 |
| 000011 | OUT | 000204 |

Esempio 4: Bit di lampeggiamento

Negli esempi di programmi seguenti vengono descritte due procedure per la creazione di bit di lampeggiamento. Nel secondo esempio viene solamente simulato un impulso di clock.

Due istruzioni TIM

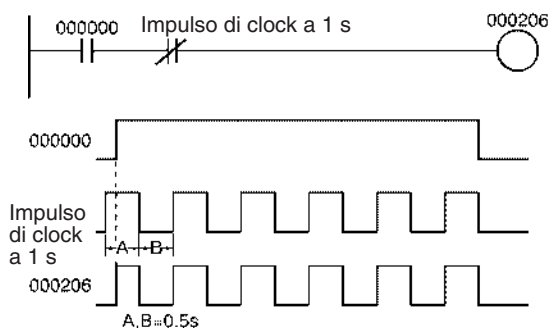
Quando la condizione di esecuzione è impostata su ON è possibile combinare due temporizzatori per fare in modo che un bit passi da ON a OFF a intervalli regolari. In questo esempio, CIO 000205 viene disattivato per 1,0 secondi e quindi attivato per 1,5 secondi fino a quando CIO 000000 è impostato su ON.



| Indirizzo | Istruzione | Operandi |
|-----------|------------|---------------|
| 000000 | LD | 000000 |
| 000001 | AND | T0002 |
| 000002 | TIM | 0001 #0010 |
| 000003 | LD | 000205 |
| 000004 | TIM | 0002 #0015 |
| 000005 | LD | T0001 |
| 000006 | OUT | 000205 |

Impulso di clock

Per simulare l'impulso di clock (0,1 s, 0,2 s o 1,0 s) è possibile combinare la condizione di esecuzione prescelta con un impulso di clock.



| Indirizzo | Istruzione | Operandi |
|-----------|------------|----------|
| 000000 | LD | 000000 |
| 000001 | AND | 1s |
| 000002 | OUT | 000206 |

3-6-11 Indirizzamento indiretto dei numeri di temporizzatore/contatore

Utilizzando registri indice è possibile eseguire un indirizzamento indiretto dei numeri di temporizzatore e contatore. Se per l'indirizzamento indiretto si utilizzano registri indice, specificare MOV_{RW}(561) (MOVE TIMER/COUNTER PV TO REGISTER) per impostare sul registro indice desiderato l'indirizzo di memoria del PLC del valore attuale (PV) del temporizzatore o contatore prescelto.

Utilizzando registri indice è possibile eseguire un indirizzamento indiretto dei seguenti temporizzatori e contatori. TIM, TIMX(550), TIMH(015), TIMHX(551), TTIM(087), TTIMX(555), TMHH(540), TMHHX(552), TIMW(813), TIMWX(816), TMHW(815), TMHWX(817), CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) e CNTWX(818). Si tratta di temporizzatori e contatori che utilizzano numeri di temporizzatori e contatori.

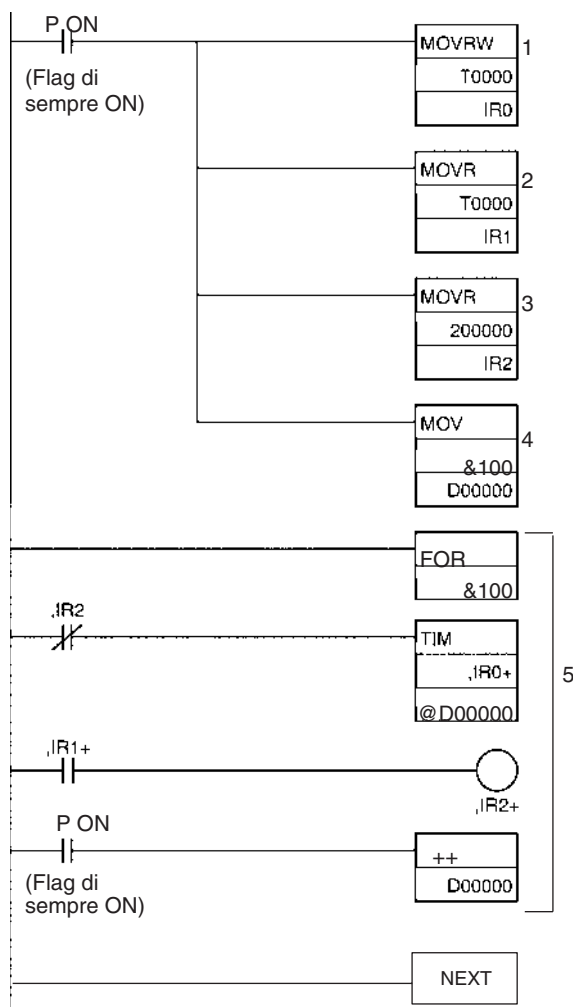
Se l'indirizzo di memoria del PLC specificato nel registro indice non è l'indirizzo di un valore attuale (PV) di un temporizzatore o di un contatore, l'istruzione per temporizzatore o contatore non viene eseguita.

L'utilizzo dei registri indice per eseguire un indirizzamento indiretto di temporizzatori e contatori può ridurre le dimensioni di un programma e aumentare la flessibilità. Ad esempio, è possibile creare normali subroutine.

Esempio

Nell'esempio seguente viene illustrata una sezione del programma che utilizza l'indirizzamento indiretto per definire e avviare 100 temporizzatori con i valori impostati (SV) compresi tra D00100 e D00109. IR0 contiene l'indirizzo di memoria del PLC del valore attuale (PV) del temporizzatore; IR1 contiene l'indirizzo di memoria del PLC del flag di completamento del temporizzatore.

| Indirizzo DM | Contenuto | Funzione |
|--------------|-----------|--------------|
| D00100 | 0010 | SV per T0000 |
| D00101 | 0100 | SV per T0001 |
| D00102 | 0050 | SV per T0002 |
| . | . | . |
| . | . | . |
| . | . | . |
| D00199 | 0999 | SV per T0099 |



- 1,2,3...** 1. L'istruzione MOV RW(561) sposta a IR0 l'indirizzo di memoria del PLC del valore attuale (PV) per il temporizzatore T0000. In seguito IR0 può essere utilizzato al posto del numero di temporizzatore.

2. L'istruzione MOVR (560) sposta a IR1 l'indirizzo di memoria del PLC del flag di completamento per il temporizzatore T0000.
3. L'istruzione MOVR (560) sposta l'indirizzo di memoria del PLC di CIO 200000 in IR2.
4. MOV(021) sposta &100 in D00000 per l'indirizzamento indiretto dei valori impostati (SV) del temporizzatore.
5. Il contenuto di IR0, IR1, IR2 e D00000 viene incrementato di 1 ogni volta: questo ciclo, infatti, viene eseguito 100 volte avviando i temporizzatori compresi tra T0000 e T0099.

Il ciclo del programma appena osservato è provvisto di quattro parametri di ingresso utilizzati per avviare tutti i 100 temporizzatori con questa normale subroutine.

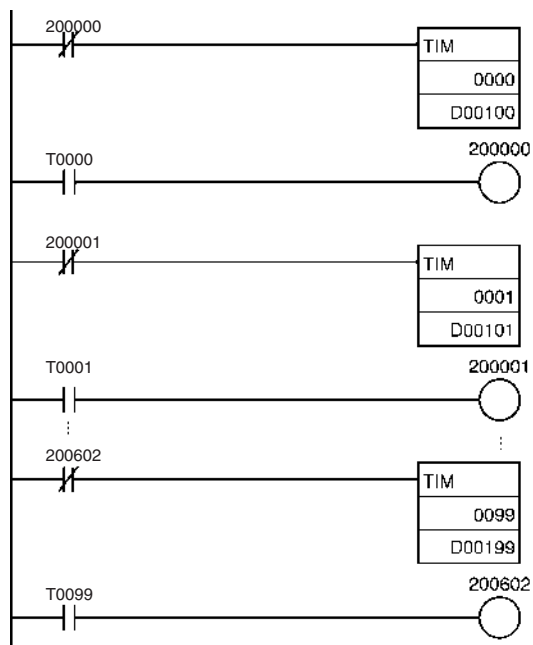
IR0 L'indirizzo di memoria del PLC del valore attuale (PV) del temporizzatore

IR1 L'indirizzo di memoria del PLC del flag di completamento del temporizzatore

IR2 L'indirizzo di memoria del PLC della condizione di esecuzione del temporizzatore

D00000 l'indirizzo DM del canale che contiene il valore impostato (SV) del temporizzatore

Questa subroutine equivale alle 400 istruzioni indicate qui di seguito.



| Indirizzo | Istruzione | Operandi |
|-----------|------------|----------|
| 000000 | LD NOT | 200000 |
| 000001 | TIM | 0000 |
| | | D00100 |
| 000002 | LD | T0000 |
| 000003 | OUT | 200000 |
| 000004 | LD NOT | 200001 |
| 000005 | TIM | 0001 |
| | | D00101 |
| 000006 | LD | T0001 |
| 000007 | OUT | 200001 |
| 000008 | LD NOT | 200002 |
| 000009 | TIM | 0002 |
| | | D00102 |
| 000010 | LD | T0002 |
| 000011 | OUT | 200002 |
| ≈ | | |
| 000396 | LD NOT | 200602 |
| 000397 | TIM | 0099 |
| | | D00199 |
| 000398 | LD | T0000 |
| 000399 | OUT | 200602 |

3-7 Istruzioni di confronto

In questa sezione vengono descritte le istruzioni utilizzate per confrontare in vari modi i dati di diverse lunghezze.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|---------------------------------------|---|-----------------|--------|
| Istruzioni di confronto di ingresso | =, <>, <, <=, >, >= (S, L) (LD, AND, OR) | Da 300 a 328 | 275 |
| Istruzioni di confronto di date e ore | =DT, <>DT, <DT, <=DT, >DT, >=DT (LD, AND, OR) | Da 341 a 346 | 281 |
| COMPARE | CMP | 020 | 287 |
| DOUBLE COMPARE | CMPL | 060 | 290 |
| SIGNED BINARY COMPARE | CPS | 114 | 293 |
| DOUBLE SIGNED BINARY COMPARE | CPSL | 115 | 296 |
| MULTIPLE COMPARE | MCMP | 019 | 299 |
| TABLE COMPARE | TCMP | 085 | 301 |
| BLOCK COMPARE | BCMP | 068 | 304 |
| EXPANDED BLOCK COMPARE | BCMP2 | 502 | 306 |

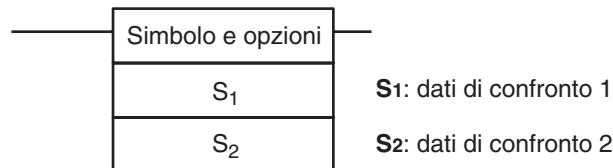
3-7-1 Istruzioni di confronto di ingresso (da 300 a 328)

Scopo

Le istruzioni di confronto di ingresso confrontano due valori (costanti e/o il contenuto di canali specifici) e creano una condizione di esecuzione ON quando la condizione di confronto è vera. Le istruzioni di confronto di ingresso vengono utilizzate per confrontare dati con segno o senza segno a un canale o a doppia lunghezza.

Nota Consultare la sezione 3-15-21 *Istruzioni di confronto a virgola mobile in singola precisione* per ulteriori informazioni sulle istruzioni di confronto di ingresso a virgola mobile in singola precisione e la sezione 3-16-21 *Istruzioni di ingresso a virgola mobile in doppia precisione* per ulteriori informazioni sulle istruzioni di confronto di ingresso a virgola mobile in doppia precisione.

Simbolo programmazione ladder



Variazioni

| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il confronto risulta vero | Istruzione di confronto di ingresso |
|-------------------------|--|-------------------------------------|
| Aggiornamento immediato | | Non supportata |

Arete di programma applicabili

| Arete di programma a blocchi | Arete di programma step | Subroutine | Task ad interrupt |
|------------------------------|-------------------------|------------|-------------------|
| OK | OK | OK | OK |

Specifiche dell'operando nelle istruzioni per dati a un canale

| Area | S ₁ | S ₂ |
|----------------|------------------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |

| Area | S ₁ | S ₂ |
|--|--|----------------|
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

**Specifiche dell'operando
nelle istruzioni per dati a
doppia lunghezza**

| Area | S ₁ | S ₂ |
|--------------------------------------|--|----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFF (binario) | |
| Registri dati | --- | |

| Area | S ₁ | S ₂ |
|--|--|----------------|
| Registri indice | Da IR0 a IR15 (solo per dati senza segno) | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

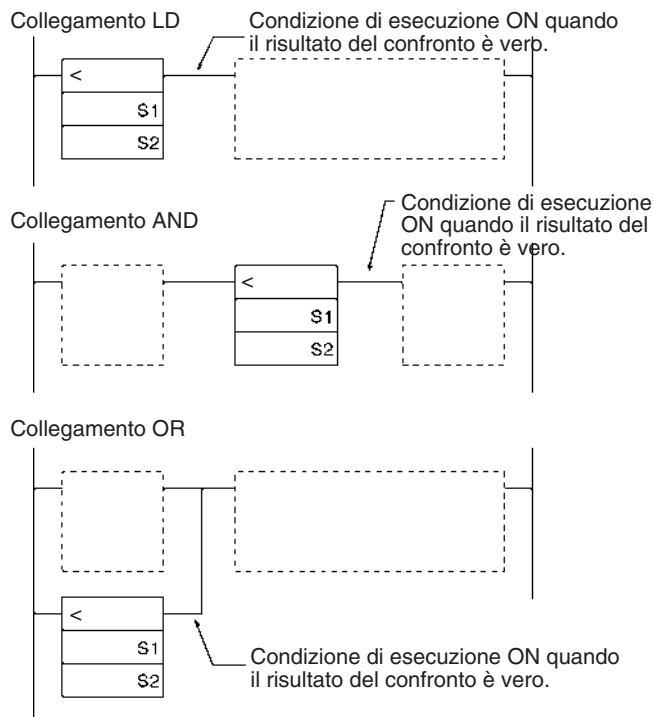
Descrizione

L'istruzione di confronto di ingresso confronta S₁ e S₂ come valori con segno o senza segno e crea una condizione di esecuzione ON quando la condizione di confronto è vera. Diversamente dalle istruzioni come CMP(020) e CMPL(060), il risultato di un'istruzione di confronto di ingresso si riflette direttamente come condizione di esecuzione, quindi non è necessario accedere al risultato del confronto tramite un flag aritmetico e il programma è più semplice e più veloce.

Immissione di istruzioni

Le istruzioni di confronto di ingresso vengono interpretate esattamente come le istruzioni LD, AND e OR per controllare l'esecuzione delle istruzioni successive.

| Tipo di ingresso | Operazione |
|------------------|---|
| LD | È possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |
| AND | Non è possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |
| OR | È possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |



Opzioni

Le istruzioni di confronto di ingresso possono confrontare dati con segno o senza segno e valori a un canale o doppi. Se non è stata specificata alcuna

opzione, il confronto verrà eseguito su dati senza segno a un canale. Utilizzando questi tre tipi di ingresso e due opzioni, ne derivano 72 diverse istruzioni di confronto di ingresso.

| Simbolo | Opzione (formato dati) | Opzione (lunghezza dati) |
|------------------------|---------------------------|----------------------------|
| = (uguale) | Nessuno: dati senza segno | Nessuno: dati a un canale |
| < > (diverso) | S: dati con segno | L: dati a doppia lunghezza |
| < (minore di) | | |
| <= (minore o uguale) | | |
| > (maggiore di) | | |
| >= (maggiore o uguale) | | |

Le istruzioni di confronto di ingresso senza segno, ovvero le istruzioni senza l'opzione S, possono gestire dati binari o dati BCD senza segno. Le istruzioni di confronto di ingresso con segno, ovvero le istruzioni con l'opzione S, possono gestire dati binari con segno.

Sommario delle istruzioni di confronto di ingresso

Nella tabella riportata di seguito sono indicati i codici funzione, i mnemonici e le funzioni delle 72 istruzioni di confronto di ingresso. Per i confronti a un canale $C1=S_1$ e $C2=S_2$; per i confronti doppi $C1=S_1+1$, S_1 e $C2=S_2+1$, S_2 .

| Codice | Mnemonico | Nome | Funzione |
|--------|-----------|------------------------------|-------------------------|
| 300 | LD = | LOAD EQUAL | Vera se $C1 = C2$ |
| | AND = | AND EQUAL | |
| | OR = | OR EQUAL | |
| 301 | LD=L | LOAD DOUBLE EQUAL | |
| | AND=L | AND DOUBLE EQUAL | |
| | OR=L | OR DOUBLE EQUAL | |
| 302 | LD=S | LOAD SIGNED EQUAL | |
| | AND=S | AND SIGNED EQUAL | |
| | OR=S | OR SIGNED EQUAL | |
| 303 | LD=SL | LOAD DOUBLE SIGNED EQUAL | |
| | AND=SL | AND DOUBLE SIGNED EQUAL | |
| | OR=SL | OR DOUBLE SIGNED EQUAL | |
| 305 | LD <> | LOAD NOT EQUAL | Vera se $C1 \neq C2$ |
| | AND <> | AND NOT EQUAL | |
| | OR <> | OR NOT EQUAL | |
| 306 | LD <>L | LOAD DOUBLE NOT EQUAL | |
| | AND <>L | AND DOUBLE NOT EQUAL | |
| | OR <>L | OR DOUBLE NOT EQUAL | |
| 307 | LD <>S | LOAD SIGNED NOT EQUAL | |
| | AND <>S | AND SIGNED NOT EQUAL | |
| | OR <>S | OR SIGNED NOT EQUAL | |
| 308 | LD <>SL | LOAD DOUBLE SIGNED NOT EQUAL | |
| | AND <>SL | AND DOUBLE SIGNED NOT EQUAL | |
| | OR <>SL | OR DOUBLE SIGNED NOT EQUAL | |

| Codice | Mnemonico | Nome | Funzione |
|--------|-----------|---------------------------------------|--------------------|
| 310 | LD < | LOAD LESS THAN | Vera se C1 < C2 |
| | AND < | AND LESS THAN | |
| | OR < | OR LESS THAN | |
| 311 | LD <L | LOAD DOUBLE LESS THAN | |
| | AND <L | AND DOUBLE LESS THAN | |
| | OR <L | OR DOUBLE LESS THAN | |
| 312 | LD <S | LOAD SIGNED LESS THAN | |
| | AND <S | AND SIGNED LESS THAN | |
| | OR <S | OR SIGNED LESS THAN | |
| 313 | LD <SL | LOAD DOUBLE SIGNED LESS THAN | |
| | AND <SL | AND DOUBLE SIGNED LESS THAN | |
| | OR <SL | OR DOUBLE SIGNED LESS THAN | |
| 315 | LD <= | LOAD LESS THAN OR EQUAL | Vera se C1 ≤ C2 |
| | AND <= | AND LESS THAN OR EQUAL | |
| | OR <= | OR LESS THAN OR EQUAL | |
| 316 | LD <=L | LOAD DOUBLE LESS THAN OR EQUAL | |
| | AND <=L | AND DOUBLE LESS THAN OR EQUAL | |
| | OR <=L | OR DOUBLE LESS THAN OR EQUAL | |
| 317 | LD <=S | LOAD SIGNED LESS THAN OR EQUAL | |
| | AND <=S | AND SIGNED LESS THAN OR EQUAL | |
| | OR <=S | OR SIGNED LESS THAN OR EQUAL | |
| 318 | LD <=SL | LOAD DOUBLE SIGNED LESS THAN OR EQUAL | Vera se C1 ≤ C2 |
| | AND <=SL | AND DOUBLE SIGNED LESS THAN OR EQUAL | |
| | OR <=SL | OR DOUBLE SIGNED LESS THAN OR EQUAL | |
| 320 | LD > | LOAD GREATER THAN | Vera se C1 > C2 |
| | AND > | AND GREATER THAN | |
| | OR > | OR GREATER THAN | |
| 321 | LD >L | LOAD DOUBLE GREATER THAN | |
| | AND >L | AND DOUBLE GREATER THAN | |
| | OR >L | OR DOUBLE GREATER THAN | |
| 322 | LD >S | LOAD SIGNED GREATER THAN | |
| | AND >S | AND SIGNED GREATER THAN | |
| | OR >S | OR SIGNED GREATER THAN | |
| 323 | LD >SL | LOAD DOUBLE SIGNED GREATER THAN | |
| | AND >SL | AND DOUBLE SIGNED GREATER THAN | |
| | OR >SL | OR DOUBLE SIGNED GREATER THAN | |
| 325 | LD >= | LOAD GREATER THAN OR EQUAL | Vera se C1 ≥ C2 |
| | AND >= | AND GREATER THAN OR EQUAL | |
| | OR >= | OR GREATER THAN OR EQUAL | |
| 326 | LD >=L | LOAD DOUBLE GREATER THAN OR EQUAL | |
| | AND >=L | AND DOUBLE GREATER THAN OR EQUAL | |
| | OR >=L | OR DOUBLE GREATER THAN OR EQUAL | |
| 327 | LD >=S | LOAD SIGNED GREATER THAN OR EQUAL | |
| | AND >=S | AND SIGNED GREATER THAN OR EQUAL | |
| | OR >=S | OR SIGNED GREATER THAN OR EQUAL | |
| 328 | LD >=SL | LOAD DBL SIGNED GREATER THAN OR EQUAL | |
| | AND >=SL | AND DBL SIGNED GREATER THAN OR EQUAL | |
| | OR >=SL | OR DBL SIGNED GREATER THAN OR EQUAL | |

Flag

| Nome | Etichetta | Operazione |
|---------------------------|-----------|---|
| Flag di errore | ER | OFF o invariato (vedere nota). |
| Flag di maggiore | > | ON se $S_1 > S_2$ con dati a un canale. ON se $S_1+1, S_1 > S_2+1, S_2$ con dati a doppia lunghezza. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | > = | ON se $S_1 \geq S_2$ con dati a un canale. ON se $S_1+1, S_1 \geq S_2+1, S_2$ con dati a doppia lunghezza. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se $S_1 = S_2$ con dati a un canale. ON se $S_1+1, S_1 = S_2+1, S_2$ con dati a doppia lunghezza. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | ≠ | ON se $S_1 \neq S_2$ con dati a un canale. ON se $S_1+1, S_1 \neq S_2+1, S_2$ con dati a doppia lunghezza. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se $S_1 < S_2$ con dati a un canale. ON se $S_1+1, S_1 < S_2+1, S_2$ con dati a doppia lunghezza. OFF in tutti gli altri casi. |
| Flag di minore o uguale | < = | ON se $S_1 \leq S_2$ con dati a un canale. ON se $S_1+1, S_1 \leq S_2+1, S_2$ con dati a doppia lunghezza. OFF in tutti gli altri casi. |
| Flag negativo | N | OFF o invariato (vedere nota). |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

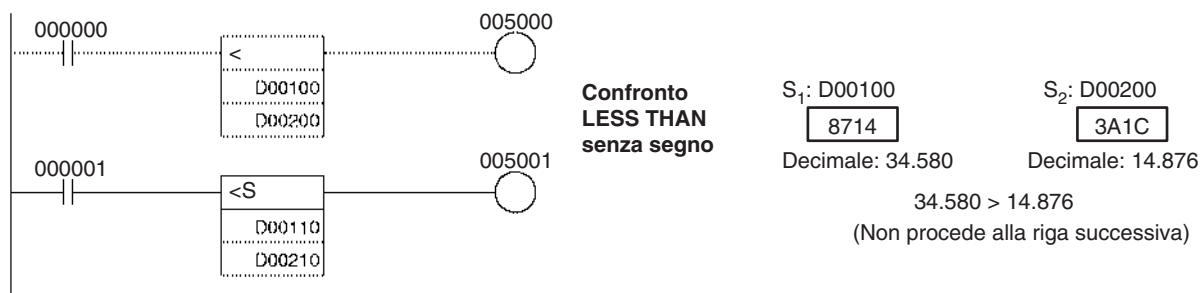
Avvertenze

Non è possibile utilizzare le istruzioni di confronto di ingresso come istruzioni sul lato destro: tra queste istruzioni e la barra di distribuzione destra è infatti necessario specificare un'altra istruzione.

Esempi

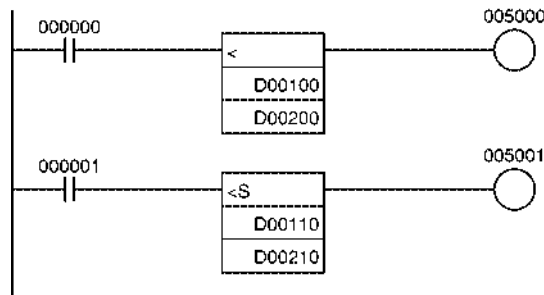
AND LESS THAN: AND<(310)

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i contenuti di D00100 e D00200 vengono confrontati come dati binari senza segno. Se il contenuto di D00100 è inferiore a quello di D00200, CIO 005000 viene attivato e l'esecuzione prosegue dalla riga successiva. Se il contenuto di D00100 non è inferiore a quello di D00200, il resto della riga di istruzione viene ignorato e l'esecuzione passa alla riga successiva dell'istruzione.

**AND SIGNED LESS THAN: AND<S(312)**

Nell'esempio seguente, quando CIO 000001 è impostato su ON, i contenuti di D00110 e D00210 vengono confrontati come dati binari con segno. Se il contenuto di D00110 è inferiore a quello di D00210, CIO 005001 viene attivato e l'esecuzione prosegue dalla riga successiva. Se il contenuto di

D00110 non è inferiore a quello di D00210, il resto della riga di istruzione viene ignorato e l'esecuzione passa alla riga successiva dell'istruzione.



Confronto LESS THAN con segno

S₁: D00110
 8714
 Decimale: -30.956

S₂: D00210
 3A1C
 Decimale: 14.876

-30.956 < 14.876
 (Procede alla riga)

3-7-2 Istruzioni di confronto di date e ore (da 341 a 346)

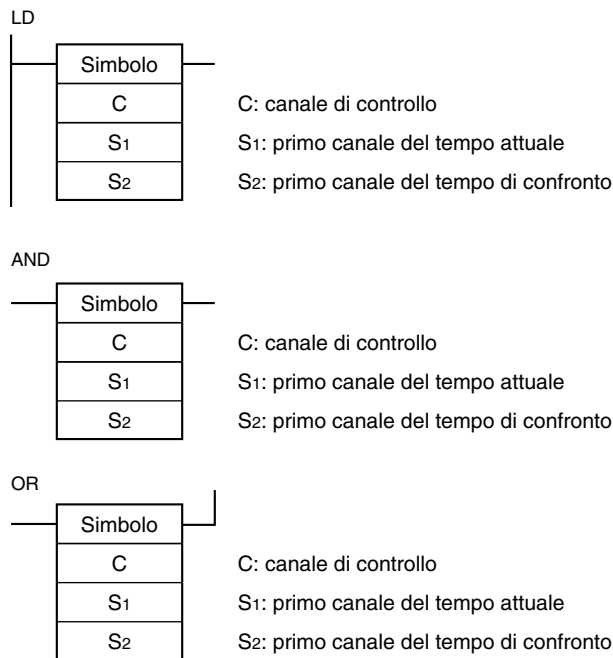
Scopo

Le istruzioni di confronto di date e ore confrontano due valori temporali in formato BCD e creano una condizione di esecuzione ON quando la condizione di confronto è soddisfatta.

Le istruzioni di confronto di date e ore vengono interpretate esattamente come le istruzioni LD, AND e OR per controllare l'esecuzione delle istruzioni successive.

Queste istruzioni sono supportate solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|---------------------------------------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il confronto risulta vero | Istruzioni di confronto di date e ore |
| Aggiornamento immediato | | Non supportata |

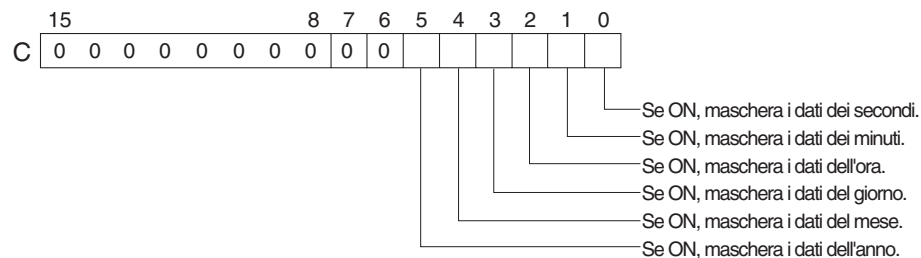
Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

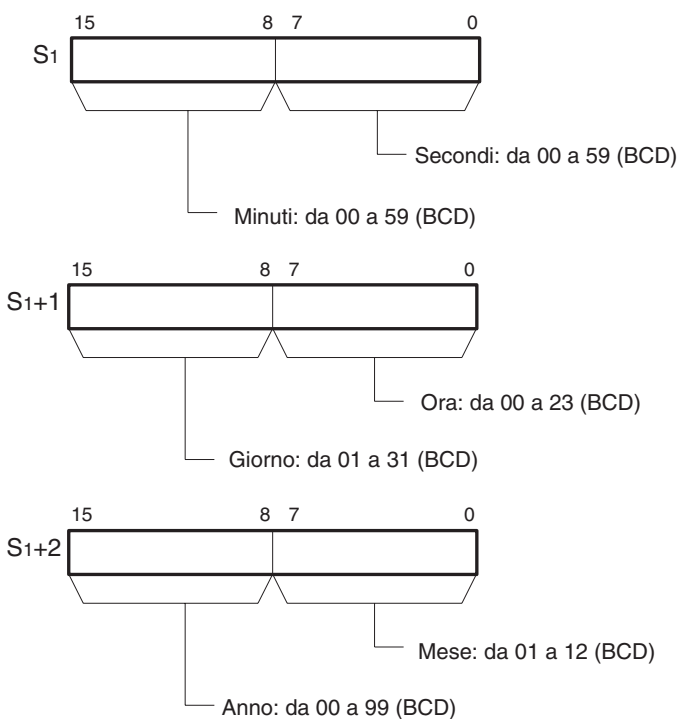
Operandi

C: canale di controllo

I bit da 00 a 05 di C specificano se i dati di data e ora verranno o meno mascherati per il confronto. I bit da 00 a 05 mascherano rispettivamente i secondi, i minuti, le ore, il giorno, il mese e l'anno. Se tutti i 6 valori vengono mascherati, l'istruzione non verrà eseguita, la condizione di esecuzione sarà OFF e verrà attivato il flag di errore.

**Da S₁ a S₁+2: dati del tempo attuale**

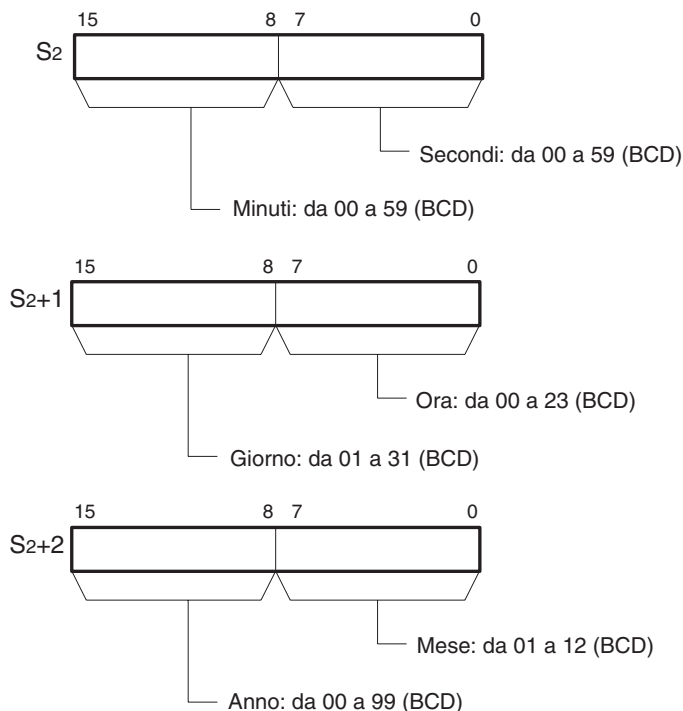
Da S₁ a S₁+2 vi sono i dati del tempo attuale. Da S₁ a S₁+2 devono essere nella stessa area dati.



Nota Quando per il confronto si utilizzano i dati dell'orologio interno della CPU, impostare S₁ su A351 per specificare i dati dell'orologio interno della CPU (A351 - A353).

Da S₂ a S₂+2: Dati temporali di confronto

Da S₂ a S₂+2 vi sono i dati temporali di confronto. Da S₂ a S₂+2 devono essere nella stessa area dati.



Nota Il valore dell'anno indica le ultime due cifre dell'anno. I valori da 00 a 97 vengono interpretati come da 2000 a 2097. I valori da 98 a 99 vengono interpretati come da 1998 a 1999.

Caratteristiche operando

| Area | C | S ₁ | S ₂ |
|--------------------------------------|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6141 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | Da W000 a W509 | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H509 | Da H000 a H510 |
| Area bit ausiliaria | Da A448 a A959 | Da A000 ad A957 | Da A000 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4093 | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4093 | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32765 | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32765 | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32765 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |

| Area | C | S ₁ | S ₂ |
|--|--|------------------------------|----------------|
| Costanti | Vedere la pagina precedente. | Vedere la pagina precedente. | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

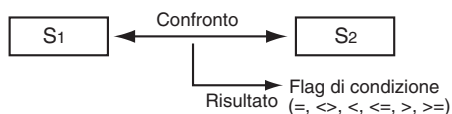
L'istruzione di confronto di date e ore confronta valori non mascherati (bit corrispondente di C impostato su 0) dei dati del tempo attuale in S₁ - S₁+2 con i dati temporali di confronto in S₂ - S₂+2 e crea una condizione di esecuzione ON quando la condizione di confronto risulta vera. Contemporaneamente, il risultato di un'istruzione di confronto di date e ore si riflette nei flag aritmetici (=, <>, <, <=, >, >=).

Esistono 18 combinazioni possibili di istruzioni di confronto di date e ore.

I valori temporali mascherati nel canale di controllo (C) non vengono inclusi nel confronto.

Nella tabella seguente viene riportato lo stato ON/OFF di ciascun flag per ogni risultato di confronto.

| Risultato | Stato del flag | | | | | |
|---------------------------------|----------------|-----|-----|-----|-----|-----|
| | = | <> | < | <= | > | >= |
| S ₁ = S ₂ | ON | OFF | OFF | ON | OFF | ON |
| S ₁ > S ₂ | OFF | ON | OFF | OFF | ON | ON |
| S ₁ < S ₂ | OFF | ON | ON | ON | OFF | OFF |

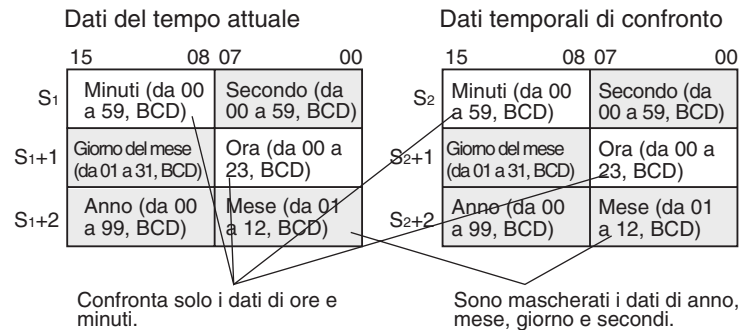


Mascheramento dei valori temporali

È possibile mascherare singoli valori temporali ed escluderli dall'operazione di confronto. Per mascherare un valore temporale, impostare su 1 il bit corrispondente nel canale di controllo (C). I bit da 00 a 05 di C mascherano rispettivamente i secondi, i minuti, le ore, il giorno, il mese e l'anno.

Esempio:

Quando C = 39 esadecimale, i 6 bit più a destra sono 111001 (anno=1, mese=1, giorno=1, ore=0, minuti=0 e secondi=1), quindi verranno confrontate solo le ore e minuti. Questa impostazione di mascheramento può essere utilizzata per eseguire una particolare operazione ogni giorno a un orario stabilito (ora e minuti).



Dati confrontati nelle precedenti istruzioni di confronto di date e ore in unità a 16 bit. Le istruzioni di confronto di date e ore si limitano a confrontare i valori temporali a 8 bit.

Nella tabella seguente viene illustrata la struttura dell'area interna dell'orologio calendario della CPU.

| Indirizzi | Contenuto |
|---------------------|-----------------------------------|
| Da A35100 ad A35107 | Secondo (da 00 a 59, BCD) |
| Da A35108 ad A35115 | Minuti (da 00 a 59, BCD) |
| Da A35200 ad A35207 | Ora (da 00 a 23, BCD) |
| Da A35208 ad A35215 | Giorno del mese (da 01 a 31, BCD) |
| Da A35300 ad A35307 | Mese (da 01 a 12, BCD) |
| Da A35308 ad A35315 | Anno (da 00 a 99, BCD) |

È possibile impostare l'area dell'orologio calendario tramite un dispositivo di programmazione (inclusa una console di programmazione), l'istruzione DATE(735) o il comando "CLOCK WRITE" FINS (0702 esadecimale).

Sommario delle istruzioni di confronto di date e ore

Nella tabella riportata di seguito sono indicati i codici funzione, i mnemonici e le funzioni delle 18 istruzioni di confronto di date e ore.

| Codice | Mnemonico | Nome | Funzione |
|--------|-----------|----------------------------|-----------------|
| 341 | LD =DT | LOAD EQUAL | Vera se S1 = S2 |
| | AND=DT | AND EQUAL | |
| | OR=DT | OR EQUAL | |
| 342 | LD <>DT | LOAD NOT EQUAL | Vera se S1 ≠ S2 |
| | AND <> DT | AND NOT EQUAL | |
| | OR <>DT | OR NOT EQUAL | |
| 343 | LD <DT | LOAD LESS THAN | Vera se S1 < S2 |
| | AND <DT | AND LESS THAN | |
| | OR <DT | OR LESS THAN | |
| 344 | LD <=DT | LOAD LESS THAN OR EQUAL | Vera se S1 ≤ S2 |
| | AND <=DT | AND LESS THAN OR EQUAL | |
| | OR <=DT | OR LESS THAN OR EQUAL | |
| 345 | LD >DT | LOAD GREATER THAN | Vera se S1 > S2 |
| | AND >DT | AND GREATER THAN | |
| | OR >DT | OR GREATER THAN | |
| 346 | LD >=DT | LOAD GREATER THAN OR EQUAL | Vera se S1 ≥ S2 |
| | AND >=DT | AND GREATER THAN OR EQUAL | |
| | OR >=DT | OR GREATER THAN OR EQUAL | |

Flag

| Nome | Etichetta | Operazione |
|---------------------------|-----------|---|
| Flag di errore | ER | ON se tutti i 6 bit di mascheramento (bit di C da 00 a 05) sono ON. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se $S_1 > S_2$. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | > = | ON se $S_1 \geq S_2$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se $S_1 = S_2$. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | ≠ | ON se $S_1 \neq S_2$. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se $S_1 < S_2$. OFF in tutti gli altri casi. |
| Flag di minore o uguale | < = | ON se $S_1 \leq S_2$. OFF in tutti gli altri casi. |
| Flag negativo | N | OFF o invariato (vedere nota). |

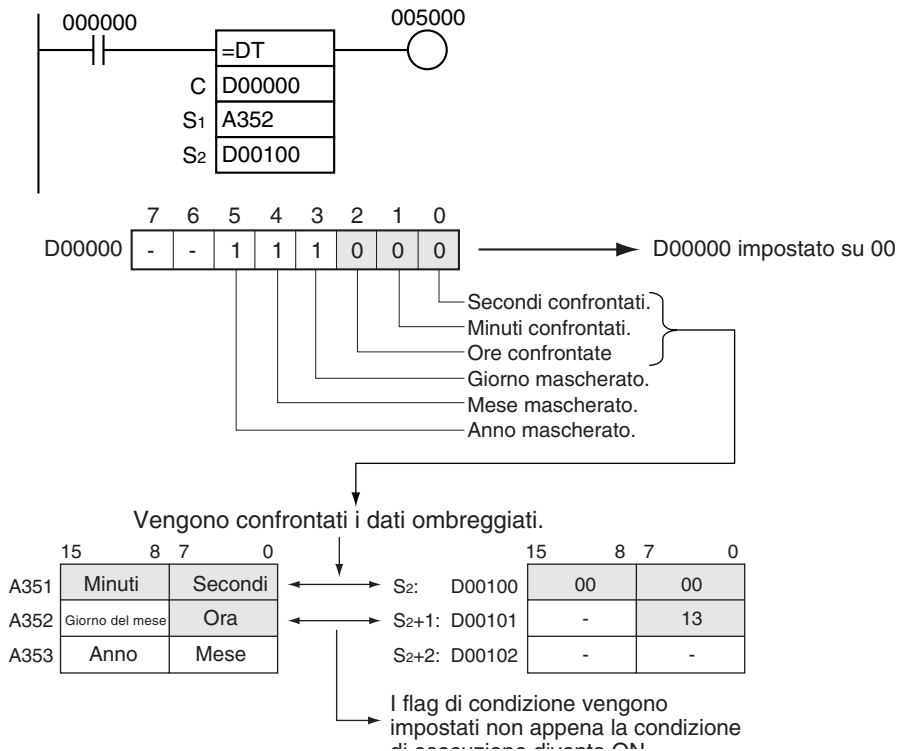
Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

Non è possibile utilizzare le istruzioni di confronto di date e ore come istruzioni sul lato destro: tra queste istruzioni e la barra di distribuzione destra è infatti necessario specificare un'altra istruzione.

Esempio

Quando CIO 000000 è ON e l'ora è 13:00:00, CIO 005000 viene attivato. I contenuti di A351 - A353 (i dati dell'orologio calendario interni della CPU) vengono utilizzati come dati del tempo attuale e i contenuti di D00100 - D00102 come dati temporali di confronto. I valori di anno, mese e giorno sono mascherati, quindi verranno confrontati soltanto i dati relativi a ora, minuti e secondi.

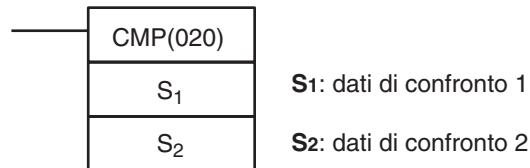


3-7-3 COMPARE: CMP(020)

Scopo

Confronta due valori binari senza segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | CMP(020) |
|--|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota). | | !CMP(020) |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D per sistemi a due CPU.

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

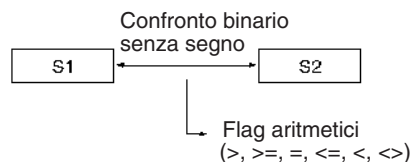
Caratteristiche operando

| Area | S ₁ | S ₂ |
|--------------------------------------|--|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | |
| Registri dati | Da DR0 a DR15 | |

| Area | S ₁ | S ₂ |
|--|--|----------------|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

CMP(020) confronta i dati binari senza segno in S₁ e S₂ e invia il risultato ai flag aritmetici (flag di maggiore, maggiore o uguale, uguale, minore o uguale, minore e di non uguaglianza) nell'area ausiliaria.

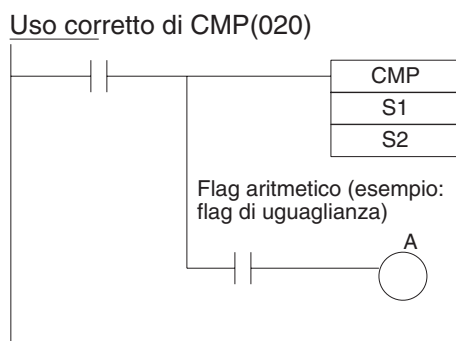
**Stato del flag di condizione**

Nella tabella seguente viene mostrato lo stato dei flag aritmetici dopo l'esecuzione di CMP(020). Lo stato "---" indica che il flag può essere ON o OFF.

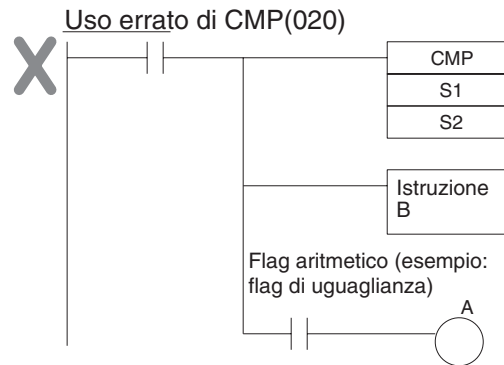
| Risultato di CMP(020) | Stato del flag | | | | | |
|---------------------------------|----------------|-----|-----|-----|-----|-----|
| | > | > = | = | < = | < | < > |
| S ₁ > S ₂ | ON | ON | OFF | OFF | OFF | ON |
| S ₁ = S ₂ | OFF | ON | ON | ON | OFF | OFF |
| S ₁ < S ₂ | OFF | OFF | OFF | ON | ON | ON |

Utilizzo dei risultati di CMP(020) nel programma

Quando viene eseguita l'istruzione CMP(020), il risultato si riflette sui flag aritmetici. Controllare l'uscita desiderata o l'istruzione sul lato destro con una diramazione dalla stessa condizione di ingresso che controlla CMP(020), come mostrato nello schema riportato di seguito. In questo caso, il flag di uguaglianza e l'uscita A vengono attivati quando S₁ = S₂.

**Utilizzo dei risultati di CMP(020) nel programma**

Non programmare un'altra istruzione tra CMP(020) e l'istruzione controllata dal flag aritmetico, poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico. In questo caso, i risultati dell'istruzione B potrebbero modificare i risultati di CMP(020).



La variazione di aggiornamento immediato (!CMP(020)) può essere utilizzata con canali allocati a ingressi esterni specificati in S_1 e/o in S_2 . Quando viene eseguita l'istruzione !CMP(020), viene eseguito l'aggiornamento del canale di ingresso esterno specificato in S_1 e/o in S_2 e viene confrontato il valore aggiornato. Non è possibile eseguire l'aggiornamento immediato sugli ingressi assegnati ai moduli di I/O ad alta densità gruppo 2 o a moduli montati su rack di sistemi slave.

Flag

| Nome | Etichetta CX-Programmer | Etichetta console di programmazione | Operazione |
|---------------------------|-------------------------|-------------------------------------|--|
| Flag di errore | P_ER | ER | OFF o invariato (vedere nota). |
| Flag di maggiore | P_GT | > | ON se $S_1 > S_2$. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | P_GE | > = | ON se $S_1 \geq S_2$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | P_EQ | = | ON se $S_1 = S_2$. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | P_NE | ≠ | ON se $S_1 \neq S_2$. OFF in tutti gli altri casi. |
| Flag di minore | P_LT | < | ON se $S_1 < S_2$. OFF in tutti gli altri casi. |
| Flag di minore o uguale | P_LE | < = | ON se $S_1 \leq S_2$. OFF in tutti gli altri casi. |
| Flag negativo | P_N | N | OFF o invariato (vedere nota). |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1/H, CJ1/H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

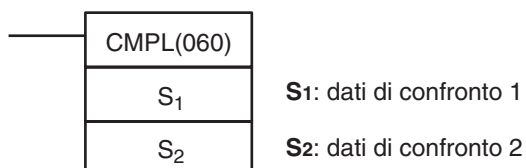
Non programmare un'altra istruzione tra CMP(020) e la condizione di ingresso che accede al risultato di CMP(020), poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico.

3-7-4 DOUBLE COMPARE: CMPL(060)

Scopo

Confronta due valori binari doppi senza segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | CMPL(060) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

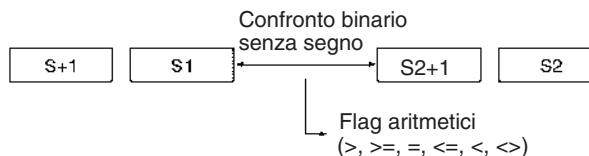
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S ₁ | S ₂ |
|--|---|----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | |
| Registri dati | --- | |
| Registri indice | Da IR0 a IR15 | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

CMPL(060) confronta i dati binari senza segno in $S_1 + 1$, S_1 e $S_2 + 1$, S_2 e invia il risultato ai flag aritmetici (flag di maggiore, maggiore o uguale, uguale, minore o uguale, minore e di non uguaglianza) nell'area ausiliaria.

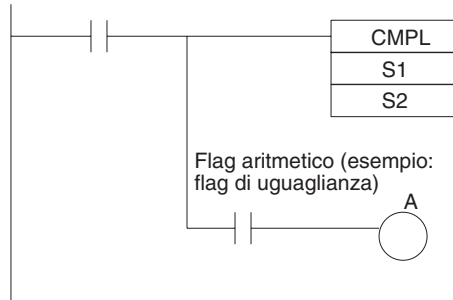
**Stato del flag aritmetico**

Nella tabella seguente viene mostrato lo stato dei flag aritmetici dopo l'esecuzione di CMPL(060). Lo stato "---" indica che il flag può essere ON o OFF.

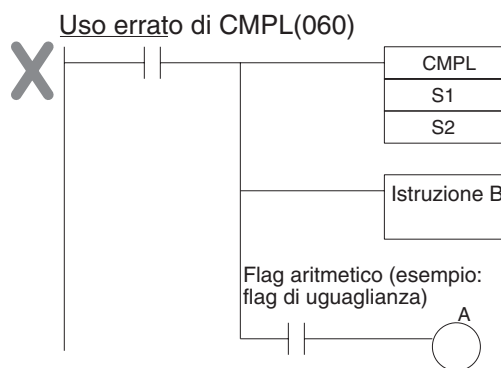
| Risultato di CMPL(060) | Stato del flag | | | | | |
|-------------------------------|----------------|-----|-----|-----|-----|-----|
| | > | > = | = | < = | < | < > |
| $S_1 + 1, S_1 > S_2 + 1, S_2$ | ON | ON | OFF | OFF | OFF | ON |
| $S_1 + 1, S_1 = S_2 + 1, S_2$ | OFF | ON | ON | ON | OFF | OFF |
| $S_1 + 1, S_1 < S_2 + 1, S_2$ | OFF | OFF | OFF | ON | ON | ON |

Utilizzo dei risultati di CMPL(060) nel programma

Quando viene eseguita l'istruzione CMPL(060), il risultato si riflette sui flag aritmetici. Controllare l'uscita desiderata o l'istruzione sul lato destro con una diramazione dalla stessa condizione di ingresso che controlla CMPL(060), come mostrato nello schema riportato di seguito. In questo caso, il flag di uguaglianza e l'uscita A vengono attivati quando $S_1 + 1, S_1 = S_2 + 1, S_2$.

Uso corretto di CMPL(060)**Utilizzo dei risultati di CMPL(060) nel programma**

Non programmare un'altra istruzione tra CMPL(060) e l'istruzione controllata dal flag aritmetico, poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico. In questo caso, i risultati dell'istruzione B potrebbero modificare i risultati di CMPL(060).



Flag

| Nome | Etichetta CX-Programmer | Etichetta console di programmazione | Operazione |
|---------------------------|-------------------------|-------------------------------------|--|
| Flag di errore | P_ER | ER | OFF o invariato (vedere nota). |
| Flag di maggiore | P_GT | > | ON se $S_1 + 1, S_1 > S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | P_GE | > = | ON se $S_1 + 1, S_1 \geq S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | P_EQ | = | ON se $S_1 + 1, S_1 = S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | P_NE | <> | ON se $S_1 + 1, S_1 \neq S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di minore | P_LT | < | ON se $S_1 + 1, S_1 < S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di minore o uguale | P_LE | < = | ON se $S_1 + 1, S_1 \leq S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag negativo | P_N | N | OFF o invariato (vedere nota). |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.

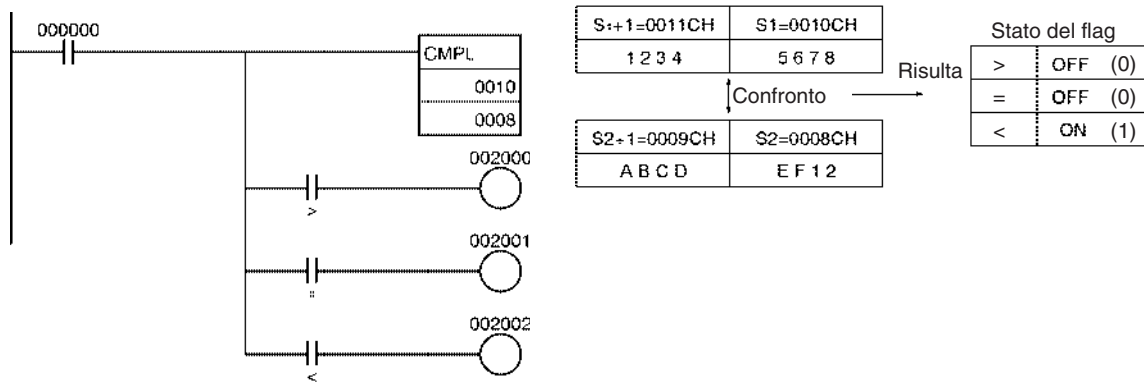
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

Non programmare un'altra istruzione tra CMPL(060) e la condizione di ingresso che accede al risultato di CMPL(060), poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico.

Esempio

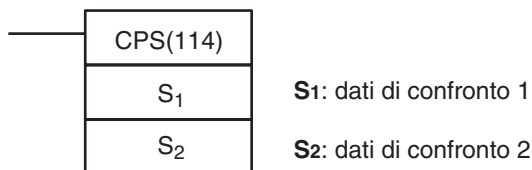
Nell'esempio seguente, quando CIO 000000 è impostato su ON, i dati binari a otto cifre senza segno in CIO 0011 e CIO 0010 vengono confrontati con i dati binari a otto cifre senza segno in CIO 0009 e CIO 0008 e il risultato viene inviato ai flag aritmetici. I risultati registrati nei flag Maggiore di, Uguale e Minore di vengono salvati immediatamente in CIO 000200 (Maggiore di), CIO 000201 (Uguale) e CIO 000202 (Minore di).



3-7-5 SIGNED BINARY COMPARE: CPS(114)

Scopo Confronta due valori binari con segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.

Simbolo programmazione ladder



Variazioni

| | | |
|---|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CPS(114) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota). | | !CPS(114) |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

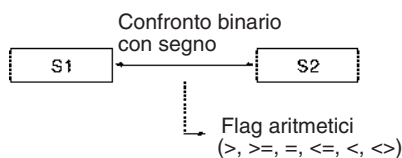
Caratteristiche operando

| Area | S ₁ | S ₂ |
|--------------------------------------|--|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |

| Area | S ₁ | S ₂ |
|--|--|----------------|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

CPS(114) confronta i dati binari con segno in S₁ e S₂ e invia il risultato ai flag aritmetici (flag di maggiore, maggiore o uguale, uguale, minore o uguale, minore e di non uguaglianza) nell'area ausiliaria.



Nota CPS(114) considera i dati in S₁ e S₂ come dati binari con segno compresi nell'intervallo da 8000 a 7FFF (da -32.768 a 32.767 decimale).

Stato del flag aritmetico

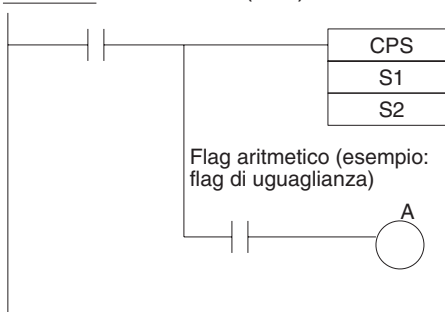
Nella tabella seguente viene mostrato lo stato dei flag aritmetici dopo l'esecuzione di CPS(114). Lo stato "---" indica che il flag può essere ON o OFF.

| Risultato di CPS(114) | Stato del flag | | | | | |
|---------------------------------|----------------|-----|-----|-----|-----|-----|
| | > | > = | = | < = | < | < > |
| S ₁ > S ₂ | ON | ON | OFF | OFF | OFF | ON |
| S ₁ = S ₂ | OFF | ON | ON | ON | OFF | OFF |
| S ₁ < S ₂ | OFF | OFF | OFF | ON | ON | ON |

Utilizzo dei risultati di CPS(114) nel programma

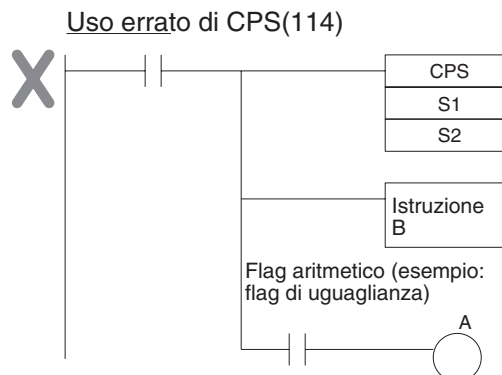
Quando viene eseguita l'istruzione CPS(114), il risultato si riflette sui flag aritmetici. Controllare l'uscita desiderata o l'istruzione sul lato destro con una diramazione dalla stessa condizione di ingresso che controlla CPS(114), come mostrato nello schema riportato di seguito. In questo caso, il flag di uguaglianza e l'uscita A vengono attivati quando S₁ = S₂.

Uso corretto di CPS(114)



Utilizzo dei risultati di CPS(114) nel programma

Non programmare un'altra istruzione tra CPS(114) e l'istruzione controllata dal flag aritmetico, poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico. In questo caso, i risultati dell'istruzione B potrebbero modificare i risultati di CPS(114).



La variazione di aggiornamento immediato (!CPS(114)) può essere utilizzata con canali allocati a ingressi esterni specificati in S_1 e/o in S_2 . Quando viene eseguita l'istruzione !CPS(114), viene eseguito l'aggiornamento del canale di ingresso esterno specificato in S_1 e/o in S_2 e viene confrontato il valore aggiornato. Non è possibile eseguire l'aggiornamento immediato sugli ingressi assegnati ai moduli di I/O ad alta densità gruppo 2 o a moduli montati su rack di sistemi slave.

Flag

| Nome | Etichetta | Operazione |
|---------------------------|-----------|--|
| Flag di errore | ER | OFF o invariato (vedere nota). |
| Flag di maggiore | > | ON se $S_1 > S_2$. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | > = | ON se $S_1 \geq S_2$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se $S_1 = S_2$. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | <> | ON se $S_1 \neq S_2$. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se $S_1 < S_2$. OFF in tutti gli altri casi. |
| Flag di minore o uguale | < = | ON se $S_1 \leq S_2$. OFF in tutti gli altri casi. |
| Flag negativo | N | OFF o invariato (vedere nota). |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

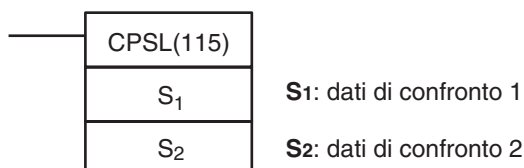
Avvertenze

Non programmare un'altra istruzione tra CPS(114) e la condizione di ingresso che accede al risultato di CPS(114), poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico.

3-7-6 DOUBLE SIGNED BINARY COMPARE: CPSL(115)

Scopo Confronta due valori binari doppi con segno (costanti e/o il contenuto di canali specifici) e invia il risultato ai flag aritmetici dell'area ausiliaria.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | CPSL(115) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

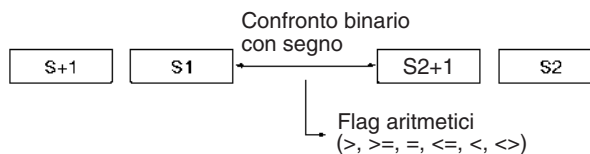
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S ₁ | S ₂ |
|--|---|----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

CPSL(115) confronta i dati binari doppi con segno in $S_1 + 1$, S_1 e $S_2 + 1$, S_2 e invia il risultato ai flag aritmetici (flag di maggiore, maggiore o uguale, uguale, minore o uguale, minore e di non uguaglianza) nell'area ausiliaria.



Nota CPSL(115) considera i dati in S_1 e S_2 come dati binari con segno compresi nell'intervallo da 8000 a 7FFF (da $-2.147.483.648$ a $2.147.483.647$ decimale).

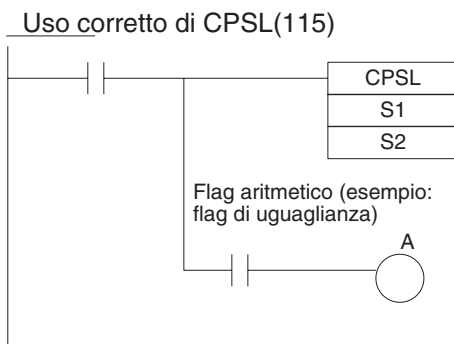
Stato del flag aritmetico

Nella tabella seguente viene mostrato lo stato dei flag aritmetici dopo l'esecuzione di CPSL(115). Lo stato "---" indica che il flag può essere ON o OFF.

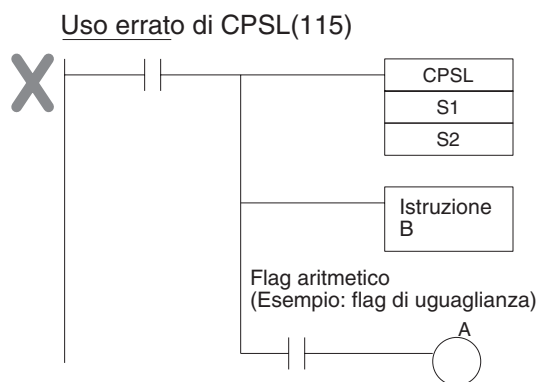
| Risultato di CPSL(115) | Stato del flag | | | | | |
|-------------------------------|----------------|-----|-----|-----|-----|-----|
| | > | >= | = | <= | < | <> |
| $S_1 + 1, S_1 > S_2 + 1, S_2$ | ON | ON | OFF | OFF | OFF | ON |
| $S_1 + 1, S_1 = S_2 + 1, S_2$ | OFF | ON | ON | ON | OFF | OFF |
| $S_1 + 1, S_1 < S_2 + 1, S_2$ | OFF | OFF | OFF | ON | ON | ON |

Utilizzo dei risultati di CPSL(115) nel programma

Quando viene eseguita l'istruzione CPSL(115), il risultato si riflette sui flag aritmetici. Controllare l'uscita desiderata o l'istruzione sul lato destro con una diramazione dalla stessa condizione di ingresso che controlla CPSL(115), come mostrato nello schema riportato di seguito. In questo caso, il flag di uguaglianza e l'uscita A vengono attivati quando $S_1 + 1, S_1 = S_2 + 1, S_2$.

**Utilizzo dei risultati di CPSL(115) nel programma**

Non programmare un'altra istruzione tra CPSL(115) e l'istruzione controllata dal flag aritmetico, poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico. In questo caso, i risultati dell'istruzione B potrebbero modificare i risultati di CPSL(115).



Flag

| Nome | Etichetta | Operazione |
|---------------------------|-----------|--|
| Flag di errore | ER | OFF o invariato (vedere nota). |
| Flag di maggiore | > | ON se $S_1 + 1, S_1 > S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | > = | ON se $S_1 + 1, S_1 \geq S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se $S_1 + 1, S_1 = S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | ≠ | ON se $S_1 + 1, S_1 \neq S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se $S_1 + 1, S_1 < S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag di minore o uguale | < = | ON se $S_1 + 1, S_1 \leq S_2 + 1, S_2$. OFF in tutti gli altri casi. |
| Flag negativo | N | OFF o invariato (vedere nota). |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

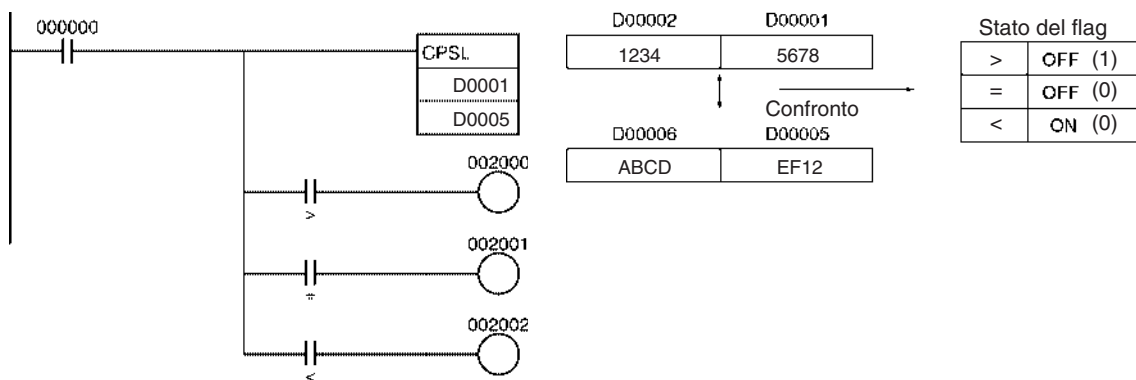
Avvertenze

Non programmare un'altra istruzione tra CPSL(115) e la condizione di ingresso che accede al risultato di CPSL(115), poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico.

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i dati binari a otto cifre con segno in D00002 e D00001 vengono confrontati con i dati binari a otto cifre con segno in D00006 e D00005 e il risultato viene inviato ai flag aritmetici.

- Se il contenuto di D00002 e D00001 è maggiore di quello di D00006 e D00005, verrà attivato il flag Maggiore di e CIO 002000 verrà impostato su ON.
- Se il contenuto di D00002 e D00001 è uguale a quello di D00006 e D00005, verrà attivato il flag Uguale a e CIO 002001 verrà impostato su ON.
- Se il contenuto di D00002 e D00001 è minore di quello di D00006 e D00005, verrà attivato il flag Minore di e CIO 002002 verrà impostato su ON.

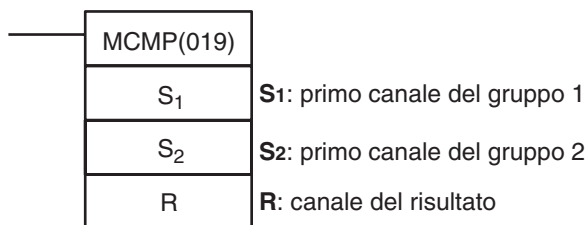


3-7-7 MULTIPLE COMPARE: MCMP(019)

Scopo

Confronta 16 canali consecutivi con altri 16 canali consecutivi e, nel caso in cui il contenuto di un canale **non** è uguale, imposta su ON il bit corrispondente nel canale del risultato.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MCMP(019) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MCMP(019) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S₁: primo canale del gruppo 1

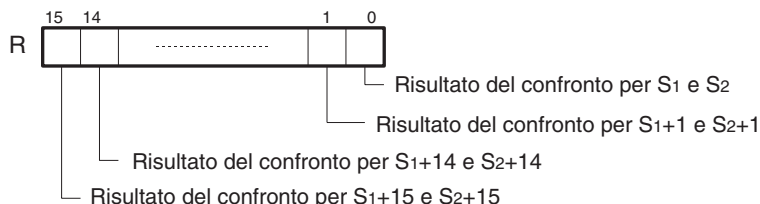
Specifica l'inizio del primo intervallo di 16 canali. S₁ e S₁+15 devono essere nella stessa area dati.

S₂: primo canale del gruppo 2

Specifica l'inizio del secondo intervallo di 16 canali. S₂ e S₂+15 devono essere nella stessa area dati.

R: canale del risultato

Ogni bit di R contiene il risultato di un confronto tra due canali nei gruppi di 16 canali. Il bit n di R (n = 00 - 15) contiene il risultato del confronto tra i canali S₁+n e S₂+n.



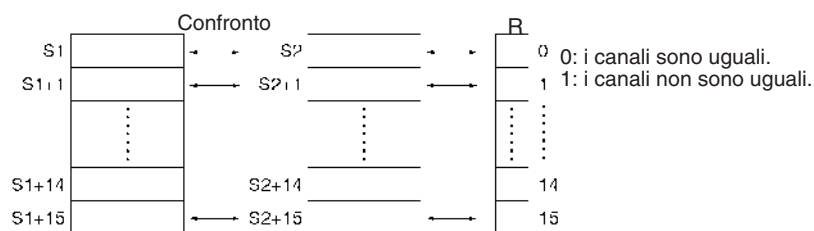
Caratteristiche operando

| Area | S ₁ | S ₂ | R |
|--|--|----------------|--|
| Area CIO | Da CIO 0000 a CIO 6128 | | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W496 | | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H496 | | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A944 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4080 | | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4080 | | Da C0000 a C4095 |
| Area DM | Da D00000 a D32752 | | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32752 | | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a 32752 (n = da 0 a C) | | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

MCMP(019) confronta i contenuti dei 16 canali da S₁ a S₁+15 con i contenuti dei 16 canali da S₂ a S₂+15 e imposta su ON il bit corrispondente nel canale R quando i contenuti **non sono** uguali.

Il contenuto di S₁ viene confrontato con il contenuto di S₂, il contenuto di S₁+1 con il contenuto di S₂+1, ..., e il contenuto di S₁+15 con il contenuto di S₂+15. Il bit n di R viene impostato su OFF se il contenuto di S₁+n è uguale al contenuto di S₂+n, mentre viene impostato su ON se i contenuti non sono uguali. Se i contenuti di tutte le 16 coppie di canali sono identici, dopo l'esecuzione dell'istruzione verrà attivato il flag di uguaglianza.

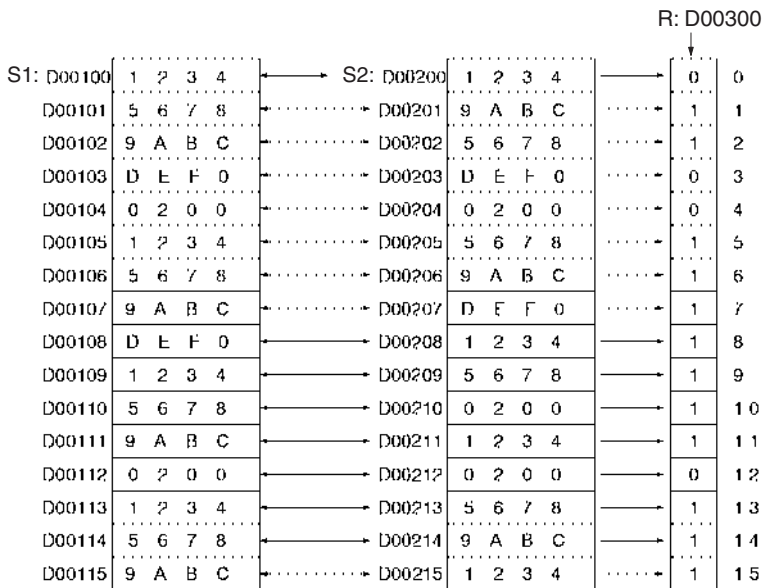
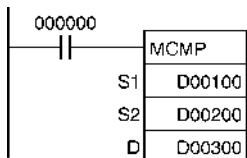


Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il canale del risultato è 0000. (I due gruppi di 16 canali contengono gli stessi dati). OFF in tutti gli altri casi. |

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, MCMP(019) confronta i canali da D00100 a D00115 con i canali da D00200 a D00215 e attiva i bit corrispondenti in D00300 quando i canali **non sono** uguali.

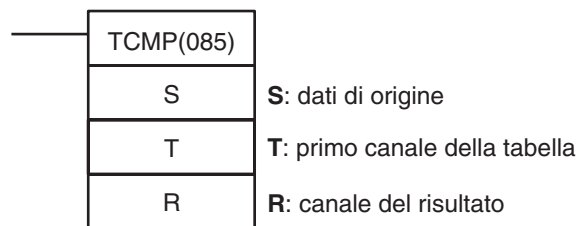


3-7-8 TABLE COMPARE: TCMP(085)

Scopo

Confronta i dati di origine con i contenuti dei 16 canali consecutivi e, se i contenuti dei canali **sono** uguali, imposta su ON il bit corrispondente nel canale del risultato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TCMP(085) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TCMP(085) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

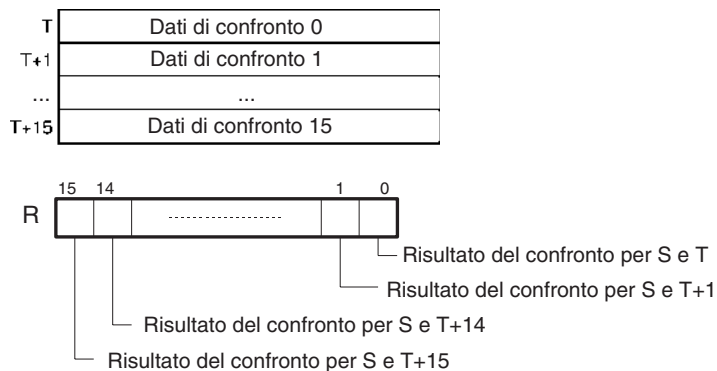
Operandi

T: primo canale della tabella

Specifica l'inizio della tabella di 16 canali. T e T+15 devono essere nella stessa area dati.

R: canale del risultato

Ogni bit di R contiene il risultato di un confronto tra S e un canale nella tabella di 16 canali. Il bit n di R (n = 00 - 15) contiene il risultato del confronto tra S e T+n.



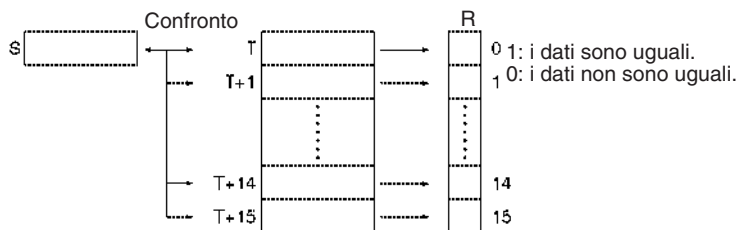
Caratteristiche operando

| Area | S | T | R |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6128 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W496 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H496 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 a A944 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4080 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4080 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32752 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32752 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32752 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @E00000 a @E32767 Da @En_00000 a @En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

TCMP(085) confronta i dati di origine (S) con ciascuno dei 16 canali da T a T+15 e attiva il bit corrispondente nel canale R quando i dati **sono** uguali. Il bit n di R viene impostato su ON se il contenuto di T+n è uguale a S e viene impostato su OFF se i contenuti non sono uguali.

S viene confrontato con il contenuto di T e il bit 00 di R viene impostato su ON se i contenuti sono uguali o su OFF se non lo sono, S viene confrontato con il contenuto di T+1 e il bit 01 di R viene impostato su ON se i contenuti sono uguali o su OFF se non lo sono, ..., e S viene confrontato con il contenuto di T+15 e il bit 15 di R viene impostato su ON se i contenuti sono uguali o su OFF se non lo sono.

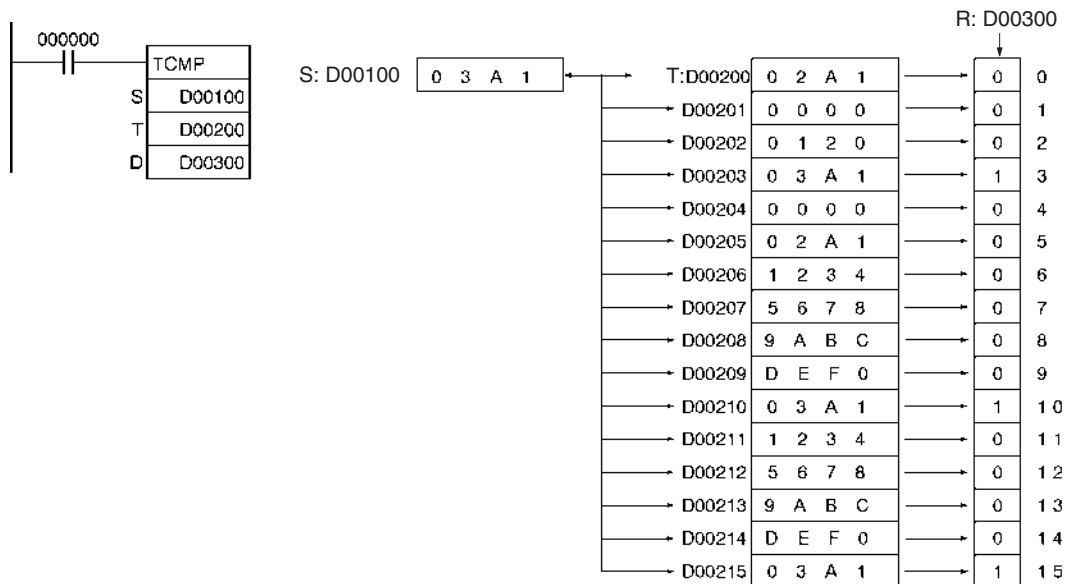


Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il canale del risultato è 0000. (Nessuno dei 16 canali nella tabella è uguale a S.) OFF in tutti gli altri casi. |

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, TCMP(085) confronta il contenuto di D00100 con il contenuto dei canali da D00200 a D00215 e attiva i bit corrispondenti in D00300 quando i contenuti sono uguali o li disattiva se risultano diversi.

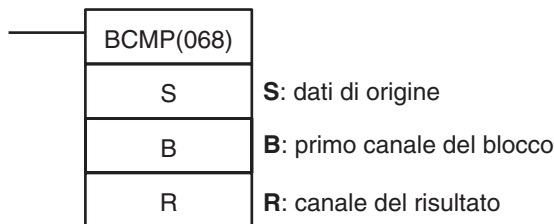


3-7-9 BLOCK COMPARE: BCMP(068)

Scopo

Confronta i dati di origine con 16 intervalli (definiti da 16 limiti inferiori e 16 limiti superiori) e, se i dati di origine rientrano nell'intervallo, imposta su ON il bit corrispondente nel canale del risultato.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BCMP(068) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BCMP(068) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

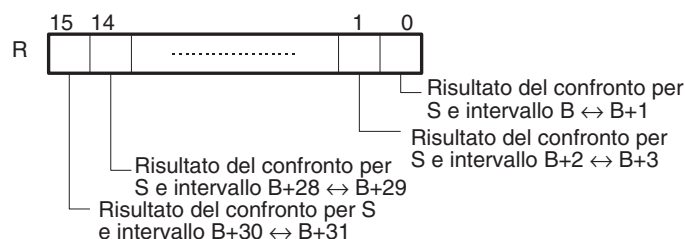
Operandi

B: Primo canale di blocco

Specifica l'inizio del blocco di 32 canali (16 coppie con limite inferiore/superiore). B e B+31 devono essere nella stessa area dati.

R: canale del risultato

Ogni bit di R contiene il risultato di un confronto tra S e uno dei 16 intervalli definiti nel blocco di 32 canali. Il bit n di R (n = 00 - 15) contiene il risultato del confronto tra S e l'ⁿesima coppia di canali.



Caratteristiche operando

| Area | S | B | R |
|-------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6112 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W0000 a W480 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H480 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A928 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4064 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4064 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32736 | Da D00000 a D32767 |

| Area | S | B | R |
|--|--|--|--|
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32736 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32736 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

BCMP(068) confronta i dati di origine (S) con i 16 intervalli definiti dalle coppie di valori del limite inferiore e superiore in B - B+31. Il primo canale in ciascuna coppia (B+2n) fornisce il limite inferiore e il secondo canale (B+2n+1) fornisce il limite superiore dell'intervallo n (n = 0 - 15). Se S è incluso in uno di questi intervalli, compresi i limiti superiore e inferiore, il bit corrispondente in R verrà impostato su ON. I restanti bit in R verranno impostati su OFF.

| | | | |
|------|-------|------|-------------|
| B | ≤ S ≤ | B+1 | Bit 00 di R |
| B+2 | ≤ S ≤ | B+3 | Bit 01 di R |
| B+4 | ≤ S ≤ | B+5 | Bit 02 di R |
| B+6 | ≤ S ≤ | B+7 | Bit 03 di R |
| B+8 | ≤ S ≤ | B+9 | Bit 04 di R |
| B+10 | ≤ S ≤ | B+11 | Bit 05 di R |
| B+12 | ≤ S ≤ | B+13 | Bit 06 di R |
| B+14 | ≤ S ≤ | B+15 | Bit 07 di R |
| B+16 | ≤ S ≤ | B+17 | Bit 08 di R |
| B+18 | ≤ S ≤ | B+19 | Bit 09 di R |
| B+20 | ≤ S ≤ | B+21 | Bit 10 di R |
| B+22 | ≤ S ≤ | B+23 | Bit 11 di R |
| B+24 | ≤ S ≤ | B+25 | Bit 12 di R |
| B+26 | ≤ S ≤ | B+27 | Bit 13 di R |
| B+28 | ≤ S ≤ | B+29 | Bit 14 di R |
| B+30 | ≤ S ≤ | B+31 | Bit 15 di R |

Ad esempio, il bit 00 di R viene impostato su ON se S è compreso nel primo intervallo ($B \leq S \leq B+1$), il bit 01 di R viene impostato su ON se S è compreso nel secondo intervallo ($B+2 \leq S \leq B+3$), ..., e il bit 15 di R viene impostato su ON se S è compreso nel quindicesimo intervallo ($B+30 \leq S \leq B+31$). Tutti gli altri bit in R verranno impostati su OFF.

Flag

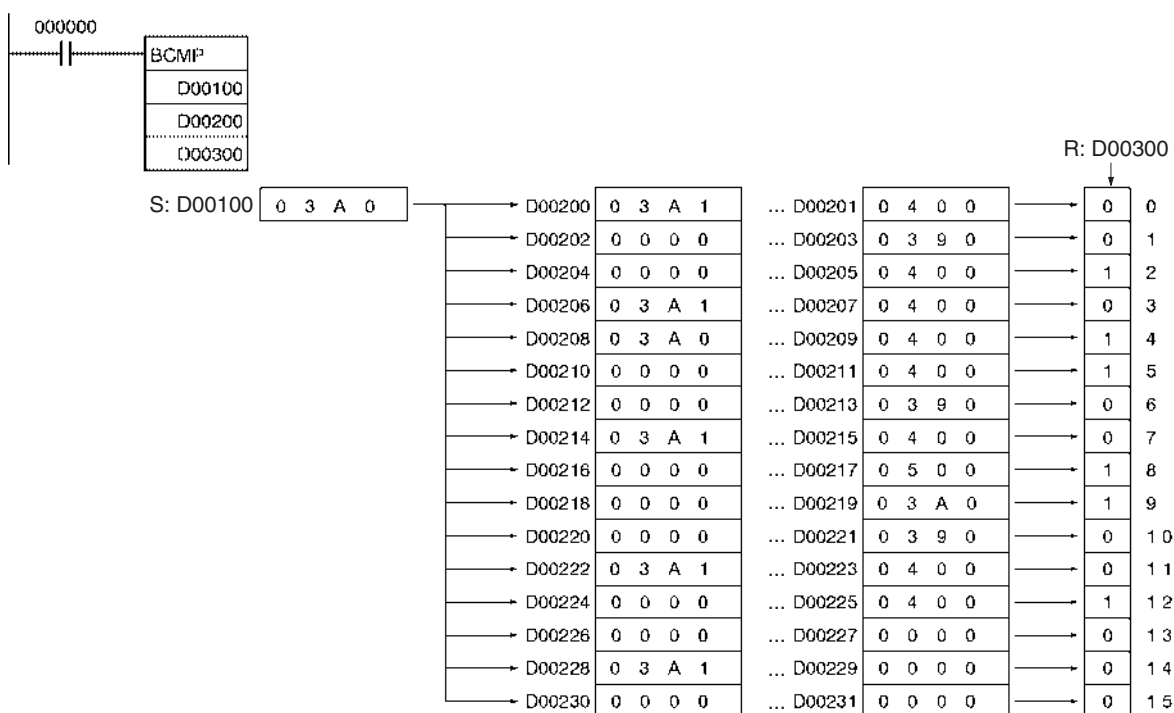
| Nome | Etichetta | Operazione |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il canale del risultato è 0000. (S non è compreso in nessuno dei 16 intervalli). OFF in tutti gli altri casi. |

Avvertenze

Se il limite inferiore è maggiore del limite superiore non verrà generato alcun errore, ma al bit corrispondente di R verrà inviato 0 (non incluso nell'intervallo).

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, BCMP(068) confronta il contenuto di D00100 con i 16 intervalli definiti in D00200 - D00231 e attiva i bit corrispondenti in D00300 quando S è incluso nell'intervallo o li disattiva quando S non è incluso nell'intervallo.

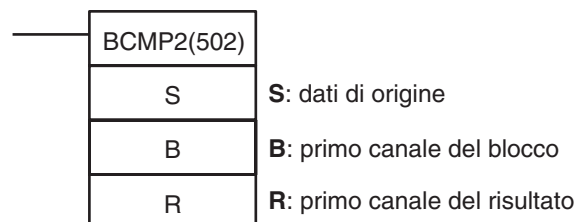


3-7-10 EXPANDED BLOCK COMPARE: BCMP2(502)

Scopo

Confronta i dati di origine con fino a 256 intervalli (definiti da 256 limiti inferiori e 256 limiti superiori) e, se i dati di origine rientrano nell'intervallo, imposta su ON il bit corrispondente nel canale del risultato. L'istruzione BCMP2(502) è supportata solo dalle CPU CS1-H, CJ1-H o CS1D versione 2.0 o successiva e dalle CPU CJ1M (precedenti alla versione 2.0, versione 2.0 o successiva).

Simbolo programmazione ladder



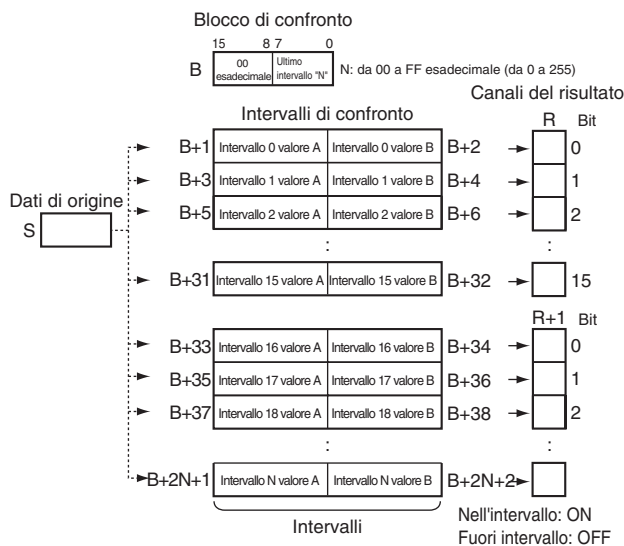
Caratteristiche operando

| Area | S | B | R |
|--|--|-----|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | --- | | |
| Area EM con banco | --- | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

BCMP2(502) confronta i dati di origine (S) con gli intervalli definiti dalle coppie di valori del limite inferiore e superiore nel blocco di confronto. Se S è incluso in uno di questi intervalli, compresi i limiti superiore e inferiore, i bit corrispondenti nei canali del risultato (da R a R+15 max) verranno impostati su ON. I restanti bit in R verranno impostati su OFF.

Il numero di intervalli viene determinato dal valore N impostato nel byte inferiore di B. N può essere compreso tra 0 e 255. Il byte superiore di B deve essere 00 esadecimale.

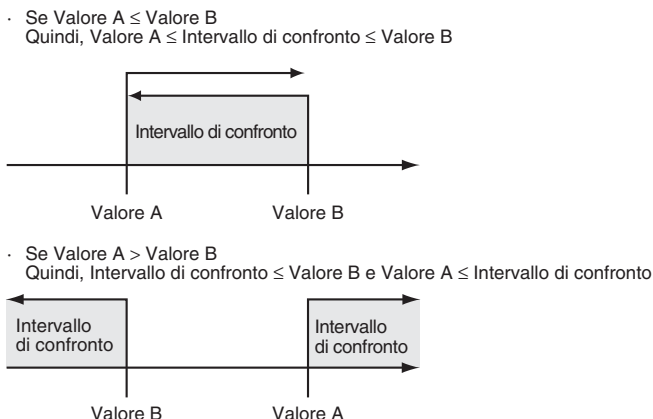


Numero di intervalli

Il numero di intervalli del blocco di confronto viene impostato nel primo canale del blocco. È possibile impostare fino a 256 intervalli.

Impostazione degli intervalli

I valori A e B di ciascun intervallo determinano il modo in cui il confronto viene eseguito, in base al valore maggiore, come mostrato di seguito.



Esempio

Quando $B+1 \leq B+2$

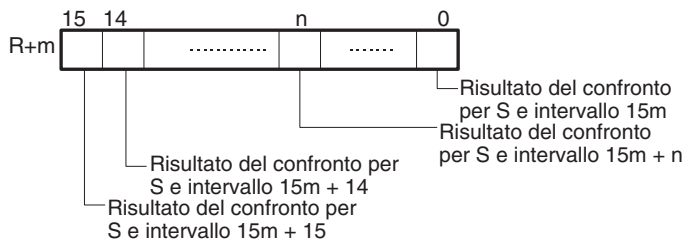
- Se $B+1 \leq S \leq B+2$, il bit 0 di R verrà impostato su ON,
- Se $B+3 \leq S \leq B+4$, il bit 1 di R verrà impostato su ON,
- Se $S < B+5$ e $B+6 < S$, il bit 2 di R verrà impostato su OFF e
- Se $S < B+7$ e $B+8 < S$, il bit 3 di R verrà impostato su OFF.

Quando $B+1 > B+2$

- Se $S \leq B+2$ e $B+1 \leq S$, il bit 0 di R verrà impostato su ON,
- Se $S \leq B+4$ e $B+3 \leq S$, il bit 1 di R verrà impostato su ON,
- Se $B+6 < S < B+5$, il bit 2 di R verrà impostato su OFF e
- Se $B+8 < S < B+7$, il bit 3 di R verrà impostato su OFF.

Area di memorizzazione dei risultati

I risultati vengono inviati ai bit corrispondenti nel canale R. Se vi sono più di 16 intervalli di confronto, verranno utilizzati i canali consecutivi successivi a R. Il numero massimo di canali del risultato è 16, ossia m è uguale a 0 - 15.



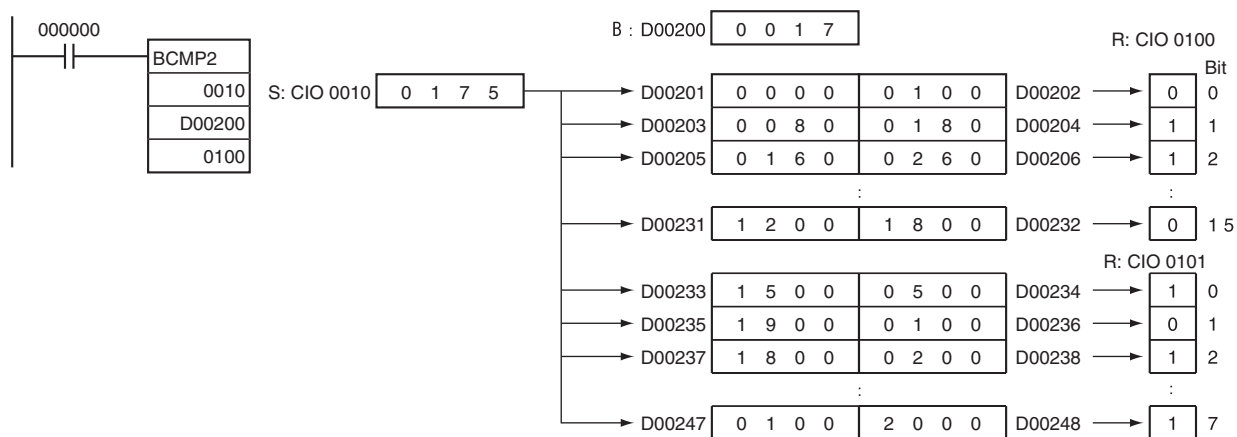
Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|------------|
| Flag di errore | ER | OFF |

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, BCMP2(502) confronta il contenuto di CIO 0010 con i 24 intervalli definiti da D00200 a D00247 (N = 17 esadecimale = 23 decimale, ossia 24 intervalli) e attiva i bit corrispondenti in CIO 0100 e CIO 0101 quando S è incluso nell'intervallo e li disattiva quando S non è incluso nell'intervallo. Ad esempio, se i dati di origine in CIO 0010 sono compresi nell'intervallo definito da D00201 e da D00202, allora il bit 00 di CIO 0100 verrà impostato su ON, mentre se non sono compresi nell'intervallo il bit 00 di CIO 0100 verrà impostato su OFF. In modo simile, i dati di origine in CIO 0010 vengono

confrontati con gli intervalli definiti da D00203 e da D00204, da D00247 e da D00248 e dagli altri canali nel blocco di confronto, il bit 1 in CIO 0100, il bit 7 in CIO 1010 e gli altri bit nei canali del risultato verranno manipolati in base ai risultati del confronto.



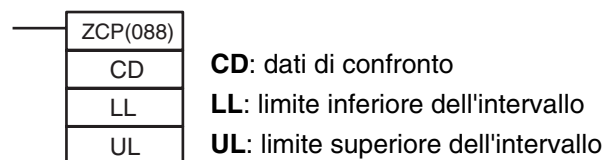
3-7-11 AREA RANGE COMPARE: ZCP(088)

Scopo

Confronta un valore binario a 16 bit senza segno (CD) con l'intervallo definito dal limite inferiore LL e dal limite superiore UL. I risultati vengono inviati ai flag aritmetici.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ZCP(088) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| | | | |
|-------------------------|------------------------|-----------|-----------|
| Area | CD | LL | UL |
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |

| Area | CD | LL | UL |
|--|--|----|----|
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

ZCP(088) confronta i dati binari a 16 bit con segno in CD con l'intervallo definito da LL e UL e invia il risultato ai flag Maggiore di, Uguale a e Minore di nell'area ausiliaria. (I flag Minore di o uguale, Maggiore di o uguale e il flag di non uguaglianza restano invariati).

Stato del flag aritmetico

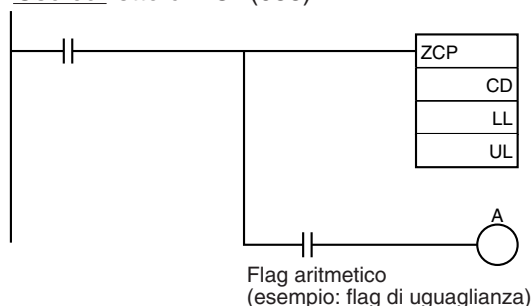
Nella tabella seguente viene mostrato lo stato dei flag aritmetici dopo l'esecuzione di ZCP(088).

| Risultato di ZCP(088) | Stato del flag | | |
|-----------------------|----------------|-----|-----|
| | > | = | < |
| CD > UL | ON | OFF | OFF |
| CD = UL | OFF | ON | |
| LL < CD < UL | | | |
| CD = LL | | | |
| CD < LL | | OFF | ON |

Utilizzo dei risultati di ZCP(088) nel programma

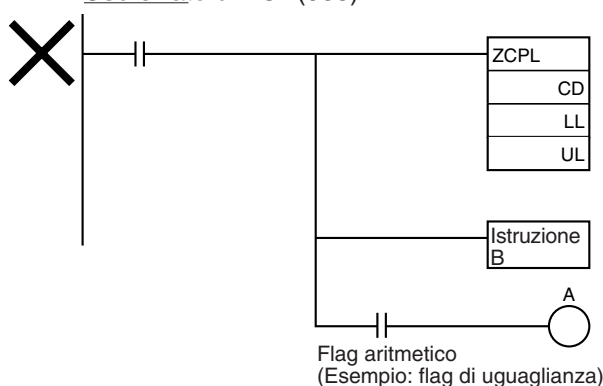
Quando viene eseguita l'istruzione ZCP(088), il risultato si riflette sui flag aritmetici. Controllare l'uscita desiderata o l'istruzione sul lato destro con una diramazione dalla stessa condizione di ingresso che controlla ZCP(088), come mostrato nello schema riportato di seguito. In questo caso, il flag di uguaglianza e l'uscita A vengono attivati quando $LL \leq CD \leq UL$.

Uso corretto di ZCP(088)



Non programmare un'altra istruzione tra ZCP(088) e l'istruzione controllata dal flag aritmetico, poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico. In questo caso, i risultati dell'istruzione B potrebbero modificare i risultati di ZCP(088).

Uso errato di ZCP(088)



Flag

| Nome | Etichetta | Operazione |
|---------------------------|-----------|---|
| Flag di errore | ER | ON se LL > UL. |
| Flag di maggiore | > | ON se CD > UL. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | > = | Rimane invariata. |
| Flag di uguaglianza | = | ON se LL ≤ CD ≤ UL. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | <> | Rimane invariata. |
| Flag di minore | < | ON se CD < LL. OFF in tutti gli altri casi. |
| Flag di minore o uguale | < = | Rimane invariata. |
| Flag negativo | N | Rimane invariata. |

Avvertenze

Non programmare un'altra istruzione tra ZCP(088) e la condizione di ingresso che accede al risultato di ZCP(088), poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico.

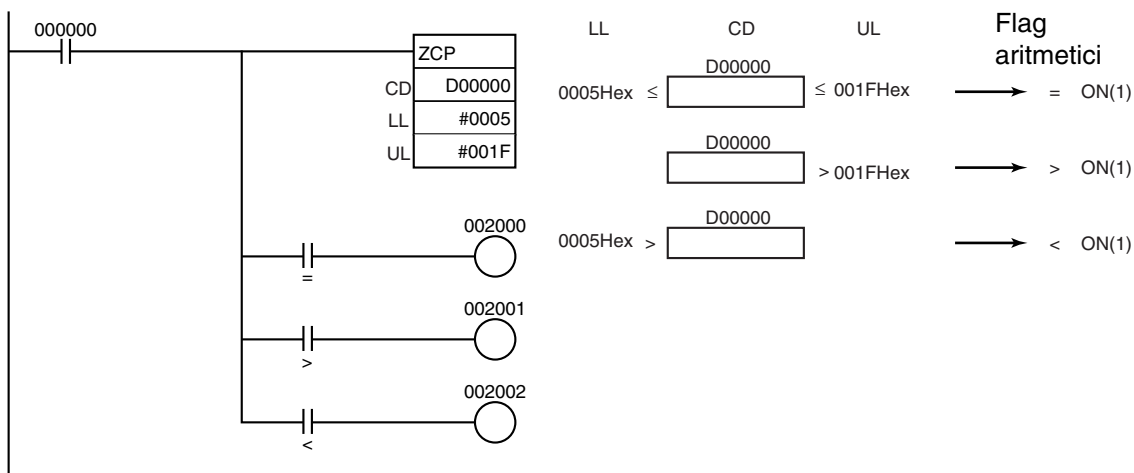
Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i dati binari a 16 bit senza segno in D00000 vengono confrontati con l'intervallo 0005 - 001F esadecimale (5 - 31 decimale) e il risultato viene inviato ai flag aritmetici.

CIO 000200 viene impostato su ON se 0005 esadecimale ≤ contenuto di D00000 ≤ 001F esadecimale.

CIO 000201 viene impostato su ON se il contenuto di D00000 > 001F esadecimale.

CIO 000202 viene impostato su ON se il contenuto di D00000 < 0005 esadecimale.



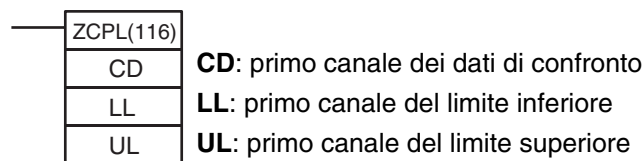
3-7-12 DOUBLE AREA RANGE COMPARE: ZCPL(116)

Scopo

Confronta un valore binario a 32 bit senza segno (CD+1, CD) con l'intervallo definito dal limite inferiore (LL+1, LL) e dal limite superiore (UL+1, UL). I risultati vengono inviati ai flag aritmetici.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ZCP(088) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | CD | LL | UL |
|-------------------------|--|----|----|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritenività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |

| Area | CD | LL | UL |
|--|--|----|----|
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 0000 a #FFFF FFFF (binario) | | |
| Registri dati | --- | | |
| Registri indice | Da IR0 a IR15 | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

ZCPL(116) confronta i dati binari a 32 bit con segno in CD+1, CD con l'intervallo definito da LL+1, LL e UL+1, UL e invia il risultato ai flag Maggiore di, Uguale a e Minore di nell'area ausiliaria. (I flag Minore di o uguale, Maggiore di o uguale e il flag di non uguaglianza restano invariati).

Stato del flag aritmetico

Nella tabella seguente viene mostrato lo stato dei flag aritmetici dopo l'esecuzione di ZCPL(116).

| Risultato di ZCPL(116) | Stato del flag | | |
|--------------------------------|----------------|-----|-----|
| | > | = | < |
| CD+1, CD > UL+1, UL | ON | OFF | OFF |
| CD+1, CD = UL+1, UL | OFF | ON | OFF |
| LL+1, LL < CD+1, CD < UL+1, UL | | | |
| CD+1, CD = LL+1, LL | | | |
| CD+1, CD < LL+1, LL | | OFF | ON |

Utilizzo dei risultati di ZCPL(116) nel programma

Quando viene eseguita l'istruzione ZCPL(116), il risultato si riflette sui flag aritmetici. Controllare l'uscita desiderata o l'istruzione sul lato destro con una diramazione dalla stessa condizione di ingresso che controlla ZCPL(116).

Non programmare un'altra istruzione tra ZCPL(116) e l'istruzione controllata dal flag aritmetico, poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico.

Il funzionamento di ZCPL(116) è quasi identico a quello della funzione ZCP(088), con la differenza che ZCPL(116) confronta valori a 32 bit anziché valori a 16 bit. Per gli schemi che mostrano l'utilizzo dei risultati nel programma e per una sezione di esempio del programma, consultare 3-7-11 AREA RANGE COMPARE: ZCP(088).

Flag

| Nome | Etichetta | Operazione |
|------------------|-----------|--|
| Flag di errore | ER | ON se LL+1, LL > UL+1, UL. |
| Flag di maggiore | > | ON se CD > UL+1, UL. OFF in tutti gli altri casi. |

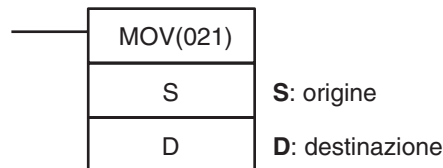
| Nome | Etichetta | Operazione |
|---------------------------|-----------|---|
| Flag di maggiore o uguale | > = | Rimane invariata. |
| Flag di uguaglianza | = | ON se LL+1, LL ≤ CD+1, CD ≤ UL+1, UL. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | <> | Rimane invariata. |
| Flag di minore | < | ON se CD+1, CD < LL+1, LL. OFF in tutti gli altri casi. |
| Flag di minore o uguale | < = | Rimane invariata. |
| Flag negativo | N | Rimane invariata. |

Avvertenze

Non programmare un'altra istruzione tra ZCPL(116) e la condizione di ingresso che accede al risultato di ZCPL(116), poiché l'altra istruzione potrebbe modificare lo stato del flag aritmetico.

3-8 Istruzioni di spostamento dati**3-8-1 MOVE: MOV(021)****Scopo**

Trasferisce un canale di dati nel canale specificato.

Simbolo programmazione ladder**Variazioni**

| | | |
|---|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MOV(021) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOV(021) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato (vedere nota). | | !MOV(021) |
| Variazioni combinate | Eseguita una sola volta con aggiornamento immediato della destinazione per differenziazione sul fronte di salita (vedere nota). | !@MOV(021) |

Nota L'aggiornamento immediato non è supportato dalle CPU CS1D.

Are di programma applicabili

| Are di programma a blocchi | Are di programma step | Subroutine | Task ad interrupt |
|----------------------------|-----------------------|------------|-------------------|
| OK | OK | OK | OK |

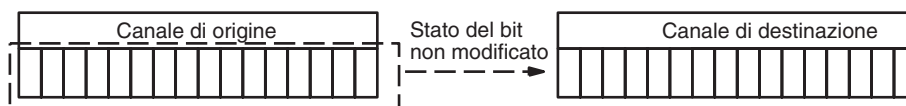
Caratteristiche operando

| Area | S | D |
|-------------------------|------------------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |

| Area | S | D |
|--|--|-----|
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

Trasferisce S a D. Se S è una costante, il valore può essere utilizzato per impostare i dati.



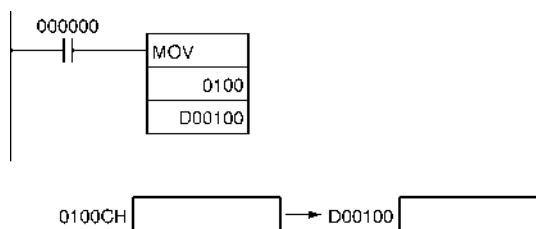
MOV(021) ha una variazione di aggiornamento immediato (!MOV(021)). È possibile specificare per S i bit di ingresso esterni e per D i bit di uscita esterni. I bit di ingresso utilizzati per S verranno aggiornati subito prima e i bit di uscita utilizzati per D verranno aggiornati subito dopo l'esecuzione, a meno che i bit siano allocati a un modulo di I/O ad alta densità gruppo 2, a un modulo di I/O speciale ad alta densità o a un modulo installato sul rack di un sistema slave di I/O remoto SYSMAC BUS.

Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se i dati da trasferire sono 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit più a sinistra dei dati da trasferire è 1. OFF in tutti gli altri casi. |

Esempio

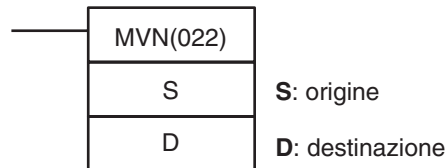
Nell'esempio seguente, quando CIO 000000 è impostato su ON, il contenuto di CIO 0100 viene copiato in D00100.



3-8-2 MOVE NOT: MVN(022)

Scopo Trasferisce il complemento di un canale di dati nel canale specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MVN(022) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MVN(022) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|--|--|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione MVN(022) inverte i bit in S e trasferisce il risultato in D. Il contenuto di S resta invariato.

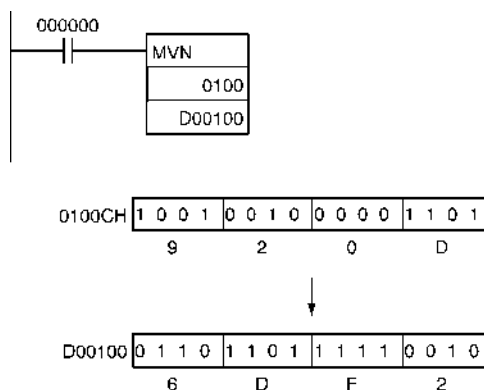


Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il contenuto di D è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il contenuto del bit più a sinistra di D è 1 dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempio

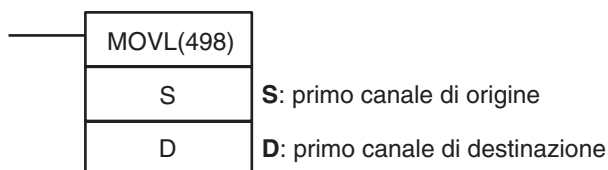
Nell'esempio seguente, quando CIO 000000 è impostato su ON, lo stato dei bit in CIO 0100 viene invertito e il risultato viene copiato in D00100.



3-8-3 DOUBLE MOVE: MOVL(498)

Scopo Trasferisce due canali di dati nei canali specificati.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | MOVL(498) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVL(498) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|--|---|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | Da IR0 a IR15 | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,1-(--)IR5 | |

Descrizione

MOVL(498) trasferisce S+1 e S a D+1 e D. Se S+1 e S sono costanti, il valore può essere utilizzato per impostare i dati.

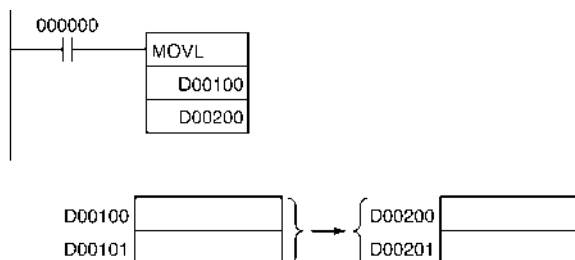


Flag

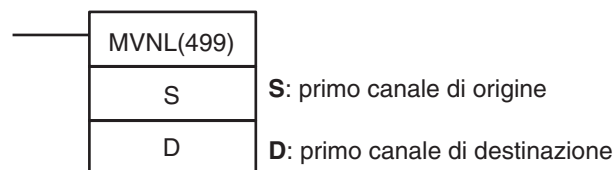
| Nome | Etichetta | Operazione |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il contenuto di D+1 e D è 0000 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il contenuto del bit più a sinistra di D+1 è 1 dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, il contenuto di D00101 e D00100 viene copiato in D00201 e D00200.

**3-8-4 DOUBLE MOVE NOT: MVNL(499)****Scopo**

Trasferisce il complemento di due canali di dati nei canali specificati.

Simbolo programmazione ladder**Variazioni**

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MVNL(499) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MVNL(499) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

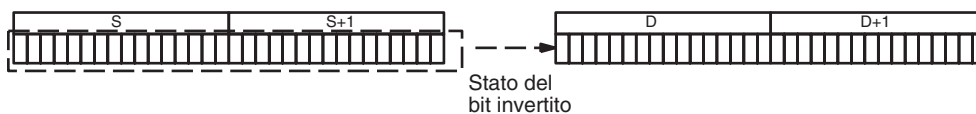
Caratteristiche operando

| Area | S | D |
|--------------------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |

| Area | S | D |
|--|--|-----|
| Costanti | Da #00000000 a #FFFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

MVNL(499) inverte i bit in S+1 e S e trasferisce il risultato in D+1 e D. Il contenuto di S+1 e S resta invariato.

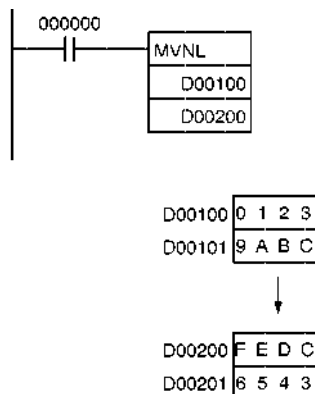


Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il contenuto di D+1 e D è 0000 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il contenuto del bit più a sinistra di D+1 è 1 dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, lo stato dei bit in D00101 e D00100 viene invertito e il risultato viene copiato in D00201 e D00200. (Il contenuto originale di D00101 e di D00100 resta invariato).

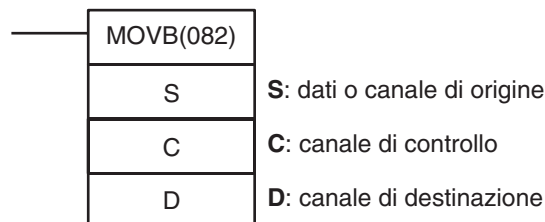


3-8-5 MOVE BIT: MOVB(082)

Scopo

Trasferisce il bit specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MOVB(082) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVB(082) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

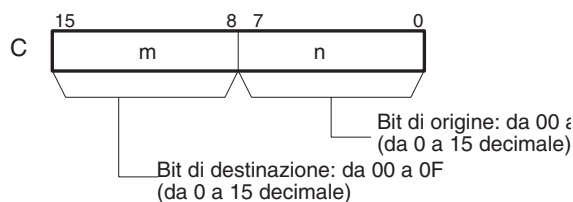
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo

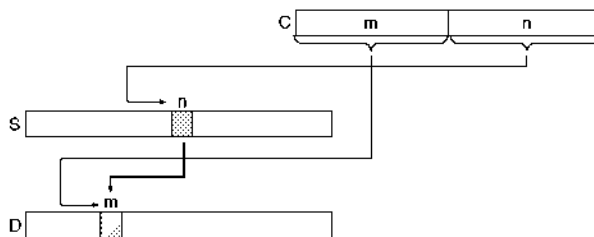
Le prime due cifre a destra di C indicano che il bit di S è il bit di origine, mentre le prime due cifre a sinistra di C indicano che il bit di D è il bit di destinazione.



Caratteristiche operando

| Area | S | C | D |
|--|--|-------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | Solo valori specificati | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione MOV(082) copia il bit specificato (n) da S al bit specificato (m) in D. Gli altri bit nel canale di destinazione vengono lasciati inalterati.



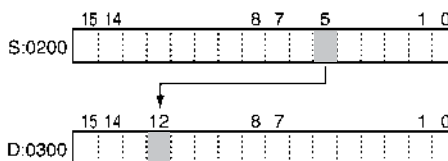
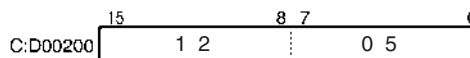
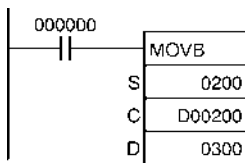
Nota Per copiare un bit in un canale, è possibile specificare lo stesso canale per S e D.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se le cifre più a destra e più a sinistra di C non sono nell'intervallo specificato da 00 a 0F. OFF in tutti gli altri casi. |

Esempi

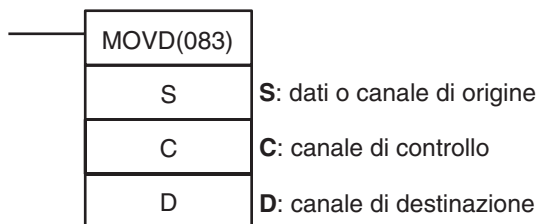
Nell'esempio seguente, quando CIO 000000 è ON, il 5° bit del canale di origine (CIO 0200) viene copiato nel 12° bit del canale di destinazione (CIO 0300) in base al valore 0C05 del canale di controllo.



3-8-6 MOVE DIGIT: MOVD(083)

Scopo Trasferisce la cifra o le cifre specificate (ogni cifra è composta da 4 bit).

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | MOVD(083) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVD(083) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

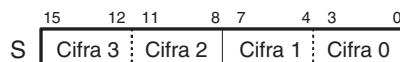
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

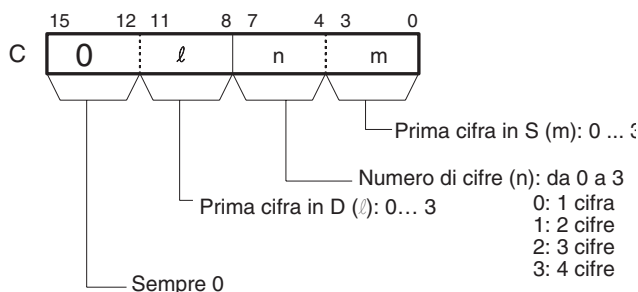
Operandi

S: canale di origine

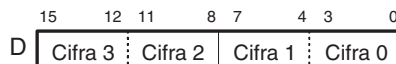
Le cifre di origine vengono lette da destra a sinistra e, se necessario, è possibile tornare alla cifra più a destra (cifra 0).

**C: canale di controllo**

Come illustrato nel diagramma seguente, le prime tre cifre di C indicano la prima cifra di origine (m), il numero di cifre da trasferire (n) e la prima cifra di destinazione (l).

**D: canale di destinazione**

Le cifre di destinazione vengono scritte da destra a sinistra e, se necessario, è possibile tornare alla cifra più a destra (cifra 0).



Caratteristiche operando

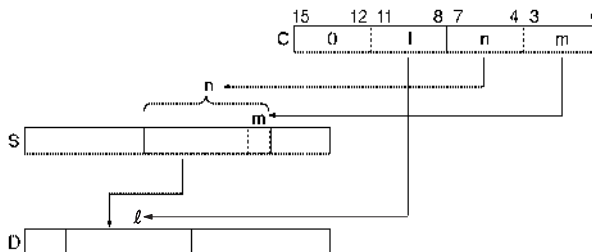
| Area | S | C | D |
|--------------------------------------|--|-------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | Solo valori specificati | --- |
| Registri dati | Da DR0 a DR15 | | |

| Area | S | C | D |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

MOVD(083) copia il contenuto di n cifre da S (iniziando dalla cifra m) a D (iniziando dalla cifra l). La modifica interviene soltanto sulle cifre specificate e le altre vengono lasciate inalterate.

Se il numero di cifre lette o scritte supera la cifra più a sinistra di S o di D, MOVD(083) torna alla cifra più a destra dello stesso canale.



Nota Per copiare un bit in un canale, è possibile specificare lo stesso canale per S e D.

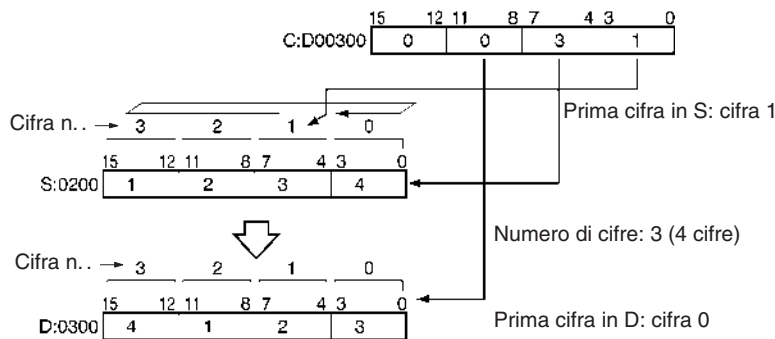
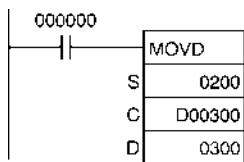
Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se una delle tre prime cifre di C non è nell'intervallo specificato compreso tra 0 e 3. OFF in tutti gli altri casi. |

Esempi

Trasferimento di quattro cifre

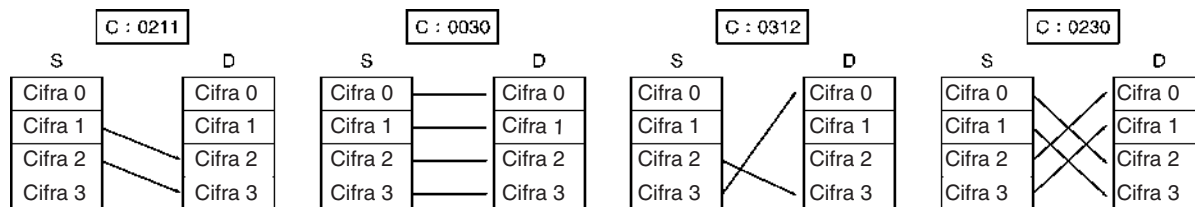
Nell'esempio che segue, quando CIO 000000 è ON, quattro cifre di dati vengono copiate da CIO 0200 a CIO 0300. Il trasferimento comincia dalla cifra 1 di CIO 0200 e dalla cifra 0 di CIO 0300, in base al valore 0031 del canale di controllo.



Nota Dopo avere letto la cifra più a sinistra di S (cifra 3), MOVD(083) torna alla cifra più a destra (cifra 0).

Esempi di C

Nel diagramma che segue sono riportati alcuni esempi di trasferimenti di dati per differenti valori di C.

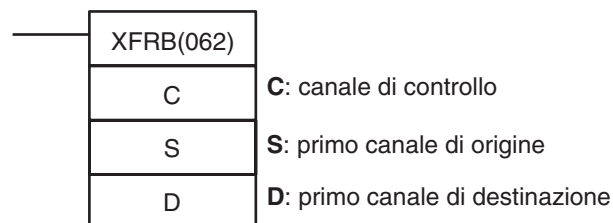


3-8-7 MULTIPLE BIT TRANSFER: XFRB(062)

Scopo

Trasferisce il numero di bit consecutivi specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | XFRB(062) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XFRB(062) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

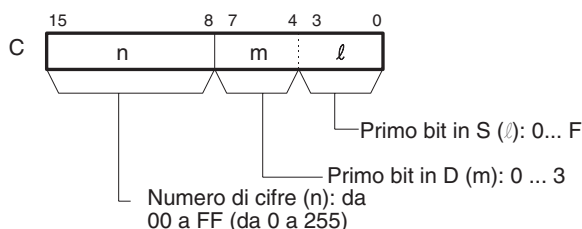
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

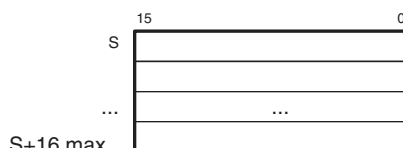
C: canale di controllo

Come illustrato nel diagramma seguente, le prime tre cifre di C indicano la prima cifra di origine (m), il numero di cifre da trasferire (n) e la prima cifra di destinazione (l).



S: primo canale di origine

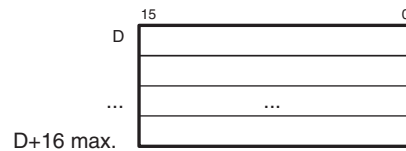
Specifica il primo canale di origine. I bit vengono letti da destra a sinistra e, se necessario, l'elaborazione continua con i canali consecutivi fino a S+16.



Nota I canali di origine devono essere nella stessa area dati.

D: primo canale di destinazione

Specifica il primo canale di destinazione. I bit vengono scritti da destra a sinistra e, se necessario, l'elaborazione continua con i canali consecutivi fino a D+16.



Nota I canali di destinazione devono essere nella stessa area dati.

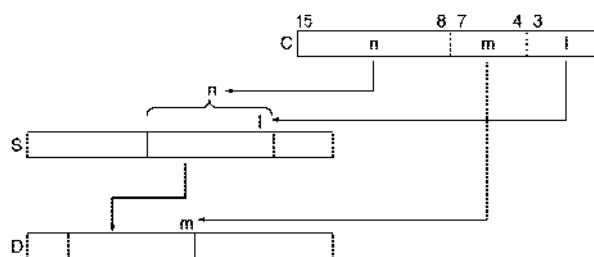
Caratteristiche operando

| Area | C | S | D |
|--|--|-----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | --- | --- |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a 5+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

XFRB(062) trasferisce fino a 255 bit consecutivi dai canali di origine (iniziando dal bit ℓ di S) ai canali di destinazione (iniziando dal bit m di D). I bit nei canali di destinazione che non sono sovrascritti dai bit di origine vengono lasciati inalterati.

Come illustrato nel seguente diagramma, i bit e i numeri di bit iniziali sono specificati in C.



È possibile sovrapporre i canali di origine e di destinazione. Nel trasferimento di dati che si sovrappongono in numerosi canali, nell'area dati è possibile comprimere i dati con maggiore efficacia. Questa operazione è particolarmente utile quando si gestiscono dati di posizione per il controllo della posizione.

Poiché i canali di origine e di destinazione possono sovrapporsi, è possibile combinare XFRB(062) con l'istruzione ANDW(034) al fine di spostare i bit m di n spazi.

Flag

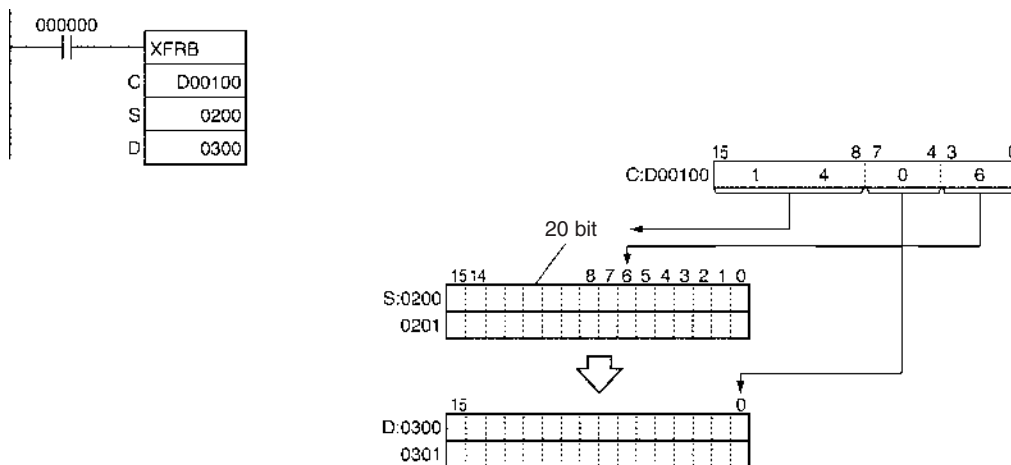
| Nome | Etichetta | Funzionamento |
|----------------|-----------|---------------|
| Flag di errore | ER | OFF |

Avvertenze

A ogni esecuzione di XFRB(062) è possibile trasferire fino a 255 bit di dati. Assicurarsi che i canali di origine e di destinazione non superino la fine dell'area dati.

Esempi

Nell'esempio che segue, quando CIO 000000 è ON, i 20 bit che iniziano con CIO 020006 vengono copiati nei 20 bit che iniziano con CIO 030000.

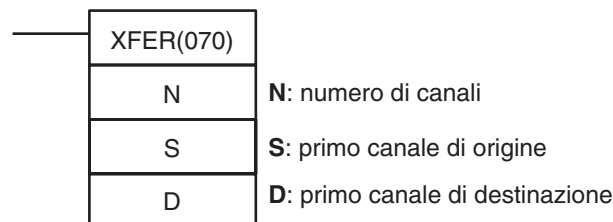


3-8-8 BLOCK TRANSFER: XFER(070)

Scopo

Trasferisce il numero di canali consecutivi specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | XFER(070) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XFER(070) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

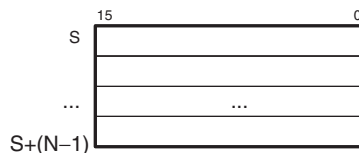
Operandi

N: numero di canali

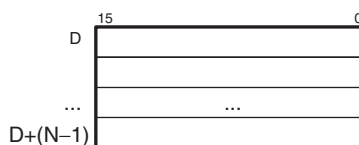
Specifica il numero di canali da trasferire. L'intervallo possibile per il valore di N è compreso tra 0000 ed FFFF (da 0 a 65.535 decimale).

S: primo canale di origine

Specifica il primo canale di origine.

**D: primo canale di destinazione**

Specifica il primo canale di destinazione.



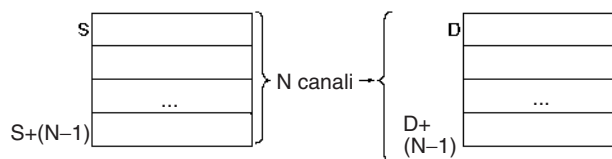
Caratteristiche operando

| Area | N | S | D |
|--------------------------------------|--|---|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |

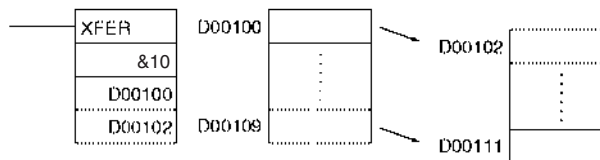
| Area | N | S | D |
|--|--|-----|-----|
| Costanti | Da #0000 a #FFFF (in formato binario) o da &0 a &65535 | --- | --- |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

XFER(070) copia N canali che iniziano con S (da S a S+(N-1)) negli N canali che iniziano con D (da D a D+(N-1)).



È possibile sovrapporre i canali di origine e di destinazione, per cui XFER(070) consente di eseguire operazioni di scorrimento del canale.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---------------|
| Flag di errore | ER | OFF |

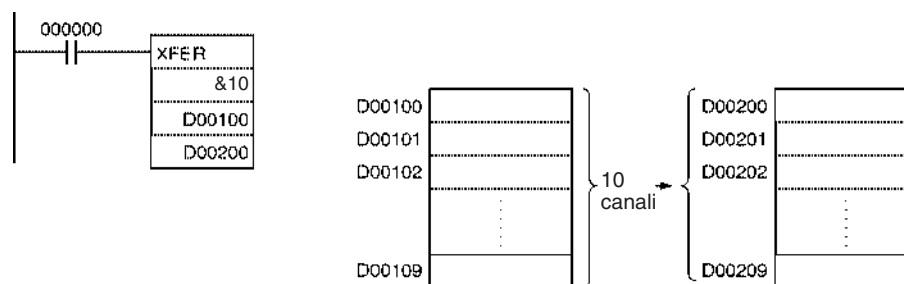
Avvertenze

Assicurarsi che i canali di origine (da S a S+N-1) e di destinazione (da D a D+N-1) non superino la fine dell'area dati.

Quando viene trasferito un numero elevato di canali, il completamento di XFER(070) può non essere rapido. In questo caso, se avviene una caduta di tensione durante l'esecuzione dell'istruzione, è possibile che il trasferimento di XFER(070) non venga completato.

Esempio

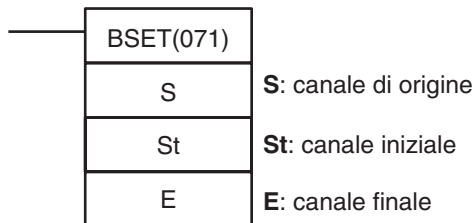
Quando CIO 000000 è ON, i 10 canali da D00100 a D00109 vengono copiati nell'intervallo compreso tra D00200 e D00209.



3-8-9 BLOCK SET: BSET(071)

Scopo Copia lo stesso canale in un intervallo di canali consecutivi.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BSET(071) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BSET(071) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: canale di origine

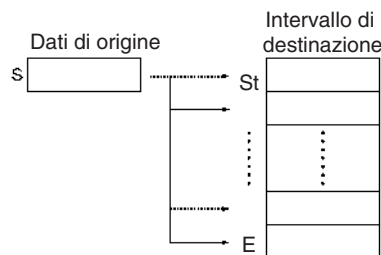
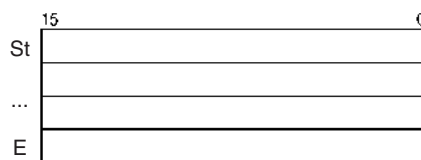
Specifica i dati di origine o il canale con i dati di origine.

St: canale iniziale

Specifica il primo canale nell'intervallo di destinazione.

E: canale finale

Specifica l'ultimo canale nell'intervallo di destinazione.



Nota St ed E devono essere nella stessa area dati.

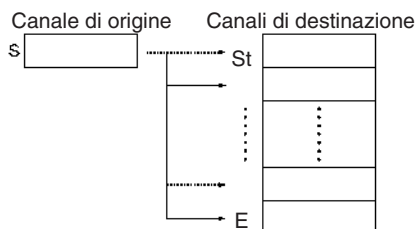
Caratteristiche operando

| Area | S | St | E |
|-------------------------|------------------------|-----------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |

| Area | S | St | E |
|--|--|-----|---|
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

BSET(071) copia lo stesso canale di origine (S) in tutti i canali di destinazione nell'intervallo da St a E.

**Flag**

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se St è maggiore di E. OFF in tutti gli altri casi. |

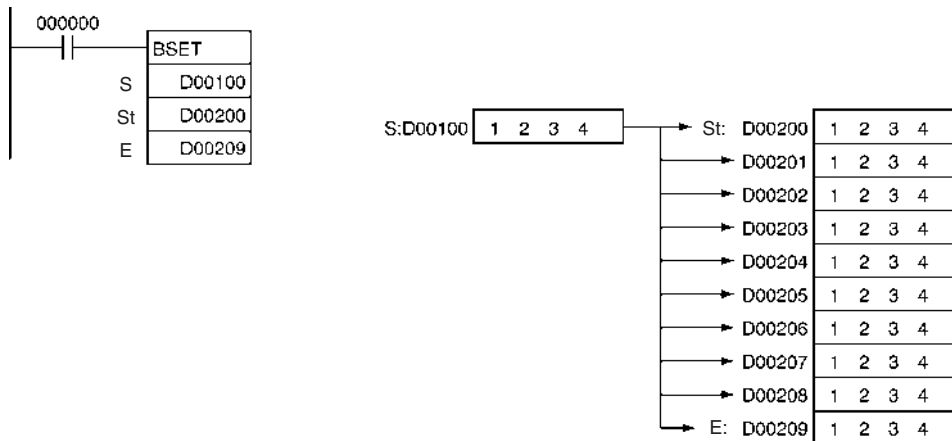
Avvertenze

Assicurarsi che il canale iniziale (St) e finale (E) siano nella stessa area dati e che $St \leq E$.

Quando i dati di origine vengono trasferiti in un numero elevato di canali, il completamento di BSET(071) può non essere rapido. In questo caso, se avviene una caduta di tensione durante l'esecuzione dell'istruzione, è possibile che il trasferimento di BSET(071) non venga completato.

Esempio

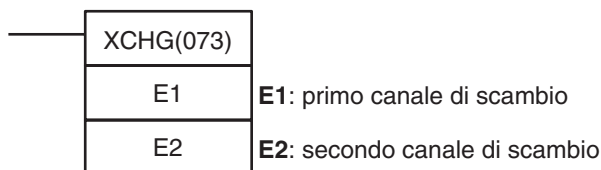
Quando CIO 000000 è ON, i dati di origine in D00100 vengono copiati nell'intervallo compreso tra D00200 e D00209.



3-8-10 DATA EXCHANGE: XCHG(073)

Scopo Scambia il contenuto dei due canali specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | XCHG(073) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XCHG(073) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | E1 | E2 |
|--------------------------------------|--|-----------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |

| Area | E1 | E2 |
|--|--|----|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

XCHG(073) scambia il contenuto di E1 ed E2.

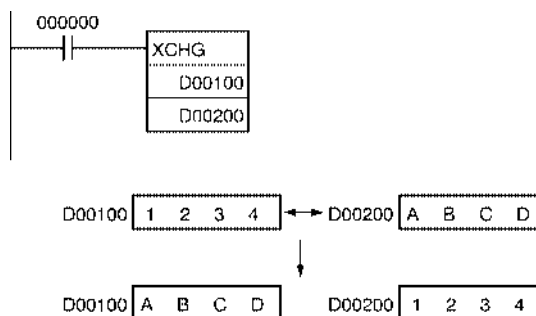
**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF o non modificato (vedere nota) |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

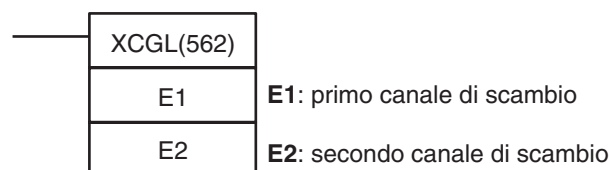
Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Esempio

Quando CIO 000000 è ON, il contenuto di D00100 viene scambiato con il contenuto di D00200.

**3-8-11 DOUBLE DATA EXCHANGE: XCGL(562)****Scopo**

Scambia il contenuto di una coppia di canali consecutivi con il contenuto di un'altra coppia di canali consecutivi.

Simbolo programmazione ladder

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | XCGL(562) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XCGL(562) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

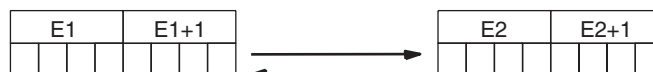
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

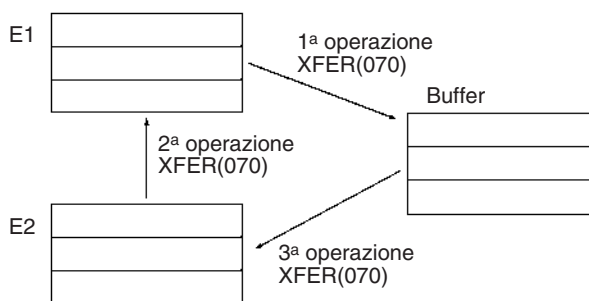
| Area | E1 | E2 |
|--|--|-----|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A448 ad A958 | |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | --- |
| Registri dati | --- | |
| Registri indice | Da IR0 a IR15 | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

XCHG(073) scambia il contenuto di E1+1 ed E1 con il contenuto di E2+1 ed E2.



Per scambiare 3 o più canali, utilizzare XFER(070) e trasferire i canali in una terza serie di canali (buffer) come illustrato nel seguente diagramma.



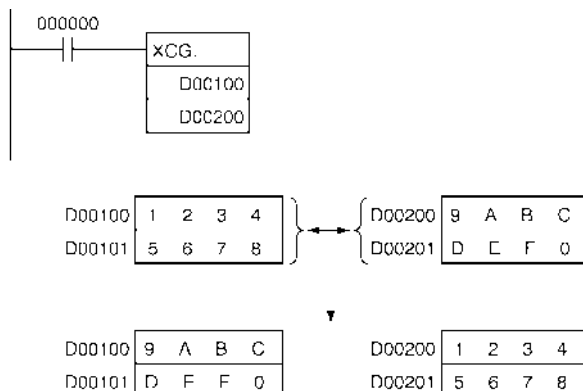
Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF o non modificato (vedere nota) |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF. Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Esempio

Quando CIO 000000 è ON, il contenuto di D00100 e D00101 viene scambiato con il contenuto di D00200 e D00201.

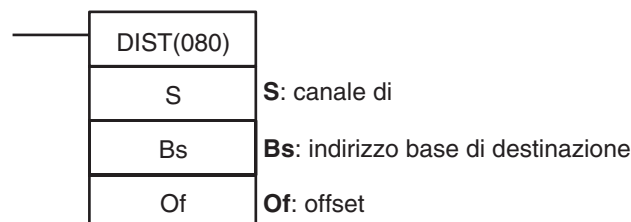


3-8-12 SINGLE WORD DISTRIBUTE: DIST(080)

Scopo

Trasferisce il canale di origine in un canale di destinazione calcolato aggiungendo il valore di offset all'indirizzo di base.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | DIST(080) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @DIST(080) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

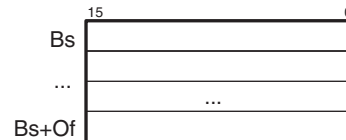
Operandi

Bs: indirizzo base di destinazione

Specifica l'indirizzo base di destinazione. L'offset viene aggiunto a questo indirizzo per calcolare il canale di destinazione.

Of: offset

Questo valore viene aggiunto all'indirizzo base per calcolare il canale di destinazione. L'offset può avere qualsiasi valore da 0000 a FFFF (da 0 a 65.535 decimale), ma Bs e Bs+Of devono essere nella stessa area dati.

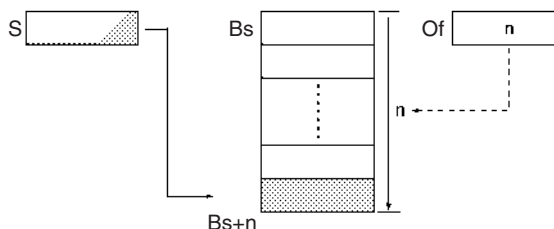


Caratteristiche operando

| Area | S | Bs | Of |
|--|--|-----------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | Da A000 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | Da #0000 a #FFFF (in formato binario) o da &0 a &65535 |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

DIST(080) copia S sul canale di destinazione calcolato aggiungendo Of a Bs. La stessa istruzione DIST(080) può essere utilizzata per distribuire il canale origine nell'area dei dati modificando il valore di Of.

**Flag**

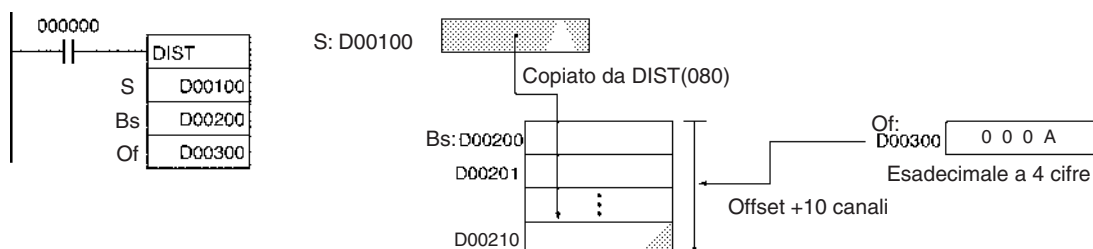
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se i dati di origine sono 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra dei dati di origine è 1. OFF in tutti gli altri casi. |

Avvertenze

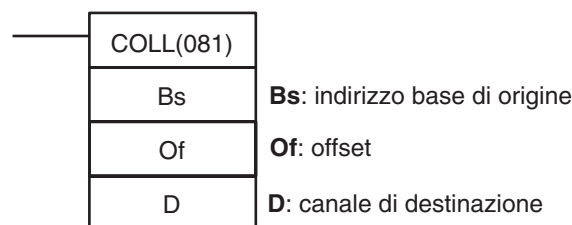
Assicurarsi che l'offset non superi la fine dell'area dati, ossia che Bs e Bs+Of siano nella stessa area dati.

Esempio

Quando CIO 000000 è ON, il contenuto di D00100 viene copiato in D00210 (D00200 + 10) purché il contenuto di D00300 sia 10 (0A esadecimale). Modificando l'offset in D00300, è possibile copiare il contenuto di D00100 in altri canali.

**3-8-13 DATA COLLECT: COLL(081)****Scopo**

Trasferisce il canale di origine, calcolato aggiungendo un valore di offset all'indirizzo di base, nel canale di destinazione.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | COLL(081) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @COLL(081) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

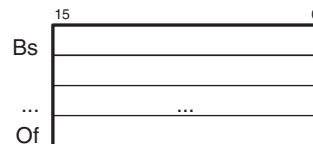
Operandi

Bs: indirizzo base di origine

Specifica l'indirizzo base di origine. L'offset viene aggiunto a questo indirizzo per calcolare il canale di origine.

Of: offset

Questo valore viene aggiunto all'indirizzo base per calcolare il canale di origine. L'offset può avere qualsiasi valore da 0000 a FFFF (da 0 a 65.535 decimale), ma Bs e Bs+Of devono essere nella stessa area dati.

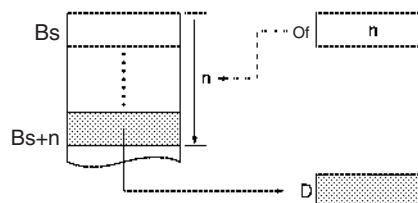


Caratteristiche operando

| Area | Bs | Of | D |
|--|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a #FFFF (in formato binario) o da &0 a &65535 | --- |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

COLL(081) copia il canale di origine (calcolato addizionando Of a Bs) nel canale di destinazione. La stessa istruzione COLL(081) può essere utilizzata per raccogliere i dati provenienti da diversi canali di origine nell'area dei dati modificando il valore di Of.

**Flag**

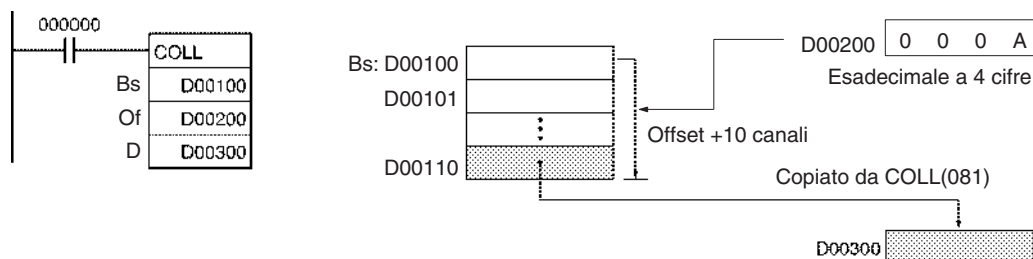
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se i dati di origine sono 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra dei dati di origine è 1. OFF in tutti gli altri casi. |

Avvertenze

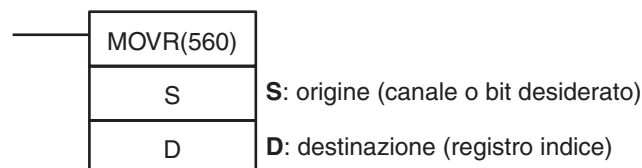
Assicurarsi che l'offset non superi la fine dell'area dati, ossia che Bs e Bs+Of siano nella stessa area dati.

Esempio

Quando CIO 000000 è ON, il contenuto di D00110 (D00100 + 10) viene copiato in D00300 purché il contenuto di D00200 sia 10 (0A esadecimale). Modificando l'offset in D00200, è possibile copiare il contenuto degli altri canali in D00300.

**3-8-14 MOVE TO REGISTER: MOVR(560)****Scopo**

Imposta l'indirizzo di memoria del PLC del canale, del bit o del flag di completamento del temporizzatore/contatore specificato nel registro degli indici specificato. Usare l'istruzione MOVRW(561) per impostare l'indirizzo di memoria del PLC relativo a un valore attuale del temporizzatore o del contatore in un registro indice.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | MOVR(560) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVR(560) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

D: destinazione

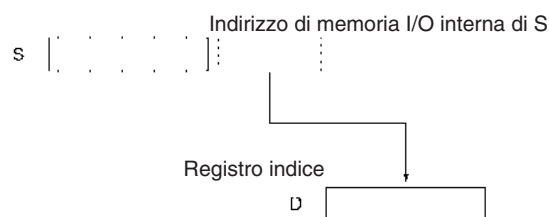
La destinazione deve essere un registro indice (da IR0 a IR15).

Caratteristiche operando

| Area | S | D |
|--|--|---------------|
| Area CIO | Da CIO 0000 a CIO 6143 Da CIO 000000 a CIO 614315 | --- |
| Area di lavoro | Da W000 a W511 Da W00000 a W51115 | --- |
| Area bit di ritentività | Da H000 a H511 Da H00000 a H51115 | --- |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 Da A00000 ad A44715 Da A44800 ad A95915 | --- |
| Area del temporizzatore | Da T0000 a T4095 (flag di completamento) | --- |
| Area del contatore | Da C0000 a C4095 (flag di completamento) | --- |
| Flag dei task | Da TK0000 a TK0031 | --- |
| Area DM | Da D00000 a D32767 | --- |
| Area EM senza banco | Da E00000 a E32767 | --- |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- |
| Indirizzi DM/EM indiretti in BCD | --- | --- |
| Costanti | --- | --- |
| Registri dati | --- | --- |
| Registri indice | --- | Da IR0 a IR15 |
| Indirizzamento indiretto con i registri indice | --- | --- |

Descrizione

MOVR(560) trova l'indirizzo di memoria del PLC (indirizzo assoluto) di S e lo scrive in D (un registro indice).



Se un temporizzatore o un contatore è specificato in S, MOVR(560) scrive l'indirizzo di memoria del PLC del flag di completamento del temporizzatore/contatore in D. Per scrivere in D l'indirizzo di memoria del PLC del valore attuale del temporizzatore/contatore, utilizzare MOVRW(561).

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF o non modificato (vedere nota) |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

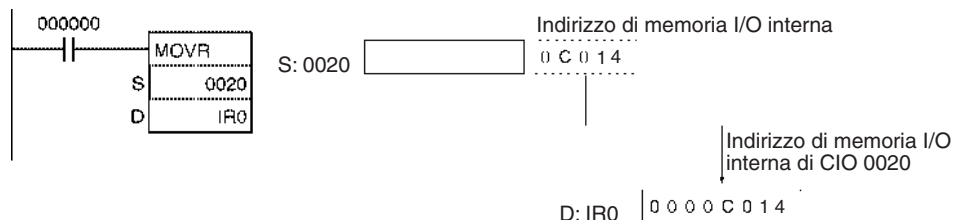
Con MOVR(560) non è possibile impostare gli indirizzi di memoria del PLC dei valori attuali di temporizzatore/contatore. Per impostare gli indirizzi di memoria del PLC dei valori attuali di temporizzatore/contatore, utilizzare MOVRW(561).

Il contenuto di un registro indice in un task ad interrupt non è prevedibile se non viene impostato. In un task ad interrupt, assicurarsi di impostare un registro mediante MOVR(560) prima di utilizzarlo.

Qualsiasi modifica al contenuto di un IR o DR apportata in un task ad interrupt non influisce sul contenuto del registro in un task ciclo.

Esempio

Quando CIO 000000 è ON, MOVR(560) scrive l'indirizzo di memoria del PLC di CIO 0020 in IR0.

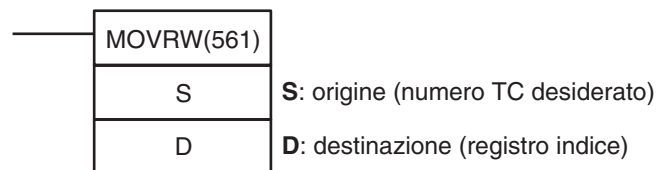


3-8-15 MOVE TIMER/COUNTER PV TO REGISTER: MOVRW(561)

Scopo

Imposta l'indirizzo di memoria del PLC del valore attuale del temporizzatore/contatore specificato nel registro indice specificato. Utilizzare l'istruzione MOVR(560) per impostare l'indirizzo di memoria del PLC relativo a un canale, a un bit o a un flag di completamento del temporizzatore/contatore in un registro indice.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | MOVR(561) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVR(561) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

D: destinazione

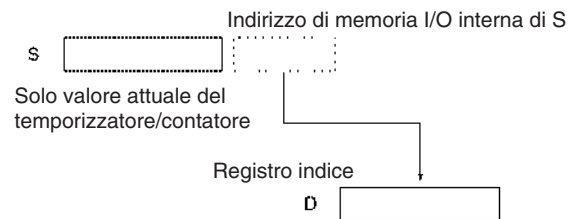
La destinazione deve essere un registro indice (da IR0 a IR15).

Caratteristiche operando

| Area | S | D |
|---|--------------------------------------|---------------|
| Area CIO | --- | |
| Area di lavoro | --- | |
| Area bit di ritenività | --- | |
| Area bit ausiliaria | --- | |
| Area del temporizzatore | Da T0000 a T4095 (valore attuale) | --- |
| Area del contatore | Da C0000 a C4095 (valore attuale) | --- |
| Area DM | --- | |
| Area EM senza banco | --- | |
| Area EM con banco | --- | |
| Indirizzi DM/EM indiretti in binario | --- | |
| Indirizzi DM/EM indiretti in BCD | --- | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | Da IR0 a IR15 |
| Indirizzamento indiretto con i registri indice | --- | |

Descrizione

MOVRW(561) trova l'indirizzo di memoria del PLC per il valore attuale di un temporizzatore o contatore specificato in S e lo scrive in D (un registro indice).



MOVRW(561) imposta l'indirizzo di memoria del PLC del valore attuale del temporizzatore o contatore in D. Per impostare l'indirizzo di memoria del PLC del flag di completamento del temporizzatore o contatore, utilizzare MOVR(560).

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF o non modificato (vedere nota) |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag negativo | N | OFF o non modificato (vedere nota) |

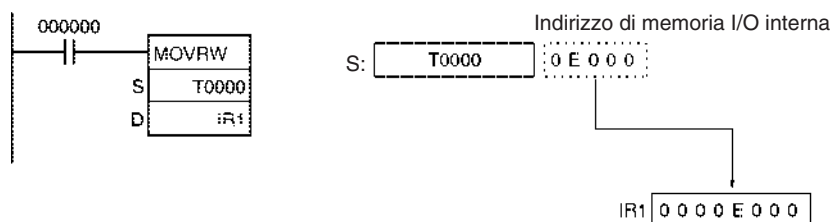
Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

Con MOVRW(561) non è possibile impostare gli indirizzi di memoria del PLC di canali, bit o flag di completamento di temporizzatori/contatori delle aree dati. Per impostarli, utilizzare MOVR(560).

Esempio

Quando CIO 000000 è ON, MOVW(561) scrive l'indirizzo di memoria del PLC per il valore attuale del temporizzatore T0000 in IR1.

**3-9 Istruzioni di scorrimento dei dati**

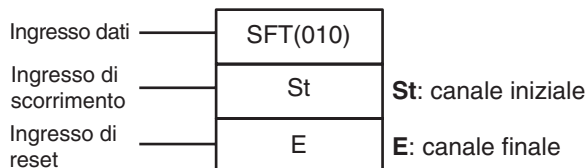
In questa sezione vengono descritte le istruzioni utilizzate per fare scorrere i dati nei o tra canali, ma in numero e con direzione differenti.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|-----------------------------------|-----------|-----------------|--------|
| SHIFT REGISTER | SFT | 010 | 345 |
| REVERSIBLE SHIFT REGISTER | SFTR | 084 | 346 |
| ASYNCHRONOUS SHIFT REGISTER | ASFT | 017 | 349 |
| WORD SHIFT | WSFT | 016 | 352 |
| ARITHMETIC SHIFT LEFT | ASL | 025 | 354 |
| DOUBLE SHIFT LEFT | ASLL | 570 | 355 |
| ARITHMETIC SHIFT RIGHT | ASR | 026 | 357 |
| DOUBLE SHIFT RIGHT | ASRL | 571 | 358 |
| ROTATE LEFT | ROL | 027 | 360 |
| DOUBLE ROTATE LEFT | ROLL | 572 | 362 |
| ROTATE LEFT WITHOUT CARRY | RLNC | 574 | 367 |
| DOUBLE ROTATE LEFT WITHOUT CARRY | RLNL | 576 | 369 |
| ROTATE RIGHT | ROR | 028 | 364 |
| DOUBLE ROTATE RIGHT | RORL | 573 | 365 |
| ROTATE RIGHT WITHOUT CARRY | RRNC | 575 | 371 |
| DOUBLE ROTATE RIGHT WITHOUT CARRY | RRNL | 577 | 372 |
| ONE DIGIT SHIFT LEFT | SLD | 074 | 374 |
| ONE DIGIT SHIFT RIGHT | SRD | 075 | 376 |
| SHIFT N-BIT DATA LEFT | NSFL | 578 | 377 |
| SHIFT N-BIT DATA RIGHT | NSFR | 579 | 379 |
| SHIFT N-BITS LEFT | NASL | 580 | 381 |
| DOUBLE SHIFT N-BITS LEFT | NSLL | 582 | 384 |
| SHIFT N-BITS RIGHT | NASR | 581 | 387 |
| DOUBLE SHIFT N-BITS RIGHT | NSRL | 583 | 389 |

3-9-1 SHIFT REGISTER: SFT(010)

Scopo Utilizza un registro di scorrimento.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | SFT(010) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

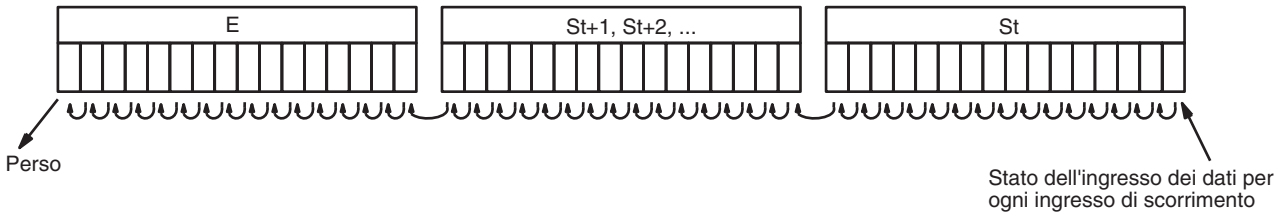
Nota St ed E devono essere nella stessa area dati.

Caratteristiche operando

| Area | St | E |
|--|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 ad A959 | |
| Area del temporizzatore | --- | |
| Area del contatore | --- | |
| Area DM | --- | |
| Area EM senza banco | --- | |
| Area EM con banco | --- | |
| Indirizzi DM/EM indiretti in binario | --- | |
| Indirizzi DM/EM indiretti in BCD | --- | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

Quando la condizione di esecuzione dell'ingresso di scorrimento passa da OFF a ON, tutti i dati da St a E vengono fatti scorrere a sinistra di un bit (dal bit più a destra al bit più a sinistra) e lo stato ON/OFF dei dati di ingresso viene scritto nel bit più a destra.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se l'indirizzo indiretto di IR per St ed E non è nelle aree dati di CIO, AR, HR o WR. OFF in tutti gli altri casi. |

Avvertenze

I dati del bit che sono stati fatti scorrere fuori dal registro di shift vengono scartati.

Quando l'ingresso di reset passa a ON, tutti i bit nel registro di shift a partire dal canale designato più a destra (St) al canale designato più a sinistra (E) vengono reimpostati, ossia impostati su 0. Rispetto ad altri ingressi, l'ingresso di reset ha la precedenza.

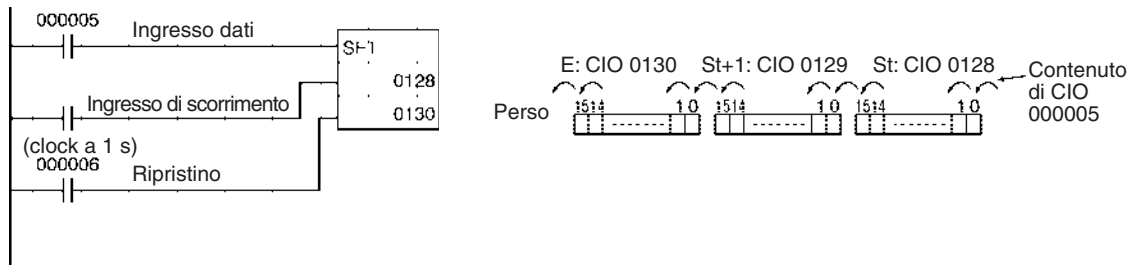
St deve essere minore o uguale a E; tuttavia, nel caso in cui St venisse impostato su un valore maggiore di E, non si verificherebbe alcun errore e un canale di dati in St verrebbe fatto scorrere.

Quando St ed E vengono designati indirettamente mediante registri indice e gli indirizzi effettivi nella memoria di I/O non sono in aree di memoria per i dati, si verifica un errore e viene attivato il flag di errore.

Esempio

Registro di shift che supera 16 bit

In questo caso, un registro di shift a 48 bit utilizza i canali da CIO 0128 a CIO 0130. Un impulso di clock a 1 s viene utilizzato per fare scorrere a ogni secondo la condizione di esecuzione prodotta da CIO 000005 in un registro a 3 canali da CIO 012800 a CIO 013015.

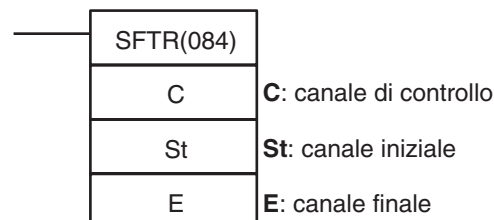


3-9-2 REVERSIBLE SHIFT REGISTER: SFTR(084)

Scopo

Crea un registro di shift che fa scorrere i dati a destra o a sinistra.

Simbolo programmazione ladder



Variazioni

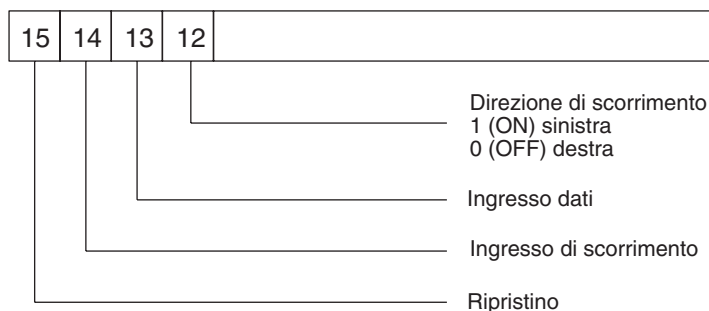
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SFTR(084) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SFTR(084) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

C: canale di controllo



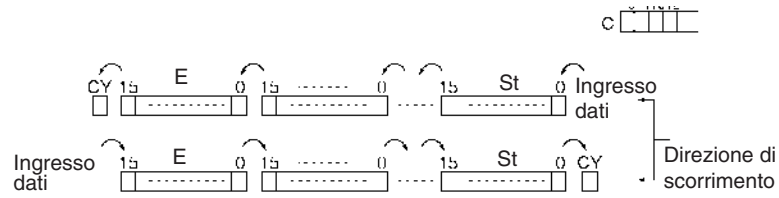
Nota St ed E devono essere nella stessa area dati.

Caratteristiche operando

| Area | C | St | E |
|--|--|-----------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Quando la condizione di esecuzione del bit dell'ingresso di scorrimento (bit 14 di C) passa a ON, tutti i dati da St a E vengono spostati di un bit nella direzione di scorrimento designata dal bit 12 di C e lo stato ON/OFF dei dati di ingresso viene scritto nel bit più a destra o nel bit più a sinistra. I dati del bit spostato dal registro di shift vengono scritti nel flag di riporto (CY).



Flag

| Nome | Etichetta | Funzionamento |
|-----------------|-----------|---|
| Flag di errore | ER | ON se St è maggiore di E. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 vi viene fatto scorrere. OFF quando 0 vi viene fatto scorrere. OFF quando il ripristino è impostato su 1. |

Avvertenze

Le operazioni di scorrimento sopra illustrate sono applicabili quando il bit di reset (bit 15 di C) viene disattivato.

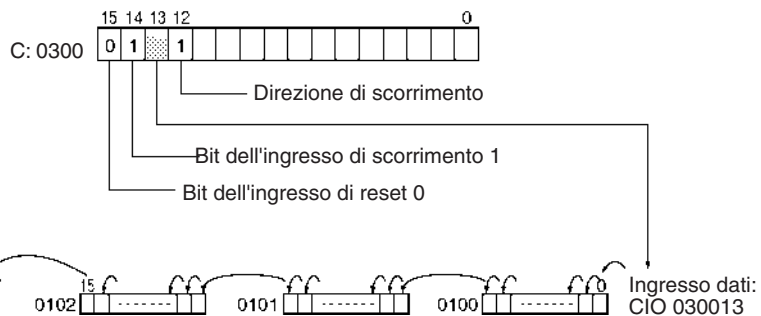
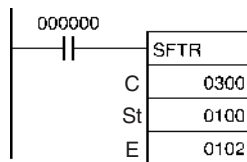
All'attivazione del bit di reset (bit 15 di C), vengono reimpostati, cioè impostati su 0, tutti i bit del registro di shift compresi tra St ed E.

Quando St è maggiore di E, viene generato un errore e verrà attivato il flag di errore.

Esempio

Scorrimento di dati

Se l'ingresso di scorrimento CIO 030014 passa a ON quando CIO 000000 è ON e il bit di reset CIO 030015 è OFF, i canali da CIO 0100 a CIO 0102 vengono fatti scorrere di un bit nella direzione designata da CIO 030012 (ossia a 1: destra) e il contenuto del bit di ingresso CIO 030013 viene fatto scorrere nel bit più a destra, cioè CIO 010000. Il contenuto di CIO 010215 verrà fatto scorrere al flag di scorrimento (CY).



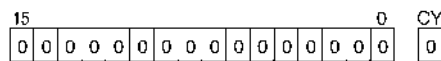
Reimpostazione dei dati

Se CIO 030014 è ON quando CIO 000000 è ON e il bit di reset CIO 030015 è ON, i canali da CIO 0100 a CIO 0102 e il flag di riporto vengono impostati su OFF.

Controllo dei dati

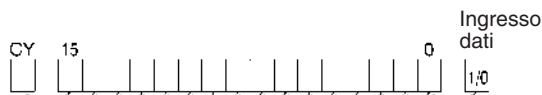
Reimpostazione dei dati

Quando il bit dell'ingresso di reset (bit 15 di C) è ON, tutti i bit da St a E e il flag di riporto vengono impostati su 0 e non è possibile ricevere altri dati.



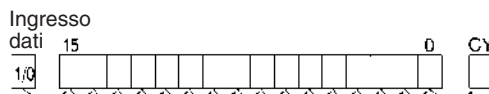
Scorrimento dei dati a sinistra (dal bit più a destra al bit più a sinistra)

Quando il bit dell'ingresso di scorrimento (bit 14 di C) è ON, il contenuto del bit di ingresso (bit 13 di C) viene fatto scorrere al bit 00 del canale iniziale e tutti i bit successivi vengono fatti scorrere a sinistra di un bit. Lo stato del bit 15 del canale finale viene fatto scorrere al flag di riporto.



Scorrimento dei dati a destra (dal bit più a sinistra al bit più a destra)

Quando il bit dell'ingresso di scorrimento (bit 14 di C) è ON, il contenuto del bit di ingresso (bit 13 di C) (I/O) viene fatto scorrere al bit 15 del canale finale e tutti i bit successivi vengono fatti scorrere a destra di un bit. Lo stato del bit 00 del canale iniziale viene fatto scorrere al flag di riporto.

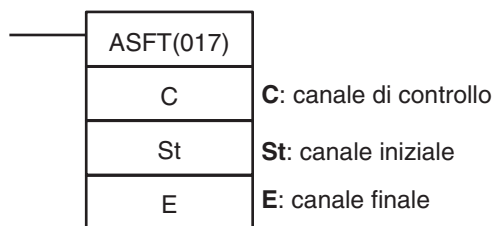


3-9-3 ASYNCHRONOUS SHIFT REGISTER: ASFT(017)

Scopo

Fa scorrere tutti i dati diversi da zero all'interno dell'intervallo di canali specificato verso St o verso E e sostituisce i dati esadecimali 0000 nel canale.

Simbolo programmazione ladder



Variazioni

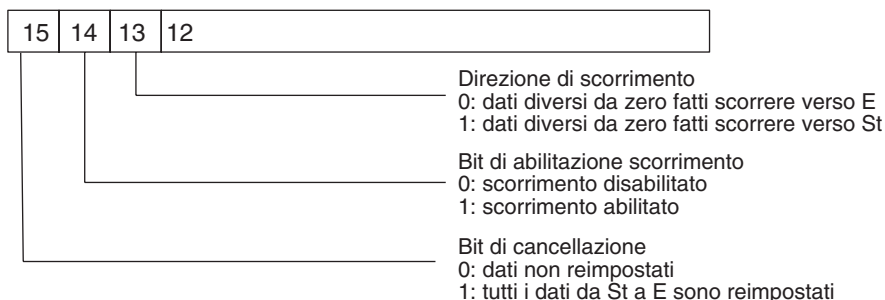
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ASFT(017) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASFT(017) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

C: canale di controllo



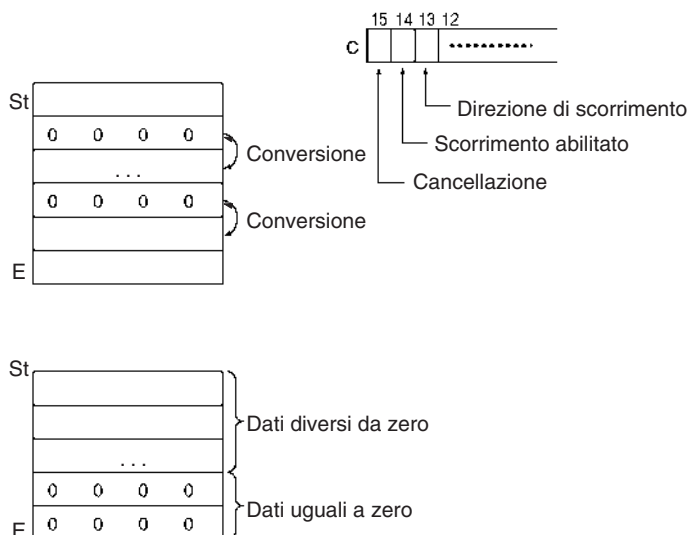
Nota St ed E devono essere nella stessa area dati.

Caratteristiche operando

| Area | C | St | E |
|--|--|-----------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Quando il bit di abilitazione scorrimento (bit 14 di C) è ON, tutti i canali con contenuto diverso da zero compresi tra St ed E vengono fatti scorrere di un canale nella direzione determinata dal bit della direzione di scorrimento (bit 13 di C) purché il canale nella direzione dello scorrimento contenga tutti dati uguali a zero. Se ASFT(017) viene ripetuta un numero sufficiente di volte, la totalità dei canali contenenti tutti dati uguali a zero viene sostituita da canali contenenti dati diversi da zero. Di conseguenza, tutti i dati tra St ed E vengono suddivisi tra dati uguali a zero e dati diversi da zero.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se St è maggiore di E. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione in background è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |

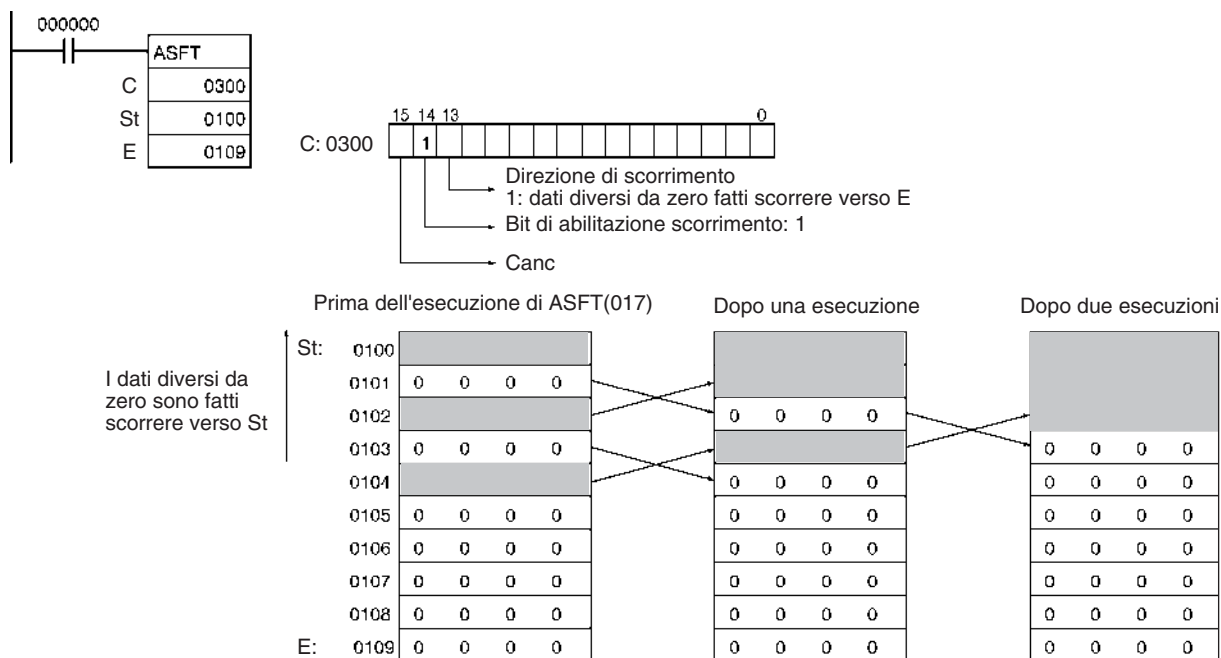
Avvertenze

All'attivazione del flag di cancellazione (bit 15 di C), vengono reimpostati, cioè impostati su 0, tutti i bit del registro di shift compresi tra da St ed E. Il flag di cancellazione ha la precedenza sul bit di abilitazione scorrimento (bit 14 di C). Quando St è maggiore di E, viene generato un errore e verrà attivato il flag di errore.

Esempio

Scorrimento di dati

Se il bit di abilitazione scorrimento CIO 030014 passa a ON quando CIO 000000 è ON, tutti i canali che contengono dati diversi da zero compresi tra CIO 0100 e CIO 0109 vengono fatti scorrere nella direzione designata dal bit della direzione di scorrimento CIO 030013 (ossia, 1: verso St), purché il canale a sinistra dei dati diversi da zero contenga tutti dati uguali a zero.

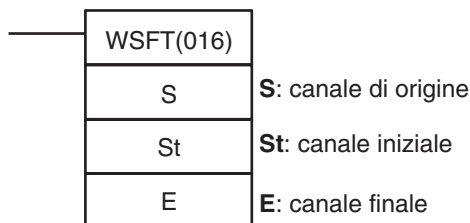


3-9-4 WORD SHIFT: WSFT(016)

Scopo

Fa scorrere i dati tra St ed E, un canale alla volta.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | WSFT(016) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @WSFT(016) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota St ed E devono essere nella stessa area dati.

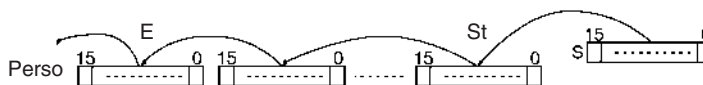
Caratteristiche operando

| Area | S | St | E |
|-------------------------|------------------------|-----------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |

| Area | S | St | E |
|--|--|-----|---|
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

WSFT(016) fa scorrere i dati da St a E in unità canale e i dati dal canale di origine S sono inviati in St. Il contenuto di E viene perso.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se St è maggiore di E. OFF in tutti gli altri casi. |

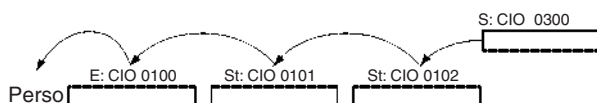
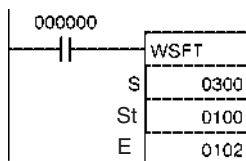
Avvertenze

Quando St è maggiore di E, viene generato un errore e verrà attivato il flag di errore.

Nota Quando viene fatto scorrere un numero elevato di dati, il tempo di esecuzione dell'istruzione si allunga. Assicurarsi che durante l'esecuzione di WSFT(016) non avvengano interruzioni nell'alimentazione poiché potrebbero provocare il mancato completamento dell'operazione di scorrimento.

Esempio

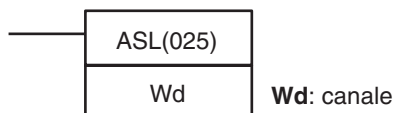
Quando CIO 000000 è ON, i dati da CIO 0100 a CIO 0102 vengono fatti scorrere di un canale verso E. Il contenuto di CIO 0300 viene memorizzato in CIO 0100 mentre quello di CIO 0102 verrà perso.



3-9-5 ARITHMETIC SHIFT LEFT: ASL(025)

Scopo Fa scorrere il contenuto del canale Wd di un bit a sinistra.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ASL(025) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASL(025) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

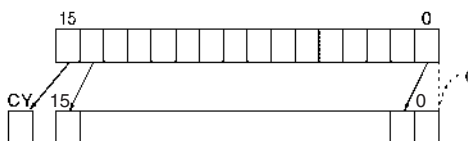
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

ASL(025) fa scorrere il contenuto di Wd di un bit a sinistra, dal bit più a destra al bit più a sinistra. "0" viene inviato nel bit più a destra e i dati dal bit più a sinistra vengono fatti scorrere al flag di riporto (CY).



Flag

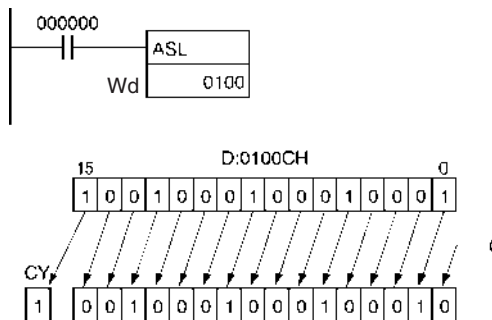
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Quando ASL(025) viene eseguita, il flag di errore viene disattivato.
 Se come risultato dello scorrimento il contenuto di Wd è zero, verrà attivato il flag di uguaglianza.
 Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, CIO 0100 viene fatto scorrere di un bit a sinistra. "0" è inviato in CIO 010000 e il contenuto di CIO 010115 viene fatto scorrere al flag di riporto (CY).

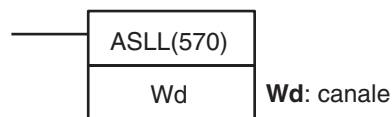


3-9-6 DOUBLE SHIFT LEFT: ASLL(570)

Scopo

Fa scorrere il contenuto dei canali Wd e Wd+1 di un bit a sinistra.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ASLL(570) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASLL(570) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

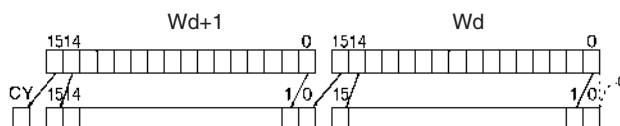
Caratteristiche operando

| Area | Wd |
|----------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |

| Area | Wd |
|--|--|
| Area bit di ritentività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--),IR0 a ,-(--),IR15 |

Descrizione

ASLL(570) fa scorrere il contenuto dei canali Wd e Wd + 1 di un bit a sinistra, dal bit più a destra al bit più a sinistra. "0" è inviato nel bit più a destra di Wd e il contenuto del bit più a sinistra di Wd e Wd + 1 viene fatto scorrere al flag di riporto (CY).

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

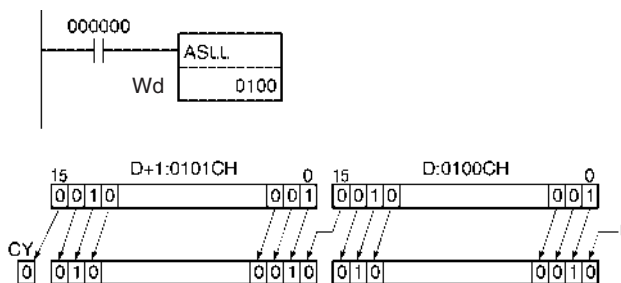
Quando ASLL(570) viene eseguita, il flag di errore viene disattivato.

Se come risultato dello scorrimento il contenuto dei canali Wd e Wd + 1 è zero, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd + 1 è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, CIO 0100 e CIO 0101 del canale vengono fatti scorrere di un bit a sinistra. "0" è inviato in CIO 010000 e il contenuto di CIO 010015 viene fatto scorrere al flag di riporto (CY).

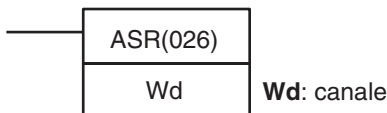


3-9-7 ARITHMETIC SHIFT RIGHT: ASR(026)

Scopo

Fa scorrere il contenuto del canale Wd di un bit a destra.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ASR(026) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASR(026) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

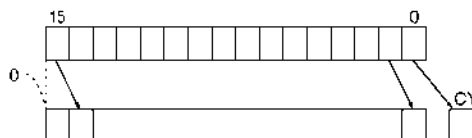
Caratteristiche operando

| Area | Wd |
|--------------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @E00000 a @E32767 Da @En_00000 a @En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |

| Area | Wd |
|--|--|
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

ASR(026) fa scorrere il contenuto di Wd di un bit a destra, dal bit più a sinistra al bit più a destra. "0" è inviato nel bit più a sinistra e il contenuto del bit più a destra viene fatto scorrere al flag di riporto (CY).

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | OFF |

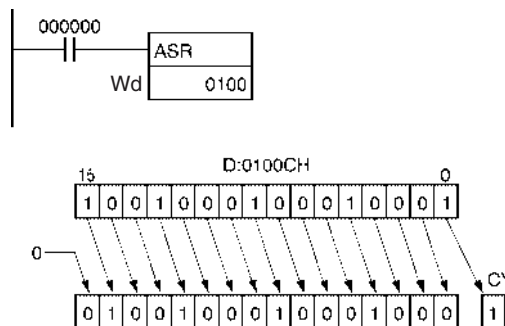
Avvertenze

Quando ASR(026) viene eseguita, il flag di errore e il flag negativo vengono disattivati.

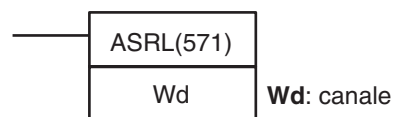
Se come risultato dello scorrimento il contenuto di Wd è zero, verrà attivato il flag di uguaglianza.

Esempio

Quando CIO 000000 è ON, CIO 0100 del canale viene fatto scorrere di un bit a destra. "0" è inviato in CIO 010015 e il contenuto di CIO 010000 viene fatto scorrere al flag di riporto (CY).

**3-9-8 DOUBLE SHIFT RIGHT: ASRL(571)****Scopo**

Fa scorrere il contenuto dei canali Wd e Wd+1 di un bit a destra.

Simbolo programmazione ladder

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ASRL(571) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASRL(571) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

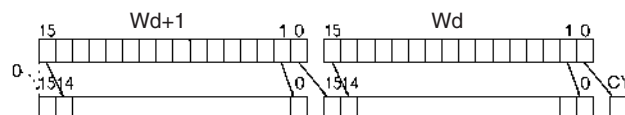
| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

ASRL(571) fa scorrere il contenuto dei canali Wd e Wd + 1 di un bit a destra, dal bit più a sinistra al bit più a destra. "0" è inviato nel bit più a sinistra di Wd + 1 e il contenuto del bit più a destra di Wd viene fatto scorrere al flag di riporto (CY).



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |

| Nome | Etichetta | Funzionamento |
|-----------------|-----------|---|
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | OFF |

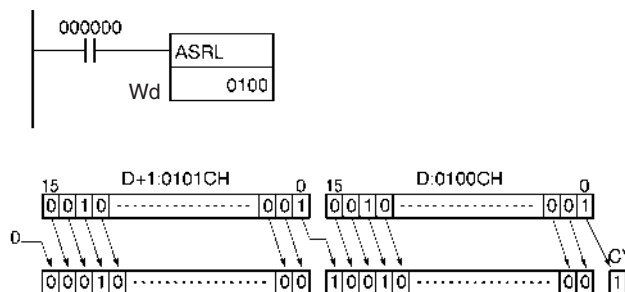
Avvertenze

Quando ASRL (571) viene eseguita, il flag di errore e il flag negativo vengono disattivati.

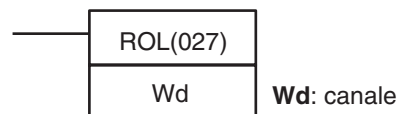
Se come risultato dello scorrimento il contenuto dei canali Wd e Wd + 1 è zero, verrà attivato il flag di uguaglianza.

Esempio

Quando CIO 000000 è ON, CIO 0100 e CIO 0101 del canale vengono fatti scorrere di un bit a destra. "0" è inviato in CIO 010115 e il contenuto di CIO 010000 viene fatto scorrere al flag di riporto (CY).

**3-9-9 ROTATE LEFT: ROL(027)****Scopo**

Fa scorrere tutti i bit del canale Wd di un bit a sinistra, incluso il flag di riporto (CY).

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | ROL(027) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ROL(027) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

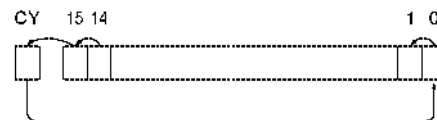
Caratteristiche operando

| Area | Wd |
|-------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |

| Area | Wd |
|--|--|
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

ROL(027) fa scorrere tutti i bit del canale Wd, incluso il flag di riporto (CY), a sinistra, dal bit più a destra al bit più a sinistra.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Quando ROL(027) viene eseguita, il flag di errore viene disattivato.

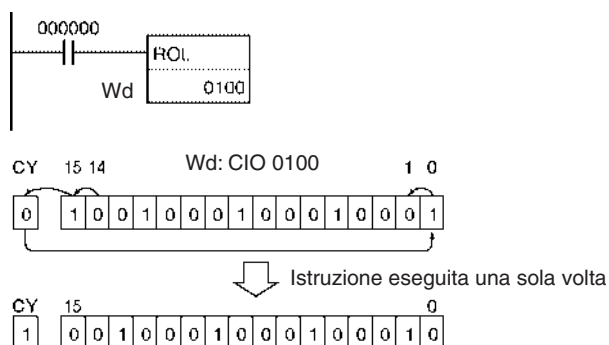
Se come risultato dello scorrimento il contenuto di Wd è zero, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd è 1, verrà attivato il flag negativo.

Nota Utilizzando l'istruzione Set Carry (STC(040)) o Clear Carry (CLC(041)), è possibile impostare il contenuto del flag di riporto su 1 o 0 appena prima dell'esecuzione di ROL (027).

Esempio

Quando CIO 000000 è ON, CIO 0100 e il flag di riporto (CY) del canale vengono fatti scorrere di un bit a sinistra. Il contenuto di CIO 010015 viene fatto scorrere al flag di riporto (CY) e il contenuto di tale flag verrà fatto scorrere in CIO 010000.

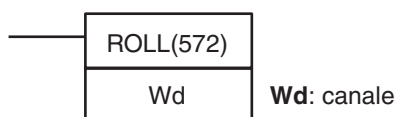


3-9-10 DOUBLE ROTATE LEFT: ROLL(572)

Scopo

Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a sinistra, incluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ROLL(572) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ROLL(572) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

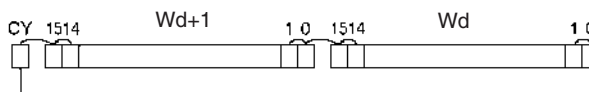
Caratteristiche operando

| Area | Wd |
|--------------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |

| Area | Wd |
|--|--|
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

ROLL(572) fa scorrere tutti i bit dei canali Wd e Wd + 1, incluso il flag di riporto (CY), a sinistra, dal bit più a destra al bit più a sinistra.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

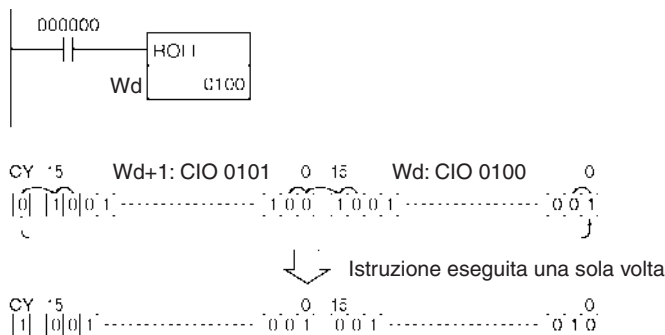
Avvertenze

Quando ROLL(572) viene eseguita, il flag di errore viene disattivato.
Se come risultato dello scorrimento il contenuto di Wd e Wd + 1 è zero, verrà attivato il flag di uguaglianza.
Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd + 1 è 1, verrà attivato il flag negativo.

Nota Utilizzando l'istruzione Set Carry (STC(040)) o Clear Carry (CLC(041)), è possibile impostare il contenuto del flag di riporto su 1 o 0 appena prima dell'esecuzione di ROL (027).

Esempio

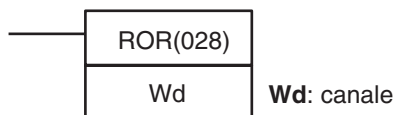
Quando CIO 000000 è ON, CIO 0100, CIO 0101 e il flag di riporto (CY) del canale vengono fatti scorrere di un bit a sinistra. Il contenuto di CIO 010015 viene fatto scorrere al flag di riporto (CY) e il contenuto di tale flag verrà fatto scorrere in CIO 010000.



3-9-11 ROTATE RIGHT: ROR(028)

Scopo Fa scorrere tutti i bit del canale Wd di un bit a destra, incluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ROR(028) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ROR(028) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

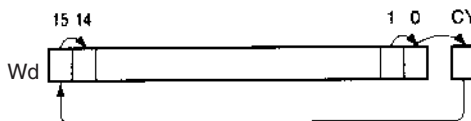
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

ROR(028) fa scorrere tutti i bit del canale Wd, incluso il flag di riporto (CY), a destra, dal bit più a sinistra al bit più a destra.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

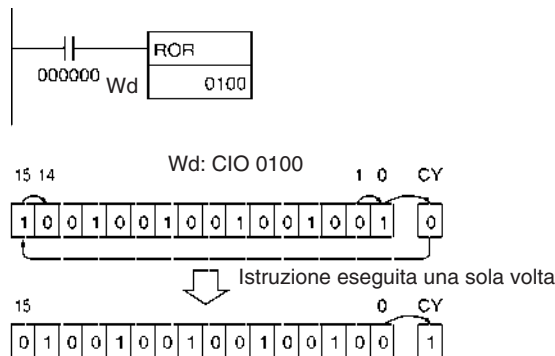
Avvertenze

Quando ROR(028) viene eseguita, il flag di errore viene disattivato.
 Se come risultato dello scorrimento il contenuto di Wd è zero, verrà attivato il flag di uguaglianza.
 Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd è 1, verrà attivato il flag negativo.

Nota Utilizzando l'istruzione Set Carry (STC(040)) o Clear Carry (CLC(041)), è possibile impostare il contenuto del flag di riporto su 1 o 0 appena prima dell'esecuzione di ROL (027).

Esempio

Quando CIO 000000 è ON, CIO 0100 e il flag di riporto (CY) del canale vengono fatti scorrere di un bit a destra. Il contenuto di CIO 010000 viene fatto scorrere al flag di riporto (CY) e il contenuto di tale flag verrà fatto scorrere in CIO 010015.

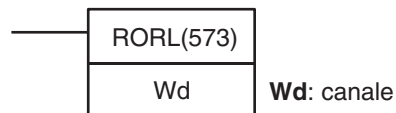


3-9-12 DOUBLE ROTATE RIGHT: RORL(573)

Scopo

Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a destra, incluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RORL(573) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RORL(573) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

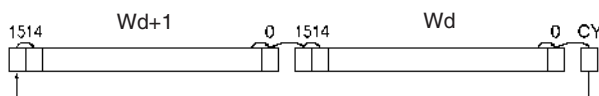
| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

RORL(573) fa scorrere tutti i bit dei canali Wd e Wd + 1, incluso il flag di riporto (CY), a destra, dal bit più a sinistra al bit più a destra.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |

| Nome | Etichetta | Funzionamento |
|-----------------|-----------|---|
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

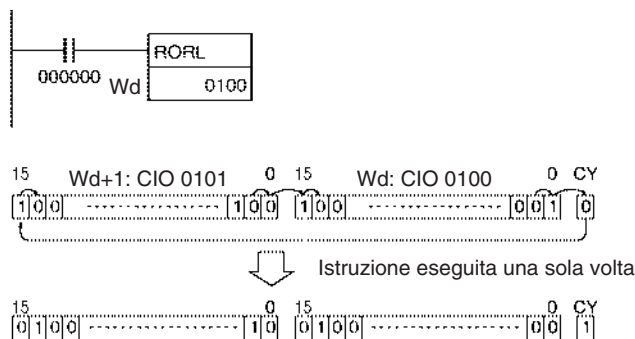
Avvertenze

Quando RORL(573) viene eseguita, il flag di errore viene disattivato.
 Se come risultato dello scorrimento il contenuto di Wd e Wd + 1 è zero, verrà attivato il flag di uguaglianza.
 Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd +1 è 1, verrà attivato il flag negativo.

Nota Utilizzando l'istruzione Set Carry (STC(040)) o Clear Carry (CLC(041)), è possibile impostare il contenuto del flag di riporto su 1 o 0 appena prima dell'esecuzione di ROL (027).

Esempio

Quando CIO 000000 è ON, CIO 0100, CIO 0101 e il flag di riporto (CY) del canale vengono fatti scorrere di un bit a destra. Il contenuto di CIO 010000 viene fatto scorrere al flag di riporto (CY) e il contenuto di tale flag verrà fatto scorrere in CIO 010115.

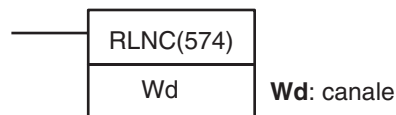


3-9-13 ROTATE LEFT WITHOUT CARRY: RLNC(574)

Scopo

Fa scorrere tutti i bit del canale Wd di un bit a sinistra, escluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | RLNC(574) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RLNC(574) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

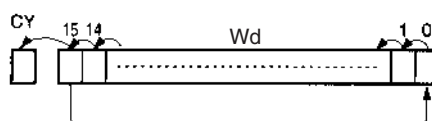
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

RLNC(574) fa scorrere tutti i bit di Wd a sinistra, dal bit più a destra al bit più a sinistra. Il contenuto del bit più a sinistra del canale Wd viene fatto scorrere nel bit più a destra e nel flag di riporto (CY).



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

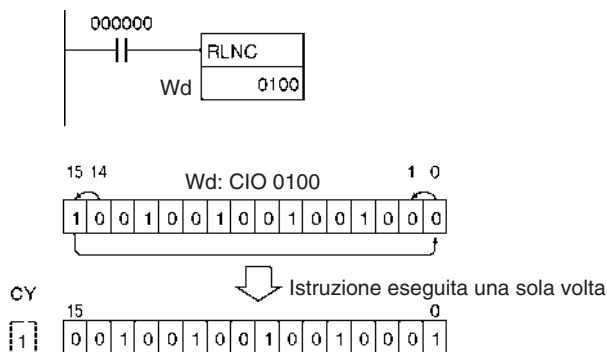
Quando RLNC(574) viene eseguita, il flag di errore viene disattivato.

Se come risultato dello scorrimento il contenuto di Wd è zero, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, CIO 0100 del canale viene fatto scorrere di un bit a sinistra, escluso il flag di riporto (CY). Il contenuto di CIO 010015 viene fatto scorrere a CIO 010000.

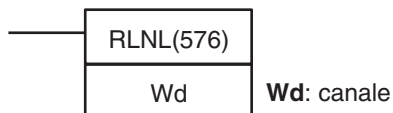


3-9-14 DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576)

Scopo

Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a sinistra, escluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RLNL(576) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RLNL(576) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

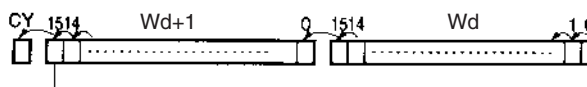
Caratteristiche operando

| Area | Wd |
|--------------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |

| Area | Wd |
|--|---|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

RLNL(576) fa scorrere tutti i bit dei canali Wd e Wd + 1 a sinistra, dal bit più a destra al bit più a sinistra. Il contenuto del bit più a sinistra del canale Wd+1 si sposta nel bit più a destra del canale Wd e nel flag di riporto (CY).



Flag

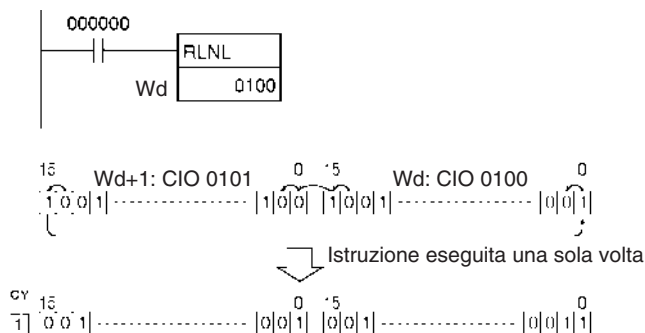
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Quando RLNL(576) viene eseguita, il flag di errore viene disattivato.
Se come risultato dello scorrimento il contenuto di Wd e Wd + 1 è zero, verrà attivato il flag di uguaglianza.
Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd + 1 è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, CIO 0100 e CIO 0101 del canale vengono fatti scorrere di un bit a sinistra, escluso il flag di riporto (CY). Il contenuto di CIO 010115 viene fatto scorrere a CIO 010000.

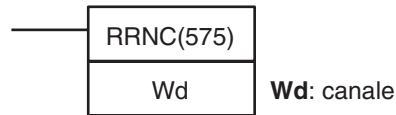


3-9-15 ROTATE RIGHT WITHOUT CARRY: RRNC(575)

Scopo

Fa scorrere tutti i bit del canale Wd di un bit a destra, escluso il flag di riporto (CY). Il contenuto del bit più a destra del canale Wd si sposta nel bit più a sinistra e nel flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RRNC(575) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RRNC(575) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

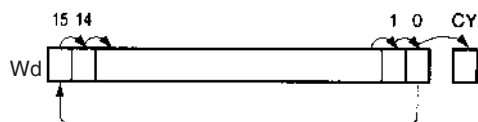
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

RRNC(575) fa scorrere tutti i bit del canale Wd a destra, dal bit più a sinistra al bit più a destra, escluso il flag di riporto (CY).



Flag

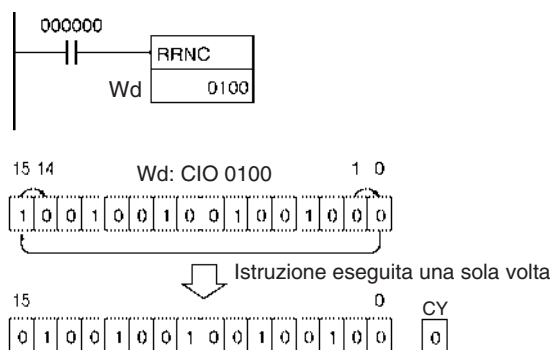
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Quando RRNC(575) viene eseguita, il flag di errore viene disattivato.
 Se come risultato dello scorrimento il contenuto di Wd è zero, verrà attivato il flag di uguaglianza.
 Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, CIO 0100 del canale viene fatto scorrere di un bit a destra, escluso il flag di riporto (CY). Il contenuto di CIO 010000 viene fatto scorrere a CIO 010015.

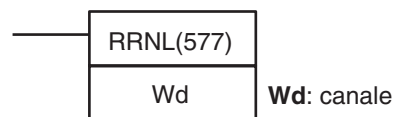


3-9-16 DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577)

Scopo

Fa scorrere tutti i bit dei canali Wd e Wd +1 di un bit a destra, escluso il flag di riporto (CY). Il contenuto del bit più a destra del canale Wd+1 si sposta nel bit più a sinistra del canale Wd e nel flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RRNL(577) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RRNL(577) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

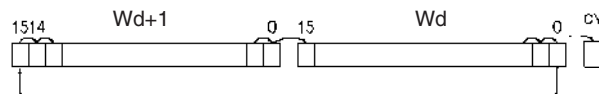
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

RRNC(577) fa scorrere tutti i bit dei canali Wd e Wd + 1 a destra, dal bit più a sinistra al bit più a destra, escluso il flag di riporto (CY).



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Quando RRNL(577) viene eseguita, il flag di errore viene disattivato.

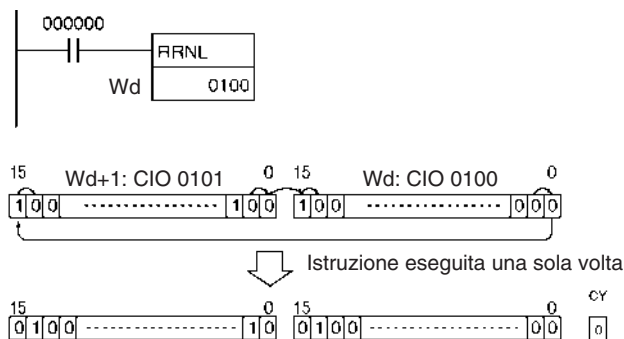
Se come risultato dello scorrimento il contenuto dei canali Wd e Wd + 1 è zero, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di Wd + 1 è 1, verrà attivato il flag negativo.

Nota Utilizzando l'istruzione Set Carry (STC(040)) o Clear Carry (CLC(041)), è possibile impostare il contenuto del flag di riporto su 1 o 0 appena prima dell'esecuzione di ROL (027).

Esempio

Quando CIO 000000 è ON, CIO 0100 e CIO 0101 dei canali vengono fatti scorrere di un bit a destra, escluso il flag di riporto (CY). Il contenuto di CIO 010000 viene fatto scorrere a CIO 010115.

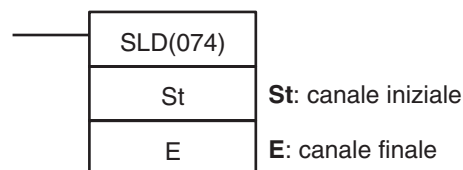


3-9-17 ONE DIGIT SHIFT LEFT: SLD(074)

Scopo

Fa scorrere i dati di una cifra (4 bit) a sinistra.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SLD(074) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SLD(074) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota St ed E devono essere nella stessa area dati.

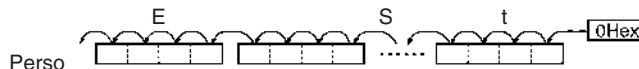
Caratteristiche operando

| Area | St | E |
|-------------------------|------------------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |

| Area | St | E |
|--|--|---|
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

SLD(074) fa scorrere i dati compresi tra St ed E di una cifra (4 bit) a sinistra. "0" viene inviato nella cifra più a destra (bit da 3 a 0 di St) e il contenuto della cifra più a sinistra (bit da 15 a 12 di E) viene perso.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se St è maggiore di E. OFF in tutti gli altri casi. |

Avvertenze

Quando St è maggiore di E, viene generato un errore e verrà attivato il flag di errore.

Nota Quando viene fatto scorrere un numero elevato di dati, il tempo di esecuzione dell'istruzione si allunga. Assicurarsi che durante l'esecuzione di SLD(074) non avvengano interruzioni nell'alimentazione poiché potrebbero provocare il mancato completamento dell'operazione di scorrimento.

Esempio

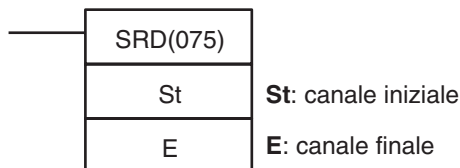
Quando CIO 000000 è ON, CIO 0100 - CIO 0101 dei canali vengono fatti scorrere di una cifra(4 bit) a sinistra. Uno zero viene inviato nei bit da 0 a 3 di CIO 0100 del canale e il contenuto dei bit da 12 a 15 di CIO 0102 sarà perso.



3-9-18 ONE DIGIT SHIFT RIGHT: SRD(075)

Scopo Fa scorrere i dati di una cifra (4 bit) a destra.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SRD(075) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SRD(075) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

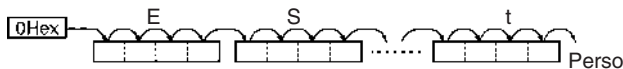
Nota St ed E devono essere nella stessa area dati.

Caratteristiche operando

| Area | St | E |
|--|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

SRD(075) fa scorrere i dati compresi tra St ed E di una cifra (4 bit) a destra. "0" viene inviato nella cifra più a sinistra (bit da 15 a 12 di E) e il contenuto della cifra più a destra (bit da 3 a 0 di St) viene perso.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se St è maggiore di E. OFF in tutti gli altri casi. |

Avvertenze

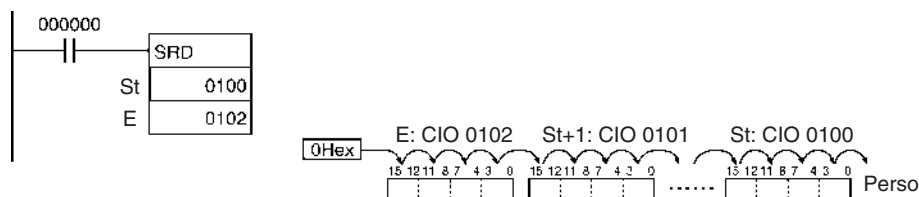
Quando St è maggiore di E, viene generato un errore e verrà attivato il flag di errore.

Quando SRD(075) viene eseguita, il flag di uguaglianza e il flag negativo vengono disattivati.

Nota Quando viene fatto scorrere un numero elevato di dati, il tempo di esecuzione dell'istruzione si allunga. Assicurarsi che durante l'esecuzione di SRD(075) non avvengano interruzioni nell'alimentazione poiché potrebbero provocare il mancato completamento dell'operazione di scorrimento.

Esempio

Quando CIO 000000 è ON, CIO 0100 ... CIO 0102 dei canali vengono fatti scorrere di una cifra (4 bit) a destra. Uno zero viene inviato nei bit da 12 a 15 di CIO 0102 e il contenuto dei bit da 0 a 3 di CIO 0100 del canale sarà perso.

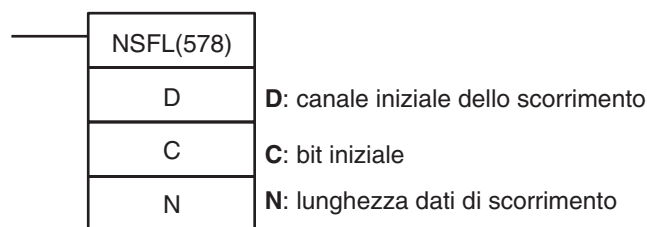


3-9-19 SHIFT N-BIT DATA LEFT: NSFL(578)

Scopo

Fa scorrere a sinistra il numero di bit specificato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | NSFL(578) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NSFL(578) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Flag

| Nome | Etichetta | Funzionamento |
|-----------------|-----------|---|
| Flag di errore | ER | ON quando i dati di C non sono tra 0000 e 000F esadecimale. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |

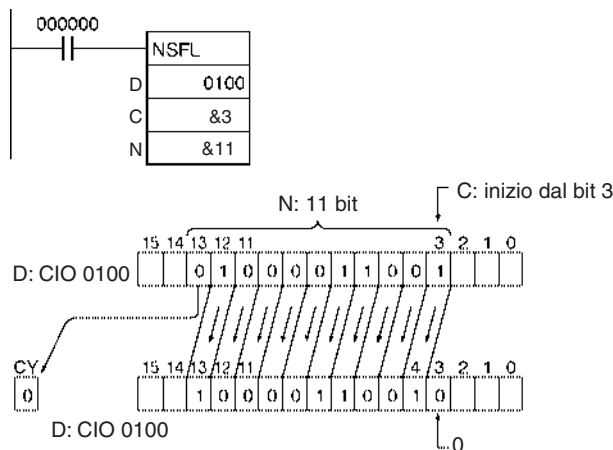
Avvertenze

Quando la lunghezza dati di scorrimento (N) è 0, il contenuto del bit iniziale viene copiato nel flag di riporto (CY) e resta inalterato.

Vengono modificati soltanto i bit fatti scorrere nel canale più a destra dell'area di scorrimento, ossia i dati del canale più a sinistra.

Esempio

Quando CIO 000000 è ON, tutti i bit dal bit iniziale 3 alla lunghezza dati di scorrimento (B esadecimale) vengono fatti scorrere di un bit a sinistra, dal bit più a destra al bit più a sinistra. "0" è inviato nel bit 3 di CIO 0100. Il contenuto del bit più a sinistra nell'area di scorrimento (bit 13 di CIO 0100) viene copiato nel flag di riporto (CY).

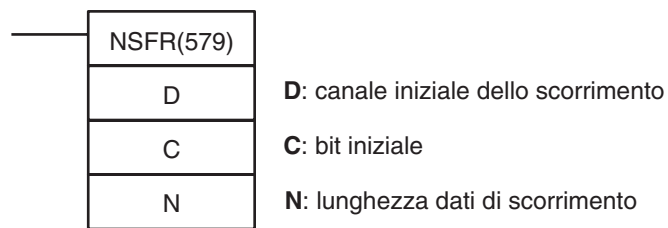


3-9-20 SHIFT N-BIT DATA RIGHT: NSFR(579)

Scopo

Fa scorrere a destra il numero di bit specificato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | NSFR(579) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NSFR(579) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: da 0000 a 000F esadecimale (da 0 a 15)

N: da 0000 a FFFF esadecimale (da 0 a 65535)

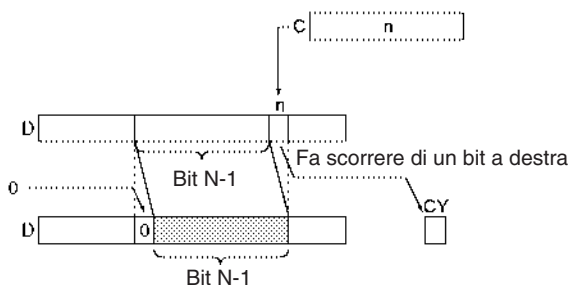
Nota Tutti i canali nel registro di shift devono essere nella stessa area.

Caratteristiche operando

| Area | D | C | N |
|--|--|---|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A448 ad A959 | Da A000 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a #000F (in formato binario) o da &0 a &15 | Da #0000 a #FFFF (in formato binario) o da &0 a &65535 |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

NSFR(579) fa scorrere il numero di bit specificato della lunghezza dati di scorrimento (N) dal bit iniziale (C) nel canale più a destra, come designato da D di un bit a destra, verso il canale e il bit più a destra. "0" è inviato nel bit iniziale e il contenuto del bit più a destra dell'area di scorrimento viene fatto scorrere al flag di riporto (CY).



Flag

| Nome | Etichetta | Funzionamento |
|-----------------|-----------|---|
| Flag di errore | ER | ON quando i dati di C non sono tra 0000 e 000F esadecimale. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |

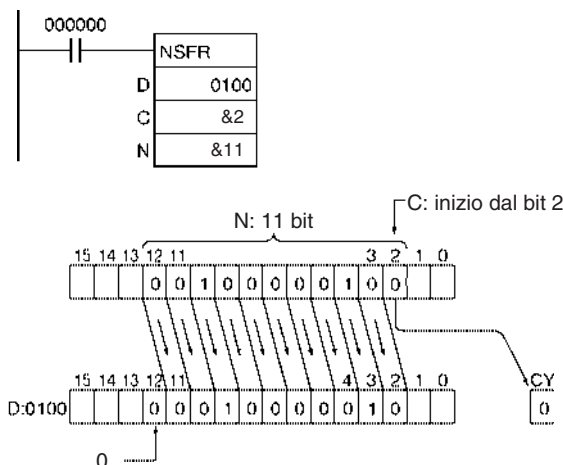
Avvertenze

Quando la lunghezza dati di scorrimento (N) è 0, il contenuto del bit iniziale viene copiato nel flag di riporto (CY) e resta inalterato.

Vengono modificati soltanto i bit fatti scorrere nel canale più a destra dell'area di scorrimento, ossia i dati del canale più a sinistra.

Esempio

Quando CIO 000000 è ON, tutti i bit dal bit iniziale 2 alla fine della lunghezza dati di scorrimento di 11 bit (B esadecimale) vengono fatti scorrere di un bit a destra, dal bit più a sinistra al bit più a destra. "0" è spostato nel bit 12 di CIO 0100. Il contenuto del bit più a destra nell'area di scorrimento (bit 2 di CIO 0100) viene copiato nel flag di riporto (CY).

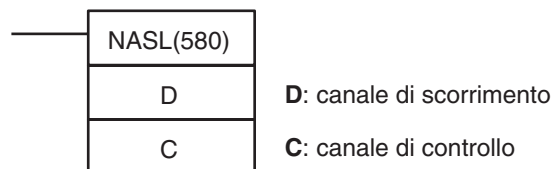


3-9-21 SHIFT N-BITS LEFT: NASL(580)

Scopo

Fa scorrere i 16 bit specificati dei dati canale a sinistra del numero di bit specificato.

Simbolo programmazione ladder

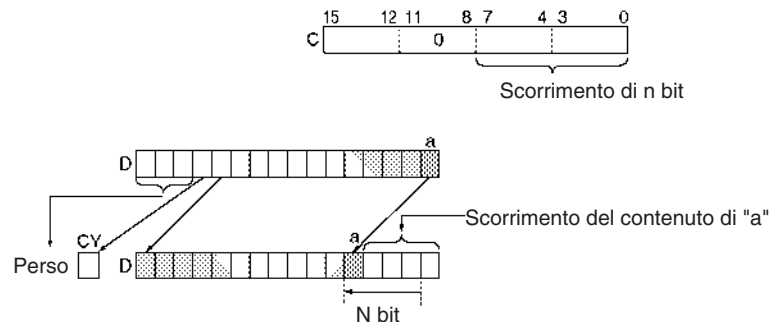


Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | NASL(580) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NASL(580) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON quando il canale di controllo C (numero di bit da fare scorrere) non è nell'intervallo. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Per ciascun bit fatto scorrere fuori dal canale specificato, il contenuto dell'ultimo bit viene fatto scorrere al flag di riporto (CY) e gli altri dati saranno persi.

Se il numero di bit da fare scorrere specificato in C è "0", i dati non saranno spostati. In ogni caso, i flag appropriati si attiveranno o disattiveranno in base ai dati nel canale specificato.

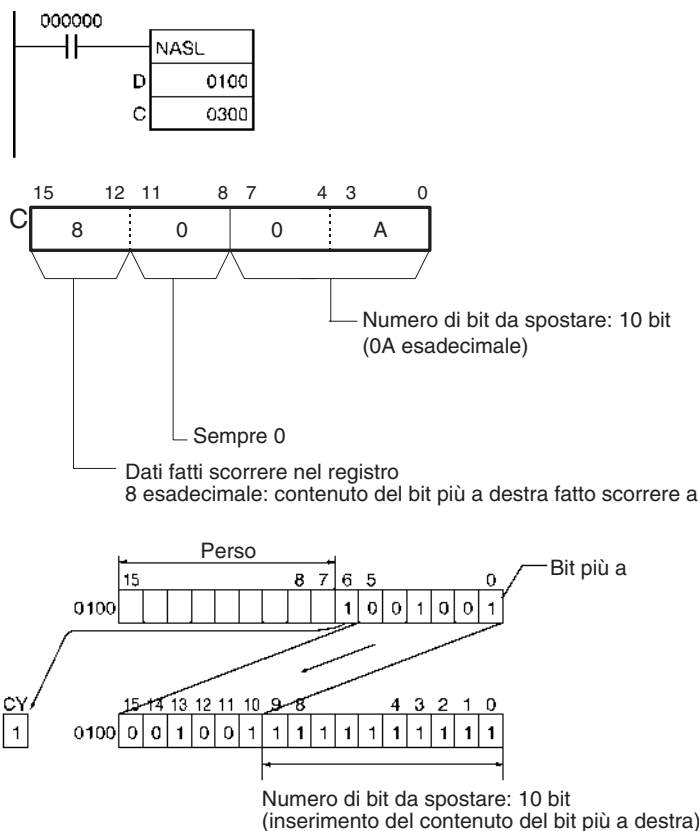
Se il contenuto del canale di controllo C non è nell'intervallo, verrà generato un errore e si attiverà il flag di errore.

Se come risultato dello scorrimento il contenuto di D è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di D è 1, verrà attivato il flag negativo.

Esempio

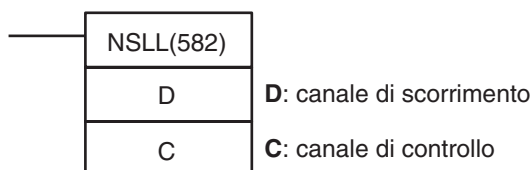
Quando CIO 000000 è ON, il contenuto di CIO 0100 viene fatto scorrere di 10 bit a sinistra, dal bit più a destra al bit più a sinistra. Il numero di bit da fare scorrere è specificato nei bit da 0 a 7 di CIO 0300 del canale (dati di controllo). Il contenuto del bit 0 di CIO 0100 viene copiato nei bit dai quali erano stati spostati i dati e il contenuto del bit più a destra, che era stato fatto scorrere fuori dall'intervallo, viene spostato nel flag di riporto (CY). Tutti gli altri dati vengono persi.



3-9-22 DOUBLE SHIFT N-BITS LEFT: NSLL(582)

Scopo Fa scorrere a sinistra i 32 bit specificati dei dati canale del numero di bit specificato.

Simbolo programmazione ladder



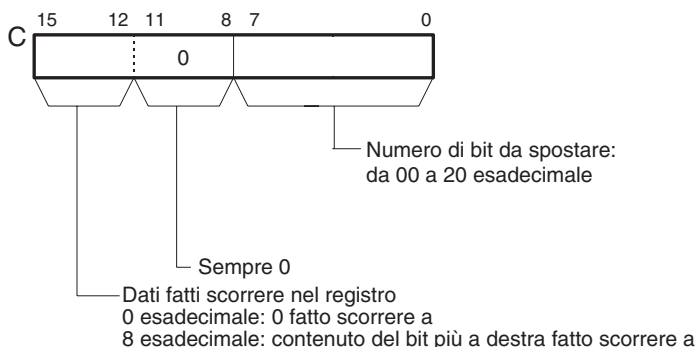
Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | NSLL(582) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NSLL(582) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi C: canale di controllo

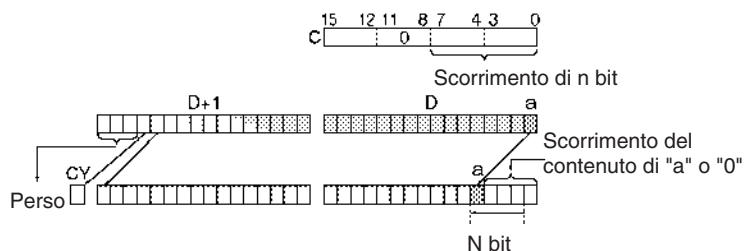


Caratteristiche operando

| Area | D | C |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A958 | Da A000 ad A959 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Solo valori specificati |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

NSLL(582) fa scorrere D e D + 1, cioè i canali di scorrimento, del numero specificato di bit binari indicato in C verso sinistra, dal bit più a destra al bit più a sinistra. Gli zeri o il valore del bit più a destra vengono inviati nel numero specificato di bit del canale di scorrimento a partire dal bit più a destra.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON quando il canale di controllo C, ossia il numero di bit da fare scorrere, non è nell'intervallo. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Per ciascun bit fatto scorrere fuori dal canale specificato, il contenuto dell'ultimo bit viene fatto scorrere al flag di riporto (CY) e gli altri dati saranno persi.

Se il numero di bit da fare scorrere specificato in C è "0", i dati non saranno spostati. In ogni caso, i flag appropriati si attiveranno o disattiveranno in base ai dati nel canale specificato.

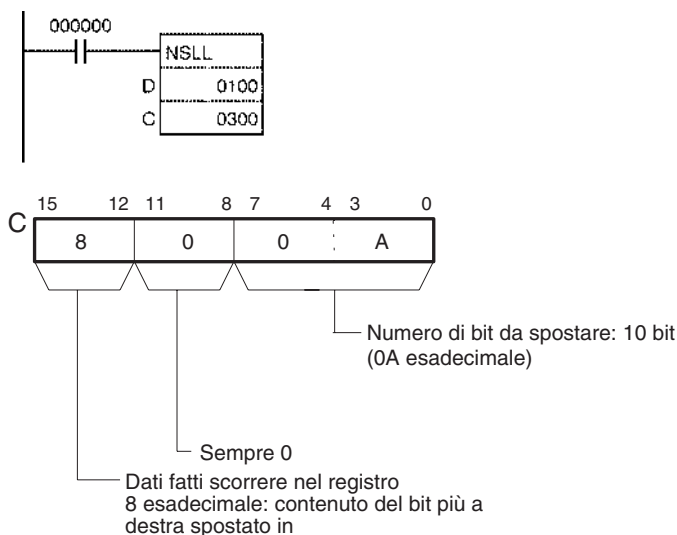
Se il contenuto del canale di controllo C non è nell'intervallo, verrà generato un errore e si attiverà il flag di errore.

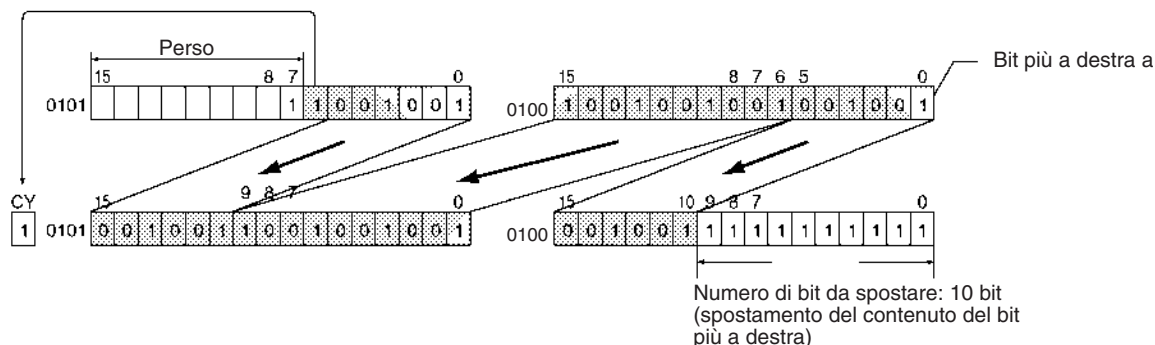
Se come risultato dello scorrimento il contenuto di D è 0000, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di D e D +1 è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, il contenuto di CIO 0100 e CIO 0101 viene fatto scorrere di 10 bit a sinistra, dal bit più a destra al bit più a sinistra. Il numero di bit da fare scorrere è specificato nei bit da 0 a 7 di CIO 0300 del canale (dati di controllo). Il contenuto del bit 0 di CIO 0100 viene copiato nei bit dai quali erano stati spostati i dati e il contenuto del bit più a destra, che era stato fatto scorrere fuori dall'intervallo, viene spostato nel flag di riporto (CY). Tutti gli altri dati vengono persi.

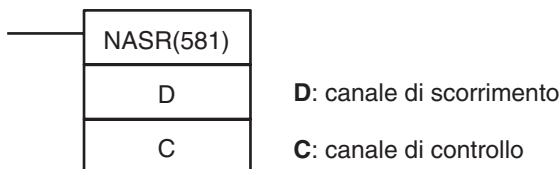




3-9-23 SHIFT N-BITS RIGHT: NASR(581)

Scopo Fa scorrere a destra i 16 bit specificati dei dati canale del numero di bit specificato.

Simbolo programmazione ladder



Variazioni

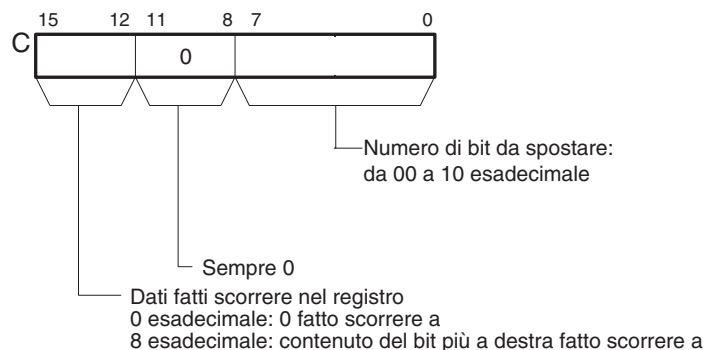
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | NASR(581) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NASR(581) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Arete di programma applicabili

| Arete di programma a blocchi | Arete di programma step | Subroutine | Task ad interrupt |
|------------------------------|-------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo



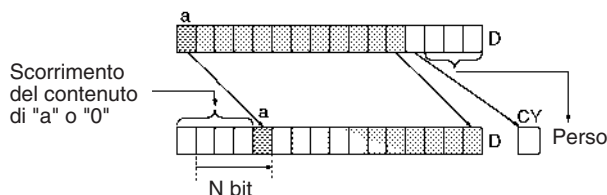
Caratteristiche operando

| Area | D | C |
|-------------------------|------------------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |

| Area | D | C |
|--|--|------------------------------------|
| Area bit ausiliaria | Da A448 ad A959 | Da A000 ad A447 Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Solo valori specificati |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

NASR(581) fa scorrere D, cioè il canale di scorrimento, del numero specificato di bit binari indicato in C verso destra, dal bit più a sinistra al bit più a destra. Gli zeri o il valore del bit più a destra vengono inviati nel numero specificato di bit del canale di scorrimento a partire dal bit più a destra.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON quando il canale di controllo C, ossia il numero di bit da fare scorrere, non è nell'intervallo. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Per ciascun bit fatto scorrere fuori dal canale specificato, il contenuto dell'ultimo bit viene fatto scorrere al flag di riporto (CY) e gli altri dati saranno scartati.

Se il numero di bit da fare scorrere specificato in C è "0", i dati non saranno spostati. In ogni caso, i flag appropriati si attiveranno o disattiveranno in base ai dati nel canale specificato.

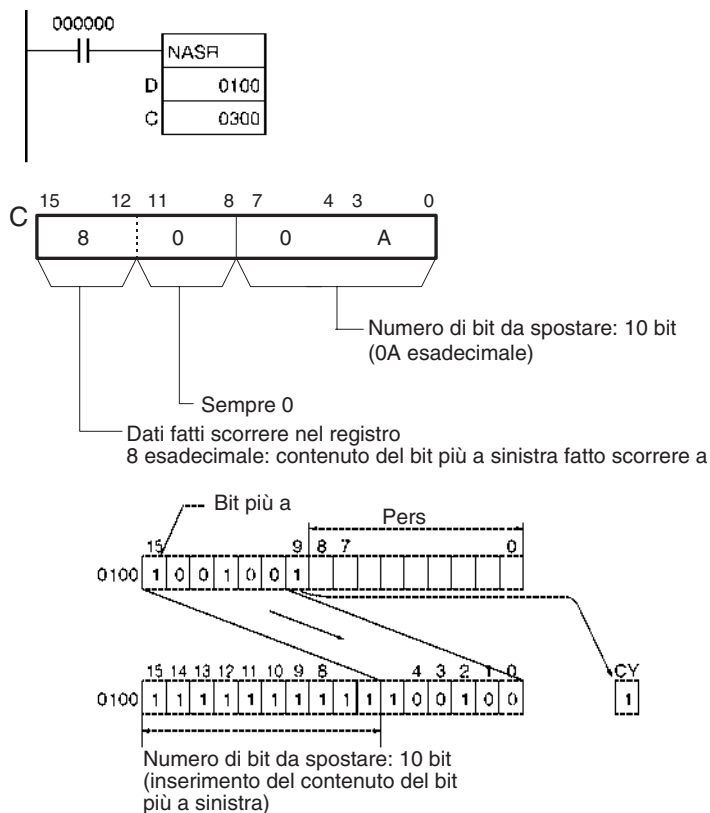
Se il contenuto del canale di controllo C non è nell'intervallo, verrà generato un errore e si attiverà il flag di errore.

Se come risultato dello scorrimento il contenuto di D è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di D è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, CIO 0100 viene fatto scorrere di 10 bit a destra, dal bit più a sinistra al bit più a destra. Il numero di bit da fare scorrere è specificato nei bit da 0 a 7 di CIO 0300 del canale. Il contenuto del bit 15 di CIO 0100 viene copiato nei bit dai quali erano stati spostati i dati e il contenuto del bit di dati più a sinistra, che era stato fatto scorrere fuori dall'intervallo, viene spostato nel flag di riporto (CY)(CY). Tutti gli altri dati vengono persi.

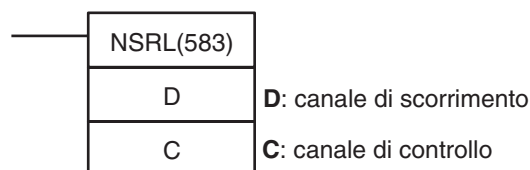


3-9-24 DOUBLE SHIFT N-BITS RIGHT: NSRL(583)

Scopo

Fa scorrere a destra i 32 bit specificati dei dati canale del numero di bit specificato.

Simbolo programmazione ladder



Variazioni

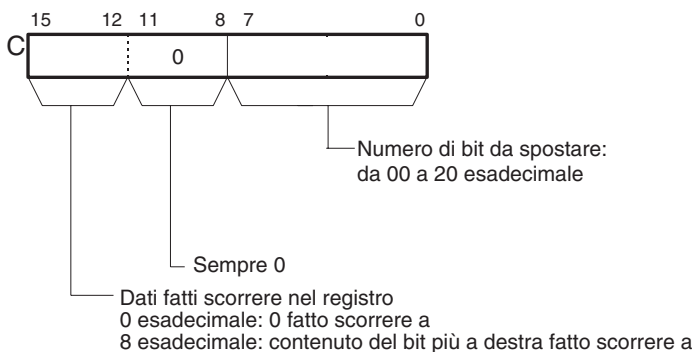
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | NSRL(583) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NSRL(583) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

C: canale di controllo

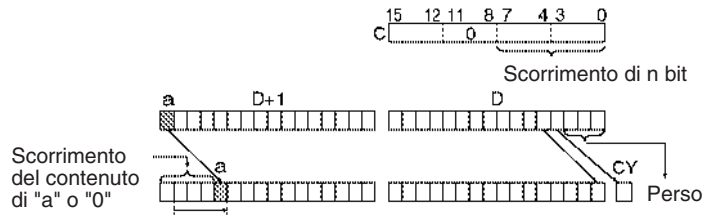


Caratteristiche operando

| Area | D | C |
|--|---|--|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A958 | Da A000 ad A959 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @E00000 a @E32767 Da @En_00000 a @En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Solo valori specificati |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a -2048 a +2047 a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) ,-(-)IR0 ... ,-(-)IR15 | |

Descrizione

NSRL(583) fa scorrere D e D + 1, cioè i canali di scorrimento, del numero specificato di bit binari indicato in C verso destra, dal bit più a sinistra al bit più a destra. Gli zeri o il valore del bit più a destra vengono inviati nel numero specificato di bit del canale di scorrimento a partire dal bit più a destra.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON quando il canale di controllo C, ossia il numero di bit da fare scorrere, non è nell'intervallo. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato dello scorrimento è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando 1 viene fatto scorrere al flag di riporto (CY). OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra è 1 come risultato dello scorrimento. OFF in tutti gli altri casi. |

Avvertenze

Per ciascun bit fatto scorrere fuori dal canale specificato, il contenuto dell'ultimo bit viene fatto scorrere al flag di riporto (CY) e gli altri dati saranno persi.

Se il numero di bit da fare scorrere specificato in C è "0", i dati non saranno spostati. In ogni caso, i flag appropriati si attiveranno e disattiveranno in base ai dati nel canale specificato.

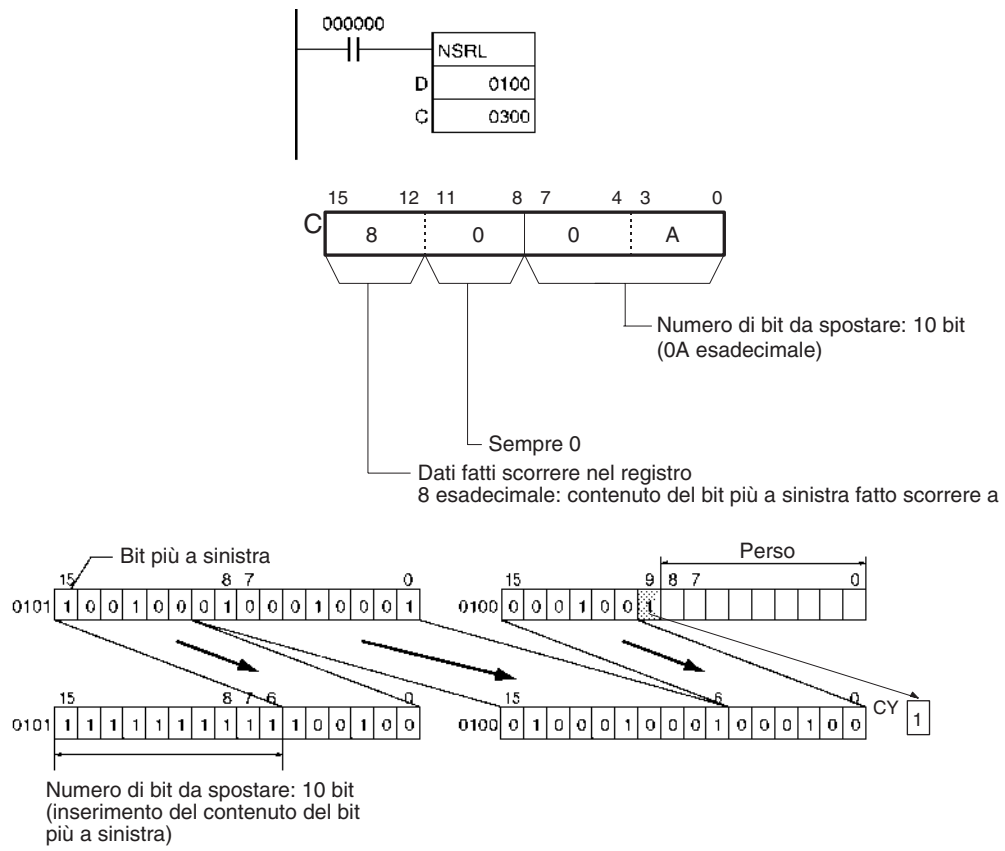
Se il contenuto del canale di controllo C non è nell'intervallo, verrà generato un errore e si attiverà il flag di errore.

Se come risultato dello scorrimento il contenuto di D +1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato dello scorrimento il contenuto del bit più a sinistra di D +1 è 1, verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, CIO 0100 e CIO 0101 vengono fatti scorrere di 10 bit a destra, dal bit più a sinistra al bit più a destra. Il numero di bit da fare scorrere è specificato nei bit da 0 a 7 di CIO 0300 del canale (dati di controllo). Il contenuto del bit 15 di CIO 0100 viene copiato nei bit dai quali erano stati spostati i dati e il contenuto del bit di dati più a sinistra, che era stato fatto scorrere fuori dall'intervallo, viene spostato nel flag di riporto (CY). Tutti gli altri dati vengono persi.

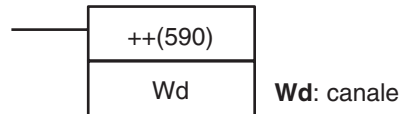


3-10 Istruzioni di incremento e decremento

3-10-1 INCREMENT BINARY: ++(590)

Scopo Incrementa di 1 il contenuto esadecimale a 4 cifre del canale specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ++(590) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @++(590) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

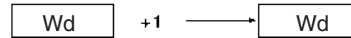
Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione ++(590) aggiunge 1 al contenuto binario di Wd. Il canale specificato verrà incrementato di 1 ad ogni ciclo finché la condizione di esecuzione di ++(590) è ON. Quando viene utilizzata la variazione

differenziata sul fronte di salita di questa istruzione (@++(590)), il canale specificato viene incrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000, il flag di riporto quando una cifra cambia da F a 0 e il flag negativo quando il bit 15 di Wd è ON nel risultato.

Verranno attivati sia il flag di uguaglianza che il flag di riporto quando il contenuto di Wd cambia da FFFF a 0000.

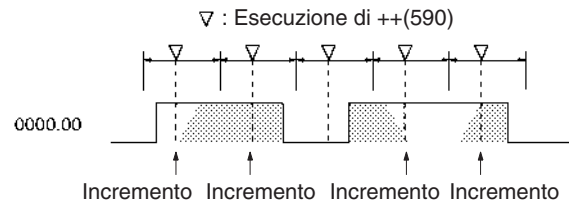
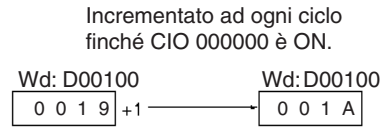
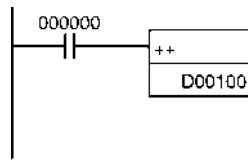
Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il contenuto di Wd è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se una cifra in Wd è passata da F a 0 durante l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di Wd è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempi

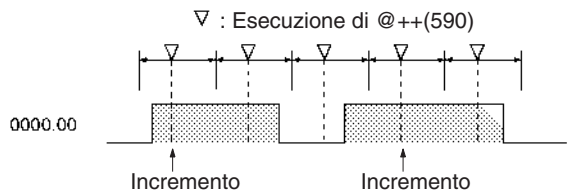
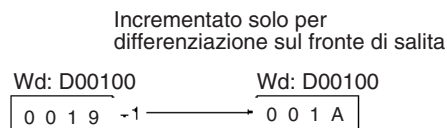
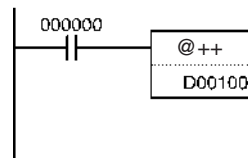
Funzionamento di ++(590)

Nell'esempio seguente, il contenuto di D00100 verrà incrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @++(590)

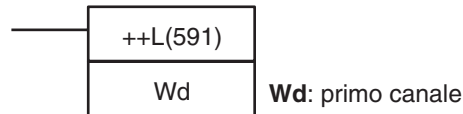
Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto di D00100 venga incrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-10-2 DOUBLE INCREMENT BINARY: ++L(591)

Scopo Incrementa di 1 il contenuto esadecimale a 8 cifre dei canali specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ++L(591) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @++L(591) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

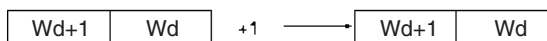
Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | Da IR0 a IR15 |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione ++L(591) aggiunge 1 al contenuto esadecimale a 8 cifre di Wd+1 e Wd. Il contenuto dei canali specificati verrà incrementato di 1 ad ogni ciclo finché la condizione di esecuzione di ++L(591) è ON. Quando viene utilizzata la variazione differenziata sul fronte di salita di questa istruzione

(@++L(591)), il contenuto dei canali specificati viene incrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000 0000, il flag di riporto quando una cifra cambia da F a 0 e il flag negativo quando il bit 15 di Wd+1 è ON nel risultato.

Verranno attivati sia il flag di uguaglianza che il flag di riporto quando il contenuto cambia da FFFF FFFF a 0000 0000.

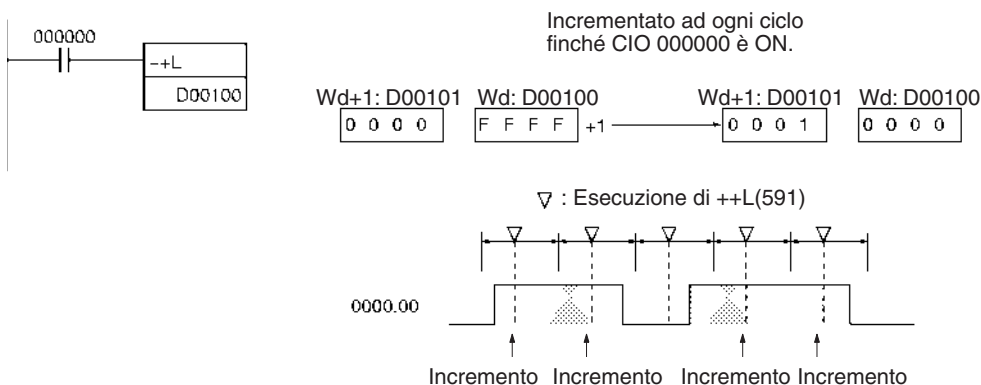
Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se una cifra in Wd+1 o Wd è passata da F a 0 durante l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di Wd+1 è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempi

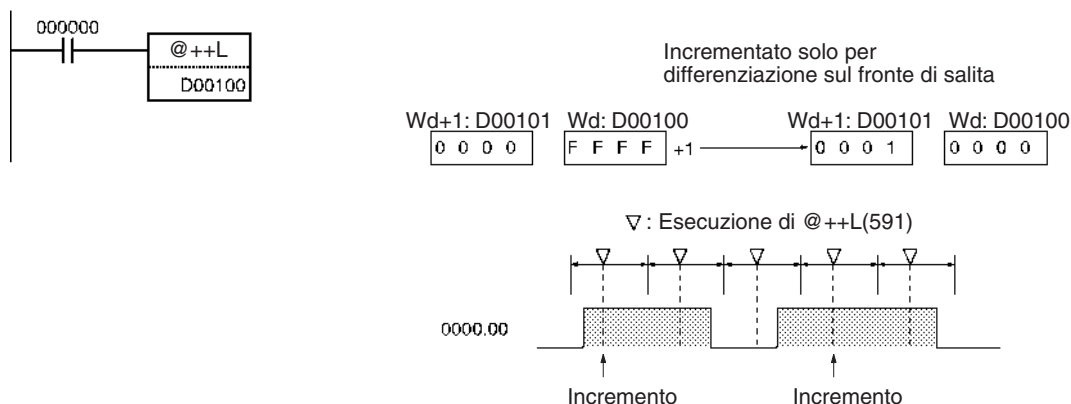
Funzionamento di ++L(591)

Nell'esempio seguente, il contenuto esadecimale a 8 cifre di D00101 e D00100 verrà incrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @++L(591)

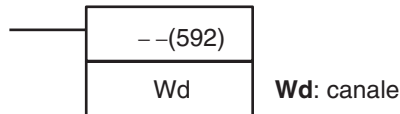
Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto di D00101 e D00100 venga incrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-10-3 DECREMENT BINARY: --(592)

Scopo Decrementa di 1 il contenuto esadecimale a 4 cifre del canale specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -- (592) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-- (592) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

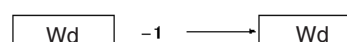
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione --(592) sottrae 1 dal contenuto binario di Wd. Il canale specificato verrà decrementato di 1 ad ogni ciclo finché la condizione di esecuzione di --(592) è ON. Quando viene utilizzata la variazione differenziata sul fronte di salita di questa istruzione (@--(592)), il canale specificato viene decrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000, il flag di riporto quando una cifra cambia da F a 0 e il flag negativo quando il bit 15 di Wd è ON nel risultato.

Verranno attivati sia il flag di riporto che il flag negativo quando il contenuto di Wd cambia da 0000 a FFFF.

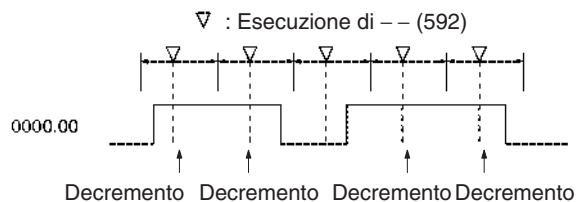
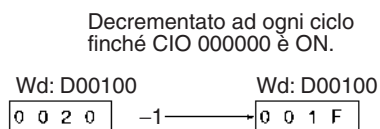
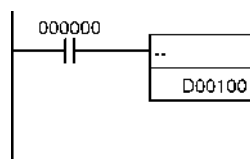
Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il contenuto di Wd è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se una cifra in Wd è passata da 0 a F durante l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di Wd è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempi

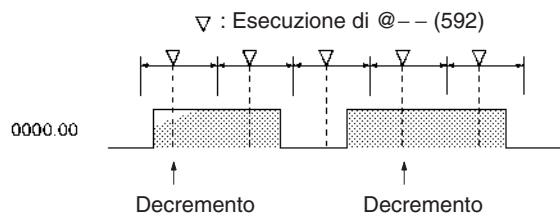
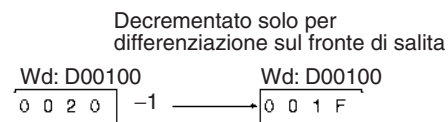
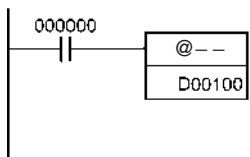
Funzionamento di --(592)

Nell'esempio seguente, il contenuto di D00100 verrà decrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @--(592)

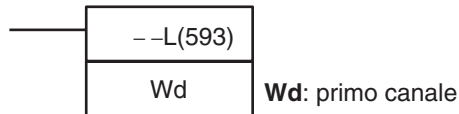
Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto di D00100 venga decrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-10-4 DOUBLE DECREMENT BINARY: --L(593)

Scopo Decrementa di 1 il contenuto esadecimale a 8 cifre dei canali specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | --L(593) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @--L(593) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

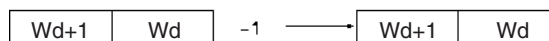
Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | Da IR0 a IR15 |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione --L(593) sottrae 1 dal contenuto esadecimale a 8 cifre di Wd+1 e Wd. Il contenuto dei canali specificati verrà decrementato di 1 ad ogni ciclo finché la condizione di esecuzione di --L(593) è ON. Quando viene utilizzata la variazione sul fronte di salita di questa istruzione (@--L(593)), il contenuto

dei canali specificati viene decrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000 0000, il flag di riporto quando una cifra cambia da 0 a F e il flag negativo quando il bit 15 di Wd+1 è ON nel risultato.

Verranno attivati sia il flag di riporto che il flag negativo quando il contenuto cambia da 0000 0000 a FFFF FFFF.

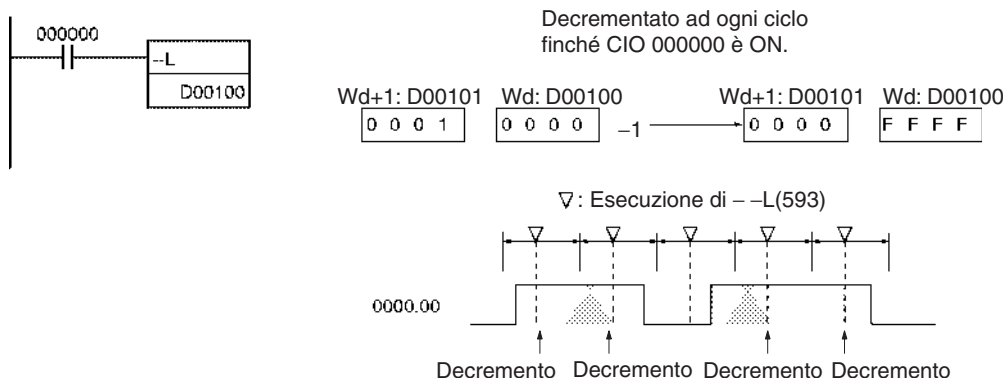
Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se una cifra in Wd+1 o Wd è passata da 0 a F durante l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di Wd+1 è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempi

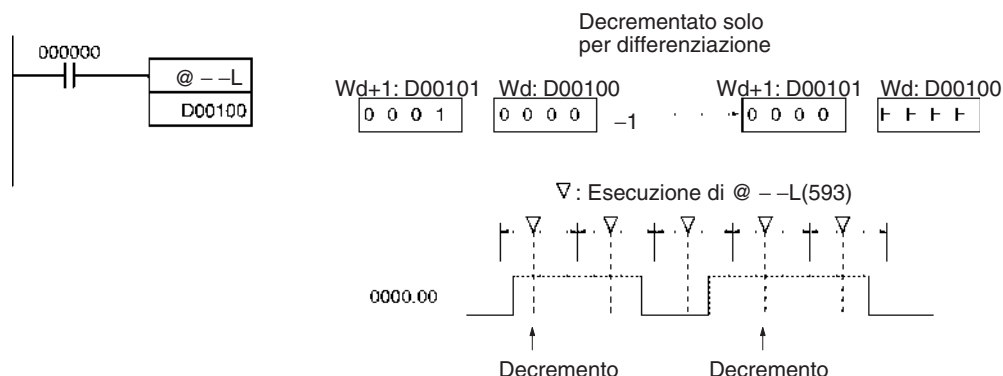
Funzionamento di --L(593)

Nell'esempio seguente, il contenuto esadecimale a 8 cifre di D00101 e D00100 verrà decrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @--L(593)

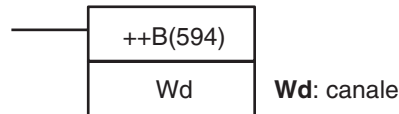
Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto di D00101 e D00100 venga decrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-10-5 INCREMENT BCD: ++B(594)

Scopo Incrementa di 1 il contenuto in formato BCD a 4 cifre del canale specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ++B(594) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @++B(594) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

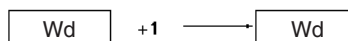
Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione ++B(594) aggiunge 1 al contenuto BCD di Wd. Il canale specificato verrà incrementato di 1 ad ogni ciclo finché la condizione di esecuzione di ++B(594) è ON. Quando viene utilizzata la variazione sul fronte di salita di questa istruzione (@++B(594)), il canale specificato viene

incrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000 e il flag di riporto quando una cifra cambia da 9 a 0.

Verranno attivati sia il flag di uguaglianza che il flag di riporto quando il contenuto cambia da 9999 a 0000.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se il contenuto di Wd non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il contenuto di Wd è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se è una cifra in Wd è passata da 9 a 0 durante l'esecuzione. OFF in tutti gli altri casi. |

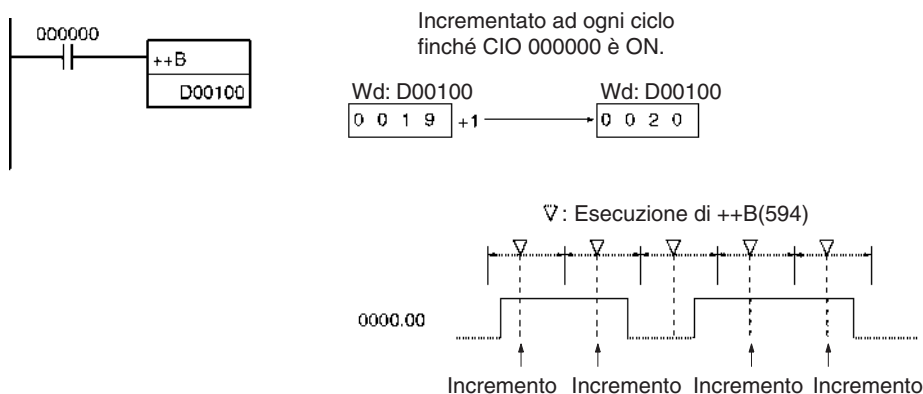
Precauzioni

Il contenuto di Wd deve essere in formato BCD. In caso contrario si verificherà un errore e verrà attivato il flag di errore.

Esempi

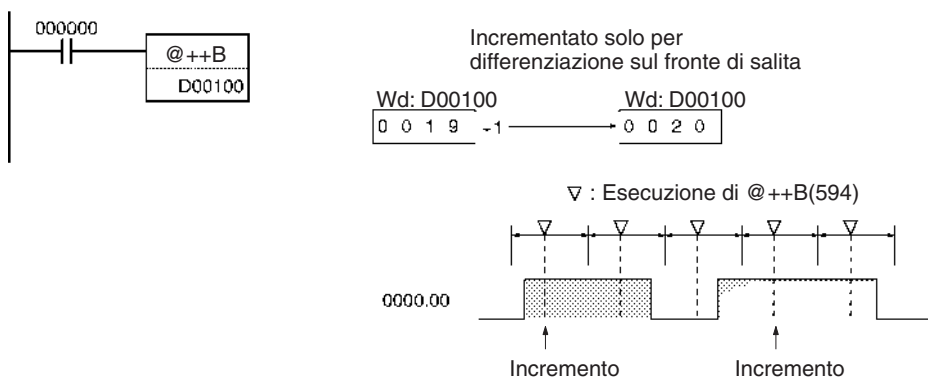
Funzionamento di ++B(594)

Nell'esempio seguente, il contenuto BCD di D00100 verrà incrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @++B(594)

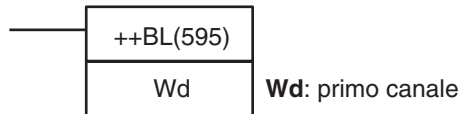
Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto di D00100 venga incrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-10-6 DOUBLE INCREMENT BCD: ++BL(595)

Scopo Incrementa di 1 il contenuto in formato BCD a 8 cifre dei canali specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ++BL(595) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @++BL(595) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

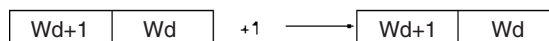
Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione ++BL(595) aggiunge 1 al contenuto BCD a 8 cifre di Wd+1 e Wd. Il contenuto dei canali specificati verrà incrementato di 1 ad ogni ciclo finché la condizione di esecuzione di ++BL(595) è ON. Quando viene utilizzata la variazione sul fronte di salita di questa istruzione (@++BL(595)), il contenuto

dei canali specificati viene incrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000 0000 e il flag di riporto quando una cifra cambia da 9 a 0.

Verranno attivati sia il flag di uguaglianza che il flag di riporto quando il contenuto cambia da 9999 9999 a 0000 0000.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di Wd+1 e Wd non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se una cifra in Wd+1 o Wd è passata da 9 a 0 durante l'esecuzione. OFF in tutti gli altri casi. |

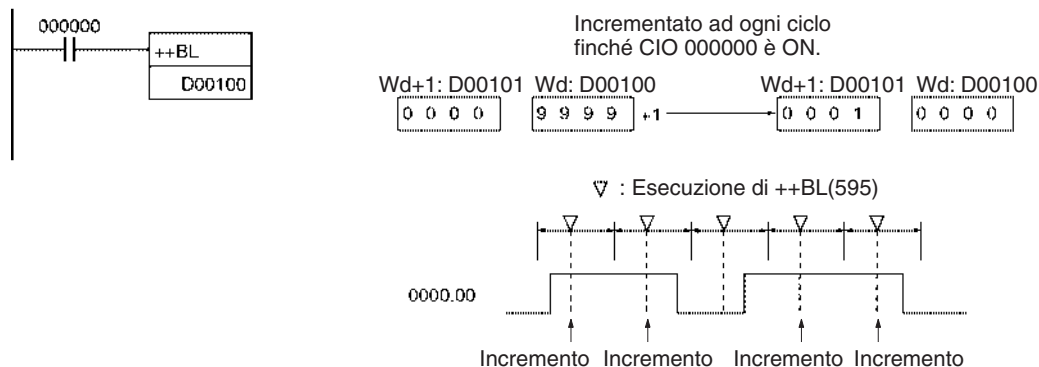
Precauzioni

Il contenuto di Wd+1 e Wd deve essere in formato BCD. In caso contrario si verificherà un errore e verrà attivato il flag di errore.

Esempi

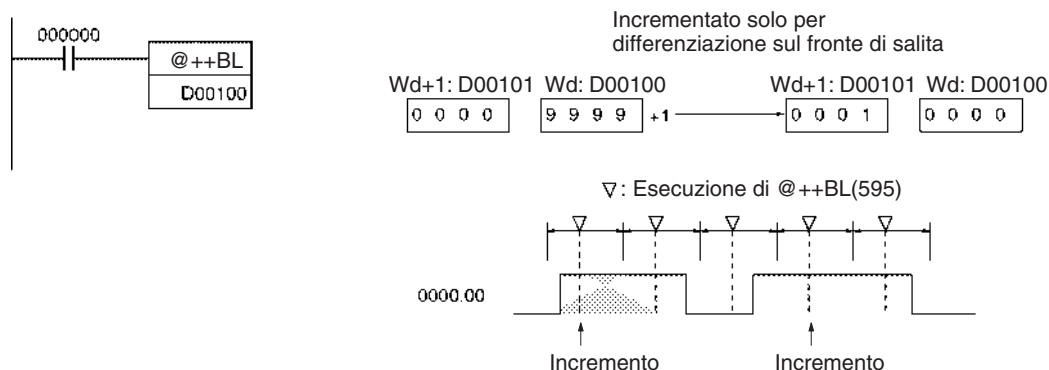
Funzionamento di ++BL(595)

Nell'esempio seguente, il contenuto BCD a 8 cifre di D00101 e D00100 verrà incrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @++BL(595)

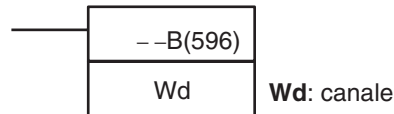
Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto BCD di D00101 e D00100 venga incrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-10-7 DECREMENT BCD: --B(596)

Scopo Decrementa di 1 il contenuto in formato BCD a 4 cifre del canale specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | --B(596) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @--B(596) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

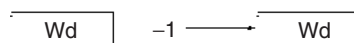
Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione --B(596) sottrae 1 dal contenuto BCD di Wd. Il canale specificato verrà decrementato di 1 ad ogni ciclo finché la condizione di esecuzione di --B(596) è ON. Quando viene utilizzata la variazione differenziata sul fronte di salita di questa istruzione (@--B(596)), il canale

specificato viene decrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000 e il flag di riporto quando una cifra cambia da 0 a 9.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se il contenuto di Wd non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il contenuto di Wd è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se una cifra in Wd è passata da 0 a 9 durante l'esecuzione. OFF in tutti gli altri casi. |

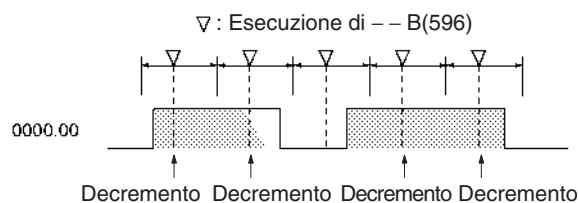
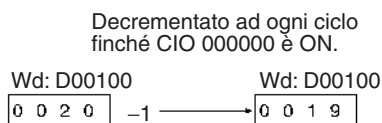
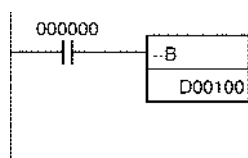
Precauzioni

Il contenuto di Wd deve essere in formato BCD. In caso contrario si verificherà un errore e verrà attivato il flag di errore.

Esempi

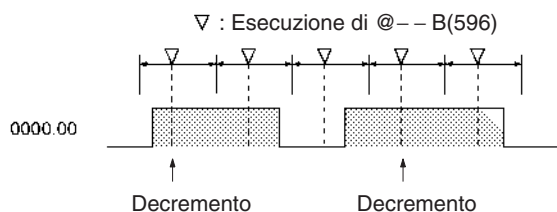
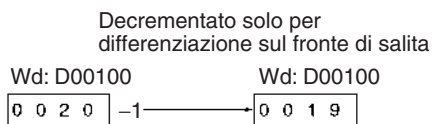
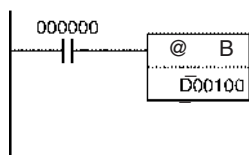
Funzionamento di --B(596)

Nell'esempio seguente, il contenuto BCD di D00100 verrà decrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @--B(596)

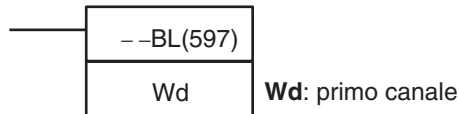
Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto BCD di D00100 venga decrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-10-8 DOUBLE DECREMENT BCD: --BL(597)

Scopo Decrementa di 1 il contenuto in formato BCD a 8 cifre dei canali specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | --BL(597) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @--BL(597) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

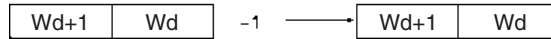
Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione --BL(597) sottrae 1 dal contenuto BCD a 8 cifre di Wd+1 e Wd. Il contenuto dei canali specificati verrà decrementato di 1 ad ogni ciclo finché la condizione di esecuzione di --BL(597) è ON. Quando viene utilizzata la variazione sul fronte di salita di questa istruzione (@--BL(597)), il contenuto

dei canali specificati viene decrementato soltanto quando la condizione di esecuzione passa da OFF a ON.



Verrà attivato (condizione ON) rispettivamente il flag di uguaglianza quando il risultato è 0000 0000 e il flag di riporto quando una cifra cambia da 0 a 9.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di Wd+1 e Wd non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se una cifra in Wd+1 o Wd è passata da 0 a 9 durante l'esecuzione. OFF in tutti gli altri casi. |

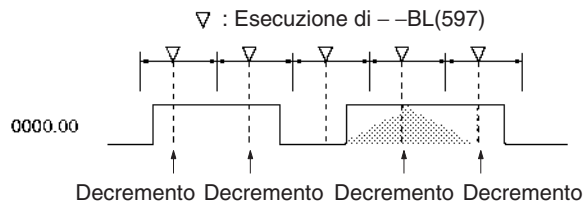
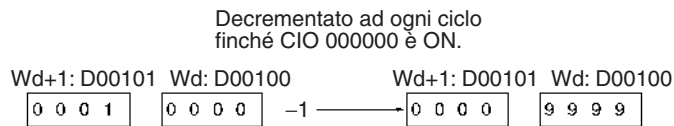
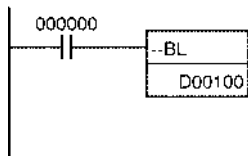
Precauzioni

Il contenuto di Wd+1 e Wd deve essere in formato BCD. In caso contrario si verificherà un errore e verrà attivato il flag di errore.

Esempi

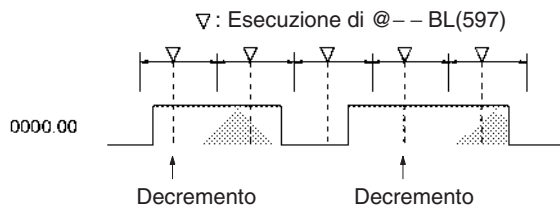
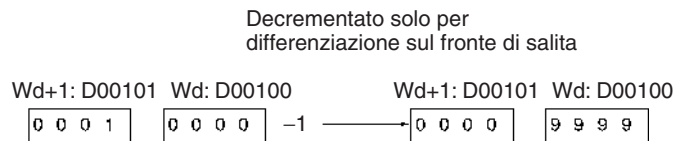
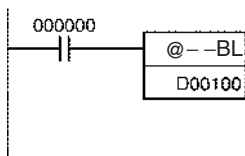
Funzionamento di --BL(597)

Nell'esempio seguente, il contenuto BCD a 8 cifre di D00101 e D00100 verrà decrementato di 1 ad ogni ciclo finché CIO 000000 è ON.



Funzionamento di @--BL(597)

Nell'esempio seguente viene utilizzata la variazione differenziata sul fronte di salita, in modo che il contenuto BCD di D00101 e D00100 venga decrementato di 1 soltanto quando CIO 000000 è passato da OFF a ON.



3-11 Istruzioni matematiche con simboli

In questa sezione vengono descritte le istruzioni matematiche con simboli, che permettono di eseguire operazioni aritmetiche su dati BCD o binari.

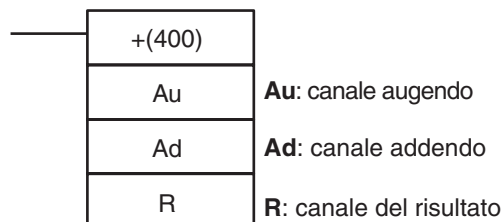
| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|---|------------------|-----------------|--------|
| SIGNED BINARY ADD WITHOUT CARRY | + | 400 | 410 |
| DOUBLE SIGNED BINARY ADD WITHOUT CARRY | +L | 401 | 412 |
| SIGNED BINARY ADD WITH CARRY | +C | 402 | 414 |
| DOUBLE SIGNED BINARY ADD WITH CARRY | +CL | 403 | 416 |
| BCD ADD WITHOUT CARRY | +B | 404 | 418 |
| DOUBLE BCD ADD WITHOUT CARRY | +BL | 405 | 419 |
| BCD ADD WITH CARRY | +BC | 406 | 421 |
| DOUBLE BCD ADD WITH CARRY | +BCL | 407 | 423 |
| SIGNED BINARY SUBTRACT WITHOUT CARRY | - | 410 | 424 |
| DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY | -L | 411 | 426 |
| SIGNED BINARY SUBTRACT WITH CARRY | -C | 412 | 430 |
| DOUBLE SIGNED BINARY SUBTRACT WITH CARRY | -CL | 413 | 432 |
| BCD SUBTRACT WITHOUT CARRY | -B | 414 | 435 |
| DOUBLE BCD SUBTRACT WITHOUT CARRY | -BL | 415 | 436 |
| BCD SUBTRACT WITH CARRY | -BC | 416 | 440 |
| DOUBLE BCD SUBTRACT WITH CARRY | -BCL | 417 | 441 |
| SIGNED BINARY MULTIPLY | * | 420 | 443 |
| DOUBLE SIGNED BINARY MULTIPLY | *L | 421 | 445 |
| UNSIGNED BINARY MULTIPLY | *U | 422 | 447 |
| DOUBLE UNSIGNED BINARY MULTIPLY | *UL | 423 | 449 |
| BCD MULTIPLY | *B | 424 | 450 |
| DOUBLE BCD MULTIPLY | *BL | 425 | 452 |
| SIGNED BINARY DIVIDE | / | 430 | 454 |
| DOUBLE SIGNED BINARY DIVIDE | /L | 431 | 456 |
| UNSIGNED BINARY DIVIDE | /U | 432 | 458 |
| DOUBLE UNSIGNED BINARY DIVIDE | /UL | 433 | 460 |
| BCD DIVIDE | /B | 434 | 462 |
| DOUBLE BCD DIVIDE | /BL | 435 | 464 |

3-11-1 SIGNED BINARY ADD WITHOUT CARRY: +(400)

Scopo

Somma dati e/o costanti esadecimali a 4 cifre (canale singolo).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +(400) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+(400) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

3-11-2 DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401)

Scopo

Somma dati e/o costanti esadecimali a 8 cifre (doppio canale).

Simbolo programmazione ladder

| | | | |
|---|---|--------|--------------------------------------|
| — | + | L(401) | |
| | | Au | Au: primo canale augendo |
| | | Ad | Ad: primo canale addendo |
| | | R | R: primo canale del risultato |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +L(401) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+L(401) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

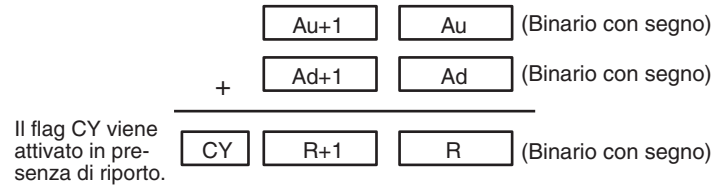
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | Da IR0 a IR15 | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

+L(401) somma i valori binari in Au e Au+1 e in Ad e Ad+1 e invia il risultato a R.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato dell'addizione produce un riporto. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON quando il risultato della somma di due numeri positivi è compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale. OFF in tutti gli altri casi. |
| flag di underflow | UF | ON quando il risultato della somma di due numeri negativi è compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Precauzioni

Quando +L(401) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'addizione il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se il risultato dell'addizione produce un riporto, verrà attivato il flag di riporto.

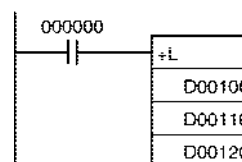
Se il risultato della somma di due numeri positivi è negativo (compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale), verrà attivato il flag di overflow.

Se il risultato della somma di due numeri negativi è positivo (compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale), verrà attivato il flag di underflow.

Se come risultato dell'addizione il contenuto del bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Quando CIO 000000 è ON, D00100 e D00101 verranno sommati a D00110 e D00111 come valori binari a 8 cifre con segno e il risultato verrà inviato a D00120 e a D00121.

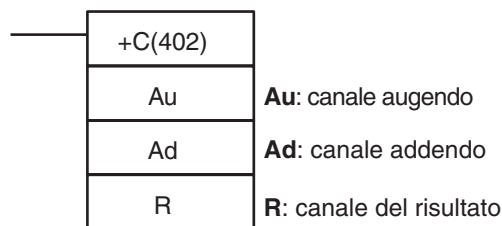


3-11-3 SIGNED BINARY ADD WITH CARRY: +C(402)

Scopo

Somma dati e/o costanti esadecimali a 4 cifre (canale singolo), incluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +C(402) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+C(402) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

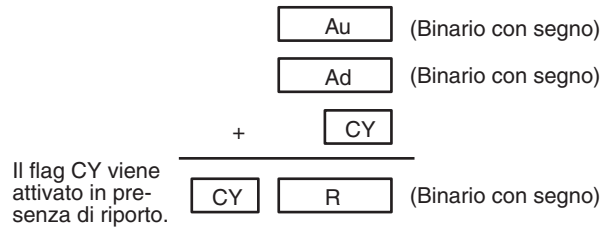
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,(- -)IR0 a ,(- -)IR15 | | |

Descrizione +C(402) somma i valori binari in Au, Ad e CY e invia il risultato a R.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato dell'addizione è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato dell'addizione produce un riporto. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON quando il risultato della somma di due numeri positivi e CY è compreso nell'intervallo da 8000 a FFFF esadecimale. OFF in tutti gli altri casi. |
| flag di underflow | UF | ON quando il risultato della somma di due numeri negativi e CY è compreso nell'intervallo da 0000 a 7FFF esadecimale. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Precauzioni

Quando +C(402) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'addizione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se il risultato dell'addizione produce un riporto, verrà attivato il flag di riporto.

Se il risultato della somma di due numeri positivi e CY è negativo (compreso nell'intervallo da 8000 a FFFF esadecimale), verrà attivato il flag di overflow.

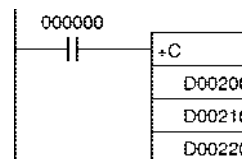
Se il risultato della somma di due numeri negativi e CY è positivo (compreso nell'intervallo da 0000 a 7FFF esadecimale), verrà attivato il flag di underflow.

Se come risultato dell'addizione il contenuto del bit più a sinistra di R è 1, verrà attivato il flag negativo.

Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

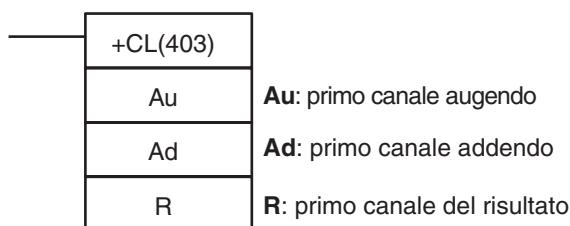
Quando CIO 000000 è ON, D00100, D00110 e CY verranno aggiunti come valori binari a 4 cifre con segno e il risultato verrà inviato a D00220.



3-11-4 DOUBLE SIGNED BINARY ADD WITH CARRY: +CL(403)

Scopo Somma dati e/o costanti esadecimali a 8 cifre (doppio canale), incluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +CL(403) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+CL(403) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

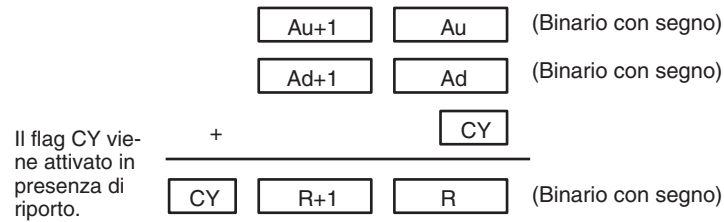
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,(- -)IR0 a ,(- -)IR15 | | |

Descrizione

+CL(403) somma i valori binari in Au e Au+1 e in Ad e Ad+1 e CY e invia il risultato a R.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato produce un riporto. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON quando il risultato della somma di due numeri positivi e CY è compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale. OFF in tutti gli altri casi. |
| flag di underflow | UF | ON quando il risultato della somma di due numeri negativi e CY è compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

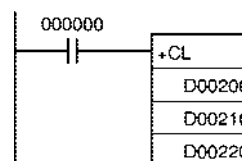
Precauzioni

Quando +CL(403) viene eseguita, il flag di errore viene disattivato.
 Se come risultato dell'addizione il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.
 Se il risultato dell'addizione produce un riporto, verrà attivato il flag di riporto.
 Se il risultato della somma di due numeri positivi e CY è negativo (compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale), verrà attivato il flag di overflow.
 Se il risultato della somma di due numeri negativi e CY è positivo (compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale), verrà attivato il flag di underflow.
 Se come risultato dell'addizione il contenuto del bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

Quando CIO 000000 è ON, D00201, D00200, D00211, D00210 e CY verranno aggiunti come valori binari a 8 cifre con segno e il risultato verrà inviato a D00221 e a D00220.



3-11-5 BCD ADD WITHOUT CARRY: +B(404)**Scopo**

Somma dati e/o costanti in formato BCD a 4 cifre (canale singolo).

Simbolo programmazione ladder

| | | |
|---|---------|--------------------------------|
| — | +B(404) | |
| | Au | Au: canale augendo |
| | Ad | Ad: canale addendo |
| | R | R: canale del risultato |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +B(404) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ +B(404) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

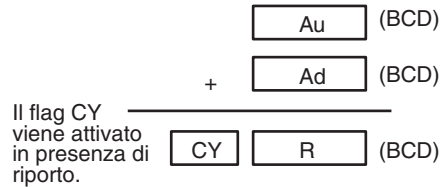
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da 0000 a 9999 (BCD) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione +B(404) somma i valori in formato BCD in Au e Ad e invia il risultato a R.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando Au non è in formato BCD. ON quando Ad non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato dell'addizione produce un riporto. OFF in tutti gli altri casi. |

Precauzioni

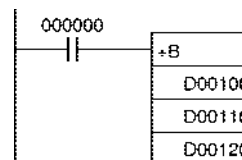
Se Au o Ad non è in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato dell'addizione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se un'addizione produce come risultato un riporto, verrà attivato il flag di riporto.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100 e D00110 verranno aggiunti come valori BCD a 4 cifre e il risultato verrà inviato a D00120.

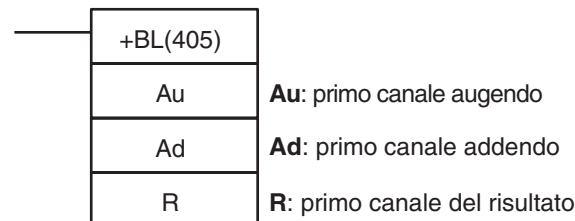


3-11-6 DOUBLE BCD ADD WITHOUT CARRY: +BL(405)

Scopo

Somma dati e/o costanti in formato BCD a 8 cifre (doppio canale).

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | +BL(405) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+BL(405) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

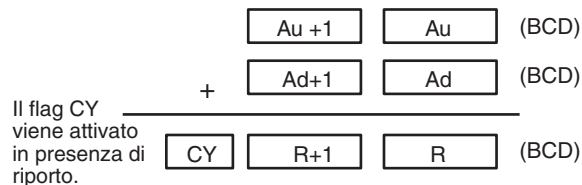
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #99999999 (BCD) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

+BL(405) somma i valori BCD in Au e Au+1 e in Ad e Ad+1 invia il risultato a R, R+1.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando il contenuto di Au, Au +1 non è in formato BCD. ON quando il contenuto di Ad, Ad +1 non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato dell'addizione produce un riporto. OFF in tutti gli altri casi. |

Avvertenze

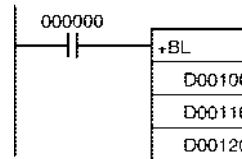
Se Au, Au +1 o Ad, Ad +1 non sono in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato dell'addizione il contenuto di R, R +1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se un'addizione produce come risultato un riporto, verrà attivato il flag di riporto.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00101, D00100, D00111 e D110 verranno aggiunti come valori BCD a 8 cifre e il risultato verrà inviato a D00121 e a D00120.

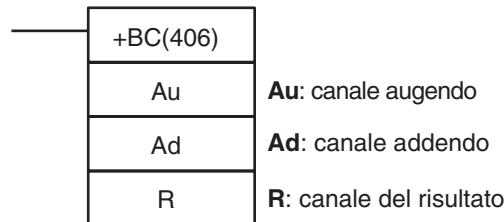


3-11-7 BCD ADD WITH CARRY: +BC(406)

Scopo

Somma dati e/o costanti in formato BCD a 4 cifre (canale singolo), incluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +BC(406) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+BC(406) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

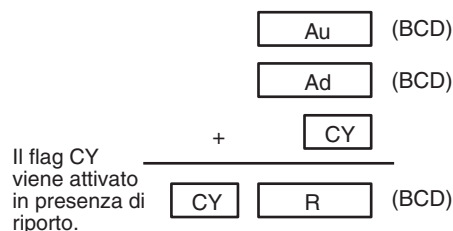
Caratteristiche operando

| Area | Au | Ad | R |
|-------------------------|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |

| Area | Au | Ad | R |
|--|--|----|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a 9999 (BCD) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

+BC(406) somma i valori BCD in Au, Ad e CY e invia il risultato a R.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando Au non è in formato BCD. ON quando Ad non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato dell'addizione produce un riporto. OFF in tutti gli altri casi. |

Avvertenze

Se Au o Ad non è in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

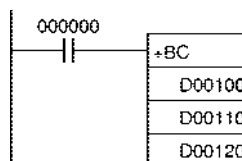
Se come risultato dell'addizione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se un'addizione produce come risultato un riporto, verrà attivato il flag di riporto.

Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100 e D00110 e CY verranno aggiunti come valori BCD a 4 cifre e il risultato verrà inviato a D00120.



3-11-8 DOUBLE BCD ADD WITH CARRY: +BCL(407)

Scopo

Somma dati e/o costanti in formato BCD a 8 cifre (doppio canale), incluso il flag di riporto (CY).

Simbolo programmazione ladder

| | |
|-----------|--------------------------------------|
| +BCL(407) | |
| Au | Au: primo canale augendo |
| Ad | Ad: primo canale addendo |
| R | R: primo canale del risultato |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +BCL(407) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+BCL(407) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

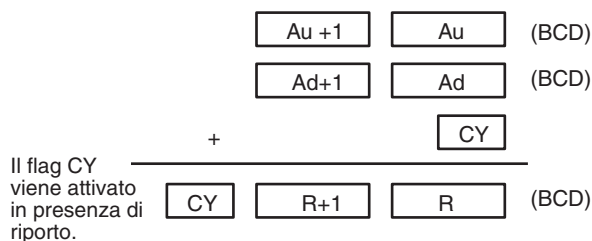
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #99999999 (BCD) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione +BCL(407) somma i valori BCD in Au e Au+1 e in Ad e Ad+1 e CY e invia il risultato a R,R+1.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando il contenuto di Au, Au +1 non è in formato BCD. ON quando il contenuto di Ad, Ad +1 non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato dell'addizione produce un riporto. OFF in tutti gli altri casi. |

Avvertenze

Se Au, Au +1 o Ad, Ad +1 non sono in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

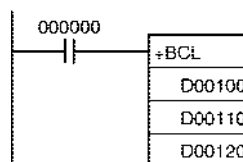
Se come risultato dell'addizione il contenuto di R, R +1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se un'addizione produce come risultato un riporto, verrà attivato il flag di riporto.

Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00101, D00100, D00111, D110 e CY verranno aggiunti come valori BCD a 8 cifre e il risultato verrà inviato a D00121 e a D00120.

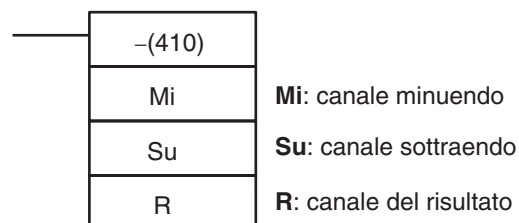


3-11-9 SIGNED BINARY SUBTRACT WITHOUT CARRY: -(410)

Scopo

Esegue una sottrazione tra dati e/o costanti esadecimali a 4 cifre (canale singolo).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -(410) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-(410) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Mi | Su | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D0000 a D4095 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

-(400) sottrae i valori binari in Su da Mi e invia il risultato a R. Quando il risultato è negativo, viene inviato a R come complemento a 2. Per ulteriori informazioni, consultare la sezione 3-11-10 *DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: -L(411)* nella quale è fornito un esempio sulla gestione dei complementi a 2.

(Binario con segno)

(Binario con segno)

Il flag CY viene attivato in presenza di prestito.

(Binario con segno)

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato della sottrazione produce un prestito. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON quando il risultato della sottrazione di un numero negativo da un numero positivo è compreso nell'intervallo da 8000 a FFFF esadecimale. OFF in tutti gli altri casi. |
| flag di underflow | UF | ON quando il risultato della sottrazione di un numero negativo da un numero positivo è compreso nell'intervallo da 0000 a 7FFF esadecimale. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando $-(410)$ viene eseguita, il flag di errore viene disattivato.

Se come risultato della sottrazione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se il risultato della sottrazione produce un prestito, verrà attivato il flag di riporto.

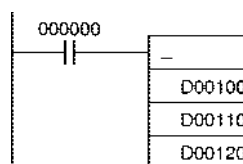
Se la sottrazione di un numero negativo da un numero positivo produce un risultato negativo (compreso nell'intervallo da 8000 a FFFF esadecimale), verrà attivato il flag di overflow.

Se la sottrazione di un numero positivo da un numero negativo produce un risultato positivo (compreso nell'intervallo da 0000 a 7FFF esadecimale), verrà attivato il flag di underflow.

Se come risultato della sottrazione il contenuto del bit più a sinistra di R è 1, verrà attivato il flag negativo.

Esempi

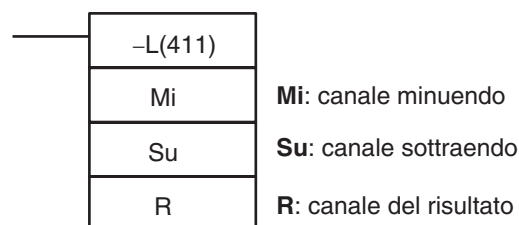
Quando nell'esempio seguente CIO 000000 è ON, D00110 verrà sottratto da D00100 come valore binario a 4 cifre con segno e il risultato verrà inviato a D00120.

3-11-10 DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: $-L(411)$

Scopo

Esegue una sottrazione tra dati e/o costanti esadecimali a 8 cifre (doppio canale).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -L(411) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-L(411) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

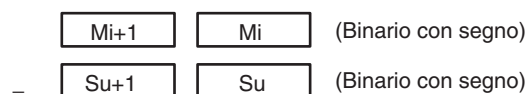
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Mi | Su | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | Da IR0 a IR15 | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

-L(411) sottrae i valori binari in Su e Su+1 da Mi e Mi+1 e invia il risultato a R, R+1. Quando il risultato è negativo, viene inviato a R e a R+1 come complemento a 2.



Il flag CY viene attivato in presenza di prestito.

| | | | |
|----|-----|---|---------------------|
| CY | R+1 | R | (Binario con segno) |
|----|-----|---|---------------------|

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato della sottrazione produce un prestito. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON quando il risultato della sottrazione di un numero negativo da un numero positivo è compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale. OFF in tutti gli altri casi. |
| flag di underflow | UF | ON quando il risultato della sottrazione di un numero positivo da un numero negativo è compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando $-L(411)$ viene eseguita, il flag di errore viene disattivato.

Se come risultato della sottrazione il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se il risultato della sottrazione produce un prestito, verrà attivato il flag di riporto.

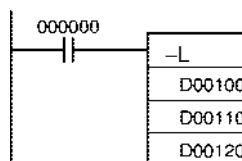
Se la sottrazione di un numero negativo da un numero positivo produce un risultato negativo (compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale), verrà attivato il flag di overflow.

Se la sottrazione di un numero positivo da un numero negativo produce un risultato positivo (compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale), verrà attivato il flag di underflow.

Se come risultato della sottrazione il contenuto del bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00111 e D00110 verranno sottratti da D00101 e D00100 come valori binari a 8 cifre con segno e il risultato verrà inviato a D00121 e a D00120.



Esempi

Se il risultato della sottrazione è un numero negativo ($M_i < S_u$ o $M_{i+1}, M_i < S_{u+1}, S_u$), il risultato verrà inviato come complemento a 2 e verrà attivato il flag di riporto (CY), ad indicare che il risultato della sottrazione è un valore negativo. Per convertire il complemento a 2 in un numero effettivo, è necessaria un'istruzione che sottragga il risultato da 0 utilizzando il flag di riporto (CY) come condizione di esecuzione.

Nota Complemento a 2

Un complemento a 2 è il valore ottenuto sottraendo ciascuna cifra binaria da 1 e aggiungendo uno al risultato. Ad esempio, il complemento a 2 di 1101 viene calcolato nel modo seguente: 1111 (F esadecimale) - 1101 (D esadecimale) + 1 (1 esadecimale) = 0011 (3 esadecimale). Il complemento a 2 di 3039 (esadecimale) viene calcolato nel modo seguente: FFFF (esadecimale) - 3039 (esadecimale) + 0001 (esadecimale) - CFC7 (esadecimale). Pertanto, nel caso di un valore esadecimale a 4 cifre, il complemento a 2 può essere calcolato nel modo seguente: FFFF (esadecimale) - a (esadecimale) + 0001 (esadecimale) = b (esadecimale). Per ottenere il numero effettivo dal complemento a 2 b (esadecimale): a (esadecimale) = 10000 (esadecimale) - b (esadecimale). Ad esempio, per ottenere il numero effettivo dal complemento a 2 CFC7 (esadecimale): 10000 (esadecimale) - CFC7 = 3039.

Esempio 1 Dati con segno Dati senza segno

| | | | |
|---------------------|---|-----------|--------------|
| FFFF esadecimale | | | |
| → | | | |
| 0001 esadecimale | → | -1 | 65535 |
| -) FFFF esadecimale | → | -) +1 | -) 1 |
| → | | | |
| FFFE esadecimale | | -2 Nota 1 | 65534 Nota 2 |
| Flag negativo ON | | | |
| Flag di riporto OFF | | | |

- Nota**
- Poiché il flag negativo è ON, il risultato (FFFE esadecimale) è un valore negativo (complemento a 2) e perciò è -2.
 - Poiché il flag di riporto è OFF, il risultato (FFFE esadecimale) è un valore positivo senza segno di 65534.

Esempio 2 Dati con segno Dati senza segno

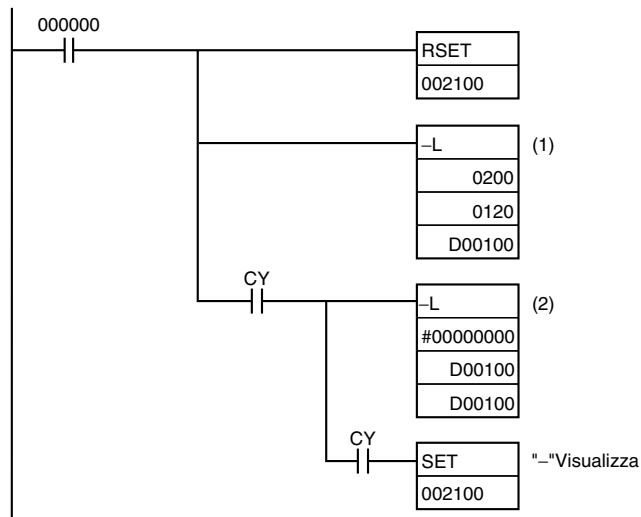
| | | | |
|---------------------|---|-----------|--------------|
| FFFD esadecimale | | | |
| → | | | |
| FFFF esadecimale | → | -3 | 65533 |
| -) FFFF esadecimale | → | -) -1 | -) 65535 |
| → | | | |
| FFFE esadecimale | | -2 Nota 3 | 65534 Nota 4 |
| Flag negativo ON | | | |
| Flag di riporto OFF | | | |

- Poiché il flag negativo è ON, il risultato (FFFE esadecimale) è un valore negativo (complemento a 2) e perciò è -2.
- Poiché il flag di riporto è ON, il risultato (FFFE esadecimale) è un valore negativo (complemento a 2) e diventa -2 quando viene convertito in un valore effettivo.

Esempio di programma

$$20F55A10 - B8A360E3 = -97AE06D3.$$

In questo esempio il valore binario a otto cifre in CIO 0121 e CIO 0120 viene sottratto dal valore in CIO 0201 e CIO 0200 e il risultato viene inviato a D00101 e a D00100 come valore binario a otto cifre. Se il risultato è negativo, verrà eseguita l'istruzione a (2) e il risultato effettivo verrà quindi inviato a D00101 e a D00100.



Sottrazione a 1

| | |
|----------------|----------------------------|
| Mi+1: CIO 0201 | Mi: CIO 0200 |
| 2 0 F 5 | 5 A 1 0 |
| Su+1: CIO 0121 | Su: CIO 0120 |
| - B 8 A 3 | 6 0 E 3 |
| <hr/> | |
| CY | R+1: D00101 R+1: D00100 |
| 1 | 6 8 5 1 F 9 2 D |

Il flag di riporto (CY) è ON, quindi il risultato viene sottratto da 0000 0000 per ottenere il numero effettivo.

Sottrazione a 2

| | | | | | | | | |
|--------------|-------------|-------------|---|-------------|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Su+1: D00101 | | | | Su: D00100 | | | | |
| - | 6 | 8 | 5 | 1 | F | 9 | 2 | D |
| | | | | | | | | |
| CY | R+1: D00101 | R+1: D00100 | | R+1: D00100 | | | | |
| 1 | 9 | 7 | A | E | 0 | 6 | D | 3 |

Risultato finale sottrazione

| | | | | | | | | |
|----------------|-------------|-------------|---|--------------|---|---|---|---|
| Mi+1: CIO 0201 | | | | Mi: CIO 0200 | | | | |
| 2 | 0 | F | 5 | 5 | A | 1 | 0 | |
| Su+1: D00101 | | | | Su: D00100 | | | | |
| - | 6 | 8 | 5 | 1 | F | 9 | 2 | D |
| | | | | | | | | |
| CY | R+1: D00101 | R+1: D00100 | | R+1: D00100 | | | | |
| 1 | 9 | 7 | A | E | 0 | 6 | D | 3 |

Il flag di riporto (CY) è ON, quindi il numero effettivo è -97AE06D3. Poiché il contenuto di D00101 e D00100 è negativo, CY viene usato per attivare CIO 002100 ad indicare questa situazione.

3-11-11 SIGNED BINARY SUBTRACT WITH CARRY: -C(412)**Scopo**

Esegue una sottrazione tra dati e/o costanti esadecimali a 4 cifre (canale singolo), incluso il flag di riporto (CY).

Simbolo programmazione ladder

| | |
|---------|--------------------------------|
| -C(412) | |
| Mi | Mi: canale minuendo |
| Su | Su: canale sottraendo |
| R | R: canale del risultato |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -C(412) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-C(412) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

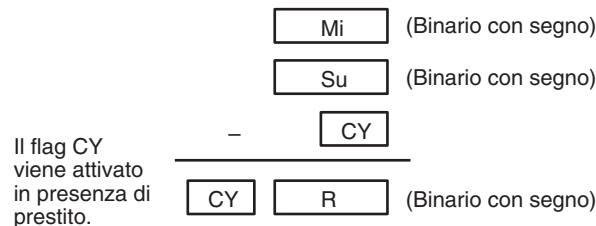
Caratteristiche operando

| | | | |
|-------------------------|------------------------|-----------|-----------------|
| Area | Mi | Su | R |
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |

| Area | Mi | Su | R |
|--|--|----|-----|
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

-C(412) sottrae i valori binari in Su e CY da Mi e invia il risultato a R. Quando il risultato è negativo, viene inviato a R come complemento a 2.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato della sottrazione è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato della sottrazione produce un prestito. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON quando il risultato della sottrazione di un numero negativo e CY da un numero positivo è compreso nell'intervallo da 8000 a FFFF esadecimale. OFF in tutti gli altri casi. |
| flag di underflow | UF | ON quando il risultato della sottrazione di un numero positivo e CY da un numero negativo è compreso nell'intervallo da 0000 a 7FFF esadecimale. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando $-CL(412)$ viene eseguita, il flag di errore viene disattivato.

Se come risultato della sottrazione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se il risultato della sottrazione produce un prestito, verrà attivato il flag di riporto.

Se la sottrazione di un numero negativo e di CY da un numero positivo produce un risultato negativo (compreso nell'intervallo da 8000 a FFFF esadecimale), verrà attivato il flag di overflow.

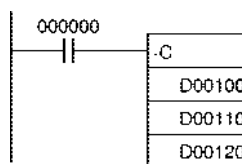
Se la sottrazione di un numero positivo e di CY da un numero negativo produce un risultato positivo (compreso nell'intervallo da 0000 a 7FFF esadecimale), verrà attivato il flag di underflow.

Se come risultato della sottrazione il contenuto del bit più a sinistra di R è 1, verrà attivato il flag negativo.

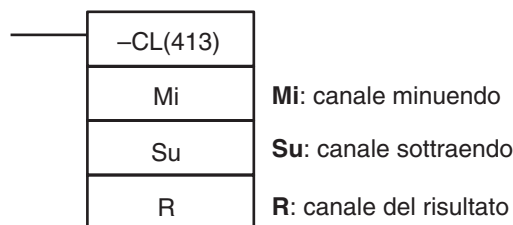
Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00110 e CY verrà sottratto da D00100 come valore binario a 4 cifre con segno e il risultato verrà inviato a D00120.

**3-11-12 DOUBLE SIGNED BINARY SUBTRACT WITH CARRY: $-CL(413)$** **Scopo**

Esegue una sottrazione tra dati e/o costanti esadecimali a 8 cifre (doppio canale), incluso il flag di riporto (CY).

Simbolo programmazione ladder**Variazioni**

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -CL(413) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-CL(413) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

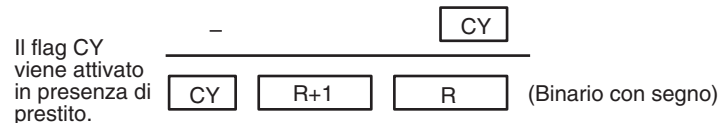
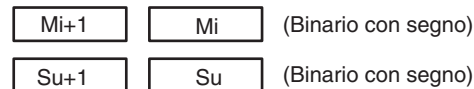
Caratteristiche operando

| Area | Mi | Su | R |
|-------------------------|------------------------|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |

| Area | Mi | Su | R |
|--|--|----|-----|
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

-CL(413) sottrae i valori binari in Su e Su+1 e CY da Mi e Mi+1 e invia il risultato a R, R+1. Quando il risultato è negativo, viene inviato a R e a R+1 come complemento a 2.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato produce un prestito. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON quando il risultato della sottrazione di un numero negativo e CY da un numero positivo è compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale. OFF in tutti gli altri casi. |
| flag di underflow | UF | ON quando il risultato della sottrazione di un numero positivo e di CY da un numero negativo è compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando –CL(413) viene eseguita, il flag di errore viene disattivato.

Se come risultato della sottrazione il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se il risultato della sottrazione produce un prestito, verrà attivato il flag di riporto.

Se la sottrazione di un numero negativo e di CY da un numero positivo produce un risultato negativo (compreso nell'intervallo da 80000000 a FFFFFFFF esadecimale), verrà attivato il flag di overflow.

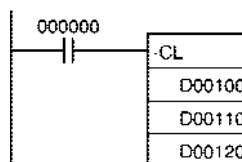
Se la sottrazione di un numero positivo e di CY da un numero negativo produce un risultato positivo (compreso nell'intervallo da 00000000 a 7FFFFFFF esadecimale), verrà attivato il flag di underflow.

Se come risultato della sottrazione il contenuto del bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00111 e D00110 e CY verranno sottratti da D00101 e D00100 come valori binari a 8 cifre con segno e il risultato verrà inviato a D00121 e a D00120.



Se il risultato della sottrazione è un numero negativo ($M_i < S_u$ o M_{i+1} , $M_i < S_{u+1}$, S_u), il risultato verrà inviato come complemento a 2 e verrà attivato il flag di riporto (CY). Per convertire il complemento a 2 in un numero effettivo, è necessario utilizzare un programma che sottragga il risultato da 0, come condizione di ingresso del flag di riporto (CY). Quando il flag di riporto passa allo stato ON significa quindi che il risultato della sottrazione è negativo.

Nota Complemento a 2

Un complemento a 2 è il valore ottenuto sottraendo ciascuna cifra binaria da 1 e aggiungendo uno al risultato.

Esempio: il complemento a 2 del numero binario 1101 è il seguente:

1111 (F esadecimale) – 1101 (D esadecimale) + 1 (1 esadecimale) = 0011 (3 esadecimale).

Esempio: il complemento a 2 del numero esadecimale a 4 cifre è il seguente:

$FFFF$ esadecimale – 3039 esadecimale + 0001 esadecimale = $CFC7$ esadecimale.

Di conseguenza, il complemento a 2 del numero esadecimale a 4 cifre "a" è il seguente:

$FFFF$ esadecimale – a esadecimale + 0001 esadecimale = b esadecimale.

Per ottenere il numero effettivo "a" esadecimale dal complemento a 2 "b" esadecimale:

a esadecimale + 10000 esadecimale – b esadecimale.

Esempio: per ottenere il numero effettivo dal complemento a 2 CFC7 esadecimale:

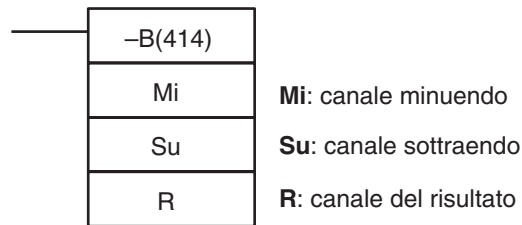
10000 esadecimale – $CFC7$ esadecimale = 3039 esadecimale.

3-11-13 BCD SUBTRACT WITHOUT CARRY: -B(414)

Scopo

Esegue una sottrazione tra dati e/o costanti in formato BCD a 4 cifre (canale singolo).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -B(414) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-B(414) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

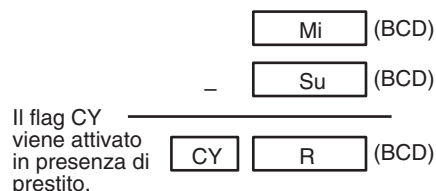
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Mi | Su | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da 0000 a 9999 (BCD) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

–B(414) sottrae i valori BCD in Su da Mi e invia il risultato a R. Se il risultato della sottrazione è negativo, il risultato viene inviato come complemento a 10.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando Mi non è in formato BCD. ON quando Su non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato della sottrazione produce un prestito. OFF in tutti gli altri casi. |

Avvertenze

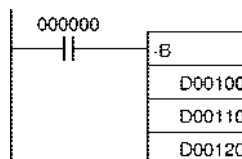
Se il contenuto di Mi e/o Su non è in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato della sottrazione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

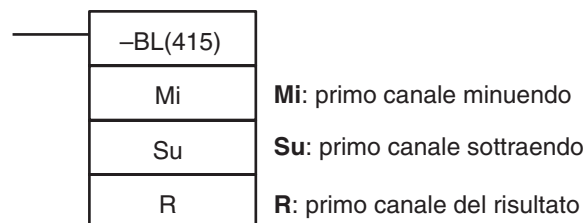
Se un'addizione produce come risultato un prestito, verrà attivato il flag di riporto.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100 viene sottratto da D00100 come valore a 4 cifre in formato BCD e il risultato verrà inviato a D00120.

**3-11-14 DOUBLE BCD SUBTRACT WITHOUT CARRY: –BL(415)****Scopo**

Esegue una sottrazione tra dati e/o costanti in formato BCD a 8 cifre (doppio canale).

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | –BL(415) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @–BL(415) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Mi | Su | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritenività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #99999999 (BCD) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

–BL(415) sottrae i valori BCD in Su e Su+1 da Mi e Mi+1 e invia il risultato a R, R+1. Se il risultato è negativo, viene inviato a R e a R+1 come complemento a 10.

$$\begin{array}{r} \boxed{Mi+1} \quad \boxed{Mi} \quad (\text{BCD}) \\ - \quad \boxed{Su+1} \quad \boxed{Su} \quad (\text{BCD}) \end{array}$$

Il flag CY viene attivato in presenza di prestito.

$$\boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{BCD})$$

Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|--|
| Flag di errore | ER | ON quando Mi e/o Mi +1 non sono in formato BCD. ON quando Su e/o Su +1 non sono in formato BCD. OFF in tutti gli altri casi. |

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato della sottrazione produce un prestito. OFF in tutti gli altri casi. |

Avvertenze

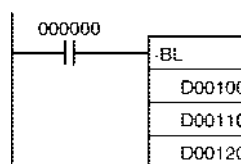
Se Mi, Mi +1 e/o Su, Su +1 non sono in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato della sottrazione il contenuto di R, R +1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se un'addizione produce come risultato un prestito, verrà attivato il flag di riporto.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00111 e D00110 verranno sottratti da D00101 e D00100 come valori BCD a 8 cifre e il risultato verrà inviato a D00121 e a D00120.



Se il risultato della sottrazione è un numero negativo (Mi < Su o Mi+1, Mi < Su+1, Su), il risultato verrà inviato come complemento a 10 e verrà attivato il flag di riporto (CY). Per convertire il complemento a 10 in un numero effettivo, è necessario utilizzare un programma che sottragga il risultato da 0, come condizione di ingresso del flag di riporto (CY). Quando il flag di riporto passa allo stato ON significa quindi che il risultato della sottrazione è negativo.

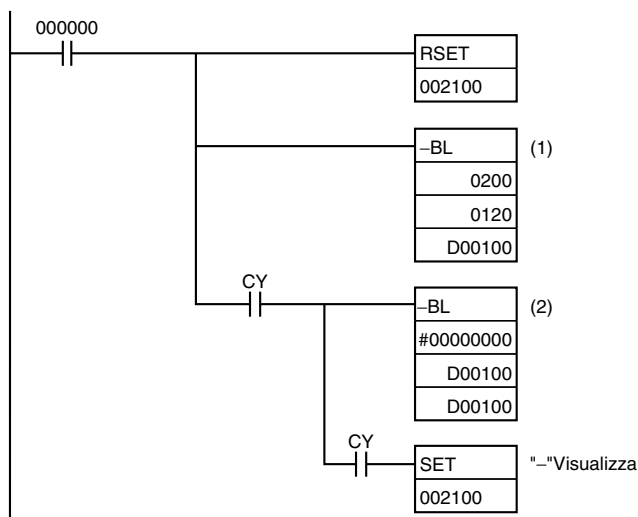
Nota Complemento a 10

Un complemento a 10 è il valore ottenuto sottraendo ciascuna cifra binaria da 9 e aggiungendo uno al risultato. Ad esempio, il complemento a 10 di 7556 viene calcolato nel modo seguente: $9999 - 7556 + 1 = 2444$. Per un numero a quattro cifre, il complemento a 10 di A è $9999 - A + 1 = B$. Per ottenere il numero effettivo dal complemento a 10 B: $A = 10000 - B$. Ad esempio, per ottenere il numero effettivo dal complemento a 10 2444: $10000 - 2444 = 7556$.

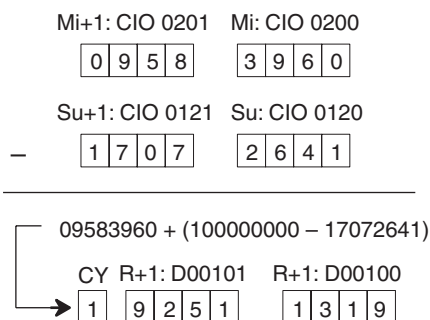
Esempio di programma

$9.583.960 - 17.072.641 = -7.488.681$.

In questo esempio il contenuto BCD a otto cifre in CIO 0121 e CIO 0120 viene sottratto dal contenuto di CIO 0201 e CIO 0200 e il risultato viene inviato a D00101 e a D00100 come valore BCD a otto cifre. Il risultato è negativo, quindi verrà eseguita l'istruzione a (2) e il valore effettivo verrà quindi inviato a D00101 e a D00100.

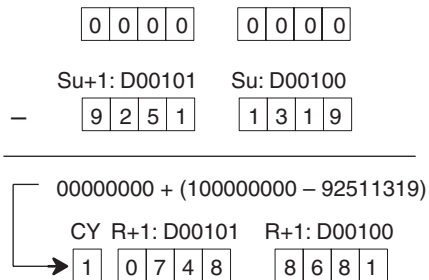


Sottrazione a 1

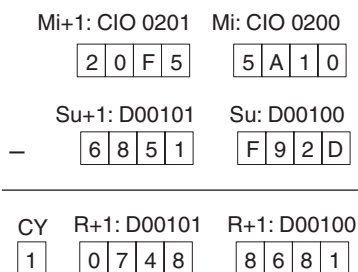


Il flag di riporto (CY) è ON, quindi il risultato viene sottratto da 0000 0000.

Sottrazione a 2



Risultato finale sottrazione



Il flag di riporto (CY) è ON, quindi il numero effettivo è -7.488.681. Poiché il contenuto di D00101 e D00100 è negativo, CY viene usato per attivare CIO 002100 ad indicare questa situazione.

3-11-15 BCD SUBTRACT WITH CARRY: -BC(416)

Scopo

Esegue una sottrazione tra dati e/o costanti in formato BCD a 4 cifre (canale singolo), incluso il flag di riporto (CY).

Simbolo programmazione ladder

| | | |
|---|----------|--------------------------------|
| — | -BC(416) | |
| | Mi | Mi: canale minuendo |
| | Su | Su: canale sottraendo |
| | R | R: canale del risultato |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -BC(416) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-BC(416) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

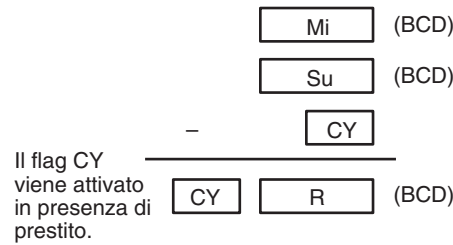
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Mi | Su | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a D32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #9999 (BCD) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,(- -)IR0 a ,(- -)IR15 | | |

Descrizione

-BC(416) sottrae i valori BCD in Su e CY da Mi e invia il risultato a R. Se il risultato è negativo, viene inviato a R come complemento a 2.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando Mi non è in formato BCD. ON quando Su non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato della sottrazione produce un prestito. OFF in tutti gli altri casi. |

Avvertenze

Se il contenuto di Mi e/o Su non è in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

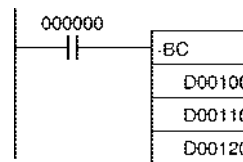
Se come risultato della sottrazione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se un'addizione produce come risultato un prestito, verrà attivato il flag di riporto.

Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00110 e CY verrà sottratto da D00100 come valore BCD a 4 cifre e il risultato verrà inviato a D00120.

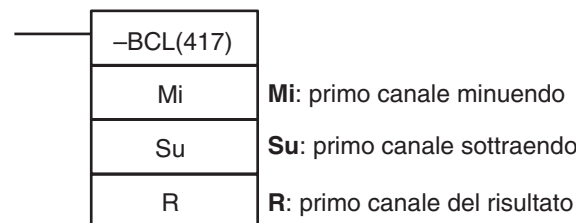


3-11-16 DOUBLE BCD SUBTRACT WITH CARRY: -BCL(417)

Scopo

Esegue una sottrazione tra dati e/o costanti in formato BCD a 8 cifre (doppio canale), incluso il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -BCL(417) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-BCL(417) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

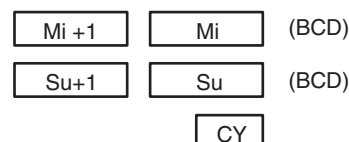
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Mi | Su | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #99999999 (BCD) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

-BCL(417) sottrae i valori BCD in Su, Su+1 e CY da Mi e Mi+1 e invia il risultato a R, R+1. Se il risultato è negativo, viene inviato a R e a R+1 come complemento a 10.



Il flag CY viene attivato in presenza di prestito.

| | | | |
|----|-----|---|-------|
| CY | R+1 | R | (BCD) |
|----|-----|---|-------|

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando Mi e/o Mi +1 non sono in formato BCD. ON quando Su e/o Su +1 non sono in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON quando il risultato della sottrazione produce un prestito. OFF in tutti gli altri casi. |

Avvertenze

Se Mi, Mi +1 e/o Su, Su +1 non sono in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

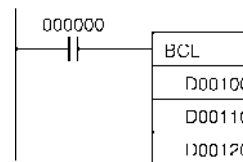
Se come risultato della sottrazione il contenuto di R, R +1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se una sottrazione produce come risultato un prestito, verrà attivato il flag di riporto.

Nota Per cancellare il flag di riporto (CY), eseguire l'istruzione CLEAR CARRY (CLC(041)).

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00111, D00110 e CY verranno sottratti da D00101 e D00100 come valori BCD a 8 cifre e il risultato verrà inviato a D00121 e a D00120.



Se il risultato della sottrazione è un numero negativo (Mi<Su o Mi+1, Mi <Su+1, Su), il risultato verrà inviato come complemento a 10 e verrà attivato il flag di riporto (CY). Per convertire il complemento a 10 in un numero effettivo, è necessario utilizzare un programma che sottragga il risultato da 0, come condizione di ingresso del flag di riporto (CY). Quando il flag di riporto passa allo stato ON significa quindi che il risultato della sottrazione è negativo.

Nota Complemento a 10

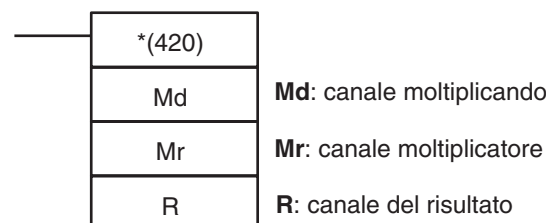
Un complemento a 10 è il valore ottenuto sottraendo ciascuna cifra binaria da 9 e aggiungendo uno al risultato. Ad esempio, il complemento a 10 di 7556 viene calcolato nel modo seguente: 9999 – 7556 + 1 = 2444. Per un numero a quattro cifre, il complemento a 10 di A è 9999 – A + 1 = B. Per ottenere il numero effettivo dal complemento a 10 B: A = 10000 – B. Ad esempio, per ottenere il numero effettivo dal complemento a 10 2444: 10000 – 2444 = 7556.

3-11-17 SIGNED BINARY MULTIPLY: *(420)

Scopo

Moltiplica dati e/o costanti esadecimali a 4 cifre con segno.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | *(420) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @*(420) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Md | Mr | R |
|--|--|----|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

*(420) moltiplica i valori binari con segno in Md e Mr e invia il risultato a R, R+1.



Flag

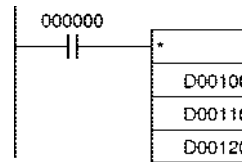
| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando *(420) viene eseguita, il flag di errore viene disattivato.
 Se come risultato della moltiplicazione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.
 Se come risultato della moltiplicazione il contenuto del bit più a sinistra di R+1 e R è 1, verrà attivato il flag negativo.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100 e D00110 verranno moltiplicati come valori esadecimali a 4 cifre con segno e il risultato verrà inviato a D00120.

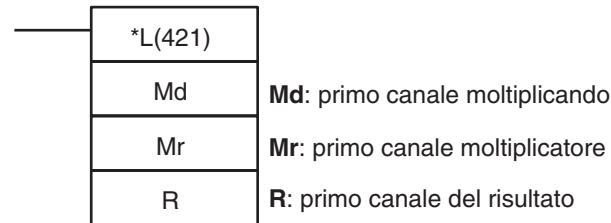


3-11-18 DOUBLE SIGNED BINARY MULTIPLY: *L(421)

Scopo

Moltiplica dati e/o costanti esadecimali a 8 cifre con segno.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | *L(421) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @*L(421) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

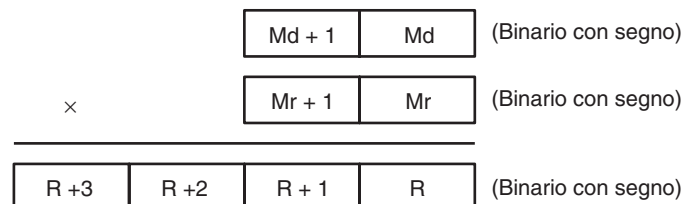
Caratteristiche operando

| Area | Md | Mr | R |
|-------------------------|------------------------|----|------------------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W510 | | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H510 | | Da H000 a H508 |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4094 | | Da T0000 a T4092 |

| Area | Md | Mr | R |
|--|--|----|--|
| Area del contatore | Da C0000 a C4094 | | Da C0000 a C4092 |
| Area DM | Da D00000 a D32766 | | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32766 | | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

*L(421) moltiplica i valori binari con segno in Md e Md+1 e in Mr e Mr+1 e invia il risultato a R, R+1, R+2 e R+3.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

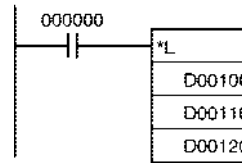
Quando *L(421) viene eseguita, il flag di errore viene disattivato.

Se come risultato della moltiplicazione il contenuto di R, R+1, R+2, R+3 è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato della moltiplicazione il contenuto del bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100, D00110, D00111 e D00110 verranno moltiplicati come valori esadecimali a 8 cifre con segno e il risultato verrà inviato a D00121 e a D00120.

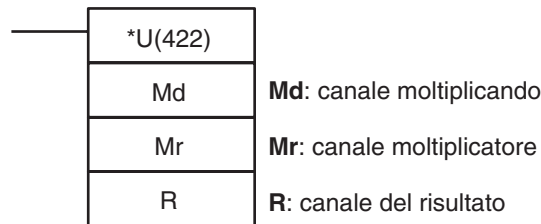


3-11-19 UNSIGNED BINARY MULTIPLY: *U(422)

Scopo

Moltiplica dati e/o costanti esadecimali a 4 cifre senza segno.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | *U(422) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @*U(422) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

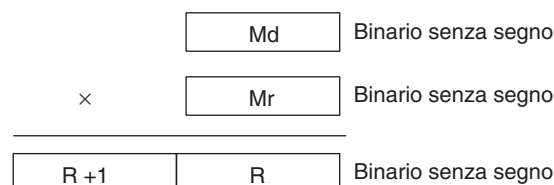
Caratteristiche operando

| Area | Md | Mr | R |
|-------------------------|--|----|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32766 (n = da 0 a C) |

| Area | Md | Mr | R |
|--|--|----|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

*U(420) moltiplica i valori binari in Md e Mr e invia il risultato a R, R+1.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

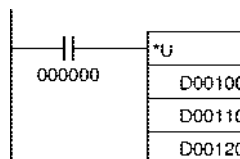
Quando *U(422) viene eseguita, il flag di errore viene disattivato.

Se come risultato della moltiplicazione il contenuto di R, R+1 è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato della moltiplicazione il contenuto del bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

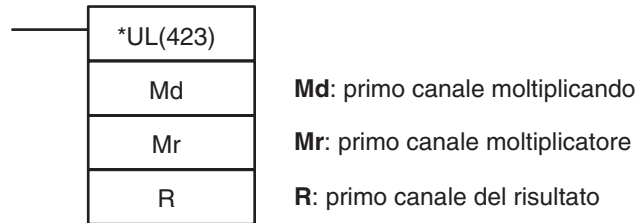
Quando nell'esempio seguente CIO 000000 è ON, D00100 e D00110 verranno moltiplicati come valori binari a 4 cifre senza segno e il risultato verrà inviato a D00121 e a D00120.



3-11-20 DOUBLE UNSIGNED BINARY MULTIPLY: *UL(423)

Scopo Moltiplica dati e/o costanti esadecimali a 8 cifre senza segno.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | *UL(423) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @*UL(423) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

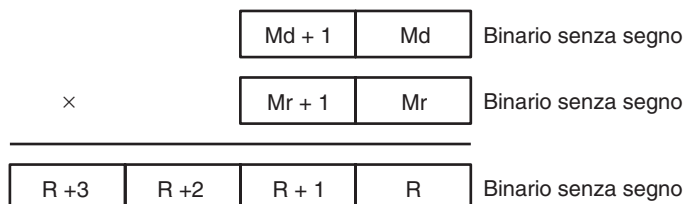
Caratteristiche operando

| Area | Md | Mr | R |
|--------------------------------------|--|----|--|
| Area CIO | Da CIO 0000 a CIO 6142 | | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W510 | | Da W000 a W508 |
| Area bit di ritenività | Da H000 a H510 | | Da H000 a H508 |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4094 | | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4094 | | Da C0000 a C4092 |
| Area DM | Da D00000 a D32766 | | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32766 | | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |

| Area | Md | Mr | R |
|--|--|----|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

*UL(423) moltiplica i valori binari senza segno in Md e Md+1 e in Mr e Mr+1 e invia il risultato a R, R+1, R+2 e R+3.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è 1. OFF in tutti gli altri casi. |

Avvertenze

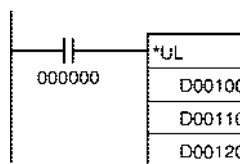
Quando *UL(423) viene eseguita, il flag di errore viene disattivato.

Se come risultato della moltiplicazione il contenuto di R, R+1, R+2, R+3 è 0000 esadecimale, verrà attivato il flag di uguaglianza.

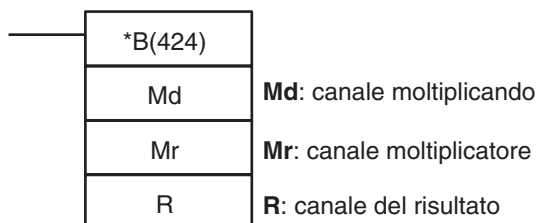
Se come risultato della moltiplicazione il contenuto del bit più a sinistra di R+3 è 1, verrà attivato il flag negativo.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100, D00110, D00111 e D00110 verranno moltiplicati come valori binari a 8 cifre senza segno e il risultato verrà inviato a D00123, D00122, D00121 e a D00120.

**3-11-21 BCD MULTIPLY: *B(424)****Scopo**

Moltiplica dati e/o costanti in formato BCD a 4 cifre (canale singolo).

Simbolo programmazione ladder

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | *B(424) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @*B(424) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Md | Mr | R |
|--|--|----|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #9999 (BCD) | | --- |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

*B(424) moltiplica il contenuto BCD in Md e Mr e invia il risultato a R, R+1.

| | | |
|---|-------|-------|
| | Md | (BCD) |
| × | Mr | (BCD) |
| | R +1 | R |
| | (BCD) | |

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando Md non è in formato BCD. ON quando Mr non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |

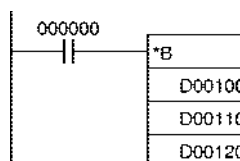
Avvertenze

Se il contenuto di Md e/o Mr non è in formato BCD, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato della moltiplicazione il contenuto di R, R+1 è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100 e D00110 verranno moltiplicati come valori BCD a 4 cifre e il risultato verrà inviato a D00121 e a D00120.

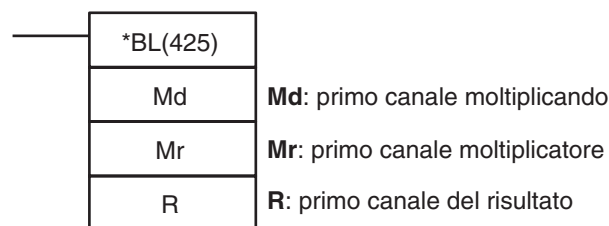


3-11-22 DOUBLE BCD MULTIPLY: *BL(425)

Scopo

Moltiplica dati e/o costanti in formato BCD a 8 cifre (doppio canale).

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | *BL(425) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @*BL(425) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

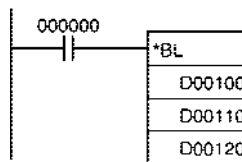
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Md | Mr | R |
|-------------------------|------------------------|----|------------------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W510 | | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H510 | | Da H000 a H508 |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A956 |

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00101, D00100, D00111 e D00110 verranno moltiplicati come valori BCD a 8 cifre senza segno e il risultato verrà inviato a D00123, D00122, D00121 e a D00120.

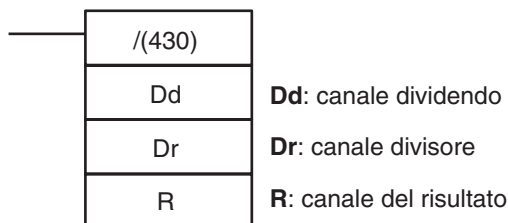


3-11-23 SIGNED BINARY DIVIDE: /(430)

Scopo

Esegue una divisione tra dati e/o costanti esadecimali a 4 cifre (canale singolo) con segno.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | /(430) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/(430) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

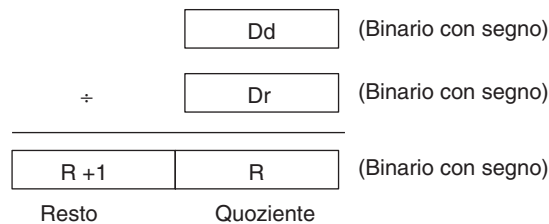
Caratteristiche operando

| Area | Dd | Dr | R |
|--------------------------------------|--|----|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |

| Area | Dd | Dr | R |
|--|--|-------------------------------|-----|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | Da #0001 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

/(430) divide i valori binari con segno (16 bit) in Dd per quelli in Dr e invia il risultato a R, R+1. Il quoziente viene inserito in R e il resto in R+1.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando come risultato della divisione R corrisponde a 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

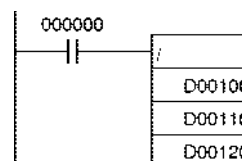
Se il contenuto di Dr è 0, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato della divisione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato della divisione il contenuto del bit più a sinistra di R è 1, verrà attivato il flag negativo.

Esempi

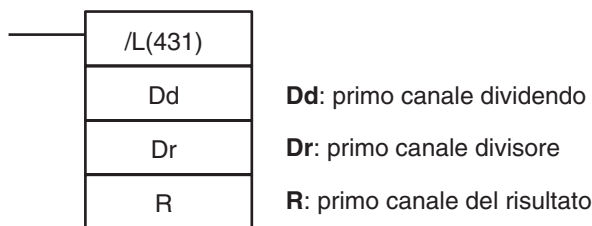
Quando nell'esempio seguente CIO 000000 è ON, D00100 verrà diviso per D00110 come valore binario a 4 cifre con segno, il quoziente verrà inviato a D00120 e il resto a D00121.



3-11-24 DOUBLE SIGNED BINARY DIVIDE: /L(431)

Scopo Esegue una divisione tra dati e/o costanti esadecimali a 8 cifre (doppio canale) con segno.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | /L(431) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/L(431) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

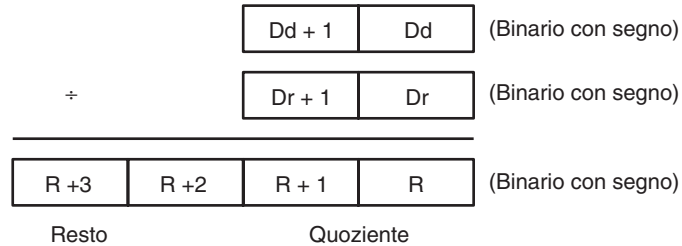
Caratteristiche operando

| Area | Dd | Dr | R |
|--------------------------------------|--|--------------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6142 | | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W510 | | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H510 | | Da H000 a H508 |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4094 | | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4094 | | Da C0000 a C4092 |
| Area DM | Da D00000 a D32766 | | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32766 | | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | Da #00000001 a #FFFFFFF (binario) | --- |
| Registri dati | --- | | |

| Area | Dd | Dr | R |
|--|--|----|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

/L(431) divide i valori binari con segno in Dd e Dd+1 per quelli contenuti in Dr e Dr+1 e invia il risultato a R, R+1, R+2 e R+3. Il quoziente viene inviato a R e R+1 e il resto a R+2 e R+3.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando come risultato della divisione R+1, R corrisponde a 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R+1, R è 1. OFF in tutti gli altri casi. |

Avvertenze

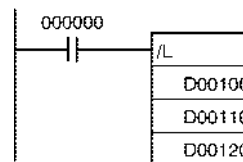
Quando il resto del risultato R+3, R+2 è 0, verrà attivato il flag di errore.

Se come risultato della divisione il contenuto di R+1, R è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato della divisione il contenuto del bit più a sinistra di R+1, R è 1, verrà attivato il flag negativo.

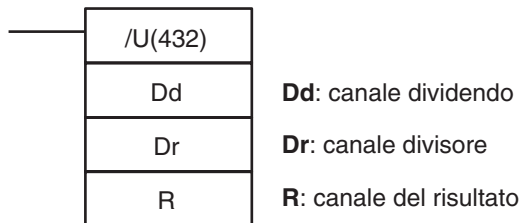
Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00101 e D00100 verranno divisi per D00111 e D00110 come valori esadecimali a 8 cifre con segno, il quoziente verrà inviato a D00121 e a D00120 e il resto a D00123 e a D00122.



3-11-25 UNSIGNED BINARY DIVIDE: /U(432)**Scopo**

Esegue una divisione tra dati e/o costanti esadecimali a 4 cifre (canale singolo) senza segno.

Simbolo programmazione ladder**Variazioni**

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | /U(432) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/U(432) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

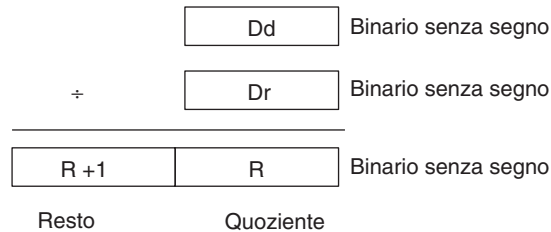
Caratteristiche operando

| Area | Dd | Dr | R |
|--------------------------------------|--|-------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | Da #0001 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | | --- |

| Area | Dd | Dr | R |
|--|--|----|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

/U(432) divide i valori binari senza segno in Dd per quelli in Dr e invia il risultato a R e il resto a R+1.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando come risultato della divisione R corrisponde a 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

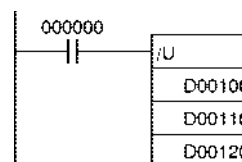
Se come risultato della divisione il contenuto di R+1 è 0, verrà attivato il flag di errore.

Se come risultato della divisione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato della divisione il contenuto del bit più a sinistra di R è 1, verrà attivato il flag negativo.

Esempi

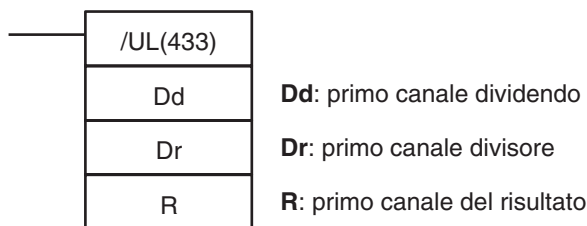
Quando nell'esempio seguente CIO 000000 è ON, D00100 verrà diviso per D00110 come valore binario a 4 cifre senza segno, il quoziente verrà inviato a D00120 e il resto a D00121.



3-11-26 DOUBLE UNSIGNED BINARY DIVIDE: /UL(433)

Scopo Esegue una divisione tra dati e/o costanti esadecimali a 8 cifre (doppio canale) senza segno.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | /UL(433) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/UL(433) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

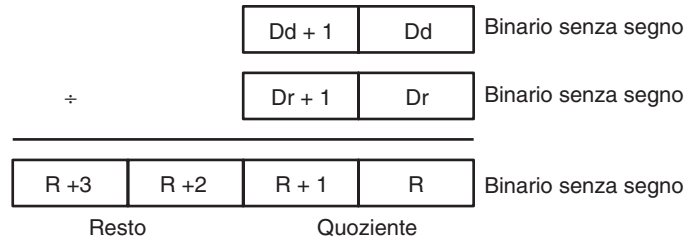
Caratteristiche operando

| Area | Dd | Dr | R |
|--------------------------------------|--|--------------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6142 | | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W510 | | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H510 | | Da H000 a H508 |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4094 | | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4094 | | Da C0000 a C4092 |
| Area DM | Da D00000 a D32766 | | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32766 | | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | Da #00000001 a #FFFFFFF (binario) | --- |
| Registri dati | --- | | |

| Area | Dd | Dr | R |
|--|--|----|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

/UL(433) divide i valori binari senza segno in Dd e Dd+1 per quelli contenuti in Dr e Dr+1 e invia il quoziente a R, R+1 e il resto a R+2 e R+3.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando come risultato della divisione R+1, R corrisponde a 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R+1, R è 1. OFF in tutti gli altri casi. |

Avvertenze

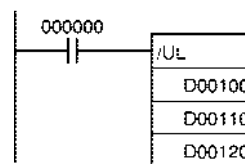
Quando il contenuto di Dr, Dr+1 è 0, verrà attivato il flag di errore.

Se come risultato della divisione il contenuto di R, R+1 è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato della divisione il contenuto del bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

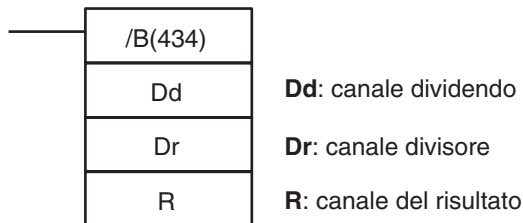
Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100 e D00101 verranno divisi per D00111 e D00110 come valori esadecimali a 8 cifre senza segno, il quoziente verrà inviato a D00121 e a D00120 e il resto a D00123 e a D00122.



3-11-27 BCD DIVIDE: /B(434)**Scopo**

Esegue una divisione tra dati e/o costanti in formato BCD a 4 cifre (canale singolo).

Simbolo programmazione ladder**Variazioni**

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | /B(434) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/B(434) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

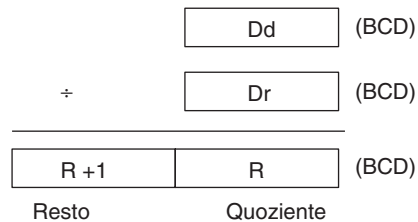
Caratteristiche operando

| Area | Dd | Dr | R |
|--------------------------------------|--|---------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #9999 (BCD) | Da #0001 a #9999 (BCD) | --- |
| Registri dati | Da DR0 a DR15 | | --- |

| Area | Dd | Dr | R |
|--|--|----|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

/B(434) divide i valori BCD contenuti in Dd per quelli in Dr e invia il risultato a R e il resto a R+1.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON quando Dd non è in formato BCD. ON quando Dr non è in formato BCD. ON quando il resto è 0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando R è 0. OFF in tutti gli altri casi. |

Avvertenze

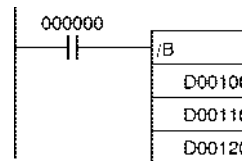
Se il contenuto di Dd o Dr non è in formato BCD o se il resto (R+1) è 0, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato della divisione il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato della divisione il bit più a sinistra di R è 1, verrà attivato il flag negativo.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00100 verrà diviso per D00110 come valori BCD a 4 cifre, il quoziente verrà inviato a D00120 e il resto a D00120.



3-11-28 DOUBLE BCD DIVIDE: /BL(435)**Scopo**

Esegue una divisione tra dati e/o costanti in formato BCD a 8 cifre (doppio canale).

Simbolo programmazione ladder

| | | |
|---|----------|---------------------------------------|
| — | /BL(435) | |
| | Dd | Dd : primo canale dividendo |
| | Dr | Dr : primo canale divisore |
| | R | R : primo canale del risultato |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | /BL(435) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/BL(435) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportate |
| Aggiornamento immediato | | Non supportate |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

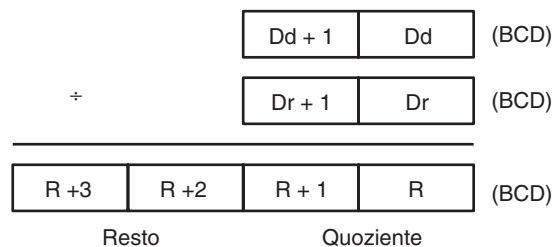
Caratteristiche operando

| Area | Dd | Dr | R |
|--------------------------------------|--|-----------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6142 | | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W510 | | Da W000 a W508 |
| Area bit di ritenività | Da H000 a H510 | | Da H000 a H508 |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4094 | | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4094 | | Da C0000 a C4092 |
| Area DM | Da D00000 a D32766 | | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32766 | | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #99999999 (BCD) | Da #00000001 a #99999999 (BCD) | --- |
| Registri dati | --- | | |

| Area | Dd | Dr | R |
|--|--|----|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

/BL(435) divide i valori BCD in Dd e Dd+1 per quelli contenuti in Dr e Dr+1 e invia il quoziente a R, R+1 e il resto a R+2 e R+3.

**Flag**

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON quando il contenuto di Dd, Dd+1 non è in formato BCD. ON quando il contenuto di Dr, Dr + 1 non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |

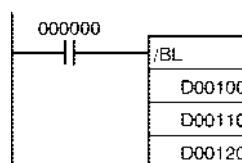
Avvertenze

Se il contenuto di Dd, Dd+1 e/o Dr, Dr+1 non è in formato BCD o se il contenuto di Dr, Dr+1 è 0, verrà generato un errore e verrà attivato il flag di errore.

Se come risultato della divisione il contenuto di R, R+1, è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Esempi

Quando nell'esempio seguente CIO 000000 è ON, D00101 e D00100 verranno divisi per D00111 e D00110 come valori BCD a 8 cifre, il quoziente verrà inviato a D00121 e a D00120 e il resto a D00123 e a D00122.



3-12 Istruzioni di conversione

In questa sezione vengono descritte le istruzioni utilizzate per la conversione dei dati.

| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|-----------------------------|------------------|-----------------|--------|
| BCD-TO-BINARY | BIN | 023 | 466 |
| DOUBLE BCD-TO-DOUBLE BINARY | BINL | 058 | 467 |
| BINARY-TO-BCD | BCD | 024 | 469 |
| DOUBLE BINARY-TO-DOUBLE BCD | BCDL | 059 | 470 |

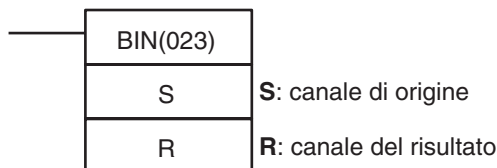
| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|--------------------------------|------------------|-----------------|--------|
| 2'S COMPLEMENT | NEG | 160 | 472 |
| DOUBLE 2'S COMPLEMENT | NEGL | 161 | 474 |
| 16-BIT TO 32-BIT SIGNED BINARY | SIGN | 600 | 476 |
| DATA DECODER | MLPX | 076 | 477 |
| DATA ENCODER | DMPX | 077 | 482 |
| ASCII CONVERT | ASC | 086 | 486 |
| ASCII TO HEX | HEX | 162 | 490 |
| COLUMN TO LINE | LINE | 063 | 494 |
| LINE TO COLUMN | COLM | 064 | 496 |
| SIGNED BCD-TO-BINARY | BINS | 470 | 499 |
| DOUBLE SIGNED BCD-TO-BINARY | BISL | 472 | 502 |
| SIGNED BINARY-TO-BCD | BCDS | 471 | 505 |
| DOUBLE SIGNED BINARY-TO-BCD | BDSL | 473 | 507 |
| GRAY CODE CONVERSION | GRY | 474 | 511 |

3-12-1 BCD-TO-BINARY: BIN(023)

Scopo

Converte dati in formato BCD in dati binari.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | BIN(023) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BIN(023) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|-------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |

| Area | S | R |
|--|--|---|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

BIN(023) converte i dati BCD in S in dati binari e scrive il risultato in R.

S → R

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di S non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | OFF |

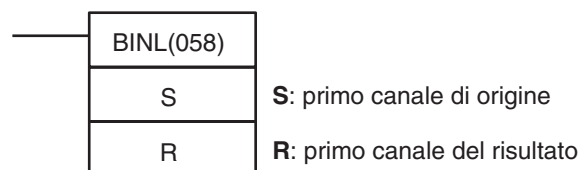
Esempio

Nello schema riportato di seguito viene illustrato un esempio di conversione da BCD in binario.

$$\begin{array}{cccccc}
 15 & 12 & 11 & 8 & 7 & 4 & 3 & 0 \\
 S & \boxed{3} & \boxed{4} & \boxed{5} & \boxed{2} & & & \\
 & \times 10^3 & \times 10^2 & \times 10^1 & \times 10^0 & & &
 \end{array}
 \rightarrow
 \begin{array}{cccccc}
 15 & 12 & 11 & 8 & 7 & 4 & 3 & 0 \\
 R & \boxed{0} & \boxed{D} & \boxed{7} & \boxed{C} & & & \\
 & \times 16^3 & \times 16^2 & \times 16^1 & \times 16^0 & & &
 \end{array}$$

3-12-2 DOUBLE BCD-TO-DOUBLE BINARY: BINL(058)**Scopo**

Converte dati in formato BCD a 8 cifre in dati esadecimali a 8 cifre (binari a 32 bit).

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | BINL(058) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BINL(058) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

BINL(058) converte i dati BCD a 8 cifre in S e S+1 in dati a 8 cifre esadecimali (binari a 32 bit) e scrive il risultato in R e in R+1.

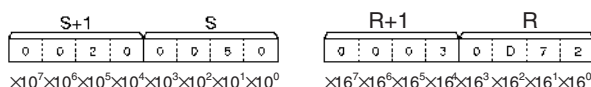


Flag

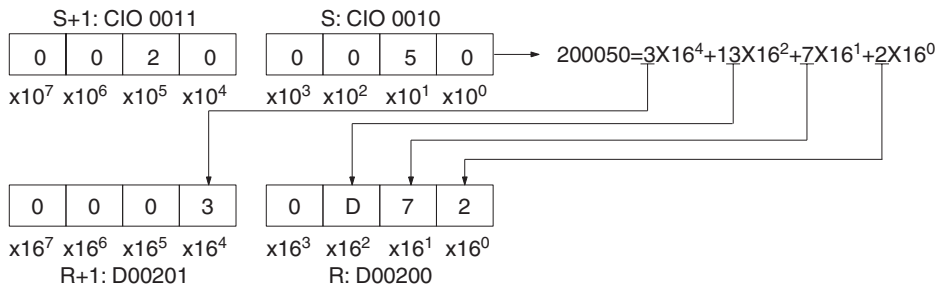
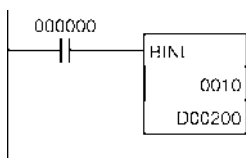
| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di S+1 non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | OFF |

Esempi

Nello schema riportato di seguito viene illustrato un esempio di conversione a 8 cifre da BCD in binario.



Nell'esempio seguente, quando CIO 000000 è ON, il valore BCD a 8 cifre in CIO 0010 e CIO 0011 viene convertito in valore esadecimale e memorizzato in D00200 e in D00201.

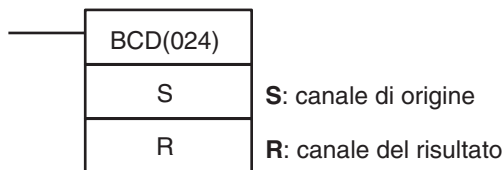


3-12-3 BINARY-TO-BCD: BCD(024)

Scopo

Converte un canale di dati binari in un canale di dati in formato BCD.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BCD(024) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BCD(024) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

S: canale di origine

S deve essere compreso tra 0000 e 270F esadecimale (0000 e 9999 decimale).

Caratteristiche operando

| Area | S | R |
|-------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |

| Area | S | R |
|--|--|---|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione BCD(024) converte i dati binari in S in dati BCD e scrive il risultato in R.

⊘ → R

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se il contenuto di S supera 270F (9999 decimale). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |

Avvertenze

Il contenuto di S non deve superare 270F (9999 decimale).

Esempio

Nello schema riportato di seguito viene illustrato un esempio di conversione da BCD in binario.

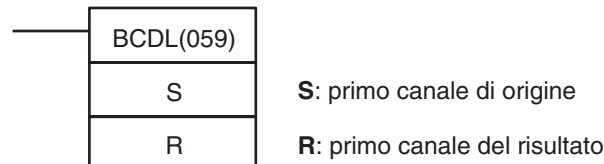
| | | | | | | | | | | |
|---|---|---|---|---|--|---|---|---|---|---|
| 15 12 11 8 7 4 3 0 | → | 15 12 11 8 7 4 3 0 | | | | | | | | |
| ⊘ <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">E</td><td style="padding: 2px 5px;">C</td></tr></table> | 1 | 0 | E | C | | R <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px 5px;">4</td><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;">2</td></tr></table> | 4 | 3 | 3 | 2 |
| 1 | 0 | E | C | | | | | | | |
| 4 | 3 | 3 | 2 | | | | | | | |
| $\times 16^3 \times 16^2 \times 16^1 \times 16^0$ | | $\times 10^3 \times 10^2 \times 10^1 \times 10^0$ | | | | | | | | |

3-12-4 DOUBLE BINARY-TO-DOUBLE BCD: BCDL(059)

Scopo

Converte dati esadecimali a 8 cifre (binari a 32 bit) in dati in formato BCD a 8 cifre.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | BCDL(059) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BCDL(059) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: primo canale di origine

Il contenuto di S+1 e S deve essere compreso tra 0000 0000 e 05F5 E0FF esadecimale (0000 0000 e 9999 9999 decimale).

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

BCDL(059) converte i dati esadecimale a 8 cifre (binari a 32 bit) in S e S+1 in dati BCD a 8 cifre e scrive il risultato in R e in R+1.



Flag

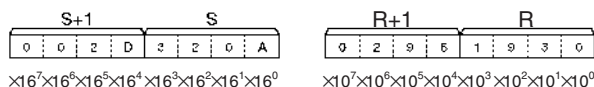
| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se il contenuto di S e S+1 supera 05F5 E0FF (9999 9999 decimale). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |

Avvertenze

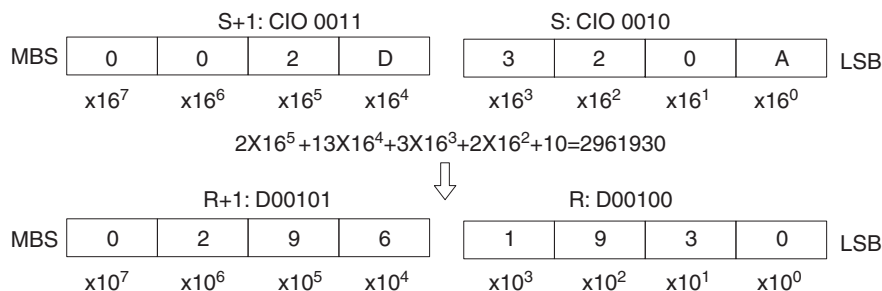
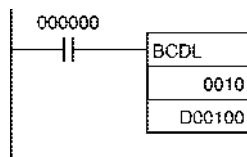
Il contenuto di S+1 e S non deve superare 05F5 E0FF (9999 9999 decimale).

Esempi

Nello schema riportato di seguito viene illustrato un esempio di conversione a 8 cifre da BCD in binario.



Nell'esempio seguente, quando CIO 000000 è ON, il valore esadecimale in CIO 0011 e CIO 0010 viene convertito in valore BCD e memorizzato in D00200 e in D00201.

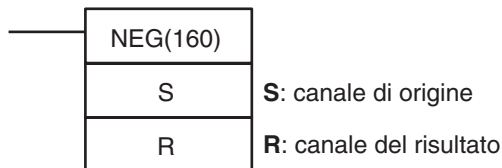


3-12-5 2'S COMPLEMENT: NEG(160)

Scopo

Calcola il complemento a 2 di un canale di dati esadecimali.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | NEG(160) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NEG(160) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|-------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |

| Area | S | R |
|--|--|-----|
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

NEG(160) calcola il complemento a 2 di S e scrive il risultato in R. Sostanzialmente il calcolo del complemento a 2 inverte lo stato dei bit in S e aggiunge 1.

$$\overline{(S)} \xrightarrow{\substack{\text{Complemento a 2} \\ \text{(complemento + 1)}}} (R)$$

Nota Questa operazione, ossia l'inversione dello stato dei bit e l'aggiunta di 1, equivale a sottrarre il contenuto di S da 0000.

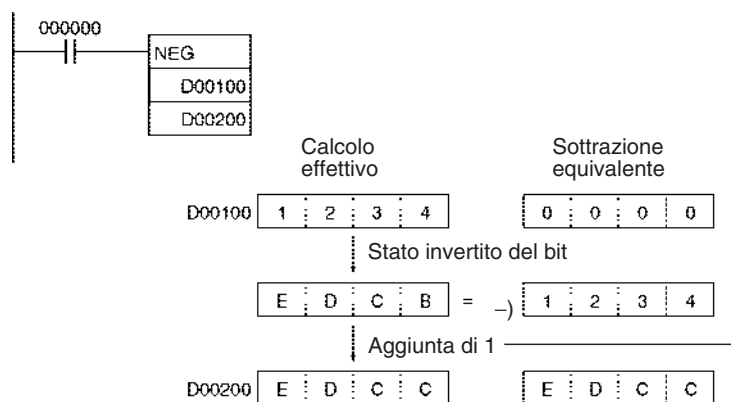
Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 del risultato è ON. OFF in tutti gli altri casi. |

Nota Il risultato per 8000 esadecimale sarà 8000 esadecimale.

Esempio

Nell'esempio seguente, quando CIO 000000 è ON, NEG(160) calcola il complemento a 2 del contenuto di D00100 e scrive il risultato in D00200.

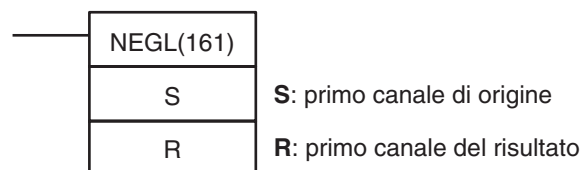


3-12-6 DOUBLE 2'S COMPLEMENT: NEGL(161)

Scopo

Calcola il complemento a 2 di due canali di dati esadecimali.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | NEGL(161) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @NEGL(161) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--------------------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |

| Area | S | R |
|--|--|-----|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Nota R e R+1 devono essere nella stessa area dati

Descrizione

NEGL(161) calcola il complemento a 2 di S+1 e S e scrive il risultato in R+1 e R. Sostanzialmente il calcolo del complemento a 2 inverte lo stato dei bit in S+1 e S e aggiunge 1.

$$\overline{(S+1, S)} \xrightarrow{\text{Complemento a 2 (complemento + 1)}} (R+1, R)$$

Nota Questa operazione, ossia l'inversione dello stato dei bit e l'aggiunta di 1, equivale a sottrarre il contenuto di S+1 e S da 0000 0000.

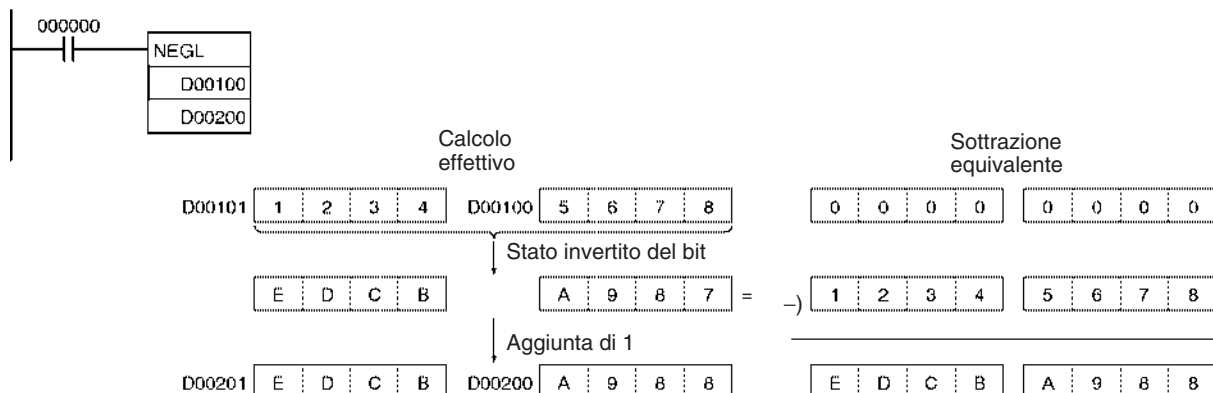
Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di R+1 è ON. OFF in tutti gli altri casi. |

Nota Il risultato per 8000 esadecimale sarà 8000 esadecimale.

Esempio

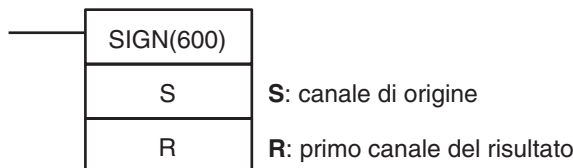
Nell'esempio seguente, quando CIO 000000 è ON, NEGL(161) calcola il complemento a 2 del contenuto di D00101 e D00100 e scrive il risultato in D00201 e D00200.



3-12-7 16-BIT TO 32-BIT SIGNED BINARY: SIGN(600)

Scopo Convertire un valore binario con segno a 16 bit nell'equivalente a 32 bit.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SIGN(600) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SIGN(600) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

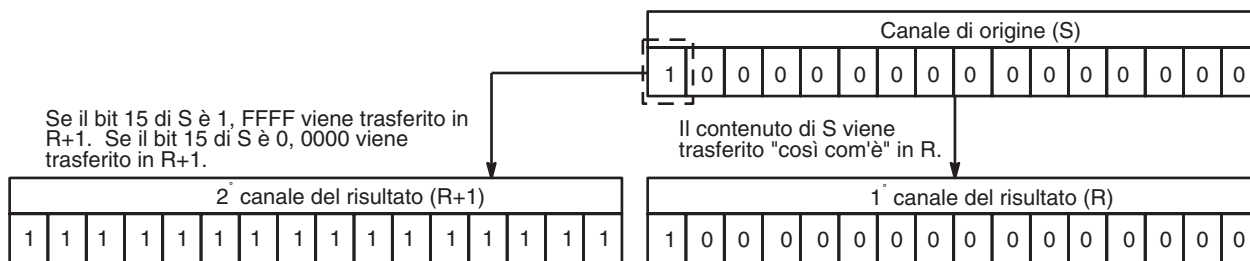
| Area | S | R |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | --- |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Nota R e R+1 devono essere nella stessa area dati.

Descrizione

SIGN(600) converte il numero binario in S a 16 bit con segno nel numero binario equivalente a 32 bit con segno e scrive il risultato in R+1 e in R.

La conversione viene eseguita copiando il contenuto di S in R e scrivendo FFFF in R+1 se il bit 15 di S è 1 o 0000 in R+1 se il bit 15 di S è 0.

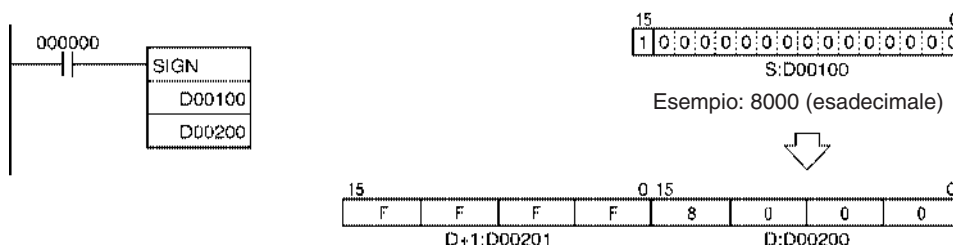


Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di R+1 è ON. OFF in tutti gli altri casi. |

Esempio

Nell'esempio seguente, quando CIO 000000 è ON, SIGN(600) converte il contenuto binario a 16 bit con segno di D00100 (#8000 = -32.768 decimale) nell'equivalente a 32 bit (#FFFF 8000 = -32.768 decimale) e scrive il risultato in D00201 e in D00200.

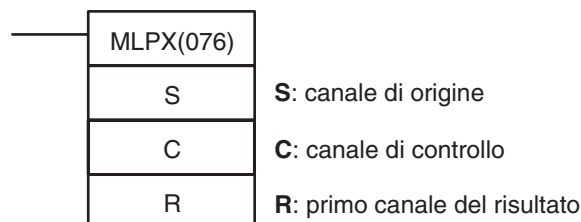


3-12-8 DATA DECODER: MLPX(076)

Scopo

Legge il valore numerico nella cifra o nel byte specificato e lo copia nel canale di origine, imposta su ON il bit corrispondente e su OFF tutti gli altri bit nel canale del risultato o nell'intervallo di 16 canali.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | MLPX(076) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MLPX(076) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

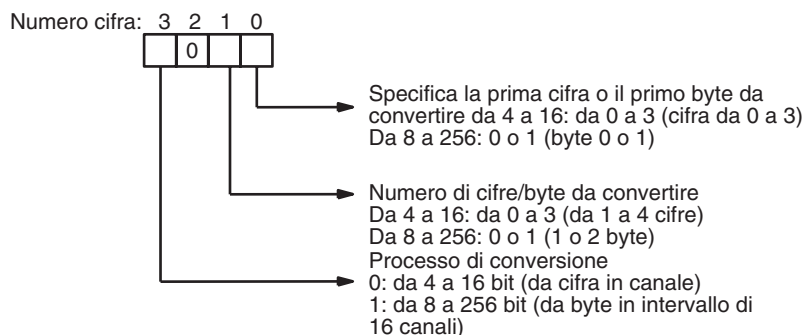
Operandi

S: canale di origine

I dati nel canale di origine indicano la posizione del o dei bit che verranno impostati su ON.

C: canale di controllo

Il canale di controllo specifica se MLPX(076) eseguirà una conversione da 4 a 16 bit o una conversione da 8 a 256 bit, il numero di cifre o byte da convertire e la cifra o il byte di inizio.



R: primo canale del risultato

I canali del risultato possono essere in qualsiasi posizione da 1 a 32, in base al tipo di processo di conversione e al numero di cifre/byte da convertire. I canali del risultato devono essere nella stessa area dati.

Caratteristiche operando

| Area | S | C | R |
|--------------------------------------|--|-------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | --- |
| Registri dati | Da DR0 a DR15 | | --- |

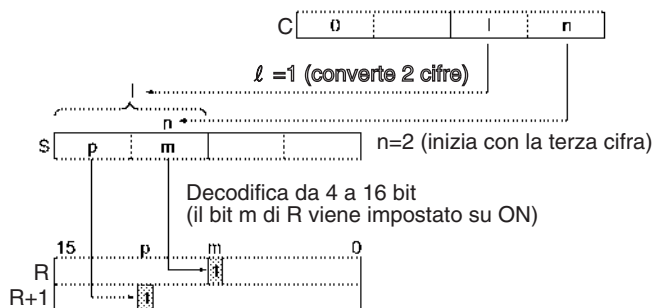
| Area | S | C | R |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

MLPX(076) può eseguire conversioni da 4 a 16 bit o da 8 a 256 bit. Impostare il bit più a sinistra di C su 0 per specificare la conversione da 4 a 16 bit e impostarlo su 1 per specificare la conversione da 8 a 256 bit.

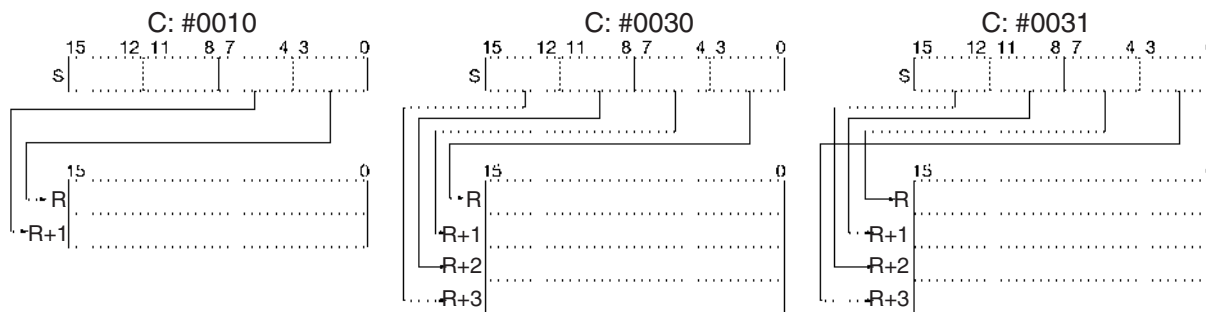
Conversione da 4 a 16 bit

Quando la cifra più a sinistra di C è 0, MLPX(076) assume il valore della cifra specificata in S (da 0 a F) e imposta su ON il bit corrispondente nel canale del risultato. Tutti gli altri bit nel canale del risultato verranno impostati su OFF. È possibile convertire fino a quattro cifre.



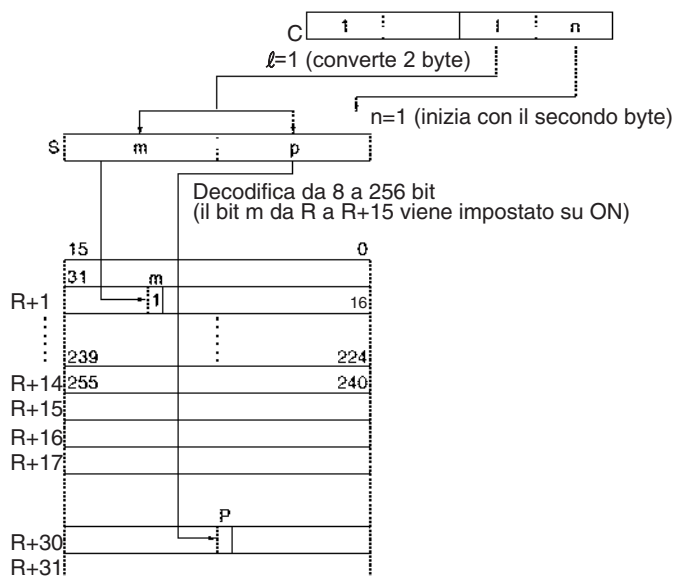
Quando vengono convertite due o più cifre, MLPX(076) leggerà le cifre in S da destra verso sinistra e dopo avere raggiunto la cifra più a sinistra tornerà a capo alla cifra più a destra, se necessario.

Nello schema riportato di seguito sono illustrati alcuni valori di esempio per C e le conversioni da 4 a 16 bit prodotte.



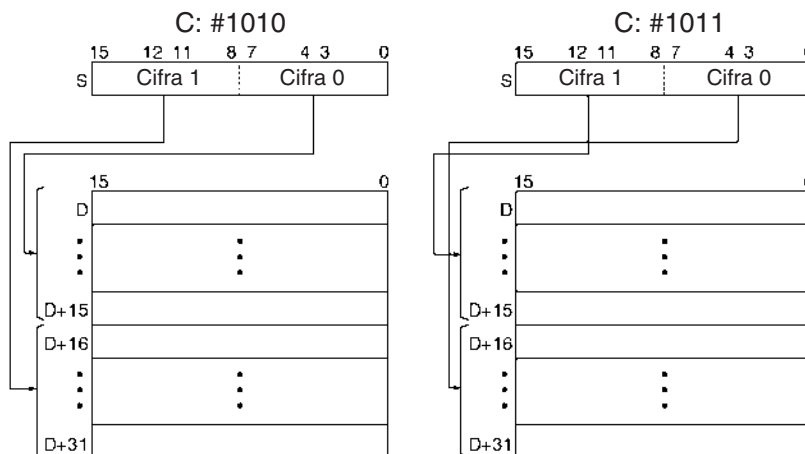
Conversione da 8 a 25 bit

Quando la cifra più a sinistra di C è 1, MLPX(076) assume il valore del byte specificato in S (da 00 a FF) e imposta su ON il bit corrispondente nell'intervallo dei 16 canali del risultato. Tutti gli altri bit nei canali del risultato verranno impostati su OFF. È possibile convertire fino a due byte.



Quando vengono convertiti due o più byte, MLPX(076) leggerà i byte in S da destra verso sinistra e tornerà a capo alla cifra più a destra se il byte più a sinistra (byte 1) è stato specificato come byte di inizio.

Nello schema riportato di seguito sono illustrati alcuni valori di esempio per C e le conversioni da 8 a 256 bit prodotte.



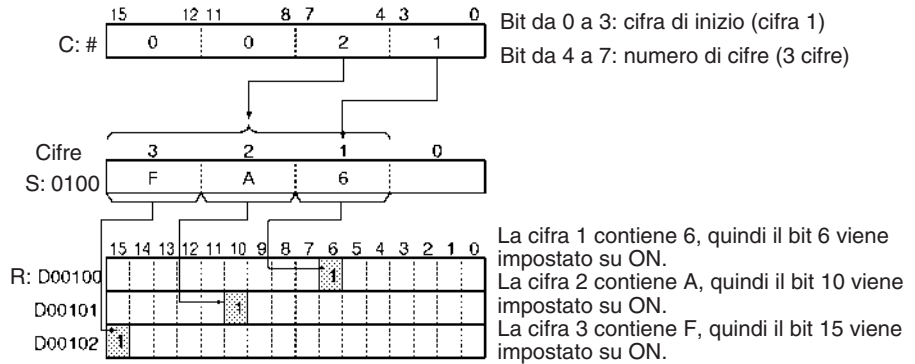
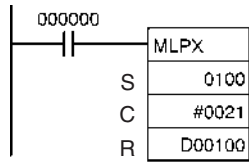
Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se C non è incluso negli intervalli specificati. OFF in tutti gli altri casi. |

Esempi

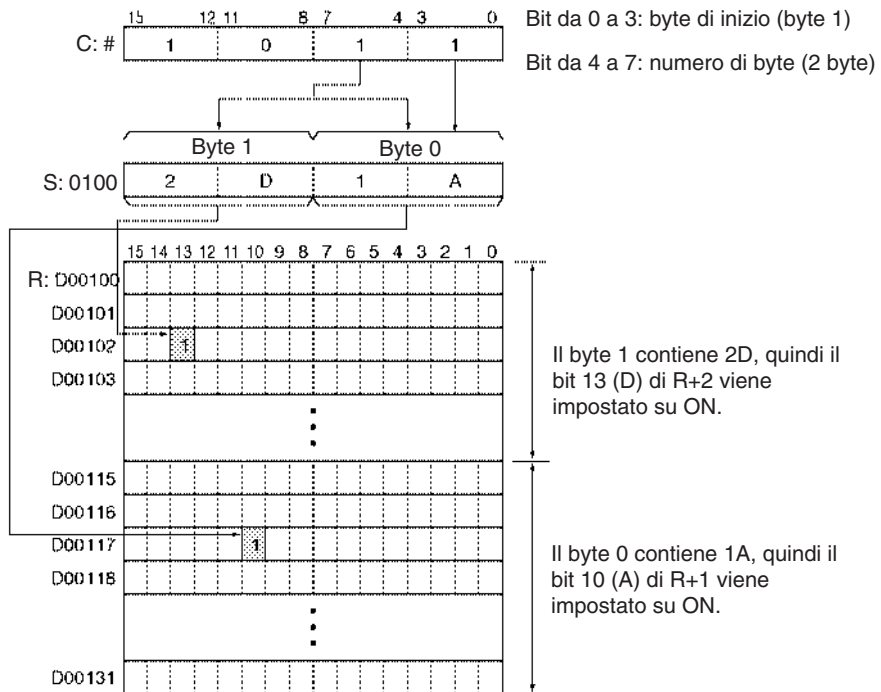
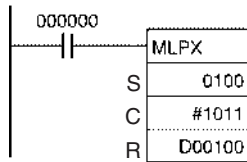
Conversione da 4 a 16 bit

Nell'esempio seguente, quando CIO 000000 è ON, MLPX(076) convertirà 3 cifre in S cominciando dalla cifra 1 (la seconda cifra), come indicato da C (#0021). I bit corrispondenti in D00100, D00101 e D00102 verranno impostati su ON.



Conversione da 8 a 25 bit

Nell'esempio seguente, quando CIO 000000 è ON, MLPX(076) convertirà i 2 byte in S cominciando dal byte 1 (il byte più a sinistra), come indicato da C (#1011). I bit corrispondenti in D00100-D00115 e D00116-D00131 verranno impostati su ON.

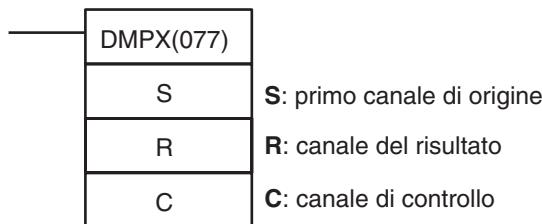


3-12-9 DATA ENCODER: DMPX(077)

Scopo

Determina la posizione del primo o dell'ultimo bit impostato su ON all'interno del canale di origine o dell'intervallo di 16 canali e scrive tale valore nella cifra o nel byte specificato nel canale del risultato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| | Eseguita a ogni ciclo con condizione ON | DMPX(077) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DMPX(077) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: primo canale di origine

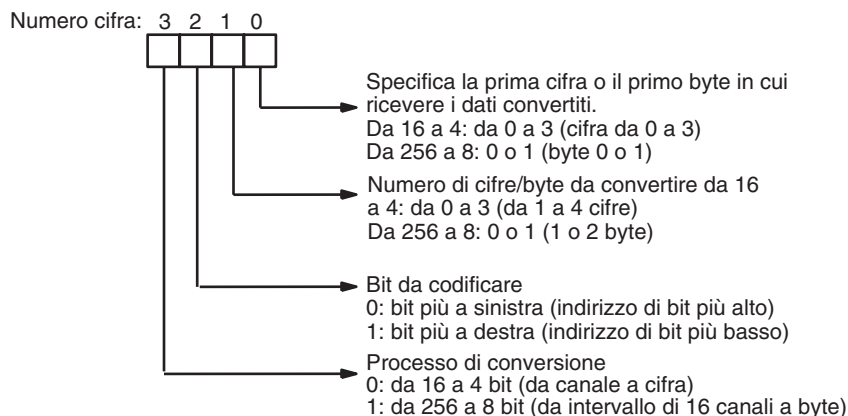
I canali di origine possono essere in qualsiasi posizione da 1 a 32, in base al tipo di processo di conversione e al numero di cifre/byte da convertire. I canali di origine devono essere nella stessa area dati.

R: canale del risultato

Le posizioni dei bit che erano impostati su ON nel canale o nei canali di origine vengono scritte nelle cifre/byte in R cominciando dalla prima cifra o dal primo byte specificato.

C: canale di controllo

Il canale di controllo specifica se DMPX(077) eseguirà una conversione da 16 a 4 bit o una conversione da 256 a 8 bit, se verrà codificato il bit più a sinistra o più a destra impostato su ON, il numero di cifre o byte da convertire e la cifra o il byte di inizio in cui verranno scritti i risultati.



Caratteristiche operando

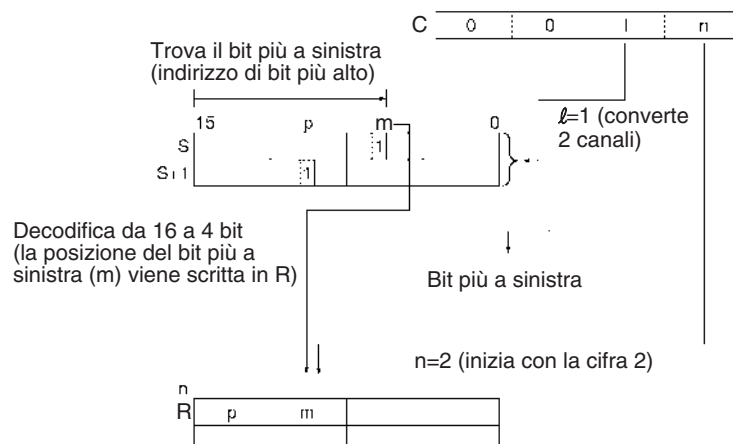
| Area | S | R | C |
|--|--|-----------------|-------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | Da A000 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | --- | Solo valori specificati |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

DMPX(077) può eseguire conversioni da 16 a 4 bit o da 256 a 8 bit. Impostare il bit più a sinistra di C su 0 per specificare la conversione da 16 a 4 bit e impostarlo su 1 per specificare la conversione da 256 a 8 bit.

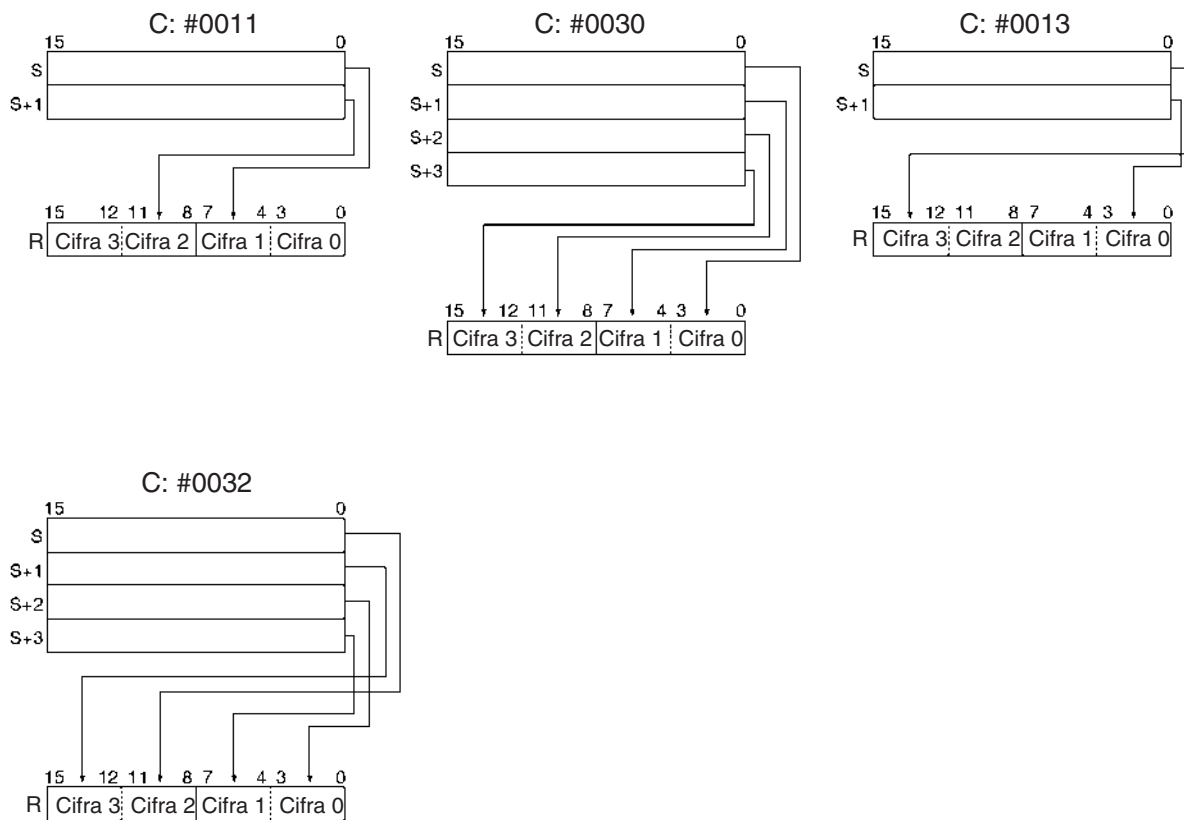
Conversione da 16 a 4 bit

Quando la quarta cifra (più a sinistra) di C è 0, DMPX(077) trova le posizioni dei bit più a sinistra o più a destra impostati su ON in un numero massimo di 4 canali di origine e scrive queste posizioni in R cominciando dalla cifra specificata. (Impostare la terza cifra di C su 0 o su 1 per trovare rispettivamente i bit più a sinistra o i bit più a destra impostati su ON).



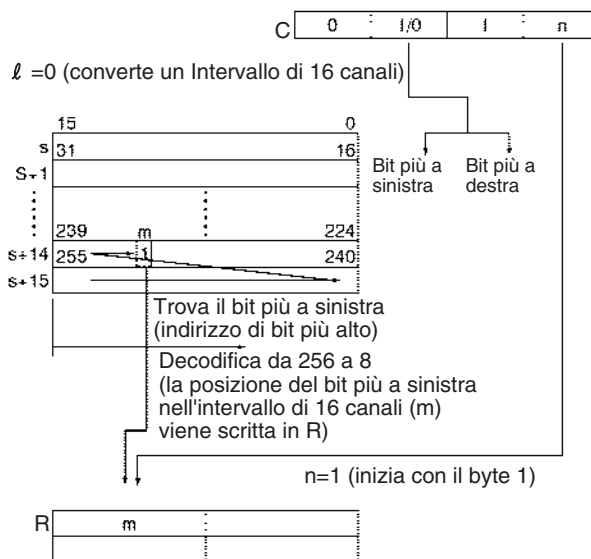
Quando vengono convertite due o più cifre, DMPX(077) scriverà i valori nelle cifre in R da destra verso sinistra e dopo avere raggiunto la cifra più a sinistra tornerà a capo alla cifra più a destra, se necessario.

Nello schema riportato di seguito sono illustrati alcuni valori di esempio per C e le conversioni da 16 a 4 bit prodotte.



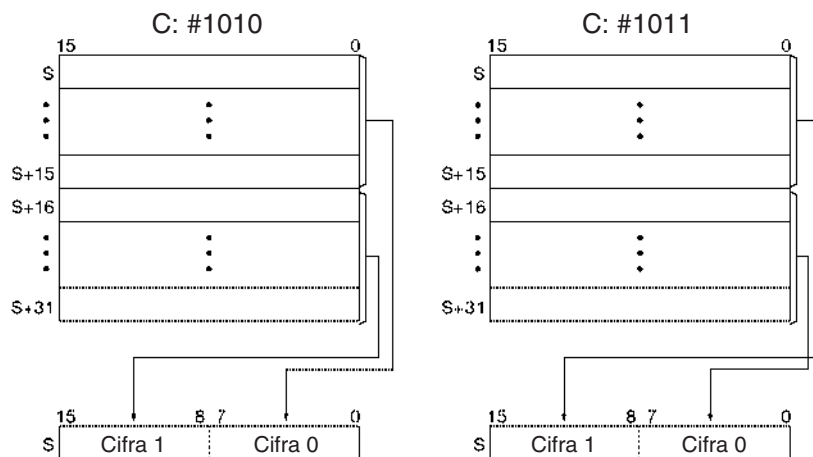
Conversione da 256 a 8 bit

Quando la quarta cifra (più a sinistra) di C è 1, DMPX(077) trova le posizioni dei bit più a sinistra (indirizzo di bit più alto) o più a destra (indirizzo di bit più basso) impostati su ON in uno o due intervalli di 16 canali di origine. Le posizioni di questi bit vengono scritte in R, cominciando dal byte specificato. (Impostare la terza cifra di C su 0 o su 1 per trovare rispettivamente i bit più a sinistra o i bit più a destra impostati su ON).



Quando vengono convertiti due o più byte, DMPX(077) scriverà i valori nei byte in R da destra verso sinistra e tornerà a capo alla cifra più a destra se il byte più a sinistra (byte 1) è stato specificato come byte di inizio.

Nello schema riportato di seguito sono illustrati alcuni valori di esempio per C e le conversioni da 256 a 8 bit prodotte.



Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|--|
| Flag di errore | ER | ON se uno dei canali di origine contiene il valore 0000 esadecimale (nessun bit da codificare). ON se C non è incluso negli intervalli specificati. OFF in tutti gli altri casi. |

Avvertenze

Se i dati di conversione contengono il valore 0000 esadecimale, ma devono essere codificati altri dati, separare le conversioni utilizzando più istruzioni DMPX(077).

DMPX(077) D0000 D0100 #0300

DMPX(077) D0000 D0100 #0000

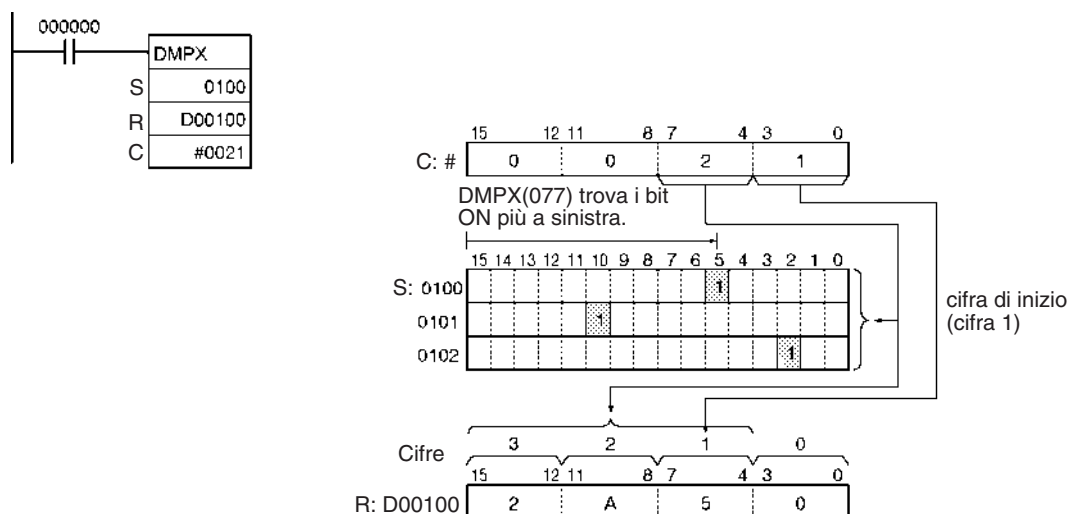
DMPX(077) D0001 D0100 #0001

DMPX(077) D0002 D0100 #0002

DMPX(077) D0003 D0100 #0003

Esempi

Nell'esempio seguente, quando CIO 000000 è ON, DMPX(077) troverà i bit più a sinistra impostati su ON in CIO 0100, CIO 0101 e in CIO 0102 e scriverà queste posizioni nelle 3 cifre in R, cominciando con la cifra 1 (la seconda cifra), come indicato da C (#0021).

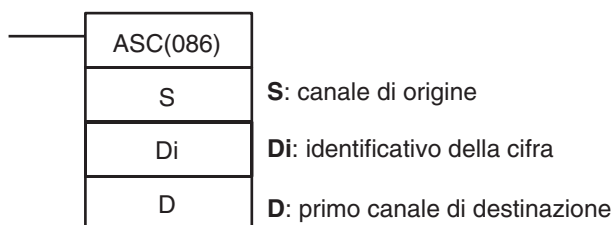


3-12-10 ASCII CONVERT: ASC(086)

Scopo

Converte cifre esadecimali a 4 bit nel canale di origine negli equivalenti ASCII a 8 bit.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ASC(086) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASC(086) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

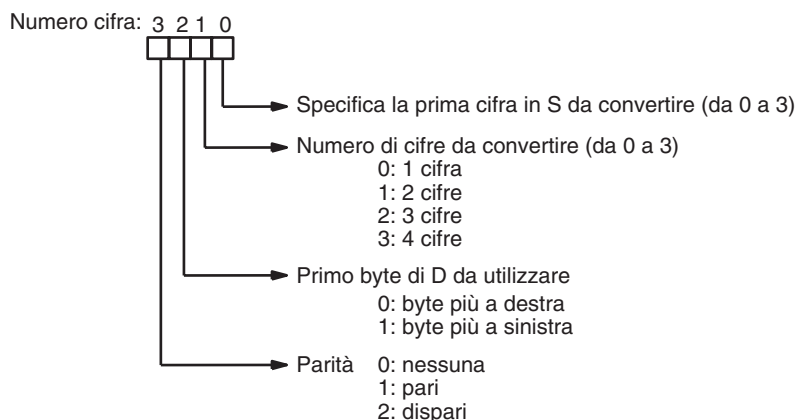
Operandi

S: canale di origine

Nel canale di origine è possibile convertire fino a quattro cifre. Le cifre sono numerate da 0 a 3, da destra a sinistra.

Di: identificativo della cifra

L'identificativo della cifra specifica vari parametri di conversione, come illustrato nello schema seguente.

**D: Primo canale di destinazione**

I dati ASCII convertiti vengono scritti nel canale o nei canali di destinazione, cominciando con il byte specificato in D. Se si intende convertire 4 cifre e il byte più a sinistra è selezionato come primo byte in D, sono necessari tre canali di destinazione (da D a D+3). I canali di destinazione devono essere nella stessa area dati.

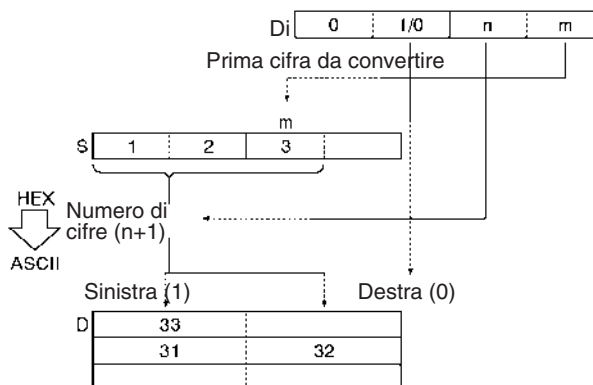
Tutti i byte nei canali di destinazione che non vengono sovrascritti con i dati ASCII resteranno inalterati.

Caratteristiche operando

| Area | S | Di | D |
|--|--|-------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | --- |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

ASC(086) gestisce il contenuto di S come 4 cifre esadecimali, converte la cifra o le cifre specificate di S nell'equivalente ASCII a 8 bit e scrive questi dati nel canale o nei canali di destinazione, cominciando con il byte specificato in D.



Nota Per una tabella dei caratteri ASCII estesi, consultare l'Appendice A del manuale *CS/CJ-series Programming Consoles Operation Manual (W341)*.

Parità

È possibile specificare la parità dei dati ASCII per l'utilizzo nel controllo degli errori durante le trasmissioni di dati. Il bit più a sinistra di ciascun carattere ASCII verrà impostato automaticamente per pari, dispari o nessuna parità.

Quando l'impostazione è nessuna parità (0), il bit più a sinistra sarà sempre zero. Quando l'impostazione è parità pari (1), il bit più a sinistra verrà impostato in modo che il numero totale di bit ON sia pari. Quando l'impostazione è parità dispari (2), il bit più a sinistra di ciascun carattere ASCII verrà impostato in modo che vi sia un numero dispari di bit ON. Lo stato del bit di parità non influisce sul significato del codice ASCII.

Esempi di parità pari:

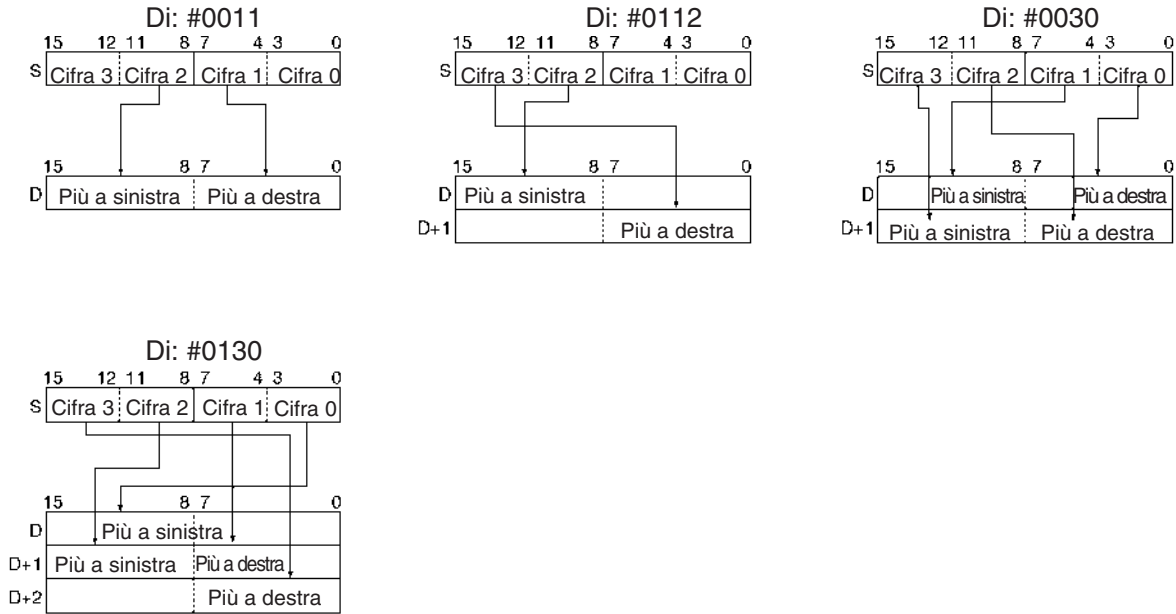
Quando è regolato per parità pari, ASCII "31" (00110001) sarà "B1" (10110001: bit di parità ON per creare un numero pari di bit ON); ASCII "36" (00110110) sarà "36" (00110110: il bit di parità resta OFF poiché il numero di bit ON è già pari).

Esempi di parità dispari:

Quando è regolato per parità dispari, ASCII "36" (00110110) sarà "B6" (10110110: bit di parità ON per creare un numero dispari di bit ON); ASCII "46" (01000110) sarà "46" (01000110: il bit di parità resta OFF poiché il numero di bit ON è già dispari).

Esempi di Di

Quando vengono convertite due o più cifre, ASC(086) leggerà i byte in S da destra verso sinistra, quindi tornerà a capo al byte più a destra, se necessario. Nello schema riportato di seguito sono illustrati alcuni valori di esempio per Di e le conversioni prodotte.

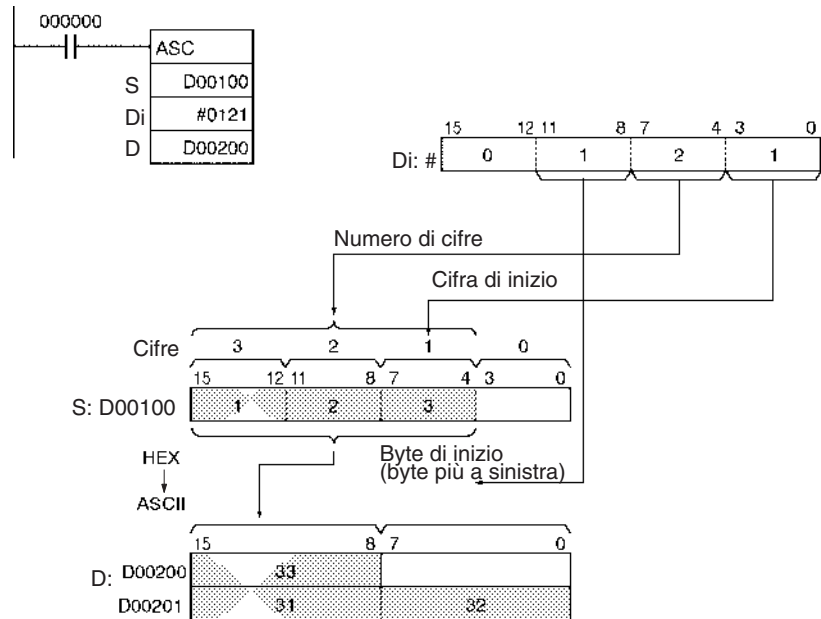


Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|--|
| Flag di errore | ER | ON se il contenuto di Di non è incluso negli intervalli specificati. OFF in tutti gli altri casi. |

Esempio

Nell'esempio seguente, quando CIO 000000 è ON, ASC(086) converte tre cifre esadecimali in D00100 (cominciando con la cifra 1) nei corrispettivi ASCII e scrive questi dati in D00200 e in D00201, cominciando con il byte più a sinistra in D00200. In questo caso, un identificativo della cifra di #0121 specifica nessuna parità, il byte di inizio (per la scrittura) = byte più a sinistra, il numero di cifre da leggere = 3 e la cifra di inizio (per la lettura) = cifra 1.

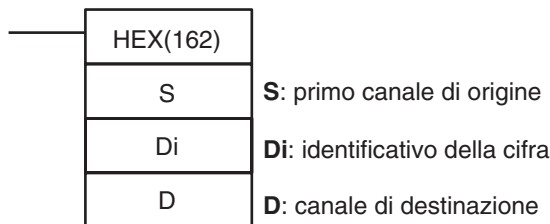


3-12-11 ASCII TO HEX: HEX(162)

Scopo

Converte fino a 4 byte di dati ASCII all'interno del canale di origine negli equivalenti esadecimali e scrive tali cifre nel canale di destinazione specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | HEX(162) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @HEX(162) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

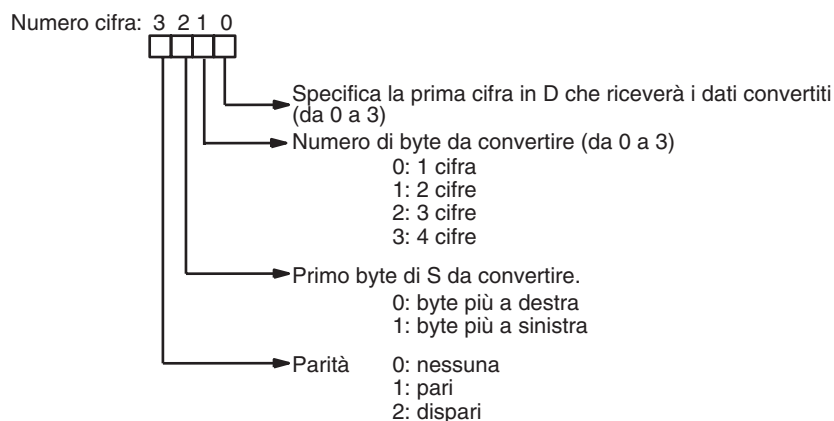
Operandi

S: primo canale di origine

Il contenuto dei canali di origine viene interpretato come dati ASCII. È possibile utilizzare fino a tre canali di origine. Se si intende convertire 4 byte e il byte più a sinistra è selezionato come primo byte in S, sono necessari tre canali di origine. I canali di origine devono essere nella stessa area dati.

Di: identificativo della cifra

L'identificativo della cifra specifica vari parametri di conversione, come illustrato nello schema seguente.



D: canale di destinazione

Le cifre esadecimali convertite vengono scritte in D da destra a sinistra, cominciando con la prima cifra specificata. Tutte le cifre nel canale di destinazione che non vengono sovrascritte con i dati convertiti resteranno inalterate.

Caratteristiche operando

| Area | S | Di | D |
|--|--|-------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | --- |
| Registri dati | --- | Da DR0 a DR15 | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

HEX(162) interpreta il contenuto del canale o dei canali di origine come dati ASCII che rappresentano le cifre esadecimali (da 0 a 9 e da A a F), converte in valori esadecimali il numero specificato di byte e scrive i dati esadecimali nel canale di destinazione, cominciando con la cifra specificata.

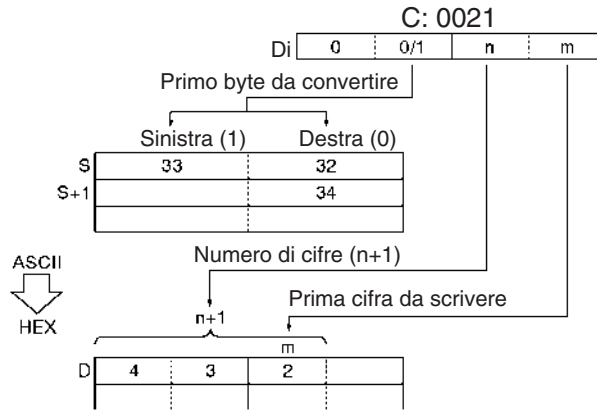
Se i canali di origine contengono dati che non corrispondono all'equivalente ASCII delle cifre esadecimali, verrà generato un errore. Nella tabella seguente sono riportate le cifre esadecimali e gli equivalenti ASCII, esclusi i bit di parità.

Flag

| Cifre esadecimali (4 bit) | Equivalente ASCII (2 cifre esadecimali) |
|---------------------------|---|
| Da 0 a 9 | Da 30 a 39 |
| Da A a F | Da 41 a 46 |

Nota Per una tabella dei caratteri ASCII estesi, consultare l'Appendice A del manuale *CS/CJ-series Programming Consoles Operation Manual (W341)*.

Nello schema seguente è illustrato il funzionamento di base di HEX(162) con Di=0021.



Parità

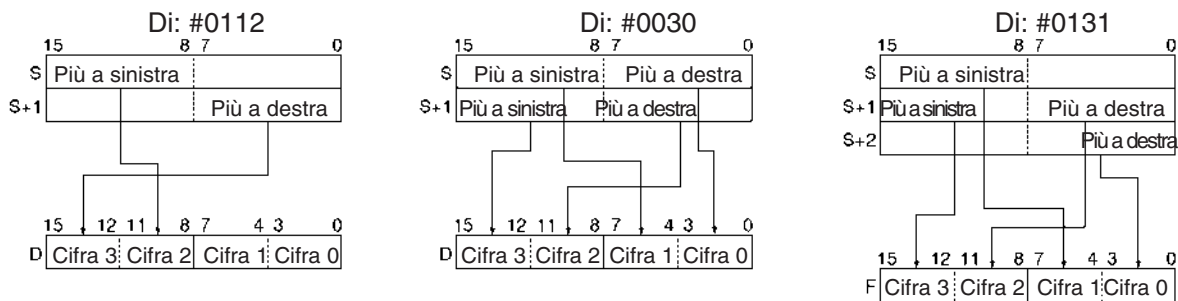
È possibile specificare la parità dei dati ASCII per l'utilizzo nel controllo degli errori durante le trasmissioni di dati. In ciascun byte, il bit più a sinistra è il bit di parità. Quando è impostata nessuna parità, il bit di parità dovrebbe sempre essere zero, con parità pari lo stato del bit di parità dovrebbe dare come risultato un numero pari di bit ON e con parità dispari lo stato del bit di parità dovrebbe dare come risultato un numero dispari di bit ON.

Nella tabella seguente è illustrato il funzionamento di HEX(162) per ogni impostazione di parità.

| Impostazione parità (cifra più a sinistra di Di) | Funzionamento di HEX(162) |
|--|--|
| Nessuna parità (0) | L'istruzione HEX(162) verrà eseguita soltanto quando il bit di parità è 0 in ogni byte. Se il bit di parità è diverso da zero, verrà generato un errore. |
| Parità pari (1) | L'istruzione HEX(162) verrà eseguita soltanto quando ogni byte contiene un numero pari di bit ON. Se in un byte il numero di bit ON è dispari, verrà generato un errore. |
| Parità dispari (2) | L'istruzione HEX(162) verrà eseguita soltanto quando ogni byte contiene un numero dispari di bit ON. Se in un byte il numero di bit ON è pari, verrà generato un errore. |

Esempi di Di

Quando vengono convertiti due o più byte, HEX(162) scriverà le cifre convertite da destra verso sinistra nel canale di destinazione, quindi tornerà a capo alla cifra più a destra, se necessario. Nello schema riportato di seguito sono illustrati alcuni valori di esempio per Di e le conversioni prodotte.



Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se si verifica un errore di parità nei dati ASCII. ON se i dati ASCII nei canali di origine non equivalgono alle cifre esadecimali. ON se il contenuto di Di non è incluso negli intervalli specificati. OFF in tutti gli altri casi. |

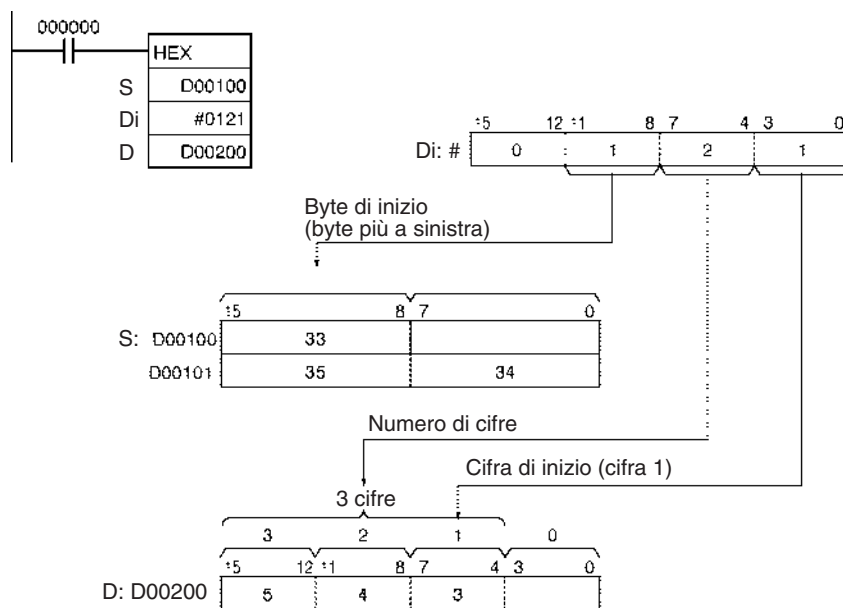
Avvertenze

Verrà generato un errore e il flag di errore verrà attivato se si verifica un errore di parità nei dati ASCII, se i dati ASCII nei canali di origine non equivalgono alle cifre esadecimali o se il contenuto di Di non è incluso negli intervalli specificati.

Esempi

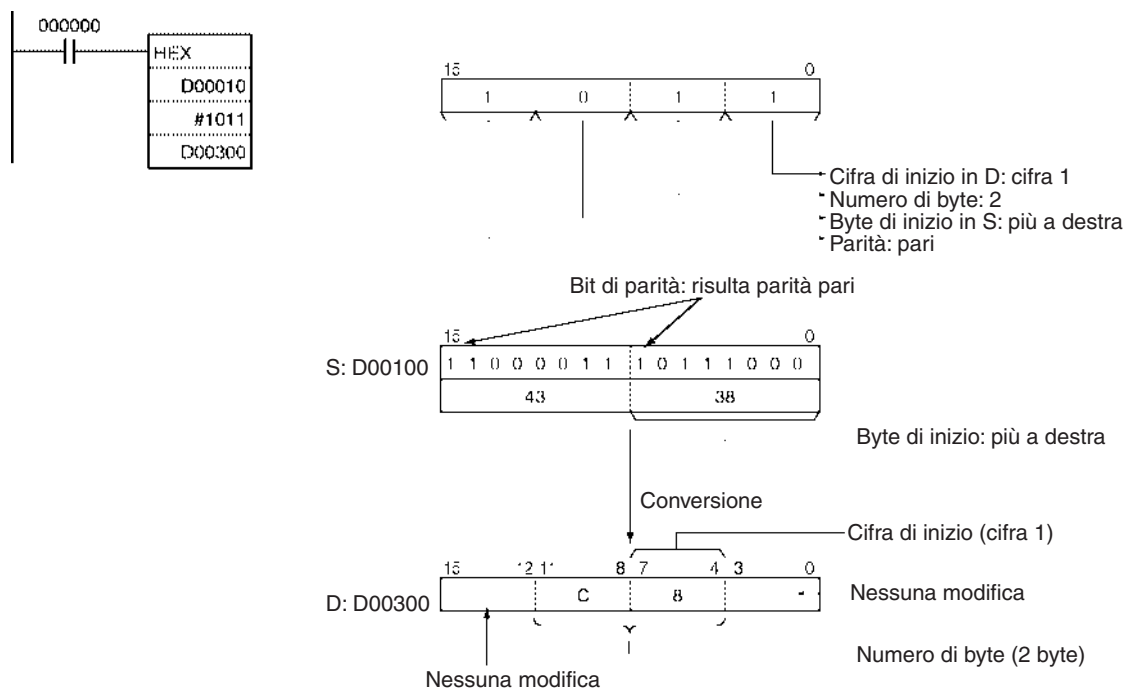
Nel seguente esempio, quando CIO 000000 è ON, HEX(162) convertirà i dati ASCII in D00100 e D00101 in base alle impostazioni dell'identificativo della cifra. D=#0121 specifica nessuna parità, il byte di inizio (per la lettura) = byte più a sinistra, il numero di byte da leggere = 3 e la cifra di inizio (per la scrittura) = cifra 1.

HEX(162) converte tre byte dei dati ASCII (3 caratteri), cominciando dal byte più a sinistra di D00100, negli equivalenti esadecimali e scrive tali dati in D00200, cominciando dalla cifra 1.



Nell'esempio seguente, quando CIO 000000 è ON, HEX(162) converte i dati ASCII in D00010 cominciando dal byte più a destra e scrive gli equivalenti esadecimali in D00300 cominciando con la cifra 1.

L'impostazione dell'identificativo della cifra di #1011 specifica parità pari, il byte di inizio (per la lettura) = byte più a destra, il numero di byte da leggere = 2 e la cifra di inizio (per la scrittura) = cifra 1.

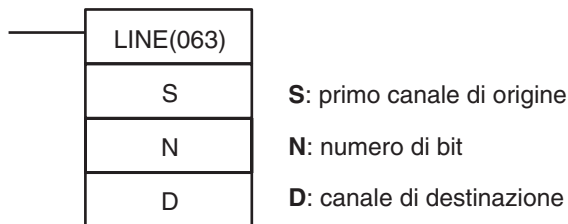


3-12-12 COLUMN TO LINE: LINE(063)

Scopo

Converte una colonna di bit all'interno di un intervallo di 16 canali (lo stesso numero di bit in 16 canali consecutivi) nei 16 bit del canale di destinazione

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | LINE(063) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @LINE(063) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: primo canale di origine

Specifica il primo canale di origine. S e S+15 devono essere nella stessa area dati.

N: numero di bit

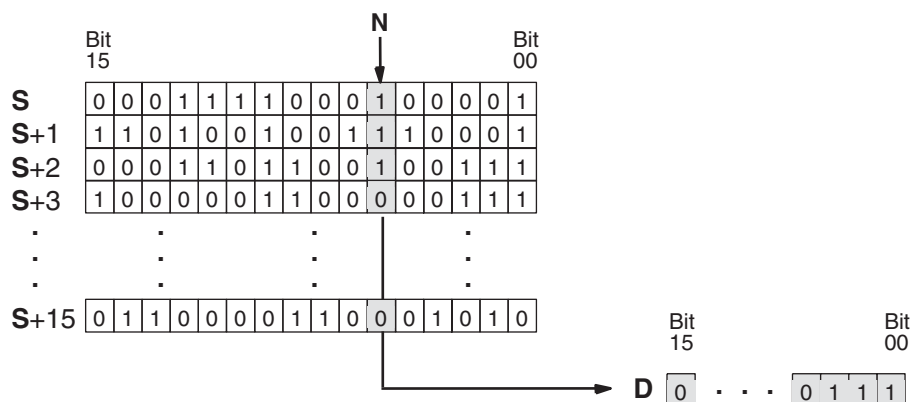
Specifica il numero di bit (da 0000 a 000F o da &0 a &15) da copiare dai canali di origine.

Caratteristiche operando

| Area | S | N | D |
|--|--|---|----------------|
| Area CIO | Da CIO 0000 a CIO 6128 | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W496 | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H496 | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A944 | Da A000 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4080 | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4080 | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32752 | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32752 | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32752 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a 000F (binario) o da &0 a &15 | --- |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

LINE(063) copia i 16 bit con il numero N dall'intervallo di 16 canali da S a S+15 nel canale di destinazione D. Il bit N di S+m viene copiato nel bit m di D, ossia il bit N di S viene copiato nel bit 00 di D e il bit N di S+15 nel bit 15 di D.

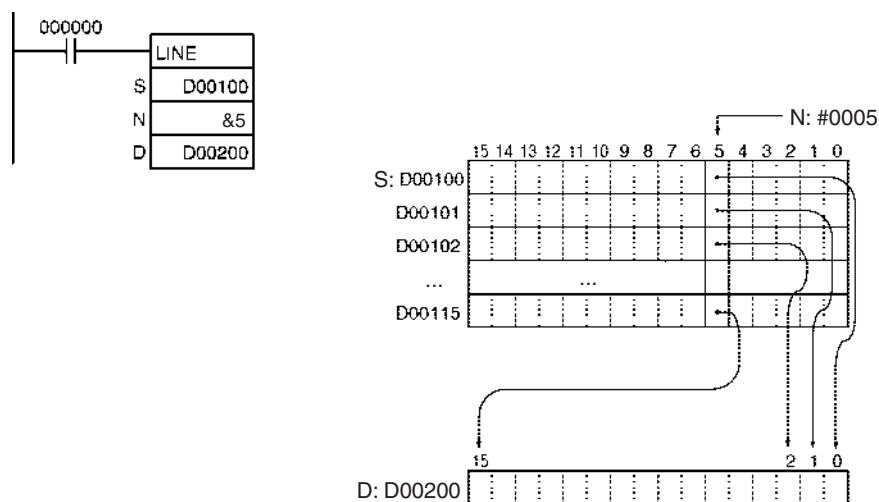


Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato 0000 - 000F. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se D è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempio

Nell'esempio seguente, quando CIO 000000 è ON, LINE(063) copia il bit 5 da D00100 a D00115 nei 16 bit in D00200.

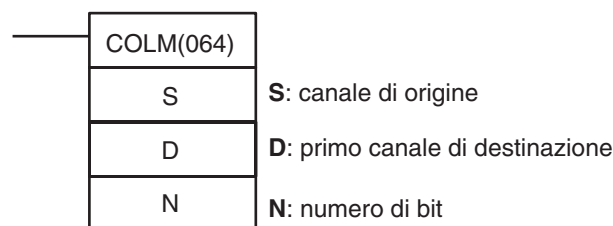


3-12-13 LINE TO COLUMN: COLM(064)

Scopo

Converte i 16 bit del canale di origine in una colonna di bit all'interno di un intervallo di 16 canali di destinazione (lo stesso numero di bit in 16 canali consecutivi).

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | COLM(064) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @COLM(064) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

D: Primo canale di destinazione

Specifica il primo canale di destinazione. D e D+15 devono essere nella stessa area dati.

N: numero di bit

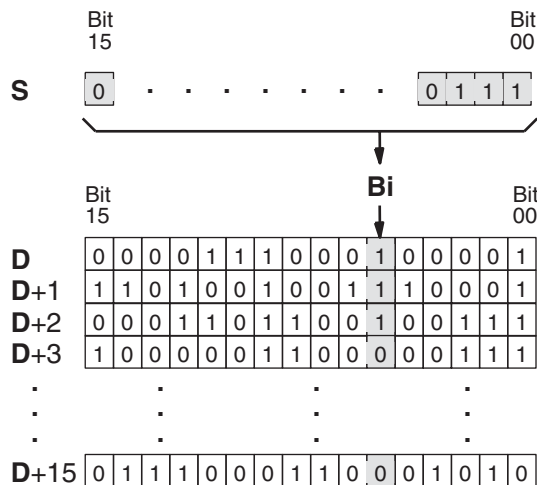
Specifica il numero di bit (da 0000 a 000F o da &0 a &15) che verranno sovrascritti dal canale di origine.

Caratteristiche operando

| Area | S | D | N |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6128 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W496 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H496 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A944 | Da A000 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4080 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4080 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32752 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32752 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32752 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | Da #0000 a #000F (binario) o da &0 a &15 |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

COLM(064) copia i 16 bit da S ai 16 bit con il numero di bit N nell'intervallo di 16 canali da D a D+15. Il bit m di S viene copiato nel bit N di D+m, ossia il bit 00 di S viene copiato nel bit N di D e il bit 15 di S nel bit N di D+15.

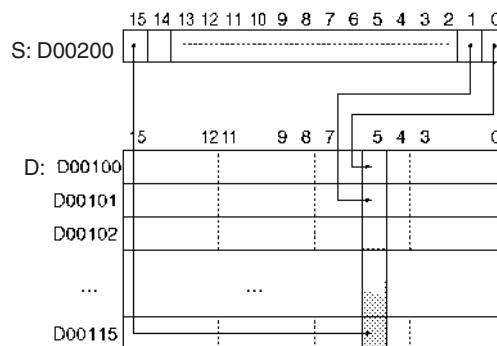
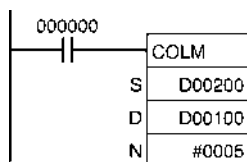


Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato 0000 - 000F. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il bit N è 0 in tutti i 16 canali da D a D+15 dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempio

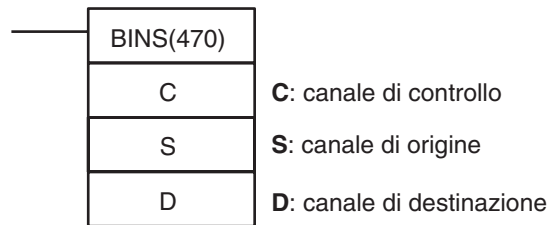
Nell'esempio seguente, quando CIO 000000 è ON, COLM copia i 16 bit in D00200 (i bit da 00 a 15) nel bit 5 da D00100 a D00115.



3-12-14 SIGNED BCD-TO-BINARY: BINS(470)

Scopo Convertere un canale di dati in formato BCD con segno in un canale di dati binari con segno.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BINS(470) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BINS(470) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo

Specifica il formato BCD con segno. C deve essere da 0000 a 0003.

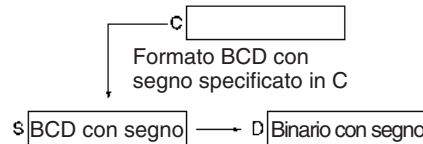
Caratteristiche operando

| Area | C | S | D |
|--------------------------------------|--|-----|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #0003 (binario) | --- | |
| Registri dati | Da DR0 a DR15 | | |

| Area | C | S | D |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

BINS(470) converte i dati BCD con segno in dati binari con segno. Innanzitutto il formato e l'intervallo dei dati BCD con segno nel canale S vengono controllati e confrontati con le impostazioni del canale di controllo (C). Se i dati di origine sono corretti, i dati BCD con segno in S vengono convertiti in dati binari con segno e inviati a D. Se i dati di origine non sono corretti, viene attivato il flag di errore e l'istruzione non verrà eseguita.



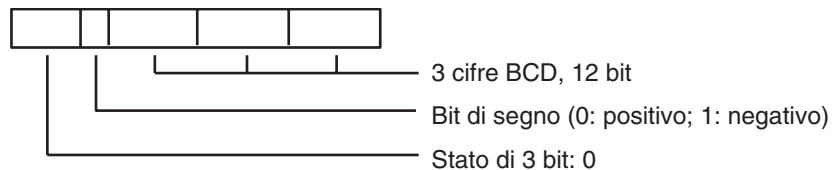
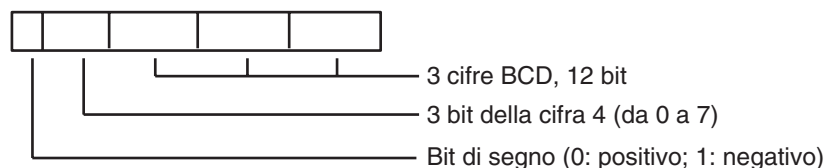
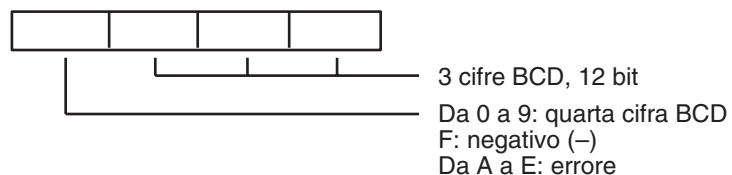
Quando i dati convertiti sono negativi, verranno inviati come complemento a 2 e verrà attivato il flag negativo. È possibile utilizzare l'istruzione NEG(160) per determinare il valore assoluto di un numero binario negativo con segno.

Per ulteriori informazioni, consultare la sezione 3-12-52'S COMPLEMENT: NEG(160).

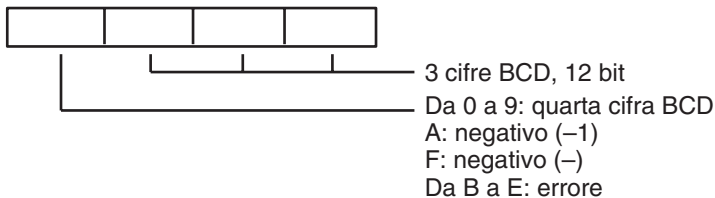
Un valore di -0 nei dati di origine verrà interpretato come 0 e non provocherà alcun errore. Inoltre, lo stato dei bit da 13 a 15 di S non viene controllato quando C=0000.

Nota Alcuni moduli di I/O speciali inviano dati BCD con segno. Generalmente è più semplice eseguire calcoli con questi dati se prima li si converte in dati binari con segno mediante l'istruzione BINS(470).

Il canale di controllo specifica il formato BCD con segno, come mostrato di seguito.

C = 0000 (intervallo dati di ingresso: da -999 a 999 BCD)**C = 0001 (intervallo dati di ingresso: da -7999 a 7999 BCD)****C = 0002 (intervallo dati di ingresso: da -999 a 9999 BCD)**

C = 0003 (intervallo dati di ingresso: da -1999 a 9999 BCD)



Nella tabella seguente sono riportati i valori BCD possibili per ciascun formato BCD con segno e i valori binari con segno corrispondenti.

| Imposta- zione | Valori BCD con segno | Valori binari con segno |
|-------------------|-----------------------------|---------------------------------|
| C=0000 | Da -999 a -1 e da 0 a 999 | Da FC19 a FFFF e da 0000 a 03E7 |
| C=0001 | Da -7999 a -1 e da 0 a 7999 | Da E0C1 a FFFF e da 0000 a 1F3F |
| C=0002 | Da -999 a -1 e da 0 a 9999 | Da FC19 a FFFF e da 0000 a 270F |
| C=0003 | Da -1999 a -1 e da 0 a 9999 | Da F831 a FFFF e da 0000 a 270F |

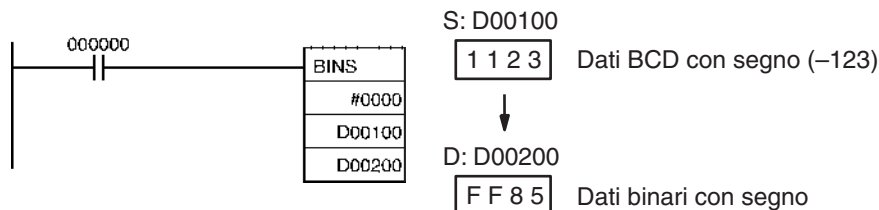
Flag

| Nome | Eti- chetta | Operazione |
|---------------------|----------------|---|
| Flag di errore | ER | ON se C non rientra nell'intervallo specificato 0000 - 0003. ON se C=0002 e la cifra più a sinistra di S è da A a E. ON se C=0003 e la cifra più a sinistra di S è da B a E. ON se il contenuto di S non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se D è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di D è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempi

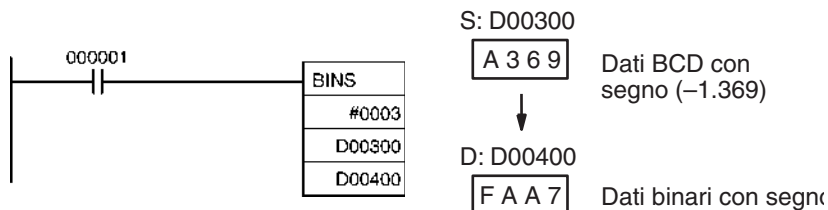
Formato BCD 0 (C=#0000)

Nell'esempio seguente, quando CIO 000000 è ON, il formato e l'intervallo dei dati BCD con segno in D00100 vengono controllati e confrontati con il formato specificato nel canale di controllo (0000). Se i dati di origine sono corretti, i dati BCD con segno in D00100 vengono convertiti in dati binari con segno e inviati a D00200.



Formato BCD 0 (C=#0003)

Nell'esempio seguente, quando CIO 000001 è ON, il formato e l'intervallo dei dati BCD con segno in D00300 vengono controllati e confrontati con il formato specificato nel canale di controllo (0003). Se i dati di origine sono corretti, i dati BCD con segno in D00300 vengono convertiti in dati binari con segno e inviati a D00400.



3-12-15 DOUBLE SIGNED BCD-TO-BINARY: BISL(472)

Scopo

Converte dati in formato BCD con segno in doppia precisione in dati binari con segno in doppia precisione.

Simbolo programmazione ladder

| | | |
|---|-----------|---------------------------------|
| — | BISL(472) | |
| | C | C: canale di controllo |
| | S | S: primo canale di origine |
| | D | D: primo canale di destinazione |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BISL(472) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BISL(472) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo

Specifica il formato BCD con segno. C deve essere da 0000 a 0003.

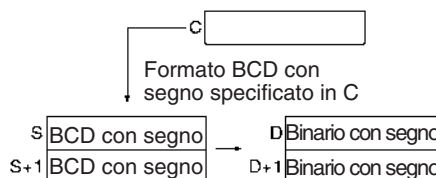
Caratteristiche operando

| Area | C | S | D |
|--------------------------------------|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #0003 (binario) | --- | |

| Area | C | S | D |
|--|--|-----|---|
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

BISL(472) converte i dati in formato BCD con segno in doppia precisione in S+1 e in S in dati binari con segno in doppia precisione e scrive il risultato in D+1 e in D. Innanzitutto il formato e l'intervallo dei dati BCD con segno nei canali S+1 e S vengono controllati e confrontati con le impostazioni del canale di controllo (C). Se i dati di origine sono corretti, i dati BCD con segno in S+1 e S vengono convertiti in dati binari con segno e inviati a D+1 e D. Se i dati di origine non sono corretti, viene attivato il flag di errore e l'istruzione non verrà eseguita.



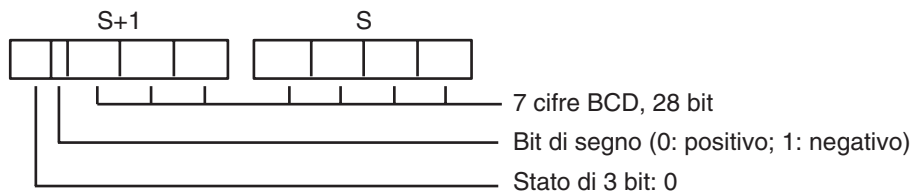
Quando i dati convertiti sono negativi, verranno inviati come complemento a 2 e verrà attivato il flag negativo. È possibile utilizzare l'istruzione NEGL(161) per determinare il valore assoluto di un numero binario negativo doppio con segno. Per ulteriori informazioni, consultare la sezione 3-12-6 *DOUBLE 2'S COMPLEMENT: NEGL(161)*.

I valori di -0 nei dati di origine verranno interpretati come 0 e non provocheranno alcun errore. Inoltre, lo stato dei bit da 13 a 15 di S+1 non viene controllato quando C=0000.

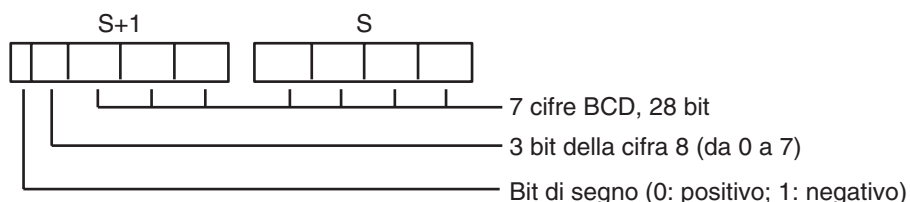
Nota Alcuni moduli di I/O speciali inviano dati BCD con segno. Generalmente è più semplice eseguire calcoli con questi dati se prima li si converte in dati binari con segno mediante l'istruzione BILS(472).

Il canale di controllo specifica il formato BCD con segno, come mostrato di seguito.

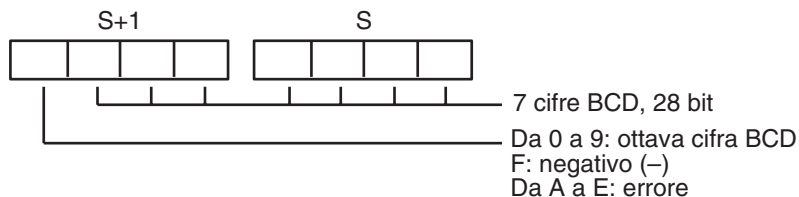
C = 0000 (intervallo dati di ingresso: da -999 9999 a 999 9999 BCD)



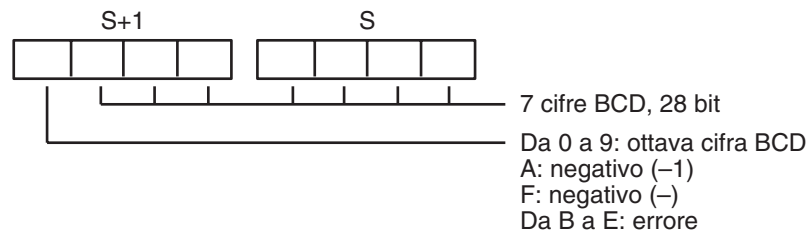
C = 0001 (intervallo dati di ingresso: da -7999 9999 a 7999 9999 BCD)



C = 0002 (intervallo dati di ingresso: da -999 9999 a 9999 9999 BCD)



C = 0003 (intervallo dati di ingresso: da -1999 9999 a 9999 9999 BCD)



Nella tabella seguente sono riportati i valori BCD possibili per ciascun formato BCD con segno e i valori binari con segno corrispondenti.

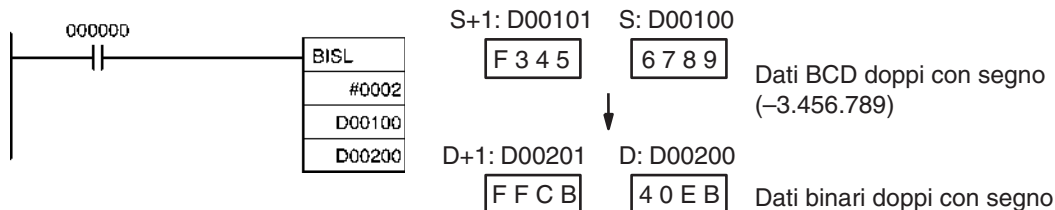
| Impostazione | Valori BCD con segno | Valori binari con segno |
|--------------|----------------------|--------------------------|
| C=0000 | Da -999 9999 a -1 | Da FF67 6981 a FFFF FFFF |
| | Da 0 a 999 9999 | Da 0000 0000 a 0098 967F |
| C=0001 | Da -7999 9999 a -1 | Da FB3B 4C01 a FFFF FFFF |
| | Da 0 a 7999 9999 | Da 0000 0000 a 04C4 B3FF |
| C=0002 | Da -999 9999 a -1 | Da FF67 6981 a FFFF FFFF |
| | Da 0 a 9999 9999 | Da 0000 0000 a 05F5 E0FF |
| C=0003 | Da -1999 9999 a -1 | Da FECE D301 a FFFF FFFF |
| | Da 0 a 9999 9999 | Da 0000 0000 a 05F5 E0FF |

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se C non rientra nell'intervallo specificato 0000 - 0003. ON se C=0002 e la cifra più a sinistra di S+1 è da A a E. ON se C=0003 e la cifra più a sinistra di S+1 è da B a E. ON se il contenuto di S+1 e S non è in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se D+1 contiene 0000 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di D+1 è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

Esempio

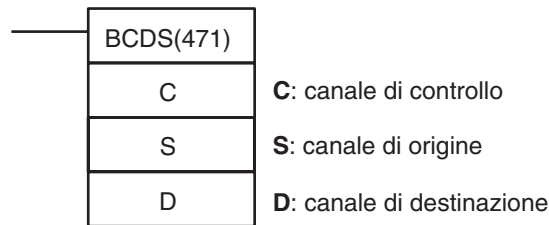
Nell'esempio seguente, quando CIO 000000 è ON, il formato e l'intervallo dei dati BCD doppi con segno in D00101 e in D00100 vengono controllati e confrontati con il formato specificato nel canale di controllo (0002). Se i dati di origine sono corretti, i dati BCD doppi con segno in D00101 e D00100 vengono convertiti in dati binari doppi con segno e inviati a D00201 e D00200.



3-12-16 SIGNED BINARY-TO-BCD: BCDS(471)

Scopo Convertere un canale di dati binari con segno in un canale di dati in formato BCD con segno.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BCDS(471) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BCDS(471) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operando

C: canale di controllo

Specifica il formato BCD con segno. C deve essere da 0000 a 0003.

S: canale di origine

Contiene i dati binari con segno da convertire. Il contenuto di S deve essere compreso nell'intervallo valido del formato BCD specificato in C.

| Impostazione | Valori ammessi per S |
|--------------|---------------------------------|
| C=0000 | Da FC19 a FFFF o da 0000 a 03E7 |
| C=0001 | Da E0C1 a FFFF o da 0000 a 1F3F |
| C=0002 | Da FC19 a FFFF o da 0000 a 270F |
| C=0003 | Da F831 a FFFF o da 0000 a 270F |

D: canale di destinazione

Contiene i dati BCD con segno convertiti. Per una spiegazione dei formati BCD, consultare la descrizione fornita più avanti.

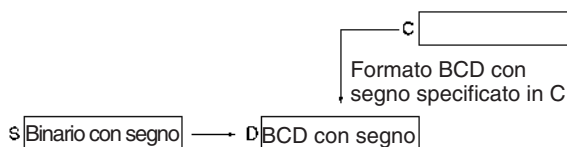
Caratteristiche operando

| Area | C | S | D |
|-------------------------|--|---|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |

| Area | C | S | D |
|--|--|-----|---|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #0003 (binario) | --- | |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047,IR0 a, da -2048 a +2047,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

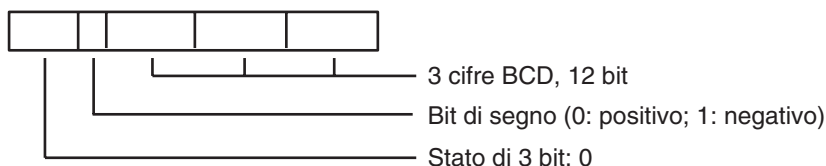
BCDS(471) converte i dati binari con segno in dati BCD con segno. Innanzitutto i dati binari con segno nel canale S vengono controllati per verificare che rientrino nell'intervallo valido del formato BCD con segno specificato nel canale di controllo (C). Se i dati di origine sono corretti, i dati binari con segno in S vengono convertiti in dati BCD con segno e inviati a D. Se i dati di origine non sono corretti, viene attivato il flag di errore e l'istruzione non verrà eseguita.



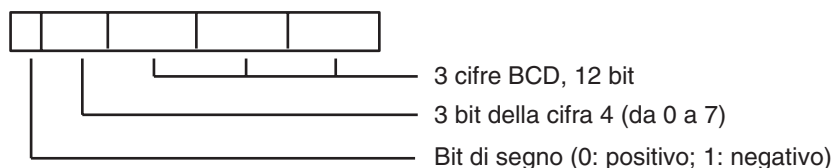
- Nota**
1. I valori di -0 nei dati di origine verranno interpretati come 0 e non provocheranno alcun errore.
 2. Alcuni moduli di I/O speciali necessitano di ingressi di dati BCD con segno. È possibile utilizzare l'istruzione BCDS(471) per convertire i dati binari con segno per l'invio a tali moduli.

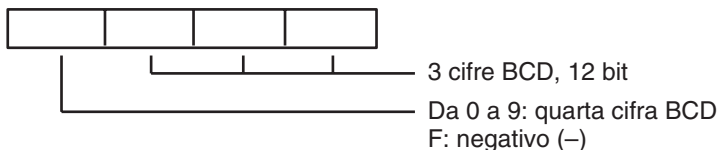
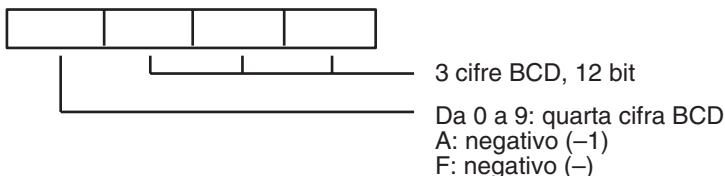
Il canale di controllo specifica il formato BCD con segno che verrà usato per il risultato, come mostrato di seguito.

C = 0000 (intervallo dati di uscita: da -999 a 999 BCD)



C = 0001 (intervallo dati di uscita: da -7999 a 7999 BCD)



C = 0002 (intervallo dati di uscita: da -999 a 9999 BCD)**C = 0003 (intervallo dati di uscita: da -1999 a 9999 BCD)**

Nella tabella seguente sono riportati i valori binari con segno possibili per ciascun formato BCD con segno. Se i dati di origine non rientrano nell'intervallo ammesso per il formato BCD con segno specificato, si verificherà un errore.

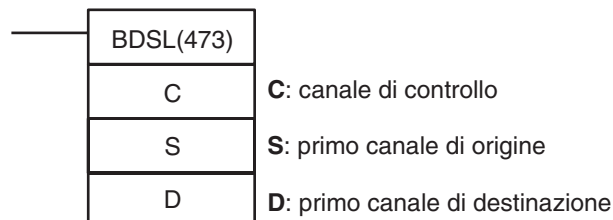
| Imposta- zione | Valori binari con segno | Valori BCD con segno |
|-------------------|---------------------------------|-----------------------------|
| C=0000 | Da FC19 a FFFF e da 0000 a 03E7 | Da -999 a -1 e da 0 a 999 |
| C=0001 | Da E0C1 a FFFF e da 0000 a 1F3F | Da -7999 a -1 e da 0 a 7999 |
| C=0002 | Da FC19 a FFFF e da 0000 a 270F | Da -999 a -1 e da 0 a 9999 |
| C=0003 | Da F831 a FFFF e da 0000 a 270F | Da -1999 a -1 e da 0 a 9999 |

Flag

| Nome | Eti- chetta | Operazione |
|---------------------|----------------|--|
| Flag di errore | ER | ON se C non rientra nell'intervallo specificato 0000 - 0003. ON se C=0000 e i dati di origine non rientrano negli intervalli consentiti (FC19 - FFFF o 0000 - 03E7). ON se C=0001 e i dati di origine non rientrano negli intervalli consentiti (E0C1 - FFFF o 0000 - 1F3F). ON se C=0002 e i dati di origine non rientrano negli intervalli consentiti (FC19 - FFFF o 0000 - 270F). ON se C=0003 e i dati di origine non rientrano negli intervalli consentiti (F831 - FFFF o 0000 - 270F). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se D è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se C=0000 o 0001 e il bit di segno del risultato è ON dopo l'esecuzione. ON se C=0002 e la cifra più a sinistra del risultato è F. ON se C=0003 e la cifra più a sinistra del risultato è A o F. OFF in tutti gli altri casi. |

3-12-17 DOUBLE SIGNED BINARY-TO-BCD: BDSL(473)**Scopo**

Converte dati binari con segno in doppia precisione in dati in formato BCD con segno in doppia precisione.

Simbolo programmazione ladder

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BDSL(473) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BDSL(473) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Operandi

C: canale di controllo

Specifica il formato BCD con segno. C deve essere da 0000 a 0003.

S: primo canale di origine

I canali di origine S+1 e S contengono dati binari doppi con segno da convertire. Il contenuto di tali canali deve essere compreso nell'intervallo valido del formato BCD specificato in C.

| Impostazione | Valori ammessi per S+1 e S |
|--------------|---|
| C=0000 | Da FF67 6981 a FFFF FFFF o da 0000 0000 a 0098 967F |
| C=0001 | Da FB3B 4C01 a FFFF FFFF o da 0000 0000 a 04C4 B3FF |
| C=0002 | Da FF67 6981 a FFFF FFFF o da 0000 0000 a 05F5 E0FF |
| C=0003 | Da FECE D301 a FFFF FFFF o da 0000 0000 a 05F5 E0FF |

D: primo canale di destinazione

I canali D+1 e D contengono i dati BCD doppi con segno convertiti. Per una spiegazione dei formati BCD, consultare la descrizione fornita più avanti.

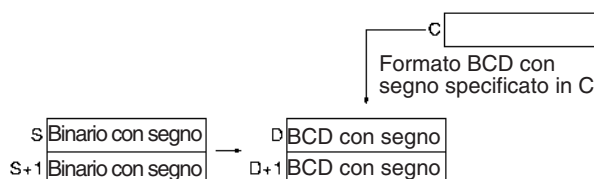
Caratteristiche operando

| Area | C | S | D |
|--------------------------------------|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #0003 (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | |

| Area | C | S | D |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

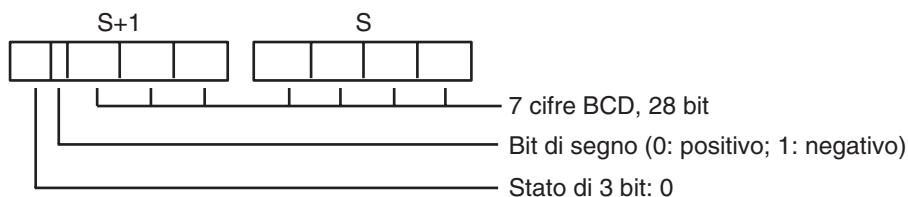
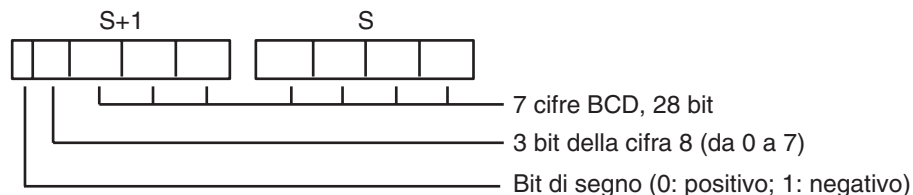
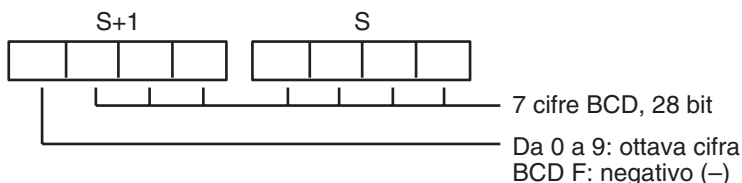
Descrizione

BDSL(473) converte dati binari con segno in doppia precisione in dati in formato BCD con segno in doppia precisione. Innanzitutto i dati binari doppi con segno in S+1 e S vengono controllati per verificare che rientrino nell'intervallo valido del formato BCD con segno specificato nel canale di controllo (C). Se i dati di origine sono corretti, i dati binari doppi con segno in S+1 e S vengono convertiti in dati in formato BCD doppi con segno e inviati a D+1 e D. Se i dati di origine non sono corretti, viene attivato il flag di errore e l'istruzione non verrà eseguita.

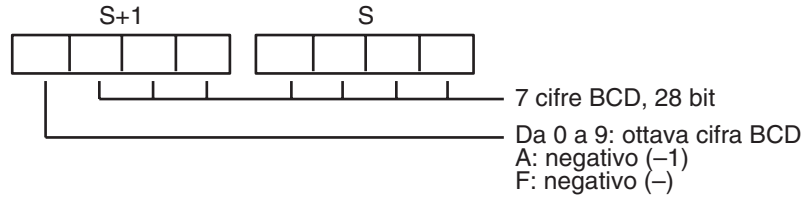


- Nota**
1. I valori di -0 nei dati di origine verranno interpretati come 0 e non provocheranno alcun errore.
 2. Alcuni moduli di I/O speciali necessitano di ingressi di dati BCD con segno. È possibile utilizzare l'istruzione BDSL(473) per convertire i dati binari doppi con segno per l'invio a tali moduli.

Il canale di controllo specifica il formato BCD con segno, che verrà usato per il risultato come mostrato di seguito.

C = 0000 (intervallo dati di uscita: da -999 9999 a 999 9999 BCD)**C = 0001 (intervallo dati di uscita: da -7999 9999 a 7999 9999 BCD)****C = 0002 (intervallo dati di uscita: da -999 9999 a 9999 9999 BCD)**

C = 0003 (intervallo dati di uscita: da -1999 9999 a 9999 9999 BCD)



Nella tabella seguente sono riportati i valori binari doppi con segno possibili per ciascun formato BCD con segno. Se i dati di origine non rientrano nell'intervallo ammesso per il formato BCD con segno specificato, si verificherà un errore.

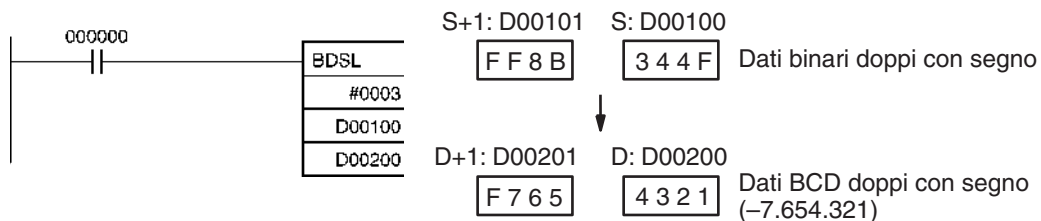
| Impostazione | Valori binari con segno | Valori BCD con segno |
|--------------|------------------------------|----------------------|
| C=0000 | Da FF67 6981 a FFFF FFFF | Da -999 9999 a -1 |
| | 0000 0000 a 0098 967F | Da 0 a 999 9999 |
| C=0001 | Da FB3B 4C01 a FFFF FFFF | Da -7999 9999 a -1 |
| | 0000 0000 a 04C4 B3FF | Da 0 a 7999 9999 |
| C=0002 | Da FF67 6981 a FFFF FFFF | Da -999 9999 a -1 |
| | 0000Da 0000 0000 a 05F5 E0FF | Da 0 a 9999 9999 |
| C=0003 | Da FECE D301 a FFFF FFFF | Da -1999 9999 a -1 |
| | 0000Da 0000 0000 a 05F5 E0FF | Da 0 a 9999 9999 |

Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|--|
| Flag di errore | ER | ON se C non rientra nell'intervallo specificato 0000 - 0003. ON se C=0000 e i dati di origine non sono inclusi nell'intervallo:da FF67 6981 a FFFF FFFF o da 0000 0000 a 0098 967F. ON se C=0001 e i dati di origine non sono inclusi nell'intervallo:da FB3B 4C01 a FFFF FFFF o da 0000 0000 a 04C4 B3FF. ON se C=0002 e i dati di origine non sono inclusi nell'intervallo:da FF67 6981 a FFFF FFFF o da 0000 0000 a 05F5 E0FF. ON se C=0003 e i dati di origine non sono inclusi nell'intervallo:da FECE D301 a FFFF FFFF o da 0000 0000 a 05F5 E0FF. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se D è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se C=0000 o 0001 e il bit di segno del risultato è ON dopo l'esecuzione. ON se C=0002 e la cifra più a sinistra del risultato è F. ON se C=0003 e la cifra più a sinistra del risultato è A o F. OFF in tutti gli altri casi. |

Esempio

Nell'esempio seguente, quando CIO 000000 è ON, i dati binari doppi con segno in D00101 e in D00100 vengono controllati e confrontati con il formato specificato nel canale di controllo (0003). Se i dati di origine sono corretti, i dati binari doppi con segno in D00101 e D00100 vengono convertiti in dati BCD doppi con segno e inviati a D00201 e D00200.



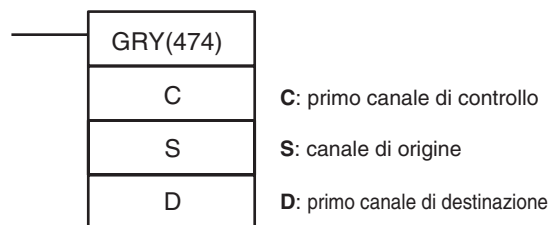
3-12-18 GRAY CODE CONVERT: GRY(474)

Scopo

Converte i dati in codice Gray binario nel canale specificato in dati standard in formato binario, BCD o angolare alla risoluzione specificata.

Questa istruzione è supportata solo dalla CPU della serie CS/CJ versione 2.0 o successiva, comprese le CPU CS1-H, CJ1-H e CJ1M con numero di lotto 030201 e successivo)

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | GRY(474) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @GRY(474) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

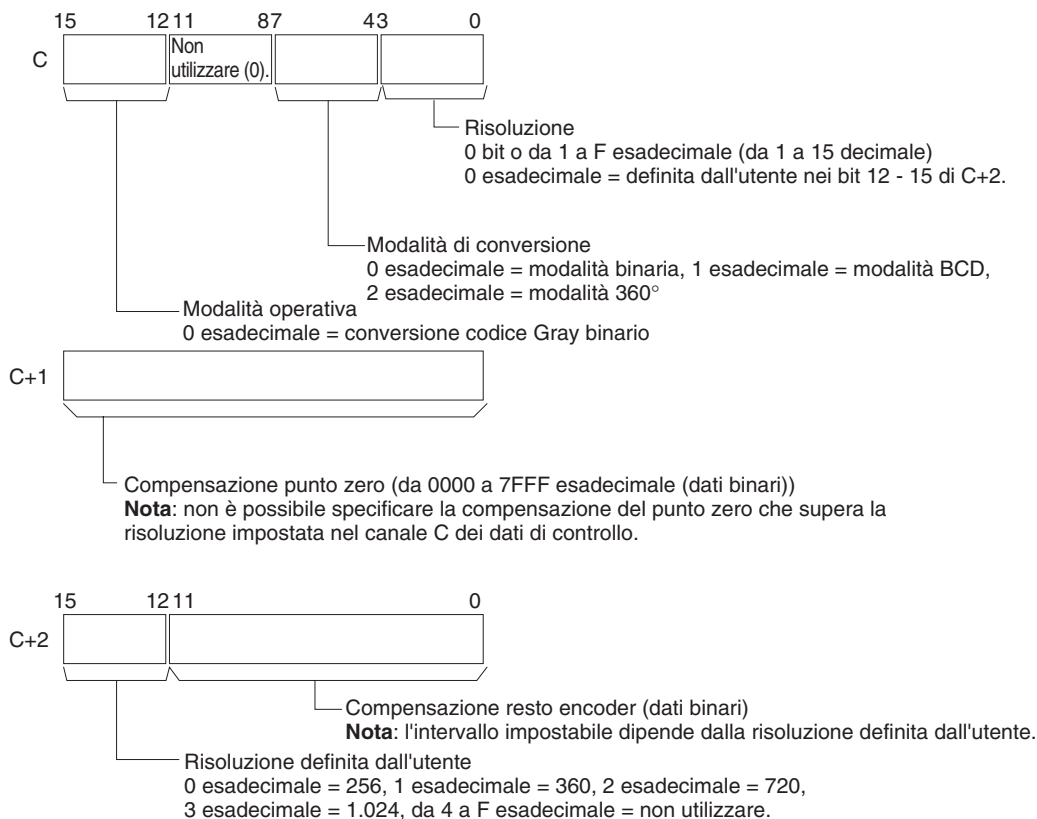
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo

Specifica i parametri per la conversione, come mostrato di seguito.



Nota: la precedente impostazione è valida quando la risoluzione è impostata su 0 esadecimale nei bit 00 - 03 di C.

S: canale di origine

Contiene il codice Gray binario da convertire. L'intervallo deve essere compreso entro il numero di bit determinato dalla risoluzione specificata nei bit 00 - 03 di C. Verranno ignorati tutti i bit non inclusi nel numero di bit della risoluzione specificata. Ad esempio, se la risoluzione specificata è 08 esadecimale e S contiene FFFF esadecimale, il codice Gray binario verrà considerato come 00FF esadecimale.

S

D: primo canale di destinazione

I canali di destinazione D+1 e D contengono i risultati della conversione del codice Gray binario alla risoluzione specificata nei bit 00 - 03 del canale C dei dati di controllo e la modalità di conversione specificata nei bit 04 - 07 del canale C dei dati di controllo. Il canale più a sinistra viene inviato a D+1 e il canale più a destra a D. Gli intervalli di dati inviati sono i seguenti:

Modalità binaria: da 0000 0000 a 0000 7FFF esadecimale
 Modalità BCD: da 0000 0000 a 0003 2767
 Modalità 360°: da 0000 0000 a 0000 3599
 (incrementi da 0,0° a 359,9° in 0.1°, BCD)

D Canale all'estrema destra
 D+1 Canale all'estrema sinistra:

Caratteristiche operando

| Area | C | S | D |
|--------------------------------------|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H511 | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A958 | Da A000 a A959 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a #FFFF (binario) | --- |
| Registri dati | --- | Da DR0 a DR15 | --- |

| Area | C | S | D |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

GRY(474) converte il codice Gray binario nel canale specificato in S alla risoluzione definita in C, utilizzando una delle modalità di conversione riportate di seguito (binaria, BCD, o 360°), specificata anch'essa in C, e inserisce i risultati in D e in D+1.

| Modalità di conversione | Funzione |
|-------------------------|---|
| Modalità binaria | Il codice Gray binario viene convertito in dati binari compresi tra 0000 0000 e 0000 7FFF esadecimale. Vengono applicati l'offset del punto zero e la compensazione del resto, quindi il risultato viene inviato a D e D+1. |
| Modalità BCD | Il codice Gray binario viene convertito in dati in formato BCD. Vengono applicati l'offset del punto zero e la compensazione del resto, i dati vengono convertiti in formato BCD da 0000 0000 a 0003 2767, quindi il risultato viene inviato a D e D+1. |
| Modalità 360° | Il codice Gray binario viene convertito in dati in formato BCD. Vengono applicati l'offset del punto zero e la compensazione del resto, i dati vengono convertiti in formato BCD da 0000 0000 a 0000 3599 (incrementi da 0,0° a 359,9° in 0,1°), quindi il risultato viene inviato a D e D+1. |

- Nota**
1. Generalmente l'istruzione GRY(474) viene utilizzata per inserire tramite un modulo di ingresso c.c. un segnale parallelo (2ⁿ) da un encoder assoluto che invia un codice Gray binario.
 2. Se il canale specificato per S è allocato a un modulo di ingresso, i dati di ingresso convertiti da GRY(474) saranno per il codice Gray binario del precedente ciclo della CPU, ossia saranno vecchi di un tempo di ciclo.

Limitazioni

All'istruzione GRY(474) vengono applicate le limitazioni riportate di seguito.

■ Limitazioni sul modulo CPU

È possibile utilizzare GRY(474) soltanto con i seguenti modelli di CPU e solo per CPU prodotte dopo il 1° febbraio 2003 (numero di lotto 030201 o successivo, incluse le CPU versione 2.0 o successiva).

- CJ1M-CPU□□
- CJ1G-CPU□□H
- CJ1H-CPU□□H
- CS1G-CPU□□H
- CS1H-CPU□□H
- CS1D-CPU□□S

Per determinare la data di produzione, verificare il numero di lotto riportato sul lato o sulla parte inferiore della CPU. Nei numeri di lotto, la data di produzione è indicata come segue:

AAMMGG nnnn

AA = le ultime due cifre dell'anno, MM = mese espresso come valore numerico, GG = giorno del mese, nnnn = numero di serie

Nota Se GRY(474) viene trasferita in una CPU che non supporta tale istruzione e il programma viene letto da una console di programmazione, verrà visualizzato un punto interrogativo "?" per GRY(474), ad indicare un'istruzione non valida. Se GRY(474) viene eseguita con una condizione di esecuzione ON in una CPU che non supporta tale istruzione, si verificherà un errore e l'esecuzione del programma verrà interrotta.

■ Limitazioni su CX-Programmer

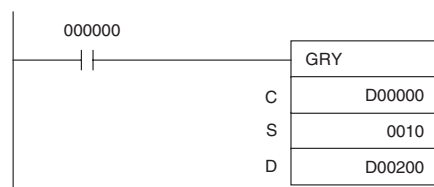
È possibile utilizzare l'istruzione GRY(474) soltanto con CX-Programmer versione 3.2 o successiva.

Flag

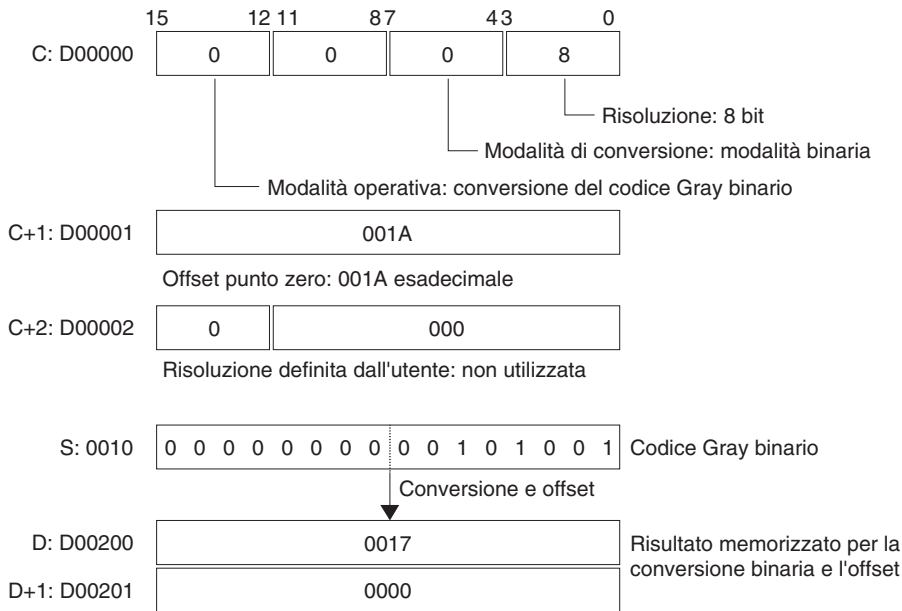
| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i bit 12 - 15 di C non sono 0 esadecimale (modalità operativa = conversione codice Gray binario). ON se l'offset del punto zero in C+1 non rientra tra le risoluzioni specificate (incluse le risoluzioni definite dall'utente). ON se i bit 04 - 07 di C non sono 0 esadecimale (= modalità binaria), 1 esadecimale (= modalità BCD) o 2 esadecimale (= modalità 360°). ON se la compensazione del resto encoder specificata supera la risoluzione definita dall'utente quando i bit 00 - 03 di C sono 0 esadecimale (= risoluzione specificata dall'utente). ON se il valore binario convertito è inferiore alla compensazione del resto encoder quando i bit 00 - 03 di C sono 0 esadecimale (= risoluzione specificata dall'utente). ON se il valore binario convertito è inferiore alla risoluzione quando i bit 00 - 03 di C sono 0 esadecimale (= risoluzione specificata dall'utente). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF in tutti i casi. |
| Flag negativo | N | OFF in tutti i casi. |

Esempi

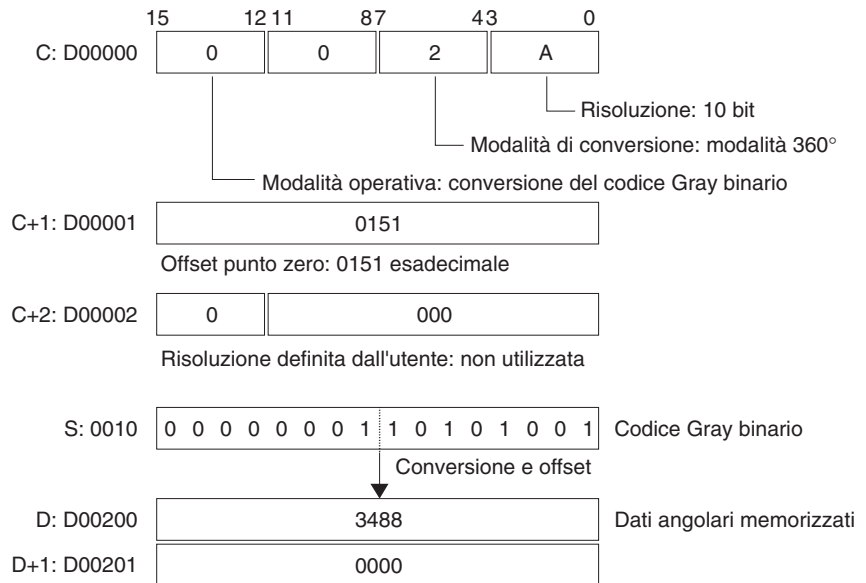
Nell'esempio seguente, quando CIO 000000 è ON, il codice Gray binario in CIO 0010 verrà convertito in base alle impostazioni dei dati di controllo in D00000 e D00002 e il risultato verrà inviato a D00200 e a D00201.



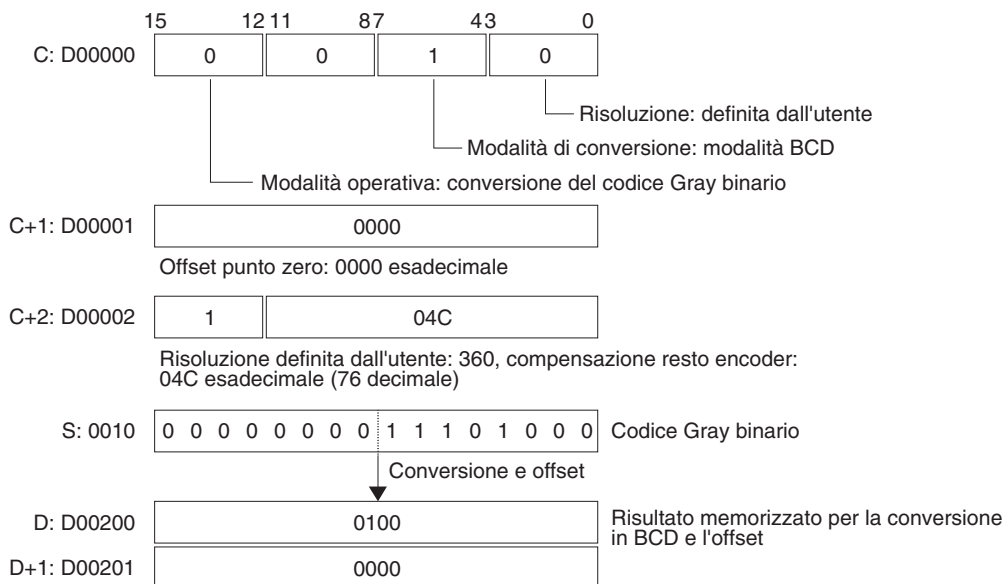
■ Esempio 1: conversione in dati binari con una risoluzione di 8 bit e un offset del punto zero di 001A esadecimale



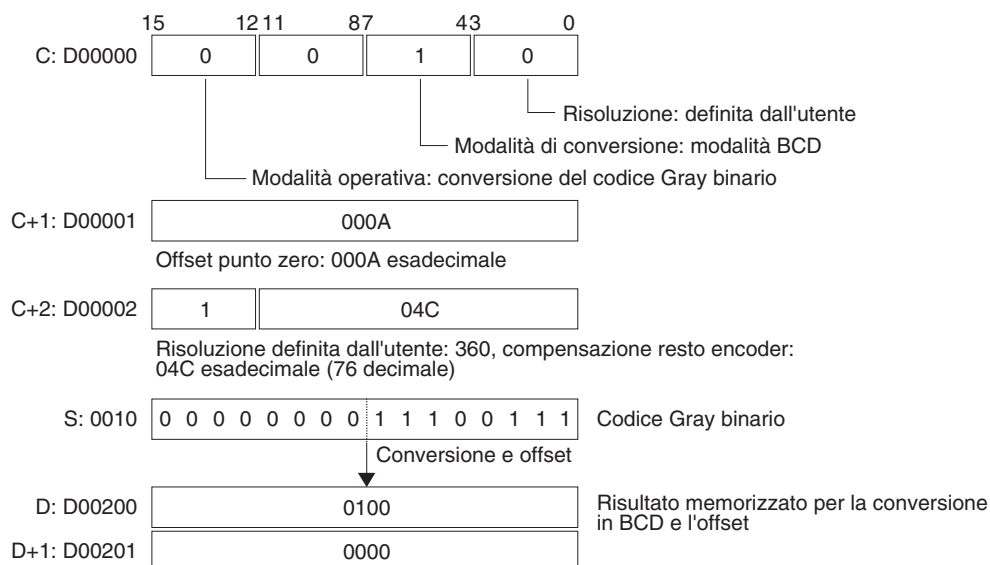
■ Esempio 2: conversione in dati angolari con una risoluzione di 10 bit e un offset del punto zero di 0151 esadecimale



■ **Esempio 3: conversione in dati BCD per un encoder assoluto OMRON E6C2-AG5C (risoluzione: 360/rotazione, compensazione resto encoder: 76) e offset del punto zero di 0000 esadecimale**



■ **Esempio 4: conversione in dati BCD per un encoder assoluto OMRON E6C2-AG5C (risoluzione: 360/rotazione, compensazione resto encoder: 76) e offset del punto zero di 000A esadecimale**



3-13 Istruzioni logiche

In questa sezione vengono illustrate le istruzioni che eseguono operazioni logiche su dati canale.

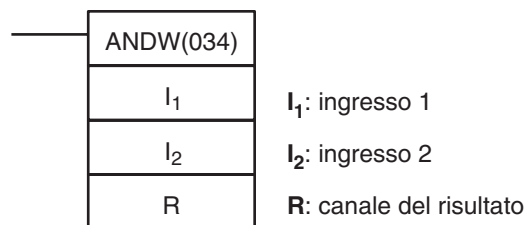
| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|----------------------|------------------|-----------------|--------|
| LOGICAL AND | ANDW | 034 | 517 |
| DOUBLE LOGICAL AND | ANDL | 610 | 519 |
| LOGICAL OR | ORW | 035 | 520 |
| DOUBLE LOGICAL OR | ORWL | 611 | 522 |
| EXCLUSIVE OR | XORW | 036 | 524 |
| DOUBLE EXCLUSIVE OR | XORL | 612 | 526 |
| EXCLUSIVE NOR | XNRW | 037 | 528 |
| DOUBLE EXCLUSIVE NOR | XNRL | 613 | 529 |
| COMPLEMENT | COM | 029 | 531 |
| DOUBLE COMPLEMENT | COML | 614 | 533 |

3-13-1 LOGICAL AND: ANDW(034)

Scopo

Assume un AND logico tra i bit corrispondenti nei canali singoli di dati canale e/o costanti.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ANDW(034) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ANDW(034) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|-------------------------|--|----------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |

| Area | I ₁ | I ₂ | R |
|--|--|----------------|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

ANDW(034) assume un AND logico tra i dati specificati in I₁ e I₂ e invia il risultato a R.

- L'AND logico viene assunto tra i bit corrispondenti in I₁ e I₂ in sequenza.
- Se il contenuto dei bit corrispondenti sia in I₁ che in I₂ è 1 oppure se uno dei due è 0, il risultato inviato al bit corrispondente in R sarà 0:

I₁, I₂ → R

| I ₁ | I ₂ | R |
|----------------|----------------|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando ANDW(034) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'AND il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

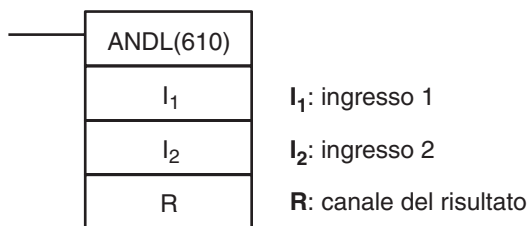
Se come risultato dell'AND il bit più a sinistra di R è 1, verrà attivato il flag negativo.

3-13-2 DOUBLE LOGICAL AND: ANDL(610)

Scopo

Assume un AND logico tra i bit corrispondenti nei canali doppi di dati canale e/o costanti.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ANDL(610) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ANDL(610) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|--|--|----------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritenività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

ANDL(610) assume un AND logico tra i dati specificati in I₁, I₁+1 e I₂, I₂+1 invia il risultato a R, R+1.

$$(I_1, I_1+1), (I_2, I_2+1) \rightarrow (R, R+1)$$

| I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 |
|------------------------------------|------------------------------------|--------|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

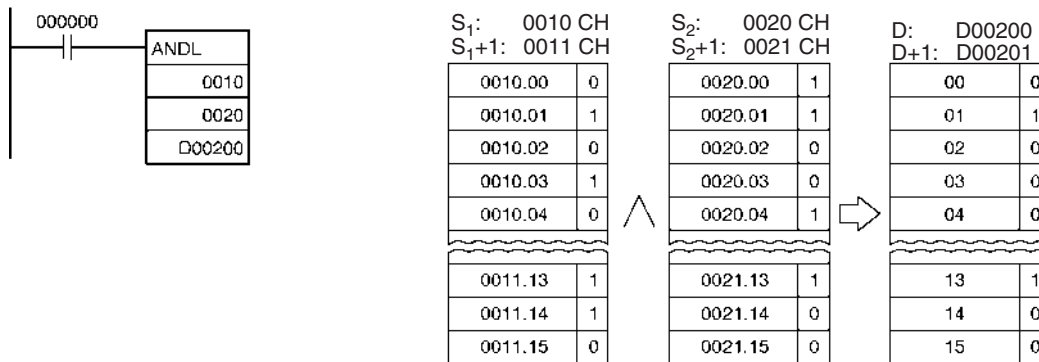
Quando ANDL(610) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'AND il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato dell'AND il bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Quando il CIO 00000000 della condizione di esecuzione è impostato su ON, l'AND logico viene assunto tra i bit corrispondenti in CIO 0011 e CIO 0010 nonché in CIO 0021 e CIO 0020 e i risultati vengono inviati ai bit corrispondenti in D00201 e D00200.



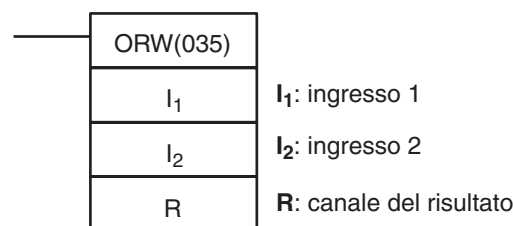
Nota: la freccia verticale indica l'AND logico.

3-13-3 LOGICAL OR: ORW(035)

Scopo

Assume un OR logico tra i bit corrispondenti nei canali singoli di dati canale e/ o costanti.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ORW(035) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ORW(035) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|--|--|----------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a+2047, da ,IR0 a -2048 a+2047 ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

ORW(035) assume un OR logico tra i dati specificati in I₁ e I₂ e invia il risultato a R.

- L'OR logico viene assunto tra i bit corrispondenti in I₁ e I₂ in sequenza.
- Se uno dei bit corrispondenti sia in I₁ che in I₂ è 1 oppure se entrambi sono 0, il risultato inviato al bit corrispondente in R sarà 0.

I₁ + I₂ → R

| I ₁ | I ₂ | R |
|----------------|----------------|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |

| I ₁ | I ₂ | R |
|----------------|----------------|---|
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando ORW(035) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'OR il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

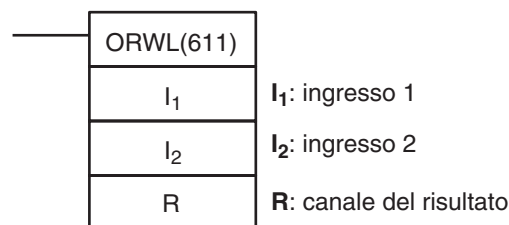
Se come risultato dell'OR il bit più a sinistra di R è 1, verrà attivato il flag negativo.

3-13-4 DOUBLE LOGICAL OR: ORWL(611)

Scopo

Assume un OR logico tra i bit corrispondenti nei canali doppi di dati canale e/ o costanti.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ORWL(611) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ORWL(611) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|-------------------------|------------------------|----------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |

| Area | I ₁ | I ₂ | R |
|--|--|----------------|-----|
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

ORWL(611) assume un OR logico tra i dati specificati in I₁ e I₂ come dati a canale doppio e invia il risultato a R, R+1.

- Se uno dei bit corrispondenti sia in I₁, I₁+1, I₂ sia in I₂+1 è 1, il risultato inviato al bit corrispondente in R+1 sarà 1. Se uno di tali bit è 0, il risultato inviato al bit corrispondente in R+1 sarà 0.

(I₁, I₁+1), (I₂, I₂+1) → (R, R+1)

| I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 |
|------------------------------------|------------------------------------|--------|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

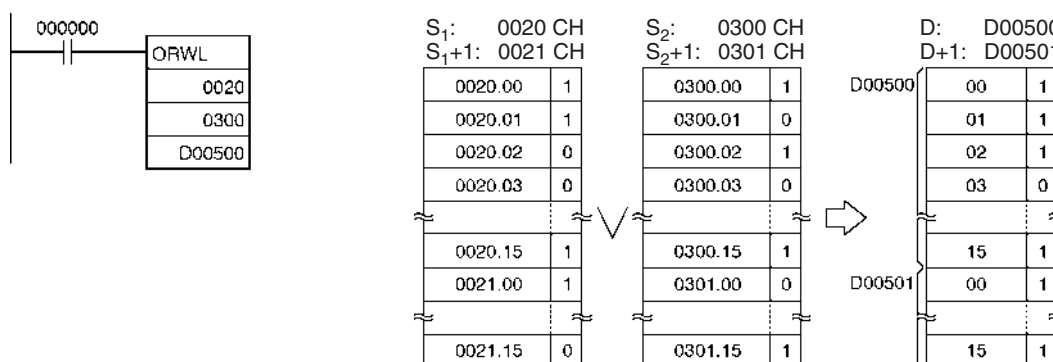
Quando ORWL(611) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'OR il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato dell'OR il bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Quando il CIO 0000000 della condizione di esecuzione è impostato su ON, l'OR logico viene assunto tra i bit corrispondenti in CIO 0021 e CIO 0020 nonché in CIO 0301 e CIO 0300 e i risultati vengono inviati ai bit corrispondenti in D00501 e D00500.



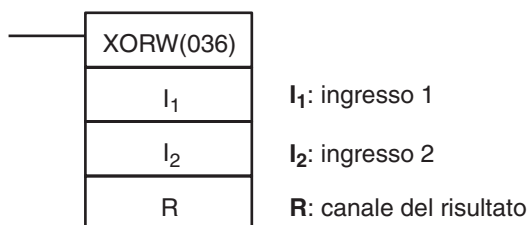
Nota: la freccia verticale indica l'OR logico.

3-13-5 EXCLUSIVE OR: XORW(036)

Scopo

Assume un OR logico esclusivo tra i bit corrispondenti nei canali singoli di dati canale e/o costanti.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | XORW(036) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XORW(036) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|-------------------------|--|----------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |

| Area | I ₁ | I ₂ | R |
|--|--|----------------|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

XORW(036) assume un OR logico esclusivo tra i dati specificati in I₁ e I₂ e invia il risultato a R.

- L'OR logico esclusivo viene assunto tra i bit corrispondenti in I₁ e I₂ in sequenza.
- Quando il contenuto dei bit corrispondenti in I₁ e I₂ è differente, il risultato inviato al bit corrispondente in R sarà 1; quando tale contenuto è uguale, il risultato inviato al bit corrispondente in R sarà 0.

$I_1, \overline{I_1}, I_2 \rightarrow R$

| I ₁ | I ₂ | R |
|----------------|----------------|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando XORW(036) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'OR il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

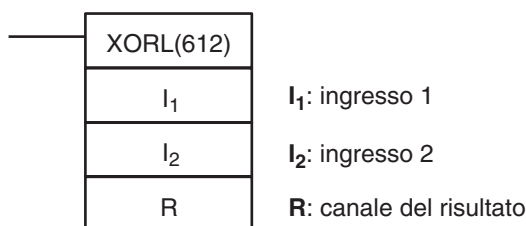
Se come risultato dell'OR il bit più a sinistra di R è 1, verrà attivato il flag negativo.

3-13-6 DOUBLE EXCLUSIVE OR: XORL(612)

Scopo

Assume un OR logico esclusivo tra i bit corrispondenti nei canali doppi di dati canale e/o costanti.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | XORL(612) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XORL(612) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|--|--|----------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

XORL(612) assume un OR logico esclusivo tra i dati specificati in I₁ e I₂ come dati a canale doppio e invia il risultato a R, R+1.

- Se il contenuto di uno dei bit corrispondenti in I₁, I₁+1, I₂ e I₂+1 è diverso, il risultato inviato al bit corrispondente in R, R+1 sarà 1. Se uno di tali contenuti è uguale, il risultato inviato al bit corrispondente in R, R+1 sarà 0.

$$(I_1, I_1+1), (\overline{I_2}, I_2+1) + (I_1, I_1+1), (I_2, I_2+1) \rightarrow (R, R+1)$$

| I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 |
|------------------------------------|------------------------------------|--------|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

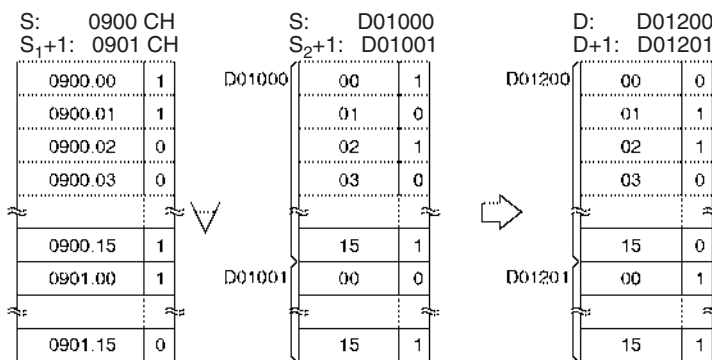
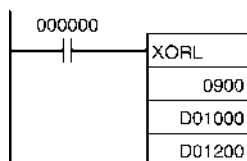
Quando XORL(612) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'OR esclusivo il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato dell'OR esclusivo il bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Quando il CIO 00000000 della condizione di esecuzione è impostato su ON, l'OR viene assunto tra i bit corrispondenti in CIO 0901 e CIO 0900 nonché in D01001 e D01000 e i risultati vengono inviati ai bit corrispondenti in D01201 e D01200.

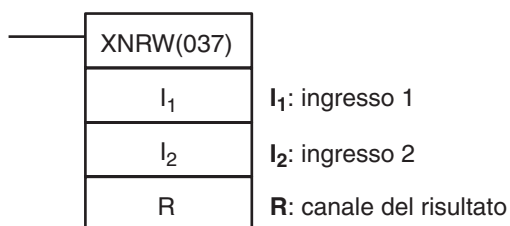


Nota: il simbolo indica l'OR logico esclusivo.

3-13-7 EXCLUSIVE NOR: XNRW(037)

Scopo Assume un NOR logico esclusivo tra i canali singoli corrispondenti di dati canale e/o costanti.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | XNRW(037) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XNRW(037) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|--|--|----------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

XNRW(037) assume un NOR logico esclusivo tra i dati specificati in I_1 e I_2 e invia il risultato a R.

- Il NOR logico esclusivo viene assunto tra i bit corrispondenti in I_1 e I_2 in sequenza.
- Quando il contenuto dei bit corrispondenti in I_1 e I_2 è differente, il risultato inviato al bit corrispondente in R sarà 0; quando tale contenuto è uguale, il risultato inviato al bit corrispondente in R sarà 1.

$$I_1, I_2 + \overline{I_1}, \overline{I_2} \rightarrow R$$

| I_1 | I_2 | R |
|-------|-------|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

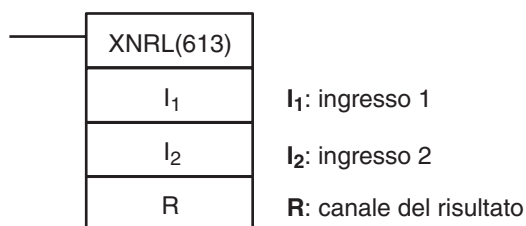
Quando XNRW(037) viene eseguita, il flag di errore viene disattivato.

Se come risultato del NOR il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato del NOR il bit più a sinistra di R è 1, verrà attivato il flag negativo.

3-13-8 DOUBLE EXCLUSIVE NOR: XNRL(613)**Scopo**

Assume un NOR logico esclusivo tra i bit corrispondenti nei canali doppi di dati canale e/o costanti.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | XNRL(613) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XNRL(613) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | I ₁ | I ₂ | R |
|--|--|----------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W 510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

XNRL(613) assume un NOR logico esclusivo tra i dati specificati in I₁ e I₂ e invia il risultato a R, R+1.

- Se il contenuto di uno dei bit corrispondenti in I₁, I₁+1, I₂ e I₂+1 è diverso, il risultato inviato al bit corrispondente in R, R+1 sarà 0. Se uno di tali contenuti è uguale, il risultato inviato al bit corrispondente in R, R+1 sarà 1.

$$(I_1, I_1+1), (I_2, I_2+1) + \overline{(I_1, I_1+1)}, \overline{(I_2, I_2+1)} \rightarrow (R, R+1)$$

| I ₁ , I ₁ +1 | I ₂ , I ₂ +1 | R, R+1 |
|------------------------------------|------------------------------------|--------|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Flag

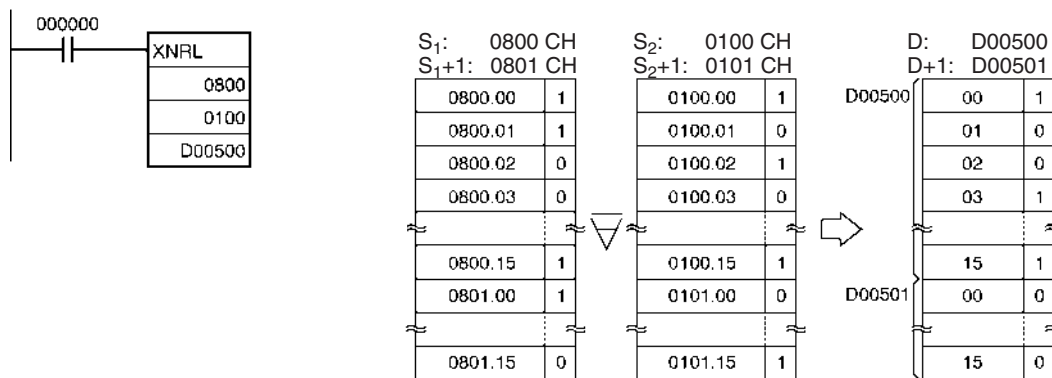
| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando XNRL(613) viene eseguita, il flag di errore viene disattivato.
 Se come risultato del NOR esclusivo il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.
 Se come risultato dell’NOR esclusivo il bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Quando il CIO 00000000 della condizione di esecuzione è impostato su ON, il NOR logico esclusivo viene assunto tra i bit corrispondenti in CIO 0801 e CIO 0800 nonché in CIO 0101 e CIO 0100 e i risultati vengono inviati ai bit corrispondenti in D00501 e D00500.



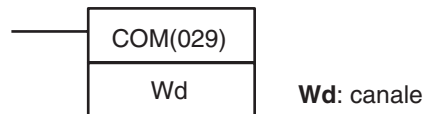
Nota: il simbolo indica il NOR logico esclusivo.

3-13-9 COMPLEMENT: COM(029)

Scopo

Imposta su OFF tutti i bit il cui stato è ON e su ON tutti i bit il cui stato è OFF nel canale Wd.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | COM(029) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @COM(029) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| | |
|-------------------------|------------------------|
| Area | Wd |
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |

| Area | Wd |
|--|--|
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 |

Descrizione

COM(029) inverte lo stato di ciascun bit specificato in Wd.
 $\overline{Wd} \rightarrow Wd$: 1 \rightarrow 0 e 0 \rightarrow 1

Nota Quando si utilizza l'istruzione COM, è necessario ricordare che lo stato di ciascun bit viene modificato in tutti i cicli nei quali la condizione di esecuzione è impostata su ON.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

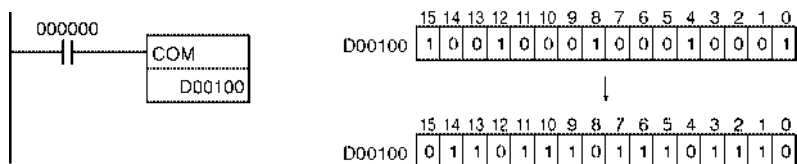
Quando COM(029) viene eseguita, il flag di errore viene disattivato.

Se come risultato dell'istruzione COM il contenuto di R è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se come risultato dell'istruzione COM il bit più a sinistra di R è 1, verrà attivato il flag negativo.

Esempi

Nell'esempio seguente quando CIO 000000 è impostato su ON, lo stato di ogni bit in D00100 viene invertito.

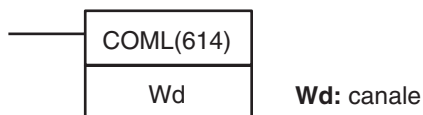


3-13-10 DOUBLE COMPLEMENT: COML(614)

Scopo

Imposta su OFF tutti i bit il cui stato è ON e su ON tutti i bit il cui stato è OFF nei canali Wd e Wd+1.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| | Eseguita a ogni ciclo con condizione ON | COML(614) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @COML(614) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Wd |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H510 |
| Area bit ausiliaria | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

COML(614) inverte lo stato di ciascun bit specificato in Wd e Wd+1.
 $(\overline{Wd+1}, \overline{Wd}) \rightarrow (Wd+1, Wd)$

Nota Quando si utilizza l'istruzione COM, è necessario ricordare che lo stato di ciascun bit viene modificato in tutti i cicli nei quali la condizione di esecuzione è impostata su ON.

Flag

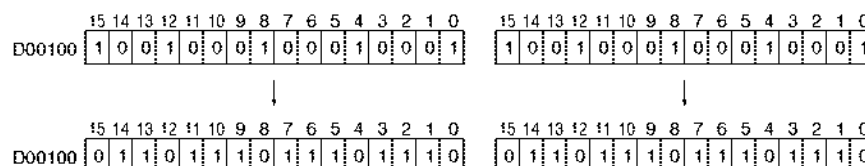
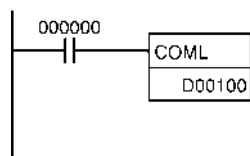
| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON quando il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra di R è 1. OFF in tutti gli altri casi. |

Avvertenze

Quando COML(614) viene eseguita, il flag di errore viene disattivato.
 Se come risultato di COML il contenuto di R, R+1 è 00000000 esadecimale, verrà attivato il flag di uguaglianza.
 Se come risultato di COML il bit più a sinistra di R+1 è 1, verrà attivato il flag negativo.

Esempi

Nell'esempio seguente quando CIO 000000 è impostato su ON, lo stato di ogni bit in D00100 e D00101 viene invertito.



3-14 Istruzioni matematiche speciali

In questa sezione vengono descritte le istruzioni utilizzate per calcoli matematici speciali.

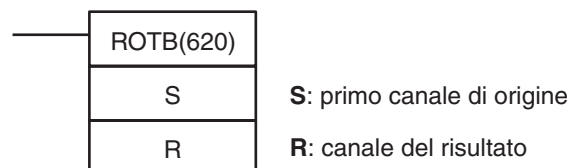
| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|-----------------------|------------------|-----------------|--------|
| BINARY ROOT | ROTB | 620 | 534 |
| BCD SQUARE ROOT | ROOT | 072 | 536 |
| ARITHMETIC PROCESS | APR | 069 | 540 |
| FLOATING POINT DIVIDE | FDIV | 079 | 552 |
| BIT COUNTER | BCNT | 067 | 556 |

3-14-1 BINARY ROOT: ROTB(620)

Scopo

Calcola la radice quadrata del contenuto binario con segno a 32 bit (valore positivo) dei canali specificati e invia la parte intera del risultato al canale del risultato specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ROTB(620) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ROTB(620) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

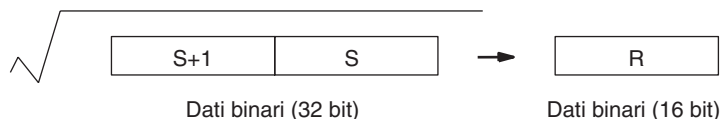
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

ROTB(620) calcola la radice quadrata del numero binario a 32 bit in S+1 ed S e invia la parte intera del risultato in R. Il resto non intero viene eliminato.



Per i canali S+1 ed S è possibile specificare l'intervallo di dati compreso tra 0000 0000 e 3FFF FFFF. Se viene indicato un numero compreso tra 4000 0000 e 7FFF FFFF, nel calcolo della radice quadrata tale numero viene considerato come 3FFF FFFF. Se il contenuto dei canali di origine è maggiore di 7FFF FFFF, cioè se il bit 15 di S+1 è 1, verrà generato un errore.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se il bit 15 di S+1 è 1 (ON). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il contenuto di S+1 ed S è compreso tra 4000 0000 e 7FFF FFFF. OFF in tutti gli altri casi. |
| Flag di underflow | UF | OFF |
| Flag negativo | N | OFF |

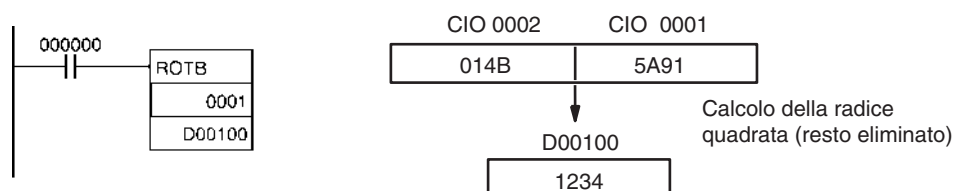
Avvertenze

Il contenuto di S+1 ed S deve essere minore di 8000 0000.

Tutti gli operandi di questa istruzione (S+1, S ed R) vengono considerati come valori binari. Se i dati di ingresso sono in formato BCD, utilizzare l'istruzione ROOT(072).

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, ROTB(620) calcola la radice quadrata dei dati in CIO 0002 e in CIO 0001 e scrive la parte intera del risultato in D00100.

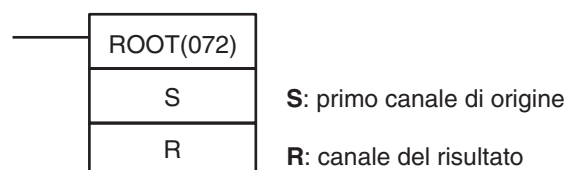


3-14-2 BCD SQUARE ROOT: ROOT(072)

Scopo

Calcola la radice quadrata di un numero in formato BCD a 8 cifre e invia la parte intera del risultato al canale del risultato specificato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ROOT(072) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ROOT(072) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

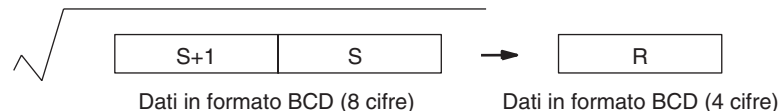
Caratteristiche operando

| Area | S | R |
|----------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 |

| Area | S | R |
|--|---|--|
| Area bit di ritenività | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #99999999 (BCD) | --- |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

ROOT(072) calcola la radice quadrata del numero BCD a 8 cifre in S+1 ed S e invia la parte intera del risultato in R. Il resto non intero viene eliminato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati in S+1 ed S non sono in formato BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |

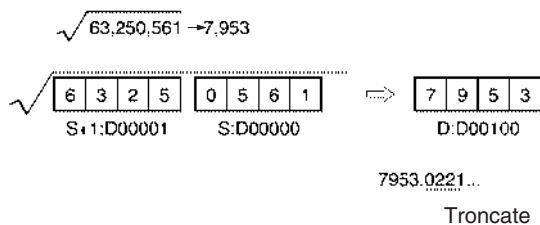
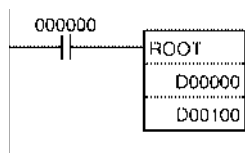
Avvertenze

Tutti gli operandi di questa istruzione (S+1, S ed R) vengono considerati come valori BCD. Se i dati di ingresso sono in formato binario, utilizzare l'istruzione ROTB(620).

Esempi**Radice quadrata di un numero a 8 cifre**

Nell'esempio seguente, quando CIO 000000 è impostato su ON, ROOT(072) calcola la radice quadrata dei dati in D00001 e D00000 e scrive la parte intera del risultato in D00100.

Nota Nei numeri a 8 cifre vengono troncate le cifre dopo la virgola decimale.



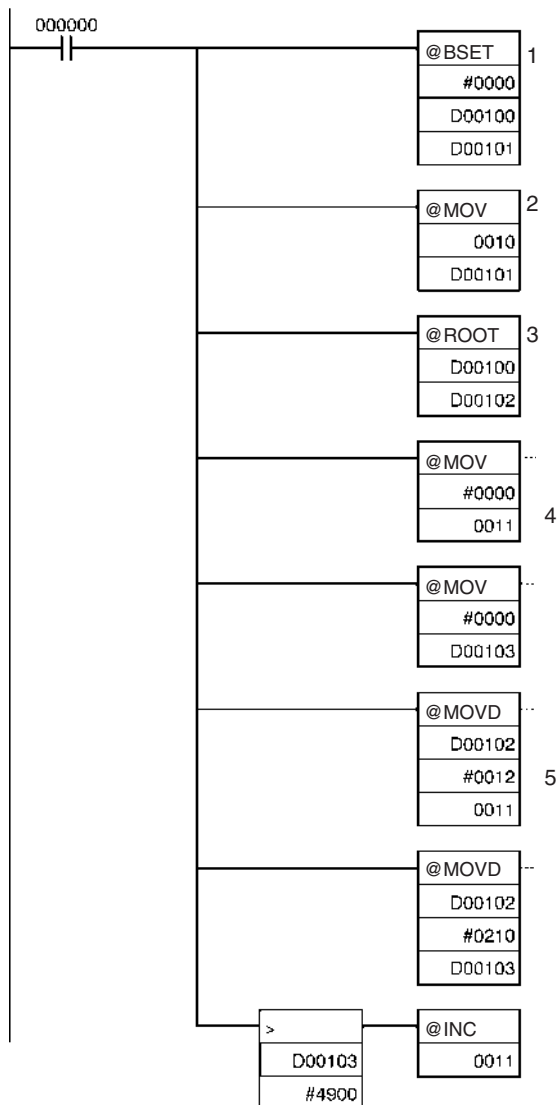
Radice quadrata di un numero a 4 cifre

Nell'esempio che segue viene illustrato come estrarre la radice quadrata di un numero a 4 cifre e arrotondare il risultato. Nell'esempio di programma viene eseguito il calcolo della radice quadrata del numero a 4 cifre in CIO 0010 e il risultato viene arrotondato e scritto in CIO 0011. In sostanza, il numero a 4 cifre viene moltiplicato per 10.000 (100²) e il risultato viene diviso per 100, ottenendo un aumento della precisione di calcolo di un fattore pari a 100.

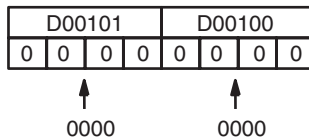
Nota Nei numeri a 4 cifre vengono arrotondate le cifre dopo la virgola decimale.

$$\sqrt{6017} = 77.56... \rightarrow 78$$

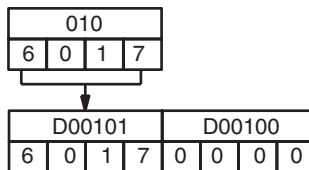
È opportuno arrotondare i valori dopo la virgola decimale.



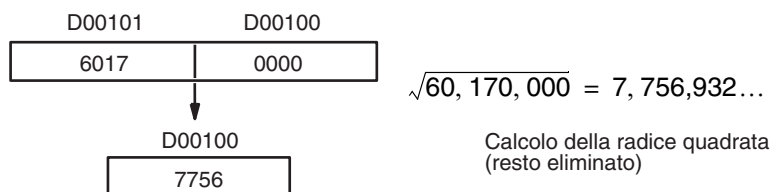
1,2,3... 1. I canali di origine D00101 e D00100 vengono azzerati (0000 0000).



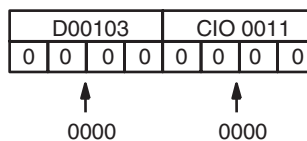
2. Il numero a 4 cifre viene spostato in D00101.



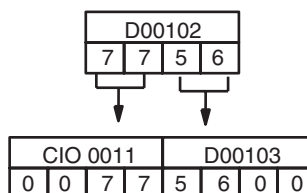
3. ROOT(072) calcola la radice quadrata di D00101 e D00100 e scrive il risultato in D00102.



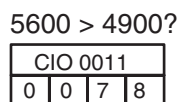
4. D00103 e il canale di origine CIO 0011 vengono azzerati (0000 0000).



5. Il risultato del calcolo della radice quadrata è diviso per 100: la parte intera viene scritta in CIO 0011 e il resto viene inviato in D00103.



6. Se il contenuto di D00103 è maggiore di 4900, CIO 0011 viene incrementato di 1. In questo caso il risultato sarà 78.

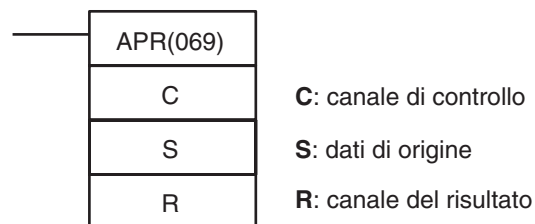


3-14-3 ARITHMETIC PROCESS: APR(069)

Scopo

Calcola seno, coseno o un'estrapolazione lineare dei dati di origine. La funzione di estrapolazione lineare consente di approssimare relazioni tra X e Y con segmenti di linea.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | APR(069) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @APR(069) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

Funzione di seno (C = 0000 esadecimale)

| Operando | Valore | Intervallo dati |
|----------|----------------------|-----------------|
| C | 0000 esadecimale | --- |
| S | Da 0000 a 0900 (BCD) | 0° ... 90° |
| D | Da 0000 a 9999 (BCD) | 0,0000...0,9999 |
| | 9999 (BCD) | 1,0000 |

Funzione di coseno (C = 0001 esadecimale)

| Operando | Valore | Intervallo dati |
|----------|----------------------|-----------------|
| C | 0001 esadecimale | --- |
| S | Da 0000 a 0900 (BCD) | 0° ... 90° |
| D | Da 0000 a 9999 (BCD) | 0,0000...0,9999 |
| | 9999 (BCD) | 1,0000 |

Funzione di estrapolazione lineare (C = indirizzo area dati)

| Operando | Valore | Intervallo dati |
|----------|---|---|
| C | Indirizzo area dati | --- |
| S | Dati BCD senza segno a 16 bit | 0000 ... 9999 |
| | Dati binari senza segno a 16 bit | 0 ... 65.535 |
| | Dati binari con segno a 16 bit ¹ | -32.768 ... 32.767 |
| | Dati binari con segno a 32 bit ¹ | -2.147.483.648 ... 2.147.483.647 |
| | Dati a virgola mobile ¹ | -∞, -3,402823 × 10 ³⁸ ... -1,175494 × 10 ⁻³⁸ , 1,175494 × 10 ⁻³⁸ ... 3,402823 × 10 ³⁸ , +∞ |
| D | Dati BCD senza segno a 16 bit | 0000 ... 9999 |
| | Dati binari senza segno a 16 bit | 0 ... 65.535 |
| | Dati binari con segno a 16 bit ¹ | -32.768 ... 32.767 |
| | Dati binari con segno a 32 bit ¹ | -2.147.483.648 ... 2.147.483.647 |
| | Dati a virgola mobile ¹ | -∞, -3.402823 × 10 ³⁸ ... -1,175494 × 10 ⁻³⁸ , 1,175494 × 10 ⁻³⁸ ... 3,402823 × 10 ³⁸ , +∞ |

- Nota**
1. I dati binari con segno e i dati a virgola mobile sono supportati solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.
 2. Se C è un indirizzo del canale, APR(069) estrapola il valore Y per il valore X di S in base a coordinate che formano segmenti di linea e che sono state specificate precedentemente in una tabella che ha inizio in C. Per ulteriori informazioni consultare la *Descrizione* qui di seguito.

Caratteristiche operando

| Area | C | S | R |
|-------------------------|--|----------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |

| Area | C | S | R |
|--|--|---------------|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | | --- |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Il funzionamento di APR(069) dipende dal canale di controllo C. Se C è 0000 o 0001, APR(069) calcola il seno o il coseno di S con S in unità di decimi di grado.

Se C è un indirizzo del canale, APR(069) estrapola il valore Y per il valore X di S in base a coordinate che formano segmenti di linea e che sono state specificate precedentemente in una tabella che ha inizio in C.

Funzione di seno (C = 0000)

Quando C è 0000, APR(069) calcola il SIN(S) e scrive il risultato in R. L'intervallo per S è compreso da 0000 a 0900 BCD (da 0,0° a 90,0°) e l'intervallo per R è compreso da 0000 a 9999 BCD (da 0,0000 a 0,9999). Il resto del risultato dopo il quarto decimale viene eliminato.

Funzione di coseno (C = 0001)

Quando C è 0001, APR(069) calcola il COS(S) e scrive il risultato in R. L'intervallo per S è compreso da 0000 a 0900 BCD (da 0,0° a 90,0°) e l'intervallo per R è compreso da 0000 a 9999 BCD (da 0,0000 a 0,9999). Il resto del risultato dopo il quarto decimale viene eliminato.

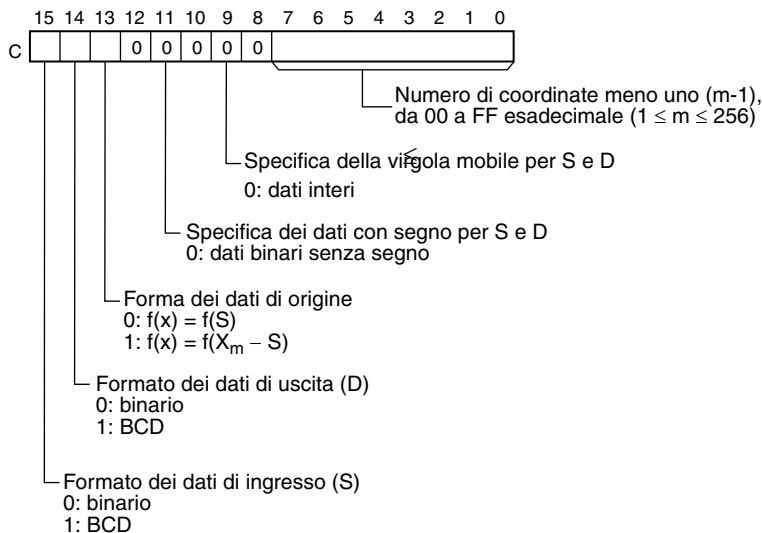
Estrapolazione lineare

L'estrapolazione lineare APR(069) viene specificata quando C è un indirizzo di canale.

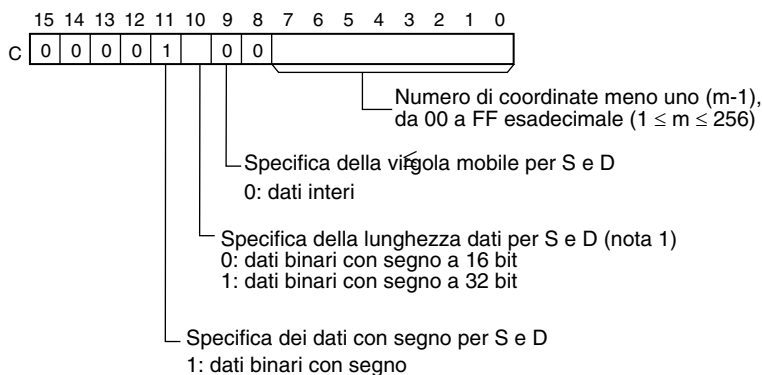
Il contenuto del canale C indica il numero di coordinate in una tabella dati che ha inizio in C+2, la forma dei dati di origine e il formato dei dati (BCD o

binario). Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D i dati di origine possono essere anche dati binari con segno o dati a virgola mobile.

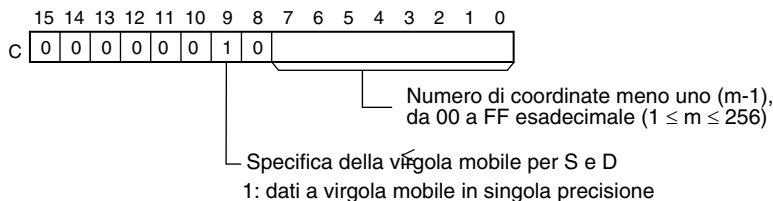
Dati interi senza segno (binari o BCD)



Dati interi con segno (binari)



Dati a virgola mobile in singola precisione



Se si utilizzano dati binari a 16 bit o dati BCD, i dati lineari a segmenti sono contenuti nei canali compresi tra C+ 1 e C+2m+2. Se si utilizzano dati binari a 32 bit o dati a virgola mobile (solo per CPU CS1-H, CJ1-H e CJ1M), i dati lineari a segmenti sono contenuti nei canali compresi tra C+ 1 e C+4m+4.

I bit compresi tra 00 e 07 contengono il numero(binario) di coordinate di linea meno 1, m-1. I bit compresi tra 08 e 12 non vengono utilizzati. Il bit 13 specifica $f(x)=f(S)$ o $f(x)=f(X_m-S)$: OFF specifica $f(x)=f(S)$ e ON specifica $f(x)=f(X_m-S)$. Il bit 14 determina il formato dell'uscita (BCD o binario): OFF

indica binario e ON indica BCD. Il bit 15 determina il formato dell'ingresso (BCD o binario): OFF indica binario e ON indica BCD.

Dati binari a 16 bit (con o senza segno) oppure dati BCD a 16 bit

Dati binari con segno a 32 bit

Dati a virgola mobile

| | |
|----------|---------|
| C+1 | X0 (*1) |
| C+2 | Y0 |
| C+3 | X1 |
| C+4 | Y1 |
| C+5 | X2 |
| C+6 | Y2 |
| | |
| | Xn |
| | Yn |
| | |
| C+(2m+1) | Xm |
| C+(2m+2) | Ym |

Nota: Scrivere X_m (valore massimo di X nella tabella) nel canale C+1 quando i dati di I/O in S e D contengono dati con segno (bit 11 di C = 0).

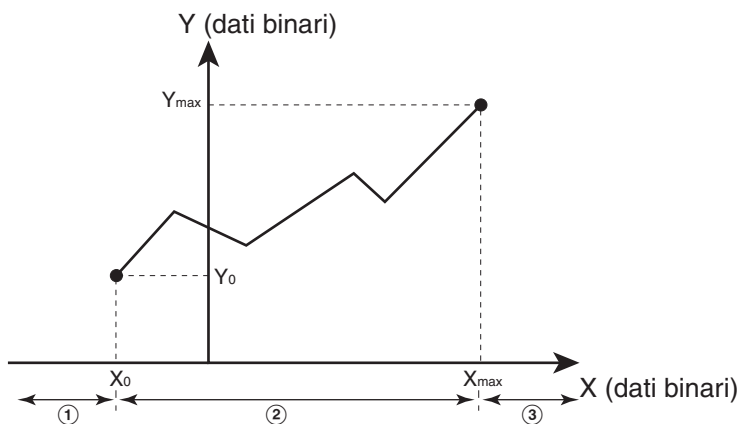
| | |
|----------|----------------------------|
| C+1 | X0 (16 bit più a destra) |
| C+2 | X0 (16 bit più a sinistra) |
| C+3 | Y0 (16 bit più a destra) |
| C+4 | Y0 (16 bit più a sinistra) |
| C+5 | X1 (16 bit più a destra) |
| C+6 | X1 (16 bit più a sinistra) |
| C+7 | Y1 (16 bit più a destra) |
| C+8 | Y1 (16 bit più a sinistra) |
| ... | ... |
| C+(4n+1) | Xn (16 bit più a destra) |
| C+(4n+2) | Xn (16 bit più a sinistra) |
| C+(4n+3) | Yn (16 bit più a destra) |
| C+(4n+4) | Yn (16 bit più a sinistra) |
| ... | ... |
| C+(4m+1) | Xm (16 bit più a destra) |
| C+(4m+2) | Xm (16 bit più a sinistra) |
| C+(4m+3) | Ym (16 bit più a destra) |
| C+(4m+4) | Ym (16 bit più a sinistra) |

| | |
|----------|----------------------------|
| C+1 | X0 (16 bit più a destra) |
| C+2 | X0 (16 bit più a sinistra) |
| C+3 | Y0 (16 bit più a destra) |
| C+4 | Y0 (16 bit più a sinistra) |
| C+5 | X1 (16 bit più a destra) |
| C+6 | X1 (16 bit più a sinistra) |
| C+7 | Y1 (16 bit più a destra) |
| C+8 | Y1 (16 bit più a sinistra) |
| ... | ... |
| C+(4n+1) | Xn (16 bit più a destra) |
| C+(4n+2) | Xn (16 bit più a sinistra) |
| C+(4n+3) | Yn (16 bit più a destra) |
| C+(4n+4) | Yn (16 bit più a sinistra) |
| ... | ... |
| C+(4m+1) | Xm (16 bit più a destra) |
| C+(4m+2) | Xm (16 bit più a sinistra) |
| C+(4m+3) | Ym (16 bit più a destra) |
| C+(4m+4) | Ym (16 bit più a sinistra) |

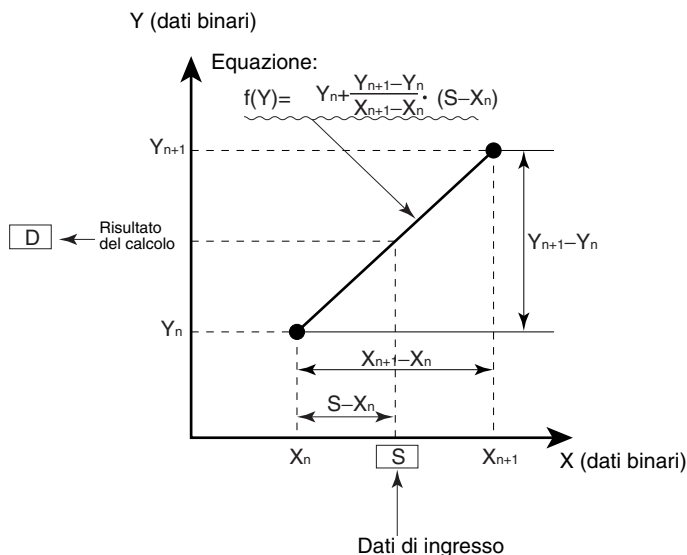
Nota Le coordinate X devono essere in ordine crescente: $X_1 < X_2 < \dots < X_m$. Inserire tutti i valori di (X_n, Y_n) come dati binari, indipendentemente dal formato di dati specificato nel canale di controllo C.

Funzionamento della funzione di estrapolazione lineare

APR(069) elabora i dati di ingresso specificati in S con l'equazione seguente e i dati lineari a segmenti (X_n, Y_n) specificati nella tabella che ha inizio in C+1. Il risultato viene inviato al canale o ai canali di destinazione specificato/i con D.



1. Per $S < X_0$
Valore convertito = Y_0
2. Per $X_0 \leq S \leq X_{max}$, se $X_n < S < X_{n+1}$
Valore convertito = $Y_n + \frac{Y_{n+1} - Y_n}{X_{n+1} - X_n} \times [Dati\ di\ ingresso\ S - X_n]$



- 3. $X_{max} < S$
 Valore convertito = Y_{max}

Nella tabella dei dati lineari a segmenti che ha inizio in C+1 è possibile memorizzare fino a 256 endpoint. È possibile utilizzare i 5 tipi di dati di I/O seguenti:

- dati BCD senza segno a 16 bit
- dati binari senza segno a 16 bit
- dati binari con segno a 16 bit (solo per CPU CS1-H, CJ1-H e CJ1M)
- dati binari con segno a 32 bit (solo per CPU CS1-H, CJ1-H e CJ1M)
- dati a virgola mobile in singola precisione (solo per CPU CS1-H, CJ1-H e CJ1M)

Impostazione del formato dei dati nel canale di controllo C

- Dati BCD senza segno a 16 bit
 I dati di ingresso e/o i dati di uscita possono essere dati BCD senza segno a 16 bit. Inoltre è possibile impostare la funzione di estrapolazione lineare per elaborare il valore specificato direttamente in S oppure $X_m - S$ (X_m è il valore massimo di X nei dati lineari a segmenti).

| Nome dell'impostazione | Bit in C | Impostazione |
|---|----------|--|
| Formato dei dati di ingresso (S) | 15 | 0: binario 1: BCD |
| Formato dei dati di uscita (D) | 14 | 0: binario 1: BCD |
| Forma dei dati di origine | 13 | 0: elaborare S 1: elabora $X_m - S$ |
| Specificazione dei dati con segno per S e D | 11 | 0: dati senza segno |
| Specificazione della lunghezza dati per S e D | 10 | Non valida (fissata a 16 bit) |
| Specificazione per la virgola mobile | 09 | 0: dati interi |

- Dati binari senza segno a 16 bit
I dati di ingresso e/o i dati di uscita possono essere dati binari senza segno a 16 bit. Inoltre è possibile impostare la funzione di estrapolazione lineare per elaborare il valore specificato direttamente in S oppure X_m-S (X_m è il valore massimo di X nei dati lineari a segmenti).

| Nome dell'impostazione | Bit in C | Impostazione |
|--|----------|--|
| Formato dei dati di ingresso (S) | 15 | 0: binario 1: BCD |
| Formato dei dati di uscita (D) | 14 | 0: binario 1: BCD |
| Forma dei dati di origine | 13 | 0: elaborare S 1: elaborare X_m-S |
| Specifica dei dati con segno per S e D | 11 | 0: dati senza segno |
| Specifica della lunghezza dati per S e D | 10 | Non valida (fissata a 16 bit) |
| Specifica per la virgola mobile | 09 | 0: dati interi |

- Dati binari con segno a 16 bit (solo per CPU CS1-H, CJ1-H, CJ1M e CS1D)

| Nome dell'impostazione | Bit in C | Impostazione |
|--|----------|-----------------------------------|
| Formato dei dati di ingresso (S) | 15 | 0: binario |
| Formato dei dati di uscita (D) | 14 | 0: binario |
| Forma dei dati di origine | 13 | 0 |
| Specifica dei dati con segno per S e D | 11 | 1: dati con segno |
| Specifica della lunghezza dati per S e D | 10 | 0: dati binari con segno a 16 bit |
| Specifica per la virgola mobile | 09 | 0: dati interi |

- Dati binari con segno a 32 bit (solo per CPU CS1-H, CJ1-H, CJ1M e CS1D)

| Nome dell'impostazione | Bit in C | Impostazione |
|--|----------|-----------------------------------|
| Formato dei dati di ingresso (S) | 15 | 0: binario |
| Formato dei dati di uscita (D) | 14 | 0: binario |
| Forma dei dati di origine | 13 | 0 |
| Specifica dei dati con segno per S e D | 11 | 1: dati con segno |
| Specifica della lunghezza dati per S e D | 10 | 1: dati binari con segno a 32 bit |
| Specifica per la virgola mobile | 09 | 0: dati interi |

Nota Se la "Specifica della lunghezza dati per S e D" nel bit 10 di C è impostata su 1 e se per S viene immessa una costante a 16 bit, i dati di ingresso saranno convertiti in binari con segno a 32 bit prima del calcolo dell'extrapolazione lineare.

- Dati a virgola mobile (solo per CPU CS1-H, CJ1-H, CJ1M e CS1D)

| Nome dell'impostazione | Bit in C | Impostazione |
|--|----------|--------------------------|
| Formato dei dati di ingresso (S) | 15 | 0: binario |
| Formato dei dati di uscita (D) | 14 | 0: binario |
| Forma dei dati di origine | 13 | 0 |
| Specifica dei dati con segno per S e D | 11 | 0 |
| Specifica della lunghezza dati per S e D | 10 | 0 |
| Specifica per la virgola mobile | 09 | 1: dati a virgola mobile |

Nota Se la "Specifica per la virgola mobile" nel bit 09 di C è impostata su 1, non è possibile immettere una costante per S.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se C è una costante maggiore di 0001. ON se C è un indirizzo del canale ma le coordinate X non sono in ordine crescente ($X_1 \leq X_2 \leq \dots \leq X_m$). ON se C è un indirizzo del canale e i bit 9, 11 e 15 di C indicano un ingresso BCD, ma S non è BCD. ON se C è un indirizzo del canale e il bit 9 di C indica dati a virgola mobile, ma S è una costante a singolo canale. ON se C è 0000 o 0001 ma S non è BCD nell'intervallo tra 0000 e 0900. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di R è ON. OFF in tutti gli altri casi. |

Avvertenze

Il risultato effettivo per SIN(90°) e COS(0°) è 1, ma a R viene inviato 9999 (0,9999).

Se C è una costante maggiore di 0001, verrà generato un errore.

Se è specificata l'extrapolazione lineare ma le coordinate X non sono in ordine crescente ($X_1 < X_2 < \dots < X_m$), verrà generato un errore.

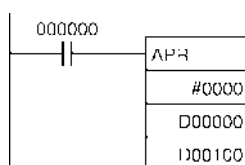
Se sono specificati l'extrapolazione lineare e l'ingresso BCD (bit 15 di C impostato su ON) ma S non è BCD, verrà generato un errore.

Se è specificata una funzione trigonometrica (C = 0000 o 0001) ma S non è BCD nell'intervallo tra 0000 e 0900, verrà generato un errore.

Esempi

Funzione di seno (C: #0000)

L'esempio fornito di seguito illustra l'utilizzo di APR(069) per il calcolo del seno di 30°.



Dati di origine

| S: D00000 | | | |
|-----------|--------|--------|-----------|
| | 10^1 | 10^0 | 10^{-1} |
| 0 | | | |
| 0 | 3 | 0 | 0 |

Impostare i dati di origine su 10^{-1} gradi. (da 0000 a 0900, BCD)

Risultato

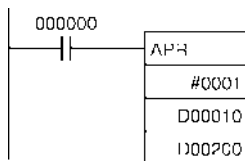
| R: D00100 | | | |
|-----------|-----------|-----------|-----------|
| 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} |
| 5 | 0 | 0 | 0 |

Il risultato presenta quattro cifre significative poiché la quinta cifra e quelle successive vengono ignorate.

Funzione di coseno (C: #0001)

L'esempio fornito di seguito illustra l'utilizzo di APR(069) per il calcolo del coseno di 30°.

(SIN(30) = 0,8660)



Dati di origine

| S: D00010 | | | |
|-----------|--------|--------|-----------|
| | 10^1 | 10^0 | 10^{-1} |
| 0 | | | |
| 0 | 3 | 0 | 0 |

Impostare i dati di origine su 10^{-1} gradi. (da 0000 a 0900, BCD)

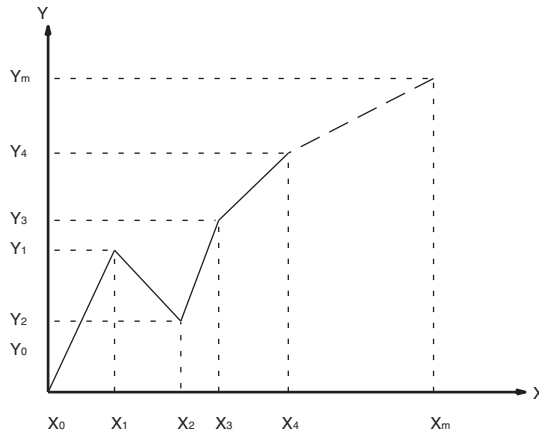
Risultato

| R: D00200 | | | |
|-----------|-----------|-----------|-----------|
| 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} |
| 8 | 6 | 6 | 0 |

Il risultato presenta quattro cifre significative poiché la quinta cifra e quelle successive vengono ignorate.

**Estrapolazione lineare (C: indirizzo del canale)
Utilizzo dei dati binari o BCD senza segno a 16 bit**

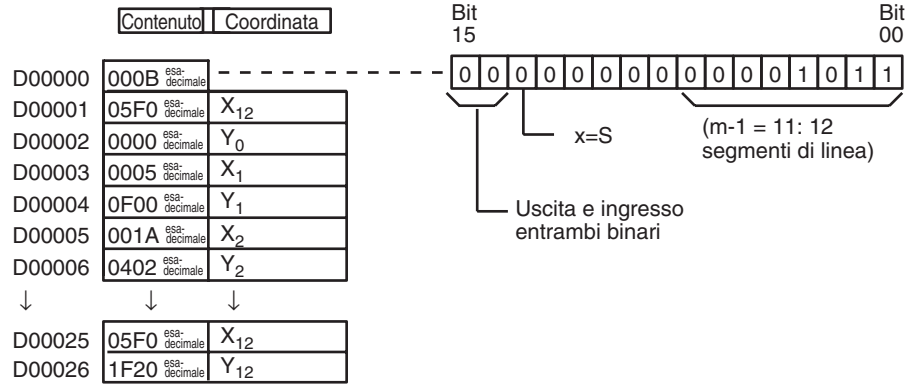
APR(069) elabora i dati di ingresso specificati in S in base ai dati di controllo in C e ai dati lineari a segmenti indicati nella tabella che ha inizio in C+1. Il risultato viene inviato a D.



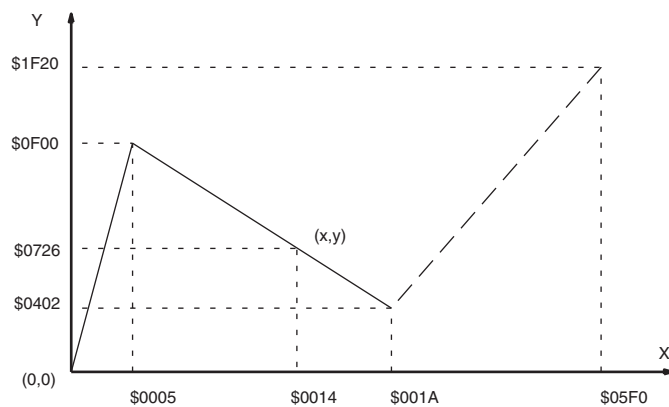
| Canale | Coordinata |
|----------|----------------------------------|
| C+1 | X _m (valore max di X) |
| C+2 | Y ₀ |
| C+3 | X ₁ |
| C+4 | Y ₁ |
| C+5 | X ₂ |
| C+6 | Y ₂ |
| ↓ | ↓ |
| C+(2m+1) | X _m (valore max di X) |
| C+(2m+2) | Y _m |

- $Y_n = f(X_n), Y_0 = f(X_0)$
- Assicurarsi che $X_{n-1} < X_n$ in tutti i casi.
- Indicare tutti i valori di (X_n, Y_n) come dati binari.

Nell'esempio viene illustrato come costruire un'estrapolazione lineare con 12 coordinate. Da D00000 a D00026 (da C a C + (2 × 12 + 2)), il blocco dei dati è correttamente ininterrotto. I dati di ingresso sono assunti da CIO 0010 e il risultato viene inviato a CIO 0011.



In questo caso, il canale di origine CIO 0010 contiene 0014; $f(0014) = 0726$ viene inviato a R, CIO 0011.



Di seguito è riportato il calcolo dell'extrapolazione lineare.

$$Y = 0F00 + \frac{0402 - 0F00}{001A - 0005} \times (0014 - 0005)$$

$$= 0F00 - (0086 \times 000F)$$

$$= 0726$$

Tutti i valori sono esadecimali.

Estrapolazione lineare (C: indirizzo del canale)

Utilizzo dei dati binari con segno a 32 bit (solo per CPU CS1-H, CJ1-H, CJ1M e CS1D)

Nell'esempio, APR(069) viene utilizzata per convertire l'altezza del fluido in un serbatoio nel volume del fluido in base alla forma del serbatoio di contenimento.

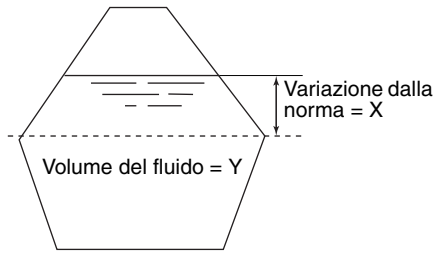
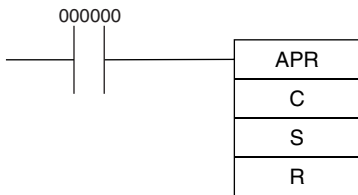
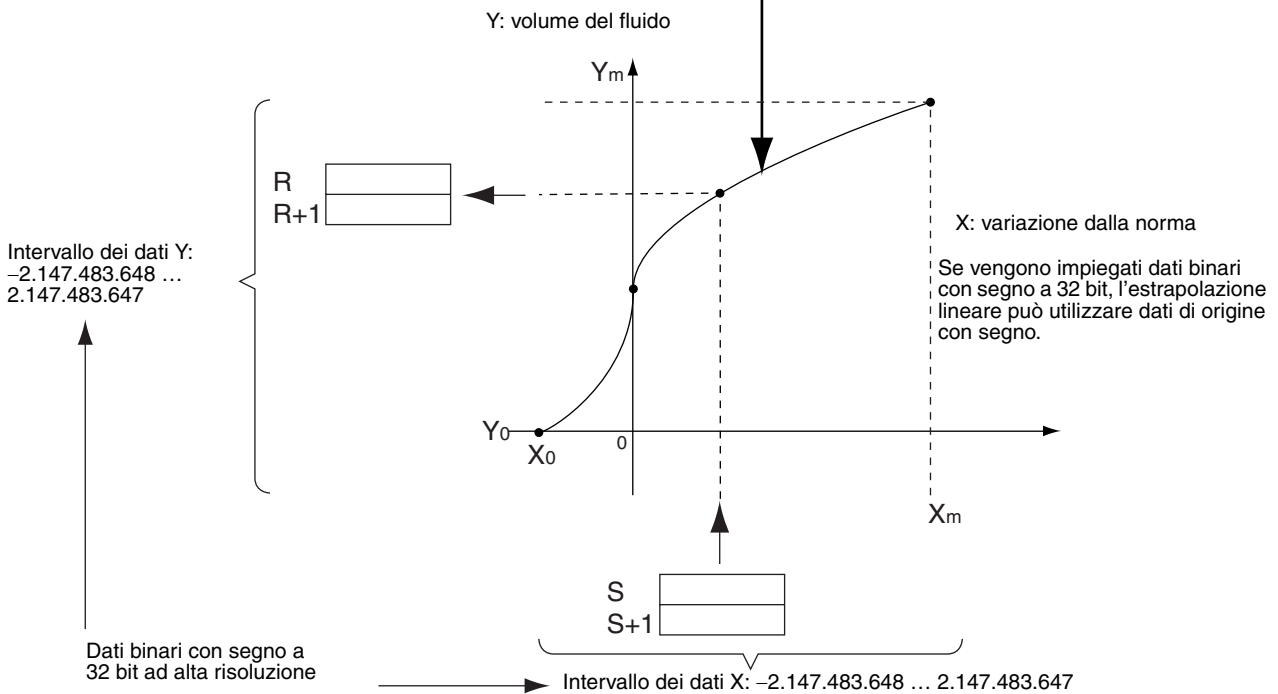


Tabella di conversione altezza / volume di un fluido (dati binari con segno a 32 bit)

| | |
|-----------|----------------------------|
| C+1 | X0 (16 bit più a destra) |
| C+2 | X0 (16 bit più a sinistra) |
| C+3 | Y0 (16 bit più a destra) |
| C+4 | Y0 (16 bit più a sinistra) |
| C+5 | X1 (16 bit più a destra) |
| C+6 | X1 (16 bit più a sinistra) |
| C+7 | Y1 (16 bit più a destra) |
| C+8 | Y1 (16 bit più a sinistra) |
| ... | ... |
| C+ (4n+1) | Xn (16 bit più a destra) |
| C+ (4n+2) | Xn (16 bit più a sinistra) |
| C+ (4n+3) | Yn (16 bit più a destra) |
| C+ (4n+4) | Yn (16 bit più a sinistra) |
| ... | ... |
| C+ (4m+1) | Xm (16 bit più a destra) |
| C+ (4m+2) | Xm (16 bit più a sinistra) |
| C+ (4m+3) | Ym (16 bit più a destra) |
| C+ (4m+4) | Ym (16 bit più a sinistra) |



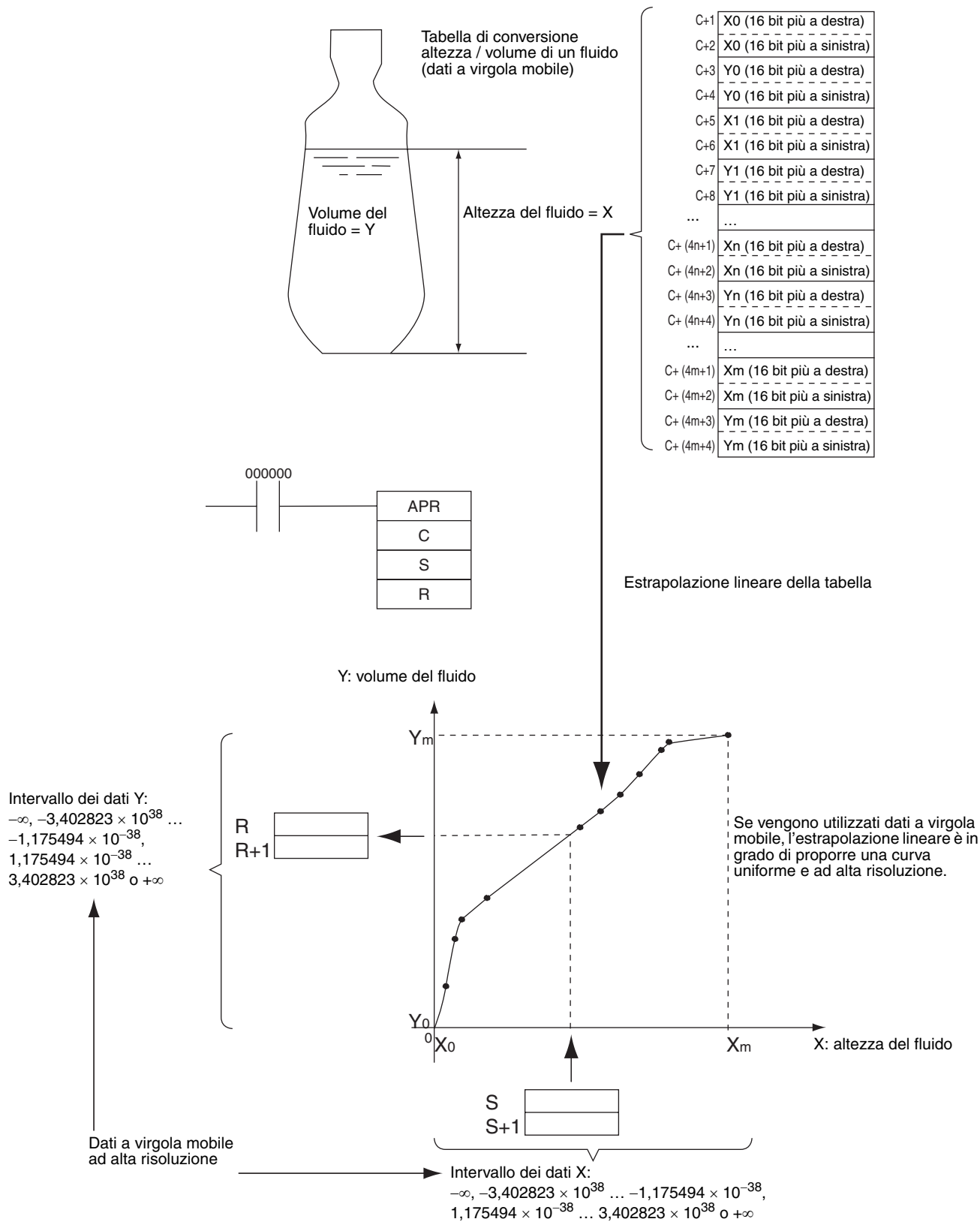
Estrapolazione lineare della tabella



Estrapolazione lineare (C: indirizzo del canale)

Utilizzo dei dati a virgola mobile (solo per CPU CS1-H, CJ1-H, CJ1M e CS1D)

Nell'esempio, APR(069) viene utilizzata per convertire l'altezza del fluido in un serbatoio nel volume del fluido in base alla forma del serbatoio di contenimento.

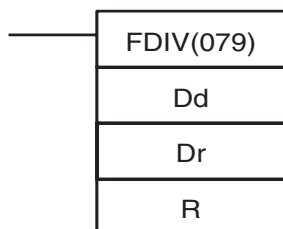


3-14-4 FLOATING POINT DIVIDE: FDIV(079)

Scopo

Divide un numero a virgola mobile a 7 cifre per un altro numero. I numeri a virgola mobile sono espressi in notazione scientifica (7 cifre di mantissa e 1 cifra di esponente).

Simbolo programmazione ladder



Dd: primo canale dividendo

Dr: primo canale divisore

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FDIV(079) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FDIV(079) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

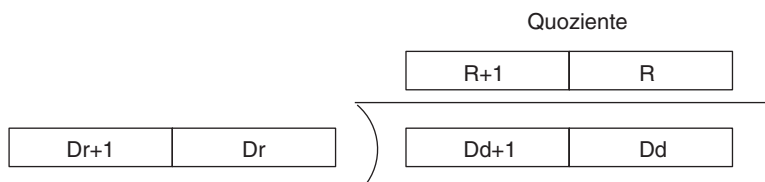
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

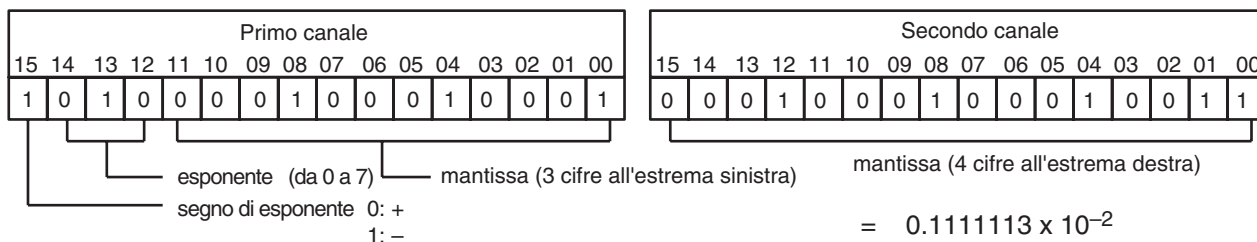
Caratteristiche operando

| Area | Dd | Dr | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione FDIV(079) divide il valore a virgola mobile in Dd e Dd+1 per quello contenuto in Dr e Dr+1 e invia il risultato a R ed R+1.



Per rappresentare i valori a virgola mobile, le sette cifre più a destra vengono utilizzate per la mantissa e quella all'estrema per l'esponente. Osservare il diagramma che segue. La cifra più a sinistra è compresa tra 0 ed F, gli esponenti positivi tra 0 e 7 e quelli negativi tra 8 ed F (da 0 a -7). Le cifre all'estrema destra devono essere in formato BCD.



Qui di seguito vengono forniti due ulteriori esempi di valori a virgola mobile:

6123 4567: $0,1234567 \times 10^6$ (6 = 0110 binario)

B123 4567: $0,1234567 \times 10^{-3}$ (B = 1011 binario)

Nella tabella seguente vengono indicati i valori massimi e minimi consentiti.

| Limite | Esadecimale a 8 cifre | Virgola mobile |
|--------------------------------------|-----------------------|----------------------------|
| Valore massimo | 7999 9999 | $0,9999999 \times 10^7$ |
| Valore minimo (divisore e dividendo) | F000 0001 | $0,0000001 \times 10^{-7}$ |
| Valore minimo (risultato) | F100 0000 | $0,1000000 \times 10^{-7}$ |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se la mantissa (7 cifre all'estrema sinistra) in Dd+1 e Dd non è in formato BCD. ON se la mantissa (7 cifre all'estrema sinistra) in Dr+1 e Dr non è in formato BCD. ON se il divisore (Dr+1 e Dr) è 0. ON se il risultato non è compreso tra $0,1000000 \times 10^{-7}$ e $0,9999999 \times 10^7$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |

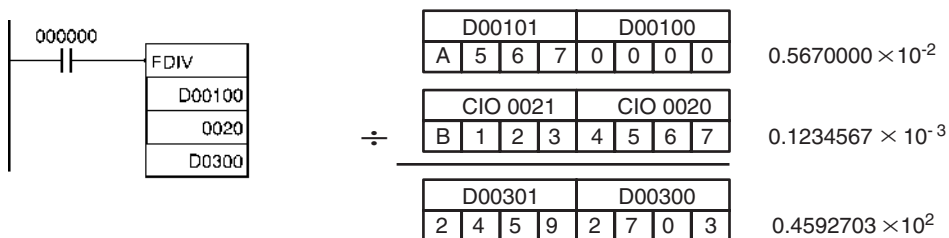
Avvertenze

Il risultato viene espresso come valore a virgola mobile e quindi ha 7 cifre significative. L'ottava cifra e quelle successive vengono eliminate. Il risultato deve essere compreso tra $0,1000000 \times 10^{-7}$ e $0,9999999 \times 10^7$.

Esempi

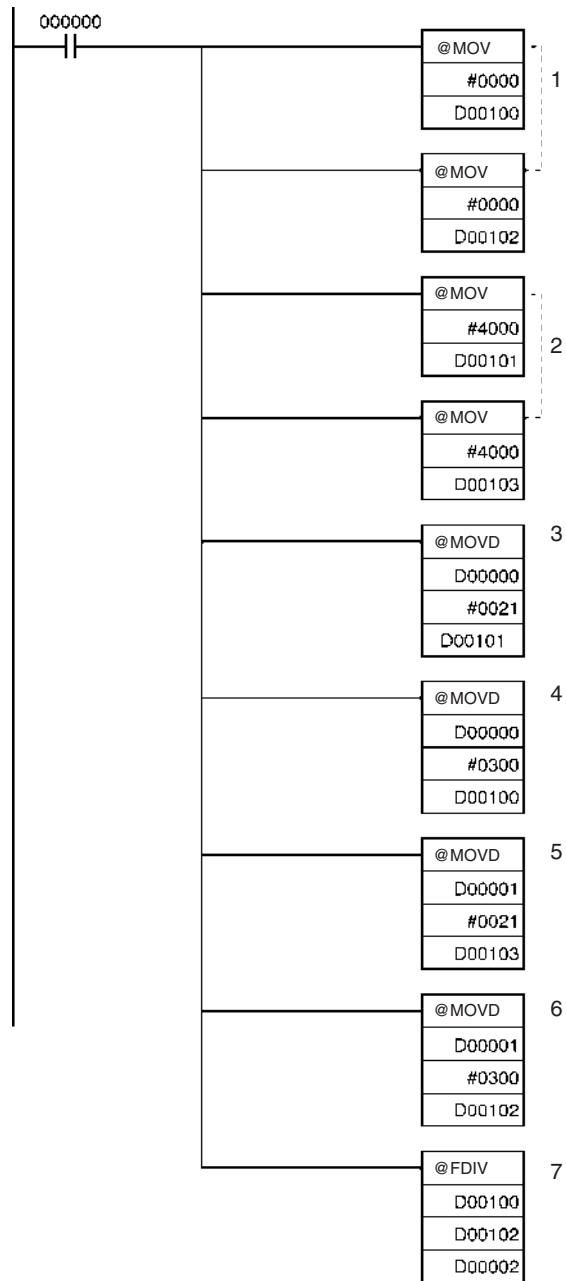
Divisione di base con virgola mobile

Nell'esempio seguente, quando CIO 000000 è impostato su ON, FDIV(079) divide il numero a virgola mobile in D00101 e D00100 per il numero a virgola mobile in CIO 0021 e CIO 0020 e scrive il risultato in D00301 e D00300.

**Divisione con virgola mobile di due numeri in formato BCD**

Nell'esempio, il numero in formato BCD a 4 cifre di D00000 è diviso per il numero in formato BCD a 4 cifre di D00001 e il risultato a virgola mobile viene scritto in D00003 e D00002.

Per eseguire la divisione con virgola mobile, il valore BCD in D00000 viene convertito in formato a virgola mobile in D00101 e D00100 e il valore BCD in D00001 viene convertito in formato a virgola mobile in D00103 e D00102.



1,2,3...

1. D00100 e D00102 vengono impostati su 0000.
2. D00101 e D00103 vengono impostati su 4000.

| | | | | | | | |
|--------|---|---|---|--------|---|---|---|
| D00101 | | | | D00100 | | | |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ↑ | | | | ↑ | | | |
| 4000 | | | | 0000 | | | |

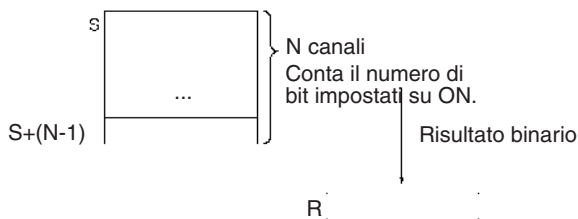
| | | | | | | | |
|--------|---|---|---|--------|---|---|---|
| D00103 | | | | D00102 | | | |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ↑ | | | | ↑ | | | |
| 4000 | | | | 0000 | | | |

3. MOVD(083) viene utilizzata per spostare le cifre dei canali di origine iniziali nelle cifre appropriate nei formati a virgola mobile del doppio canale.

| Area | N | S | R |
|--|--|-----|---------------|
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0001 a #FFFF (in formato binario) o da &1 a &65.535 | --- | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

BCNT(067) conteggia il numero totale dei bit impostati su ON in tutti i canali compresi tra S ed S+(N-1) e invia il risultato in R.



Flag

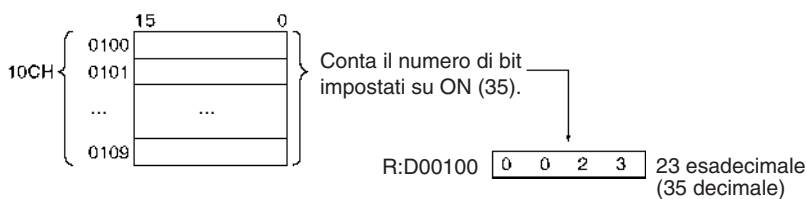
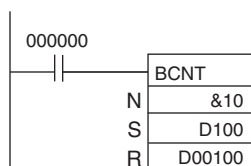
| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se N è 0000. ON se il risultato supera FFFF. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |

Avvertenze

Se N = 0000 o se il risultato supera FFFF, verrà generato un errore.

Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, BCNT(067) conta il numero totale dei bit impostati su ON nei 10 canali compresi tra CIO 0100 e CIO 0109 e scrive il risultato in D00100.



3-15 Istruzioni matematiche a virgola mobile

Le istruzioni matematiche a virgola mobile convertono i dati ed eseguono operazioni aritmetiche a virgola mobile. Le CPU della serie CS/CJ supportano le istruzioni indicate di seguito.

| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|-------------------------|------------------|-----------------|--------|
| FLOATING TO 16-BIT | FIX | 450 | 563 |
| FLOATING TO 32-BIT | FIXL | 451 | 565 |
| 16-BIT TO FLOATING | FLT | 452 | 566 |
| 32-BIT TO FLOATING | FLTL | 453 | 568 |
| FLOATING-POINT ADD | +F | 454 | 570 |
| FLOATING-POINT SUBTRACT | -F | 455 | 572 |
| FLOATING-POINT MULTIPLY | *F | 456 | 574 |
| FLOATING POINT DIVIDE | /F | 457 | 576 |
| DEGREES TO RADIANS | RAD | 458 | 578 |
| RADIANS-TO-DEGREES | DEG | 459 | 579 |
| SINE | SIN | 460 | 581 |
| COSINE | COS | 461 | 583 |
| TANGENT | TAN | 462 | 585 |
| ARC SINE | ASIN | 463 | 587 |
| ARC COSINE | ACOS | 464 | 589 |
| ARC TANGENT | ATAN | 465 | 591 |
| SQUARE ROOT | SQRT | 466 | 593 |
| EXPONENT | EXP | 467 | 595 |
| LOGARITHM | LOG | 468 | 597 |
| EXPONENTIAL POWER | PWR | 840 | 599 |

Oltre alle istruzioni citate, le CPU CS1-H e CJ1-H supportano le seguenti istruzioni di confronto e conversione a virgola mobile. Per ulteriori informazioni sulle istruzioni a virgola mobile in doppia precisione, consultare *3-16-21 Istruzioni di ingresso a virgola mobile in doppia precisione*.

| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|--|--|-----------------|--------|
| Istruzioni di confronto con simboli a virgola mobile in singola precisione (*solo CPU CS1-H, CJ1-H, CJ1M e CS1D) | LD, AND, OR + =F, <>F, <F, <=F, >F o >= F | 329 ... 334 | 600 |
| FLOATING-POINT TO ASCII (solo CPU CS1-H, CJ1-H e CJ1M) | FSTR | 448 | 604 |
| ASCII TO FLOATING-POINT (solo CPU CS1-H, CJ1-H e CJ1M) | FVAL | 449 | 609 |

Formato dei dati

I dati a virgola mobile esprimono numeri reali che utilizzano un segno, un esponente e una mantissa. Quando i dati vengono sono in formato a virgola mobile, viene applicata la formula seguente.

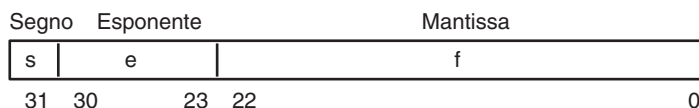
$$\text{Numero reale} = (-1)^s 2^{e-127} (1.f)$$

s: segno

e: esponente

f: mantissa

Il formato dei dati a virgola mobile è conforme agli standard IEEE754. I dati vengono espressi in 32 bit, come indicato qui di seguito:



| Dati | N. di bit | Contenuto |
|--------------|-----------|--|
| s: segno | 1 | 0: positivo, 1: negativo |
| e: esponente | 8 | Il valore dell'esponente (e) è compreso tra 0 e 255. L'esponente effettivo è il valore che resta dopo la sottrazione di 127 da e; tale valore è compreso tra -127 e 128. "e=0" ed "e=255" esprimono numeri speciali. |
| f: mantissa | 23 | La parte di mantissa dei dati binari a virgola mobile corrisponde al formale $2.0 > 1.f \geq 1.0$. |

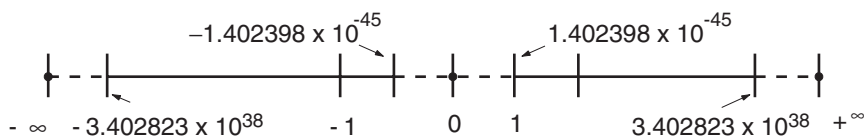
Numero di cifre

Il numero delle cifre effettive per i dati a virgola mobile è 24 bit per il formato binario (approssimativamente sette cifre per il formato decimale).

Dati a virgola mobile

I dati indicati qui di seguito possono essere espressi come dati a virgola mobile:

- $-\infty$
- $-3,402823 \times 10^{38} \leq \text{valore} \leq -1,402398 \times 10^{-45}$
- 0
- $1,402398 \times 10^{-45} \leq \text{valore} \leq 3,402823 \times 10^{38}$
- $+\infty$
- Non un numero (NaN, Not a number)



Numeri speciali

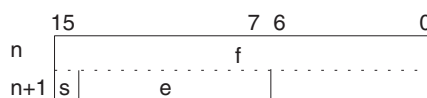
I formati per il tipo NaN, $\pm\infty$ e 0 sono i seguenti:

- NaN*: e = 255, f ≠ 0
 $+\infty$: e = 255, f = 0, s = 0
 $-\infty$: e = 255, f = 0, s = 1
 0: e = 0

*NaN non è un numero a virgola mobile valido. L'esecuzione di istruzioni di calcolo a virgola mobile non produce risultati di tipo NaN.

Scrittura di dati a virgola mobile

Quando nella finestra di modifica della memoria di I/O di CX-Programmer viene specificato il formato dei dati a virgola mobile, i numeri decimali standard inseriti nella finestra vengono convertiti automaticamente nel formato a virgola mobile. Una volta monitorati nella finestra, i dati scritti nel formato IEEE754 vengono convertiti automaticamente nel formato decimale standard.



Nella lettura e scrittura di dati a virgola mobile, non è necessario conoscere il formato dei dati IEEE754. È sufficiente ricordare che i valori a virgola mobile occupano due canali ciascuno.

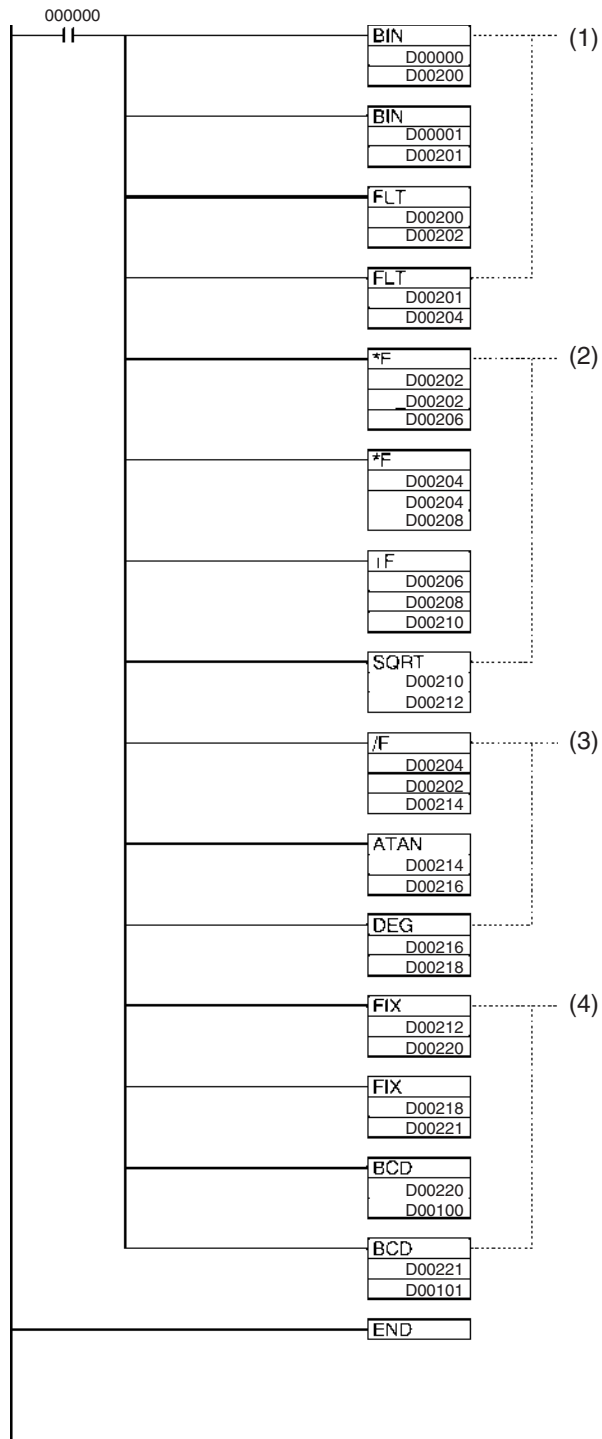
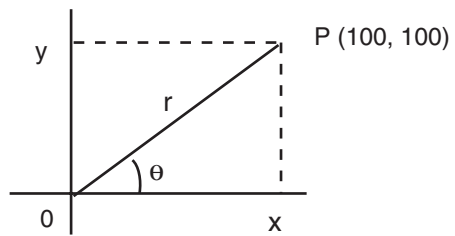
| | |
|-----------------|--|
| Infinito | Impostando il segno su 0 per i numeri positivi o su 1 per quelli negativi, è possibile esprimere i valori $+\infty$ e $-\infty$. L'esponente sarà 255 ($2^8 - 1$) e la mantissa 0. |
| NaN | Un NaN viene prodotto quando il risultato dei calcoli, ad esempio $0,0/0,0$, ∞/∞ o $\infty-\infty$ non corrisponde a un numero o a infinito. L'esponente sarà 255 ($2^8 - 1$) e la mantissa non sarà 0. |
| Nota | Non vi sono specifiche per il segno del NaN o per il valore del campo della mantissa, che comunque deve essere diverso da 0. |

Risultati aritmetici della virgola mobile

| | |
|---|--|
| Arrotondamento dei risultati | <p>Quando il numero di cifre nel risultato esatto del calcolo a virgola mobile supera quello delle cifre significative delle espressioni di elaborazione interna, utilizzare uno dei metodi seguenti per arrotondare il risultato.</p> <p>Se il risultato si avvicina a una su due espressioni a virgola mobile, verrà utilizzata quella più prossima. Se il risultato si trova in posizione intermedia tra due espressioni interne a virgola mobile, verrà arrotondato in modo che l'ultima cifra della mantissa sia 0.</p> |
| Overflow, Underflow e calcoli illegali | <p>In base al segno del risultato, l'overflow viene inviato come infinito positivo o negativo. In base al segno del risultato, l'underflow viene inviato come zero positivo o negativo.</p> <p>I calcoli illegali producono risultati di tipo NaN. Tra i calcoli illegali vengono incluse le seguenti operazioni: l'aggiunta di un numero con segno opposto, la sottrazione di infinito da un numero con segno opposto, la moltiplicazione di zero e infinito, la divisione di zero per zero o la divisione di infinito per infinito.</p> <p>Se nella conversione di un numero a virgola mobile in un numero intero si verifica un overflow, il valore del risultato potrebbe non essere corretto.</p> |
| Avvertenze per la gestione dei valori speciali | <p>Nelle operazioni con zero, infinito e NaN è necessario osservare quanto segue.</p> <ul style="list-style-type: none"> • La somma di zero positivo e zero negativo è zero positivo. • La differenza tra zeri dello stesso segno è zero positivo. • Se uno qualsiasi degli operandi è un NaN, il risultato sarà di tipo NaN. • Lo zero positivo e lo zero negativo vengono considerati equivalenti nei confronti. • Le verifiche di confronto o di equivalenza su uno o più NaN saranno sempre vere per != e sempre false per tutte le altre istruzioni. |

Risultati del calcolo a virgola mobile

| | |
|----------------|---|
| Esempio | <p>Quando il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$. Se il risultato è positivo, viene inviato come $+\infty$; se è negativo, come $-\infty$.</p> <p>Il flag di uguaglianza viene attivato solo quando esponente (e) e mantissa (f) sono entrambi 0 in seguito a un calcolo. Un risultato di calcolo verrà inoltre inviato come zero quando il valore assoluto del risultato è inferiore al valore minimo che può essere rappresentato come dato a virgola mobile. In questo caso viene attivato il flag di underflow.</p> <p>In questo esempio di programma, le coordinate x e y dell'asse X e dell'asse Y hanno un contenuto a 4 cifre in formato BCD in D00000 e D00001. Vengono rilevati la distanza (r) dall'origine e l'angolo (θ, in gradi) e inviati a D00100 e D00101. Nel risultato, le cifre a destra della virgola decimale vengono troncate.</p> |
|----------------|---|



Calcoli

Distanza $r = \sqrt{x^2 + y^2}$

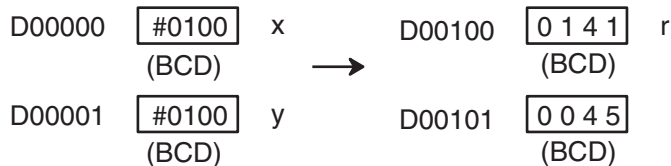
Angolo = $\tan^{-1} \left(\frac{y}{x} \right)$

Esempio

Distanza $r = \sqrt{100^2 + 100^2} = 141,4214$

Angolo = $\tan^{-1} \left(\frac{100}{100} \right) = 45.0$

Contenuto DM



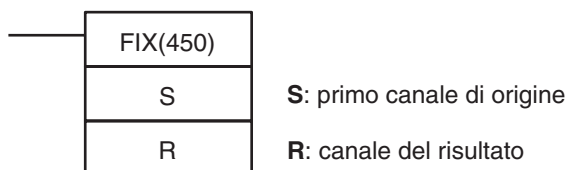
1. In questa sezione del programma vengono convertiti i dati dal formato BCD al formato a virgola mobile.
 - a) L'area dati a partire da D00200 viene utilizzata come area di lavoro.
 - b) In primo luogo BIN(023) viene utilizzata per convertire temporaneamente i dati dal formato BCD a quello binario; quindi FLT(452) serve per convertire i dati dal formato binario a quello a virgola mobile.
 - c) Il valore di x convertito in dato a virgola mobile viene inviato a D00203 e D00202.
 - d) Il valore di y convertito in dato a virgola mobile viene inviato a D00205 e D00204.
2. Per trovare la distanza r, vengono utilizzate le istruzioni matematiche a virgola mobile, che calcolano la radice quadrata di x^2+y^2 . Il risultato viene quindi inviato a D00213 e D00212 come dato a virgola mobile.
3. Per trovare l'angolo θ , vengono utilizzate le istruzioni matematiche a virgola mobile, che calcolano $\tan^{-1} (y/x)$. ATAN(465) invia il risultato in radianti; DEG(459) viene utilizzata per la conversione in gradi. Il risultato viene quindi inviato a D00219 e D00218 come dato a virgola mobile.
4. I dati vengono quindi nuovamente convertiti dal formato a virgola mobile e quello BCD.
 - a) In primo luogo FIX(450) viene utilizzata per convertire temporaneamente i dati dal formato a virgola mobile a quello binario; quindi BCD(024) serve per convertire i dati dal formato binario a quello BCD.
 - b) La distanza viene inviata a D00100.
 - c) L'angolo θ viene inviato a D00101.

3-15-1 FLOATING TO 16-BIT: FIX(450)

Scopo

Converte un valore a virgola mobile a 32 bit in dati binari con segno a 16 bit e invia il risultato al canale del risultato specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FIX(450) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FIX(450) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

FIX(450) converte la parte intera del numero a virgola mobile a 32 bit in S+1 ed S (formato IEEE754) in un dato binario con segno a 16 bit e invia il risultato a R.



Viene convertita solo la parte intera del dato a virgola mobile; la parte frazionaria viene troncata. La parte intera del dato a virgola mobile deve essere compresa tra -32.768 e 32.767.

Conversioni di esempio

Un valore a virgola mobile di 3,5 viene convertito a 3.

Un valore a virgola mobile di -3,5 viene convertito a -3.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati in S+1 ed S sono di tipo NaN. ON se la parte intera di S+1 ed S non è compresa tra -32.768 e 32.767. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 del risultato è ON. OFF in tutti gli altri casi. |

Avvertenze

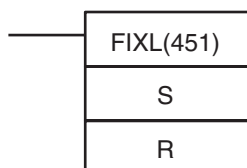
Il contenuto di S+1 ed S deve essere costituito da dati a virgola mobile e la parte intera deve essere compresa tra -32.768 e 32.767.

3-15-2 FLOATING TO 32-BIT: FIXL(451)

Scopo

Converte un valore a virgola mobile a 32 bit in dati binari con segno a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FIXL(451) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FIXL(451) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--------------------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |

| Area | S | R |
|--|--|-----|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

FIXL(451) converte la parte intera del numero a virgola mobile a 32 bit in S+1 ed S (formato IEEE754) in un dato binario con segno a 32 bit e invia il risultato a R+1 ed R.



Viene convertita solo la parte intera del dato a virgola mobile; la parte frazionaria viene troncata (la parte intera del dato a virgola mobile deve essere compresa tra -2.147.483.648 e 2.147.483.647).

Conversioni di esempio

Un valore a virgola mobile di 2.147.483.640,5 viene convertito a 2.147.483.640.

Un valore a virgola mobile di -214.748.340,5 viene convertito a -214.748.340.

Flag

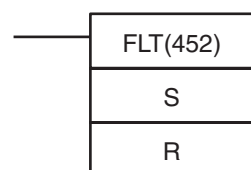
| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati in S+1 ed S sono di tipo NaN. ON se la parte intera di S+1 ed S non è compresa tra -2.147.483.648 e 2.147.483.647. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di R+1 è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

Avvertenze

Il contenuto di S+1 ed S deve essere costituito da dati a virgola mobile e la parte intera deve essere compresa tra -2.147.483.648 e 2.147.483.647.

3-15-3 16-BIT TO FLOATING: FLT(452)**Scopo**

Converte un valore binario con segno a 16 bit in dati a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder

S: canale di origine

R: primo canale del risultato

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FLT(452) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FLT(452) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 |
| Area bit ausiliaria | Da A000 a A959 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | --- |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

FLT(452) converte il valore binario con segno a 16 bit in S in un dato a virgola mobile a 32 bit (formato IEEE754) e invia il risultato a R+1 ed R. Nel risultato a virgola mobile viene aggiunto un solo 0 dopo il decimale.



Per S è possibile specificare solo valori compresi tra -32.768 e 32.767. Per convertire dati binari con segno esterni a tale intervallo, utilizzare FLT(453).

Conversioni di esempio

Un valore binario con segno di 3 viene convertito a 3,0.

Un valore binario con segno di -3 viene convertito a -3,0.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

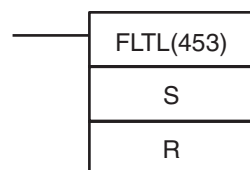
Il contenuto di S deve essere costituito da dati binari con segno con un valore (decimale) compreso tra -32.768 e 32.767.

3-15-4 32-BIT TO FLOATING: FLTL(453)

Scopo

Converte un valore binario con segno a 32 bit in dati a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | FLTL(453) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FLTL(453) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

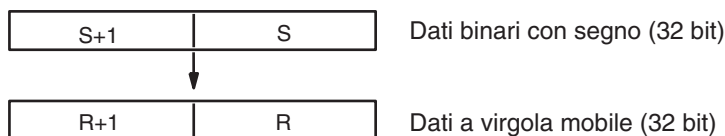
Caratteristiche operando

| Area | S | R |
|-------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |

| Area | S | R |
|--|--|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

FLTL(453) converte il valore binario con segno a 32 bit in S+1 ed S in un dato a virgola mobile a 32 bit (formato IEEE754) e invia il risultato a R+1 ed R. Nel risultato a virgola mobile viene aggiunto un solo 0 dopo il decimale.



Per S+1 ed S è possibile specificare dati binari con segno compresi tra -2.147.483.648 e 2.147.483.647. Il valore a virgola mobile ha 24 cifre binarie significative (bit). Se un numero maggiore di 16.777.215 (valore massimo esprimibile con 24 bit) viene convertito da FLTL(453), il risultato non sarà esatto.

Conversioni di esempio

Un valore binario con segno di 16.777.215 viene convertito a 16.777.215,0.

Un valore binario con segno di -16.777.215 viene convertito a -15.777.215,0.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

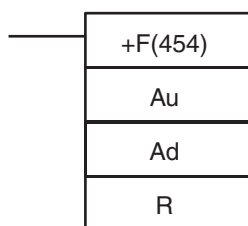
Avvertenze

Se viene convertito un numero con valore assoluto maggiore di 16.777.215 (valore massimo esprimibile con 24 bit), il risultato non sarà esatto.

3-15-5 FLOATING-POINT ADD: +F(454)

Scopo Somma due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



Au: primo canale augendo

AD: primo canale addendo

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +F(454) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ +F(454) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

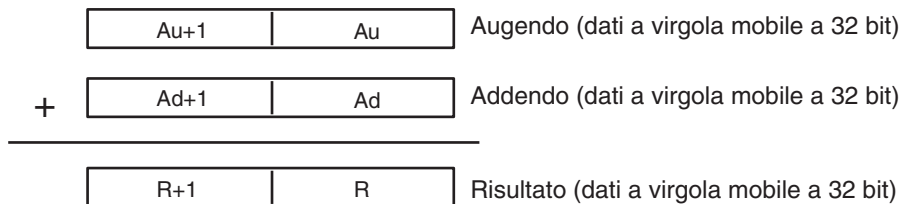
| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Au | Ad | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF7 (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

+F(454) aggiunge il numero a virgola mobile a 32 bit in A_{d+1} e A_d al numero a virgola mobile a 32 bit in A_{u+1} e A_u e invia il risultato a R_{+1} ed R (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni per dati augendo e addendo producono i risultati indicati nella tabella seguente.

| Addendo | Augendo | | | | NaN |
|-----------|-----------|------------------|------------------|------------------|------------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | 0 | Numerico | $+\infty$ | $-\infty$ | Vedere nota 2 |
| Numerico | Numerico | Vedere nota 1 | $+\infty$ | $-\infty$ | |
| $+\infty$ | $+\infty$ | $+\infty$ | $+\infty$ | Vedere nota 2 | |
| $-\infty$ | $-\infty$ | $-\infty$ | Vedere nota 2 | $-\infty$ | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Viene attivato il flag di errore e l'istruzione non viene eseguita.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati augendo o addendo non vengono riconosciuti come dati a virgola mobile. ON se i dati augendo o addendo sono di tipo NaN. ON se vengono aggiunti $+\infty$ e $-\infty$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati augendo (A_{u+1} e A_u) e addendo (A_{d+1} e A_d) siano in formato dati a virgola mobile IEEE754.

3-15-6 FLOATING-POINT SUBTRACT: -F(455)

Scopo

Esegue la sottrazione tra due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder

| | |
|--|---------|
| | -F(455) |
| | Mi |
| | Su |
| | R |

Mi: primo canale minuendo

Su: primo canale sottraendo

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | -F(455) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-F(455) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Are di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Mi | Su | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

–F(455) sottrae il numero a virgola mobile a 32 bit in Su +1 e Su dal numero a virgola mobile a 32 bit in Mi+1 e Mi e invia il risultato a R+1 ed R (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni per dati minuendo e sottraendo producono i risultati indicati nella tabella seguente.

| Sottraendo | Minuendo | | | | NaN |
|------------|-----------|---------------|---------------|---------------|---------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | 0 | Numerico | $+\infty$ | $-\infty$ | Vedere nota 2 |
| Numerico | Numerico | Vedere nota 1 | $+\infty$ | $-\infty$ | |
| $+\infty$ | $-\infty$ | $-\infty$ | Vedere nota 2 | $-\infty$ | |
| $-\infty$ | $+\infty$ | $+\infty$ | $+\infty$ | Vedere nota 2 | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Viene attivato il flag di errore e l'istruzione non viene eseguita.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i dati minuendo o sottraendo non vengono riconosciuti come dati a virgola mobile. ON se i dati minuendo o sottraendo sono di tipo NaN. ON se $+\infty$ viene sottratto da $+\infty$. ON se $-\infty$ viene sottratto da $-\infty$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

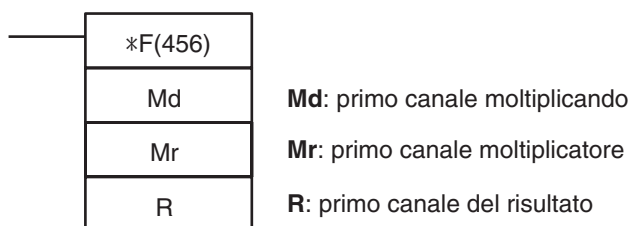
Avvertenze

È necessario che i dati minuendo (Mi+1 e Mi) e sottraendo (Su+1 e Su) siano in formato dati a virgola mobile IEEE754.

3-15-7 FLOATING-POINT MULTIPLY: *F(456)

Scopo Moltiplica due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | *F(456) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ *F(456) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

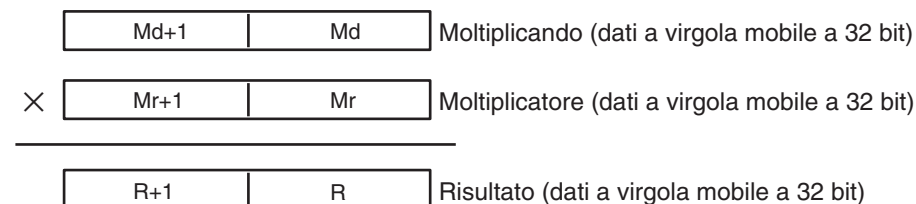
| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Md | Mr | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

*F(456) moltiplica il numero a virgola mobile a 32 bit in Md+1 ed Md per il numero a virgola mobile a 32 bit in Mr+1 ed Mr e invia il risultato a R+1 ed R (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni per dati moltiplicando e moltiplicatore producono i risultati indicati nella tabella seguente.

| Moltiplicatore | Moltiplicando | | | | NaN |
|-----------------------------|---------------|---------------|---------------|---------------|---------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | 0 | 0 | Vedere nota 2 | Vedere nota 2 | Vedere nota 2 |
| Numerico | 0 | Vedere nota 1 | $+/-\infty$ | $+/-\infty$ | |
| $+\infty$ | Vedere nota 2 | $+/-\infty$ | $+\infty$ | $-\infty$ | |
| $-\infty$ | Vedere nota 2 | $+/-\infty$ | $-\infty$ | $+\infty$ | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Viene attivato il flag di errore e l'istruzione non viene eseguita.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i dati moltiplicando o moltiplicatore non vengono riconosciuti come dati a virgola mobile. ON se i dati moltiplicando o moltiplicatore sono di tipo NaN. ON se vengono moltiplicati $+\infty$ e 0. ON se vengono moltiplicati $-\infty$ e 0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

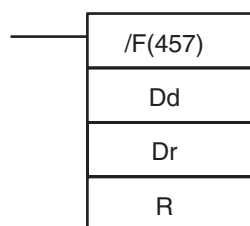
È necessario che i dati moltiplicando (Md+1 ed Md) e moltiplicatore (Mr+1 ed Mr) siano in formato dati a virgola mobile IEEE754.

3-15-8 FLOATING-POINT DIVIDE: /F(457)

Scopo

Esegue la divisione tra due numeri a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



Dd: primo canale dividendo

Dr: primo canale divisore

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | /F(457) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/F(457) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

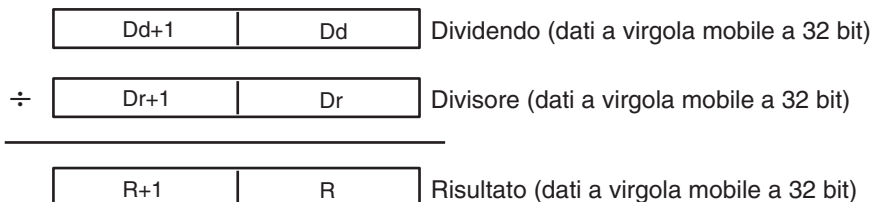
| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | Dd | Dr | R |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritentività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF7 (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

/F(457) divide il numero a virgola mobile a 32 bit in Dd+1 e Dd per il numero a virgola mobile a 32 bit in Dr+1 e Dr e invia il risultato a R+1 ed R (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni per dati dividendo e divisore producono i risultati indicati nella tabella seguente.

| Divisore | Dividendo | | | | NaN |
|-----------|---------------|---------------|---------------|---------------|---------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | Vedere nota 3 | $+/-\infty$ | $+\infty$ | $-\infty$ | Vedere nota 3 |
| Numerico | 0 | Vedere nota 1 | $+/-\infty$ | $+/-\infty$ | |
| $+\infty$ | 0 | Vedere nota 2 | Vedere nota 3 | Vedere nota 3 | |
| $-\infty$ | 0 | Vedere nota 2 | Vedere nota 3 | Vedere nota 3 | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Per gli underflow i risultati saranno zero.
 3. Viene attivato il flag di errore e l'istruzione non viene eseguita.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati dividendo o divisore non vengono riconosciuti come dati a virgola mobile. ON se i dati dividendo o divisore sono di tipo NaN. ON se dividendo e divisore sono entrambi 0. ON se dividendo e divisore sono entrambi $+\infty$ o $-\infty$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per potere essere espresso come valore a virgola mobile a 32 bit. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

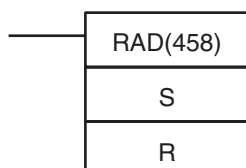
Avvertenze

È necessario che i dati dividendo (Dd+1 e Dd) e divisore (Dr+1 e Dr) siano in formato dati a virgola mobile IEEE754.

3-15-9 DEGREES TO RADIANS: RAD(458)

Scopo Convertere un numero a virgola mobile a 32 bit da gradi a radianti e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RAD(458) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RAD(458) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

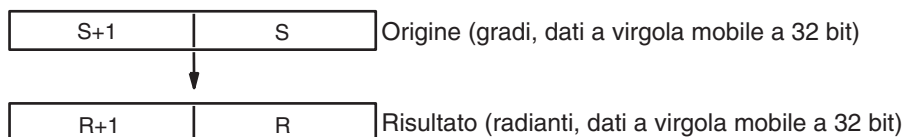
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

RAD(458) converte il numero a virgola mobile a 32 bit in S+1 ed S da gradi a radianti e invia il risultato a R ed R+1 (i dati di origine a virgola mobile devono essere nel formato IEEE754).



I gradi vengono convertiti in radianti con la formula seguente:

$$\text{gradi} \times \pi/180 = \text{radianti}$$

Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Flag

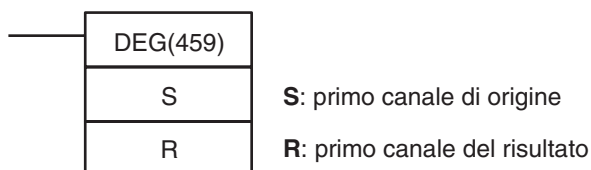
| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-10 RADIANS TO DEGREES: DEG(459)**Scopo**

Converte un numero a virgola mobile a 32 bit da radianti a gradi e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | DEG(459) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DEG(459) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

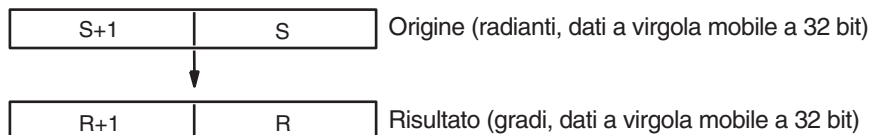
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritentività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a -2048 a +2047 ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

DEG(459) converte il numero a virgola mobile a 32 bit in S+1 ed S da radianti a gradi e invia il risultato a R+1 ed R (i dati di origine a virgola mobile devono essere nel formato IEEE754).



I radianti vengono convertiti in gradi con la formula seguente:

$$\text{radianti} \times \pi/180 = \text{gradi}$$

Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

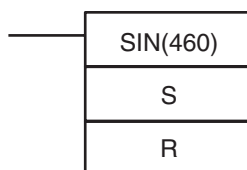
È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-11 SINE: SIN(460)

Scopo

Calcola il seno di un numero a virgola mobile a 32 bit in radianti e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | SIN(460) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SIN(460) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

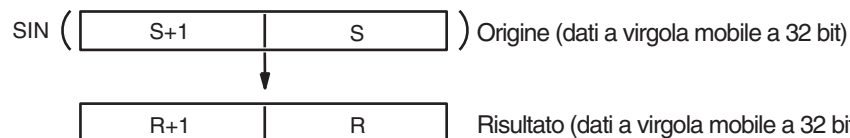
| Area | S | R |
|-------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |

| Area | S | R |
|--|--|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

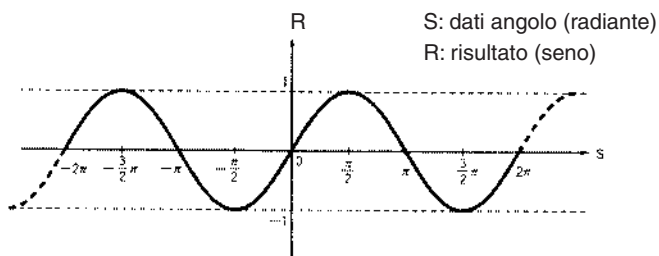
SIN(460) calcola il seno dell'angolo in radianti rappresentato come valore a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R.

I dati di origine a virgola mobile devono essere nel formato IEEE754.



Indicare l'angolo desiderato (da -65.535 a 65.535) in radianti in S+1 ed S. Se tale angolo non rientra nell'intervallo specificato, viene generato un errore e l'istruzione non sarà eseguita. Per ulteriori informazioni sulla conversione da gradi a radianti, consultare 3-15-19 LOGARITHM: LOG(468) DEGREES-TO-RADIANS: RAD(458).

Nel diagramma seguente viene illustrata la relazione tra l'angolo e il risultato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine è maggiore di 65.535. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | OFF |

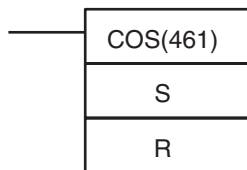
| Nome | Eti-chetta | Funzionamento |
|-------------------|------------|--|
| Flag di underflow | UF | OFF |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-12 COSINE: COS(461)**Scopo**

Calcola il coseno di un numero a virgola mobile a 32 bit in radianti e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder

S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | COS(461) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @COS(461) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--------------------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |

| Area | S | R |
|--|--|-----|
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | |

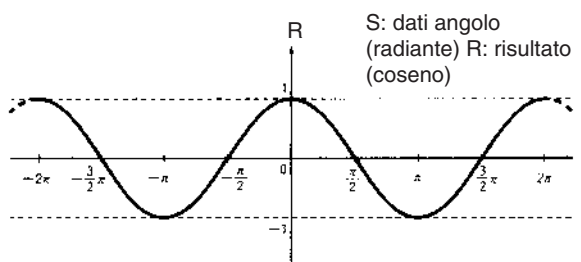
Descrizione

COS(461) calcola il coseno dell'angolo in radianti rappresentato come valore a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R. I dati di origine a virgola mobile devono essere nel formato IEEE754.



Indicare l'angolo desiderato (da -65.535 a 65.535) in radianti in S+1 ed S. Se tale angolo non rientra nell'intervallo specificato, viene generato un errore e l'istruzione non sarà eseguita. Per ulteriori informazioni sulla conversione da gradi a radianti, consultare 3-15-9 DEGREES TO RADIANS: RAD(458).

Nel diagramma seguente viene illustrata la relazione tra l'angolo e il risultato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine è maggiore di 65.535. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | OFF |
| Flag di underflow | UF | OFF |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

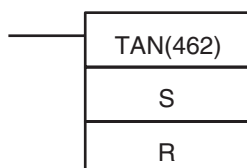
È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-13 TANGENT: TAN(462)

Scopo

Calcola la tangente di un numero a virgola mobile a 32 bit in radianti e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | TAN(462) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TAN(462) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

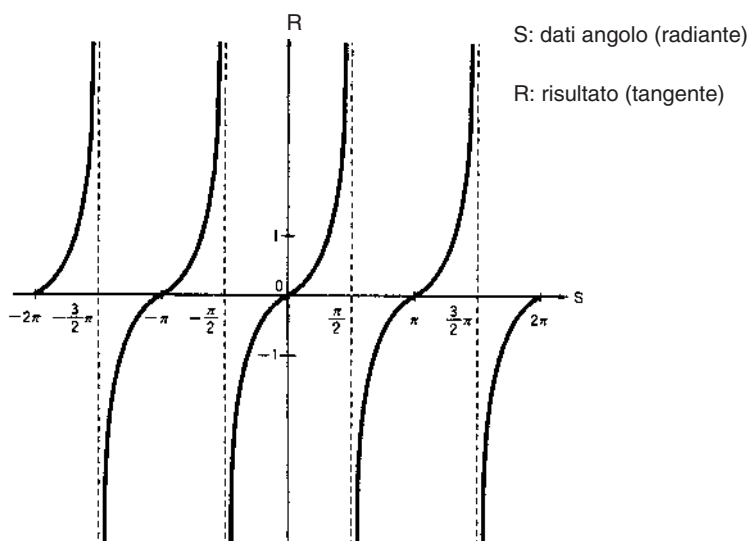
TAN(462) calcola la tangente dell'angolo in radianti rappresentata come valore a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R. I dati di origine a virgola mobile devono essere nel formato IEEE754.



Indicare l'angolo desiderato (da -65.535 a 65.535) in radianti in S+1 ed S. Se tale angolo non rientra nell'intervallo specificato, viene generato un errore e l'istruzione non sarà eseguita. Per ulteriori informazioni sulla conversione da gradi a radianti, consultare 3-15-9 DEGREES TO RADIANS: RAD(458).

Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Nel diagramma seguente viene illustrata la relazione tra l'angolo e il risultato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine è maggiore di 65.535. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | OFF |
| Flag di underflow | UF | OFF |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

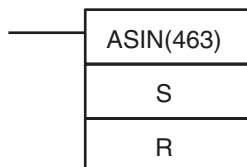
È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-14 ARC SINE: ASIN(463)

Scopo

Calcola l'arcoseno di un numero a virgola mobile a 32 bit e invia il risultato nei canali del risultato specificati. La funzione di arcoseno è l'inverso della funzione di seno e restituisce l'angolo che produce un dato valore di seno compreso tra -1 e 1.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ASIN(463) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASIN(463) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

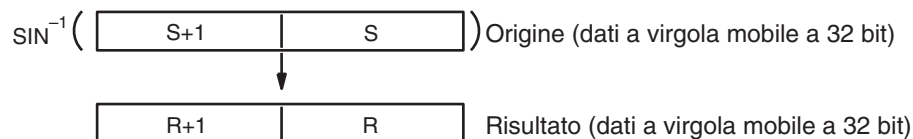
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

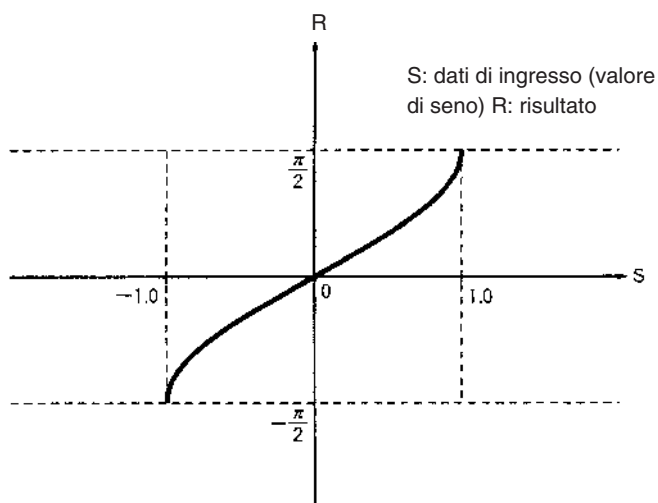
ASIN(463) calcola l'angolo in radianti per un valore di seno rappresentato come numero a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R. I dati di origine a virgola mobile devono essere nel formato IEEE754.



Il dato di origine deve essere compreso tra -1,0 e 1,0. Se il valore assoluto dei dati di origine è superiore a 1,0, viene generato un errore e l'istruzione non sarà eseguita.

Il risultato viene inviato a R+1 ed R come angolo in radianti compreso tra $-\pi/2$ e $\pi/2$.

Nel diagramma seguente viene illustrata la relazione tra i dati di ingresso e il risultato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine è maggiore di 1,0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | OFF |
| Flag di underflow | UF | OFF |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

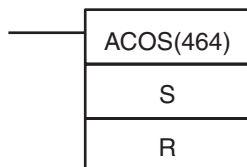
È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-15 ARC COSINE: ACOS(464)

Scopo

Calcola l'arcocoseno di un numero a virgola mobile a 32 bit e invia il risultato nei canali del risultato specificati. La funzione di arcocoseno è l'inverso della funzione di coseno e restituisce l'angolo che produce un dato valore di coseno compreso tra -1 e 1.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ACOS(464) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ACOS(464) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

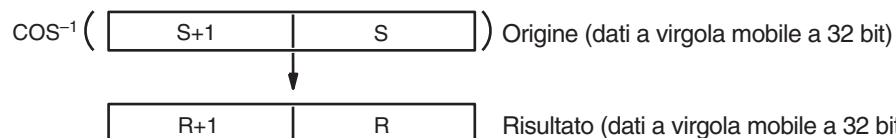
Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

ACOS(464) calcola l'angolo in radianti per un valore di coseno rappresentato come numero a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R.

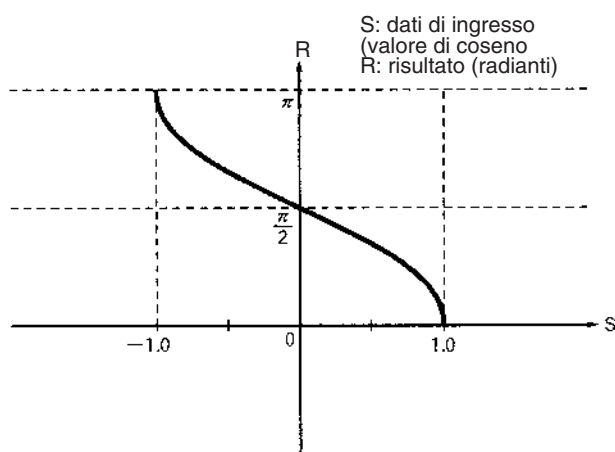
I dati di origine a virgola mobile devono essere nel formato IEEE754.



Il dato di origine deve essere compreso tra -1,0 e 1,0. Se il valore assoluto dei dati di origine è superiore a 1,0, viene generato un errore e l'istruzione non sarà eseguita.

Il risultato viene inviato a R+1 ed R come angolo in radianti compreso tra 0 e π .

Nel diagramma seguente viene illustrata la relazione tra i dati di ingresso e il risultato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine è maggiore di 1,0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | OFF |
| Flag di underflow | UF | OFF |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

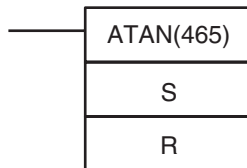
È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-16 ARC TANGENT: ATAN(465)

Scopo

Calcola l'arcotangente di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati. La funzione di arcotangente è l'inverso della funzione di tangente e restituisce l'angolo che produce un dato valore di tangente.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ATAN(465) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ATAN(465) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

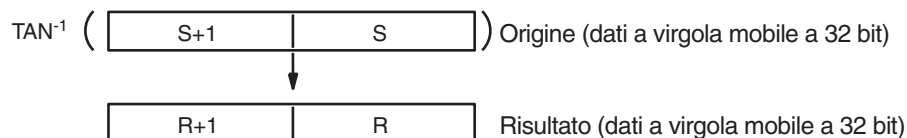
Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

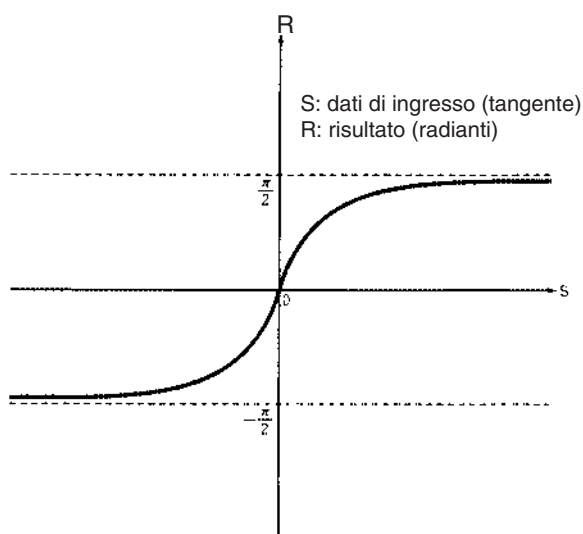
ATAN(465) calcola l'angolo in radianti per un valore di tangente rappresentato come numero a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R.

I dati di origine a virgola mobile devono essere nel formato IEEE754.



Il risultato viene inviato a R+1 ed R come angolo in radianti compreso tra $-\pi/2$ e $\pi/2$.

Nel diagramma seguente viene illustrata la relazione tra i dati di ingresso e il risultato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | OFF |
| Flag di underflow | UF | OFF |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

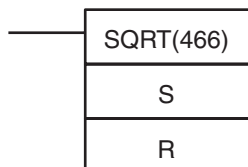
È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-17 SQUARE ROOT: SQRT(466)

Scopo

Calcola la radice quadrata di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder



S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SQRT(466) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SQRT(466) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

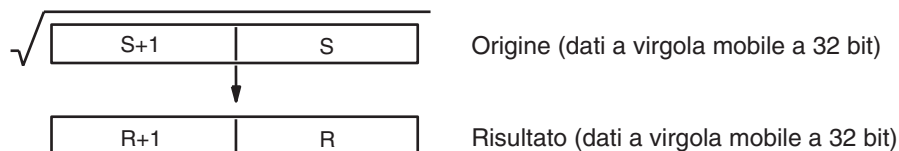
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

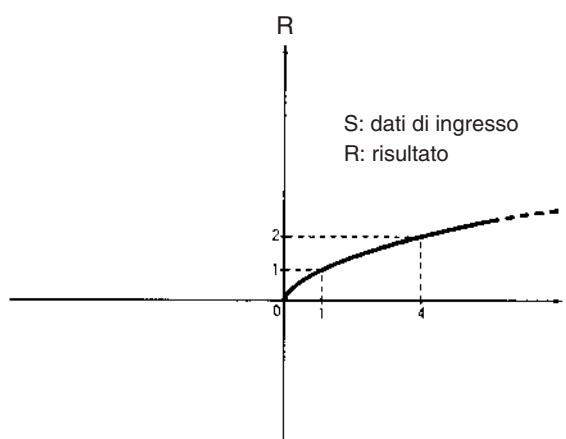
SQRT(466) calcola la radice quadrata del numero a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R (i dati di origine a virgola mobile devono essere nel formato IEEE754).



È necessario che i dati di origine siano positivi; se sono negativi, viene generato un errore e l'istruzione non sarà eseguita.

Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Nel diagramma seguente viene illustrata la relazione tra i dati di ingresso e il risultato.

**Flag**

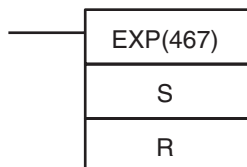
| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono negativi. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | OFF |
| Flag negativo | N | OFF |

Avvertenze

È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-18 EXPONENT: EXP(467)**Scopo**

Calcola il valore esponenziale naturale (in base e) di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder

S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | EXP(467) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @EXP(467) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

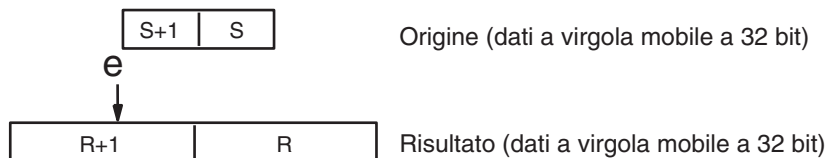
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a 4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

EXP(467) calcola il valore esponenziale naturale (in base e) del numero a virgola mobile a 32 bit in S+1 ed S e invia il risultato in R+1 ed R. In altri termini, EXP(467) calcola e^x (x = origine) e invia il risultato in R+1 ed R.

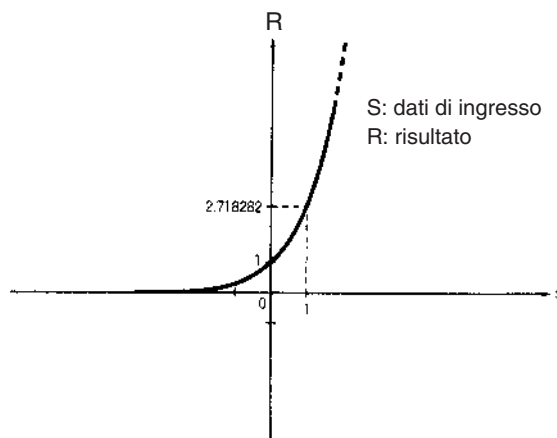


Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Nota La costante e è pari a 2,718282.

Nel diagramma seguente viene illustrata la relazione tra i dati di ingresso e il risultato.



Flag

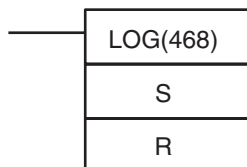
| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag negativo | N | OFF |

Avvertenze

È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-19 LOGARITHM: LOG(468)**Scopo**

Calcola il logaritmo naturale (in base e) di un numero a virgola mobile a 32 bit e invia il risultato ai canali del risultato specificati.

Simbolo programmazione ladder

S: primo canale di origine

R: primo canale del risultato

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | LOG(468) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @LOG(468) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

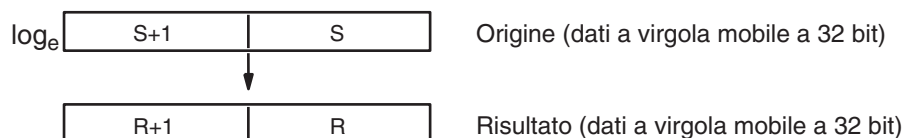
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | R |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

LOG(468) calcola il logaritmo naturale (in base e) del numero a virgola mobile a 32 bit in S+1 ed S e invia il risultato a R+1 ed R.

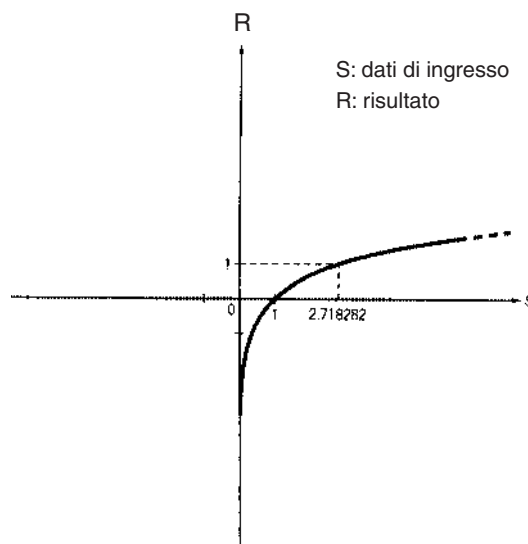


È necessario che i dati di origine siano positivi; se sono negativi, viene generato un errore e l'istruzione non sarà eseguita.

Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Nota La costante e è pari a 2,718282.

Nel diagramma seguente viene illustrata la relazione tra i dati di ingresso e il risultato.

**Flag**

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono negativi. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | OFF |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

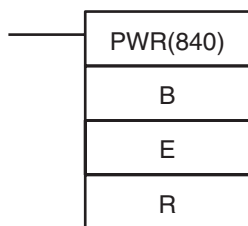
È necessario che i dati di origine in S+1 ed S siano nel formato dati a virgola mobile IEEE754.

3-15-20 EXPONENTIAL POWER: PWR(840)

Scopo

Eleva un numero a virgola mobile a 32 bit alla potenza di un altro numero a virgola mobile a 32 bit.

Simbolo programmazione ladder



B: primo canale base

E: primo canale esponente

R: primo canale del risultato

Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | PWR(840) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PWR(840) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

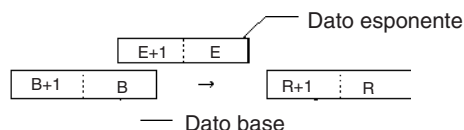
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | B | E | R |
|--|--|---|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | | |
| Area di lavoro | Da W000 a W510 | | |
| Area bit di ritenività | Da H000 a H510 | | |
| Area bit ausiliaria | Da A000 ad A958 | | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | | |
| Area del contatore | Da C0000 a C4094 | | |
| Area DM | Da D00000 a D32766 | | |
| Area EM senza banco | Da E00000 a E32766 | | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

PWR(840) eleva il numero a virgola mobile a 32 bit in B+1 e B alla potenza del numero a virgola mobile a 32 bit in E+1 ed E. In altri termini, PWR(840) calcola X^Y ($X = B+1$ e B ; $Y = E+1$ ed E).



Ad esempio, quando i canali base (B+1 e B) contengono 3,1 e i canali esponente (E+1 ed E) contengono 3, il risultato sarà $3,1^3$ oppure 29,791.

Se il valore assoluto del risultato è maggiore del valore massimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di overflow.

Se il valore assoluto del risultato è inferiore al valore minimo che può essere rappresentato come dato a virgola mobile, viene attivato il flag di underflow.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | ON se la base (B+1 e B) o l'esponente (E+1 ed E) non vengono riconosciuti come dati a virgola mobile. ON se la base (B+1 e B) o l'esponente (E+1 ed E) sono di tipo NaN. ON se la base (B+1 e B) è 0 e l'esponente (E+1 ed E) è minore di 0 (divisione per 0). ON se la base (B+1 e B) è negativa e l'esponente (E+1 ed E) è non intero (radice di un numero negativo). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo alto per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo basso per essere rappresentato come valore a virgola mobile a 32 bit. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che la base (B+1 e B) e l'esponente (E+1 ed E) siano nel formato dati a virgola mobile IEEE754.

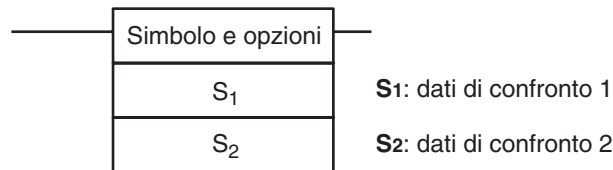
3-15-21 Istruzioni di confronto a virgola mobile in singola precisione**Scopo**

Le istruzioni di confronto di ingresso comparano due valori a virgola mobile in singola precisione (costanti a 32 bit in formato IEEE754 e/o il contenuto di canali specifici) e creano una condizione di esecuzione ON quando la condizione di confronto è vera.

Queste istruzioni sono supportate solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Nota Per ulteriori informazioni sulle istruzioni di confronto di ingresso in formato binario con e senza segno consultare *3-7-1 Istruzioni di confronto di ingresso (da 300 a 328)*; per ulteriori informazioni sulle istruzioni di confronto di ingresso a virgola mobile in doppia precisione consultare *3-16-21 Istruzioni di ingresso a virgola mobile in doppia precisione*.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|-------------------------------------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il confronto è vero | Istruzione di confronto di ingresso |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S ₁ | S ₂ |
|--|--|----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFF (binario) | |
| Registri dati | --- | |
| Registri indice | Da IR0 a IR15 (solo per dati senza segno) | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

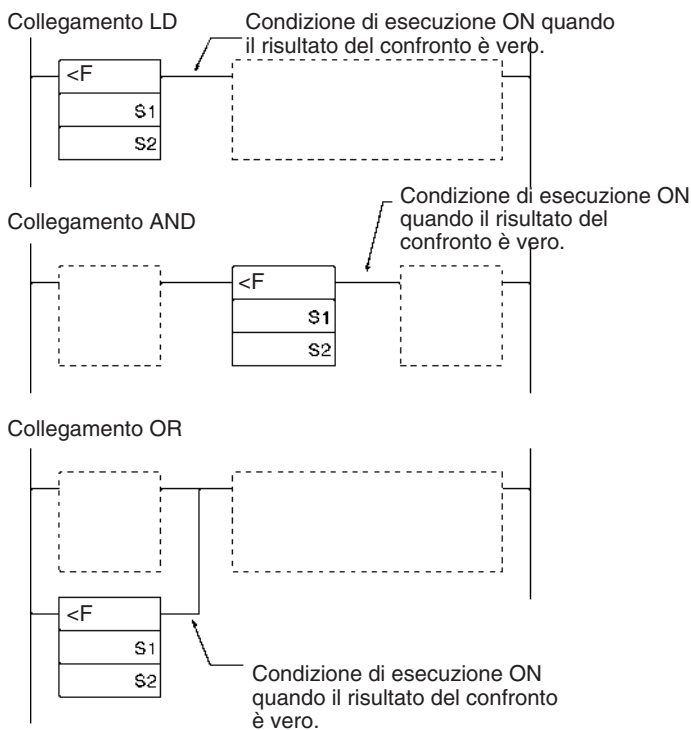
Descrizione

L'istruzione di confronto di ingresso compara i dati specificati in S₁ ed S₂ come valori a virgola mobile in singola precisione (dati a 32 bit in formato IEEE754) e crea una condizione di esecuzione ON quando la condizione di confronto è vera. Quando i dati vengono memorizzati nei canali, S₁ ed S₂ indicano i primi due canali che contengono i dati a 32 bit. Inoltre è possibile specificare il dato a virgola mobile come costante esadecimale a 8 cifre.

Specifica delle istruzioni

Le istruzioni di confronto di ingresso vengono utilizzate come le istruzioni LD, AND e OR per controllare l'esecuzione delle istruzioni successive.

| Tipo di ingresso | Funzionamento |
|------------------|---|
| LD | È possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |
| AND | Non è possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |
| OR | È possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |

**Opzioni**

I tre tipi di ingressi e i sei simboli danno luogo a 18 possibili combinazioni.

| Simbolo | Opzione (formato dei dati) |
|------------------------|--|
| = (uguale) | F: dati a virgola mobile in singola precisione |
| < > (diverso) | |
| < (minore di) | |
| <= (minore o uguale) | |
| > (maggiore di) | |
| >= (maggiore o uguale) | |

Sommario delle istruzioni di confronto di ingresso

Nella tabella seguente vengono riportati i codici di funzione, i mnemonici, i nomi e le funzioni delle 18 istruzioni di confronto di ingresso a virgola mobile in singola precisione ($C1=S_1+1$, S_1 e $C2=S_2+1$, S_2).

| Codice | Codice mnemonico | Nome | Funzione |
|--------|------------------|---------------------|--------------------------------|
| 329 | LD=F | LOAD FLOATING EQUAL | Condizione vera se $C1 = C2$: |
| | AND=F | AND FLOATING EQUAL | |
| | OR=F | OR FLOATING EQUAL | |

| Codice | Codice mnemonico | Nome | Funzione |
|--------|------------------|-------------------------------------|---------------------------------|
| 330 | LD <>F | LOAD FLOATING NOT EQUAL | Condizione vera se $C1 \neq C2$ |
| | AND <>F | AND FLOATING NOT EQUAL | |
| | OR <>F | OR FLOATING NOT EQUAL | |
| 331 | LD <F | LOAD FLOATING LESS THAN | Condizione vera se $C1 < C2$ |
| | AND <F | AND FLOATING LESS THAN | |
| | OR <F | OR FLOATING LESS THAN | |
| 332 | LD <=F | LOAD FLOATING LESS THAN OR EQUAL | Condizione vera se $C1 \leq C2$ |
| | AND <=F | AND FLOATING LESS THAN OR EQUAL | |
| | OR <=F | OR FLOATING LESS THAN OR EQUAL | |
| 333 | LD >F | LOAD FLOATING GREATER THAN | Condizione vera se $C1 > C2$ |
| | AND >F | AND FLOATING GREATER THAN | |
| | OR >F | OR FLOATING GREATER THAN | |
| 325 | LD >=F | LOAD FLOATING GREATER THAN OR EQUAL | Condizione vera se $C1 \geq C2$ |
| | AND >=F | AND FLOATING GREATER THAN OR EQUAL | |
| | OR >=F | OR FLOATING GREATER THAN OR EQUAL | |

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------------|------------|--|
| Flag di errore | ER | ON se S_1+1, S_1 o S_2+1, S_2 non è un numero a virgola mobile valido (NaN). ON se S_1+1, S_1 o S_2+1, S_2 è $+\infty$. ON se S_1+1, S_1 o S_2+1, S_2 è $-\infty$. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se $S_1+1, S_1 > S_2+1, S_2$. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | > = | ON se $S_1+1, S_1 \geq S_2+1, S_2$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se $S_1+1, S_1 = S_2+1, S_2$. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | ≠ | ON se $S_1+1, S_1 \neq S_2+1, S_2$. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se $S_1+1, S_1 < S_2+1, S_2$. OFF in tutti gli altri casi. |
| Flag di minore o uguale | < = | ON se $S_1+1, S_1 \leq S_2+1, S_2$. OFF in tutti gli altri casi. |
| Flag negativo | N | Non modificato |

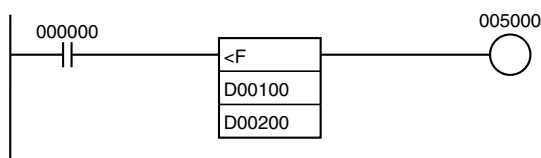
Avvertenze

Non è possibile utilizzare le istruzioni di confronto di ingresso come istruzioni sul lato destro: tra queste istruzioni e la barra di distribuzione destra è infatti necessario specificare un'altra istruzione.

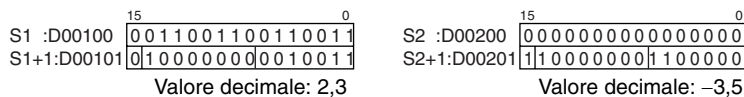
Esempio

AND FLOATING LESS THAN: AND<F(331)

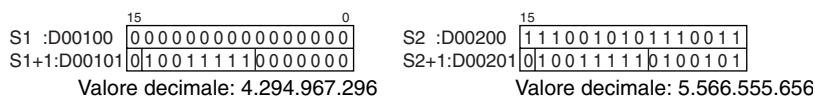
Nell'esempio seguente, quando CIO 000000 è impostato su ON, il dato a virgola mobile in D00101, D00100 viene comparato con il dato a virgola mobile in D00201, D00200. Se il contenuto di D00101, D00100 è minore di quello di D00201, D00200, l'esecuzione procede alla riga successiva e CIO 005000 viene impostato su ON. Se il contenuto di D00101, D00100 non è minore di quello di D00201, D00200, l'esecuzione non procede alla riga successiva dell'istruzione.



Confronto FLOATING LESS THAN (<F)



2.3 > -3.5
 ↓
 Non produce una condizione ON



4294967296 < 5566555656
 ↓
 Produce una condizione ON

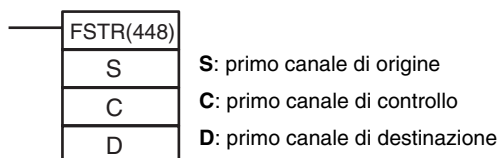
3-15-22 FLOATING-POINT TO ASCII: FSTR(448)

Scopo

Rappresenta un valore a virgola mobile a 32 bit (in formato IEEE754) in notazione decimale standard o in notazione scientifica e lo converte in testo ASCII.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FSTR(448) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FSTR(448) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

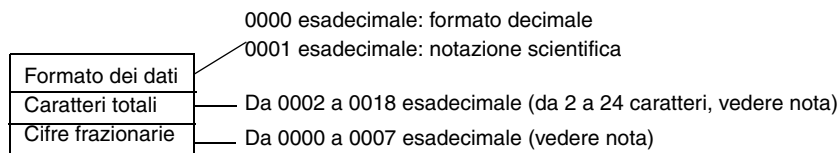
| Area | S | C | D |
|------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6141 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W510 | Da W000 a W509 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H510 | Da H000 a H509 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A958 | Da A000 ad A957 | Da A448 a A959 |

| Area | S | C | D |
|--|---|--|--|
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4093 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4093 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32765 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32765 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | En_00000 ... En_32765 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Da #00000000 a #FFFFFFFF (binario) | --- | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-()IR15 Da ,IR0 a ,IR15 | | |

Descrizione

FSTR(448) rappresenta il numero a virgola mobile a 32 bit in S+1 ed S (formato IEEE754) in notazione decimale o scientifica in base ai dati di controllo nei canali compresi tra C e C+2, converte il numero in testo ASCII e invia il risultato ai canali di destinazione iniziando da D.

Nel diagramma seguente è riportato il contenuto dei 3 canali di controllo.



Nota: il numero totale dei caratteri e il numero delle cifre frazionarie sono limitati. Per ulteriori dettagli vedere *Limiti al numero di caratteri ASCII* a pagina 607.

- Il contenuto di C (formato dei dati) determina la modalità di rappresentazione dei numeri in S+1, S: notazione decimale o scientifica.
 - Notazione decimale
Rappresenta un numero reale come parte intera e parte frazionaria.
Esempio: 124,56

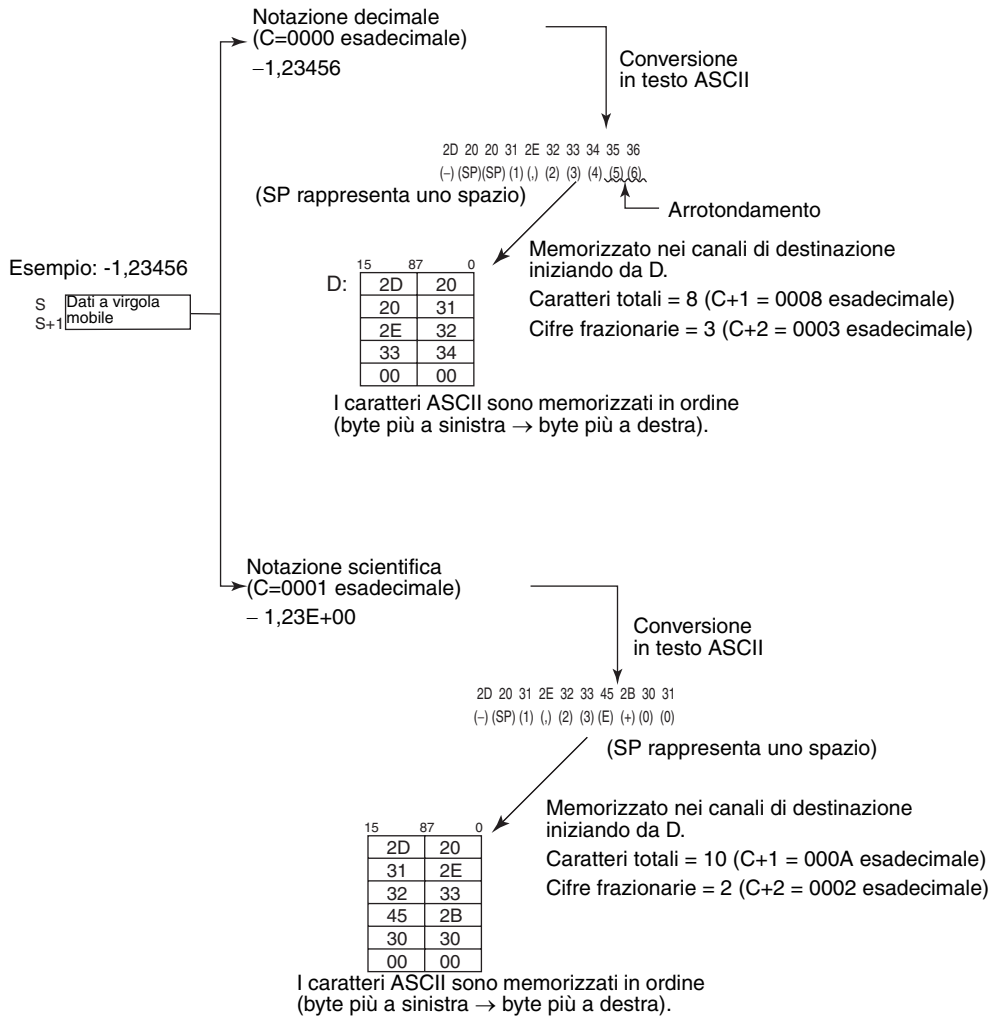
- Notazione scientifica

Rappresenta un numero reale come parte intera, parte frazionaria e parte esponenziale.

Esempio: 1,2456E-2 ($1,2456 \times 10^{-2}$)

- Il contenuto di C+1 (Caratteri totali) indica il numero di caratteri ASCII dopo la conversione, compresi il simbolo del segno, i numeri, la virgola decimale e gli spazi.
- Il contenuto di C+2 (Cifre frazionarie) indica il numero di cifre (caratteri) dopo la virgola decimale.

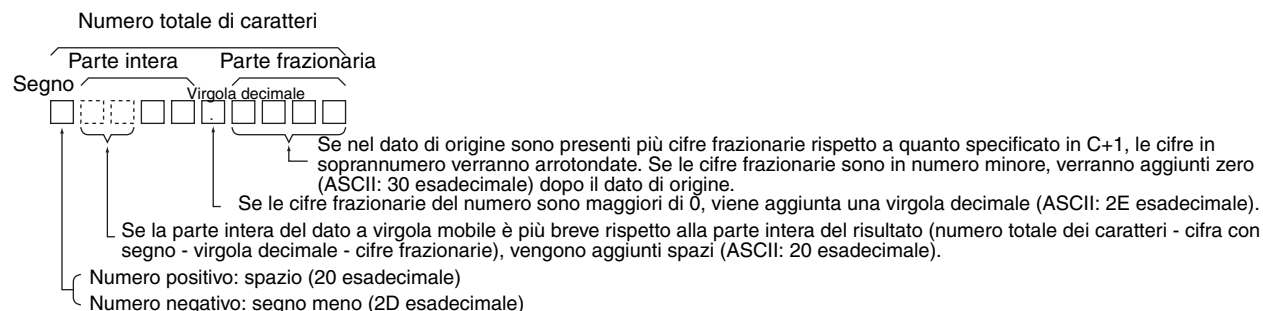
Il testo ASCII viene memorizzato in D e canali successivi nell'ordine seguente: byte più a sinistra di D, byte più a destra di D, byte più a sinistra di D+1, byte più a destra di D+1, ecc.



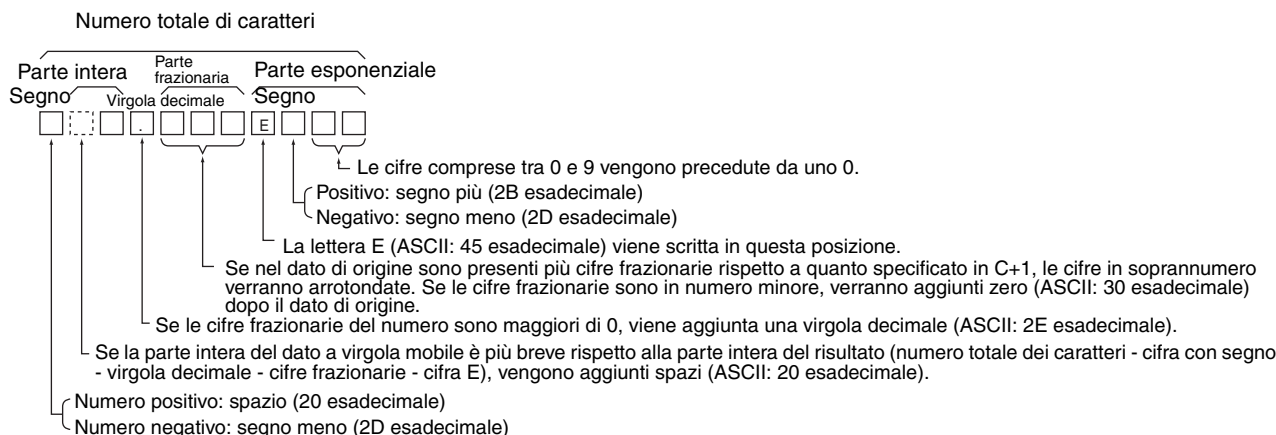
Memorizzazione di testo ASCII

Quando il numero a virgola mobile è stato convertito in testo ASCII, i caratteri ASCII vengono memorizzati nei canali di destinazione iniziando da D, come illustrato nei diagrammi seguenti. La notazione decimale e la notazione scientifica prevedono metodi differenti di memorizzazione.

Notazione decimale (C=0000 esadecimale)



Notazione scientifica (C=0001 esadecimale)



Nota Dopo il testo ASCII vengono aggiunti uno o due zeri come codice di fine.
 Numero totale di caratteri dispari: 00 esadecimale viene memorizzato dopo il testo ASCII.
 Numero totale di caratteri pari: 0000 esadecimale viene memorizzato dopo il testo ASCII.

Limiti al numero di caratteri ASCII

Il numero di caratteri ASCII nel numero convertito è limitato. Se il numero di caratteri supera il massimo consentito, viene attivato il flag di errore.

1. Limiti al numero totale di caratteri ASCII

a) Notazione decimale (C = 0000 esadecimale)

- In assenza di parte frazionaria (C+2 = 0000 esadecimale):
 $2 \leq \text{Caratteri totali} \leq 24$
- In presenza di parte frazionaria (C+2 = da 0001 a 0007 esadecimale):
 $(\text{Parte frazionaria} + 3) \leq \text{Caratteri totali} \leq 24$

b) Notazione scientifica (C = 0001 esadecimale)

- In assenza di parte frazionaria (C+2 = 0000 esadecimale):
 $6 \leq \text{Caratteri totali} \leq 24$
- In presenza di parte frazionaria (C+2 = da 0001 a 0007 esadecimale):
 $(\text{Parte frazionaria} + 7) \leq \text{Caratteri totali} \leq 24$

2. Limiti al numero di cifre nella parte intera
 - a) Notazione decimale (C = 0000 esadecimale)
 - In assenza di parte frazionaria (C+2 = 0000 esadecimale):
 $1 \leq \text{Numero di cifre intere} \leq 24$
 - In presenza di parte frazionaria (C+2 = da 0001 a 0007 esadecimale):
 $1 \leq \text{Numero di cifre intere} \leq (24 - \text{Cifre frazionarie} - 2)$
 - b) Notazione scientifica (C = 0001 esadecimale)
 1 cifra (fisso)
3. Limiti al numero di cifre nella parte frazionaria
 - a) Notazione decimale (C = 0000 esadecimale)
 - Cifre frazionarie ≤ 7
 - Inoltre: Cifre frazionarie $\leq (\text{Numero totale di caratteri ASCII} - 3)$
 - b) Notazione scientifica (C = 0001 esadecimale)
 - Cifre frazionarie ≤ 7
 - Inoltre: Cifre frazionarie $\leq (\text{Numero totale di caratteri ASCII} - 3)$

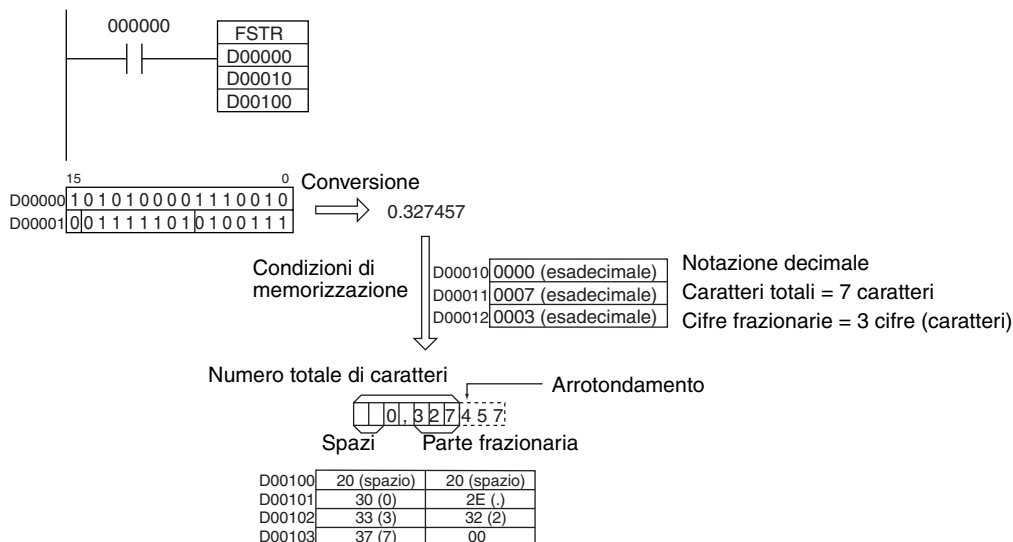
Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | <p>ON se il dato in S+1 ed S non è un numero a virgola mobile valido (NaN).</p> <p>ON se il dato in S+1 ed S è $+\infty$ o $-\infty$.</p> <p>ON se l'impostazione del formato dei dati in C non è 0000 o 0001.</p> <p>ON se l'impostazione dei caratteri totali in C+1 non rientra nell'intervallo consentito (per ulteriori informazioni, consultare 1. <i>Limiti al numero totale di caratteri ASCII</i> poco sopra).</p> <p>ON se l'impostazione delle cifre frazionarie in C+2 non rientra nell'intervallo consentito (per ulteriori informazioni, consultare 3. <i>Limiti al numero di cifre nella parte frazionaria</i> poco sopra).</p> <p>OFF in tutti gli altri casi.</p> |
| Flag di uguaglianza | = | <p>ON se il risultato della conversione è 0.</p> <p>OFF in tutti gli altri casi.</p> |

Esempi

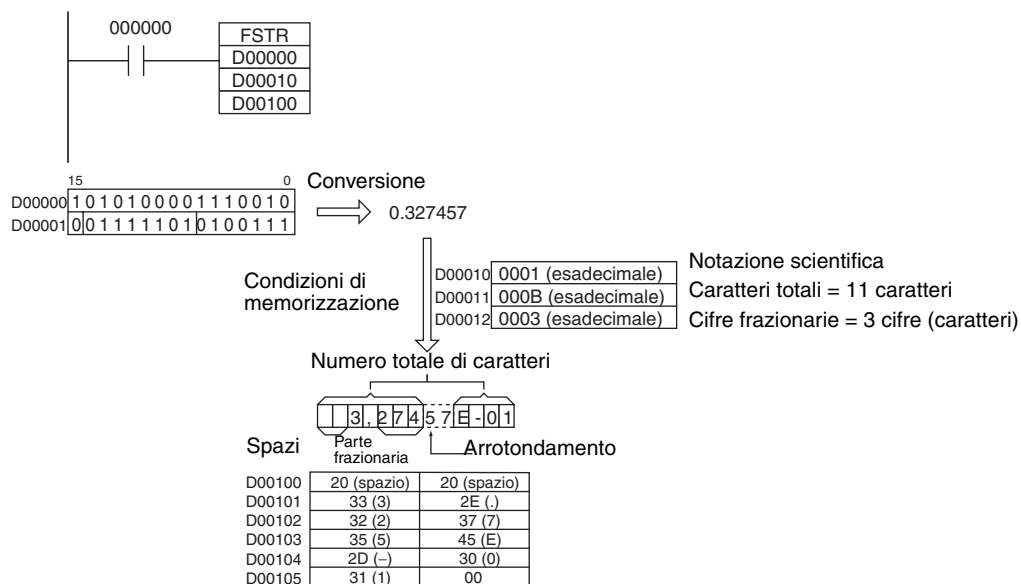
Conversione in testo ASCII con notazione decimale

Nell'esempio che segue, quando CIO 000000 è impostato su ON, FSTR(448) converte il dato a virgola mobile in D00001 e D00000 in testo ASCII con notazione decimale e lo scrive nei canali di destinazione iniziando da D00100. Il contenuto dei canali di controllo (da D00010 a D00012) specifica i dettagli sul formato dei dati: notazione decimale, 7 caratteri totali, 3 cifre frazionarie.



Conversione in testo ASCII con notazione scientifica

Nell'esempio che segue, quando CIO 000000 è impostato su ON, FSTR(448) converte il dato a virgola mobile in D00001 e D00000 in testo ASCII con notazione scientifica e lo scrive nei canali di destinazione iniziando da D00100. Il contenuto dei canali di controllo (da D00010 a D00012) specifica i dettagli sul formato dei dati: notazione scientifica, 11 caratteri totali, 3 cifre frazionarie.



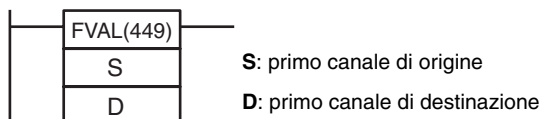
3-15-23 ASCII TO FLOATING-POINT: FVAL(449)

Scopo

Converte un numero rappresentato in testo ASCII (con notazione decimale o scientifica) in un valore a virgola mobile a 32 bit (formato IEEE754) e invia tale valore ai canali specificati.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FVAL(449) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FVAL(449) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

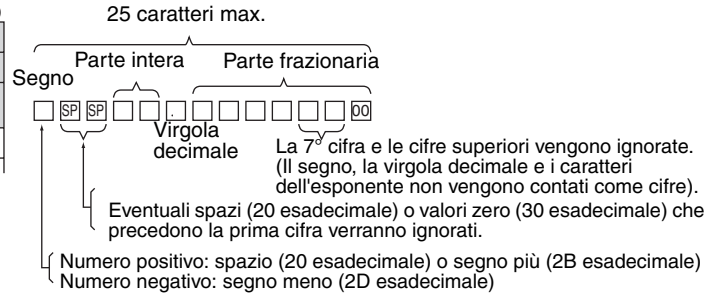
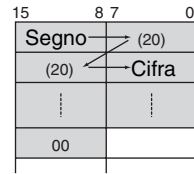
Caratteristiche operando

| Area | S | D |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 |
| Area bit ausiliaria | Da A000 a A959 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-()IR15 Da ,IR0 a ,IR15 | |

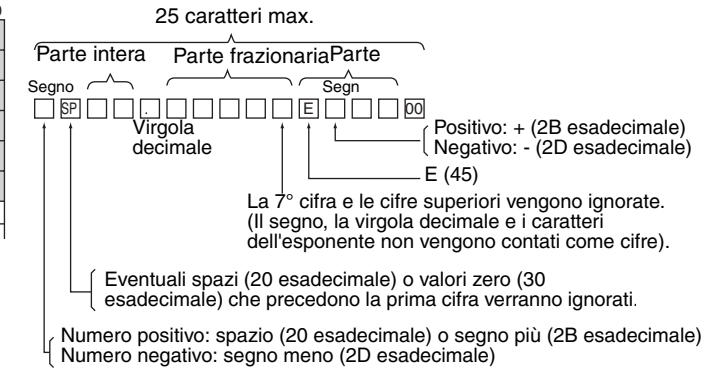
Descrizione

FVAL(449) converte il numero rappresentato in testo ASCII iniziando dal canale S in un valore a virgola mobile a 32 bit (formato IEEE754) e invia il risultato ai canali di destinazione iniziando da D.

Notazione decimale



Notazione scientifica



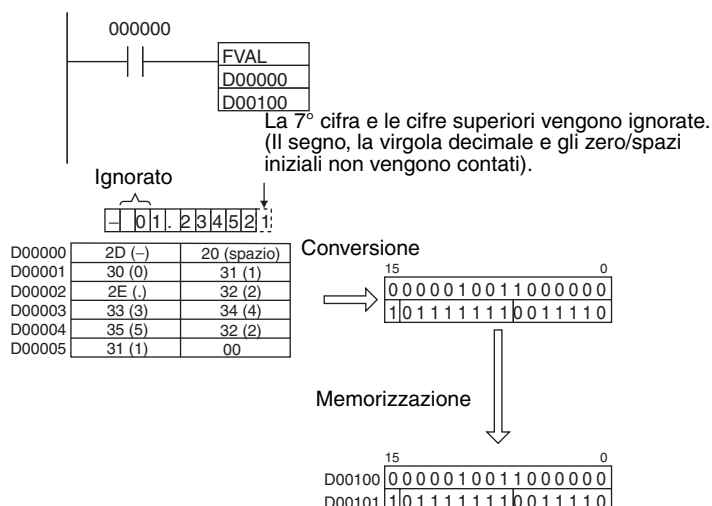
Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | <p>ON se le cifre (parti intera e frazionaria) nei dati di origine che cominciano da S non sono incluse tra 30 e 39 esadecimale (0 - 9).</p> <p>ON se le prime due cifre della parte esponenziale non contengono i valori 45 e 2B esadecimale (E+) o 45 e 2D esadecimale (E-) e se le cifre (parti intera e frazionaria) nei dati di origine che cominciano da S non sono incluse tra 30 e 39 esadecimale (0 - 9).</p> <p>ON se i dati di origine contengono due o più parti esponenziali.</p> <p>ON se i dati sono $+\infty$ o $-\infty$ dopo la conversione.</p> <p>ON se vi sono caratteri 0 nei dati di testo.</p> <p>ON se entro i primi 25 caratteri non viene trovato un byte contenente il valore 00 esadecimale.</p> <p>OFF in tutti gli altri casi.</p> |
| Flag di uguaglianza | = | <p>ON se il risultato della conversione è 0.</p> <p>OFF in tutti gli altri casi.</p> |

Esempi

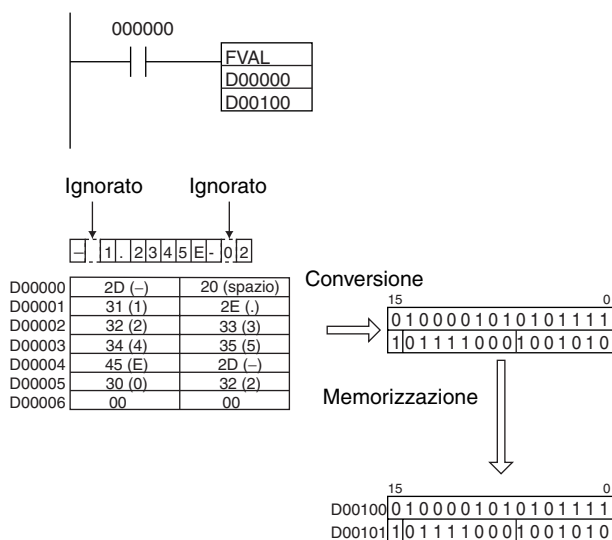
Conversione di testo ASCII con notazione decimale in dati a virgola mobile

Nell'esempio seguente, quando CIO 000000 è impostato su ON, FVAL(449) converte il numero di testo ASCII con notazione decimale specificato nei canali di origine cominciando da D00000 in dati a virgola mobile e scrive il risultato nei canali di destinazione D00100 e D00101.



Conversione di testo ASCII con notazione scientifica

Nell'esempio seguente, quando CIO 000000 è impostato su ON, FVAL(449) converte il numero di testo ASCII con notazione scientifica specificato nei canali di origine cominciando da D00000 in dati a virgola mobile e scrive il risultato nei canali di destinazione D00100 e D00101.



3-16 Istruzioni a virgola mobile in doppia precisione (solo CPU CS1-H, CJ1-H, CJ1M o CS1D)

Le istruzioni a virgola mobile in doppia precisione convertono i dati ed eseguono operazioni aritmetiche a virgola mobile su dati a virgola mobile in doppia precisione. Le CPU CS1-H/CJ1-H supportano le seguenti 20 istruzioni.

| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|--------------------------------|------------------|-----------------|--------|
| DOUBLE FLOATING TO 16-BIT | FIXD | 841 | 620 |
| DOUBLE FLOATING TO 32-BIT | FIXLD | 842 | 621 |
| 16-BIT TO DOUBLE FLOATING | DBL | 843 | 623 |
| 32-BIT TO DOUBLE FLOATING | DBLL | 844 | 624 |
| DOUBLE FLOATING-POINT ADD | +D | 845 | 626 |
| DOUBLE FLOATING-POINT SUBTRACT | -D | 846 | 628 |
| DOUBLE FLOATING-POINT MULTIPLY | *D | 847 | 630 |

| Istruzione | Codice mnemonico | Codice funzione | Pagina |
|---|--|-----------------|--------|
| DOUBLE FLOATING-POINT DIVIDE | /D | 848 | 632 |
| DOUBLE DEGREES TO RADIANS | RADD | 849 | 634 |
| DOUBLE RADIANS TO DEGREES | DEGD | 850 | 636 |
| DOUBLE SINE | SIND | 851 | 637 |
| DOUBLE COSINE | COSD | 852 | 639 |
| DOUBLE TANGENT | TAND | 853 | 641 |
| DOUBLE ARC SINE | ASIND | 854 | 643 |
| DOUBLE ARC COSINE | ACOSD | 855 | 645 |
| DOUBLE ARC TANGENT | ATAND | 856 | 647 |
| DOUBLE SQUARE ROOT | SQRTD | 857 | 649 |
| DOUBLE EXPONENT | EXPD | 858 | 651 |
| DOUBLE LOGARITHM | LOGD | 859 | 653 |
| DOUBLE EXPONENTIAL POWER | PWRD | 860 | 655 |
| Istruzioni di confronto con simboli a virgola mobile in doppia precisione | LD, AND, OR + =D, <>D, <D, <=D, >D, or >=D | Da 335 a 340 | 657 |

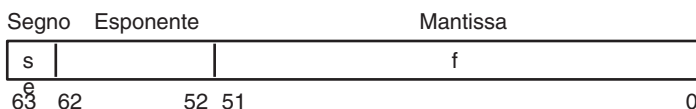
Formato dei dati

I dati a virgola mobile esprimono numeri reali tramite un segno, un esponente e la mantissa. Quando i dati vengono espressi nel formato a virgola mobile, viene applicata la formula seguente.

$$\text{Numero reale} = (-1)^s 2^{e-1.023} (1,f)$$

- s: Segno
- e: Esponente
- f: Mantissa

Il formato dei dati a virgola mobile è conforme agli standard IEEE754. I dati sono espressi in 32 bit, come mostrato di seguito:



| Dati | N. di bit | Contenuto |
|--------------|-----------|--|
| s: segno | 1 | 0: positivo; 1: negativo |
| e: esponente | 11 | Il valore dell'esponente (e) varia da 0 a 2.047. L'esponente effettivo è il valore rimanente dopo che 1.023 è stato sottratto da e, ottenendo come risultato un intervallo da -1.023 a 1.024. "e=0" e "e=2.047" esprimono numeri speciali. |
| f: mantissa | 52 | La parte della mantissa dei dati binari a virgola mobile corrisponde al formato 2,0 > 1,f ≥ 1,0. |

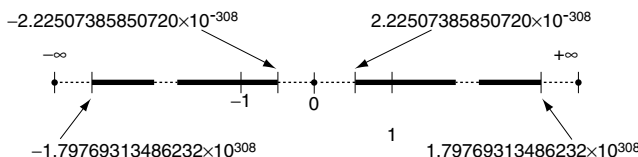
Numero di cifre

Il numero di cifre effettive per i dati a virgola mobile è 53 bit per il formato binario (circa 15 cifre decimali).

Dati a virgola mobile

I dati seguenti possono essere espressi da dati a virgola mobile:

- $-\infty$
- $-1,79769313486232 \times 10^{308} \leq \text{valore} \leq -2,22507385850720 \times 10^{-308}$
- 0
- $2,22507385850720 \times 10^{-308} \leq \text{valore} \leq 1,79769313486232 \times 10^{30}$
- $+\infty$
- Non un numero (Not a number: NaN)



Numeri speciali

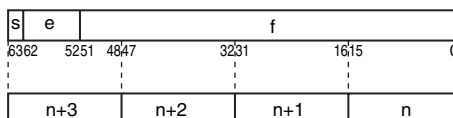
I formati per il tipo NaN, ±∞ e 0 sono i seguenti:

- NaN*: e = 1.024 e f ≠ 0
- +∞: e = 1.024, f = 0 e s = 0
- ∞: e = 1.024, f = 0 e s = 1
- 0: e = 0 e f = 0

Il tipo *NaN (not a number) non è un numero a virgola mobile valido. L'esecuzione di istruzioni a virgola mobile in doppia precisione non produce risultati di tipo NaN.

Scrittura di dati a virgola mobile

Quando viene specificata la virgola mobile in doppia precisione per il formato dei dati del display di modifica della memoria di I/O di CX-Programmer, i numeri decimali standard inseriti nel display vengono convertiti automaticamente nel formato a virgola mobile in doppia precisione illustrato in precedenza (formato IEEE754) e quindi scritti nella memoria di I/O. Una volta monitorati sul display, i dati scritti nel formato IEEE754 vengono convertiti automaticamente nel formato decimale standard.



Nella lettura e scrittura di dati a virgola mobile in doppia precisione, non è necessario conoscere il formato dei dati IEEE754, ma è sufficiente ricordare che i valori a virgola mobile in doppia precisione occupano quattro canali ciascuno.

Numeri espressi come valori a virgola mobile

È possibile utilizzare i seguenti tipi di numeri a virgola mobile.

| Mantissa (f) | Esponente (e) | | |
|--------------|-------------------------|-------------------------------|-------------------|
| | 0 | Non 0 e non tutto a 1 (1.024) | Tutto a 1 (1.024) |
| 0 | 0 | Numeri normalizzati | Infinito |
| Non 0 | Numero non normalizzato | | NaN |

Nota Un numero non è normalizzato quando il rispettivo valore assoluto è troppo piccolo per essere espresso come numero normalizzato. I numeri non normalizzati hanno meno cifre significative. Se il risultato dei calcoli è un numero non normalizzato (inclusi i risultati intermedi), il numero di cifre significative verrà ridotto.

Numeri normalizzati

I numeri normalizzati esprimono numeri reali. Il bit di segno sarà 0 per un numero positivo e 1 per un numero negativo.

L'esponente (e) verrà espresso da 1 a 2.046 e l'esponente reale sarà la sottrazione di 1.023, ovvero da -1.022 a 1.023.

La mantissa (f) sarà espressa da 0 a (2⁵² - 1) e si presume che, nella mantissa reale, il bit 2⁵² sia 1 e che la virgola decimale segua subito dopo.

I numeri normalizzati sono espressi nel modo seguente:

$$(-1)^{(\text{segno } s)} \times 2^{(\text{esponente } e) - 1,023} \times (1 + \text{mantissa} \times 2^{-52})$$

Overflow, underflow e calcoli non validi

Gli overflow verranno inviati come infinito positivo o negativo, in base al segno del risultato. Gli underflow verranno inviati come zero positivo o negativo, in base al segno del risultato.

I calcoli non validi producono risultati di tipo NaN. I calcoli non validi includono l'addizione dell'infinito con un numero di segno opposto, la sottrazione dell'infinito da un numero di segno opposto, la moltiplicazione di zero e infinito, la divisione di zero per zero o la divisione di infinito per infinito.

Se nella conversione di un numero a virgola mobile in un numero intero si verifica un overflow, il valore del risultato potrebbe non essere corretto.

Avvertenze per l'uso di valori speciali

Quando si usano i valori zero, infinito e NaN, è necessario seguire le avvertenze fornite di seguito.

- La somma di zero positivo e zero negativo dà come risultato zero positivo.
- La differenza tra zeri dello stesso segno dà come risultato zero positivo.
- Se uno qualsiasi degli operandi è un NaN, il risultato sarà di tipo NaN.
- Lo zero positivo e lo zero negativo vengono considerati equivalenti nei confronti.
- I test di confronto e equivalenza su uno o più NaN saranno sempre veri per != e sempre falsi per tutte le altre istruzioni.

Risultati dei calcoli a virgola mobile in doppia precisione

Quando il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$. Se il risultato è positivo, verrà inviato come $+\infty$, mentre se è negativo, come $-\infty$.

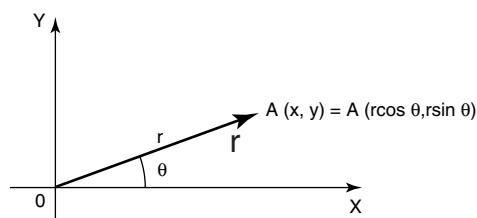
Il flag di uguaglianza verrà attivato soltanto quando sia l'esponente (e) che la mantissa (f) risultano zero dopo un calcolo. Un risultato di calcolo verrà inoltre inviato come zero quando il valore assoluto del risultato è inferiore al valore minimo che può essere espresso come dato a virgola mobile. In questo caso verrà attivato il flag di underflow.

Confronto di calcoli in singola precisione e in doppia precisione

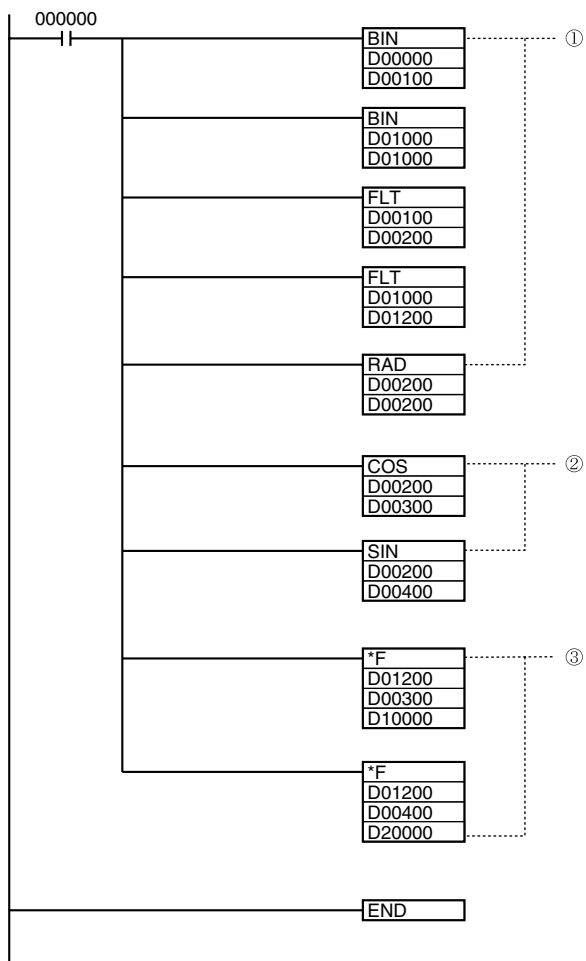
Nell'esempio che segue vengono mostrate le differenze tra i calcoli in singola precisione e in doppia precisione quando il seguente vettore espresso in coordinate polari viene convertito in coordinate rettangolari A (x,y).

$$r = re^{j \left(\frac{\pi}{360} \right) \theta}$$

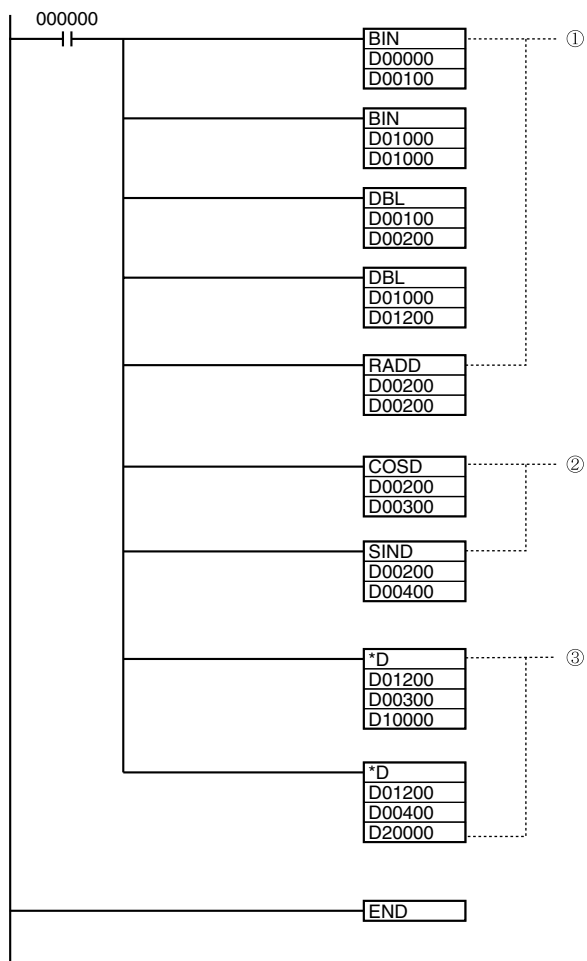
In questo esempio l'angolo BCD a 4 cifre (θ , in gradi) viene letto da D00000 e la distanza BCD a 4 cifre (r) viene letta da D01000.



- Programma ladder per calcolo in singola precisione



- Programma ladder per calcolo in doppia precisione



1. Questa sezione del programma converte i dati BCD in dati a virgola mobile in singola precisione (32 bit, formato IEEE754).
 - a) Le istruzioni BIN(023) convertono i dati BCD in dati binari, mentre le istruzioni FLT(452) convertono i dati binari in dati a virgola mobile in singola precisione.
 - b) I dati a virgola mobile per l'angolo θ vengono inviati a D00200 e a D00201.
 - c) RAD(458) converte in radianti i dati angolari in D00200 e in D00201.
 - d) I dati a virgola mobile per il raggio r vengono inviati a D01200 e a D01201.
2. Questa sezione del programma calcola il seno θ e il coseno θ come valori a virgola mobile in singola precisione.
 - a) Il valore del coseno θ viene inviato a D00300 e a D00301.
 - b) Il valore del seno θ viene inviato a D00400 e a D00401.
3. Questa sezione del programma calcola x ($r \times \coseno \theta$) e y ($r \times \seno \theta$).
 - a) Il valore di x ($r \times \coseno \theta$) viene inviato a D10000 e a D10001.
 - b) Il valore di y ($r \times \seno \theta$) viene inviato a D20000 e a D20001.

| Coordinate | Numero a virgola mobile | Numero reale |
|------------|-------------------------|-----------------|
| x | 4116 59CF | 3,4202015399933 |
| y | 405A E495 | 9,3969259262085 |

1. Questa sezione del programma converte i dati BCD in dati a virgola mobile in doppia precisione (64 bit, formato IEEE754).
 - a) Le istruzioni BIN(023) convertono i dati BCD in dati binari, mentre le istruzioni DBL(843) convertono i dati binari in dati a virgola mobile in doppia precisione.
 - b) I dati a virgola mobile per l'angolo θ vengono inviati ai canali da D00200 a D00203.
 - c) RADD(849) converte in radianti i dati angolari nei canali D00200 e D00203.
 - d) I dati a virgola mobile per il raggio r vengono inviati ai canali da D01200 a D01203.
2. Questa sezione del programma calcola il seno θ e il coseno θ come valori a virgola mobile in doppia precisione.
 - a) Il valore del coseno θ viene inviato ai canali da D00300 a D00303.
 - b) Il valore del seno θ viene inviato ai canali D00400 e D00403.
3. Questa sezione del programma calcola x ($r \times \coseno \theta$) e y ($r \times \seno \theta$).
 - a) Il valore di x ($r \times \coseno \theta$) viene inviato ai canali da D10000 a D10003.
 - b) Il valore di y ($r \times \seno \theta$) viene inviato a D20000 e a D20003.

| Coordinate | Numero a virgola mobile | Numero reale |
|------------|-------------------------|-----------------|
| x | 4022 CB39 E973 5C32 | 3,4202014332567 |
| y | 400B 5C92 91AC 8EEB | 9,3969262078591 |

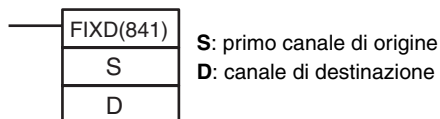
Confronto dei risultati del calcolo

Quando si confrontano i risultati dei numeri reali, è chiaro che il calcolo in doppia precisione produce un risultato più accurato.

3-16-1 DOUBLE FLOATING TO 16-BIT: FIXD(841)

Scopo Convertire un valore a virgola mobile in doppia precisione (64 bit) in dati binari a 16 bit con segno e invia il risultato al canale del risultato specificato.
 Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FIXD(841) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FIXD(841) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

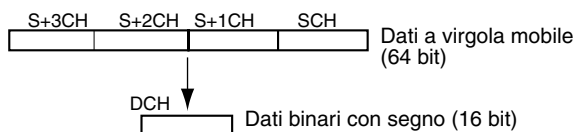
| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6140 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W508 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H508 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A956 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4092 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4092 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32764 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32764 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

FIXD(841) converte la parte intera del numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 (formato IEEE754) in dati binari con segno a 16 bit e inserisce il risultato in D.



Viene convertita solo la parte intera del dato a virgola mobile; la parte frazionaria viene troncata. La parte intera dei dati a virgola mobile deve essere compresa nell'intervallo da -32.768 a 32.767 .

Conversioni di esempio

Un valore a virgola mobile di 3,5 viene convertito in 3.

Un valore a virgola mobile di $-3,5$ viene convertito in -3 .

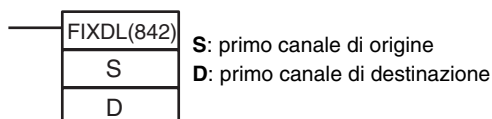
Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine (da S a S+3) sono di tipo NaN. ON se la parte intera dei dati di origine (da S a S+3) non è compresa nell'intervallo da -32.768 a 32.767 . OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 del risultato è ON. OFF in tutti gli altri casi. |

3-16-2 DOUBLE FLOATING TO 32-BIT: FIXLD(842)**Scopo**

Converte un valore a virgola mobile in doppia precisione (64 bit) in dati binari a 32 bit con segno e invia il risultato ai canali del risultato specificato.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | FIXLD(842) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FIXLD(842) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

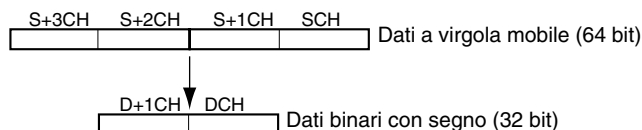
Caratteristiche operando

| Area | S | D |
|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6140 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W508 | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H508 | Da H000 a H510 |

| Area | S | D |
|--|--|--|
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4092 | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4092 | Da C0000 a C4094 |
| Area DM | Da D00000 a D32764 | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32764 | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

FIXLD(842) converte la parte intera del numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 (formato IEEE754) in dati binari con segno a 32 bit e inserisce il risultato in D.



Viene convertita solo la parte intera del dato a virgola mobile; la parte frazionaria viene troncata. La parte intera dei dati a virgola mobile deve essere compresa nell'intervallo da -2.147.483.648 a 2.147.483.647.

Conversioni di esempio

Un valore a virgola mobile di 2.147.483.640,5 viene convertito in 2.147.483.640.

Un valore a virgola mobile di -2.147.483.640,5 viene convertito in -2.147.483.640.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati nei canali da S a S+3 sono di tipo NaN. ON se la parte intera dei dati nei canali da S a S+3 non è compresa nell'intervallo da -2.147.483.648 a 2.147.483.647. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 di D+1 è ON dopo l'esecuzione. OFF in tutti gli altri casi. |

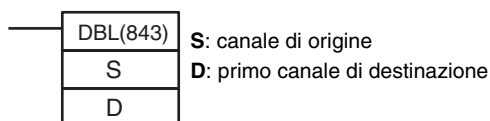
Avvertenze

I canali da S a S+3 devono contenere dati in virgola mobile e la parte intera deve essere compresa nell'intervallo da -2.147.483.648 a 2.147.483.647.

3-16-3 16-BIT TO DOUBLE FLOATING: DBL(843)**Scopo**

Converte un valore binario a 16 bit con segno in dati a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali di destinazione specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | DBL(843) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DBL(843) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

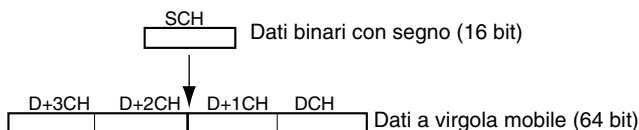
Caratteristiche operando

| Area | S | D |
|--------------------------------------|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W511 | Da W000 a W508 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H508 |
| Area bit ausiliaria | Da A000 a A959 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4092 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #FFFF (binario) | --- |
| Registri dati | Da DR0 a DR15 | --- |

| Area | S | D |
|--|--|---|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

DBL(843) converte il valore binario a 16 bit con segno in S in dati a virgola mobile in doppia precisione (64 bit) (formato IEEE754) e invia il risultato ai canali da D a D+3. Nel risultato a virgola mobile viene aggiunto un unico 0 dopo la virgola decimale.



Per S è possibile specificare solo valori compresi nell'intervallo da -32.768 a 32.767. Per convertire i dati binari con segno al di fuori di tale intervallo, utilizzare l'istruzione DBLL(844).

Conversioni di esempio

Un valore binario con segno di 3 viene convertito in 3,0.

Un valore binario con segno di -3 viene convertito in -3,0.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

Il contenuto di S deve includere dati binari con segno aventi un valore (decimale) compreso tra -32.768 e 32.767.

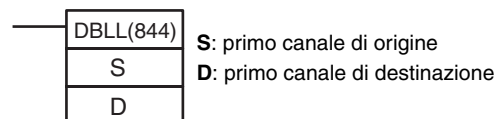
3-16-4 32-BIT TO DOUBLE FLOATING: DBLL(844)

Scopo

Converte un valore binario a 32 bit con segno in dati a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali di destinazione specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | DBLL(844) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DBLL(844) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

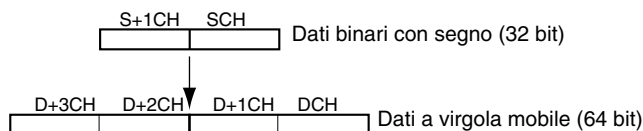
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W510 | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H508 |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4092 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #00000000 a #FFFFFFF (binario) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

DBLL(844) converte il valore binario a 32 bit con segno in S+1 e S in dati a virgola mobile in doppia precisione (64 bit) (formato IEEE754) e invia il risultato ai canali da D a D+3. Nel risultato a virgola mobile viene aggiunto un unico 0 dopo la virgola decimale.



Per S+1 ed S è possibile specificare dati binari con segno compresi tra -2.147.483.648 e 2.147.483.647. Il valore a virgola mobile ha 24 cifre binarie significative (bit). Il risultato non sarà esatto se viene convertito da DBLL(844) un numero maggiore di 16.777.215 (il valore massimo che può essere espresso in 24 bit).

Conversioni di esempio

Un valore binario con segno di 16.777.215 viene convertito in 16.777.215,0.
Un valore binario con segno di -16.777.215 viene convertito in -15.777.215,0.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | OFF |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

Il risultato non sarà esatto se viene convertito un numero con un valore assoluto maggiore di 16.777.215 (il valore massimo che può essere espresso in 24 bit).

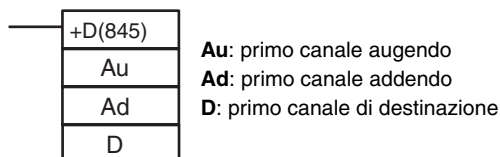
3-16-5 DOUBLE FLOATING-POINT ADD: +D(845)

Scopo

Somma due numeri a virgola mobile in doppia precisione a 64 bit e invia il risultato ai canali di destinazione specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | +D(845) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+D(845) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

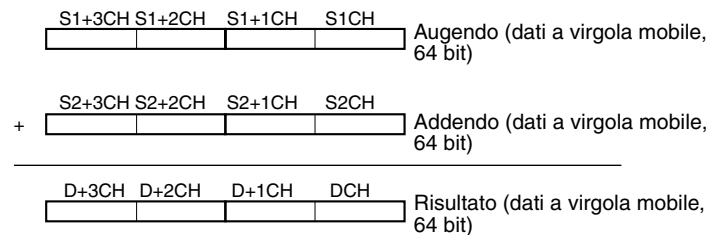
Caratteristiche operando

| Area | Au | Ad | D |
|--------------------------------------|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | | |
| Area di lavoro | Da W000 a W508 | | |
| Area bit di ritentività | Da H000 a H508 | | |
| Area bit ausiliaria | Da A000 a A956 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | | |
| Area del contatore | Da C0000 a C4092 | | |
| Area DM | Da D00000 a D32764 | | |
| Area EM senza banco | Da E00000 a E32764 | | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |

| Area | Au | Ad | D |
|--|--|----|---|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(- -)IR0 a ,-(- -)IR15 | | |

Descrizione

+D(845) aggiunge il numero a virgola mobile in doppia precisione (64 bit) nei canali da Ad a Ad+3 al numero a virgola mobile in doppia precisione (64 bit) nei canali Au - Au+3 e invia il risultato ai canali D - D+3 (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni per dati augendo e addendo producono i risultati indicati nella tabella seguente.

| Addendo | Augendo | | | | NaN |
|-----------|-----------|----------------|----------------|----------------|----------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | 0 | Numerico | $+\infty$ | $-\infty$ | Vedere nota 2. |
| Numerico | Numerico | Vedere nota 1. | $+\infty$ | $-\infty$ | |
| $+\infty$ | $+\infty$ | $+\infty$ | $+\infty$ | Vedere nota 2. | |
| $-\infty$ | $-\infty$ | $-\infty$ | Vedere nota 2. | $-\infty$ | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Verrà attivato il flag di errore e l'istruzione non verrà eseguita.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati augendo o addendo non vengono riconosciuti come dati a virgola mobile. ON se i dati augendo o addendo sono di tipo NaN. ON se $+\infty$ viene aggiunto a $-\infty$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati augendo (da Au a Au+3) e addendo (da Ad a Ad+3) siano nel formato dati a virgola mobile IEEE754.

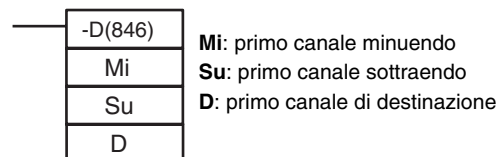
3-16-6 DOUBLE FLOATING-POINT SUBTRACT: -D(846)

Scopo

Esegue la sottrazione tra due numeri a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali di destinazione specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | -D(846) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @-D(846) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

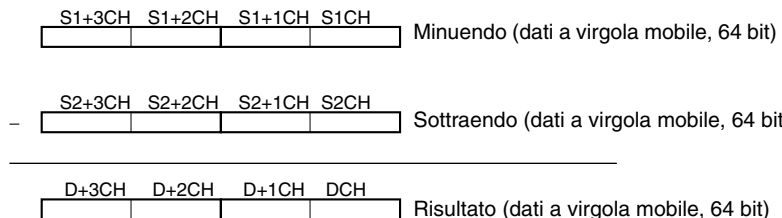
Caratteristiche operando

| Area | Mi | Su | D |
|-------------------------|------------------------|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | | |
| Area di lavoro | Da W000 a W508 | | |
| Area bit di ritentività | Da H000 a H508 | | |
| Area bit ausiliaria | Da A000 a A956 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | | |
| Area del contatore | Da C0000 a C4092 | | |
| Area DM | Da D00000 a D32764 | | |

| Area | Mi | Su | D |
|--|--|----|---|
| Area EM senza banco | Da E00000 a E32764 | | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 | | |

Descrizione

–D(846) sottrae il numero a virgola mobile in doppia precisione (64 bit) nei canali da Su a Su+3 dal numero a virgola mobile in doppia precisione (64 bit) nei canali Mi - Mi+3 e invia il risultato ai canali D - D+3 (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni di dati minuendo e sottraendo producono i risultati indicati nella tabella seguente.

| Sottraendo | Minuendo | | | | NaN |
|------------|-----------|----------------|----------------|----------------|----------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | 0 | Numerico | $+\infty$ | $-\infty$ | Vedere nota 2. |
| Numerico | Numerico | Vedere nota 1. | $+\infty$ | $-\infty$ | |
| $+\infty$ | $-\infty$ | $-\infty$ | Vedere nota 2. | $-\infty$ | |
| $-\infty$ | $+\infty$ | $+\infty$ | $+\infty$ | Vedere nota 2. | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Verrà attivato il flag di errore e l'istruzione non verrà eseguita.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati minuendo o sottraendo non vengono riconosciuti come dati a virgola mobile. ON se i dati minuendo o sottraendo sono di tipo NaN. ON se $+\infty$ viene sottratto da $+\infty$. ON se $-\infty$ viene sottratto da $-\infty$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati minuendo (da Mi a Mi+3) e sottraendo (da Su a Su+3) siano nel formato dati a virgola mobile IEEE754.

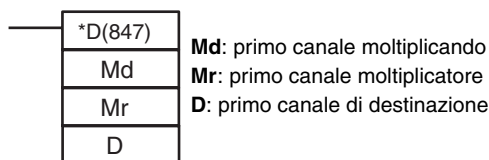
3-16-7 DOUBLE FLOATING-POINT MULTIPLY: *D(847)

Scopo

Moltiplica due numeri a virgola mobile in doppia precisione a 64 bit e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | *D(847) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @*D(847) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

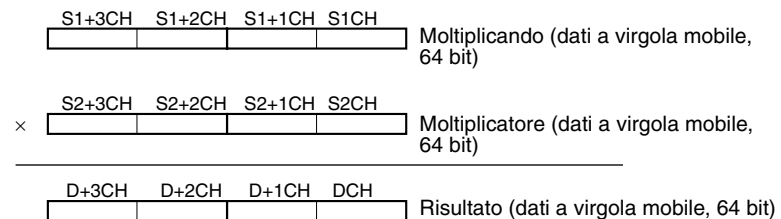
Caratteristiche operando

| Area | Md | Mr | D |
|-------------------------|------------------------|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | | |
| Area di lavoro | Da W000 a W508 | | |
| Area bit di ritentività | Da H000 a H508 | | |
| Area bit ausiliaria | Da A000 a A956 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | | |
| Area del contatore | Da C0000 a C4092 | | |
| Area DM | Da D00000 a D32764 | | |

| Area | Md | Mr | D |
|--|--|----|---|
| Area EM senza banco | Da E00000 a E32764 | | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(- -)IR0 a ,-(- -)IR15 | | |

Descrizione

*D(847) moltiplica il numero a virgola mobile in doppia precisione (64 bit) nei canali da Md a Md+3 per il numero a virgola mobile in doppia precisione (64 bit) nei canali Mr - Mr+3 e invia il risultato ai canali D - D+3 (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni dei dati moltiplicando e moltiplicatore producono i risultati indicati nella tabella seguente.

| Moltiplicatore | Moltiplicando | | | | NaN |
|----------------|----------------|----------------|----------------|----------------|----------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | 0 | 0 | Vedere nota 2. | Vedere nota 2. | Vedere nota 2. |
| Numerico | 0 | Vedere nota 1. | $+/-\infty$ | $+/-\infty$ | |
| $+\infty$ | Vedere nota 2. | $+/-\infty$ | $+\infty$ | $-\infty$ | |
| $-\infty$ | Vedere nota 2. | $+/-\infty$ | $-\infty$ | $+\infty$ | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Verrà attivato il flag di errore e l'istruzione non verrà eseguita.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati moltiplicando o moltiplicatore non vengono riconosciuti come dati a virgola mobile. ON se i dati moltiplicando o moltiplicatore sono di tipo NaN. ON se vengono moltiplicati $+\infty$ e 0. ON se vengono moltiplicati $-\infty$ e 0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati moltiplicando (da Md a Md+3) e moltiplicatore (da Mr a Mr+3) siano nel formato dati a virgola mobile IEEE754.

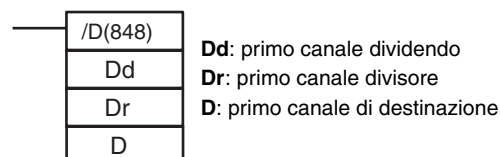
3-16-8 DOUBLE FLOATING-POINT DIVIDE: /D(848)

Scopo

Esegue la divisione di un numero a virgola mobile in doppia precisione (64 bit) per un altro e invia il risultato ai canali di destinazione specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | /D(848) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @/D(848) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

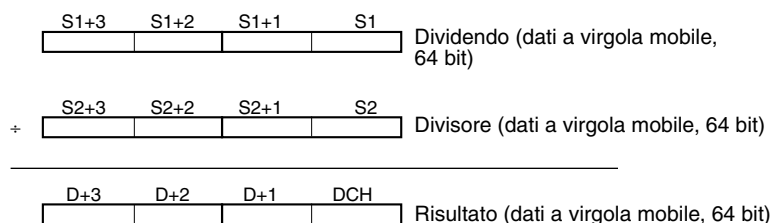
Caratteristiche operando

| Area | Dd | Dr | D |
|-------------------------|------------------------|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | | |
| Area di lavoro | Da W000 a W508 | | |
| Area bit di ritentività | Da H000 a H508 | | |
| Area bit ausiliaria | Da A000 a A956 | | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | | |
| Area del contatore | Da C0000 a C4092 | | |

| Area | Dd | Dr | D |
|--|--|----|---|
| Area DM | Da D00000 a D32764 | | |
| Area EM senza banco | Da E00000 a E32764 | | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 | | |

Descrizione

/D(848) divide il numero a virgola mobile in doppia precisione (64 bit) nei canali da Dd a Dd+3 per il numero a virgola mobile in doppia precisione (64 bit) nei canali Dr - Dr+3 e invia il risultato ai canali D - D+3 (i dati a virgola mobile devono essere nel formato IEEE754).



Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Le diverse combinazioni dei dati dividendo e divisore producono i risultati indicati nella tabella seguente.

| Divisore | Dividendo | | | | NaN |
|-----------|----------------|----------------|----------------|----------------|----------------|
| | 0 | Numerico | $+\infty$ | $-\infty$ | |
| 0 | Vedere nota 3. | $+/-\infty$ | $+\infty$ | $-\infty$ | Vedere nota 3. |
| Numerico | 0 | Vedere nota 1. | $+/-\infty$ | $+/-\infty$ | |
| $+\infty$ | 0 | Vedere nota 2. | Vedere nota 3. | Vedere nota 3. | |
| $-\infty$ | 0 | Vedere nota 2. | Vedere nota 3. | Vedere nota 3. | |
| NaN | | | | | |

- Nota**
1. I risultati possono essere zero (compresi gli underflow), un numero, $+\infty$ o $-\infty$.
 2. Per gli underflow i risultati saranno zero.
 3. Verrà attivato il flag di errore e l'istruzione non verrà eseguita.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati dividendo o divisore non vengono riconosciuti come dati a virgola mobile. ON se i dati dividendo o divisore sono di tipo NaN. ON se il dividendo e il divisore sono entrambi 0. ON se il dividendo e il divisore sono entrambi $+\infty$ o $-\infty$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati dividendo (da Dd1 a Dd+3) e divisore (da Dr a Dr+3) siano in formato dati a virgola mobile IEEE754.

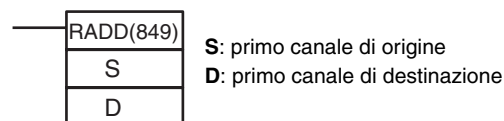
3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849)

Scopo

Converte un numero a virgola mobile in doppia precisione (64 bit) da gradi a radianti e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | RADD(849) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RADD(849) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

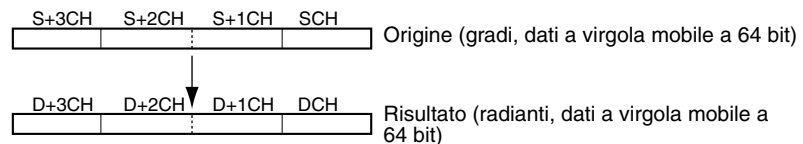
Caratteristiche operando

| Area | S | D |
|-------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritentività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |

| Area | S | D |
|--|--|---|
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

RADD(849) converte il numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 da gradi a radianti invia il risultato ai canali D - D+3 (i dati a virgola mobile devono essere nel formato IEEE754).



Per convertire i gradi in radianti, utilizzare la formula seguente:

$$\text{Gradi} \times \pi/180 = \text{radianti}$$

Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

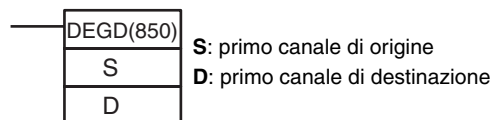
3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850)

Scopo

Converte un numero a virgola mobile in doppia precisione (64 bit) da radianti a gradi e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | DEGD(850) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DEGD(850) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

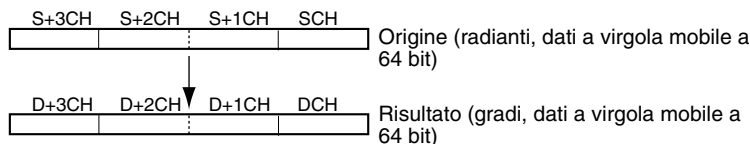
Caratteristiche operando

| Area | S | D |
|--------------------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritentività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |

| Area | S | D |
|--|--|---|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

DEGD(850) converte il numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 da radianti a gradi e invia il risultato ai canali D - D+3 (i dati a virgola mobile devono essere nel formato IEEE754).



Per convertire i radianti in gradi, utilizzare la formula seguente:

$$\text{Radianti} \times \pi/180 = \text{gradi}$$

Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

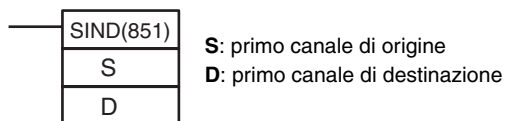
È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

3-16-11 DOUBLE SINE: SIND(851)**Scopo**

Calcola il seno di un numero a virgola mobile in doppia precisione (64 bit) (in radianti) e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SIND(851) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SIND(851) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

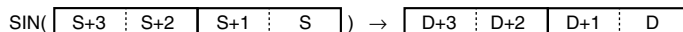
| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

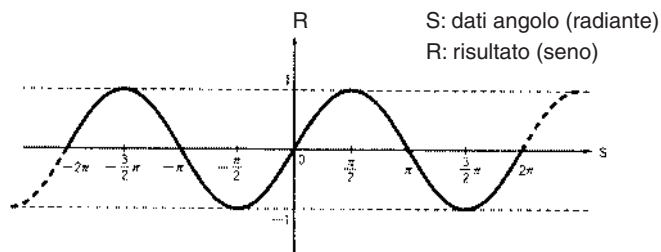
Descrizione

SIND(851) calcola il seno dell'angolo (in radianti) espresso come valore a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3.
(I dati a virgola mobile devono essere nel formato IEEE754).



Specifica l'angolo desiderato (da -65.535 a 65.535) in radianti nei canali da S a S+3. Se l'angolo non è incluso nell'intervallo da -65.535 a 65.535 , verrà generato un errore e l'istruzione non verrà eseguita. Per informazioni sulla conversione tra gradi e radianti, vedere 3-16-9 *DOUBLE DEGREES TO RADIANS: RADD(849)* o 3-16-10 *DOUBLE RADIANS TO DEGREES: DEGD(850)*.

Lo schema seguente illustra la relazione tra l'angolo e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine supera 65.535. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | Non modificato |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

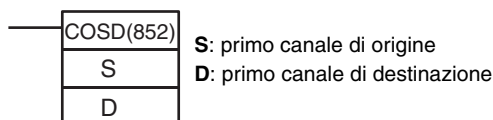
3-16-12 DOUBLE COSINE COSD(852)

Scopo

Calcola il coseno di un numero a virgola mobile in doppia precisione (64 bit) (in radianti) e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | COSD(852) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @COSD(852) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritentività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

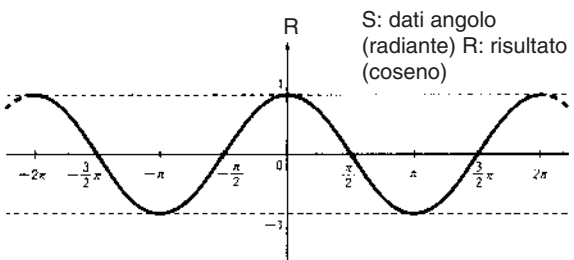
COSD(852) calcola il coseno dell'angolo (in radianti) espresso come valore a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3.

(I dati a virgola mobile devono essere nel formato IEEE754).

$$\text{COS}(\boxed{S+3} \boxed{S+2} \boxed{S+1} \boxed{S}) \rightarrow \boxed{D+3} \boxed{D+2} \boxed{D+1} \boxed{D}$$

Specifica l'angolo desiderato (da -65.535 a 65.535) in radianti nei canali da S a S+3. Se l'angolo non è incluso nell'intervallo da -65.535 a 65.535, verrà generato un errore e l'istruzione non verrà eseguita. Per informazioni sulla conversione tra gradi e radianti, vedere 3-16-9 *DOUBLE DEGREES TO RADIANS: RADD(849)* o 3-16-10 *DOUBLE RADIANS TO DEGREES: DEGD(850)*.

Lo schema seguente illustra la relazione tra l'angolo e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine supera 65.535. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | Non modificato |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

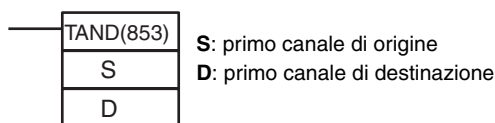
3-16-13 DOUBLE TANGENT TAND(853)

Scopo

Calcola la tangente di un numero a virgola mobile in doppia precisione (64 bit) (in radianti) e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TAND(853) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TAND(853) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|-------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |

| Area | S | D |
|--|--|---|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

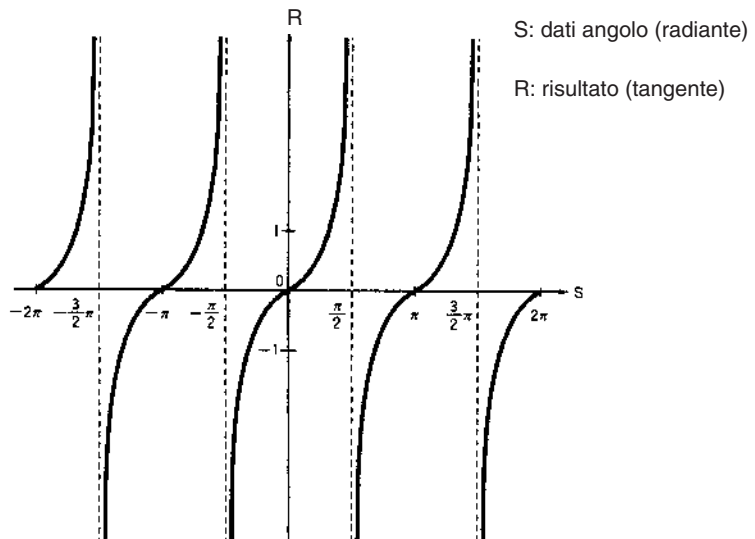
TAND(853) calcola la tangente dell'angolo (in radianti) espressa come valore a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3.
(I dati a virgola mobile devono essere nel formato IEEE754).

$$\text{TAN}(\boxed{S+3 \quad S+2 \quad S+1 \quad S}) \rightarrow \boxed{D+3 \quad D+2 \quad D+1 \quad D}$$

Specifica l'angolo desiderato (da -65.535 a 65.535) in radianti nei canali da S a S+3. Se l'angolo non è incluso nell'intervallo da -65.535 a 65.535, verrà generato un errore e l'istruzione non verrà eseguita. Per informazioni sulla conversione tra gradi e radianti, vedere 3-16-9 *DOUBLE DEGREES TO RADIANS: RADD(849)* o 3-16-10 *DOUBLE RADIANS TO DEGREES: DEGD(850)*.

Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come ±∞.

Lo schema seguente illustra la relazione tra l'angolo e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine supera 65.535. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione (64 bit). |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

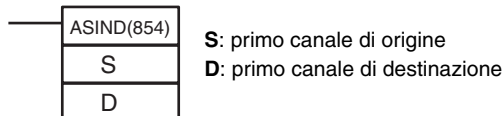
3-16-14 DOUBLE ARC SINE: ASIND(854)

Scopo

Calcola l'arcoseno di un numero a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali del risultato specificati. La funzione di arcoseno è l'inverso della funzione di seno e restituisce l'angolo che produce un dato valore di seno compreso tra -1 e 1.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ASIND(854) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ASIND(854) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|-------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |

| Area | S | D |
|--|--|---|
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

ASIND(854) calcola l'angolo (in radianti) per un valore di seno espresso come numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3.

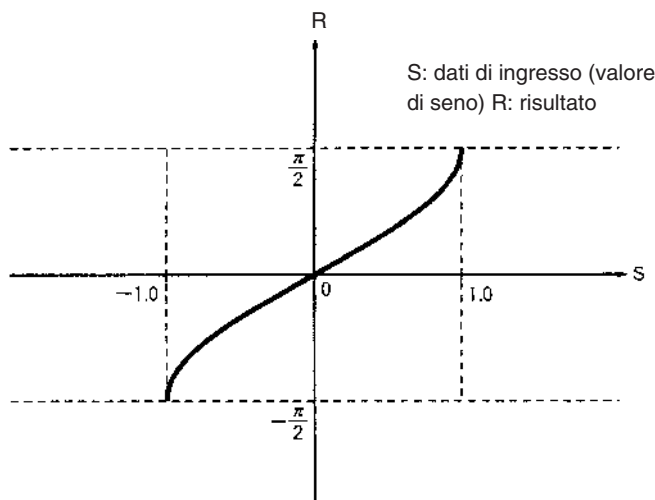
(I dati a virgola mobile devono essere nel formato IEEE754).

$$\text{SIN}^{-1}(\boxed{\text{S+3} \quad \text{S+2} \quad \text{S+1} \quad \text{S}}) \rightarrow \boxed{\text{D+3} \quad \text{D+2} \quad \text{D+1} \quad \text{D}}$$

Il dato di origine deve essere compreso tra -1,0 e 1,0. Se il valore assoluto dei dati di origine supera 1,0, verrà generato un errore e l'istruzione non verrà eseguita.

Il risultato viene inviato ai canali da D a D+3 come angolo (in radianti) compreso nell'intervallo da $-\pi/2$ a $\pi/2$.

Lo schema seguente illustra la relazione tra i dati di ingresso e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine supera 1,0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | Non modificato |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

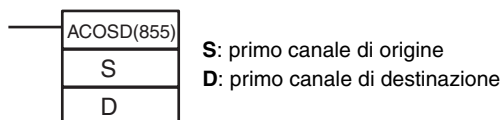
3-16-15 DOUBLE ARC COSINE: ACOSD(855)

Scopo

Calcola l'arcocoseno di un numero a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali del risultato specificati. La funzione di arcocoseno è l'inverso della funzione di coseno e restituisce l'angolo che produce un dato valore di coseno compreso tra -1 e 1.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ACOSD(855) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ACOSD(855) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|-------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |

| Area | S | D |
|--|--|---|
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

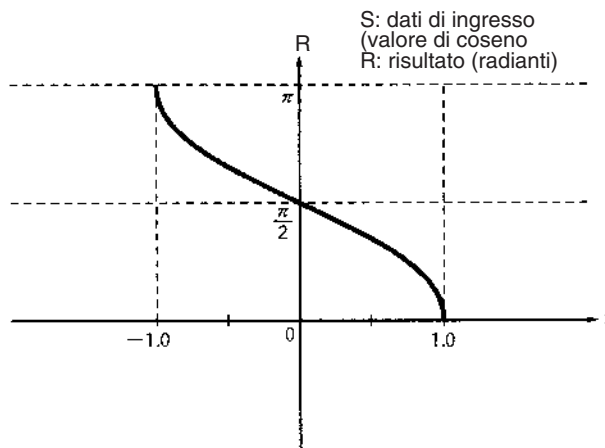
ACOSD(855) calcola l'angolo (in radianti) per un valore di coseno espresso come numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3. (I dati a virgola mobile devono essere nel formato IEEE754).

$$\text{COS}^{-1}(\boxed{S+3 \quad S+2 \quad S+1 \quad S}) \rightarrow \boxed{D+3 \quad D+2 \quad D+1 \quad D}$$

Il dato di origine deve essere compreso tra -1,0 e 1,0. Se il valore assoluto dei dati di origine supera 1,0, verrà generato un errore e l'istruzione non verrà eseguita.

Il risultato viene inviato ai canali da D a D+3 come angolo (in radianti) compreso tra 0 e π .

Lo schema seguente illustra la relazione tra i dati di ingresso e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. ON se il valore assoluto dei dati di origine supera 1,0. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | Non modificato |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | Non modificato |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

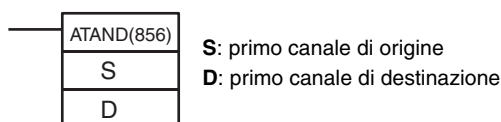
3-16-16 DOUBLE ARC TANGENT: ATAND(856)

Scopo

Calcola l'arcotangente di un numero a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali del risultato specificati. La funzione di arcotangente è l'inverso della funzione di tangente e restituisce l'angolo che produce un dato valore di tangente.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ATAND(856) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ATAND(856) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|-------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |

| Area | S | D |
|--|--|---|
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

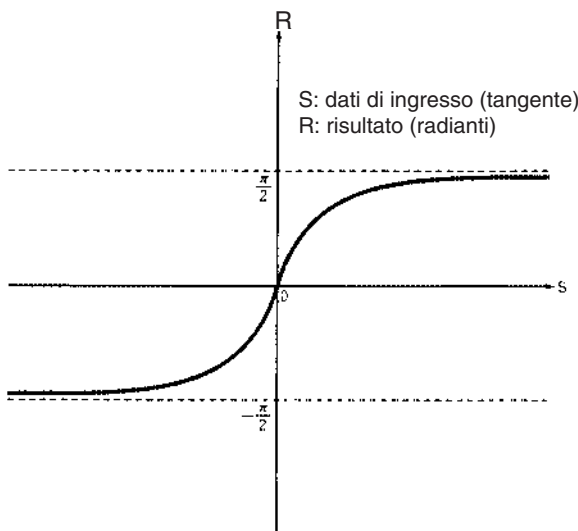
ATAND(856) calcola l'angolo (in radianti) per un valore di tangente espresso come numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato a D - D+3.

(I dati a virgola mobile devono essere nel formato IEEE754).

$$\text{TAN}^{-1}(\text{S+3} \dots \text{S+2} \dots \text{S+1} \dots \text{S}) \rightarrow \text{D+3} \dots \text{D+2} \dots \text{D+1} \dots \text{D}$$

Il risultato viene inviato ai canali da D a D+3 come angolo (in radianti) compreso nell'intervallo da $-\pi/2$ a $\pi/2$.

Lo schema seguente illustra la relazione tra i dati di ingresso e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | Non modificato |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

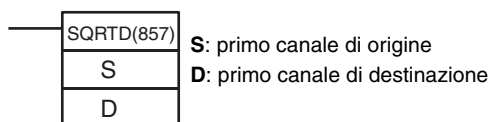
3-16-17 DOUBLE SQUARE ROOT: SQRTD(857)

Scopo

Calcola la radice quadrata di un numero a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | SQRTD(857) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SQRTD(857) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

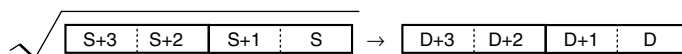
Caratteristiche operando

| Area | S | D |
|-------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |

| Area | S | D |
|--|--|---|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

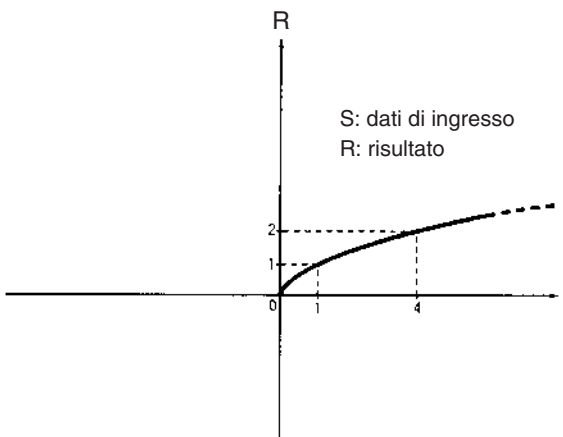
SQRD(857) calcola la radice quadrata del numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3 (i dati a virgola mobile devono essere nel formato IEEE754).



I dati di origine devono essere positivi. Se sono negativi, verrà generato un errore e l'istruzione non verrà eseguita.

Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come ±∞.

Lo schema seguente illustra la relazione tra i dati di ingresso e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono negativi. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione (64 bit). |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | Non modificato |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

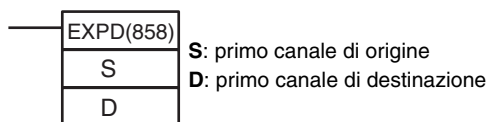
3-16-18 DOUBLE EXPONENT: EXPD(858)

Scopo

Calcola il valore esponenziale naturale (in base e) di un numero a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali del risultato specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | EXPD(858) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @EXPD(858) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

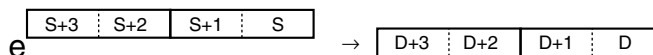
Caratteristiche operando

| Area | S | D |
|-------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |

| Area | S | D |
|--|--|---|
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

EXPD(858) calcola il valore esponenziale naturale (in base e) del numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3. In altre parole, EXP(467) calcola e^x (x = origine) e inserisce il risultato nei canali da D a D+3.

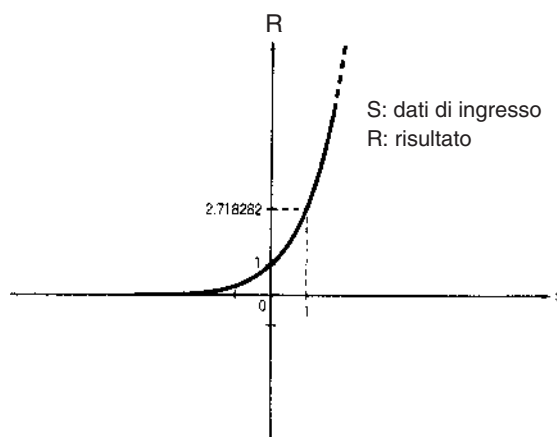


Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Se il valore assoluto del risultato è minore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow e il risultato verrà inviato come 0.

Nota La costante e è 2,718282.

Lo schema seguente illustra la relazione tra i dati di ingresso e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione (64 bit). |
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione (64 bit). |
| Flag negativo | N | Non modificato |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

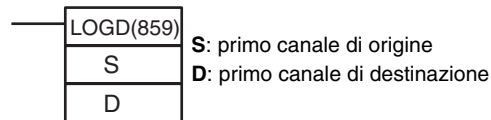
3-16-19 DOUBLE LOGARITHM: LOGD(859)

Scopo

Calcola il logaritmo naturale (in base e) di un numero a virgola mobile in doppia precisione (64 bit) e invia il risultato ai canali di destinazione specificati.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | LOGD(859) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @LOGD(859) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | D |
|-------------------------|------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |

| Area | S | D |
|--|--|---|
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

LOGD(859) calcola il logaritmo naturale (in base e) del numero a virgola mobile in doppia precisione (64 bit) nei canali da S a S+3 e invia il risultato ai canali D - D+3

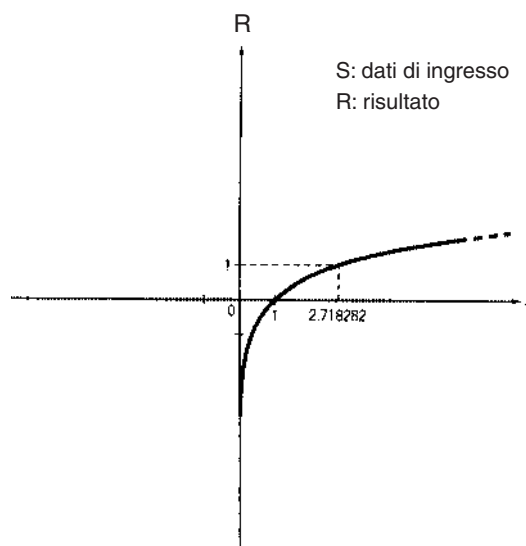
$$\log_e \begin{bmatrix} S+3 & S+2 & S+1 & S \end{bmatrix} \rightarrow \begin{bmatrix} D+3 & D+2 & D+1 & D \end{bmatrix}$$

I dati di origine devono essere positivi. Se sono negativi, verrà generato un errore e l'istruzione non verrà eseguita.

Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow e il risultato verrà inviato come $\pm\infty$.

Nota La costante e è 2,718282.

Lo schema seguente illustra la relazione tra i dati di ingresso e il risultato.



Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati di origine non vengono riconosciuti come dati a virgola mobile. ON se i dati di origine sono negativi. ON se i dati di origine sono di tipo NaN. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione (64 bit). |
| Flag di underflow | UF | Non modificato |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

È necessario che i dati di origine nei canali da S a S+3 siano nel formato dati a virgola mobile IEEE754.

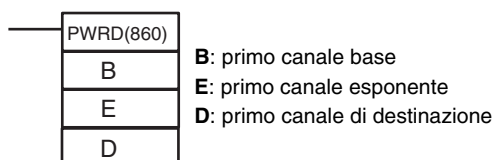
3-16-20 DOUBLE EXPONENTIAL POWER: PWRD(860)

Scopo

Eleva un numero a virgola mobile in doppia precisione (64 bit) alla potenza di un altro numero a virgola mobile in doppia precisione (64 bit).

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | PWRD(860) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PWRD(860) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

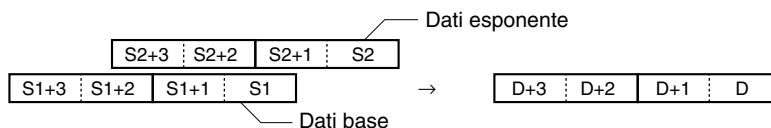
Caratteristiche operando

| Area | B | E | D |
|-------------------------|------------------------|-----------------|---|
| Area CIO | Da CIO 0000 a CIO 6140 | | |
| Area di lavoro | Da W000 a W508 | | |
| Area bit di ritentività | Da H000 a H508 | | |
| Area bit ausiliaria | Da A000 a A956 | Da A448 ad A956 | |
| Area del temporizzatore | Da T0000 a T4092 | | |
| Area del contatore | Da C0000 a C4092 | | |
| Area DM | Da D00000 a D32764 | | |

| Area | B | E | D |
|--|--|---|---|
| Area EM senza banco | Da E00000 a E32764 | | |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

PWRD(860) eleva il numero a virgola mobile in doppia precisione (64 bit) nei canali B - B+3 alla potenza del numero a virgola mobile in doppia precisione (64 bit) nei canali E - E+3. In altre parole, PWR(840) calcola X^Y (X = contenuto da B a B+3; Y = contenuto da E a E+3).



Ad esempio, quando i canali base (da B a B+3) contengono 3,1 e i canali esponente (da E a E+3) contengono 3, il risultato sarà $3,1^3$ o 29,791.

Se il valore assoluto del risultato è maggiore del valore massimo che può essere espresso come dato a virgola mobile, viene attivato il flag di overflow.

Se il valore assoluto del risultato è inferiore al valore minimo che può essere espresso come dato a virgola mobile, viene attivato il flag di underflow.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se i dati base (da B a B+3) o i dati esponente (da E a E+3) non vengono riconosciuti come dati a virgola mobile. ON se i dati base (da B a B+3) o i dati esponente (da E a E+3) non sono un numero (NaN). ON se i dati base (da B a B+3) sono 0 e i dati esponente (da E a E+3) sono inferiori a 0 (divisione per 0). ON se i dati base (da B a B+3) sono negativi e i dati esponente (da E a E+3) non sono numeri interi (radice di un numero negativo). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se esponente e mantissa del risultato sono entrambi 0. OFF in tutti gli altri casi. |
| Flag di overflow | OF | ON se il valore assoluto del risultato è troppo elevato per essere espresso come valore a virgola mobile in doppia precisione. |

| Nome | Eti-chetta | Operazione |
|-------------------|------------|--|
| Flag di underflow | UF | ON se il valore assoluto del risultato è troppo piccolo per essere espresso come valore a virgola mobile in doppia precisione. |
| Flag negativo | N | ON se il risultato è negativo. OFF in tutti gli altri casi. |

Avvertenze

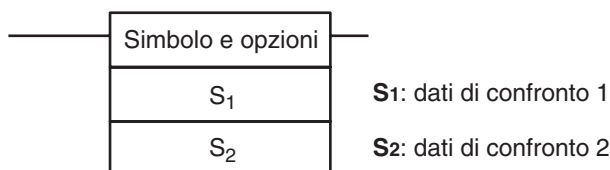
È necessario che i dati base (da B a B+3) e i dati esponente (da E a E+3) siano nel formato dati a virgola mobile IEEE754.

3-16-21 Istruzioni di ingresso a virgola mobile in doppia precisione**Scopo**

Le istruzioni di confronto di ingresso confrontano due valori a virgola mobile in doppia precisione (64 bit, formato IEEE754) e creano una condizione di esecuzione ON quando la condizione di confronto è vera.

Queste istruzioni sono supportate solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Nota Per ulteriori informazioni sulle istruzioni di confronto di ingresso di valori binari con segno e senza segno, consultare la sezione 3-7-1 *Istruzioni di confronto di ingresso (da 300 a 328)*. Per ulteriori informazioni sulle istruzioni di confronto di ingresso a virgola mobile in singola precisione, consultare la sezione 3-15-21 *Istruzioni di confronto a virgola mobile in singola precisione*.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il confronto risulta vero | Istruzione di confronto di ingresso |
|-------------------------|--|-------------------------------------|
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S ₁ | S ₂ |
|--------------------------------------|---|----------------|
| Area CIO | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | Da W000 a W508 | |
| Area bit di ritenività | Da H000 a H508 | |
| Area bit ausiliaria | Da A000 a A956 | |
| Area del temporizzatore | Da T0000 a T4092 | |
| Area del contatore | Da C0000 a C4092 | |
| Area DM | Da D00000 a D32764 | |
| Area EM senza banco | Da E00000 a E32764 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |

| Area | S ₁ | S ₂ |
|--|--|----------------|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

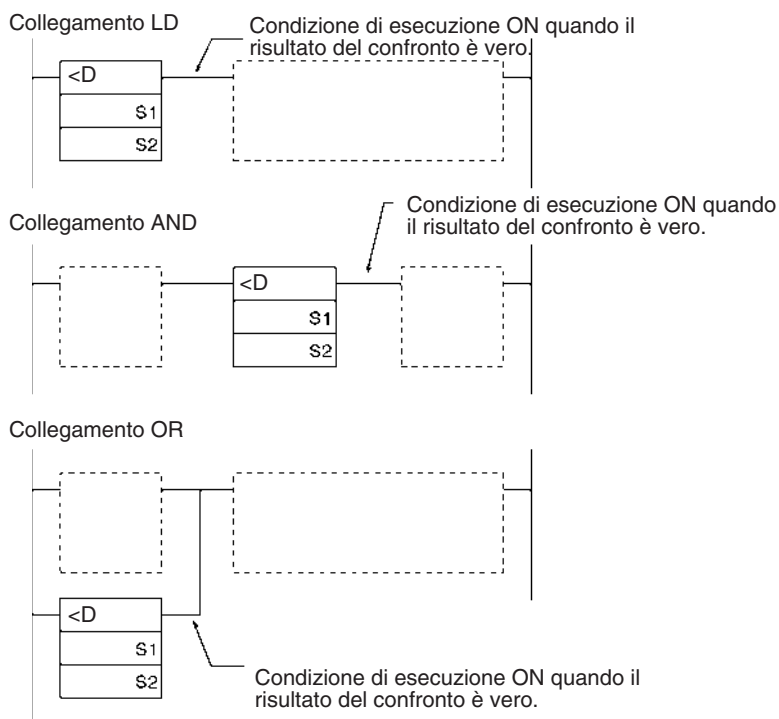
Descrizione

L'istruzione di confronto di ingresso confronta i dati specificati in S₁ e S₂ come valori a virgola mobile in doppia precisione (64 bit, formato IEEE754) e crea una condizione di esecuzione ON quando la condizione di confronto è vera. Quando i dati vengono memorizzati nei canali, S₁ e S₂ indicano i primi di quattro canali che contengono i dati a 64 bit. I dati a virgola mobile a 64 bit non possono essere immessi come costanti.

Immissione di istruzioni

Le istruzioni di confronto di ingresso vengono interpretate esattamente come le istruzioni LD, AND e OR per controllare l'esecuzione delle istruzioni successive.

| Tipo di ingresso | Operazione |
|------------------|---|
| LD | È possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |
| AND | Non è possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |
| OR | È possibile collegare l'istruzione direttamente alla barra di distribuzione sinistra. |



Opzioni

Utilizzando questi tre tipi di ingresso e sei simboli, ne derivano 18 diverse combinazioni possibili.

| Simbolo | Opzione (formato dati) |
|------------------------|---|
| = (uguale) | D: dati a virgola mobile in doppia precisione |
| < > (diverso) | |
| < (minore di) | |
| <= (minore o uguale) | |
| > (maggiore di) | |
| >= (maggiore o uguale) | |

Sommario delle istruzioni di confronto di ingresso

Nella tabella seguente vengono riportati i codici di funzione, i codici mnemonici, i nomi e le funzioni delle 18 istruzioni di confronto di ingresso a virgola mobile in singola precisione ($C1=S_1+3$, S_1+2 , S_1+1 , S_1 e $C2=S_2+3$, S_2+2 , S_2+1 , S_2).

| Codice | Mnemonico | Nome | Funzione |
|--------|-----------|--|----------------------|
| 335 | LD=D | LOAD DOUBLE FLOATING EQUAL | Vera se $C1 = C2$ |
| | AND=D | AND DOUBLE FLOATING EQUAL | |
| | OR=D | OR DOUBLE FLOATING EQUAL | |
| 336 | LD <>D | LOAD DOUBLE FLOATING NOT EQUAL | Vera se $C1 \neq C2$ |
| | AND <>D | AND DOUBLE FLOATING NOT EQUAL | |
| | OR <>D | OR DOUBLE FLOATING NOT EQUAL | |
| 337 | LD <D | LOAD DOUBLE FLOATING LESS THAN | Vera se $C1 < C2$ |
| | AND <D | AND DOUBLE FLOATING LESS THAN | |
| | OR <D | OR DOUBLE FLOATING LESS THAN | |
| 338 | LD <=D | LOAD DOUBLE FLOATING LESS THAN OR EQUAL | Vera se $C1 \leq C2$ |
| | AND <=D | AND DOUBLE FLOATING LESS THAN OR EQUAL | |
| | OR <=D | OR DOUBLE FLOATING LESS THAN OR EQUAL | |
| 339 | LD >D | LOAD DOUBLE FLOATING GREATER THAN | Vera se $C1 > C2$ |
| | AND >D | AND DOUBLE FLOATING GREATER THAN | |
| | OR >D | OR DOUBLE FLOATING GREATER THAN | |
| 340 | LD >=D | LOAD DOUBLE FLOATING GREATER THAN OR EQUAL | Vera se $C1 \geq C2$ |
| | AND >=D | AND DOUBLE FLOATING GREATER THAN OR EQUAL | |
| | OR >=D | OR DOUBLE FLOATING GREATER THAN OR EQUAL | |

Flag

In questa tabella, $C1$ = contenuto di $S1 - S1+3$ e $C2$ = contenuto di $S2 - S2+3$.

| Nome | Eti-chetta | Operazione |
|---------------------------|------------|--|
| Flag di errore | ER | ON se $C1$ o $C2$ non è un numero a virgola mobile valido (NaN). ON se $C1$ o $C2$ è $+\infty$. ON se $C1$ o $C2$ è $-\infty$. OFF in tutti gli altri casi. |
| flag di maggiore | > | ON se $C1 > C2$. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | > = | ON se $C1 \geq C2$. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se $C1 = C2$. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | = | ON se $C1 \neq C2$. OFF in tutti gli altri casi. |

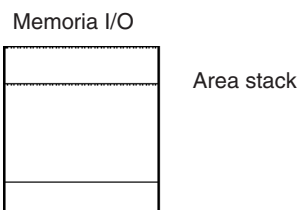
| Istruzione | Mnemonico | Codice funzione | Pagina |
|------------------------|-----------|-----------------|--------|
| LAST IN FIRST OUT | LIFO | 634 | 675 |
| DIMENSION RECORD TABLE | DIM | 631 | 678 |
| SET RECORD LOCATION | SETR | 635 | 681 |
| GET RECORD NUMBER | GETR | 636 | 683 |
| DATA SEARCH | SRCH | 181 | 685 |
| SWAP BYTES | SWAP | 637 | 687 |
| FIND MAXIMUM | MAX | 182 | 689 |
| FIND MINIMUM | MIN | 183 | 693 |
| SUM | SUM | 184 | 697 |
| FRAME CHECKSUM | FCS | 180 | 700 |
| STACK NUMBER OUTPUT | SNUM | 638 | 704 |
| STACK DATA READ | SREAD | 639 | 707 |
| STACK DATA OVERWRITE | SWRIT | 640 | 710 |
| STACK DATA INSERT | SINS | 641 | 713 |
| STACK DATA DELETE | SDEL | 642 | 716 |

Tutte queste istruzioni definiscono o elaborano un gruppo di canali. Il gruppo di canali in uno stack è definito da SSET(630), il gruppo di canali in una tabella di record è definito da DIM(631) e il gruppo di canali utilizzato nelle istruzioni su intervalli di valori è definito in modo indipendente in ciascuna istruzione.

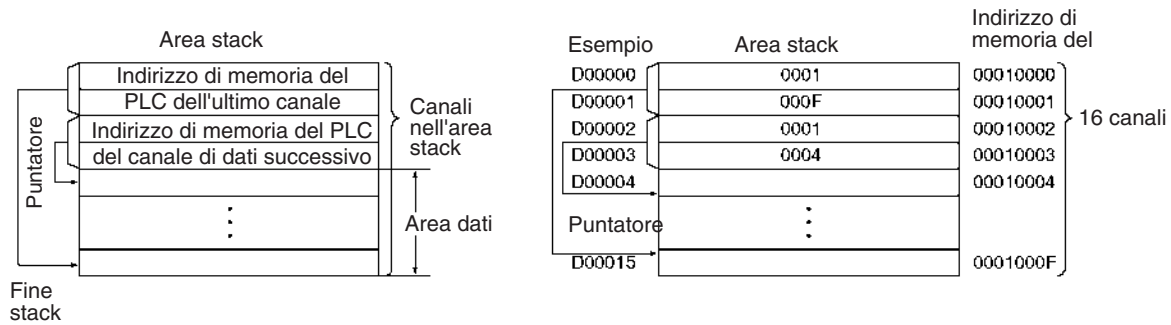
| Gruppo | Scopo | Istruzioni |
|-------------------|--|--|
| Stack | Elabora tabelle di dati FIFO (first-in first-out) o LIFO (last-in first-out). | SSET(630), PUSH(632), FIFO(633), LIFO(634), SREAD(639), SWRIT(640), SINS(641), SDEL(642) e SNUM(638) |
| Tabella di record | Elabora tabelle di dati costituite da record (le dimensioni dei record sono definite dall'utente). | DIM(631), SETR(635) e GETR(636) |
| Intervallo | Elabora un intervallo di canali per trovare i valori quali il checksum, un valore specifico, il valore massimo o minimo nell'intervallo. | FCS(180), SRCH(181), MAX(182), MIN(183), SUM(184) e SWAP(637) |

Istruzioni di stack

Le istruzioni di stack vengono utilizzate per tabelle di dati definite in modo specifico denominate stack. I primi due canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack, mentre i due canali successivi contengono il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale che verrà sovrascritto dalla successiva istruzione PUSH(632).



Nello schema seguente viene illustrata la struttura base di uno stack.

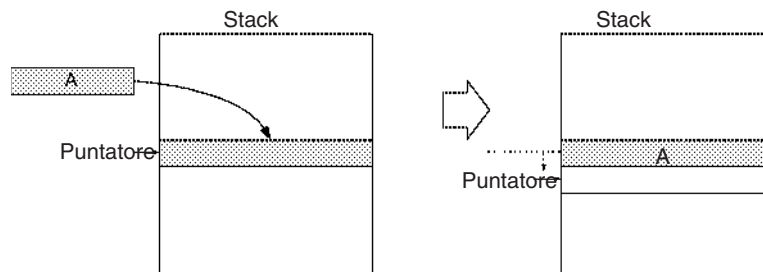


Le istruzioni riportate di seguito definiscono aree di stack o agiscono su tali aree. Fondamentalmente, l'istruzione PUSH(632) memorizza i dati nel canale successivo disponibile nello stack. FIFO(633) e LIFO(634) leggono i dati dallo stack. FIFO(633) legge il primo canale che è stato memorizzato, mentre LIFO(634) legge l'ultimo canale che è stato memorizzato.

Le ultime cinque istruzioni sono supportate solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D. SNUM(638) conta il numero di elementi di dati (canali) nello speck specificato. Questa istruzione, ad esempio, può essere utilizzata per indicare il numero di elementi su un nastro trasportatore. Le istruzioni SREAD(639), SWRIT(640), SINS(641) e SDEL(642) permettono rispettivamente di leggere, sovrascrivere, inserire ed eliminare gli elementi di dati in uno stack. Ad esempio, quando gli elementi vengono gestiti su un nastro trasportatore, queste istruzioni possono aggiungere, rimuovere o modificare un elemento di dati nello stack che corrisponde a un elemento sul nastro trasportatore.

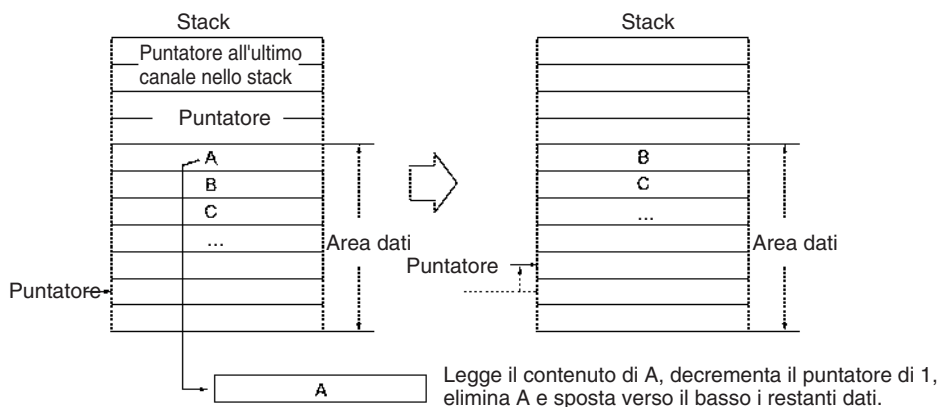
PUSH(632)

Memorizza i dati nell'indirizzo indicato dal puntatore dello stack e incrementa il puntatore di uno.



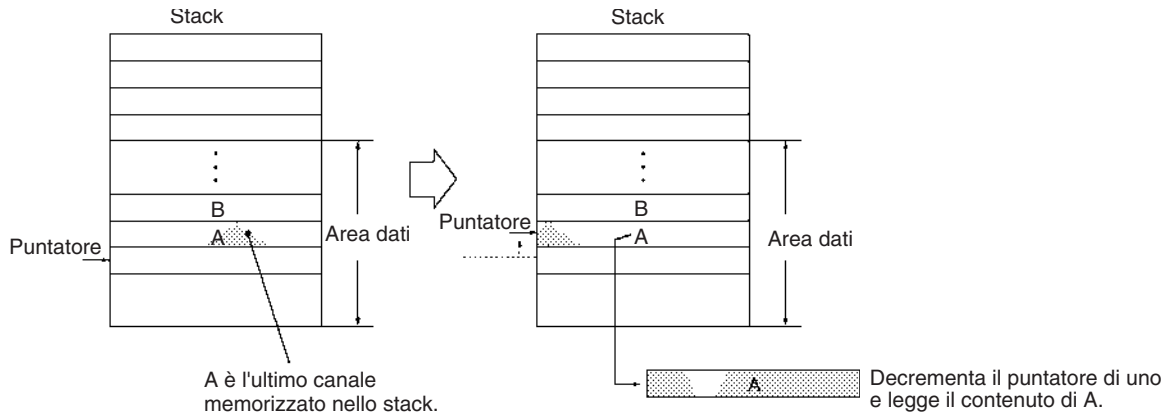
FIFO(633)

Legge il primo canale di dati (il meno recente) memorizzato nello stack, sposta i restanti dati di un canale verso il basso e decrementa il puntatore di uno.



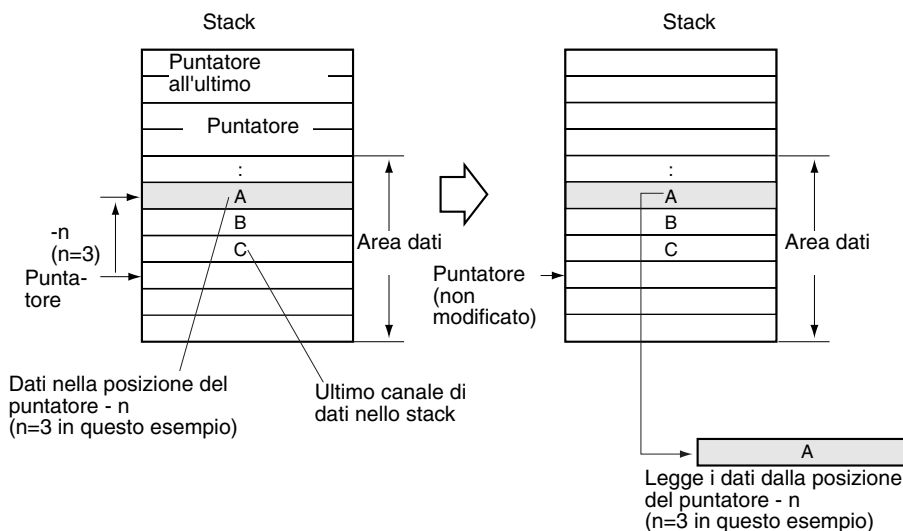
LIFO(634)

Legge l'ultimo (il più recente) canale di dati memorizzato nello stack. Decrementa il puntatore di uno e legge i dati a questo indirizzo (i dati più recenti memorizzati nello stack). I dati letti non verranno cancellati.



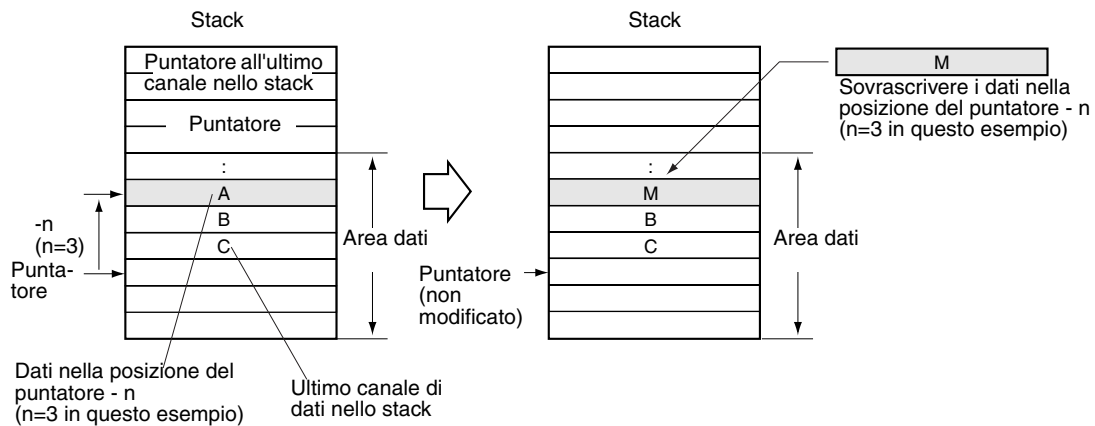
SREAD(639)

Legge i dati dall'elemento dati specificato nello stack. Il valore di offset indica la posizione del canale desiderato (il numero di canali prima della posizione corrente del puntatore).

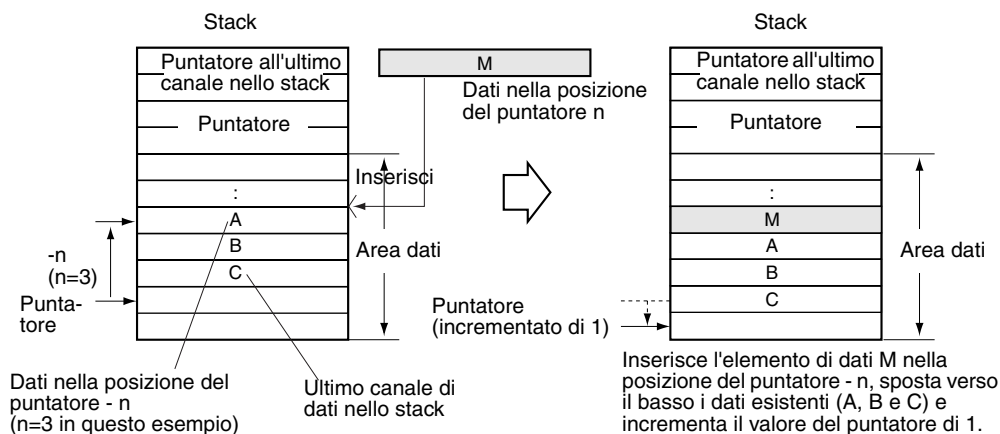


SWRIT(640)

Scrive i dati di origine nell'elemento di dati specificato nello stack, sovrascrivendo i dati già esistenti. Il valore di offset indica la posizione del canale desiderato (il numero di canali prima della posizione corrente del puntatore).

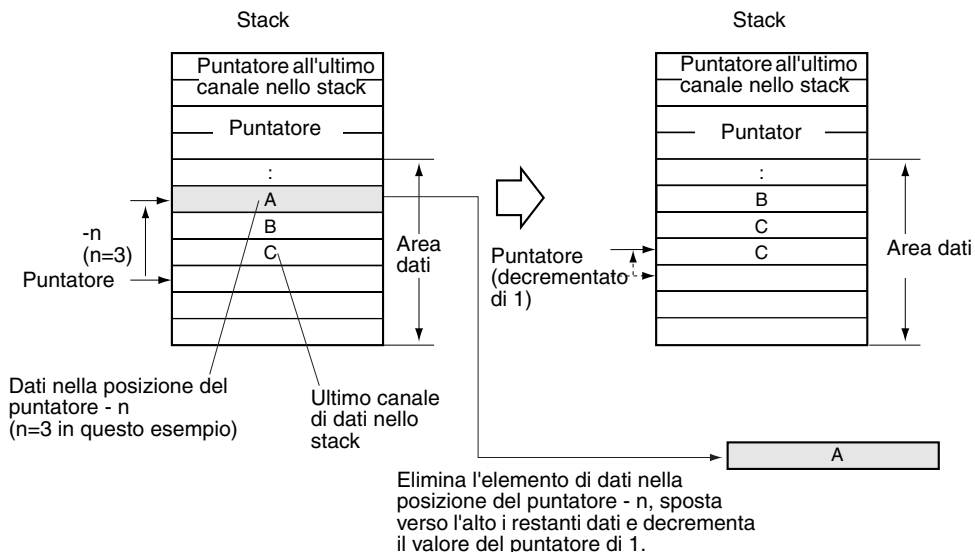
**SINS(641)**

Inserisce i dati di origine nella posizione specificata nello stack e fa scorrere verso il basso i dati rimanenti dello stack. Il valore di offset indica la posizione del canale desiderato (il numero di canali prima della posizione corrente del puntatore).



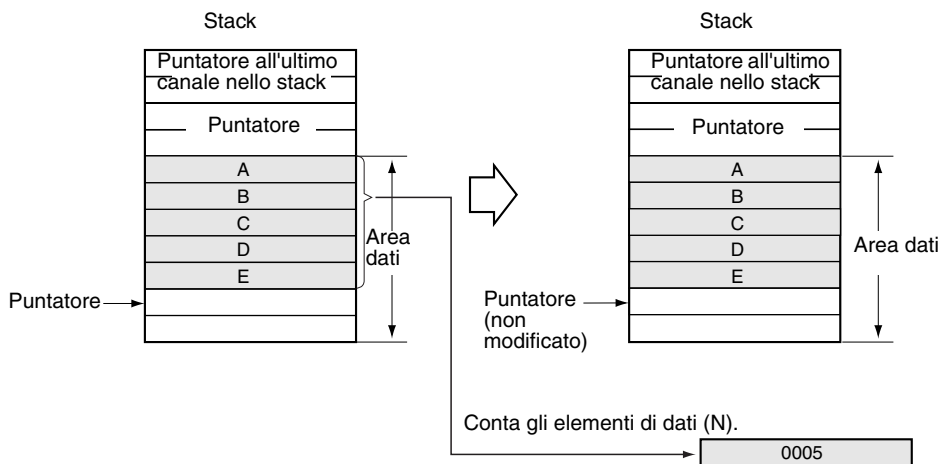
SDEL(642)

Elimina l'elemento di dati che si trova nella posizione specificata nello stack e fa scorrere verso l'alto i dati rimanenti nello stack. Il valore di offset indica la posizione del canale desiderato (il numero di canali prima della posizione corrente del puntatore).



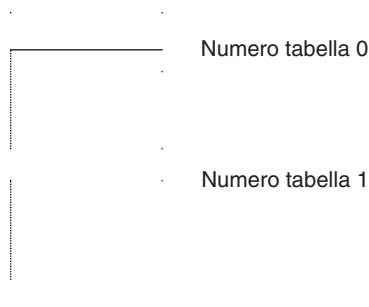
SNUM(638)

Conta la quantità di dati dello stack (numero di canali di dati) dal puntatore dello stack fino all'inizio dell'area dati.

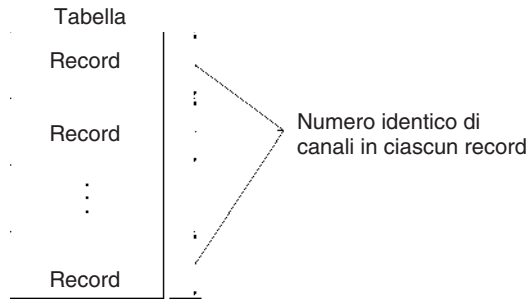


Istruzioni per le tabelle di record

Una serie di dati composta da più di un record avente lo stesso numero di canali in ciascun record viene definita dati di tabella. I dati di tabella memorizzati nella memoria di I/O specificata possono essere registrati come area tabella tramite l'istruzione DIM. È possibile definire fino a 16 tabelle distinte, con numeri di tabella da 0 a 15.



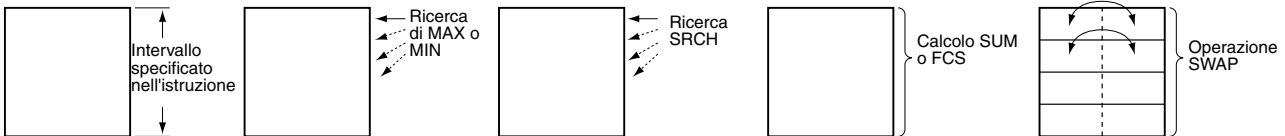
Nello schema seguente viene illustrata la struttura base di una tabella di record. Ogni record nella tabella ha lo stesso numero di canali.



È possibile utilizzare i registri indice (IR) per fare riferimento in modo indiretto ai dati della tabella. Il calcolo dell'indirizzo del record viene eseguito facilmente tramite le istruzioni SETR(635) (SET RECORD NUMBER) e GETR(636) (GET RECORD NUMBER).

Istruzioni su intervalli di valori

Le istruzioni su intervalli di valori qui indicate agiscono su un intervallo di canali specificato e consentono di trovare i valori massimo (MAX(182)) e minimo (MIN(183)), cercare un determinato valore (SRCH(181)), calcolare la somma (SUM(184)) o l'FCS (FCS(180)) e scambiare il contenuto dei byte all'estrema sinistra e all'estrema destra nei canali (SWAP(637)).

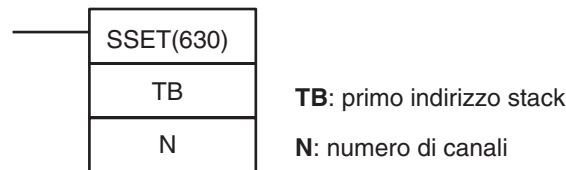


3-17-1 SET STACK: SSET(630)

Scopo

Definisce uno stack della lunghezza specificata cominciando dal canale specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SSET(630) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SSET(630) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

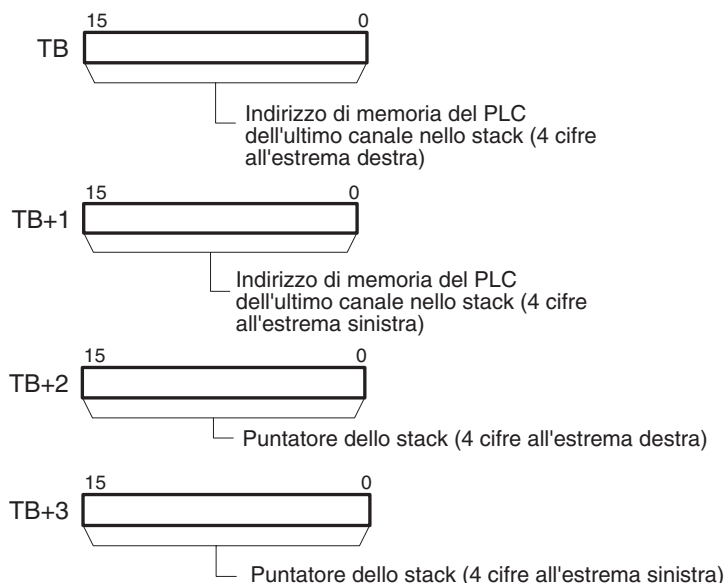
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

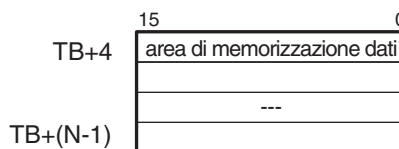
Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo che verrà sovrascritto dall'istruzione PUSH(632).



Da TB+4 a TB+(N-1): area di memorizzazione dati

Il resto dello stack viene usato per memorizzare i dati.



- Nota**
1. Il valore iniziale del puntatore dello stack è sempre l'indirizzo di memoria del PLC di TB+4.
 2. TB e TB+(N-1) devono essere nella stessa area dati.

Caratteristiche operando

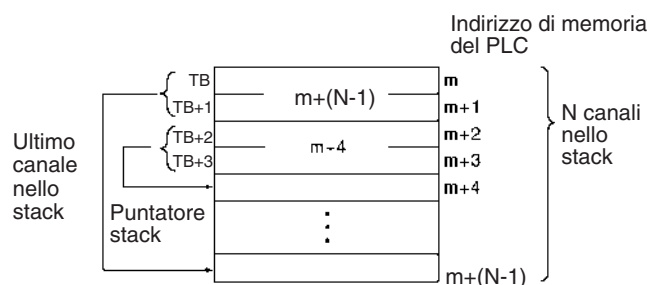
| Area | TB | N |
|--------------------------------------|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 a A959 | Da A000 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Da #0005 a #FFFF (binario) o da &5 a &65.535 |
| Registri dati | --- | Da DR0 a DR15 |

| Area | TB | N |
|--|--|---|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

SSET(630) assicura uno stack con N canali cominciando da TB e terminando a TB+(N-1). I primi due canali dello stack (TB+1 e TB) contengono l'indirizzo di memoria del PLC esadecimale a otto cifre dell'ultimo canale nello stack. Il due canali successivi (TB+3 e TB+2) contengono il puntatore dello stack. Il puntatore dello stack è l'indirizzo di memoria del PLC del canale successivo presente nello stack che verrà sovrascritto dall'istruzione PUSH(632). Il suo indirizzo iniziale è l'indirizzo di TB+4.

SSET(630) inizializza automaticamente l'area dati dello stack (da TB+4 a TB+(N-1)) sui valori zero. Nello schema seguente viene illustrata la struttura base di uno stack.



SSET(630) semplicemente stabilisce e inizializza uno stack. Utilizzare le istruzioni seguenti per memorizzare i dati e leggerli nello stack.

1,2,3...

1. PUSH(632) memorizza i dati nello stack un canale alla volta.
2. FIFO(633) e LIFO(634) leggono i dati dallo stack. FIFO(633) legge il primo canale che è stato memorizzato, mentre LIFO(634) legge l'ultimo canale che è stato memorizzato.
3. Il valore del puntatore dello stack nel canale di controllo dello stack viene aggiornato automaticamente quando viene eseguita un'istruzione PUSH(632), FIFO(633) o LIFO(634). Normalmente non è necessario occuparsi del canale di controllo dello stack. Quando, invece di utilizzare le istruzioni, si accede ai contenuti dello stack in modo diverso, è necessario impostare il valore del puntatore dello stack tramite il registro indice (IR) per il riferimento indiretto.

Flag

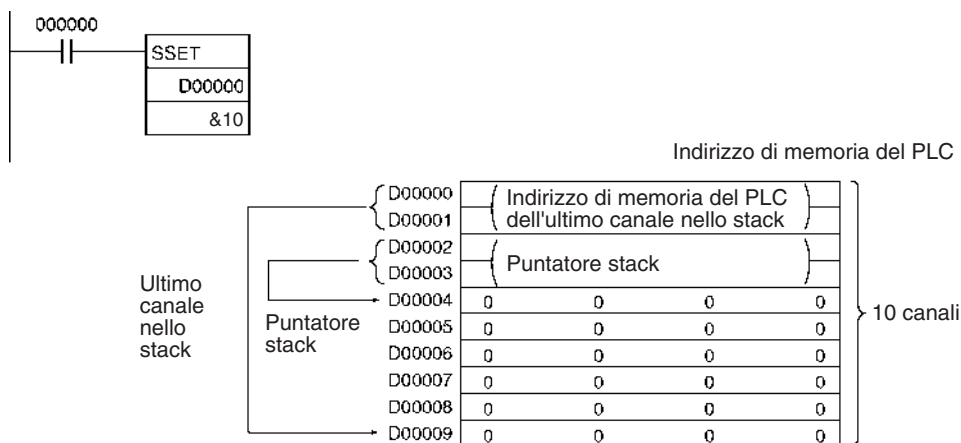
| Nome | Eti-chetta | Operazione |
|----------------|------------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato 0005 - FFFF. OFF in tutti gli altri casi. |

Avvertenze

Il valore minimo per il numero di canali nello stack (N) è 5 poiché N include i quattro canali che contengono il puntatore all'ultimo canale nello stack e il puntatore dello stack. Se N non è incluso nell'intervallo da 0005 a FFFF, si verificherà un errore.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SSET(630) assicura uno stack di 10 canali da D00000 a D00009. D00000 e D00001 contengono l'indirizzo di memoria del PLC dell'ultimo canale presente nello stack. D00002 e D00003 contengono il puntatore dello stack. Lo stack stesso comincia in D00004.

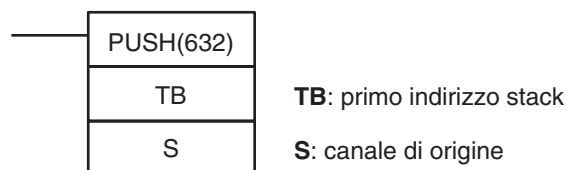


3-17-2 PUSH ONTO STACK: PUSH(632)

Scopo

Scrive un canale di dati nello stack specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | PUSH(632) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PUSH(632) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

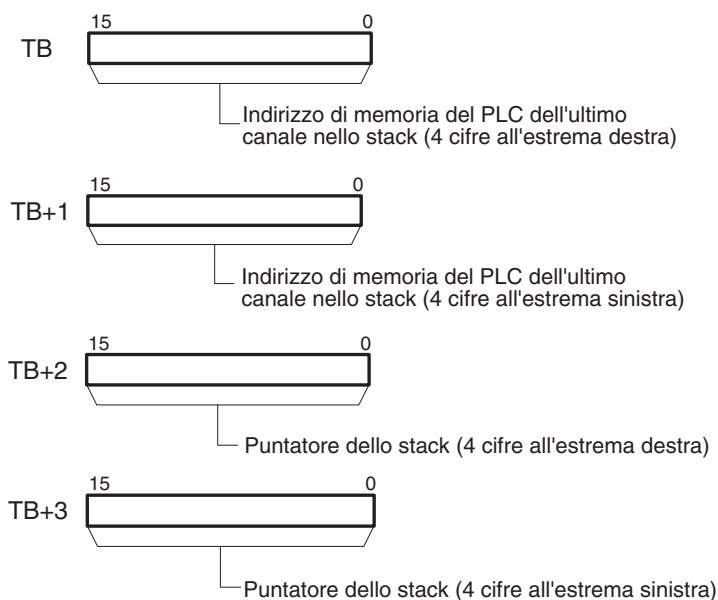
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

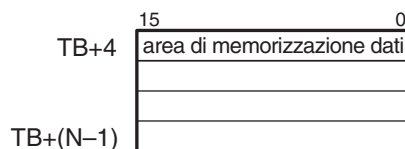
Operandi

Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo che verrà sovrascritto dall'istruzione PUSH(632).

**Da TB+4 a TB+(N-1): area di memorizzazione dati**

Il resto dello stack viene usato per memorizzare i dati.

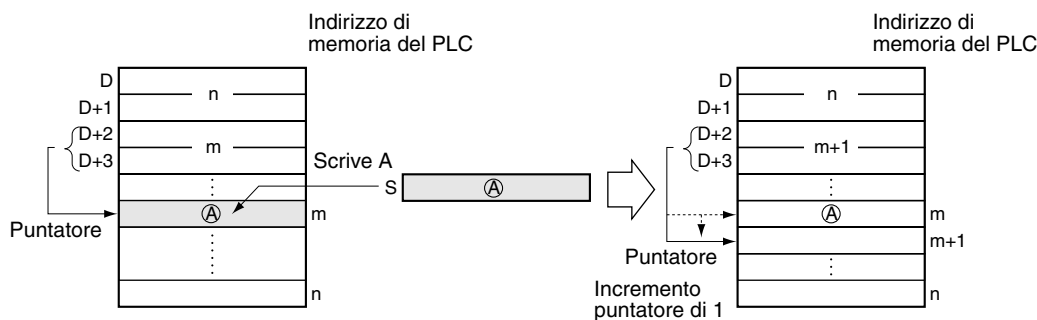
**Caratteristiche operando**

| Area | TB | S |
|--------------------------------------|--|----------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 a A959 | Da A000 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | Da #0000 a #FFFF (binario) |
| Registri dati | --- | Da DR0 a DR15 |

| Area | TB | S |
|--|--|---|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

PUSH(632) scrive il contenuto di S nell'indirizzo indicato dal puntatore dello stack (TB+3 e TB+2) e incrementa di uno il puntatore dello stack.



Dopo avere utilizzato l'istruzione PUSH(632) per scrivere i dati nello stack, è possibile leggere i dati dallo stack mediante le istruzioni FIFO(633) e LIFO(634).

Flag

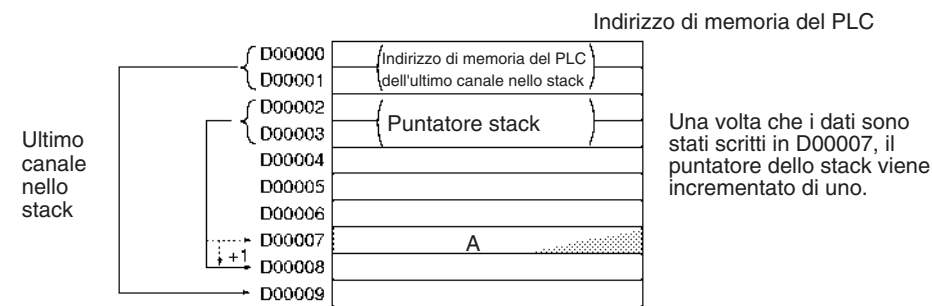
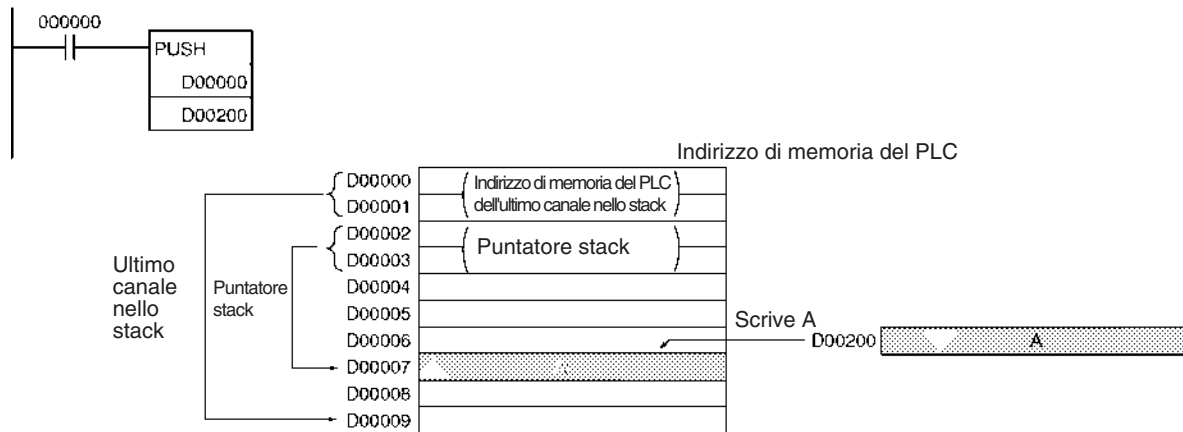
| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se l'indirizzo specificato dal puntatore dello stack (TB+3 e TB+2) supera l'ultimo canale presente nello stack. (Errore di overflow dello stack). OFF in tutti gli altri casi. |

Avvertenze

È necessario che lo stack sia stato definito in precedenza tramite l'istruzione SSET(630).

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, PUSH(632) copia il contenuto di D002000 nello stack cominciando da D00000. In questo caso il puntatore dello stack indica D00007.

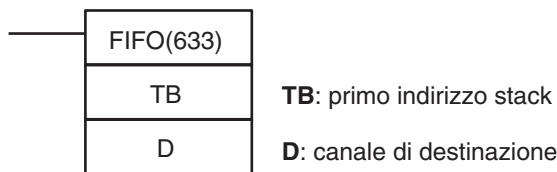


3-17-3 FIRST IN FIRST OUT: FIFO(633)

Scopo

Legge il primo canale dei dati scritti nello stack specificato, ossia i dati nello stack meno recenti.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FIFO(633) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FIFO(633) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

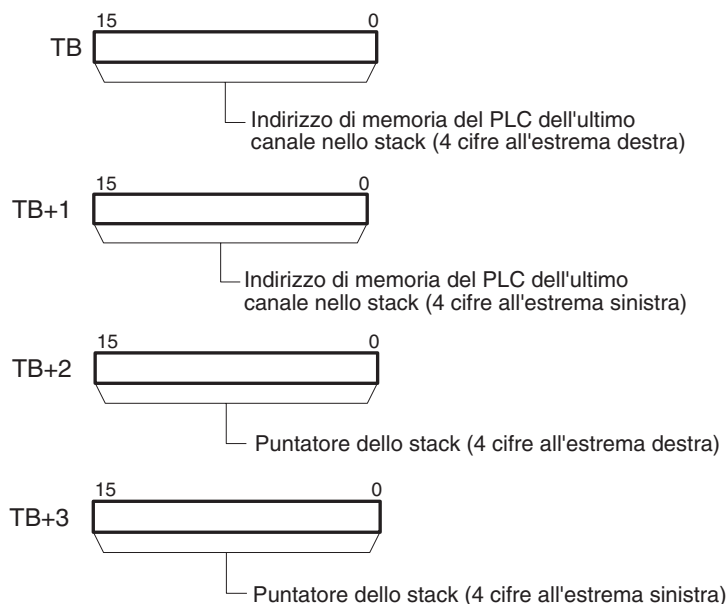
Are di programma applicabili

| | | | |
|-----------------------------------|------------------------------|-------------------|--------------------------|
| Are di programma a blocchi | Are di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

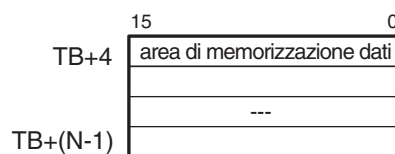
Operandi

Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo che verrà sovrascritto dall'istruzione PUSH(632).

**Da TB+4 a TB+(N-1): area di memorizzazione dati**

Il resto dello stack viene usato per memorizzare i dati.



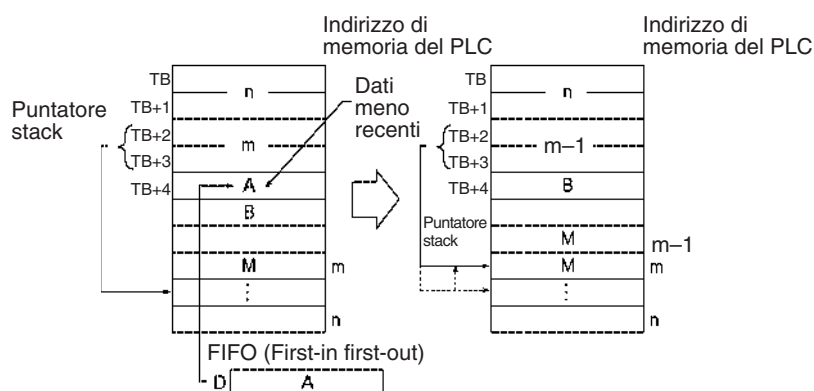
Caratteristiche operando

| Area | TB | D |
|--------------------------------------|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |

| Area | TB | D |
|--|--|---------------|
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

FIFO(633) legge dallo stack il canale di dati meno recente (TB+4) e invia tali dati a D. Successivamente il puntatore dello stack (TB+3 e TB+2) viene decrementato di uno, mentre tutti i restanti dati nello stack vengono spostati verso il basso di un canale e i dati letti da TB+4 vengono eliminati. Restano invariati i dati alla fine dello stack, ovvero l'indirizzo indicato in precedenza dal puntatore dello stack.



È possibile utilizzare l'istruzione FIFO(633) in combinazione con PUSH(632). Dopo avere utilizzato l'istruzione PUSH(632) per scrivere i dati nello stack, è possibile leggere i dati dallo stack su base first-in first-out.

FIFO(633) legge dallo stack i dati iniziali, quindi li elimina per spostare in avanti di uno i dati successivi.

Flag

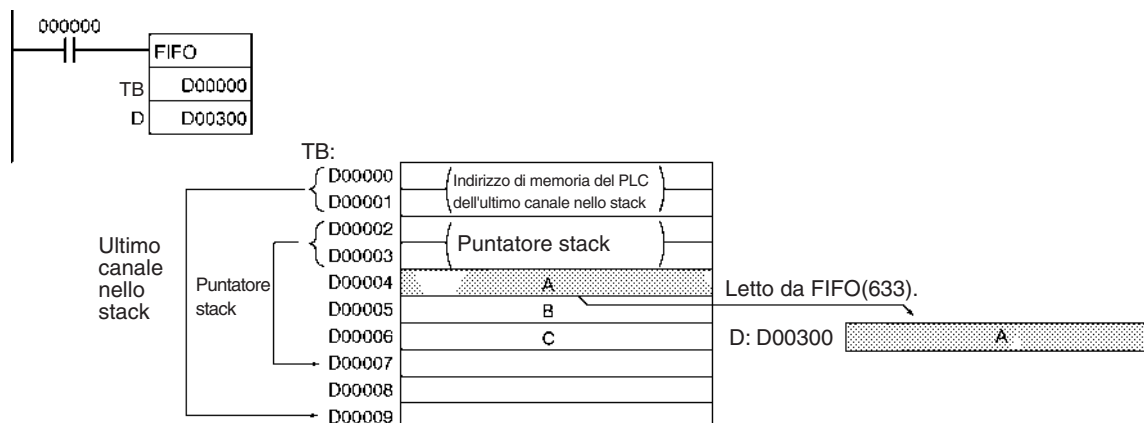
| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se il contenuto del puntatore dello stack (TB+3 e TB+2) è inferiore o uguale all'indirizzo di memoria del PLC del primo canale presente nell'area dati dello stack (TB+4). (Errore di underflow dello stack). OFF in tutti gli altri casi. |

Avvertenze

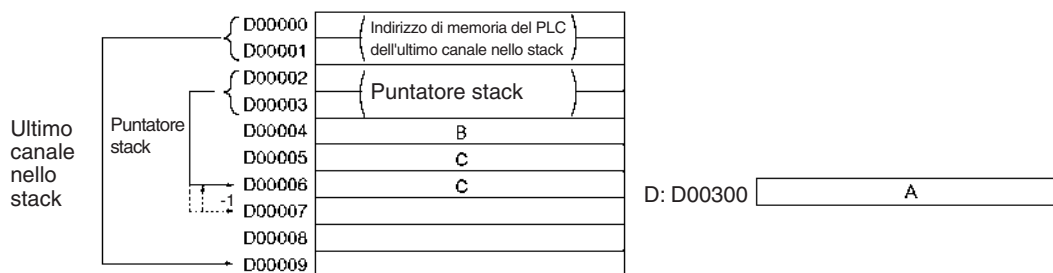
Lo stack che deve essere stato definito in precedenza tramite l'istruzione SSET(630).

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, FIFO(633) legge il contenuto di D00004 (TB+4 per lo stack che comincia da D00000) e scrive tali dati in D00300.



Una volta che i dati sono stati scritti in D00300, il puntatore dello stack viene decrementato di uno e i dati rimanenti vengono spostati verso il basso. (Il contenuto di D00005 viene spostato in D00004 e il contenuto di D00006 in D00005).

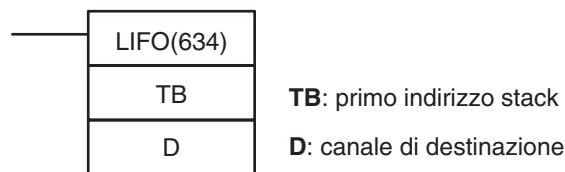


3-17-4 LAST IN FIRST OUT: LIFO(634)

Scopo

Legge l'ultimo canale dei dati scritti nello stack specificato, ossia i dati nello stack più recenti.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | LIFO(634) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @LIFO(634) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

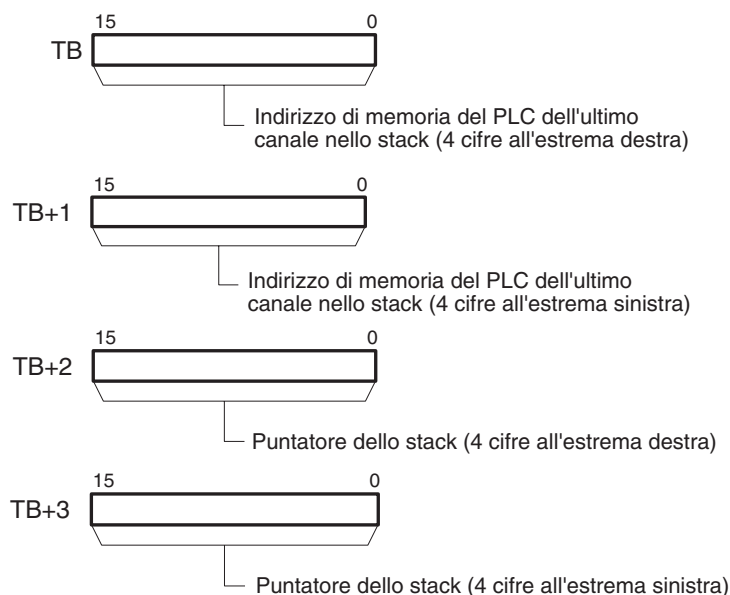
Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

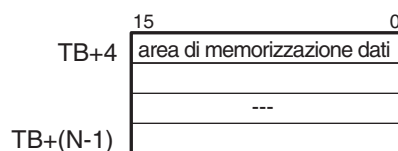
Operandi

Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo che verrà sovrascritto dall'istruzione PUSH(632).

**Da TB+4 a TB+(N-1): area di memorizzazione dati**

Il resto dello stack viene usato per memorizzare i dati.



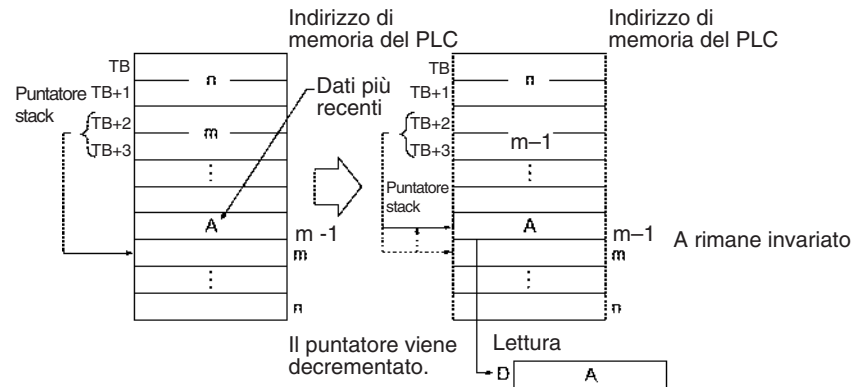
Caratteristiche operando

| Area | TB | D |
|--------------------------------------|--|---------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | Da DR0 a DR15 |

| Area | TB | D |
|--|--|---|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

LIFO(634) legge i dati dall'indirizzo indicato dal puntatore dello stack (il canale di dati più recente nello stack), decrementa il puntatore dello stack di uno e invia i dati a D. Il canale letto resta inalterato.



È possibile utilizzare l'istruzione LIFO(634) in combinazione con PUSH(632). Dopo avere utilizzato l'istruzione PUSH(632) per scrivere i dati nello stack, è possibile usare LIFO(634) per leggere i dati dallo stack su base last-in first-out. Dopo che i dati sono stati memorizzati da PUSH(632), il puntatore dello stack indica l'indirizzo successivo dopo gli ultimi dati.

Flag

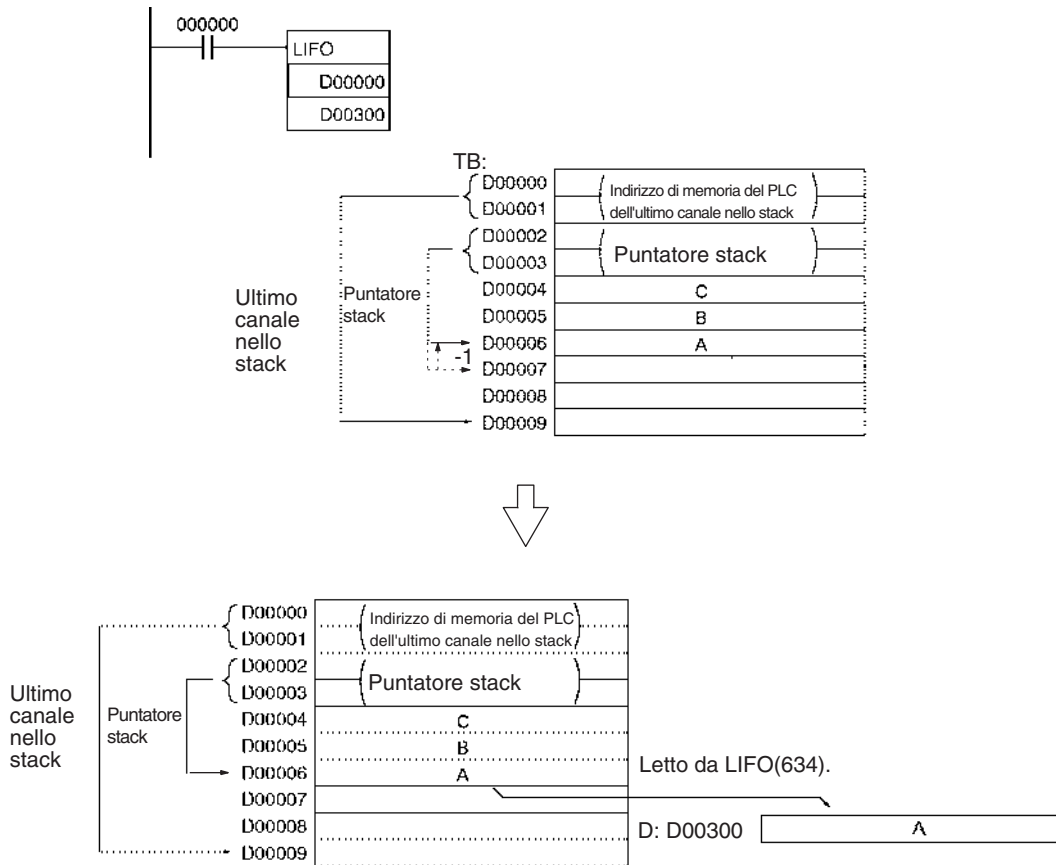
| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se il contenuto del puntatore dello stack (TB+3 e TB+2) è inferiore o uguale all'indirizzo di memoria del PLC del primo canale presente nell'area dati dello stack (TB+4). (Errore di underflow dello stack). OFF in tutti gli altri casi. |

Avvertenze

È necessario che lo stack sia stato definito in precedenza tramite l'istruzione SSET(630).

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, LIFO(634) legge il contenuto del canale indicato dal puntatore dello stack (D00006) e scrive tali dati in D00300.



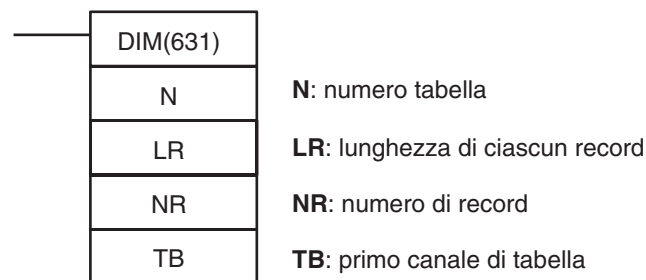
Una volta che i dati sono stati scritti in D00300, il puntatore dello stack viene decrementato di uno. Il contenuto di D00006 resta invariato.

3-17-5 DIMENSION RECORD TABLE: DIM(631)

Scopo

Definisce come tabella di record l'area di memoria I/O specificata, dichiarando la lunghezza di ciascun record e il numero di record. È possibile definire un numero massimo di 16 tabelle di record.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | DIM(631) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DIM(631) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi**N: numero tabella**

Indica il numero della tabella. N deve essere compreso tra 0 e 15.

LR: lunghezza di ciascun record

Indica il numero di canali in ciascun record. LR deve essere compreso tra 0001 e FFFF esadecimale (da 1 a 65.535 canali).

NR: numero di record

Indica il numero di record nella tabella. NR deve essere compreso tra 0001 e FFFF esadecimale (da 1 a 65.535 canali).

TB: primo canale di tabella

Indica il primo canale della tabella. Tutti i canali nella tabella devono essere nella medesima area dati. In altre parole, TB e TB+LR×NR-1 devono essere nella stessa area dati.

Caratteristiche operando

| Area | N | LR | NR | TB |
|--|-----------|--|----|----------------|
| Area CIO | --- | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | --- | Da W000 a W511 | | |
| Area bit di ritenività | --- | Da H000 a H511 | | |
| Area bit ausiliaria | --- | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 | | |
| Area del contatore | --- | Da C0000 a C4095 | | |
| Area DM | --- | Da D00000 a D32767 | | |
| Area EM senza banco | --- | Da E00000 a E32767 | | |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da 0 a 15 | Da #0001 a #FFFF (binario) o da &1 a &65.535 | | --- |
| Registri dati | --- | Da DR0 a DR15 | | --- |
| Registri indice | --- | --- | | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15, da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

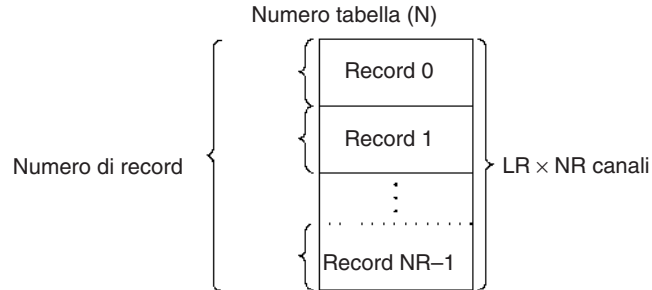
Descrizione

DIM(631) registra i canali da TB a TB+LR×NR-1 come numero di tabella N. Il numero di tabella N ha NR record e ogni record è lungo LR canali. Dopo che l'area è stata dichiarata come record, non è più possibile modificare i dati in tale area.

Utilizzare DIM(631) in combinazione con SETR(635) (SET RECORD NUMBER) o GETR(636) (GET RECORD NUMBER) per semplificare i calcoli

degli indirizzi nelle tabelle di dati. È possibile utilizzare l'istruzione DIM(631) per dividere i dati in record e l'istruzione SETR(635) per memorizzare l'indirizzo iniziale del record desiderato in un registro indice. Il registro indice può essere utilizzato come puntatore in altre istruzioni, come istruzioni di lettura, scrittura, ricerca o confronto.

Ad esempio, se alcune serie di valori quali le temperature o le pressioni vengono memorizzate come record e tali record vengono combinati in una tabella per vari scopi, è semplice leggere i valori impostati per ogni modello per ciascuna condizione particolare.



Le due istruzioni per tabelle di record associate a DIM(631) sono SETR(635) e GETR(636). SETR(635) imposta l'indirizzo iniziale della memoria del PLC del numero di record definito nel registro indice specificato. GETR(636) invia il numero di record del record che include il valore del registro indice specificato (indirizzo di memoria del PLC).

Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se LR o NR è 0000. OFF in tutti gli altri casi. |

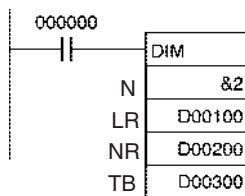
Avvertenze

I record in una tabella registrata sono identificati dai rispettivi numeri di record, compresi tra 0 e NR-1.

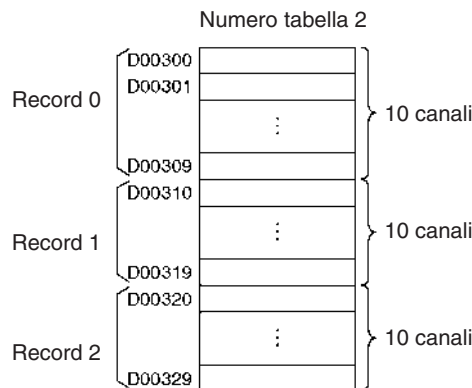
In base alla lunghezza record impostata (LR) e al numero di record (NR), è possibile che una singola tabella (da TB e TB+LR×NR-1) si sovrapponga a due aree dati. Prima di specificare una tabella che si sovrappone al contorno di un'area dati, verificare che non insorgano problemi.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, DIM(631) definisce il numero di tabella di record 2, con tre record a 10 canali. La tabella comincia a D00300.



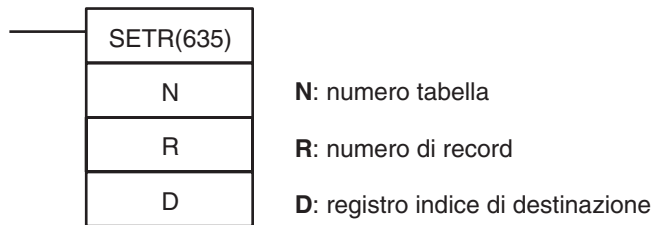
LR: D00100 0 0 0 A Lunghezza record: 10 canali
 NR: D00200 0 0 0 3 Numero di record: 3



3-17-6 SET RECORD LOCATION: SETR(635)

Scopo Scrive la posizione del record specificato (l'indirizzo di memoria del PLC relativo all'inizio del record) nel registro indice specificato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SETR(635) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SETR(635) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero tabella

Indica il numero della tabella. N deve essere compreso tra 0 e 15.

R: numero di record

Indica il numero di record del record desiderato. R deve essere compreso tra 0000 e FFFF esadecimale (da 0 a 65.534). I numeri dei record cominciano da 0, quindi i numeri di record validi sono compresi tra 0 e NR-1 per una tabella con NR record.

D: registro indice di destinazione

Indica il registro indice prescelto. D deve essere compreso tra IR0 e IR15.

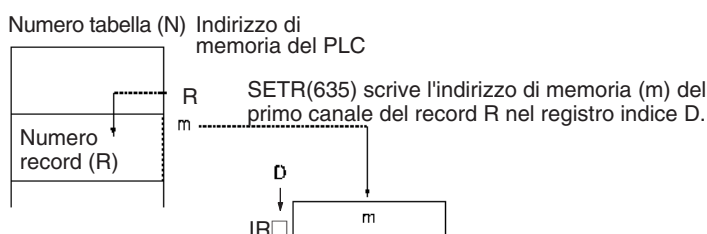
Caratteristiche operando

| Area | N | R | D |
|--------------------------------------|-----|--|-----|
| Area CIO | --- | Da CIO 0000 a CIO 6143 | --- |
| Area di lavoro | --- | Da W000 a W511 | --- |
| Area bit di ritenività | --- | Da H000 a H511 | --- |
| Area bit ausiliaria | --- | Da A000 a A959 | --- |
| Area del temporizzatore | --- | Da T0000 a T4095 | --- |
| Area del contatore | --- | Da C0000 a C4095 | --- |
| Area DM | --- | Da D00000 a D32767 | --- |
| Area EM senza banco | --- | Da E00000 a E32767 | --- |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 | --- |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 | --- |

| Area | N | R | D |
|--|-----------|--|---------------|
| Costanti | Da 0 a 15 | Da #0000 a #FFFE (binario) o da &0 a &65534 | --- |
| Registri dati | --- | Da DR0 a DR15 | --- |
| Registri indice | --- | | Da IR0 a IR15 |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | --- |

Descrizione

SETR(635) memorizza l'indirizzo della memoria del PLC del primo canale del record specificato nel registro indice specificato. Nello schema seguente è illustrato il funzionamento di base di SETR(635).



Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se il numero di tabella specificato (N) non è stato definito mediante DIM(631). ON se il numero di record specificato (R) supera il numero di record più elevato presente nella tabella (NR-1). OFF in tutti gli altri casi. |

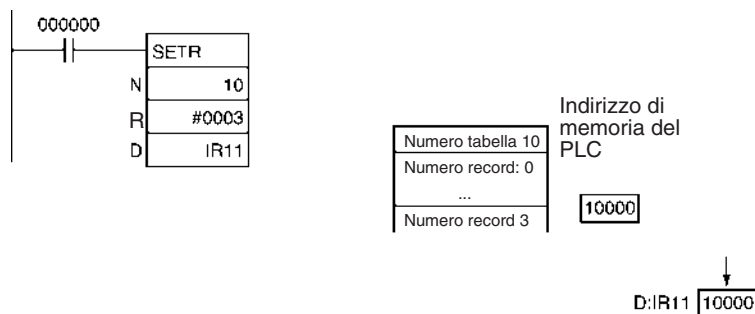
Avvertenze

È necessario che la tabella dei record sia stata definita in precedenza tramite l'istruzione DIM(631).

L'intervallo dei numeri di record validi è compreso tra 0 e NR-1, dove NR è il numero di record specificati quando è stata definita la tabella tramite l'istruzione DIM(631).

Esempi

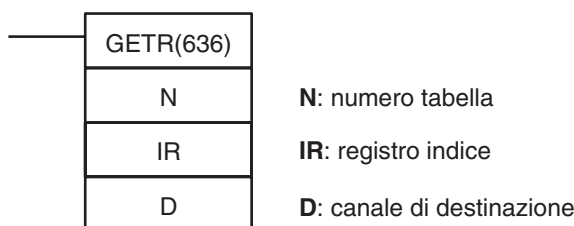
Nell'esempio seguente, quando CIO 000000 è impostato su ON, SETR(635) trova l'indirizzo di memoria del PLC del primo canale del record 3 del numero di tabella 10 e memorizza questo indirizzo nel registro indice IR11.



3-17-7 GET RECORD NUMBER: GETR(636)

Scopo Restituisce il numero del record all'indirizzo di memoria del PLC contenuto nel registro indice specificato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | GETR(636) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @GETR(636) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero tabella

Indica il numero della tabella. N deve essere compreso tra 0 e 15.

IR: registro indice

Indica il registro indice desiderato. D deve essere compreso tra IR0 e IR15.

D: canale di destinazione

Indica il canale in cui verrà scritto il numero del record.

Caratteristiche operando

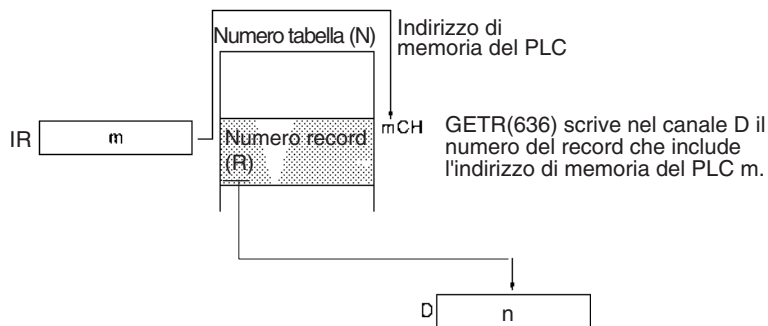
| Area | N | IR | D |
|--------------------------------------|-----------|-----|--|
| Area CIO | --- | | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | | Da W000 a W511 |
| Area bit di ritenività | --- | | Da H000 a H511 |
| Area bit ausiliaria | --- | | Da A448 a A959 |
| Area del temporizzatore | --- | | Da T0000 a T4095 |
| Area del contatore | --- | | Da C0000 a C4095 |
| Area DM | --- | | Da D00000 a D32767 |
| Area EM senza banco | --- | | Da E00000 a E32767 |
| Area EM con banco | --- | | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Da 0 a 15 | --- | --- |

| Area | N | IR | D |
|--|-----|---------------|--|
| Registri dati | --- | | Da DR0 a DR15 |
| Registri indice | --- | Da IR0 a IR15 | --- |
| Indirizzamento indiretto con i registri indice | --- | | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15, da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

GETR(636) trova il numero del record che include l'indirizzo di memoria del PLC contenuto nel registro indice specificato e scrive il numero del record in D. L'indirizzo di memoria del PLC contenuto nel registro indice non deve necessariamente essere il primo canale, ma può essere qualsiasi canale nel record.

Nello schema seguente è illustrato il funzionamento di base di GETR(636).



Flag

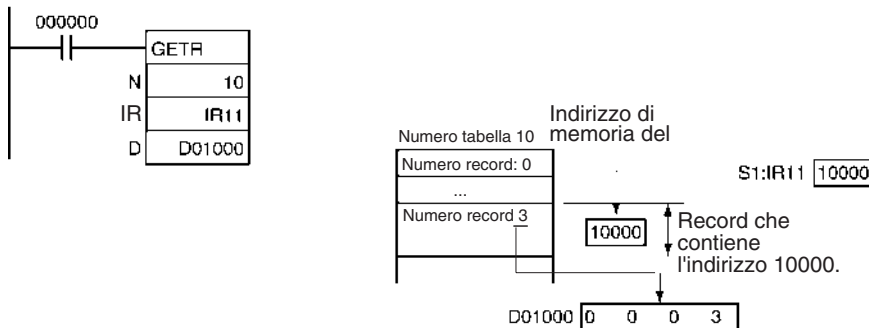
| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se l'indirizzo di memoria del PLC nel registro indice specificato non è incluso nella tabella specificata (N). ON se il numero di tabella specificato (N) non è stato definito mediante DIM(631). OFF in tutti gli altri casi. |

Avvertenze

La tabella dei record deve essere stata definita in precedenza tramite l'istruzione DIM(631) e l'indirizzo di memoria del PLC nel registro indice specificato deve essere incluso nella tabella specificata.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, GETR(636) trova il numero di record del record che contiene l'indirizzo di memoria del PLC nel registro indice IR11 e scrive tale numero in D01000.

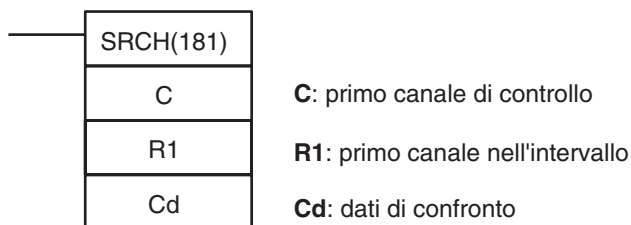


3-17-8 DATA SEARCH: SRCH(181)

Scopo

Cerca un canale di dati all'interno di un intervallo di canali. Nelle CPU CS1D per sistemi a singola CPU e nelle CPU CS1-H, CJ1-H e CJ1M questa istruzione può essere eseguita in background. Per ulteriori informazioni sull'esecuzione in background, fare riferimento al manuale *Controllori programmabili della serie CS/CJ - Manuale di programmazione*.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SRCH(181) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SRCH(181) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

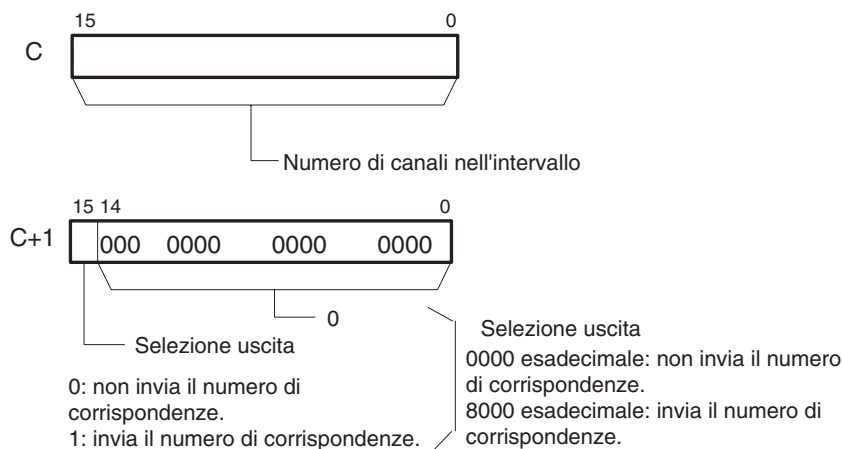
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C e C+1: canali di controllo

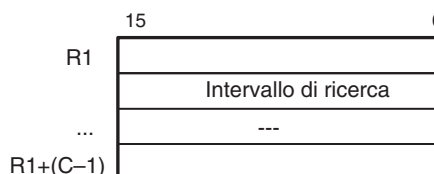
C specifica il numero di canali nell'intervallo e il bit 15 di C+1 indica se inviare o meno a DR00 il numero di corrispondenze.



Nota C e C+1 devono essere nella stessa area dati.

R1: primo canale nell'intervallo

R1 specifica il primo canale nell'intervallo di ricerca. Nei canali da R1 a R1+(C-1) viene effettuata la ricerca dei dati desiderati. C indica il numero di canali impostati in C.



Nota R1 e R1+C-1 devono essere nella stessa area dati.

Caratteristiche operando

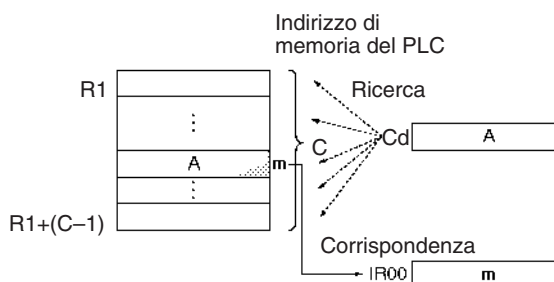
| Area | C | R1 | Cd |
|--|--|---------------------------------------|----------------------------|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A000 a A959 | |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | --- | Da #0000 a #FFFF (binario) |
| Registri dati | --- | | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SRCH(181) effettua la ricerca nell'intervallo di memoria da R1 a R1+C-1 per trovare i canali che contengono i dati di confronto (Cd). Se viene trovata una corrispondenza, SRCH(181) scrive in IR00 l'indirizzo di memoria del PLC del canale e attiva il flag di uguaglianza.

In presenza di due o più corrispondenze, in IR00 viene scritto solo l'indirizzo del primo canale che contiene i dati di confronto.

Quando il bit 15 di C+1 è stato impostato su 1, SRCH(181) scrive il numero di corrispondenze in DR00. Quando il bit 15 di C+1 è 0, DR00 non viene modificato.



SRCH(181) cerca i dati di tabella che contengono un canale in ciascun record. Per cercare i dati che contengono più di un canale per record, utilizzare DIM(631), SETR(635), GETR(636), FOR(512)–NEXT(513) o BREAK(514) insieme a un registro indice (IR).

Subito dopo l'esecuzione, è possibile controllare lo stato del flag di uguaglianza e stabilire se è stata trovata una corrispondenza.

Flag

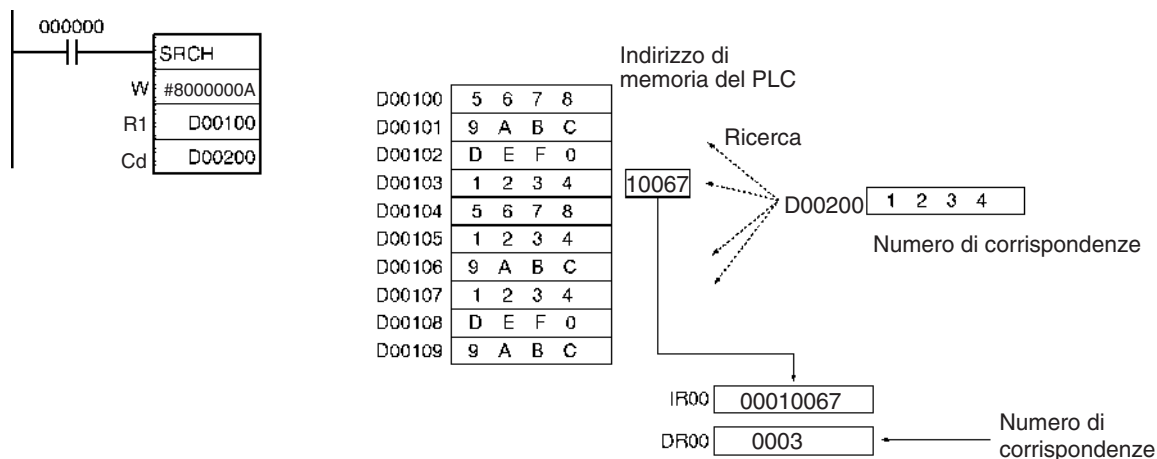
| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di C non è incluso nell'intervallo specificato 0001 - FFFF. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se uno o più canali indicati nell'intervallo di ricerca contengono i dati di confronto. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |

Avvertenze

Se non viene trovata nessuna corrispondenza, i contenuti di IR00 e DR00 restano invariati.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SRCH(181) cerca l'intervallo di 10 canali che comincia da D00100 per trovare i canali che hanno lo stesso contenuto di D00200. L'indirizzo di memoria del PLC del primo canale che contiene una corrispondenza viene scritto in IR00 e il numero totale di corrispondenze viene scritto in DR00.



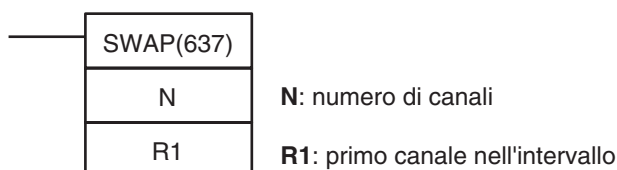
Se la lunghezza della tabella è specificata come &10 (10 decimale) o A esadecimale, il numero di corrispondenze non verrà inviato al registro dati DR00.

3-17-9 SWAP BYTES: SWAP(637)

Scopo

Scambia i byte più a sinistra e più a destra in tutti i canali dell'intervallo. Nelle CPU CS1D per sistemi a singola CPU e nelle CPU CS1-H, CJ1-H e CJ1M, questa istruzione può essere eseguita in background. Per ulteriori informazioni sull'esecuzione in background, fare riferimento al manuale *Controllori programmabili della serie CS/CJ - Manuale di programmazione*.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SWAP(637) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SWAP(637) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

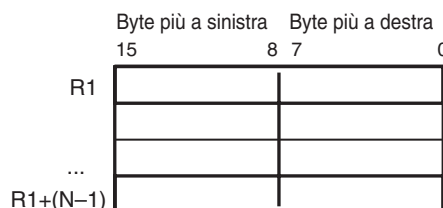
Operandi

N: Numero di canali

N specifica il numero di canali nell'intervallo e deve essere compreso tra 0001 e FFFF esadecimale (o tra &1 e &65.535).

R1: primo canale nell'intervallo

R1 specifica il primo canale nell'intervallo. R1 e R1+(N-1) devono essere nella stessa area dati.



Caratteristiche operando

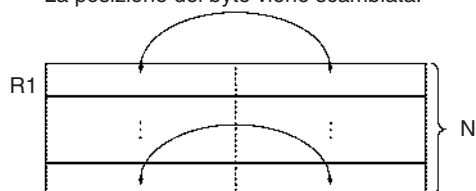
| Area | N | R1 |
|--------------------------------------|--|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |

| Area | N | R1 |
|--|--|-----|
| Costanti | Da #0001 a #FFFF (binario) o da &1 a &65.535 | --- |
| Registri dati | Da DR00 a DR15 | --- |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

SWAP(637) scambia la posizione dei due byte in tutti canali presenti nell'intervallo di memoria da R1 a R1+N-1. Questa istruzione può essere utilizzata per invertire l'ordine dei caratteri dei codici ASCII in ogni canale.

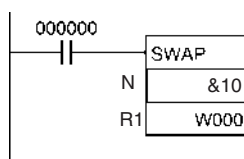
La posizione del byte viene scambiata.

**Flag**

| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se N è 0000. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l' <i>esecuzione in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SWAP(637) inverte i dati dei byte all'estrema sinistra con i dati dei byte all'estrema destra in ogni canale presente nell'intervallo di 10 canali da W000 a W009.



| | 15 | 8 | 7 | 0 |
|------|-----|-----|-----|-----|
| W000 | 4 | 1 | 4 | 2 |
| W001 | 4 | 3 | 4 | 4 |
| W002 | 4 | 5 | 4 | 6 |
| ... | ... | ... | ... | ... |
| W009 | 3 | 0 | 3 | 1 |

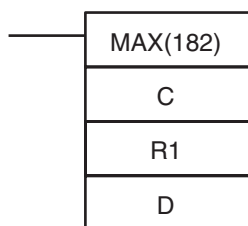
➔

| | 15 | 8 | 7 | 0 |
|------|-----|-----|-----|-----|
| W000 | 4 | 2 | 4 | 1 |
| W001 | 4 | 4 | 4 | 3 |
| W002 | 4 | 6 | 4 | 5 |
| ... | ... | ... | ... | ... |
| W009 | 3 | 1 | 3 | 0 |

3-17-10 FIND MAXIMUM: MAX(182)**Scopo**

Trova il valore massimo nell'intervallo.

Nelle CPU CS1D per sistemi a singola CPU e nelle CPU CS1-H, CJ1-H e CJ1M, questa istruzione può essere eseguita in background. Per ulteriori informazioni sull'esecuzione in background, fare riferimento al manuale *Controllori programmabili della serie CS/CJ - Manuale di programmazione*.

Simbolo programmazione ladder

C: primo canale di controllo

R1: primo canale nell'intervallo

D: canale di destinazione

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MAX(182) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MAX(182) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

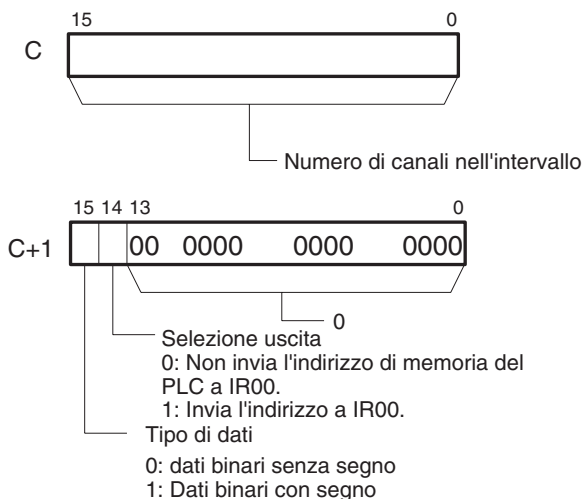
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi
C e C+1: canali di controllo

C specifica il numero di canali nell'intervallo, il bit 15 di C+1 indica se i dati verranno considerati come valori binari con segno o senza segno e il bit 14 di C+1 indica se inviare o meno a IR00 l'indirizzo di memoria del PLC del canale che contiene il valore massimo.

Nota C e C+1 devono essere nella stessa area dati.

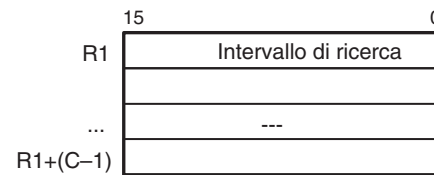


Nella tabella seguente vengono riportati i possibili valori di C.

| C+1 | Tipo di dati | Uscita registro indice |
|------|---------------------|------------------------|
| 0000 | Binario senza segno | No |
| 4000 | Binario senza segno | Sì |
| 8000 | Binario con segno | No |
| C000 | Binario con segno | Sì |

R1: primo canale nell'intervallo

R1 specifica il primo canale nell'intervallo di ricerca. Nei canali da R1 a R1+(C-1) viene effettuata la ricerca del valore massimo. C indica il numero di canali specificati in C.



Nota R1 e R1+(C-1) devono essere nella stessa area dati.

Caratteristiche operando

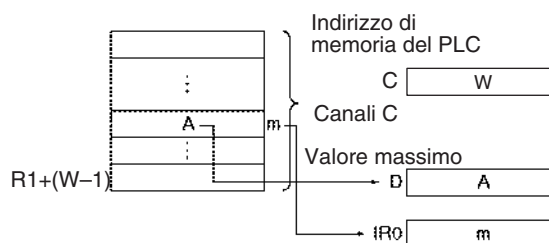
| Area | C | R1 | D |
|--|--|--|----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H510 | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A000 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | --- | |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

MAX(182) effettua la ricerca nell'intervallo di memoria da R1 a R1+C-1 per trovare il valore massimo nell'intervallo e invia tale valore a D.

Quando il bit 14 di C+1 è stato impostato su 1, MAX(182) scrive in IR00 l'indirizzo di memoria del PLC del canale contenente il valore massimo. Se due o più canali nell'intervallo contengono il valore massimo, verrà scritto in IR00 l'indirizzo del primo canale contenente tale valore.

Quando il bit 15 di C+1 è stato impostato su 1, MAX(182) interpreta i dati nell'intervallo come dati binari con segno.



Flag

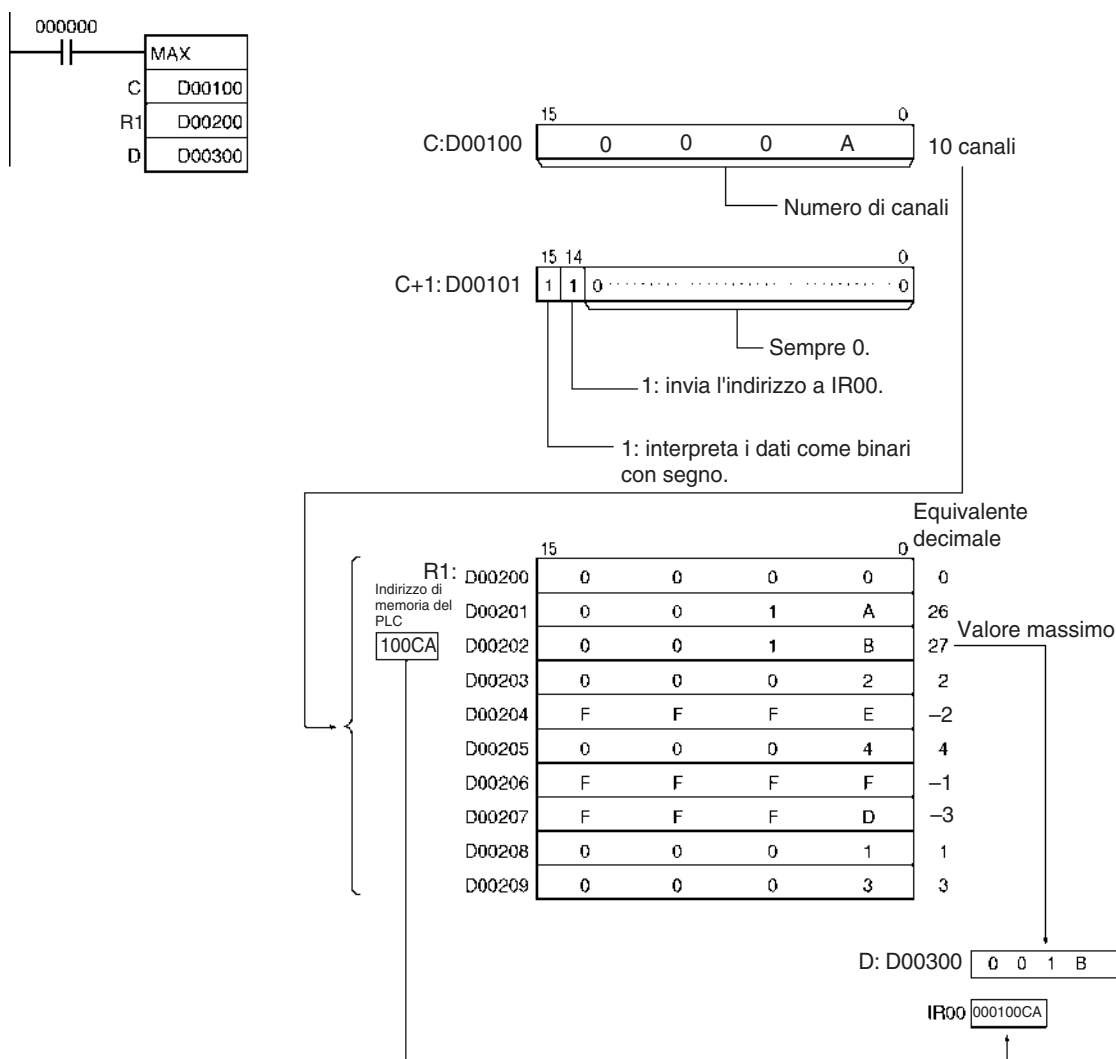
| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di C non è incluso nell'intervallo specificato 0001 - FFFF. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore massimo è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 è ON nel canale contenente il valore massimo. OFF in tutti gli altri casi. |

Avvertenze

Quando il bit 15 di C+1 è stato impostato su 1, i dati all'interno dell'intervallo verranno interpretati come dati binari con segno e i valori esadecimali da 8000 a FFFF verranno considerati negativi. Quindi i risultati della ricerca possono differire in base all'impostazione del tipo di dati.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, MAX(182) cerca l'intervallo di 10 canali che comincia da D00200 per trovare il valore massimo. Il valore massimo viene scritto in D00300 e l'indirizzo di memoria del PLC del canale contenente il valore massimo viene scritto in IR00.

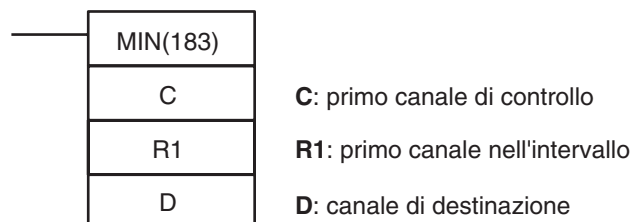


3-17-11 FIND MINIMUM: MIN(183)

Scopo

Trova il valore minimo nell'intervallo. Nelle CPU CS1D per sistemi a singola CPU e nelle CPU CS1-H, CJ1-H e CJ1M, questa istruzione può essere eseguita in background. Per ulteriori informazioni sull'esecuzione in background, fare riferimento al manuale *Controllori programmabili della serie CS/CJ - Manuale di programmazione*.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MIN(183) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MIN(183) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

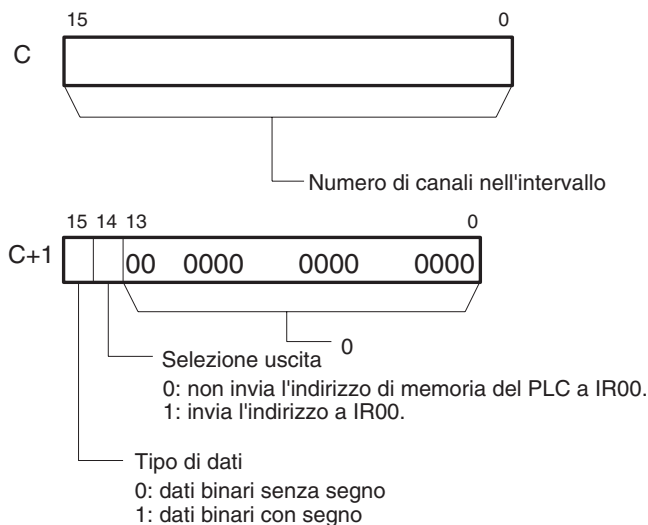
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C e C+1: canali di controllo

C specifica il numero di canali nell'intervallo, il bit 15 di C+1 indica se i dati verranno considerati come valori binari con segno o senza segno e il bit 14 di C+1 indica se inviare o meno a IR00 l'indirizzo di memoria del PLC del canale che contiene il valore minimo.

Nota C e C+1 devono essere nella stessa area dati.

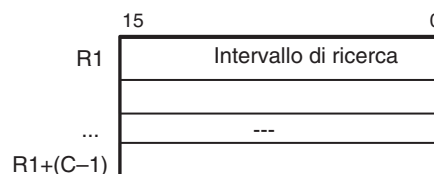


Nella tabella seguente vengono riportati i possibili valori di C.

| C+1 | Tipo di dati | Uscita registro indice |
|------|---------------------|------------------------|
| 0000 | Binario senza segno | No |
| 4000 | Binario senza segno | Sì |
| 8000 | Binario con segno | No |
| C000 | Binario con segno | Sì |

R1: primo canale nell'intervallo

R1 specifica il primo canale nell'intervallo di ricerca. Nei canali da R1 a R1+(C-1) viene effettuata la ricerca del valore minimo. C indica il numero di canali specificati in C.



Nota R1 e R1+C-1 devono essere nella stessa area dati.

Caratteristiche operando

| Area | C | R1 | D |
|-------------------------|------------------------|------------------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A000 a A959 | Da A448 a A959 |

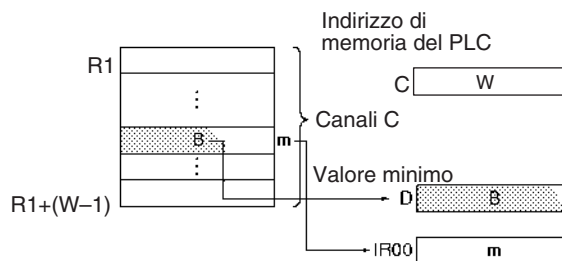
| Area | C | R1 | D |
|--|--|--|---|
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D0000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | --- | |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

MIN(183) effettua la ricerca nell'intervallo di memoria da R1 a R1+C-1 per trovare il valore minimo nell'intervallo e invia tale valore a D.

Quando il bit 14 di C+1 è stato impostato su 1, MIN(183) scrive in IR00 l'indirizzo di memoria del PLC del canale contenente il valore minimo. Se due o più canali nell'intervallo contengono il valore minimo, verrà scritto in IR00 l'indirizzo del primo canale contenente tale valore.

Quando il bit 15 di C+1 è stato impostato su 1, MIN(183) interpreta i dati nell'intervallo come dati binari con segno.



Flag

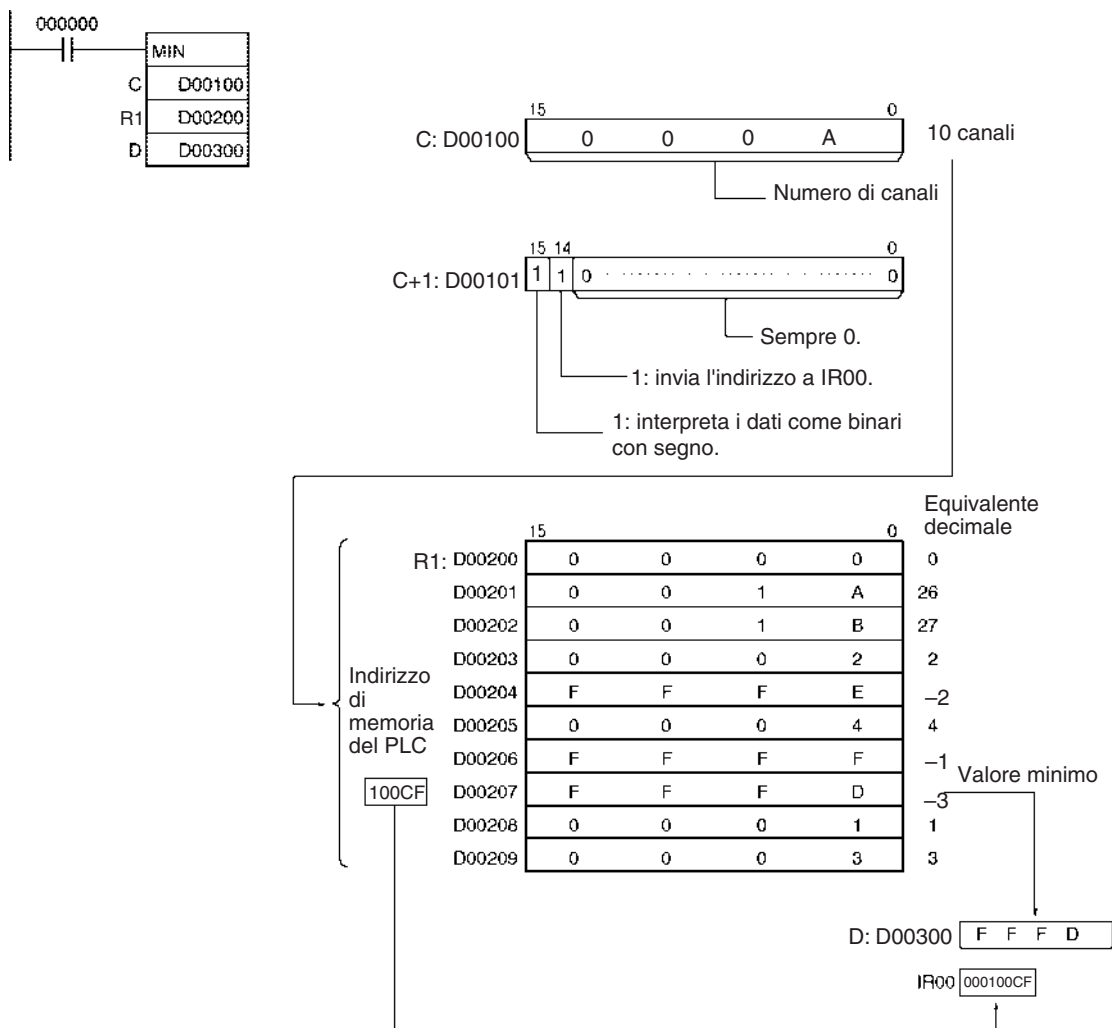
| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di C non è incluso nell'intervallo specificato 0001 - FFFF. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore minimo è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 è ON nel canale contenente il valore minimo. OFF in tutti gli altri casi. |

Avvertenze

Quando il bit 15 di C+1 è stato impostato su 1, i dati all'interno dell'intervallo verranno interpretati come dati binari con segno e i valori esadecimali da 8000 a FFFF verranno considerati negativi. Quindi i risultati della ricerca possono differire in base all'impostazione del tipo di dati.

Esempi

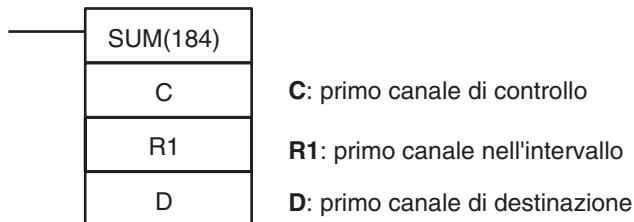
Nell'esempio seguente, quando CIO 000000 è impostato su ON, MIN(183) cerca l'intervallo di 10 canali che comincia da D00200 per trovare il valore minimo. Il valore minimo viene scritto in D00300 e l'indirizzo di memoria del PLC del canale contenente il valore minimo viene scritto in IR00.



3-17-12 SUM: SUM(184)

Scopo Aggiunge i byte o i canali nell'intervallo e invia il risultato a due canali.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SUM(184) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SUM(184) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

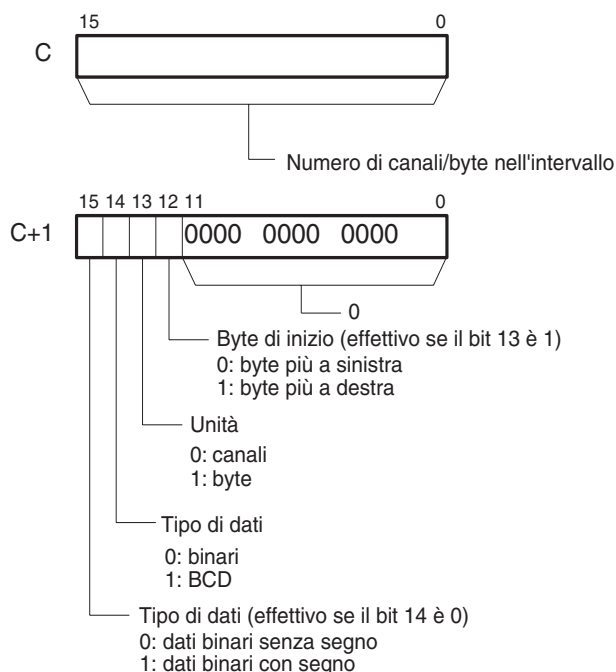
| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

C e C+1: canali di controllo

C specifica il numero di unità (byte o canali) da sommare. Il bit 13 di C+1 determina se verranno sommati byte o canali.

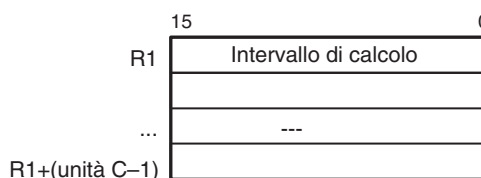
I bit da 12 a 15 di C+1 indicano il tipo di dati che verrà sommato, come mostrato nello schema seguente.



Nota C e C+1 devono essere nella stessa area dati.

R1: primo canale nell'intervallo

R1 specifica il primo canale nell'intervallo. La lunghezza dell'intervallo dipende dal numero di unità e dal byte di inizio, se vengono addizionati byte.



Nota Tutti i canali nell'intervallo di calcolo devono essere nella medesima area dati.

D: primo canale di destinazione

Il risultato del calcolo viene inviato a D+1 e a D. Le quattro cifre all'estrema sinistra verranno memorizzate in D+1, mentre le quattro cifre all'estrema destra verranno memorizzate in D.

Caratteristiche operando

| Area | C | R1 | D |
|-------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 | Da W000 a W510 |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H511 | Da H000 a H510 |
| Area bit ausiliaria | Da A000 ad A958 | Da A000 a A959 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 | Da T0000 a T4094 |

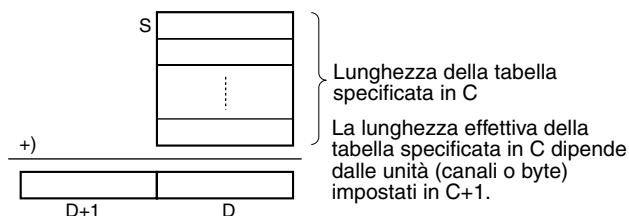
| Area | C | R1 | D |
|--|--|--|--|
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 | Da C0000 a C4094 |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | --- | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SUM(184) aggiunge le unità C dei dati che cominciano con i dati in R1 e invia il risultato a D+1 e a D. Le impostazioni in C+1 determinano se le unità sono canali o byte, se i dati sono binari (con segno o senza segno) o BCD e, se vengono addizionati byte, se iniziare con il byte più a sinistra o più a destra di R1.

Quando il bit 14 di C+1 è stato impostato su 0, SUM(184) interpreta i dati come binari. In questo caso, il bit 15 determina se i dati sono con segno (bit 15 = 1) o senza segno (bit 15 = 0).

Quando il bit 13 di C+1 è stato impostato su 1, SUM(184) aggiunge byte di dati. In questo caso, il bit 12 determina se il calcolo debba iniziare con il byte all'estrema destra di R1 (bit 12 = 1) o con il byte all'estrema sinistra di R1 (bit 12 = 0).

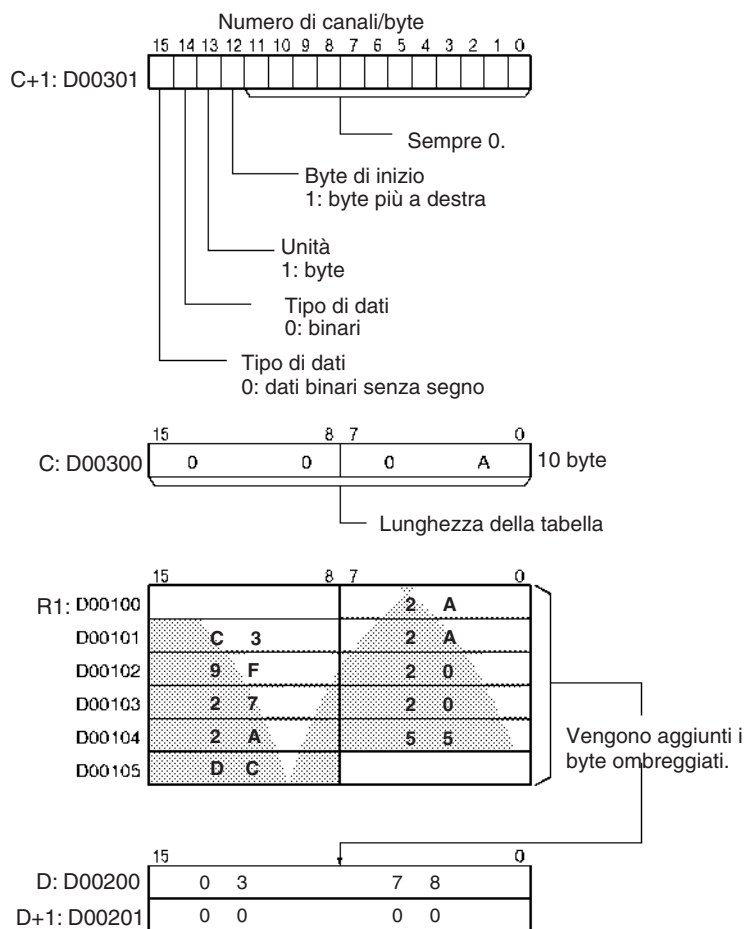
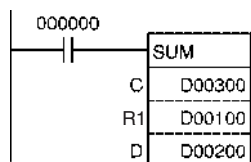


Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di C non è incluso nell'intervallo specificato 0001 - FFFF. ON se sono stati specificati dati BCD ma l'intervallo contiene dati binari. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l' <i>esecuzione in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |
| Flag negativo | N | ON se il bit 15 è ON nel risultato. OFF in tutti gli altri casi. |

Esempi

Nell'esempio seguente, quando CIO 000000 è ON, SUM(184) aggiunge 10 byte di dati binari con segno, cominciando dal byte più a destra di D00100, e scrive il risultato in D00201 e D00200.

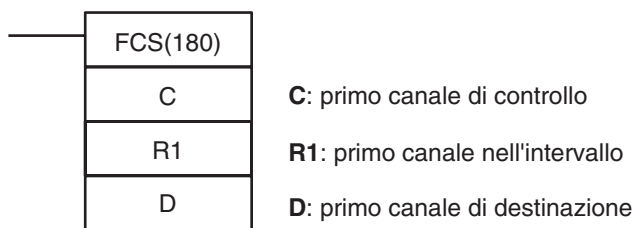


3-17-13 FRAME CHECKSUM: FCS(180)

Scopo

Calcola il valore FCS per l'intervallo specificato e invia il risultato in formato ASCII.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FCS(180) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FCS(180) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

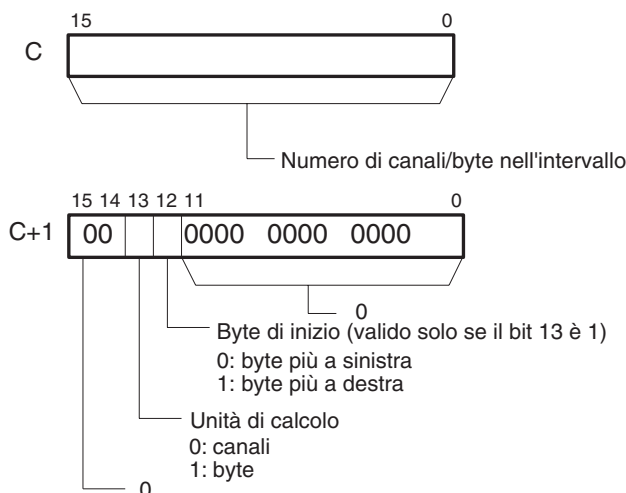
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C e C+1: canali di controllo

C specifica il numero di unità (byte o canali) da utilizzare nel calcolo FCS. Il bit 13 di C+1 determina se verranno utilizzati byte o canali.

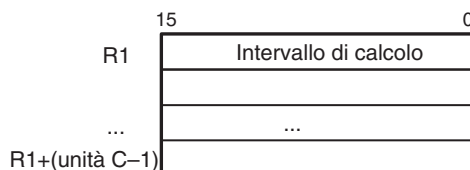
Quando il bit 13 di C+1 è stato impostato su 1, FCS(180) calcola il valore FCS per byte di dati. In questo caso, il bit 12 determina se il calcolo debba iniziare con il byte all'estrema destra di R1 (bit 12 = 1) o con il byte all'estrema sinistra di R1 (bit 12 = 0).



Nota C e C+1 devono essere nella stessa area dati.

R1: primo canale nell'intervallo

R1 specifica il primo canale nell'intervallo. La lunghezza dell'intervallo dipende dal numero di unità e dal byte di inizio, se nel calcolo vengono utilizzati byte.



Nota Tutti i canali nell'intervallo di calcolo devono essere nella medesima area dati.

D: primo canale di destinazione

Se sono stati selezionati byte, il risultato del calcolo viene inviato a D.

Se sono stati selezionati canali, il risultato del calcolo viene inviato a D+1 e a D. In questo caso, le quattro cifre all'estrema sinistra verranno memorizzate in D+1, mentre le quattro cifre all'estrema destra verranno memorizzate in D.

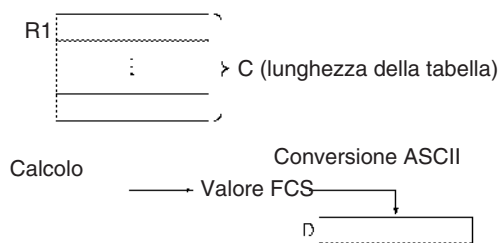
Caratteristiche operando

| Area | C | R1 | D |
|--|--|---|----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W510 | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H510 | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A000 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4094 | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4094 | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32766 | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32766 | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | Da En_0000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | --- | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

FCS(180) calcola il valore FCS per unità C di dati, cominciando dai dati in R1, converte il valore in codice ASCII e invia il risultato a D (per byte) o a D+1 e D (per canali). Le impostazioni in C+1 determinano se le unità sono canali o byte, se i dati sono binari (con segno o senza segno) o BCD e, se vengono addizionati byte, se iniziare con il byte più a sinistra o più a destra di R1.

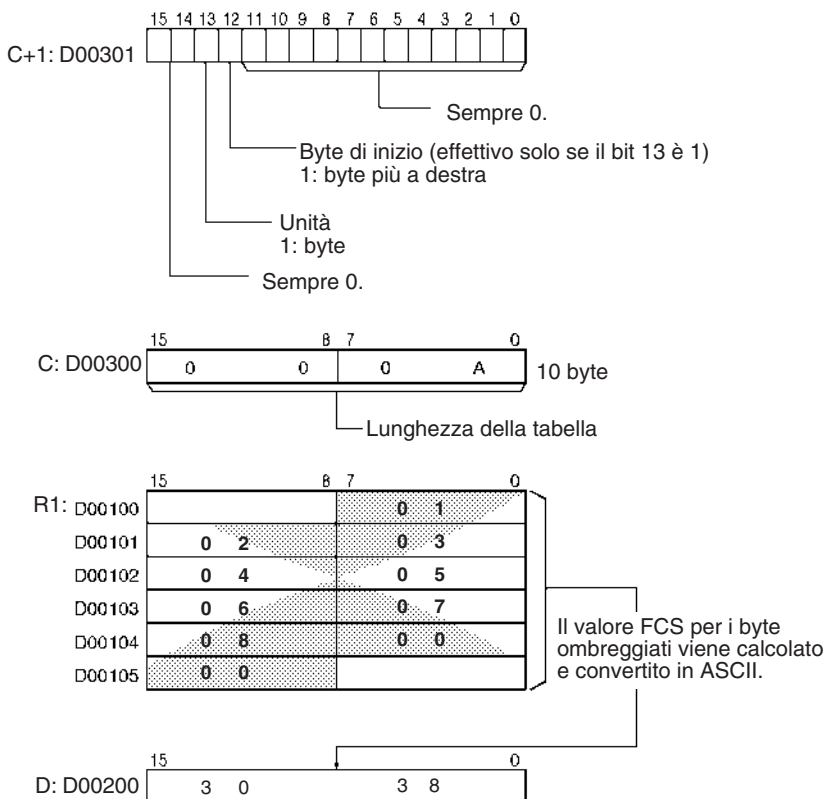
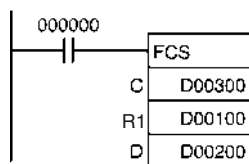
Quando il bit 13 di C+1 è stato impostato su 1, FCS(180) elabora byte di dati. In questo caso, il bit 12 determina se il calcolo debba iniziare con il byte all'estrema destra di R1 (bit 12 = 1) o con il byte all'estrema sinistra di R1 (bit 12 = 0).

**Flag**

| Nome | Eti-chetta | Operazione |
|----------------|------------|--|
| Flag di errore | ER | ON se il contenuto di C non è incluso nell'intervallo specificato 0001 - FFFF. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l' <i>esecuzione in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |

Esempi

Nell'esempio seguente, quando CIO 000000 è ON, FCS(180) calcola il valore FCS per i 10 byte di dati, cominciando dal byte più a destra di D00100, e scrive il risultato in D00200.

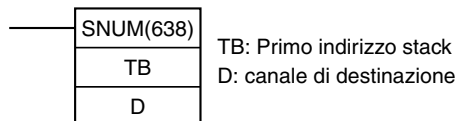


3-17-14 STACK SIZE READ: SNUM(638)

Scopo

Conta la quantità di dati di stack (numero di canali) nello stack specificato. Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SNUM(638) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SNUM(638) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

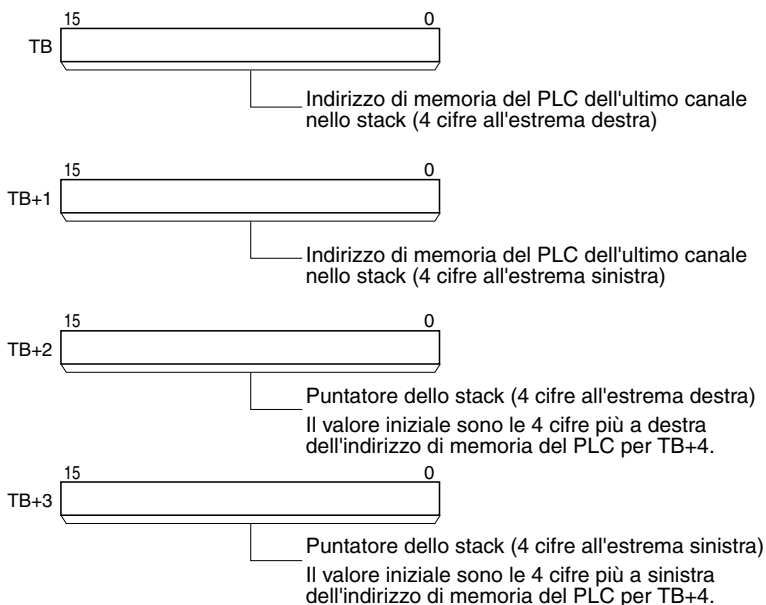
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

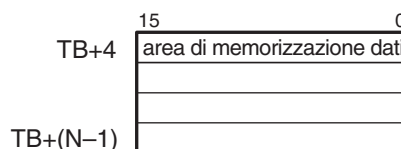
Operandi

Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo disponibile nello stack.

**Da TB+4 a TB+(N-1): area di memorizzazione dati**

Il resto dello stack viene usato per memorizzare i dati.



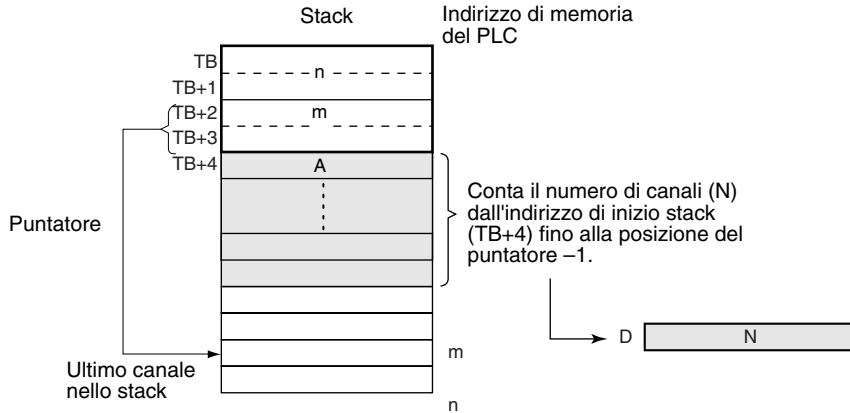
Caratteristiche operando

| Area | TB | D |
|--------------------------------------|--|---------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | Da DR0 a DR15 |

| Area | TB | D |
|--|---|---|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 | |
| | Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 | |
| | Da DR0 a DR15, da IR0 a IR15 | |
| | Da ,IR0+(++) a ,IR15+(++) | |
| | Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

SNUM(638) conta il numero di canali di dati nello stack specificato dall'inizio dell'area dati TB+4 fino all'indirizzo che precede quello indicato dal puntatore dello stack (TB+3 e TB+2). SNUM(638) non modifica i dati nello stack o il puntatore dello stack.



Flag

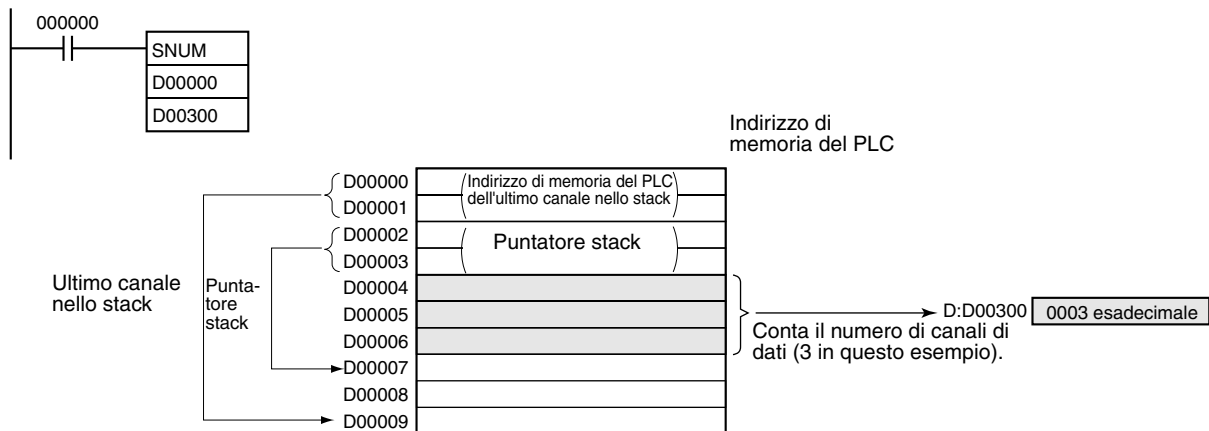
| Nome | Eti-chetta | Operazione |
|----------------|------------|--|
| Flag di errore | ER | ON se il numero di canali di dati nello stack (il valore inviato a D) è 0. OFF in tutti gli altri casi. |

Avvertenze

È necessario che lo stack sia stato definito in precedenza tramite l'istruzione SSET(630).

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SNUM(638) conta il numero di canali dall'inizio dell'area dati a D00004 fino alla posizione del puntatore dello stack -1 (D00006) e invia il risultato a D00300. In questo caso il puntatore dello stack indica D00007.



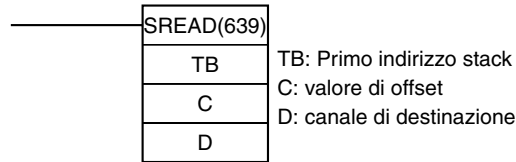
3-17-15 STACK DATA READ: SREAD(639)

Scopo

Legge i dati dall'elemento specificato dello stack. Il valore di offset (numero di elementi prima della posizione corrente del puntatore) indica la posizione dell'elemento desiderato.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SREAD(639) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SREAD(639) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

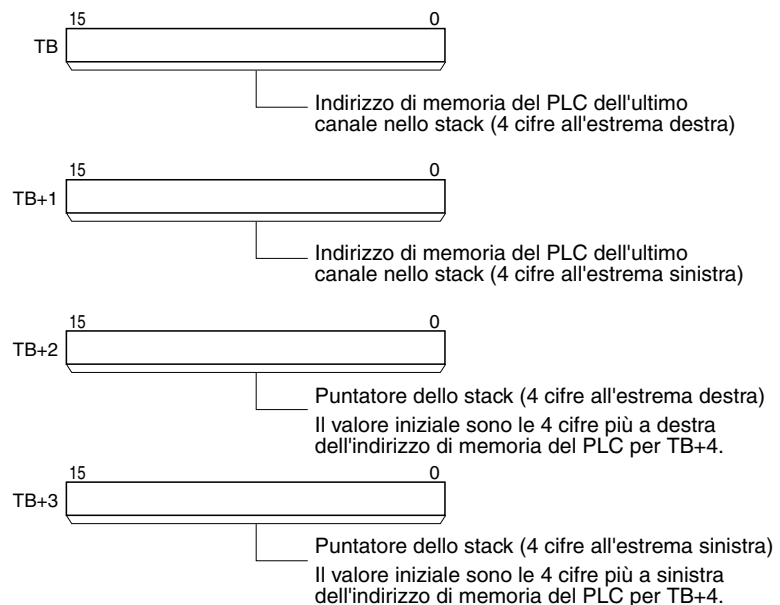
Aree di programma applicabili

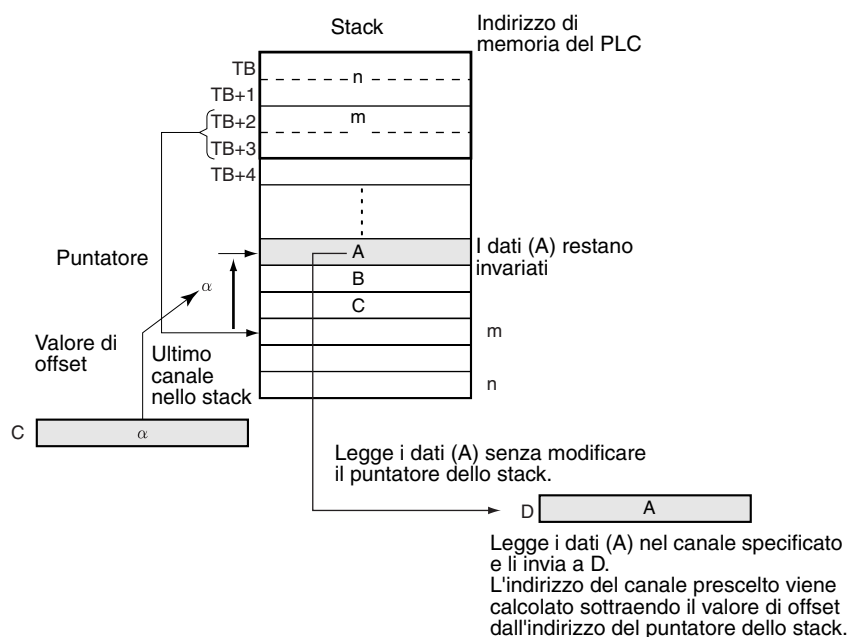
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo disponibile nello stack.





È possibile utilizzare SREAD(639) per leggere i dati relativi a un elemento che si trova attualmente su un nastro trasportatore. La posizione dell'elemento desiderato è semplicemente il numero di elementi restituiti (il valore di offset) dall'elemento più recente aggiunto al nastro trasportatore.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|---|
| Flag di errore | ER | ON se la posizione di lettura specificata non è inclusa nell'area dello stack. ON se il valore di offset specificato in C è 0 o maggiore della dimensione massima dell'area dati (FFFB esadecimale). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se i dati inviati a D sono 0000. OFF in tutti gli altri casi. |

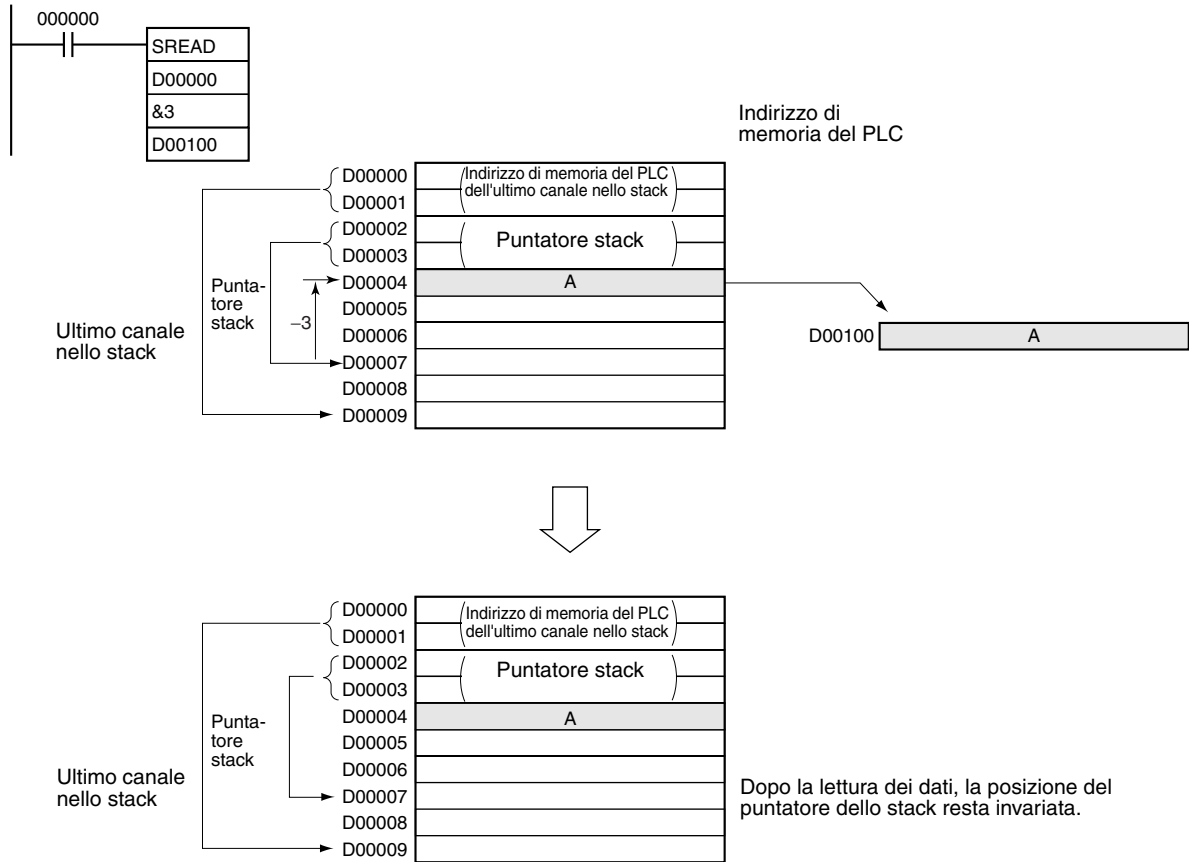
Avvertenze

È necessario che lo stack sia stato definito in precedenza tramite l'istruzione SSET(630).

L'indirizzo nel puntatore dello stack deve essere maggiore dell'indirizzo di memoria del PLC dell'inizio dell'area dati (TB+4). Se il puntatore dello stack è minore dell'indirizzo di memoria del PLC di TB+4 si verificherà un errore, ovvero verrà generato un errore di underflow dello stack.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SREAD(639) legge i dati nel canale specificato nello stack cominciando da D00000 e invia tali dati a D00100. In questo caso, il puntatore dello stack indica D00007 e il valore di offset è 3, quindi i dati verranno letti da D00004.



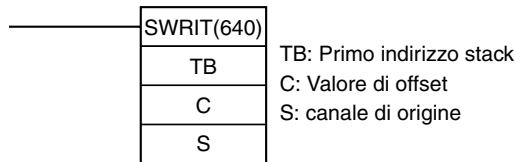
3-17-16 STACK DATA OVERWRITE: SWRIT(640)

Scopo

Scrive i dati di origine nell'elemento specificato dello stack, sovrascrivendo i dati già esistenti. Il valore di offset (numero di elementi prima della posizione corrente del puntatore) indica la posizione dell'elemento desiderato.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SWRIT(640) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SWRIT(640) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

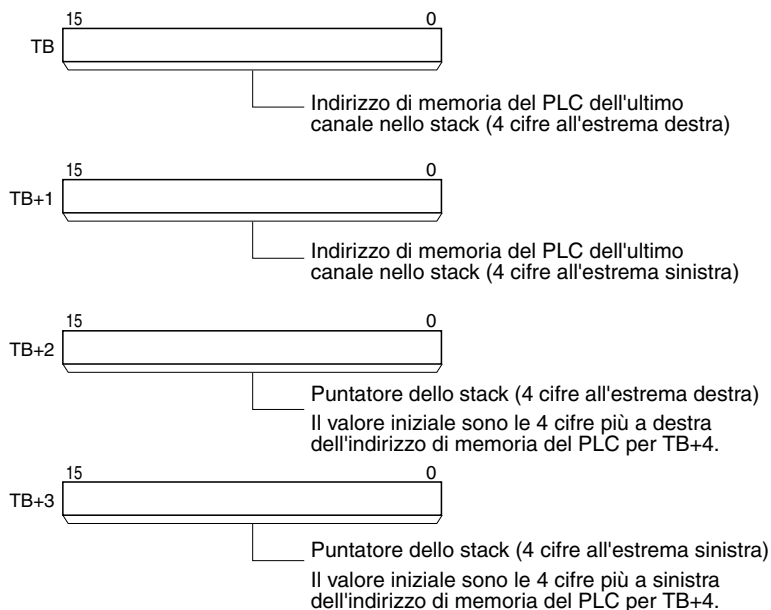
Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

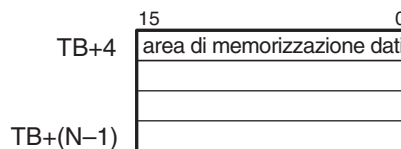
Operandi

Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo disponibile nello stack.

**Da TB+4 a TB+(N-1): area di memorizzazione dati**

Il resto dello stack viene usato per memorizzare i dati.



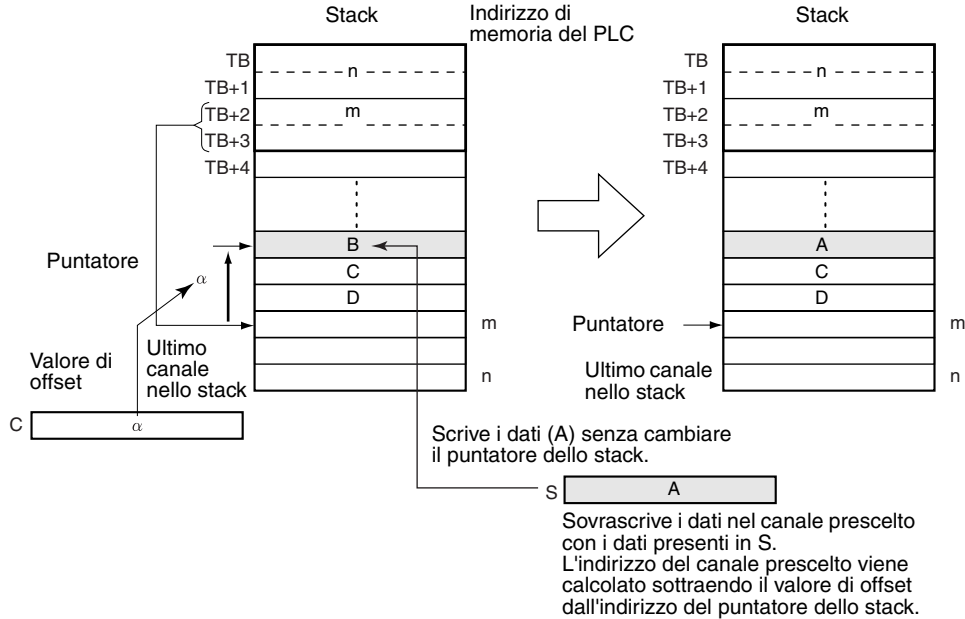
Caratteristiche operando

| Area | TB | C | S |
|--------------------------------------|--|-----------------------------------|-----------------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A448 a A959 | Da A000 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0001 a #FFFF (esadecimale) | Da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 | |

| Area | TB | C | S |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SWRIT(640) sovrascrive i dati nel canale prescelto con i dati specificati in S. La posizione del canale prescelto viene calcolata sottraendo il valore di offset in C dal puntatore dello stack (TB+3 e TB+4). SWRIT(640) non modifica il puntatore dello stack.



È possibile utilizzare SWRIT(640) per cambiare i dati relativi a un elemento che si trova attualmente su un nastro trasportatore. La posizione dell'elemento desiderato è semplicemente il numero di elementi restituiti (il valore di offset) dall'elemento più recente aggiunto al nastro trasportatore.

Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|---|
| Flag di errore | ER | ON se la posizione di scrittura specificata non è inclusa nell'area dello stack. ON se il valore di offset specificato in C è 0 o maggiore della dimensione massima dell'area dati (FFFB esadecimale). OFF in tutti gli altri casi. |

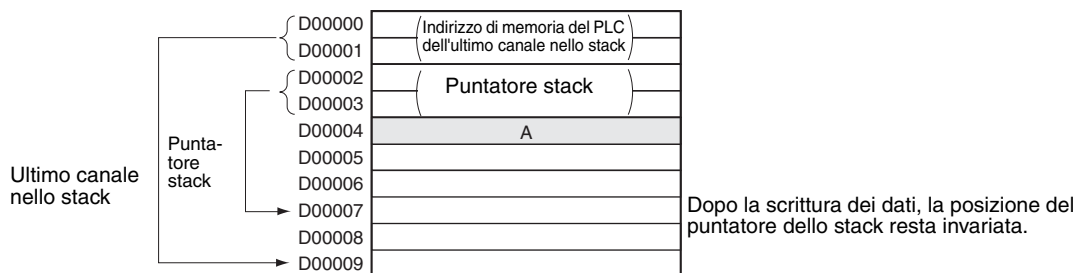
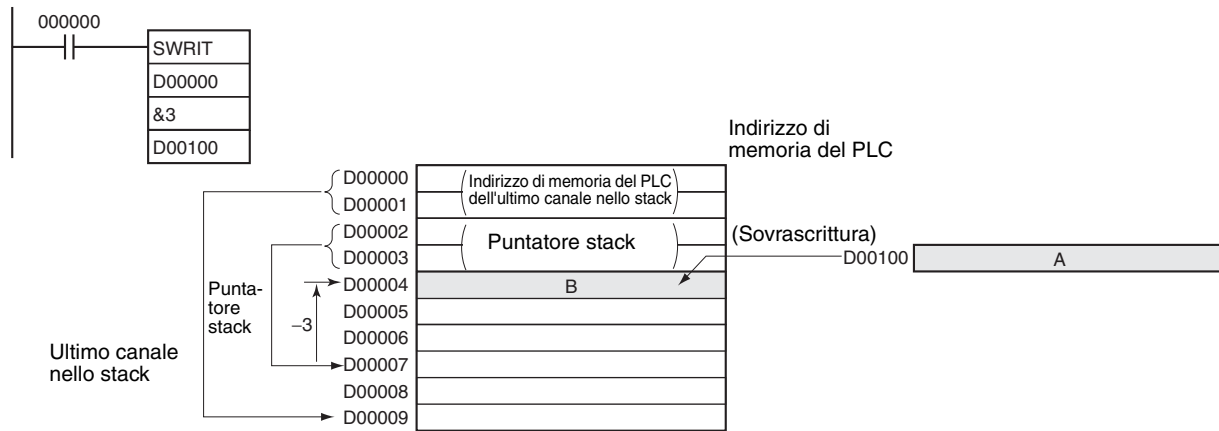
Avvertenze

È necessario che lo stack sia stato definito in precedenza tramite l'istruzione SSET(630).

L'indirizzo nel puntatore dello stack deve essere maggiore dell'indirizzo di memoria del PLC dell'inizio dell'area dati (TB+4). Se il puntatore dello stack è minore dell'indirizzo di memoria del PLC di TB+4 si verificherà un errore, ovvero verrà generato un errore di underflow dello stack.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SWRIT(640) scrive i dati in D00100 nel canale specificato nello stack cominciando da D00000. In questo caso, il puntatore dello stack indica D00007 e il valore di offset è 3, quindi i dati in D00004 verranno sovrascritti.



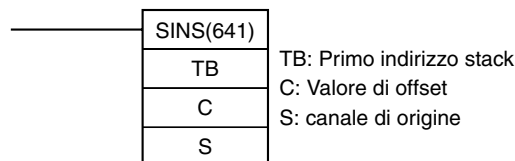
3-17-17 STACK DATA INSERT: SINS(641)

Scopo

Inserisce i dati di origine nella posizione specificata nello stack e fa scorrere verso il basso i dati rimanenti dello stack. Il valore di offset (numero di elementi prima della posizione corrente del puntatore) indica la posizione dell'elemento desiderato.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SINS(641) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SINS(641) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

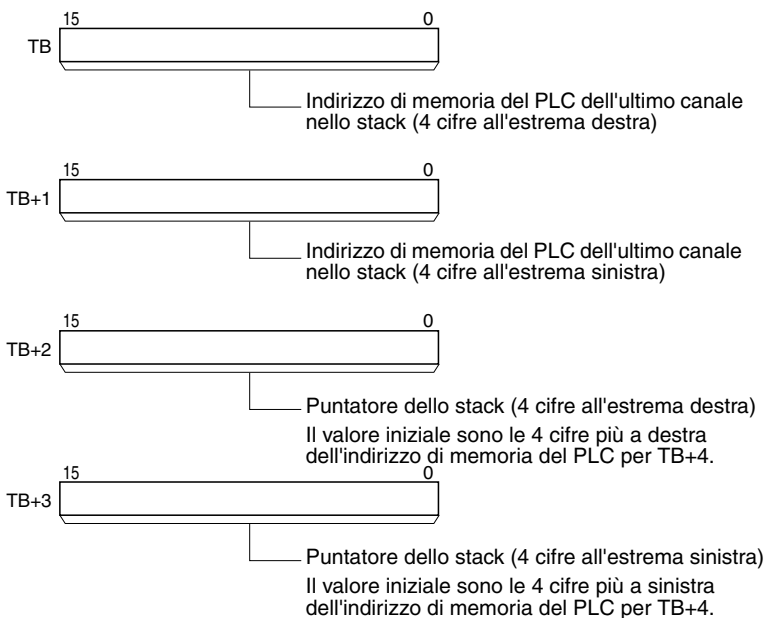
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

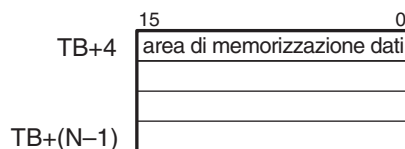
Operandi

Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo disponibile nello stack.

**Da TB+4 a TB+(N-1): area di memorizzazione dati**

Il resto dello stack viene usato per memorizzare i dati.



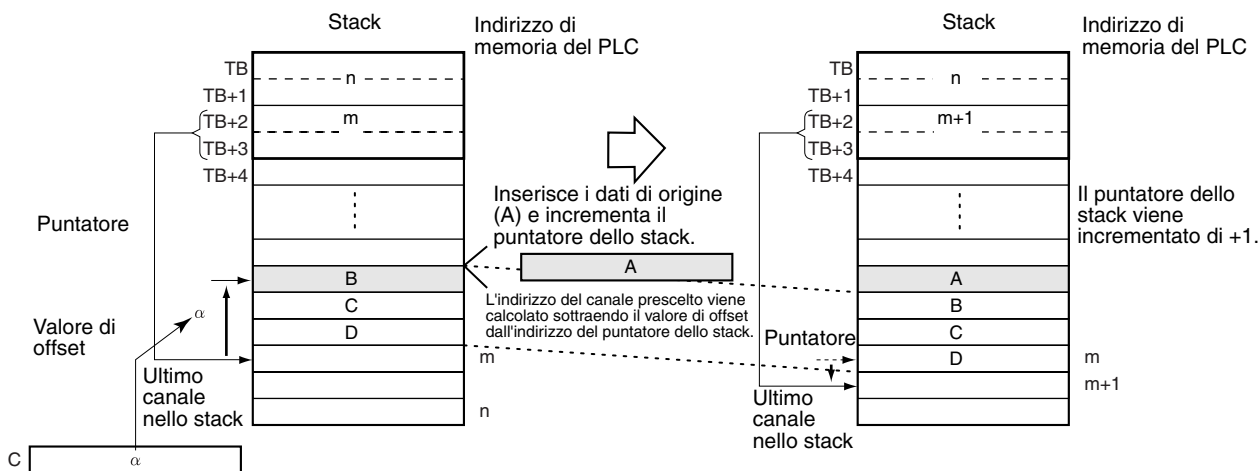
Caratteristiche operando

| Area | TB | C | S |
|--------------------------------------|--|----------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A448 a A959 | Da A000 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |

| Area | TB | C | S |
|--|--|--------------------------------|--------------------------------|
| Costanti | --- | Da #0001 a #FFFB (esadecimale) | Da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SINS(641) inserisce i dati di origine all'indirizzo desiderato e sposta i dati esistenti di un canale verso il basso. Contemporaneamente SINS(641) incrementa di 1 il puntatore dello stack (TB+3 e TB+2). La posizione dell'indirizzo prescelto viene calcolata sottraendo il valore di offset in C dal puntatore dello stack.



È possibile utilizzare SINS(641) per inserire i dati relativi a un elemento che viene inserito tra elementi già presenti su un nastro trasportatore. La posizione del punto di inserimento è semplicemente il numero di elementi restituiti (il valore di offset) dall'elemento più recente aggiunto al nastro trasportatore.

Flag

| Nome | Eti-chetta | Operazione |
|----------------|------------|--|
| Flag di errore | ER | ON se l'indirizzo indicato dal puntatore dello stack (TB+3 e TB+2) è maggiore dell'indirizzo di memoria del PLC dell'ultimo canale presente nell'area dati dello stack. (Errore di overflow dello stack). ON se il valore di offset specificato è maggiore della dimensione massima dell'area dati -1 (FFFA esadecimale). OFF in tutti gli altri casi. |

Avvertenze

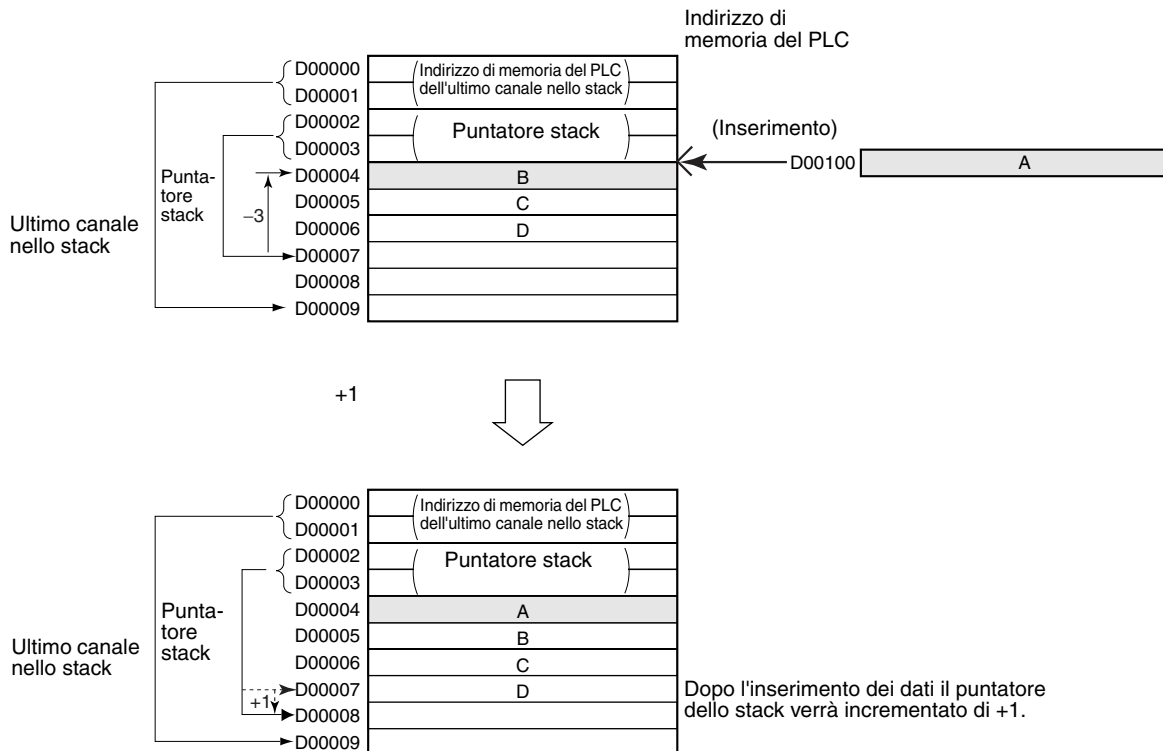
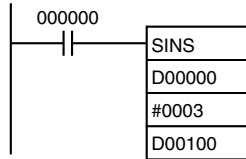
È necessario che lo stack sia stato definito in precedenza tramite l'istruzione SSET(630).

SINS(641) inserisce un canale di dati nello stack, pertanto deve esserci almeno un canale disponibile alla fine dello stack. Se lo stack è pieno, si verificherà un errore e i dati di origine non verranno inseriti.

Se quando l'istruzione SINS(641) viene eseguita l'indirizzo indicato nel puntatore dello stack (TB+3 e TB+2) è già maggiore dell'indirizzo dell'ultimo canale nello stack (TB+1 e TB), si verificherà un errore di overflow dello stack e i dati di origine non verranno inseriti.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SINS(641) inserisce i dati di origine in D00100 nell'indirizzo specificato nello stack cominciando da D00000. In questo caso, il puntatore dello stack indica D00007 e il valore di offset è 3, quindi i dati di origine verranno inseriti in D00004. I dati esistenti verranno spostati di un canale verso il basso e i dati in D00007 verranno sovrascritti. Allo stesso tempo, il puntatore dello stack verrà incrementato da D00007 a D00008.

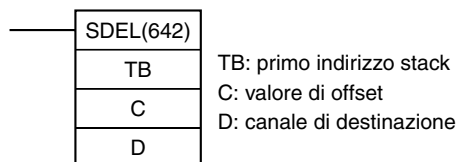


3-17-18 STACK DATA DELETE: SDEL(642)

Scopo

Elimina l'elemento di dati che si trova nella posizione specificata nello stack, invia i dati al canale di destinazione specificato e fa scorrere verso l'alto nello stack i dati rimanenti. Il valore di offset (numero di elementi prima della posizione corrente del puntatore) indica la posizione dell'elemento desiderato. Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SDEL(642) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SDEL(642) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

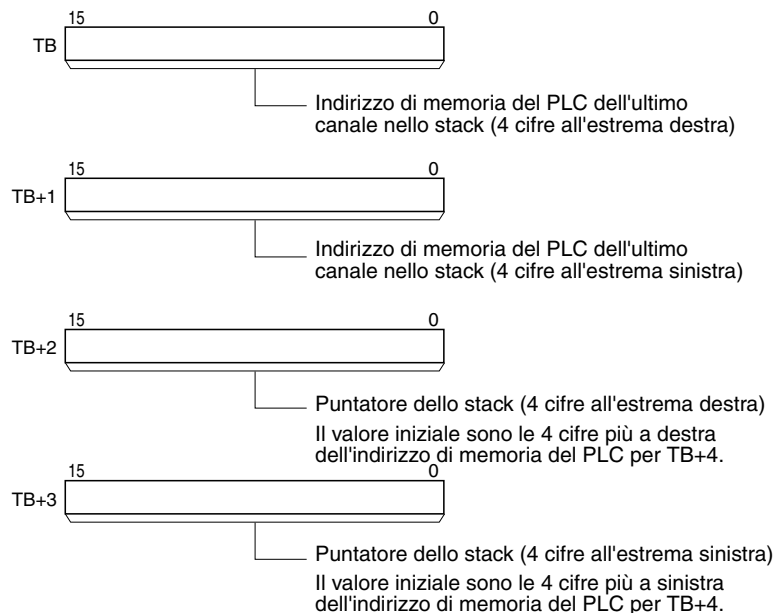
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

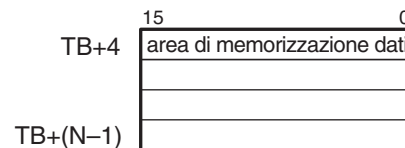
Da TB a TB+3: canali di controllo stack

I primi quattro canali dello stack contengono l'indirizzo di memoria del PLC relativo all'ultimo canale dello stack e il puntatore dello stack, ossia l'indirizzo di memoria del PLC del canale successivo disponibile nello stack.



Da TB+4 a TB+(N-1): area di memorizzazione dati

Il resto dello stack viene usato per memorizzare i dati.



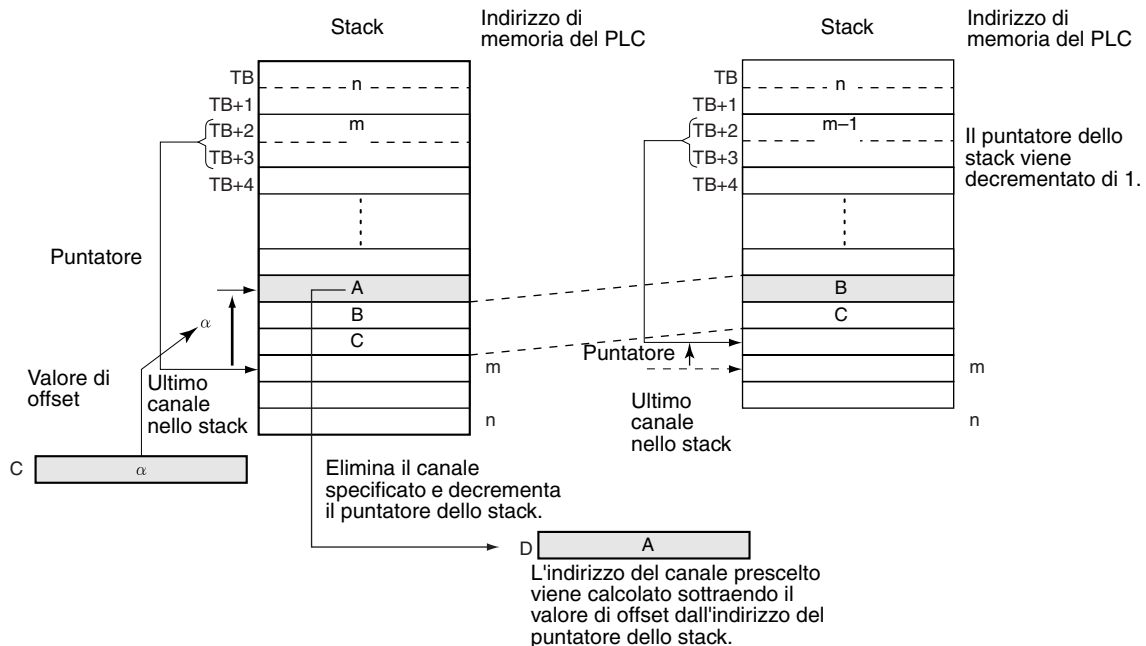
Caratteristiche operando

| Area | TB | C | D |
|-------------------------|------------------------|----------------|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A448 a A959 | Da A000 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |

| Area | TB | C | D |
|--|--|-----------------------------------|-----|
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0001 a #FFFB (esadecimale) | --- |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0(++) a ,IR15(++) Da ,-(- -)IR0 a ,-(- -)IR15 | | |

Descrizione

SDEL(642) elimina i dati che si trovano nella posizione specificata nello stack, invia i dati al canale di destinazione specificato e fa scorrere verso l'alto nello stack i dati rimanenti. Contemporaneamente SDEL(642) decrementa di 1 il puntatore dello stack (TB+3 e TB+2). La posizione dell'indirizzo prescelto viene calcolata sottraendo il valore di offset in C del puntatore dello stack.



È possibile utilizzare SDEL(642) per eliminare i dati relativi a un elemento che viene rifiutato dagli elementi presenti su un nastro trasportatore. La posizione del punto di eliminazione è semplicemente il numero di elementi restituiti (il valore di offset) dall'elemento più recente aggiunto al nastro trasportatore.

Flag

| Nome | Eti-chetta | Operazione |
|---------------------|------------|--|
| Flag di errore | ER | ON se il contenuto del puntatore dello stack (TB+3 e TB+2) è inferiore o uguale all'indirizzo di memoria del PLC del primo canale presente nell'area dati dello stack (TB+4). (Errore di underflow dello stack). ON se il valore di offset specificato in C è 0 o maggiore della dimensione massima dell'area dati (FFFB esadecimale). OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se i dati inviati a D sono 0000. OFF in tutti gli altri casi. |

Avvertenze

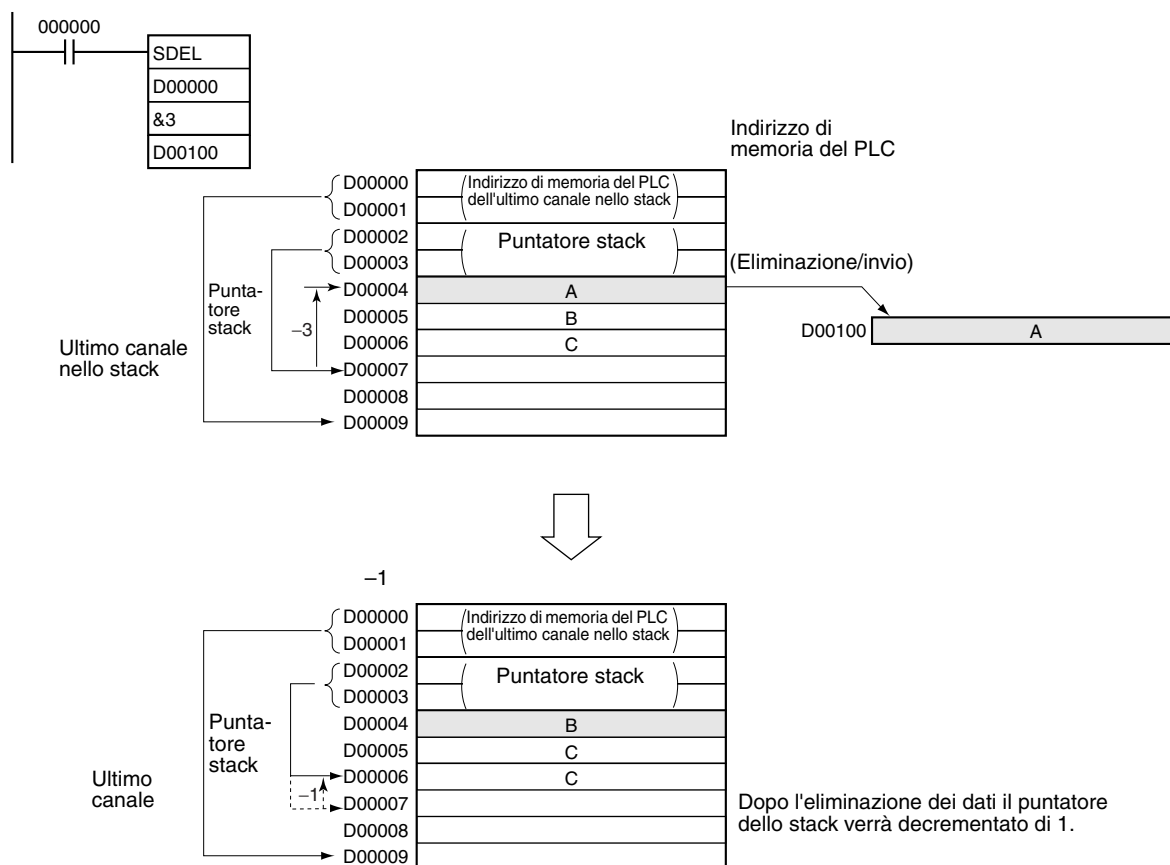
È necessario che lo stack sia stato definito in precedenza tramite l'istruzione SSET(630).

L'indirizzo nel puntatore dello stack deve essere maggiore dell'indirizzo di memoria del PLC dell'inizio dell'area dati (TB+4). Se il puntatore dello stack è minore dell'indirizzo di memoria del PLC di TB+4 si verificherà un errore, ovvero verrà generato un errore di underflow dello stack.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, SDEL(642) elimina il canale all'indirizzo specificato nello stack cominciando da D00000, invia a D00100 i dati eliminati, sposta verso l'alto in dati rimanenti e decrementa il puntatore dello stack.

In questo caso, il puntatore dello stack indica D00007 e il valore di offset è 3, quindi i dati verranno eliminati da D00004. I dati rimanenti verranno spostati di un canale verso l'alto e il puntatore dello stack verrà decrementato da D00007 a D00006.

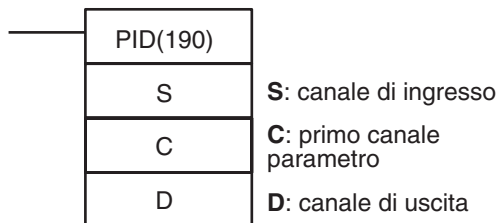


3-18 Istruzioni di controllo dei dati

3-18-1 PID CONTROL: PID(190)

Scopo Esegue un controllo PID in base ai parametri specificati.

Simbolo programmazione ladder



Variazioni

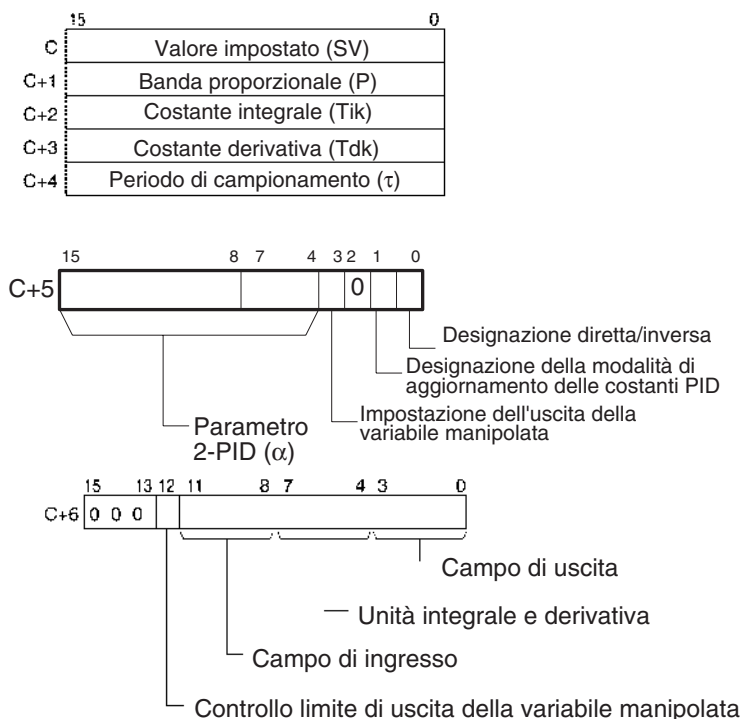
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | PID(190) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Parametri

Nei diagrammi seguenti sono indicate le posizioni dei dati di parametro. Per ulteriori dettagli sui parametri, consultare *Impostazioni dei parametri PID* in questa sezione.



Caratteristiche operando

| Area | S | C | D |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6105 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W473 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H473 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 | Da A000 ad A921 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4057 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4057 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32729 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32729 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32729 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | | |

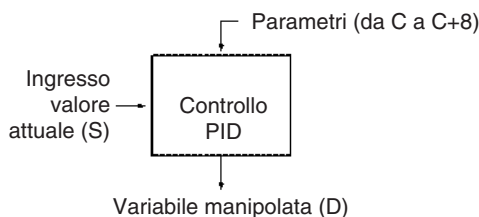
Descrizione

Quando la condizione di esecuzione è impostata su ON, PID(190) esegue un controllo PID filtrato del valore di riferimento a due gradi di libertà in base ai parametri specificati da C (valore impostato, costante PID, ecc.). Assume il campo di ingresso specificato dei dati binari dal contenuto del canale di ingresso S ed esegue l'azione PID in base ai parametri impostati. Il risultato viene quindi memorizzato come variabile manipolata nel canale di uscita D.

I parametri vengono ottenuti quando la condizione di esecuzione passa da OFF a ON e il flag di errore viene attivato se le impostazioni non rientrano nell'intervallo consentito.

Se le impostazioni sono corrette, l'elaborazione PID verrà eseguita utilizzando i valori iniziali. In questa fase non è possibile effettuare l'operazione bumpless, che verrà utilizzata per le variabili manipolate nella successiva esecuzione dell'elaborazione PID. Per operazione bumpless si intende l'elaborazione che modifica le variabili manipolate in modo graduale e ininterrotto per evitare gli effetti negativi dei cambiamenti improvvisi.

Quando la condizione di esecuzione passa a ON, viene specificato il valore attuale (PV) per il periodo di campionamento specificato e l'elaborazione ha inizio.



Il numero di bit in ingresso validi all'interno dei 16 bit dell'ingresso del PV (S) viene designato dall'impostazione del campo di ingresso in C+6, cioè i bit compresi tra 08 e 11. Ad esempio, se viene specificato 12 bit (4 esadecimale) per il campo di ingresso, l'intervallo compreso tra 0000 esadecimale e 0FFF esadecimale verrà attivato come valore attuale (PV) e i valori maggiori di 0FFF esadecimale verranno considerati come 0FFF esadecimale.

Anche l'intervallo relativo al valore impostato dipende dal campo di ingresso.

I valori misurati (PV) e i valori impostati (SV) sono in formato binario senza segno e sono compresi tra 0000 esadecimale e il valore massimo del campo di ingresso.

Il numero di bit validi dei dati di uscita tra i 16 bit dell'uscita della variabile manipolata viene designato dall'impostazione del campo di uscita in C+6, cioè i bit compresi tra 00 e 03. Ad esempio, se viene specificato 12 bit (4 esadecimale) per il campo di uscita, l'intervallo compreso tra 0000 esadecimale e 0FFF esadecimale sarà inviato come variabile manipolata.

Quando il valore attuale (PV) è uguale al valore impostato (SV) e solo nel caso di operazioni proporzionali, è possibile designare l'uscita della variabile manipolata in questo modo:

- 0: uscita 0%
- 1: uscita 50%.

La direzione dell'operazione proporzionale può essere designata sia come diretta che come inversa.

È possibile specificare i limiti superiore e inferiore dell'uscita della variabile manipolata.

Il periodo di campionamento può essere designato in unità di 10 ms (da 0,01 a 99,99 s), ma l'effettiva azione PID è determinata da una combinazione del periodo di campionamento e del tempo necessario per l'esecuzione dell'istruzione PID(190) a ogni ciclo.

È possibile impostare la modalità con cui apportare modifiche alle costanti PID in due modi: 1) all'inizio dell'esecuzione dell'istruzione PID oppure 2) all'inizio dell'esecuzione dell'istruzione PID e a ogni periodo di campionamento. Solo la banda proporzionale (P), la costante integrale (Ti) e la costante derivativa possono essere modificate a ogni ciclo di campionamento, ossia durante l'esecuzione dell'istruzione PID. La modalità di aggiornamento viene impostata al bit 1 di C+5.

Nota L'impostazione nel bit 1 di C+5 è supportata solo dalle CPU CJ1, CS1-H e CJ1-H nonché dalle CPU CS1 con numero di lotto a partire da 001201□□□□, vale a dire prodotte dall'1 dicembre 2000.

Quando la condizione di esecuzione è impostata su ON, il valore impostato (SV) è l'unico dei parametri PID (compresi tra C e C+38) che può essere modificato. Per modificare altri valori, assicurarsi di impostare la condizione di esecuzione su ON.

Flag

| Nome | Eti-chetta | Funzionamento |
|------------------|------------|--|
| Flag di errore | ER | ON se i dati in C non rientrano nell'intervallo. ON se il periodo di campionamento effettivo supera di oltre il doppio il periodo di campionamento designato. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se la variabile manipolata dopo l'azione PID supera il limite superiore. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se la variabile manipolata dopo l'azione PID è al di sotto del limite inferiore. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON in fase di esecuzione del controllo PID. OFF in tutti gli altri casi. |

Avvertenze

PID(190) viene eseguita come se la condizione di esecuzione fosse un segnale STOP-RUN. I calcoli PID vengono effettuati quando la condizione di esecuzione resta impostata su ON per il ciclo successivo all'inizializzazione dei canali compresi tra C+9 e C+38. Di conseguenza, quando si utilizza il flag di sempre ON come condizione di esecuzione per PID(190), all'avvio dell'operazione è opportuno fornire un'elaborazione distinta per l'inizializzazione di tali canali.

Se i dati in C non rientrano nell'intervallo, si verificherà un errore e verrà attivato il flag di errore.

Se il periodo di campionamento effettivo supera di oltre il doppio il periodo di campionamento designato, si verificherà un errore e verrà attivato il flag di errore. Il controllo PID viene comunque eseguito.

Il flag di riporto viene attivato in fase di esecuzione del controllo PID.

Il flag di maggiore viene attivato se la variabile manipolata dopo l'azione PID supera il limite superiore. In questa fase, i risultati vengono inviati al limite superiore.

Il flag di minore viene attivato se la variabile manipolata dopo l'azione PID è al di sotto del limite inferiore. In questa fase, i risultati vengono inviati al limite inferiore.

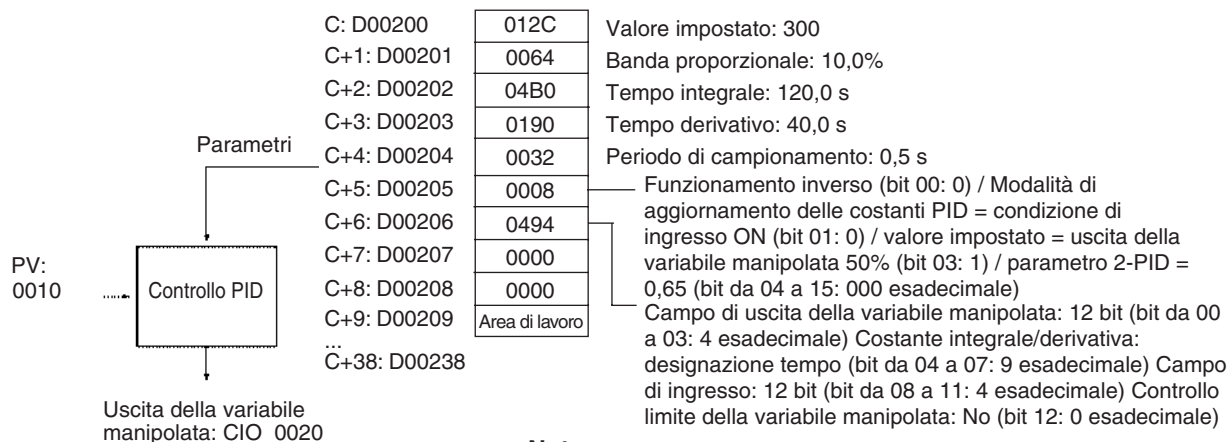
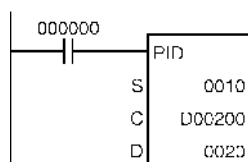
Quando la condizione di ingresso è impostata su ON, il valore impostato per C è l'unico dei parametri PID (compresi tra C e C+38) che può essere modificato. Se sono stati modificati altri valori, assicurarsi di impostare la condizione di esecuzione su ON per convalidarli.

Esempio

Al fronte di salita di CIO 000000 (da OFF a ON), l'area di lavoro da D00209 a D00238 viene inizializzata in base ai parametri impostati da D00200 a D00208 illustrati qui di seguito. In seguito all'inizializzazione dell'area di lavoro viene eseguito il controllo PID e la variabile manipolata viene inviata a CIO 0020.

Quando CIO 000000 passa a ON, viene eseguito il controllo PID agli intervalli del periodo di campionamento in base ai parametri impostati da D00200 a D00208. La variabile manipolata viene inviata a CIO 0020.

Se la banda proporzionale (P), la costante integrale (Tik) o la costante derivativa vengono modificate dopo l'attivazione di CIO 000000, le costanti PID utilizzate nei calcoli PID restano inalterate.



Nota Quando CIO 000000 è OFF, la scrittura in CIO 0020 consente di procedere come per l'operazione manuale.

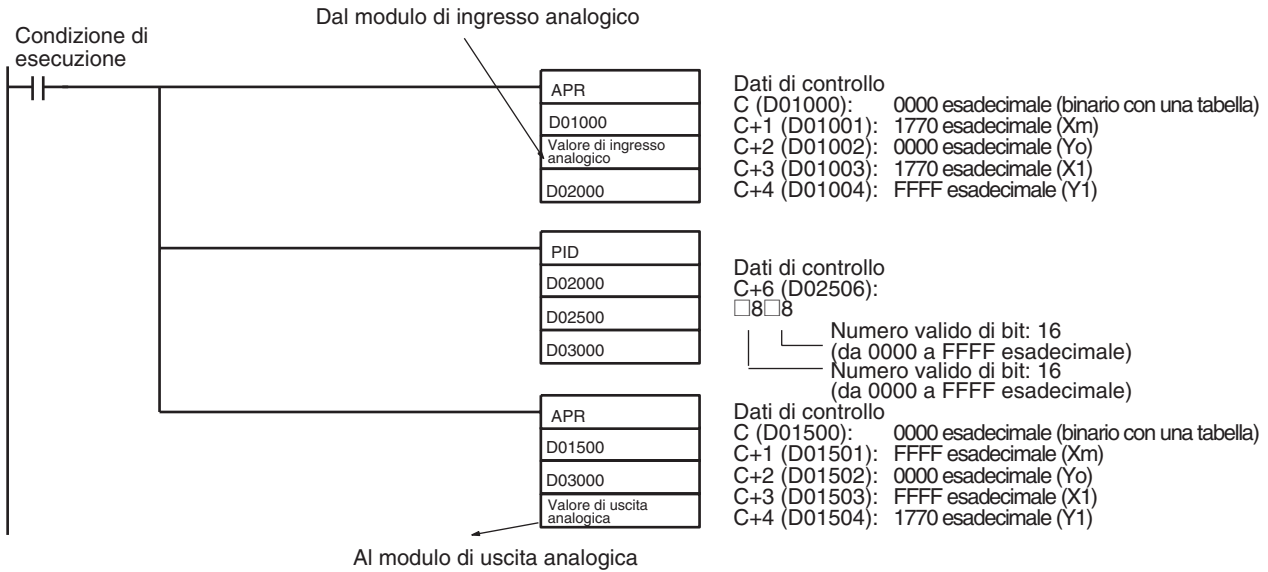
Valori di ingresso e intervalli della variabile manipolata

Il numero di bit validi dei dati di ingresso per il valore misurato viene designato dall'impostazione del campo di ingresso in C+6, cioè i bit compresi tra 08 e 11; il numero di bit validi dei dati di uscita per l'uscita della variabile manipolata è designato dall'impostazione del campo di uscita in C+6, cioè i bit compresi tra 00 e 03. Tali intervalli vengono riportati nella tabella seguente.

| C+6, bit da 08 a 11 o C+6, bit da 00 a 03 | Numero di bit validi | Intervallo |
|---|----------------------|----------------------------|
| 0 | 8 | Da 0000 a 00FF esadecimale |
| 1 | 9 | Da 0000 a 01FF esadecimale |
| 2 | 10 | Da 0000 a 03FF esadecimale |
| 3 | 11 | Da 0000 a 07FF esadecimale |
| 4 | 12 | Da 0000 a 0FFF esadecimale |
| 5 | 13 | Da 0000 a 1FFF esadecimale |
| 6 | 14 | Da 0000 a 3FFF esadecimale |
| 7 | 15 | Da 0000 a 7FFF esadecimale |
| 8 | 16 | Da 0000 a FFFF esadecimale |

Se l'intervallo dei dati gestito da un modulo di ingresso analogico o da un modulo di uscita analogico non può essere impostato con esattezza definendo il numero dei bit validi, prima e dopo PID(190) è possibile utilizzare APR(069) (ARITHMETIC PROCESS) per eseguire la conversione negli intervalli corretti.

Nella seguente sezione del programma viene illustrato l'esempio di un modulo di ingresso analogico DRT1-AD04 e di un modulo di uscita analogico DRT1-DA02 che funzionano come slave DeviceNet. L'intervallo dei dati di questi due moduli è compreso tra 0000 e 1770 esadecimale e non può essere specificato con la sola impostazione del numero valido di cifre. Di conseguenza APR(069) viene utilizzata per convertire l'intervallo compreso tra 0000 e 1770 esadecimale del modulo di ingresso analogico in un intervallo compreso tra 0000 ed FFFF esadecimale per l'ingresso in PID(190). Sempre tramite APR(190), l'uscita della variabile manipolata da PID(190) viene riconvertita nell'intervallo compreso tra 0000 e 1770 esadecimale per l'uscita dal modulo di uscita analogico.



Prestazioni dichiarate

| Elemento | | Specifiche | |
|----------------------------------|---------------------|--|--|
| Metodo di controllo PID | --- | Metodo PID di tipo filtro a due gradi di libertà per valore di riferimento (diretto/inverso) | |
| Numero di cicli di controllo PID | --- | Illimitato (1 ciclo per istruzione) | |
| Periodo di campionamento | τ | 0,01 ... 99,99 s | |
| Costante PID | Banda proporzionale | P | 0,1 ... 999,9% |
| | Costante integrale | Tik | 1 ... 8191, 9999 (nessuna azione integrale per impostazione a 9999). |
| | Costante derivativa | Tdk | 0 ... 8191 (nessuna azione derivativa per impostazione a 0). |
| Valore impostato | SV | 0 ... 65535 (valido fino al valore massimo del campo di ingresso). | |
| Valore misurato | PV | 0 ... 65535 (valido fino al valore massimo del campo di ingresso). | |
| Variabile manipolata | MV | 0 ... 65535 (valido fino al valore massimo del campo di uscita). | |

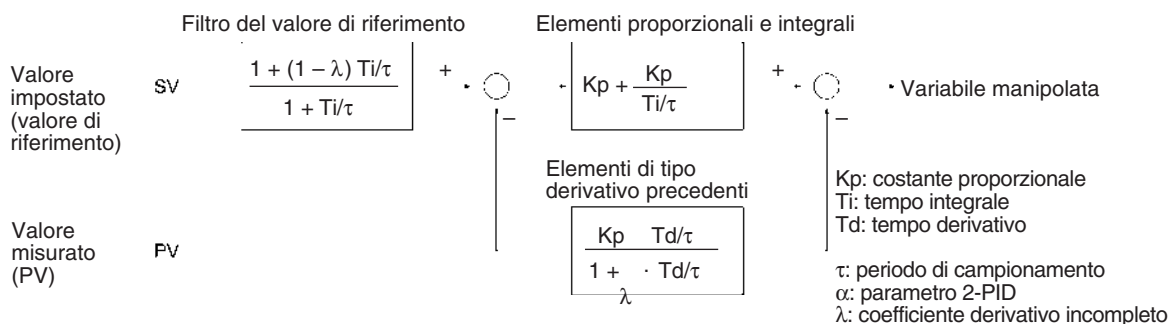
Metodo di calcolo

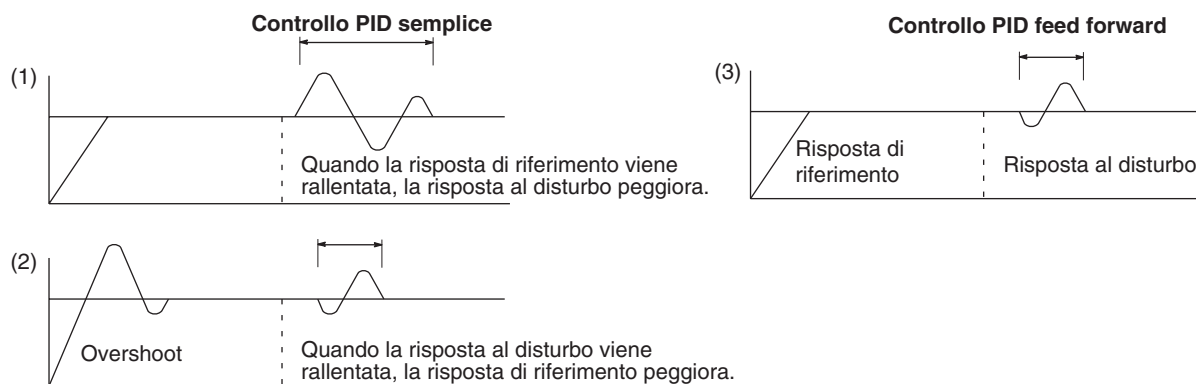
Nel controllo PID i calcoli sono eseguiti dal controllo filtrato del valore di riferimento a due gradi di libertà.

Schema a blocchi per PID del valore di riferimento a due gradi di libertà

Quando un controllo PID semplice è in grado di evitare l'overshooting, la stabilizzazione dei disturbi risulta rallentata (1). Se, d'altra parte, avviene un'accelerazione della stabilizzazione dei disturbi, si verifica l'overshooting e viene rallentata la risposta per il valore di riferimento (2).

Quando si utilizza il controllo PID del valore di riferimento a due gradi di libertà, non si verifica comunque alcun overshooting e sia la risposta per il valore di riferimento che la stabilizzazione dei disturbi vengono accelerate (3).





Impostazioni dei parametri PID

| Dati di controllo | Elemento | Contenuto | Intervallo di impostazione | Modifica con condizione di ingresso ON |
|-----------------------|---|---|---|---|
| C | Valore impostato (SV) | Il valore di riferimento del processo controllato. | Dati binari (dello stesso numero di bit specificato per il campo di ingresso) | Consentita |
| C+1 | Banda proporzionale | Il parametro per l'azione P che rappresenta l'intervallo di controllo proporzionale/totale. | Da 0001 a 270F esadecimale (da 1 a 9999); (da 0,1% a 999,9%, in unità di 0,1%) | Modifiche consentite con condizione di ingresso ON, se il bit 1 di C+5 è 1. |
| C+2 | Tik Costante integrale | Una costante che rappresenta la forza dell'azione integrale. Se questo valore aumenta, la forza integrale diminuisce. | Da 0001 a 1FFF esadecimale (da 1 a 8191); (9999 = operazione integrale non eseguita - vedere nota 1) | |
| C+3 | Tdk Costante derivativa | Una costante che rappresenta la forza dell'azione derivativa. Se questo valore aumenta, la forza derivativa diminuisce. | Da 0001 a 1FFF esadecimale (da 1 a 8191); (0000 = operazione derivativa non eseguita - vedere nota 1) | |
| C+4 | Periodo di campionamento (τ) | Imposta il periodo di esecuzione dell'azione PID. | Da 0001 a 270F esadecimale (da 1 a 9999); (da 0,01 a 99,99 s, in unità di 10 ms) | Non consentita |
| Bit da 04 a 15 di C+5 | Parametro 2-PID (α) | Il coefficiente del filtro di ingresso. Di norma utilizzare 0,65 (ossia un'impostazione di 000). L'efficienza del filtro diminuisce quando il coefficiente si avvicina allo 0. | 000 esadecimale: $\alpha = 0,65$ L'impostazione da 100 a 163 esadecimale significa che il valore delle due cifre all'estrema destra viene definito da $\alpha = 0,00$ ad $\alpha = 0,99$ (vedere nota 2) | Consentita |
| Bit 03 di C+5 | Designazione dell'uscita della variabile manipolata | Designa l'uscita della variabile manipolata quando il valore attuale (PV) è uguale al valore impostato (SV). | 0: uscita 0% 1: uscita 50% | |
| Bit 01 di C+5 | Impostazione che abilita la modifica della costante PID | I momenti in cui è consentito apportare modifiche alla banda proporzionale (P), alla costante integrale (Tik) e alla costante derivativa (Tdk) per utilizzarle nei calcoli PID. | 0: all'inizio dell'esecuzione dell'istruzione PID 1: all'inizio dell'esecuzione dell'istruzione PID e a ogni periodo di campionamento | Consentita |

| Dati di controllo | Elemento | Contenuto | Intervallo di impostazione | Modifica con condizione di ingresso ON |
|-----------------------|---|--|---|--|
| Bit 00 di C+5 | Designazione diretta/inversa per PID | Determina la direzione dell'azione proporzionale. | 0: azione inversa 1: azione diretta | Non consentita |
| Bit 12 di C+6 | Controllo limite di uscita della variabile manipolata | Determina l'eventuale controllo limite applicato all'uscita della variabile manipolata. | 0: disabilitato (nessun controllo limite) 1: abilitato (controllo limite) | |
| Bit da 08 a 11 di C+6 | Campo di ingresso | Il numero dei bit dei dati di ingresso. | 0: 8 bit 5: 13 bit 1: 9 bit 6: 14 bit 2: 10 bit 7: 15 bit 3: 11 bit 8: 16 bit 4: 12 bit | |
| Bit da 04 a 07 di C+6 | Unità integrale e derivativa | Determina l'unità che rappresenta le costanti integrali e derivative. | 1: multiplo del periodo di campionamento 9: tempo (unità: 100 ms) | |
| Bit da 00 a 03 di C+6 | Campo di uscita | Il numero dei bit dei dati di uscita (per impostazione predefinita, il numero dei bit di uscita equivale al numero dei bit di ingresso). | 0: 8 bit 5: 13 bit 1: 9 bit 6: 14 bit 2: 10 bit 7: 15 bit 3: 11 bit 8: 16 bit 4: 12 bit | |
| C+7 | Limite inferiore di uscita della variabile manipolata | Il limite inferiore nel caso in cui sia abilitato il limite di uscita della variabile manipolata. | Da 0000 a FFFF (binario) (vedere la nota 3). | |
| C+8 | Limite superiore di uscita della variabile manipolata | Il limite superiore nel caso in cui sia abilitato il limite di uscita della variabile manipolata. | Da 0000 a FFFF (binario) (vedere la nota 3). | |

- Nota**
- Quando l'unità è designata come 1, l'intervallo varia da 1 a 8.191 volte il periodo. Quando l'unità è designata come 9, l'intervallo è compreso tra 0,1 e 819,1 s. Se viene designato 9, impostare i tempi integrale e derivativo in un intervallo da 1 a 8.191 volte il periodo di campionamento.
 - L'impostazione del parametro 2-PID (α) su 000 produce 0,65, il valore normale.
 - Quando il controllo limite di uscita della variabile manipolata è abilitato (cioè è impostato su "1"), definire i valori come indicato qui di seguito.
 $0000 \leq \text{Limite inferiore uscita MV} \leq \text{Limite superiore uscita MV} \leq \text{Valore massimo campo di uscita}$

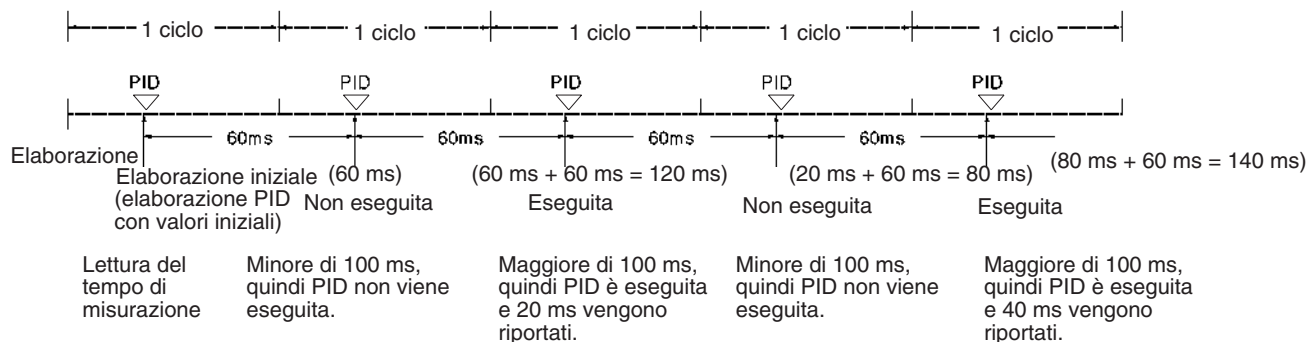
Periodo di campionamento e tempo di ciclo

Il periodo di campionamento può essere designato in unità di 10 ms (da 0,01 a 99,99 s), ma l'effettiva azione PID è determinata da una combinazione del periodo di campionamento e del tempo necessario per l'esecuzione dell'istruzione PID a ogni ciclo. La relazione tra il periodo di campionamento e il tempo di ciclo è illustrata qui di seguito.

- Se il periodo di campionamento è minore del tempo di ciclo, il controllo PID viene eseguito a ogni ciclo e non a ogni periodo di campionamento.
- Se il periodo di campionamento è maggiore al tempo di ciclo o uguale, il controllo PID non viene eseguito a ogni ciclo; tuttavia PID(190) viene eseguita quando il valore complessivo del tempo di ciclo (ossia il tempo tra le istruzioni PID) è superiore o uguale al periodo di campionamento. La parte eccedente del valore complessivo (ossia il valore complessivo del tempo di ciclo meno il periodo di campionamento) viene riportata al valore complessivo successivo.

Ad esempio, se il periodo di campionamento è pari a 100 ms e il tempo di ciclo è coerentemente di 60 ms, nel primo ciclo successivo all'esecuzione iniziale PID(190) non viene eseguita poiché 60 ms è minore di 100 ms. Nel secondo ciclo, invece, la somma di 60 ms + 60 ms è maggiore di 100 ms e quindi PID(190) sarà eseguita. Il resto di 20 ms (cioè 120 ms – 100 ms = 20 ms) viene riportato.

Nel terzo ciclo, il resto di 20 ms viene aggiunto a 60 ms. Poiché la somma, 80 ms, è minore di 100 ms, PID(190) non viene eseguita. Nel quarto ciclo, gli 80 ms vengono sommati ai 60 ms. La somma, 140 ms, è maggiore di 100 ms, quindi PID(190) viene eseguita e il resto di 40 ms (cioè $120\text{ ms} - 100\text{ ms} = 20\text{ ms}$) viene riportato. Questa procedura si ripete nei cicli successivi.



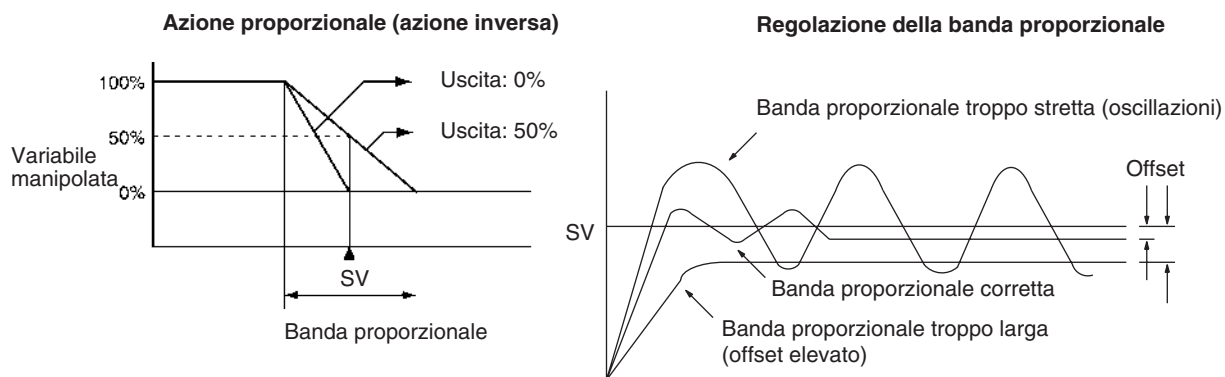
Azioni di controllo

Azione proporzionale (P)

Si tratta di un'operazione nella quale si definisce una banda proporzionale in relazione al valore impostato (SV); la variabile manipolata (MV) di tale banda, inoltre, viene resa proporzionale alla deviazione. Nell'illustrazione seguente viene riportato un esempio di funzionamento inverso.

Se viene utilizzata l'azione proporzionale e il valore attuale (PV) diventa minore della banda proporzionale, la variabile manipolata (MV) è 100% (valore massimo). Nella banda proporzionale, la MV è resa proporzionale alla deviazione (differenza tra SV e PV) e diminuisce gradualmente fino a che SV e PV equivalgono, cioè fino a quando la deviazione è 0. A questo punto la MV avrà il valore minimo di 0% (o 50%, in base all'impostazione del parametro di designazione dell'uscita della variabile manipolata). Anche quando il valore attuale è maggiore del valore impostato, la MV sarà 0%.

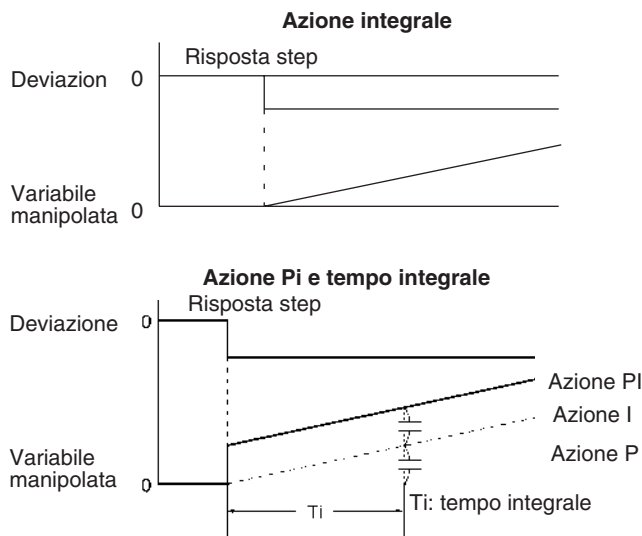
La banda proporzionale viene espressa come percentuale del campo di ingresso totale. Con la diminuzione della banda proporzionale aumenta la costante proporzionale e l'azione correttiva diventa più efficace. Con l'azione proporzionale generalmente avviene un offset (deviazione residua), che può essere ridotto diminuendo la banda proporzionale. In caso di riduzione eccessiva, tuttavia, si verificano oscillazioni.



Azione integrale (I)

La combinazione di azione integrale e proporzionale riduce l'offset in base al tempo trascorso, affinché valore attuale e valore impostato coincidano. La forza dell'azione integrale è indicata dal tempo integrale, cioè il tempo necessario alla variabile manipolata dell'azione integrale per arrivare allo stesso livello della variabile manipolata dell'azione proporzionale rispetto alla

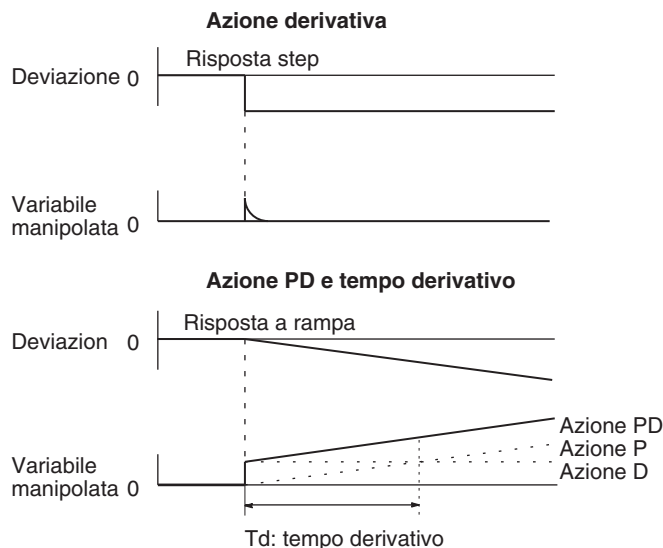
deviazione di step. Osservare la tabella seguente. Più è breve il tempo integrale, più sarà forte la correzione dell'azione integrale. Se il tempo integrale è troppo breve, la correzione sarà troppo forte e provocherà oscillazioni.



Azione derivativa (D)

L'azione proporzionale e quella integrale eseguono entrambe correzioni in base ai risultati del controllo e quindi si verifica inevitabilmente un ritardo nella risposta. L'azione derivativa compensa questo inconveniente. In risposta a un disturbo improvviso, invia una variabile manipolata grande e ripristina tempestivamente lo stato originario. Viene eseguita una correzione con la variabile manipolata resa proporzionale alla pendenza (coefficiente derivativo) causata dalla deviazione.

La forza dell'azione derivativa è indicata dal tempo derivativo, cioè il tempo necessario alla variabile manipolata dell'azione derivativa per arrivare allo stesso livello della variabile manipolata dell'azione proporzionale rispetto alla deviazione di step. Osservare la tabella seguente. Più è lungo il tempo derivativo, più sarà forte la correzione dell'azione derivativa.

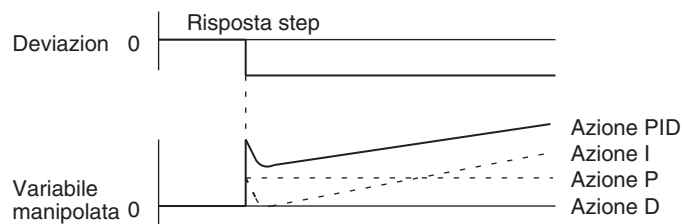


Azione PID

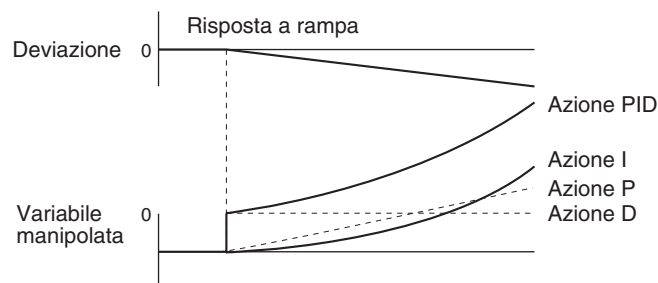
L'azione PID combina l'azione proporzionale (P), l'azione integrale (I) e l'azione derivativa (D). Produce risultati di controllo di livello superiore anche per oggetti di controllo con dead time. Utilizza l'azione proporzionale per fornire un controllo tranquillo senza oscillazioni, l'azione integrale per

correggere qualsiasi offset in modo automatico e l'azione derivativa per velocizzare la risposta ai disturbi.

Risposta step dell'uscita dell'azione di controllo PID



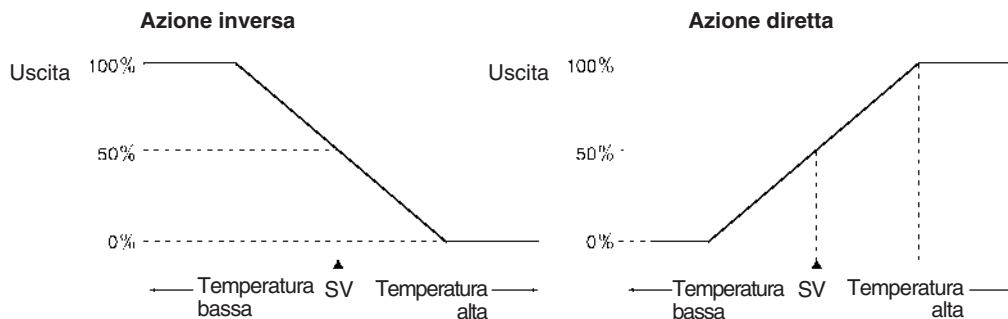
Risposta a rampa dell'uscita dell'azione di controllo PID



Direzione dell'azione

Quando si utilizza il controllo PID, selezionare una delle due seguenti direzioni di controllo, tenendo presente che in entrambi i casi la MV aumenta parallelamente al valore attuale (PV) e al valore impostato (SV).

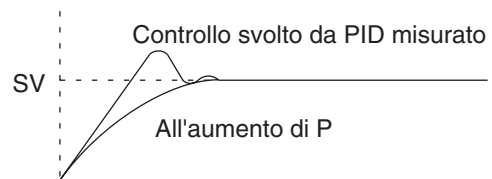
- Azione diretta: la MV aumenta quando il PV è maggiore dell'SV.
- Azione inversa: la MV aumenta quando il PV è minore dell'SV.



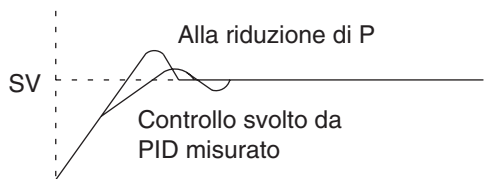
Regolazione dei parametri PID

Qui di seguito viene fornita la relazione generale tra i parametri PID e lo stato di controllo.

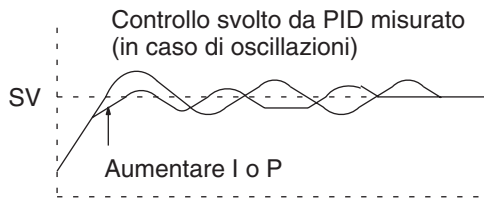
- Quando è fondamentale non provocare overshooting e si dispone del tempo necessario per la stabilizzazione (tempo di assestamento), aumentare la banda proporzionale.



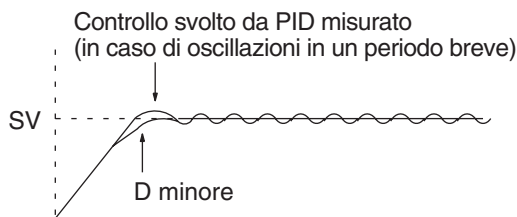
- Quando l'overshooting non rappresenta un problema ma si desidera stabilizzare il controllo rapidamente, restringere la banda proporzionale. Se la banda proporzionale viene ridotta eccessivamente, potrebbero comunque verificarsi oscillazioni.



- Quando si verificano oscillazioni marcate o quando il funzionamento avviene in presenza di overshooting e undershooting, la ragione risiede verosimilmente in un'azione integrale troppo forte. L'aumento del tempo integrale o l'allargamento della banda proporzionale consentono di ridurre le oscillazioni.



- Se il periodo è breve e si verificano oscillazioni, è possibile che la risposta del sistema di controllo sia rapida e l'azione derivativa troppo forte. In tal caso, impostare l'azione derivativa su un valore più basso.



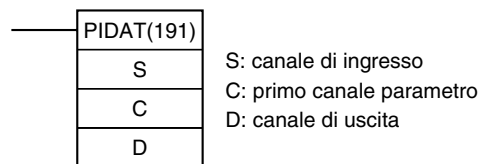
3-18-2 PID CONTROL WITH AUTOTUNING: PIDAT(191)

Scopo

Esegue un controllo PID in base ai parametri specificati. Per le costanti PID è possibile eseguire l'autotuning.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

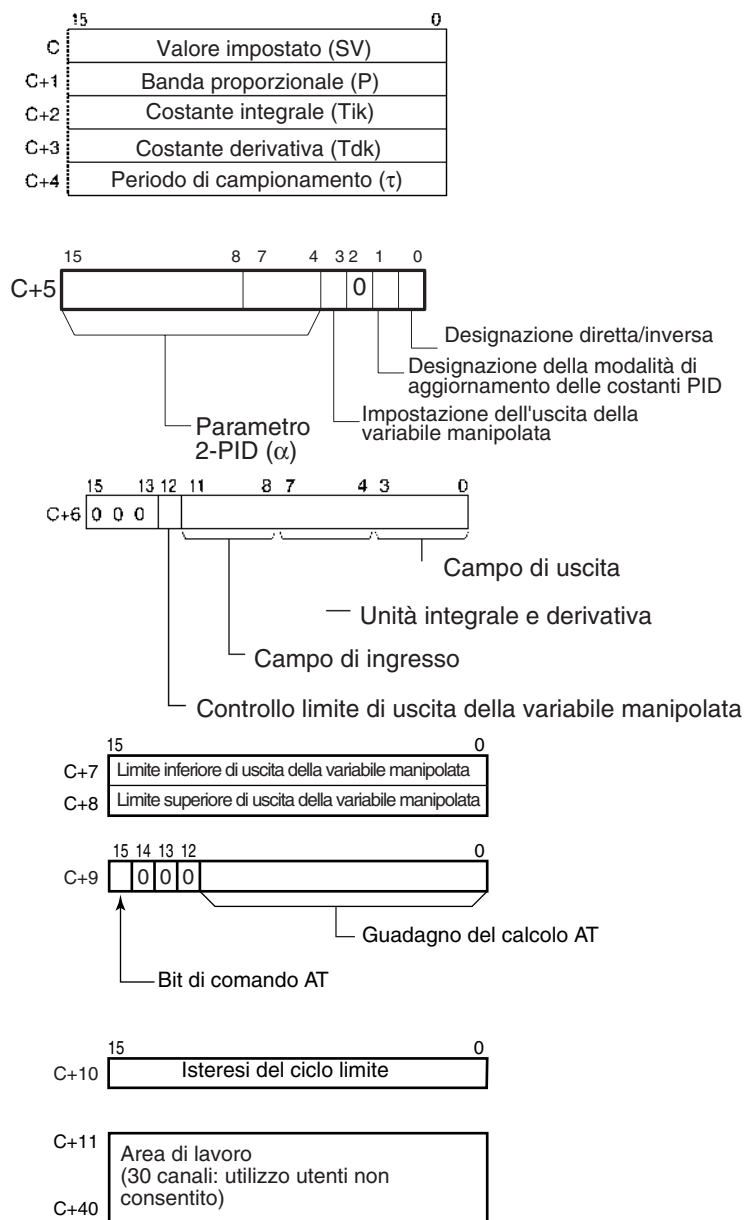
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | PIDAT(191) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| Non consentita | OK | OK | Non consentita |

Parametri

Nei diagrammi seguenti sono indicate le posizioni dei dati di parametro. Per ulteriori dettagli sui parametri, consultare *Impostazioni dei parametri PID* in questa sezione.



Caratteristiche operando

| Area | S | C | D |
|-------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6105 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W473 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H473 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 | Da A000 ad A921 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4057 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4057 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32729 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32729 | Da E00000 a E32767 |

| Area | S | C | D |
|--|--|--|--|
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32729 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | | |

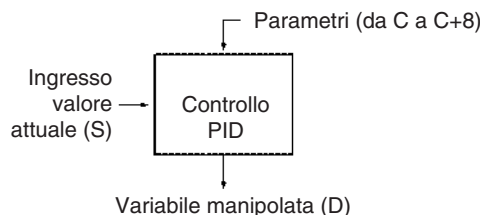
Descrizione

Quando la condizione di esecuzione è impostata su ON, PIDAT(191) esegue un controllo PID filtrato del valore di riferimento a due gradi di libertà in base ai parametri designati da C (valore impostato, costante PID, ecc.). Assume il campo di ingresso specificato dei dati binari dal contenuto del canale di ingresso S ed esegue l'azione PID in base ai parametri impostati. Il risultato viene quindi memorizzato come variabile manipolata nel canale di uscita D.

Le impostazioni dei parametri vengono lette quando la condizione di esecuzione passa da OFF a ON e il flag di errore viene attivato se le impostazioni non rientrano nell'intervallo consentito.

Se le impostazioni sono corrette, l'elaborazione PID verrà eseguita utilizzando i valori iniziali. In questa fase non è possibile effettuare l'operazione bumpless, che verrà utilizzata per le variabili manipolate nella successiva esecuzione dell'elaborazione PID. Per operazione bumpless si intende l'elaborazione che modifica le variabili manipolate in modo graduale e ininterrotto per evitare gli effetti negativi dei cambiamenti improvvisi.

Quando la condizione di esecuzione passa a ON, viene specificato il valore attuale (PV) per il periodo di campionamento specificato e l'elaborazione ha inizio.



Autotuning

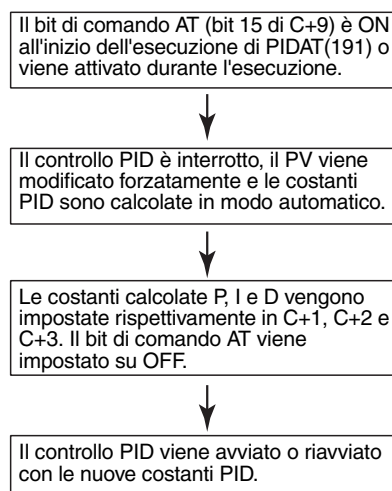
Lo stato del bit di comando AT (bit 15 di C+9) viene controllato a ogni ciclo. Se tale bit di controllo passa a ON in un determinato ciclo, PIDAT(191) inizia a eseguire l'autotuning per le costanti PID. Durante l'autotuning, non vengono aggiornate le modifiche del valore impostato.

Per l'autotuning viene utilizzato il metodo del ciclo limite. PIDAT(191) modifica efficacemente la variabile manipolata (variabile manipolata massima ↔ variabile manipolata minima) e verifica le caratteristiche del sistema controllato. Le costanti PID vengono calcolate in base alle caratteristiche riscontrate e le nuove costanti P, I e D sono memorizzate automaticamente in

C+1, C+2 e C+3. A questo punto il bit di comando AT (bit 15 di C+9) passa a OFF e il controllo PID riprende dalle nuove costanti in C+1, C+2 e C+3.

- Se all'inizio dell'esecuzione di PIDAT(191) il bit di comando AT è impostato su ON, in primo luogo viene effettuato l'autotuning e quindi viene avviato il controllo PID con le costanti PID calcolate.
- Se durante l'esecuzione di PIDAT(191) il bit di comando AT è impostato su ON, l'istruzione interrompe il controllo PID in fase di esecuzione con le costanti PID definite dall'utente, esegue l'autotuning e infine riprende il controllo PID con le costanti PID calcolate.

Il diagramma di flusso seguente illustra la procedura dell'autotuning:



- Nota**
1. Se l'autotuning viene interrotto dalla disattivazione del bit di comando AT durante l'autotuning stesso, il controllo PID viene avviato con le costanti PID utilizzate prima dell'inizio dell'autotuning.
 2. Inoltre, se si verifica un errore di esecuzione AT, il controllo PID viene avviato con le costanti PID utilizzate prima dell'inizio dell'autotuning.

Nei due casi illustrati dalle note 1 e 2, le costanti PID vengono abilitate se già calcolate al momento dell'interruzione dell'autotuning.

Controllo PID

Il numero di bit in ingresso validi all'interno dei 16 bit dell'ingresso del PV (S) viene designato dall'impostazione del campo di ingresso in C+6, cioè i bit compresi tra 08 e 11. Ad esempio, se viene specificato 12 bit (4 esadecimale) per il campo di ingresso, l'intervallo compreso tra 0000 esadecimale e 0FFF esadecimale verrà attivato come valore attuale (PV) e i valori maggiori di 0FFF esadecimale verranno considerati come 0FFF esadecimale.

Anche l'intervallo relativo al valore impostato dipende dal campo di ingresso.

I valori misurati (PV) e i valori impostati (SV) sono in formato binario senza segno e sono compresi tra 0000 esadecimale e il valore massimo del campo di ingresso.

Il numero di bit validi dei dati di uscita tra i 16 bit dell'uscita della variabile manipolata viene designato dall'impostazione del campo di uscita in C+6, cioè i bit compresi tra 00 e 03. Ad esempio, se viene designato 12 bit (4 esadecimale) per il campo di uscita, l'intervallo compreso tra 0000 esadecimale e 0FFF esadecimale sarà inviato come variabile manipolata.

Quando il valore attuale (PV) è uguale al valore impostato (SV) e solo nel caso di operazioni proporzionali, è possibile designare l'uscita della variabile manipolata in questo modo:

- 0: uscita 0%
- 1: uscita 50%.

La direzione dell'operazione proporzionale può essere designata sia come diretta che come inversa.

È possibile specificare i limiti superiore e inferiore dell'uscita della variabile manipolata.

Il periodo di campionamento può essere designato in unità di 10 ms (da 0,01 a 99,99 s), ma l'effettiva azione PID è determinata da una combinazione del periodo di campionamento e del tempo necessario per l'esecuzione dell'istruzione PIDAT(191) a ogni ciclo.

È possibile impostare la modalità con cui apportare modifiche alle costanti PID in due modi: 1) all'inizio dell'esecuzione dell'istruzione PIDAT(191) oppure 2) all'inizio dell'esecuzione dell'istruzione PID e a ogni periodo di campionamento. Solo la banda proporzionale (P), la costante integrale (TiK) e la costante derivativa possono essere modificate a ogni ciclo di campionamento, ossia durante l'esecuzione dell'istruzione PID. La modalità di aggiornamento viene impostata al bit 1 di C+5.

Quando si modificano le costanti PID manualmente, definire l'impostazione che abilita la modifica della costante PID (bit 1 di C+5) su 1 per aggiornare i valori in C+1, C+2 e C+3 a ogni periodo di campionamento nel calcolo PID. Questa impostazione consente inoltre di regolare le costanti PID manualmente al completamento dell'autotuning.

Quando la condizione di esecuzione è impostata su ON, i seguenti sono gli unici parametri PID (compresi tra C e C+38) che possono essere modificati. Nel caso di modifiche ad altri valori, assicurarsi di impostare la condizione di esecuzione su ON per abilitare le nuove impostazioni.

- Valore impostato (SV) in C
(modificabile solo durante il controllo PID. Una modifica dell'SV durante l'aggiornamento non viene applicata).
- Impostazione che abilita la modifica della costante PID (bit 1 di C+5)
- Le costanti P, I e D in C+1, C+2 e C+3.
Le modifiche di queste costanti vengono aggiornate a ogni periodo di campionamento solo se l'impostazione che abilita la modifica della costante PID (bit 1 di C+5) è definita su 1.
- Bit di comando AT (bit 15 di C+9)
- Guadagno del calcolo AT (bit da 0 a 14 di C+9) e Isteresi del ciclo limite (C+10): si tratta di valori che vengono letti all'avvio dell'autotuning.

Nota L'istruzione PIDAT(191) è uguale all'istruzione PID(190) con l'aggiunta della funzione di autotuning (AT); di conseguenza, le operazioni di controllo PID sono identiche. Per ulteriori informazioni ed esempi sulle operazioni di controllo PID, consultare *3-18-1 PID CONTROL: PID(190)*.

Flag

| Nome | Eti-chetta | Funzionamento |
|------------------|------------|--|
| Flag di errore | ER | ON se i dati in C non rientrano nell'intervallo. ON se il periodo di campionamento effettivo supera di oltre il doppio il periodo di campionamento designato. ON in caso di errore durante l'autotuning. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se la variabile manipolata dopo l'azione PID supera il limite superiore. OFF in tutti gli altri casi. |

| Nome | Eti-chetta | Funzionamento |
|-----------------|------------|---|
| Flag di minore | < | ON se la variabile manipolata dopo l'azione PID è al di sotto del limite inferiore. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON in fase di esecuzione del controllo PID. OFF in tutti gli altri casi. |

Avvertenze

PIDAT(191) viene eseguita come se la condizione di esecuzione fosse un segnale STOP-RUN. I calcoli PID vengono effettuati quando la condizione di esecuzione resta impostata su ON per il ciclo successivo all'inizializzazione dei canali compresi tra C+11 e C+40. Di conseguenza, quando si utilizza il flag di sempre ON come condizione di esecuzione per PIDAT(191), all'avvio dell'operazione è opportuno fornire un'elaborazione distinta per l'inizializzazione di tali canali.

Se i dati in C non rientrano nell'intervallo, si verificherà un errore e verrà attivato il flag di errore.

Se si è verificato un errore durante l'autotuning, verrà attivato il flag di errore.

Se il periodo di campionamento effettivo supera di oltre il doppio il periodo di campionamento designato, si verificherà un errore e verrà attivato il flag di errore. Il controllo PID viene comunque eseguito.

Il flag di riporto viene attivato in fase di esecuzione del controllo PID.

Il flag di maggiore viene attivato se la variabile manipolata dopo l'azione PID supera il limite superiore. In questa fase, i risultati vengono inviati al limite superiore.

Il flag di minore viene attivato se la variabile manipolata dopo l'azione PID è al di sotto del limite inferiore. In questa fase, i risultati vengono inviati al limite inferiore.

Impostazioni dei parametri PID

| Dati di controllo | Elemento | Contenuto | Intervallo di impostazione | Modifica con condizione di ingresso ON |
|-----------------------|---|--|---|---|
| C | Valore impostato (SV) | Il valore di riferimento del processo controllato. | Dati binari (dello stesso numero di bit specificato per il campo di ingresso) | Consentita |
| C+1 | Banda proporzionale | Il parametro per l'azione P che rappresenta l'intervallo di controllo proporzionale/totale. | Da 0001 a 270F esadecimale (da 1 a 9999); (da 0,1% a 999,9%, in unità di 0,1%) | Modifiche consentite con condizione di ingresso ON, se il bit 1 di C+5 è 1. |
| C+2 | Tik Costante integrale | Una costante che rappresenta la forza dell'azione integrale. Se questo valore aumenta, la forza integrale diminuisce. | Da 0001 a 1FFF esadecimale (da 1 a 8191); (9999 = operazione integrale non eseguita - vedere nota 1) | |
| C+3 | Tdk Costante derivativa | Una costante che rappresenta la forza dell'azione derivativa. Se questo valore aumenta, la forza derivativa diminuisce. | Da 0001 a 1FFF esadecimale (da 1 a 8191); (0000 = operazione derivativa non eseguita - vedere nota 1) | |
| C+4 | Periodo di campionamento (τ) | Imposta il periodo di esecuzione dell'azione PID. | Da 0001 a 270F esadecimale (da 1 a 9999); (da 0,01 a 99,99 s, in unità di 10 ms) | Non consentita |
| Bit da 04 a 15 di C+5 | Parametro 2-PID (α) | Il coefficiente del filtro di ingresso. Di norma utilizzare 0,65 (ossia un'impostazione di 000). L'efficienza del filtro diminuisce quando il coefficiente si avvicina allo 0. | 000 esadecimale: $\alpha = 0,65$ L'impostazione da 100 a 163 esadecimale significa che il valore delle due cifre all'estrema destra viene definito da $\alpha = 0,00$ ad $\alpha = 0,99$ (vedere nota 2) | |
| Bit 03 di C+5 | Designazione dell'uscita della variabile manipolata | Designa l'uscita della variabile manipolata quando il valore attuale (PV) è uguale al valore impostato (SV). | 0: uscita 0% 1: uscita 50% | |

| Dati di controllo | Elemento | Contenuto | Intervallo di impostazione | Modifica con condizione di ingresso ON |
|-----------------------|---|--|--|--|
| Bit 01 di C+5 | Impostazione che abilita la modifica della costante PID | I momenti in cui è consentito apportare modifiche alla banda proporzionale (P), alla costante integrale (Tik) e alla costante derivativa (Tdk) per utilizzarle nei calcoli PID. | 0: all'inizio dell'esecuzione dell'istruzione PID 1: all'inizio dell'esecuzione dell'istruzione PID e a ogni periodo di campionamento | Consentita |
| Bit 00 di C+5 | Designazione diretta/inversa per PID | Determina la direzione dell'azione proporzionale. | 0: azione inversa 1: azione diretta | Non consentita |
| Bit 12 di C+6 | Controllo limite di uscita della variabile manipolata | Determina l'eventuale controllo limite applicato all'uscita della variabile manipolata. | 0: disabilitato (nessun controllo limite) 1: abilitato (controllo limite) | |
| Bit da 08 a 11 di C+6 | Campo di ingresso | Il numero dei bit dei dati di ingresso. | 0: 8 bit 5: 13 bit 1: 9 bit 6: 14 bit 2: 10 bit 7: 15 bit 3: 11 bit 8: 16 bit 4: 12 bit | |
| Bit da 04 a 07 di C+6 | Unità integrale e derivativa | Determina l'unità che rappresenta le costanti integrali e derivate. | 1: multiplo del periodo di campionamento 9: tempo (unità: 100 ms) | |
| Bit da 00 a 03 di C+6 | Campo di uscita | Il numero dei bit dei dati di uscita (per impostazione predefinita, il numero dei bit di uscita equivale al numero dei bit di ingresso). | 0: 8 bit 5: 13 bit 1: 9 bit 6: 14 bit 2: 10 bit 7: 15 bit 3: 11 bit 8: 16 bit 4: 12 bit | |
| C+7 | Limite inferiore di uscita della variabile manipolata | Il limite inferiore nel caso in cui sia abilitato il limite di uscita della variabile manipolata. | Da 0000 a FFFF (binario) (vedere la nota 3). | |
| C+8 | Limite superiore di uscita della variabile manipolata | Il limite superiore nel caso in cui sia abilitato il limite di uscita della variabile manipolata. | Da 0000 a FFFF (binario) (vedere la nota 3). | |
| Bit 15 di C+9 | Bit di comando AT | Questo bit di controllo avvia l'autotuning. <ul style="list-style-type: none"> Per eseguire l'autotuning, impostare il bit di comando AT su 1. L'autotuning può venire avviato durante l'esecuzione di PIDAT(191). Al termine dell'autotuning, il bit viene automaticamente impostato su OFF. Se il bit di comando AT viene disattivato manualmente, l'autotuning si arresta. In tal caso, le costanti PID vengono abilitate se già calcolate al momento dell'interruzione dell'autotuning. | Come bit di controllo: <ul style="list-style-type: none"> 0 → 1: esegue l'autotuning. 1 → 0: interrompe l'autotuning. (PID(191) disattiva il bit viene automaticamente al termine dell'autotuning. Come flag: 0: autotuning non in esecuzione 1: autotuning in esecuzione | Consentita |

| Dati di controllo | Elemento | Contenuto | Intervallo di impostazione | Modifica con condizione di ingresso ON |
|-----------------------|---------------------------|---|---|--|
| Bit da 00 a 11 di C+9 | Guadagno del calcolo AT | <p>Impostare questo parametro per regolare il contributo dei risultati del calcolo PID ai valori memorizzati.</p> <p>Di norma, non modificare l'impostazione predefinita di questo parametro (0000).</p> <ul style="list-style-type: none"> • Se si privilegia la stabilità, aumentare il valore. • Se si privilegiano i tempi brevi di risposta, diminuire il valore. | <p>0000 esadecimale: 1.00 (impostazione predefinita)</p> <p>Da 0001 a 03E8 esadecimale (da 1 a 1000) (da 0,01% a 10,00%, in unità di 0,01)</p> | <p>Consentita</p> <p>La lettura di questi parametri avviene all'avvio dell'autotuning.</p> |
| C+10 | Isteresi del ciclo limite | <p>Una volta generato il ciclo limite, impostare l'isteresi. L'impostazione predefinita per il funzionamento inverso attiva la MV con un'isteresi pari a SV-20%.</p> <p>Se non è possibile generare un ciclo limite appropriato a causa dell'instabilità del valore attuale, aumentare l'impostazione. Se comunque il ciclo limite dell'isteresi è maggiore del necessario, la precisione diminuisce.</p> | <p>0000 esadecimale: 0,20% (impostazione predefinita)</p> <p>Da 0001 a 03E8 esadecimale: da 0,01% a 10,00%, in unità di 0,01%</p> <p>FFFF esadecimale: 0.00%</p> <p>Nota La percentuale è in relazione al campo di ingresso.</p> | |

- Nota**
1. Quando l'unità è designata come 1, l'intervallo varia da 1 a 8.191 volte il periodo. Quando l'unità è designata come 9, l'intervallo è compreso tra 0,1 e 819,1 s. Se viene designato 9, impostare i tempi integrale e derivativo in un intervallo da 1 a 8.191 volte il periodo di campionamento.
 2. L'impostazione del parametro 2-PID (α) su 000 produce 0,65, il valore normale.
- Quando il controllo limite di uscita della variabile manipolata è abilitato (cioè è impostato su "1"), definire i valori come indicato qui di seguito.
- $$0000 \leq \text{Limite inferiore uscita MV} \leq \text{Limite superiore uscita MV} \leq \text{Valore massimo campo di uscita}$$

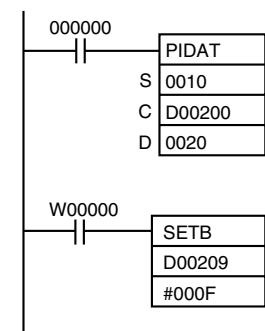
Esempio 1: Interruzione del controllo PID per eseguire l'autotuning

Al fronte di salita di CIO 000000 (da OFF a ON), l'area di lavoro da D00211 a D00240 viene inizializzata in base ai parametri impostati da D00200 a D00208 illustrati qui di seguito. In seguito all'inizializzazione dell'area di lavoro viene eseguito il controllo PID e la variabile manipolata viene inviata a CIO 0020.

Mentre CIO 000000 è ON, viene eseguito il controllo PID agli intervalli del periodo di campionamento in base ai parametri impostati da D00200 a D00210. La variabile manipolata viene inviata a CIO 0020.

Se la banda proporzionale (P), la costante integrale (Tik) o la costante derivativa vengono modificate dopo l'attivazione di CIO 000000, le costanti PID utilizzate nei calcoli PID restano inalterate.

Al fronte di salita di W 000000 (da OFF a ON), SETB(532) attiva il bit 15 di D00209 (C+9) e avvia l'autotuning. Al completamento dell'autotuning, le costanti calcolate P, I e D vengono scritte in C+1, C+2 e C+3. Infine, il controllo PID viene riavviato con le nuove costanti.



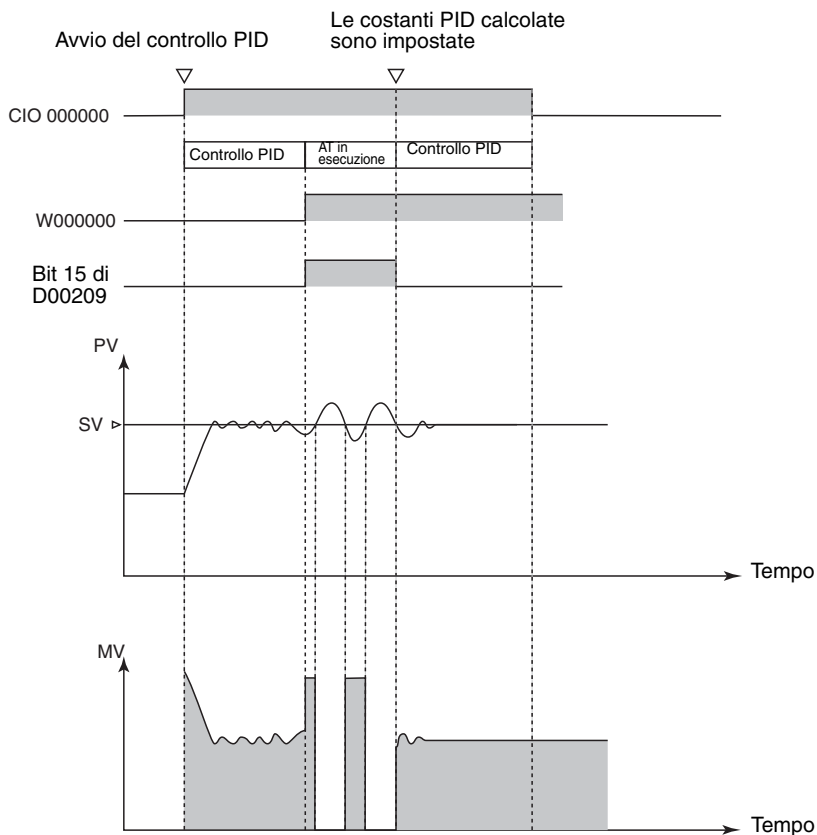
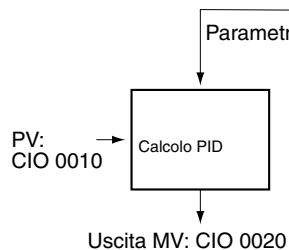
| | | | | |
|--------------|----------------|---|---|---|
| C: D00200 | 0 | 1 | 2 | C |
| C+1: D00201 | 0 | 0 | 6 | 4 |
| C+2: D00202 | 0 | 4 | B | 0 |
| C+3: D00203 | 0 | 1 | 9 | 0 |
| C+4: D00204 | 0 | 0 | 3 | 2 |
| C+5: D00205 | 0 | 0 | 0 | 8 |
| C+6: D00206 | 0 | 4 | 9 | 4 |
| C+7: D00207 | 0 | 0 | 0 | 0 |
| C+8: D00208 | 0 | 0 | 0 | 0 |
| C+9: D00209 | 0 | 0 | 0 | 0 |
| C+10: D00210 | 0 | 0 | 0 | 0 |
| C+11: D00211 | Area di lavoro | | | |
| ... | | | | |
| C+40: D00240 | Area di lavoro | | | |

Valore impostato: 300
 Banda proporzionale: 10.0%
 Tempo integrale: 120,0 s
 Tempo derivativo: 40,0 s
 Periodo di campionamento: 0,5 s

Funzionamento inverso (bit 00: 0), impostazione che abilita la modifica della costante PID = OFF (bit 01: 0), valore impostato = uscita della variabile manipolata 50% (bit 03: 1), parametro 2-PID = 0.65 (bit da 04 a 15: 000 esadecimale)

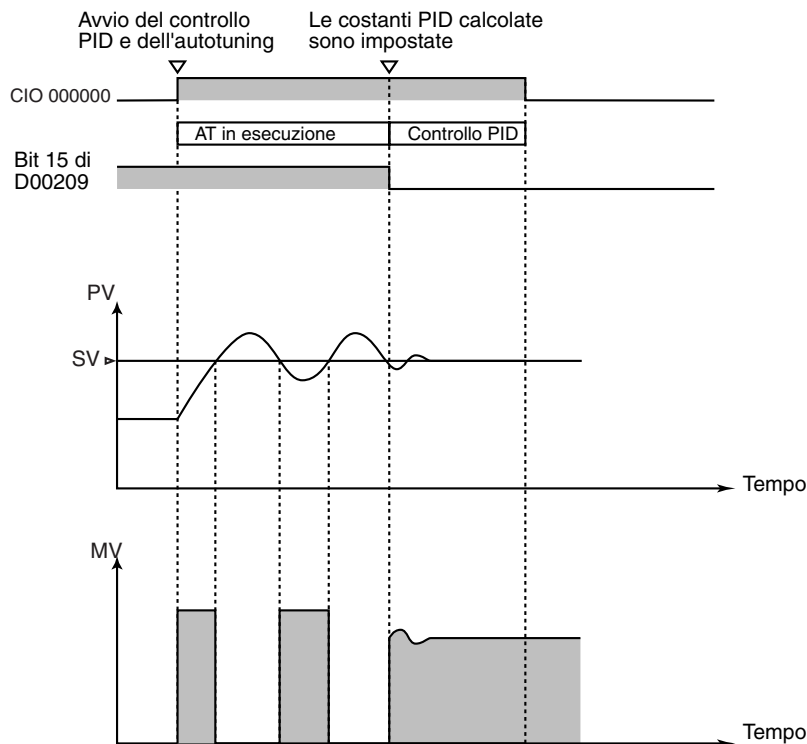
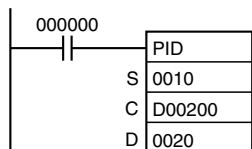
Campo di uscita della variabile manipolata: 12 bit (bit da 00 a 03: 4 esadecimale)
 Costante integrale/derivativa: designazione tempo (bit da 04 a 07: 9 esadecimale)
 Campo di ingresso: 12 bit (bit da 08 a 11: 4 esadecimale)
 Controllo disattivato del limite di uscita della variabile manipolata (bit 12:0)

Bit di comando AT (bit 15 di 0)
 Guadagno del calcolo AT = 1.00 (bit da 00 11: 000 esadecimale)
 Isteresi del ciclo limite = 0.20%



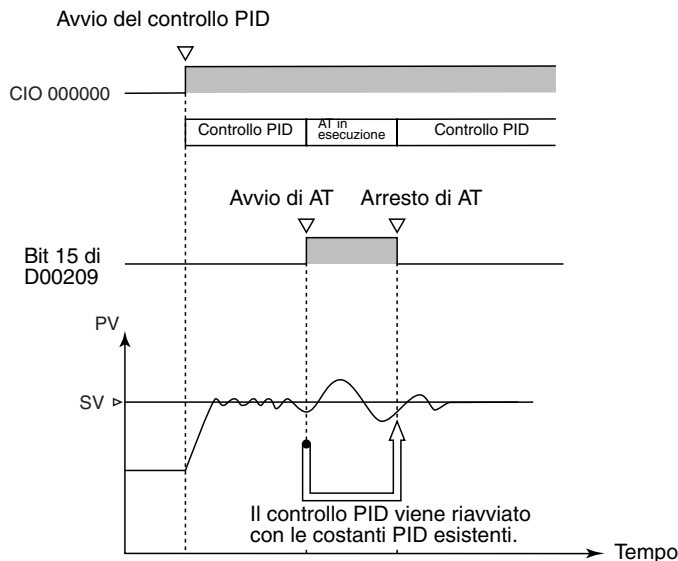
**Esempio 2:
Avvio di PIDAT(191) con
autotuning**

Al fronte di salita di CIO 000000 (da OFF a ON), l'autotuning verrà eseguito per primo se il bit 15 di D00209 (C+9) è ON. Al completamento dell'autotuning, le costanti calcolate P, I e D vengono scritte in C+1, C+2 e C+3. Infine, il controllo PID viene avviato con le costanti PID calcolate.



**Esempio 3:
Interruzione
dell'autotuning prima del
completamento**

Disattivando il bit 15 di D00209 (C+9), è possibile interrompere l'autotuning. Il controllo PID verrà quindi riavviato con le costanti P, I e D in uso prima dell'avvio dell'autotuning.



3-18-3 LIMIT CONTROL: LMT(680)

Scopo

Controlla i dati di uscita in base all'appartenenza dei dati di ingresso all'intervallo delimitato dai limiti superiore e inferiore.

Simbolo programmazione ladder

| | | |
|---|----------|------------------------|
| — | LMT(680) | |
| | S | S: canale di ingresso |
| | C | C: primo canale limite |
| | D | D: canale di uscita |

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | LMT(680) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @LMT(680) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | C | D |
|--------------------------------------|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 | Da A000 ad A958 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |

| Area | S | C | D |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Quando la condizione di esecuzione è impostata su ON, LMT(680) controlla i dati di uscita in base all'appartenenza dei dati di ingresso specificati (in formato binario con segno) all'intervallo delimitato dai limiti superiore e inferiore. Il contenuto dei canali C e C+1 è il seguente:

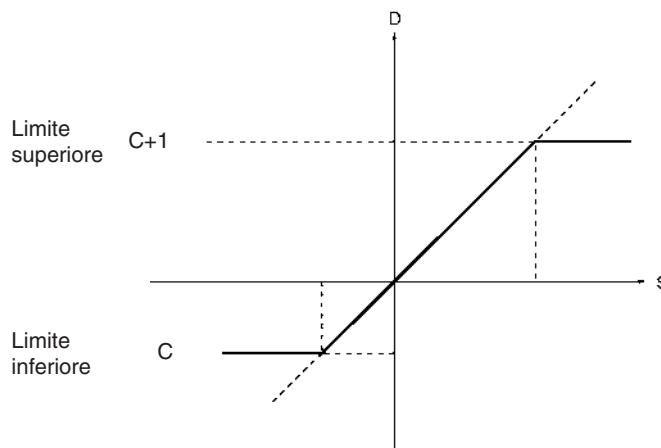
| | |
|-----|--|
| C | Dati del limite inferiore (dati minimi di uscita) |
| C+1 | Dati del limite superiore (dati massimi di uscita) |

È necessario che la classificazione di area di C e C+1 sia la stessa.

Se i dati di ingresso (S) sono minori del limite inferiore (C), i dati del limite inferiore sono inviati a D e il flag di minore viene attivato.

Se i dati di ingresso (S) sono maggiori del limite superiore (C+1), i dati del limite superiore sono inviati a D e il flag di maggiore viene attivato.

Se i dati di ingresso (S) sono maggiori o uguali al limite inferiore (C) e minori o uguali al limite superiore (C+1), i dati di ingresso (S) vengono inviati a D.



Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | Viene attivato se il limite superiore è minore del limite inferiore. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se i dati di ingresso (S) sono maggiori del limite superiore. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se i dati di ingresso (S) sono minori del limite inferiore. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è "1". OFF in tutti gli altri casi. |

Avvertenze

Se il limite superiore è minore del limite inferiore, si verificherà un errore e verrà attivato il flag di errore.

Se i dati di ingresso (S) sono maggiori del limite superiore, verrà attivato il flag di maggiore.

Se il canale di uscita D è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se i dati di ingresso (S) sono minori del limite inferiore, verrà attivato il flag di minore.

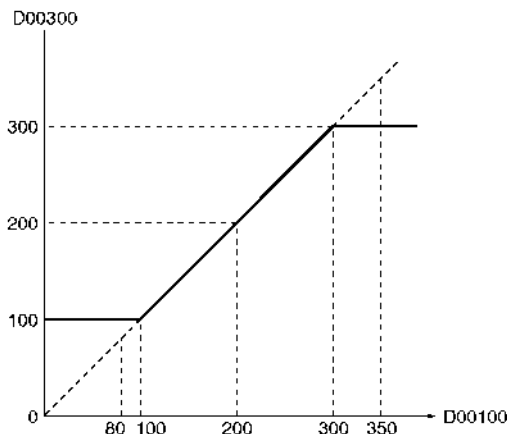
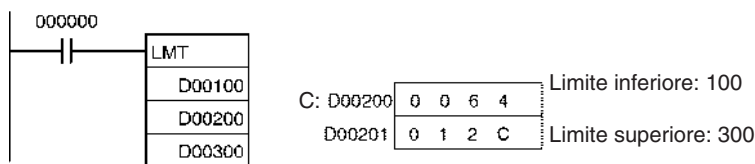
Se lo stato del bit più a sinistra del canale di uscita D è "1", verrà attivato il flag negativo.

Esempio

Se D00100 è 0050 esadecimale (80), 0064 esadecimale (100) viene inviato a D00300 poiché 80 è minore del limite inferiore di 100.

Se D00100 è 00C8 esadecimale (200), 0064 esadecimale (100) viene inviato a D00300 poiché 200 rientra nell'intervallo delimitato dai limiti superiore e inferiore.

Se D00100 è 012C esadecimale (300), 015E esadecimale (350) viene inviato a D00300 poiché 350 è maggiore del limite superiore di 300.

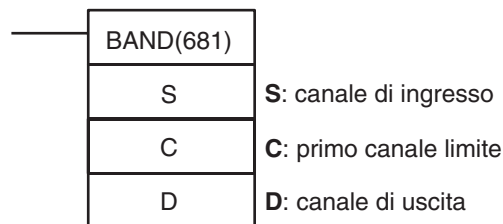


3-18-4 DEAD BAND CONTROL: BAND(681)

Scopo

Controlla i dati di uscita in base all'appartenenza dei dati di ingresso alla zona morta e cioè all'intervallo compreso tra il limite inferiore e superiore.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BAND(681) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BAND(681) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | C | D |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A958 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Quando la condizione di esecuzione è impostata su ON, BAND(681) controlla i dati di uscita in base all'appartenenza dei dati di ingresso specificati (in formato binario con segno) alla zona morta, cioè all'intervallo delimitato dai limiti superiore e inferiore. Il contenuto dei canali C e C+1 è il seguente:

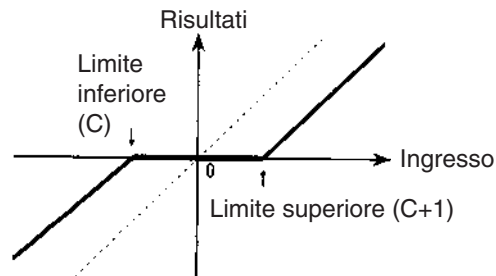
| | |
|-----|---|
| C | Dati del limite inferiore (limite inferiore della zona morta) |
| C+1 | Dati del limite superiore (limite superiore della zona morta) |

È necessario che la classificazione di area di C e C+1 sia la stessa.

Se i dati di ingresso (S) sono maggiori o uguali al limite inferiore (C) e minori o uguali al limite superiore (C+1), 0000 esadecimale è inviato a D e viene attivato il flag di uguaglianza.

Se i dati di ingresso (S) sono minori del limite inferiore (C), la differenza tra i dati di ingresso meno i dati del limite inferiore è inviata a D e il flag di minore viene attivato.

Se i dati di ingresso (S) sono maggiori del limite superiore (C+1), la differenza tra i dati di ingresso meno i dati del limite superiore è inviata a D e il flag di maggiore viene attivato.



Se i dati di ingresso sono minori di 8000 esadecimale o maggiori di 7FFF, viene invertito il segno. Ad esempio, per un limite inferiore di 0100 esadecimale e dati di ingresso di 8000 esadecimale, i dati di uscita saranno così:

8000 (esadecimale) [-32768] - 0100 (esadecimale) [256] = 7F00 (esadecimale) [32512]

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | Viene attivato se il limite superiore è minore del limite inferiore. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se i dati di ingresso (S) sono maggiori del limite superiore. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se i dati di ingresso (S) sono minori del limite inferiore. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è "1". OFF in tutti gli altri casi. |

Avvertenze

Se il limite superiore è minore del limite inferiore, si verificherà un errore e verrà attivato il flag di errore.

Se i dati di ingresso (S) sono maggiori del limite superiore, verrà attivato il flag di maggiore.

Se il canale di uscita D è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se i dati di ingresso (S) sono minori del limite inferiore, verrà attivato il flag di minore.

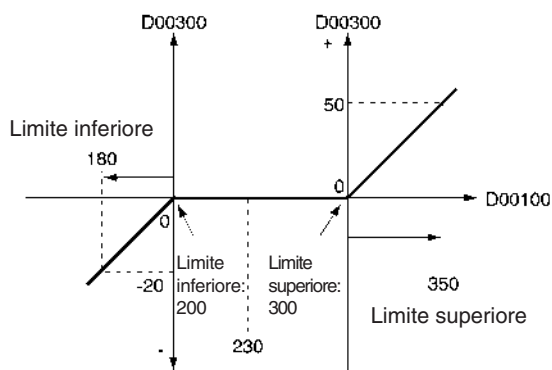
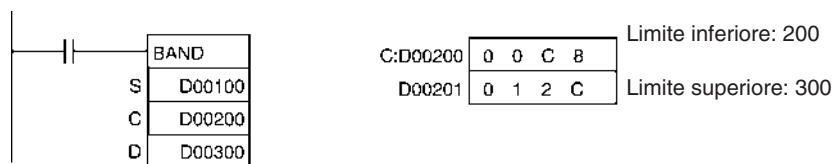
Se lo stato del bit più a sinistra del canale di uscita D è "1", verrà attivato il flag negativo.

Esempio

Se D00100 è 00B4 esadecimale (180), $180 - 200 = \text{FFEC}$ esadecimale (-20) viene inviato a D00300 poiché 180 è minore del limite inferiore di 200.

Se D00100 è 00E6 esadecimale (230), 0 viene inviato a D00300 poiché 230 rientra nell'intervallo delimitato dai limiti superiore e inferiore.

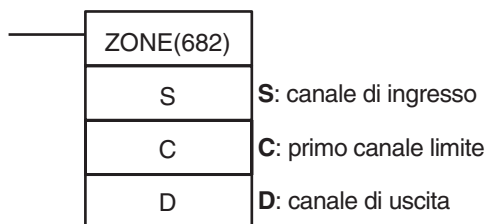
Se D00100 è 015E esadecimale (350), 350-300 = 0032 esadecimale (50) viene inviato a D00300 poiché 350 è maggiore del limite superiore di 300.



3-18-5 DEAD ZONE CONTROL: ZONE(682)

Scopo Aggiunge la polarizzazione specificata ai dati di ingresso e restituisce il risultato.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ZONE(682) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ZONE(682) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Caratteristiche operando

| Area | S | C | D |
|-------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A958 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | Da T0000 a T4095 |

| Area | S | C | D |
|--|--|--|--|
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Quando la condizione di esecuzione è impostata su ON, ZONE(682) aggiunge la polarizzazione specificata ai dati di ingresso specificati (in formato binario a 16 bit con segno) e invia il risultato in un determinato canale. Il contenuto dei canali C e C+1 è il seguente:

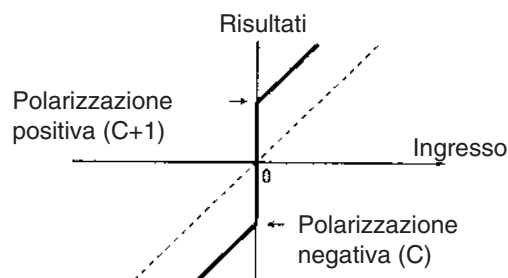
| | |
|-----|-------------------------|
| C | Polarizzazione negativa |
| C+1 | Polarizzazione positiva |

È necessario che la classificazione di area di C e C+1 sia la stessa.

Se i dati di ingresso (S) sono minori di 0, i dati di ingresso più la polarizzazione negativa sono inviati a D e il flag di minore viene attivato.

Se i dati di ingresso (S) sono maggiori di 0, i dati di ingresso più la polarizzazione positiva sono inviati a D e il flag di maggiore viene attivato.

Se i dati di ingresso (S) sono uguali a zero, 0000 è inviato a D il flag di uguaglianza viene attivato.



Se i dati di ingresso sono minori di 8000 esadecimale o maggiori di 7FFF, viene invertito il segno. Ad esempio, per un valore di polarizzazione negativa di FF00 esadecimale e dati di ingresso di 8000 esadecimale, i dati di uscita saranno così:

8000 (esadecimale) [-32768] - FF00 (esadecimale) [-256] = 7F00 (esadecimale) [32512]

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|--|
| Flag di errore | ER | Viene attivato se il limite superiore è minore del limite inferiore. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se i dati di ingresso (S) sono maggiori del limite superiore. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se i dati di ingresso (S) sono minori del limite inferiore. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando il bit più a sinistra del risultato è "1". OFF in tutti gli altri casi. |

Avvertenze

Se il limite superiore è minore del limite inferiore, si verificherà un errore e verrà attivato il flag di errore.

Se i dati di ingresso (S) sono maggiori del limite superiore, verrà attivato il flag di maggiore.

Se il canale di uscita D è 0000 esadecimale, verrà attivato il flag di uguaglianza.

Se i dati di ingresso (S) sono minori del limite inferiore, verrà attivato il flag di minore.

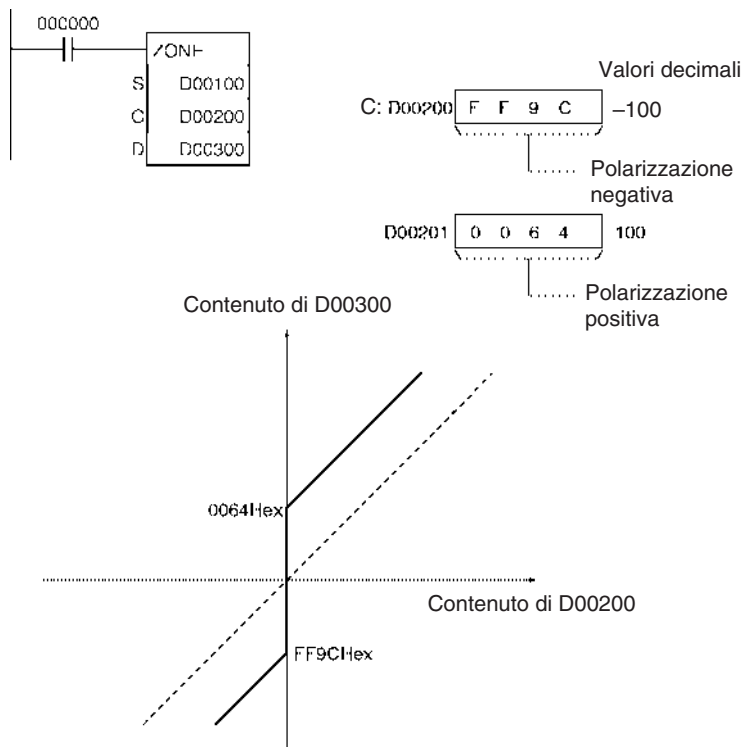
Se lo stato del bit più a sinistra del canale di uscita D è "1", verrà attivato il flag negativo.

Esempio

Quando CIO 000000 è ON, una polarizzazione di -100 viene applicata al valore di D00100, purché questo valore sia minore di 0, e il risultato è memorizzato in D00300.

Se il valore di D00100 è 0, 0000 esadecimale viene memorizzato in D00300.

Se il valore di D00100 è maggiore di 0, viene applicata una polarizzazione di +100 e il valore risultante è memorizzato in D00300.



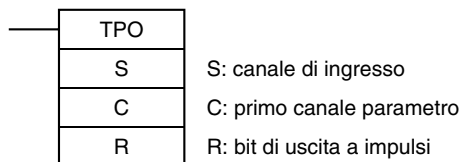
3-18-6 TIME-PROPORTIONAL OUTPUT: TPO(685)

Scopo

A partire dal rapporto di funzionamento o da una variabile manipolata contenuta nel canale specificato, converte il rapporto di funzionamento in un'uscita proporzionale al tempo in base ai parametri specificati ed emette il risultato dall'uscita specificata.

Questa istruzione è supportata solo dalla CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TPO(685) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | OK |

Operandi

S: canale di ingresso

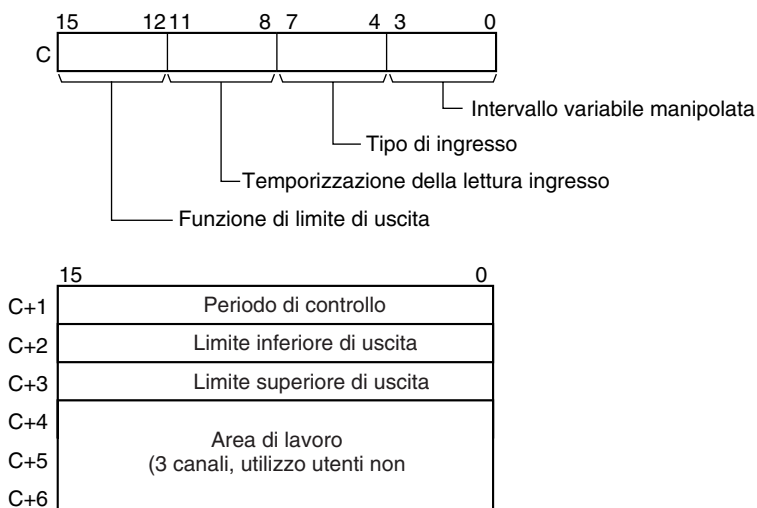
Specifica il canale di ingresso che contiene il rapporto di funzionamento di ingresso o la variabile manipolata. I bit da 04 a 07 di C indicano il tipo di ingresso, cioè se il contenuto del canale di ingresso sia rappresentato da un rapporto di funzionamento o da una variabile manipolata. Per specificare un rapporto di funzionamento impostare questi bit su 0 esadecimale; impostarli su 1 per specificare una variabile manipolata.

- Rapporto di funzionamento di ingresso: da 0000 a 2710 esadecimale (da 0,00% a 100,00%)
- Variabile manipolata di ingresso (vedere nota): da 0000 a FFFF esadecimale (da 0 a 65.535 max.). I bit da 00 a 03 di C indicano l'intervallo della variabile manipolata, vale a dire il numero dei bit validi nella variabile stessa. Indicare il numero di bit uguale a quello specificato per l'impostazione del campo di uscita PID(190).

Nota Se S è una variabile manipolata, indicare un canale che contiene l'uscita della variabile manipolata da un'istruzione PID(190) o PIDAT(191).

da C a C+6: parametri

Nel diagramma seguente sono indicate le posizioni dei dati di parametro. Per ulteriori dettagli sui parametri, consultare *Impostazione dei parametri* in questa sezione.



Nota: per ulteriori informazioni, fare riferimento alla descrizione di ciascun parametro.

R: bit di uscita a impulsi

Specifica il bit di uscita di destinazione per l'uscita a impulsi.

Di norma indica un bit di uscita allocato a un modulo di uscita a transistor e collega un relè a stato solido al modulo.

Caratteristiche operando

| Area | S | C | R |
|--------------------------------------|--|--|----------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6137 | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W000 a W511 | Da W000 a W505 | Da W00000 a W51115 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H505 | Da H00000 a H51115 |
| Area bit ausiliaria | Da A000 a 959 | Da A000 ad A953 | Da A44800 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4089 | --- |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4089 | --- |
| Area DM | Da D00000 a D32767 | Da D00000 a D32761 | --- |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32761 | --- |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32761 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | --- |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | --- |
| Costanti | Da #0000 a #FFFF (binario) | --- | --- |
| Registri dati | Da DR0 a DR15 | --- | --- |

| Area | S | C | R |
|--|--|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Riceve un ingresso di rapporto di funzionamento o di variabile manipolata dall'indirizzo del canale specificato da S, converte il rapporto di funzionamento in un'uscita proporzionale al tempo (vedere nota) in base ai parametri indicati nei canali compresi tra C e C+3 ed emette un'uscita a impulsi al bit specificato da R.

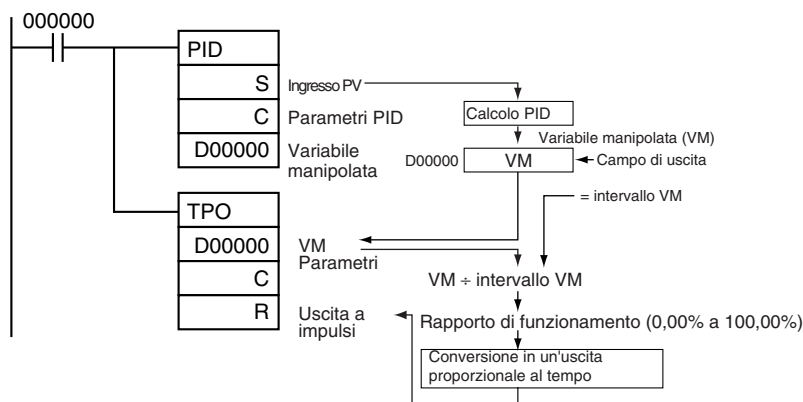
Nota Un'uscita proporzionale al tempo viene modificata in modo proporzionale in base al rapporto ON/OFF nel canale di ingresso S. Si definisce periodo di controllo il periodo di variazione dello stato ON e OFF e lo si imposta nel canale parametro C+1.

Esempio: Quando il periodo di controllo è 1 e il valore di ingresso è 50%, il bit è ON per 0,5 s e OFF per 0,5 s. Quando il periodo di controllo è 1 e il valore di ingresso è 80%, il bit è ON per 0,8 s e OFF per 0,2 s.

In genere, TPO(685) viene utilizzata insieme a PID(190) o PIDAT(191) e il canale del risultato (D) della variabile manipolata dell'istruzione PID viene specificato come il canale di ingresso (S) per l'istruzione TPO(685). Inoltre, un bit di uscita allocato a un modulo di uscita a transistor viene di norma indicato come R e un relè a stato solido è collegato a tale modulo per eseguire un controllo proporzionale al tempo di un elemento riscaldante (controllo proporzionale del rapporto ON/OFF).

Combinazione di TPO(685) con un'istruzione di controllo PID

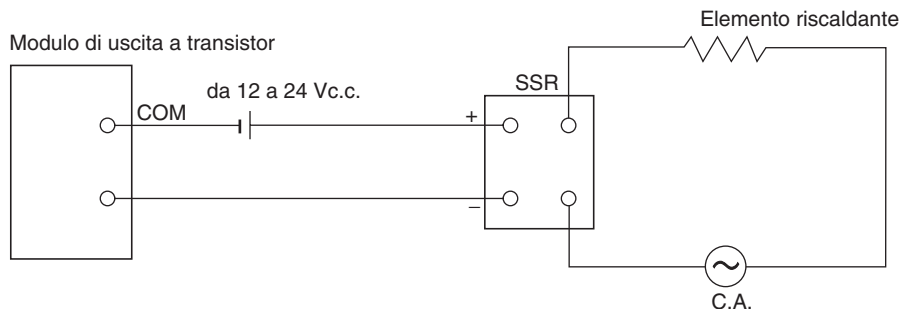
Quando si combinano TPO(685) e un'istruzione di controllo PID, l'ingresso della variabile manipolata viene diviso per l'intervallo della variabile manipolata stessa al fine di calcolare il rapporto di funzionamento; tale rapporto viene convertito in un'uscita proporzionale al tempo e gli impulsi sono inviati in uscita.



In questo caso, è necessario impostare lo stesso valore per il campo di uscita dell'istruzione di controllo PID e l'intervallo della variabile manipolata dell'istruzione TPO(685). Ad esempio, quando il campo di uscita dell'istruzione di controllo PID e l'intervallo della variabile manipolata dell'istruzione TPO(685) sono entrambi impostati su 12 bit (da 0000 a 0FFF esadecimale), il rapporto di funzionamento viene calcolato dividendo la variabile manipolata dell'istruzione di controllo PID per 0FFF esadecimale e TPO(685) converte tale rapporto di funzionamento in un'uscita proporzionale al tempo.

Esempio di cablaggio esterno

Collegare un modulo di uscita a transistor a un relè a stato solido (SSR), come indicato nel seguente diagramma.

**Impostazioni dei parametri**

| Dati di controllo | | Elemento | Contenuto | Intervallo di impostazione | Modifica con condizione di ingresso ON |
|-------------------|-----------|--|---|--|--|
| Canale | Bit | | | | |
| C | 00 ... 03 | Intervallo variabile manipolata | Indica il numero dei bit dei dati di ingresso. | 0 esadecimale: 8 bit 5 esadecimale: 13 bit 1 esadecimale: 9 bit 6 esadecimale: 14 bit 2 esadecimale: 10 bit 7 esadecimale: 15 bit 3 esadecimale: 11 bit 8 esadecimale: 16 bit 4 esadecimale: 12 bit | Consentita |
| | 04 ... 07 | Tipo di ingresso | Indica se S contiene un rapporto di funzionamento o una variabile manipolata. | 0 esadecimale: rapporto di funzionamento Intervallo d'impostazione per S: da 0000 a 2710 esadecimale (da 0,00% a 100,00%) 1 esadecimale: variabile manipolata Intervallo d'impostazione per S: da 0000 a FFFF esadecimale (da 0 a 65.535) L'impostazione massima dipende dall'intervallo VM impostato per i bit da 00 a 03 di C. | Consentita |
| | 08 ... 11 | Temporizzazione della lettura ingresso | Indica la temporizzazione della lettura di ingresso. | 0 esadecimale: utilizzare il valore iniziale del periodo di controllo 1 esadecimale: utilizzare valore inferiore 2 esadecimale: utilizzare valore superiore 3 esadecimale: regolazione continua | Consentita |
| | 12 ... 15 | Controllo limite di uscita | Indica l'attivazione o la disattivazione della funzione di limite di uscita. | 0 esadecimale: disabilitato 1 esadecimale: abilitato (vedere nota). | Consentita |
| C+1 | 00 ... 15 | Periodo di controllo | Periodo di controllo (periodo di tempo in cui vengono apportate le modifiche ON/OFF). | da 0064 a 270F esadecimale (da 1,00 a 99,99 s) Nota: ad esempio, come 0064 viene impostato 1,00 s e non 0001 esadecimale. | Consentita |
| C+2 | 00 ... 15 | Limite inferiore di uscita | Indica il limite inferiore quando avviene l'attivazione del limite di uscita. | da 0000 a 2710 esadecimale (da 0% a 100,00%) | Consentita |

| Dati di controllo | | Elemento | Contenuto | Intervallo di impostazione | Modifica con condizione di ingresso ON |
|-------------------|-----------|----------------------------|--|--|--|
| Canale | Bit | | | | |
| C+3 | 00 ... 15 | Limite superiore di uscita | Indica il limite superiore quando avviene l'attivazione del limite di uscita. | da 0000 a 2710 esadecimale (da 0% a 100,00%) | Consentita |
| C+4 | 00 ... 15 | Area di lavoro | Si tratta dell'area utilizzata dal sistema. Non può essere utilizzata dall'utente. | Impossibile utilizzare | --- |
| C+5 | 00 ... 15 | | | | |
| C+6 | 00 ... 15 | | | | |

Nota Quando la funzione di controllo limite di uscita è attivata, impostare i limiti superiore e inferiore come indicato: 0000 esadecimale \leq limite inferiore \leq limite superiore \leq 2710 esadecimale.

Esecuzione

- L'istruzione viene eseguita mentre la condizione di ingresso è impostata su ON.
- All'avvio dell'esecuzione dell'istruzione il bit di uscita (R) viene impostato su ON/OFF in base al rapporto di funzionamento.
- I parametri (da C a C+3) vengono letti in tempo reale a ogni esecuzione dell'istruzione. Quando si modificano i parametri, è opportuno intervenire su tutti i parametri allo stesso tempo, al fine di evitare confusione tra le varie serie.
- L'uscita (R) viene impostata ON/OFF all'esecuzione dell'istruzione e la precisione della temporizzazione ON/OFF dell'uscita è di 10 ms max.
- L'esecuzione dell'istruzione si arresta quando la condizione di ingresso passa a OFF. In quel momento il valore del tempo trascorso viene reimpostato e il periodo di controllo viene inizializzato.
- L'impostazione del tipo di ingresso (bit da 04 a 07 di C) determina il contenuto del canale di ingresso (S): rapporto di funzionamento o variabile manipolata. Quando S contiene la variabile manipolata, il rapporto di funzionamento viene calcolato dividendo l'ingresso della variabile manipolata per l'intervallo della variabile manipolata stessa (bit da 00 a 03 di C).

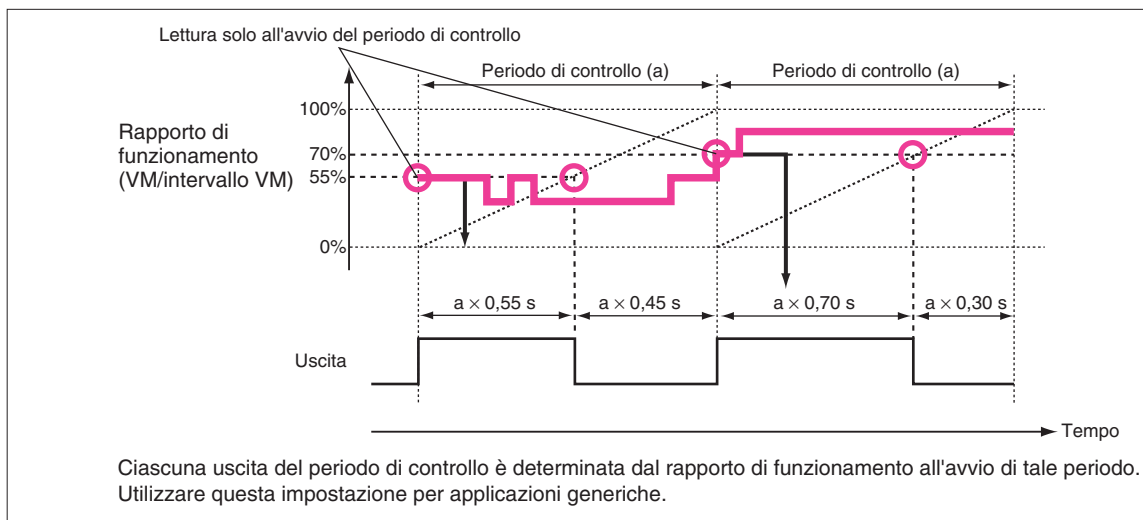
Impostazione della temporizzazione della lettura ingresso (bit da 08 a 11 di C)

L'impostazione della temporizzazione della lettura ingresso (bit da 08 a 11 di C) indica il momento in cui avviene la lettura del canale di ingresso (S). Osservare la tabella qui di seguito.

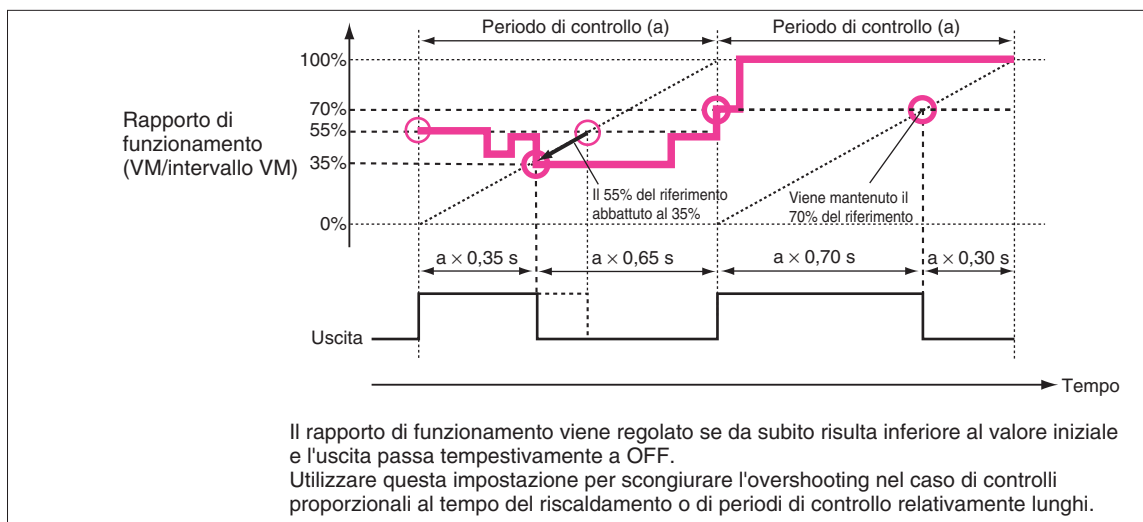
| Temporizzazione della lettura ingresso | Descrizione |
|---|---|
| 0: utilizzare il valore iniziale del periodo di controllo | L'ingresso del rapporto di funzionamento viene letto all'avvio del periodo di controllo, durante il quale non è possibile modificare il rapporto. |
| 1: utilizzare valore inferiore | Se l'ingresso del rapporto di funzionamento è inferiore al rapporto di funzionamento all'avvio del periodo di controllo, il valore inferiore avrà la precedenza e il tempo di attivazione dell'uscita verrà ridotto di conseguenza. |
| 2: utilizzare valore superiore | Se l'ingresso del rapporto di funzionamento è superiore al rapporto di funzionamento all'avvio del periodo di controllo, il valore superiore avrà la precedenza e il tempo di attivazione dell'uscita verrà aumentato di conseguenza. |
| 3: regolazione continua | Il rapporto di funzionamento viene letto in tempo reale a ogni esecuzione dell'istruzione e il funzionamento ON/OFF sarà ripetuto all'interno del periodo di controllo. |

Nei diagrammi seguenti viene illustrato il funzionamento di ciascuna impostazione di temporizzazione della lettura ingresso.

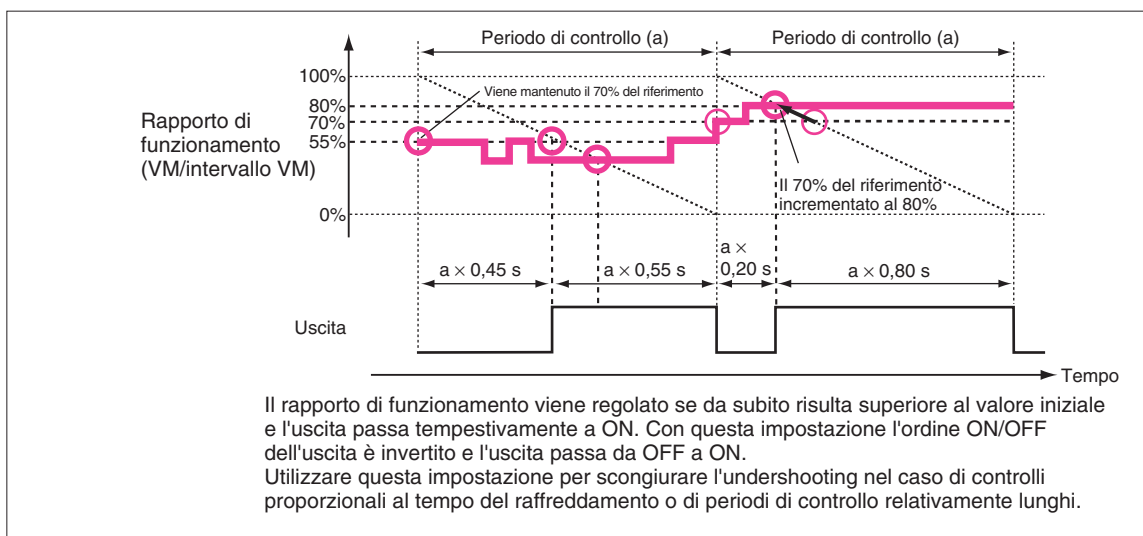
- Impostazione del tempo di ingresso = 0 (utilizzare il valore iniziale del periodo di controllo)



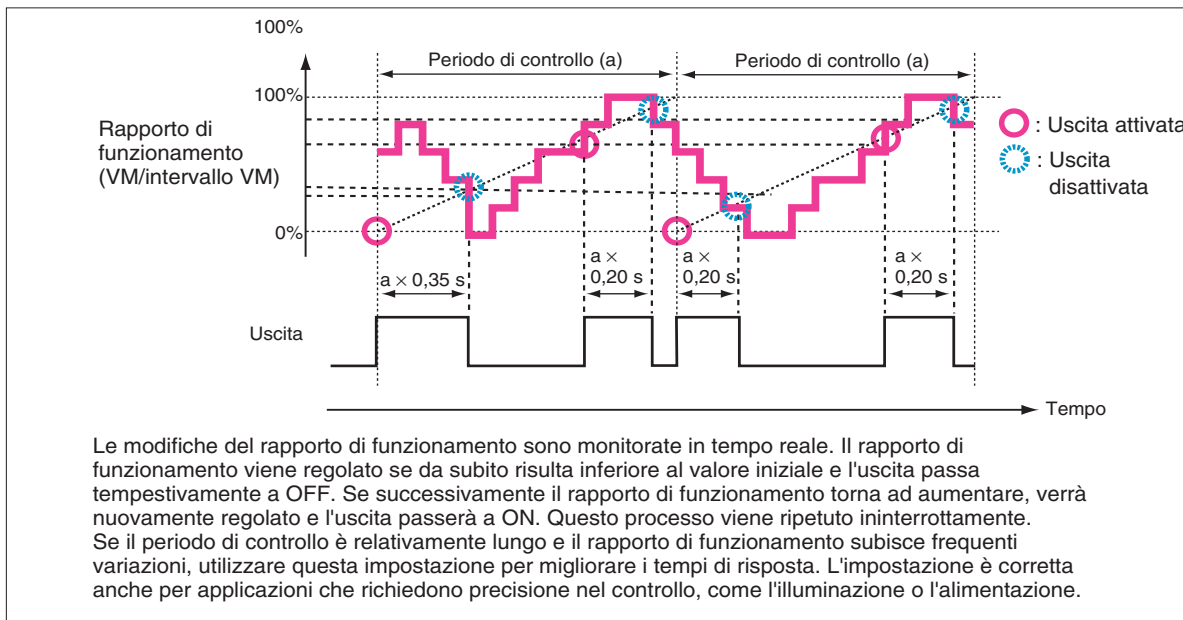
- Impostazione del tempo di ingresso = 1 (utilizzare valore inferiore)



- Impostazione del tempo di ingresso = 2 (utilizzare valore superiore)



- Impostazione del tempo di ingresso = 3 (regolazione continua)



Flag

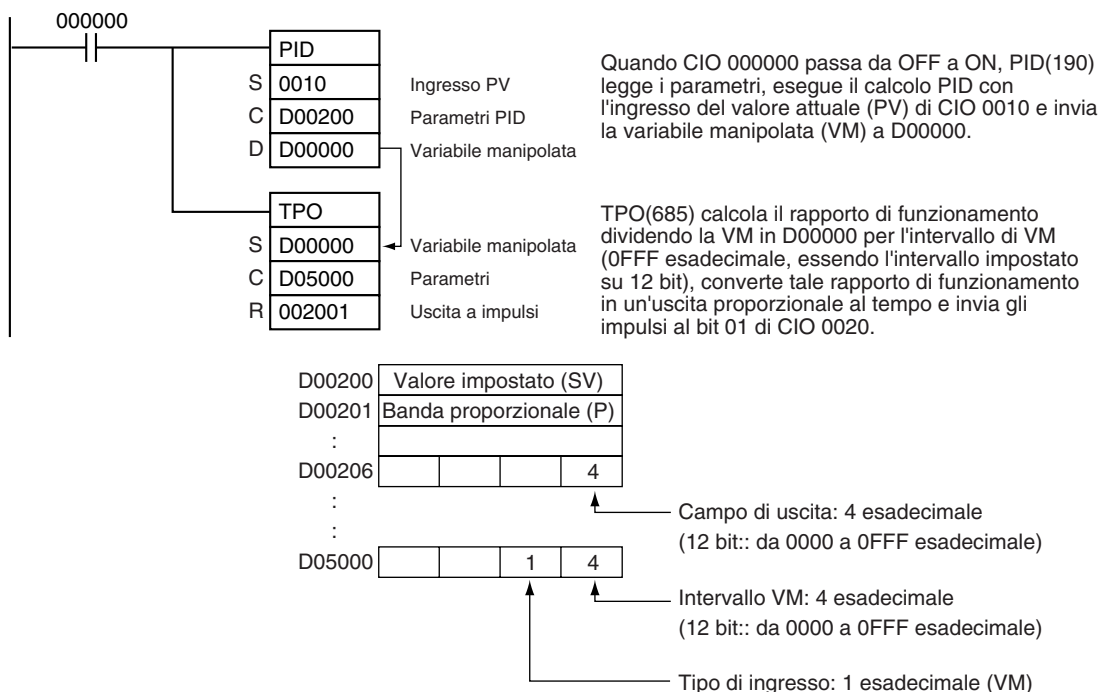
| Nome | Eti-chetta | Funzionamento |
|----------------|------------|---|
| Flag di errore | ER | ON se i dati di ingresso in S non rientrano nell'intervallo. L'intervallo di impostazione dei dati di ingresso dipende dall'impostazione del tipo di ingresso. ON se i dati in C non rientrano nell'intervallo. L'intervallo della variabile manipolata genera un errore solo quando il tipo di ingresso viene impostato su variabile manipolata. ON se il periodo di controllo in C+1 non rientra nell'intervallo. ON se la funzione di limite di uscita è attivata ma il limite inferiore (C+2) o superiore (C+3) di uscita non rientrano nell'intervallo. ON se la funzione di limite di uscita è attivata ma il limite inferiore di uscita (C+2) è minore o uguale al limite superiore di uscita (C+3). OFF in tutti gli altri casi. |

Esempio

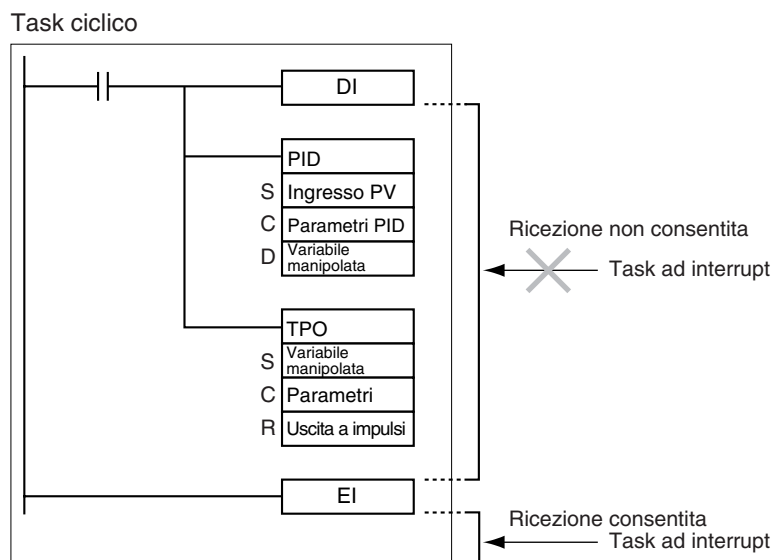
Esempio 1: Combinazione di TPO(685) con PID(190)

Quando CIO 000000 è ON, TPO(685) assume l'uscita della variabile manipolata da PID(190) (in D00000), calcola il rapporto di funzionamento dal valore di tale variabile manipolata (rapporto di funzionamento = VM ÷ intervallo VM), converte il rapporto di funzionamento in un'uscita proporzionale al tempo e invia gli impulsi a CIO 002001.

In questo caso, CIO 0020 è allocato a un modulo di uscita a transistor e il bit CIO 002001 viene collegato a un relè a stato solido per il controllo di un elemento riscaldante.



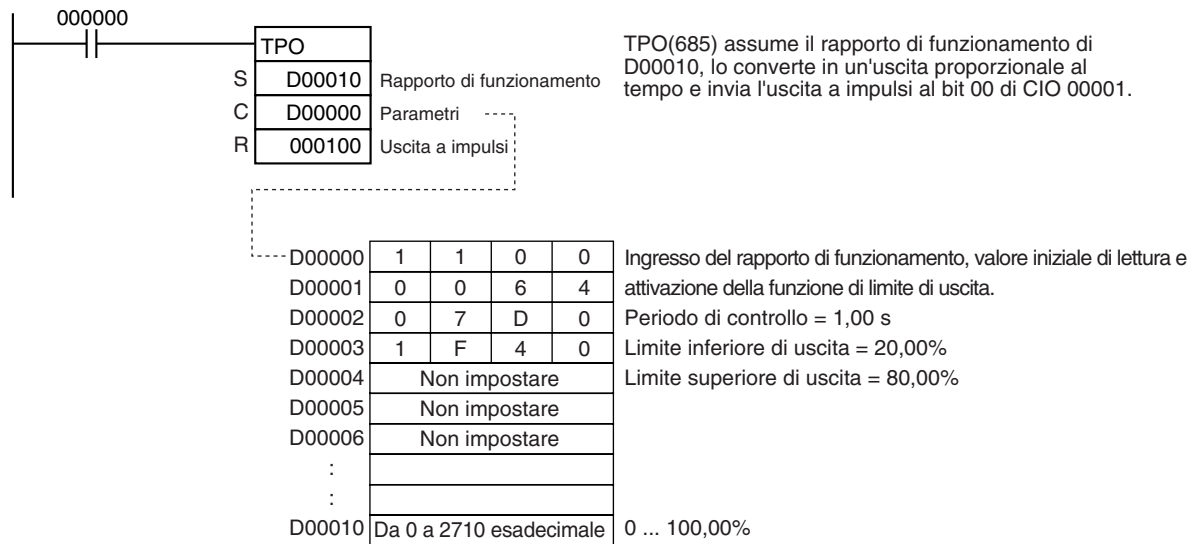
Nota Quando si utilizzano congiuntamente TPO(685) e PID(190) in un task ciclico o in un task ad interrupt, disattivare temporaneamente gli interrupt facendo precedere l'esecuzione di DI(693) (DISABLE INTERRUPTS) a quella di PID(190) e TPO(685). Se gli interrupt non sono disattivati e si verifica un interrupt tra PID(190) e TPO(685), il periodo di controllo potrebbe subire cambiamenti.



Esempio 2: Utilizzo della sola TPO(685)

Quando CIO 000000 è ON, TPO(685) assume il rapporto di funzionamento di D00010, lo converte in un'uscita proporzionale al tempo e invia gli impulsi a CIO 000100.

In questo caso, il periodo di controllo è 1 s e la funzione di limite di uscita viene attivata con un limite inferiore di 20,00% e superiore di 80,00%.

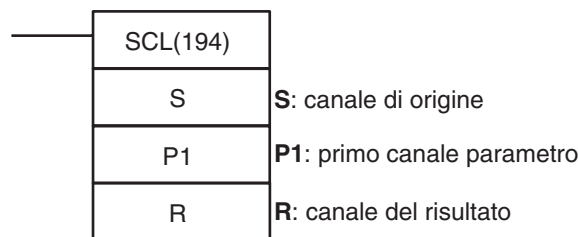


3-18-7 SCALING: SCL(194)

Scopo

Converte dati binari senza segno in dati in formato BCD senza segno in base alla funzione lineare specificata.

Simbolo programmazione ladder



Variazioni

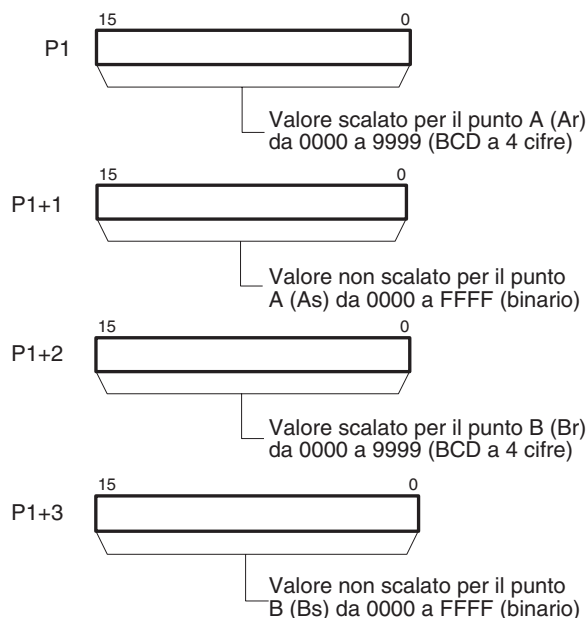
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SCL(194) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SCL(194) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Nel seguente diagramma è indicato il contenuto dei quattro canali a partire dal primo canale parametro (P1).



Nota I canali da P1 e P1+3 devono essere nella stessa area dati.

Caratteristiche operando

| Area | S | P1 | R |
|--------------------------------------|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6140 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W508 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H508 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A956 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4092 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4092 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32764 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32764 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32764 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |

| Area | S | P1 | R |
|--|--|----|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SCL(194) viene utilizzata per convertire i dati binari senza segno contenuti nel canale di origine S in dati BCD senza segno e inviare il risultato nel canale del risultato R in base alla funzione lineare definita dai punti (As, Ad) e (Bs, Bd). L'indirizzo del primo canale contenente le coordinate dei punti (As, Ar) e (Bs, Br) viene specificato per il primo canale parametro P1. Questi punti definiscono due valori (As e Bs) prima della funzione di scala e due valori (Ar e Br) in seguito.

Per la conversione si utilizzano le seguenti equazioni:

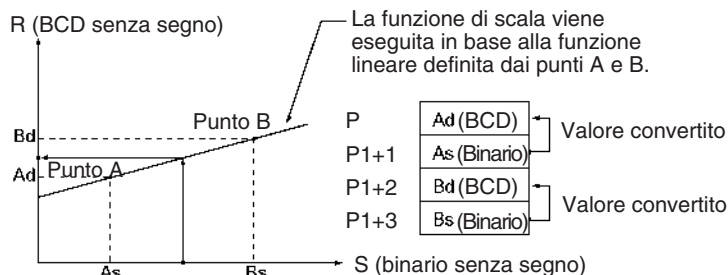
$$R = Bd - \frac{(Bd - Ad)}{\text{Conversione BCD di } (Bs - As)} \times \text{Conversione BCD di } (Bs - S)$$

La pendenza della linea è la seguente:

$$R = Bd - \frac{(Bd - Ad)}{\text{Conversione BCD di } (Bs - As)}$$

I punti A e B possono definire una linea con una pendenza positiva o negativa. L'utilizzo di una pendenza negativa consente la scala invertita.

Il risultato viene arrotondato al numero intero più vicino. Se il risultato è minore di 0000, il risultato inviato sarà 0000. Se il risultato è maggiore di 9999, verrà inviato 9999.



È possibile utilizzare SCL(194) per scalare i risultati dei valori di conversione del segnale analogico da moduli di ingresso analogico in base a parametri di scala definiti dall'utente. Ad esempio, se un ingresso compreso tra 1 e 5 V in un modulo di ingresso analogico viene immesso in memoria come un valore compreso tra 0000 e 0FA0 esadecimale, utilizzando SCL(194) è possibile scalare il valore in memoria da 50 a 200°C.

SCL(194) converte il formato binario senza segno in formato BCD senza segno. Se si desidera convertire un valore negativo, prima di eseguire SCL(194) è necessario aggiungere il valore negativo più alto nel programma. Fare riferimento all'esempio fornito.

Con SCL(194) non è possibile inviare un valore negativo al canale del risultato R. In questo caso il risultato inviato a R sarà 0000.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di C (Ar) o C+1 (Br) non è in formato BCD. ON se il contenuto di C+1 (As) e C+3 (Bs) sono uguali. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |

Avvertenze

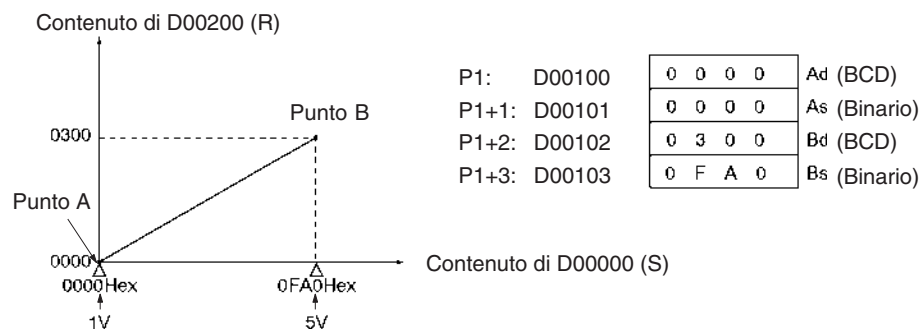
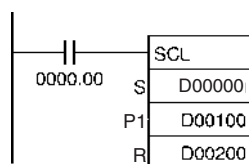
Se i valori per Ar (C) e Br (C+2) non sono in formato BCD o se i valori per As (C+1) e Bs (C+3) sono uguali, si verificherà un errore e il flag di errore verrà attivato.

Il flag di uguaglianza viene attivato quando il contenuto del canale D del risultato è 0000.

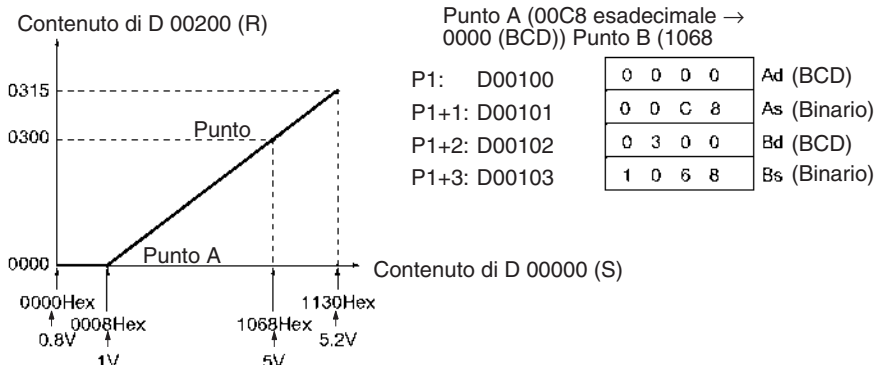
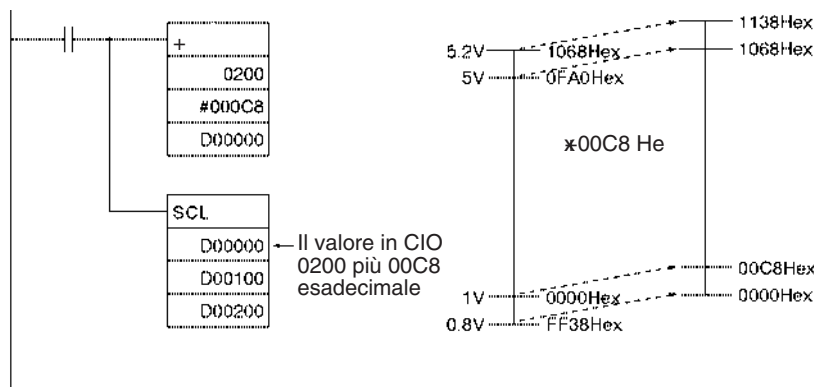
Esempi

Nell'esempio che segue, si suppone che il segnale analogico da 1 a 5 V venga convertito e immesso in D00000 come valore esadecimale compreso tra 0000 e 0FA0. Utilizzare SCL(194) per convertire/scalare il valore in CIO 0200 a un valore in formato BCD compreso tra 0000 e 0300.

Quando CIO 000000 è ON, il contenuto di D00000 viene scalato utilizzando la funzione lineare definita dal punto A (0000, 0000) e dal punto B (0FA0, 0300). Le coordinate di questi punti si trovano tra D00100 e D00103 e il risultato viene inviato in D00200.

**Valori negativi**

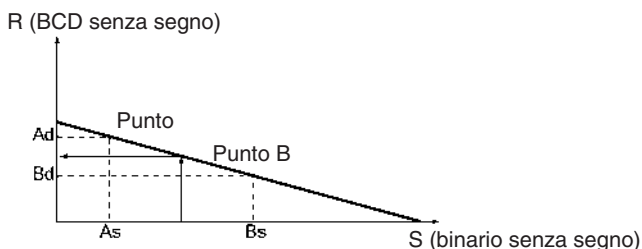
In effetti, un modulo di ingresso analogico immette valori da FF38 a 1068 esadecimale per valori compresi tra 0,8 e 5,2 V. Con SCL(194), tuttavia, è possibile gestire solo valori binari senza segno da 0000 a FFFF esadecimale e non valori binari con segno inferiori a 1 V (0000 esadecimale), come quelli compresi tra FF38 e FFFF esadecimale. In un'applicazione effettiva è quindi necessario aggiungere 00C8 esadecimale a tutti i valori in modo che FF38 esadecimale venga rappresentato come 0000 esadecimale prima dell'esecuzione di SCL(194). Osservare l'esempio seguente.



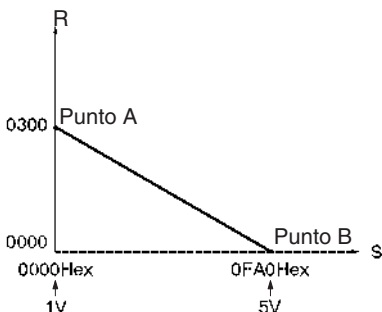
Nell'esempio, i valori compresi tra 0000 e 00C8 esadecimale vengono convertiti in valori negativi. SCL(194) può tuttavia inviare soltanto valori BCD senza segno da 0000 a 9999; pertanto quando il contenuto di D00000 è compreso tra 0000 e 00C8 esadecimale, il risultato inviato sarà 0000 BCD.

Scala invertita

È possibile utilizzare la scala invertita impostando $As < Bs$ e $Ar > Br$. Come risultato si ottiene la relazione seguente.



È possibile utilizzare la scala invertita, ad esempio, per convertire (scalare all'indietro) valori da 1 a 5 V (da 0000 a 0FA0 esadecimale) in valori compresi rispettivamente tra 0300 e 0000, come illustrato nel diagramma seguente.

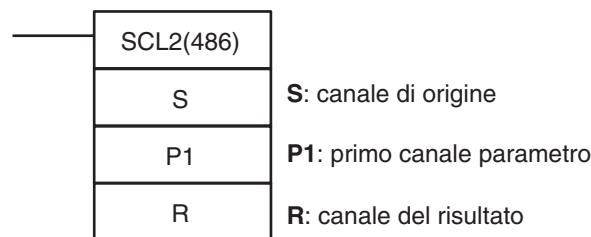


3-18-8 SCALING 2: SCL2(486)

Scopo

Converte dati binari con segno in dati in formato BCD con segno in base alla funzione lineare specificata. È possibile specificare un offset nella definizione della funzione lineare.

Simbolo programmazione ladder



Variazioni

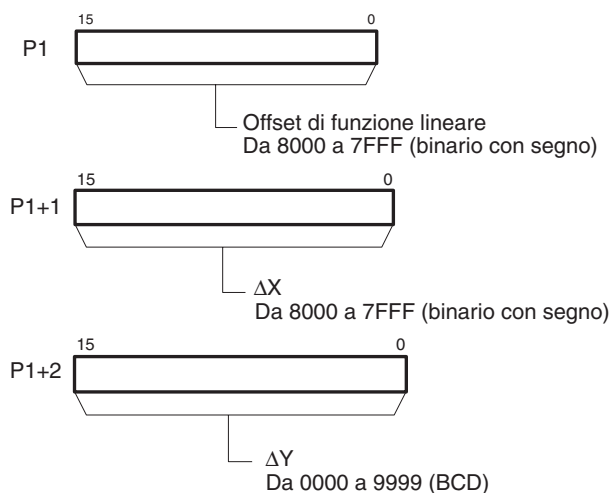
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SCL2(486) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SCL2(486) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Nel seguente diagramma è indicato il contenuto dei tre canali a partire dal primo canale parametro (P1).



Nota I canali da P1 a P1+2 devono essere nella stessa area dati.

Caratteristiche operando

| Area | S | P1 | R |
|-------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6141 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W509 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H509 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A957 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4093 | Da T0000 a T4095 |

| Area | S | P1 | R |
|--|--|---|--|
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4093 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32765 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32765 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | En_00000 ... En_32765 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SCL2(486) viene utilizzata per convertire i dati binari con segno contenuti nel canale di origine S in dati BCD con segno (i dati BCD hanno il valore assoluto e il flag di riporto indica il segno) e inviare il risultato nel canale del risultato R, in base alla funzione lineare definita dalla pendenza (ΔX , ΔY) e dall'offset. L'indirizzo del primo canale contenente ΔX , ΔY e l'offset viene specificato per il primo canale parametro P1. Il segno del risultato viene indicato dallo stato del flag di riporto (ON: negativo; OFF: positivo).

Per la conversione si utilizzano le seguenti equazioni:

$$R = \frac{\Delta Y}{\text{Conversione in BCD di } \Delta X} \times ((\text{Conversione di S in BCD}) - (\text{Conversione offset in BCD}))$$

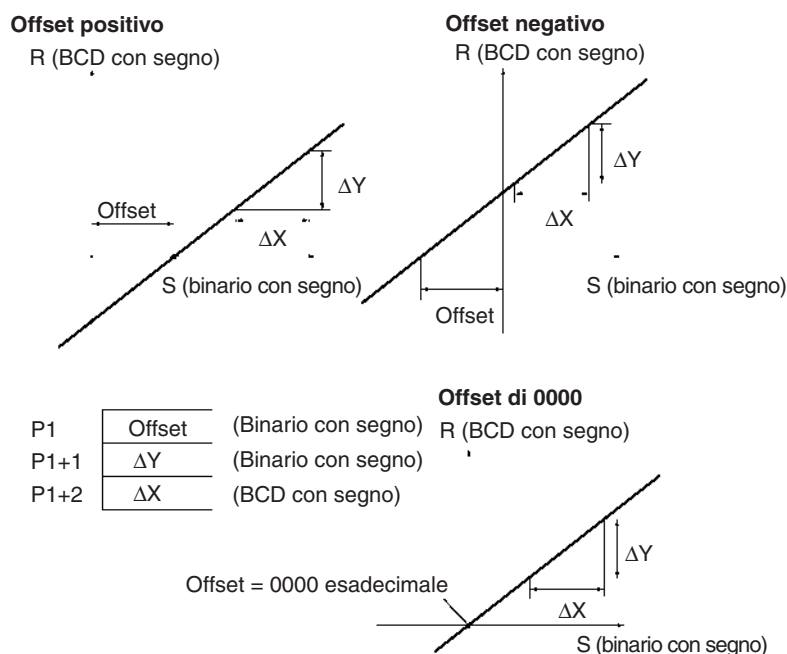
La pendenza della linea è $\Delta Y/\Delta X$.

L'offset e la pendenza possono essere valori positivi, 0 o valori negativi. L'utilizzo di una pendenza negativa consente la scala invertita.

Il risultato viene arrotondato al numero intero più vicino.

Il risultato in R sarà il valore assoluto di conversione in formato BCD e il segno verrà indicato dal flag di riporto. Pertanto il risultato può essere un valore compreso tra -9999 e 9999.

Se il risultato è minore di -9999, il risultato inviato sarà -9999. Se il risultato è maggiore di 9999, verrà inviato 9999.



È possibile utilizzare SCL2(486) per scalare i risultati dei valori di conversione del segnale analogico da moduli di ingresso analogico in base a parametri di scala definiti dall'utente. Ad esempio, se un ingresso compreso tra 1 e 5 V in un modulo di ingresso analogico viene immesso in memoria come un valore compreso tra 0000 e 0FA0 esadecimale, utilizzando SCL2(486) è possibile scalare il valore in memoria da -100 a 200°C .

SCL2(486) converte i dati binari con segno in formato BCD. I valori negativi possono quindi essere gestiti direttamente per S. È possibile utilizzare anche il risultato della funzione di scala in R e il flag di riporto per inviare valori negativi per il risultato di scala.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di C+1 (ΔX) è 0000. ON se il contenuto di C+2 (ΔY) non è BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se il risultato è negativo. OFF se il risultato è zero o positivo. |

Avvertenze

Se il valore per ΔX (C+1) è 0000 o se il valore per ΔY (C+2) non è BCD, si verificherà un errore e verrà attivato il flag di errore.

Il flag di uguaglianza viene attivato quando il contenuto del canale D del risultato è 0000.

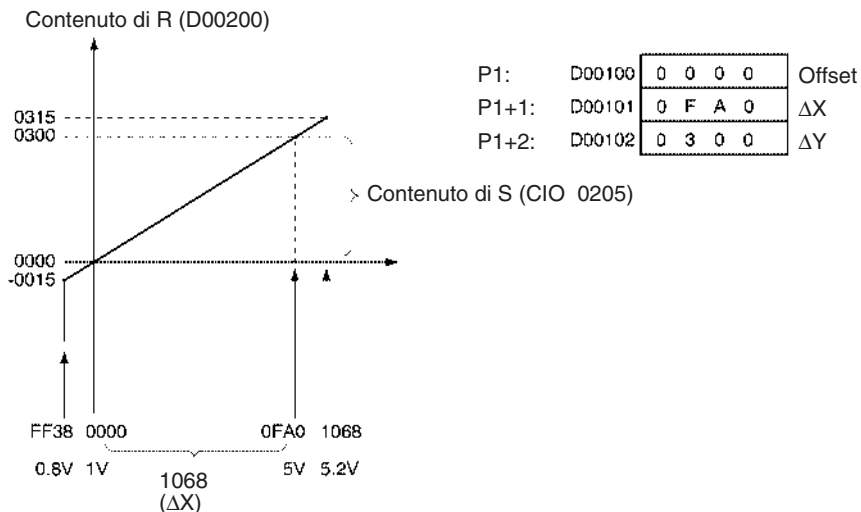
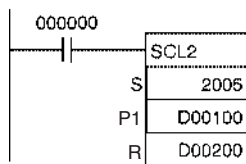
Il flag di riporto viene attivato se il valore inviato al canale del risultato è negativo.

Esempi

Scala di un ingresso analogico da 1 a 5 V a un valore compreso tra 0 e 300

Nell'esempio che segue, si suppone che il segnale analogico da 1 a 5 V venga convertito e immesso in CIO 0205 come valore esadecimale compreso tra 0000 e 0FA0. Utilizzare SCL2(486) per convertire/scalare il valore in CIO 0205 a un valore in formato BCD compreso tra 0000 e 0300.

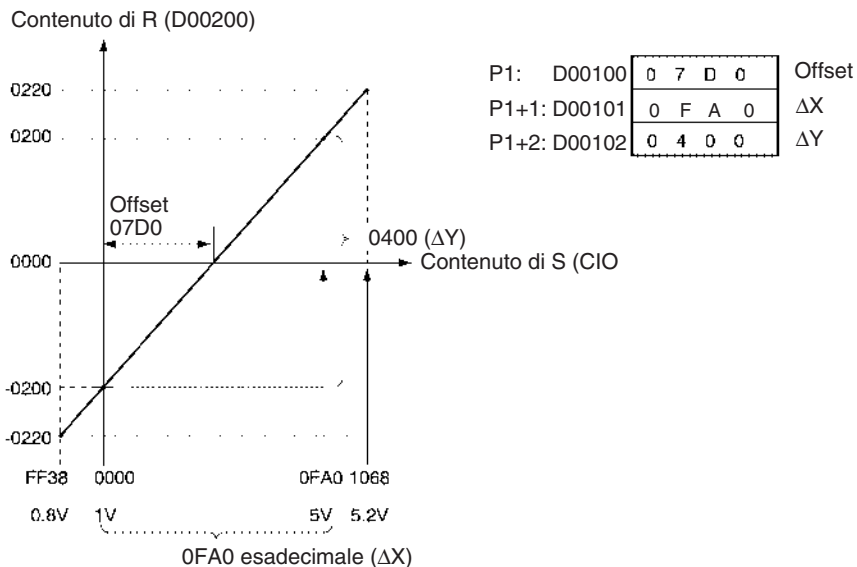
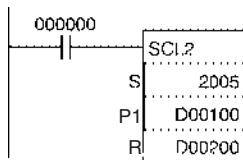
Quando CIO 000000 è ON, il contenuto di CIO 0205 viene scalato utilizzando la funzione lineare definita da ΔX (0FA0), ΔY (0300) e l'offset (0). Tali valori si trovano tra D00100 e D00102 e il risultato viene inviato in D00200.



Scala di un ingresso analogico da 1 a 5 V a un valore compreso tra -200 e 200

Nell'esempio che segue, si suppone che il segnale analogico da 1 a 5 V venga convertito e immesso in CIO 2005 come valore esadecimale compreso tra 0000 e 0FA0. Utilizzare SCL2(486) per convertire/scalare il valore in CIO 2005 a un valore in formato BCD compreso tra -0200 e 0200.

Quando CIO 000000 è ON, il contenuto di CIO 2005 viene scalato utilizzando la funzione lineare definita da ΔX (0FA0), ΔY (0400) e dall'offset (07D0). Tali valori si trovano tra D00100 e D00102 e il risultato viene inviato in D00200.

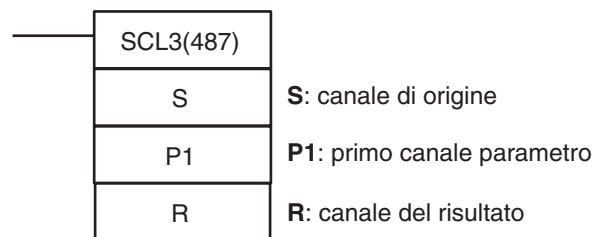


3-18-9 SCALING 3: SCL3(487)

Scopo

Converte dati in formato BCD con segno in dati binari con segno in base alla funzione lineare specificata. È possibile specificare un offset nella definizione della funzione lineare.

Simbolo programmazione ladder



Variazioni

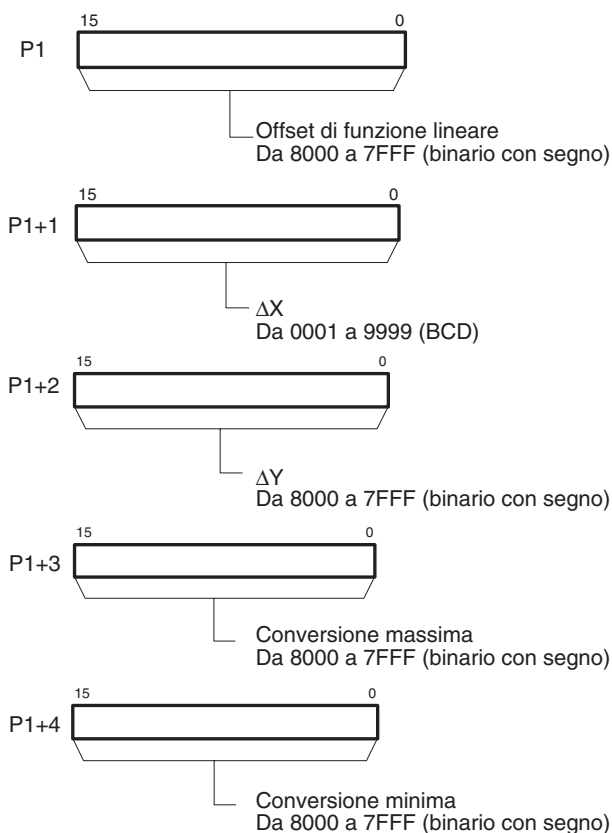
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SCL3(487) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SCL3(487) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Nel seguente diagramma è indicato il contenuto dei cinque canali a partire dal primo canale parametro (P1).



Nota I canali da P1 e P1+4 devono essere nella stessa area dati.

Caratteristiche operando

| Area | S | P1 | R |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6139 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W507 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H507 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | Da A000 ad A443 Da A448 ad A955 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4091 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4091 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32763 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32763 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | En_00000 ... En_32763 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SCL3(487) viene utilizzata per convertire i dati BCD con segno (i dati BCD hanno il valore assoluto e il flag di riporto indica il segno) contenuti nel canale di origine S in dati binari con segno e inviare il risultato nel canale del risultato R, in base alla funzione lineare definita dalla pendenza (ΔX , ΔY) e dall'offset. Vengono specificati anche i valori relativi alla conversione massima e minima. L'indirizzo del primo canale contenente ΔX , ΔY , l'offset, la conversione massima e la conversione minima viene specificato per il primo canale parametro P1.

Il segno del risultato viene indicato dallo stato del flag di riporto (ON: negativo; OFF: positivo). Per attivare e disattivare il flag di riporto, utilizzare STC(040) e CLC(041).

Per la conversione si utilizzano le seguenti equazioni:

$$R = \frac{\Delta Y}{\text{Conversione in formato binario di } \Delta X} \times ((\text{Conversione di S in formato binario}) + (\text{Offset}))$$

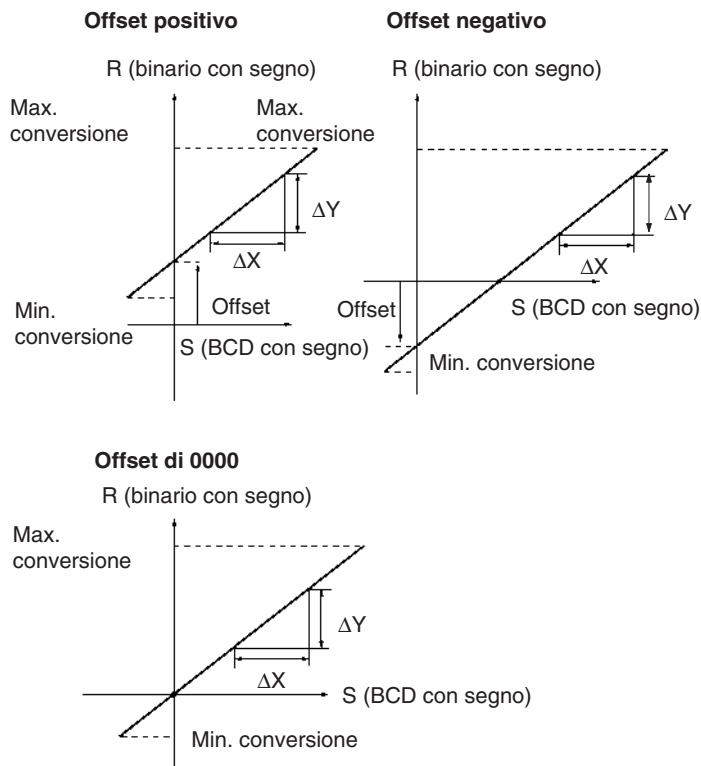
La pendenza della linea è $\Delta Y/\Delta X$.

L'offset e la pendenza possono essere valori positivi, 0 o valori negativi. L'utilizzo di una pendenza negativa consente la scala invertita.

Il risultato viene arrotondato al numero intero più vicino.

Il valore di origine in S viene considerato come un valore BCD assoluto e il segno è indicato dal flag di riporto. Di conseguenza il valore di origine può essere compreso tra -9999 e 9999.

Se il risultato è inferiore al valore minimo di conversione, quest'ultimo verrà inviato come risultato. Se il risultato è superiore al valore massimo di conversione, quest'ultimo verrà inviato come risultato.



SCL3(487) viene utilizzata per convertire i dati in formato binario con segno per moduli di uscita analogica tramite una scala definita dall'utente. Ad esempio, SCL3(487) può convertire da 0 a 200 °C in valori compresi tra 0000 e 0FA0 (esadecimale) e inviare un segnale di uscita analogica compreso tra 1 e 5 V dal modulo di uscita analogica.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se il contenuto di S non è in formato BCD. ON se il contenuto di C+1 (ΔX) non è compreso tra 0001 e 9999 BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0. OFF in tutti gli altri casi. |
| Flag negativo | N | ON quando l'MSB di R (il risultato) è 1. OFF in tutti gli altri casi. |

Avvertenze

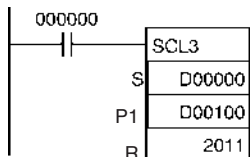
Se il contenuto di S non è BCD o se il valore per ΔX (C+1) non è compreso tra 0001 e 9999 BCD, si verificherà un errore e verrà attivato il flag di errore.

Il flag di uguaglianza viene attivato quando il contenuto del canale D del risultato è 0000.

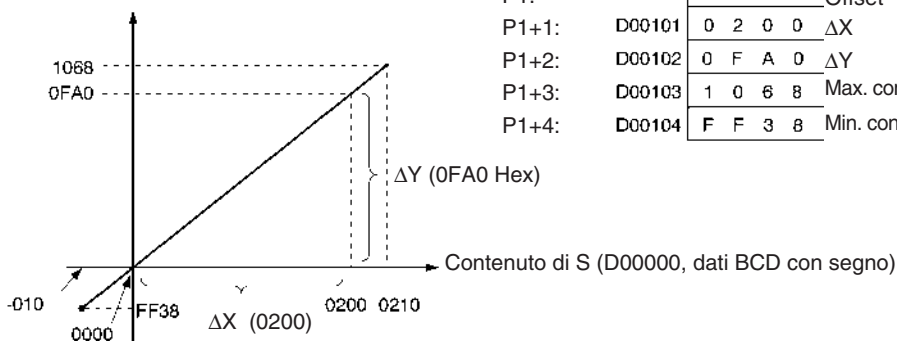
Se l'MSB del risultato in R è 1, cioè se il risultato è negativo, viene attivato il flag negativo.

Esempi

Quando un valore tra 0 e 200 viene scalato a un segnale analogico (ad esempio da 1 a 5 V), un valore BCD con segno compreso tra 0000 e 0200 viene convertito (scalato) in un valore binario con segno compreso tra 0000 e 0FA0 per un modulo di uscita analogica. Nell'esempio seguente, quando CIO 000000 passa a ON, il contenuto di D00000 viene scalato utilizzando la funzione lineare definita da ΔX (0200), ΔY (0FA0) e l'offset (0). Questi valori sono si trovano da D00100 a D00102. Il segno del valore BCD in D00000 è indicato dal flag di riporto. Il risultato viene inviato a CIO 2011.



Contenuto di R (2011, dati binari con segno)



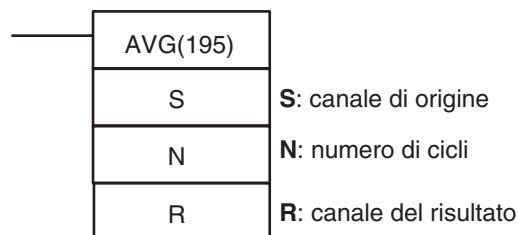
| | | | |
|-------|--------|---------|------------------|
| P1: | D00100 | 0 0 0 0 | Offset |
| P1+1: | D00101 | 0 2 0 0 | ΔX |
| P1+2: | D00102 | 0 F A 0 | ΔY |
| P1+3: | D00103 | 1 0 6 8 | Max. conversione |
| P1+4: | D00104 | F F 3 8 | Min. conversione |

3-18-10 AVERAGE: AVG(195)

Scopo

Calcola il valore medio di un canale di ingresso per il numero di cicli specificato.

Simbolo programmazione ladder



R+1: primo canale area di lavoro

Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | AVG(195) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| Non consentita | OK | OK | OK |

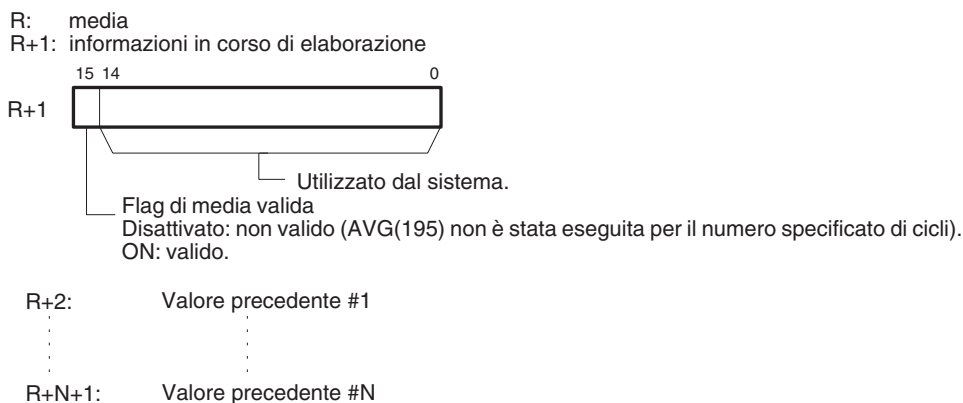
Operandi

N: numero di cicli

Il numero di cicli deve essere compreso tra 0001 e 0040 esadecimale (da 0 a 64 cicli).

R: canale del risultato ed R+1: primo canale dell'area di lavoro

R conterrà il valore medio successivo al numero specificato di cicli. R+1 fornisce informazioni sull'elaborazione della media e i canali del risultato compresi tra R+2 ed R+N+1 contengono i valori precedenti di S. Osservare il diagramma seguente.



Nota I canali da R a R+N+1 devono essere nella stessa area dati.

Caratteristiche operando

| Area | S | N | R |
|--|--|-------------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | Da #0001 a #0040 (binario) | --- |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | | |

Descrizione

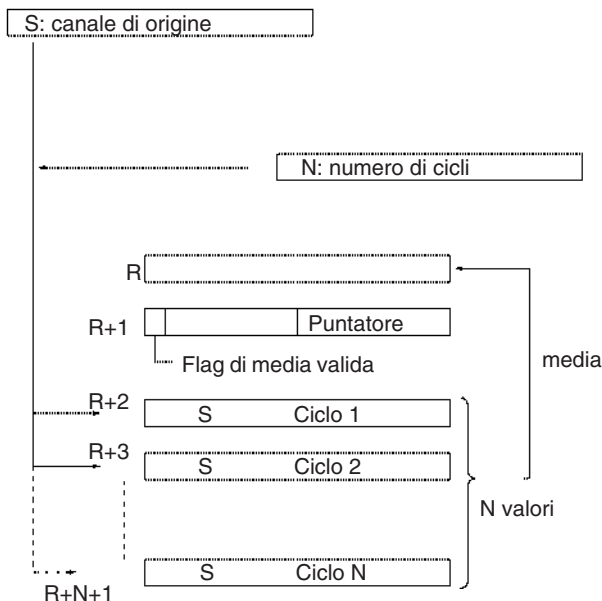
Per i primi N-1 cicli, quando la condizione di esecuzione è impostata su ON, AVG(195) scrive nei canali i valori di S in ordine, partendo da R+2. Ogni volta che un valore viene scritto, il puntatore del valore precedente (bit compresi tra 00 e 07 di R+1) viene incrementato. Una volta scritto l'ennesimo valore, il contenuto di S viene inviato in R inalterato e il flag di media valida (bit 15 di R+1) resterà disattivato.

Quando l'ennesimo valore è scritto in R+N+1, la media di tutti i valori memorizzati viene calcolata e inviata in R come valore binario senza segno e il flag di media valida (bit 15 di R+1) passerà a ON. Per tutti i cicli successivi il valore in R sarà aggiornato per gli N valori più recenti di S.

Il valore massimo di N è 64.

Una volta scritti N-1 valori, il puntatore del valore precedente sarà reimpostato su 0.

Il valore medio inviato in R viene arrotondato al numero intero più vicino.



Flag

| Nome | Eti-chetta | Funzionamento |
|----------------|------------|--|
| Flag di errore | ER | ON se il contenuto di N è 0. OFF in tutti gli altri casi. |

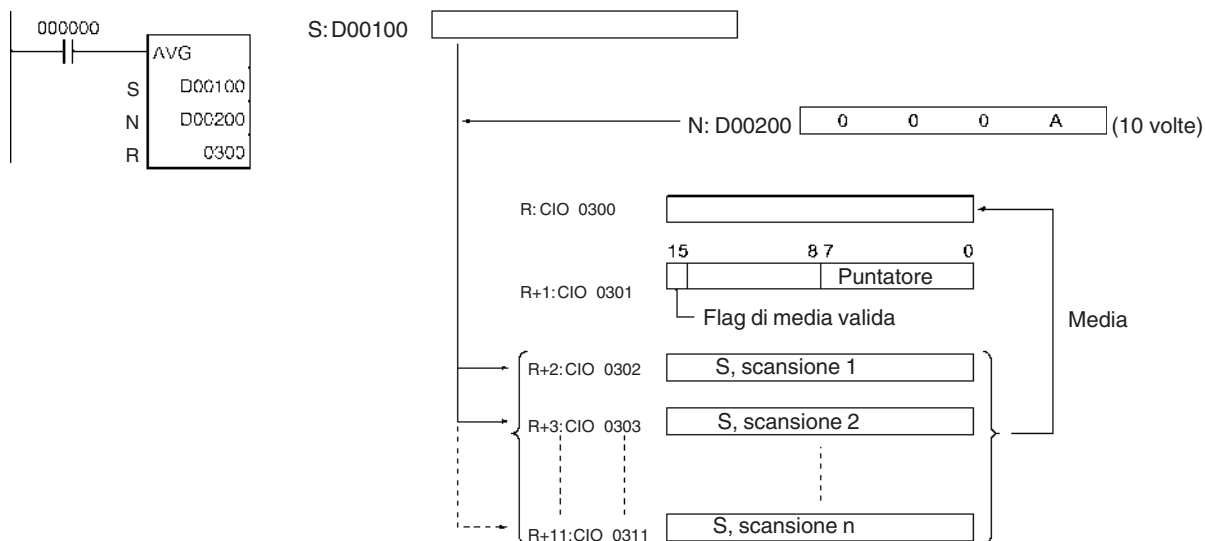
Avvertenze

Il contenuto del primo canale area di lavoro (D+1) viene azzerato (0000) ogni volta che la condizione di esecuzione passa da OFF a ON.

Il contenuto del primo canale area di lavoro (D+1) non viene azzerato in occasione della prima esecuzione del programma, all'avvio del funzionamento. Se AVG(195) deve essere eseguita nella prima scansione del programma, azzerare dal programma il primo canale dell'area di lavoro.

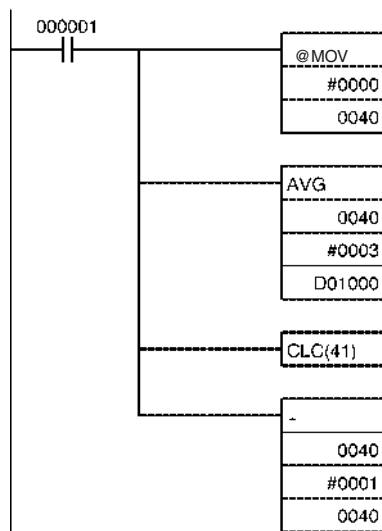
Se il contenuto di N (numero di cicli) è 0000, si verificherà un errore e verrà attivato il flag di errore.

Nell'esempio seguente, quando CIO 000000 è ON, il contenuto di D00100 verrà memorizzato una volta a ogni scansione per il numero di scansioni specificato in D00200. Il contenuto è memorizzato in sequenza nei dieci canali compresi tra CIO 0302 e CIO 0311. La media del contenuto di questi dieci canali sarà inviata in CIO 0300 e quindi verrà attivato il bit 15 di CIO 0301.



Esempi

Nell'esempio che segue, il contenuto di CIO 0040 viene impostato su #0000 e quindi incrementato di 1 a ogni ciclo. Nei primi due cicli, AVG(195) sposta il contenuto di CIO 0040 in D01002 e D01003. Modifiche intervengono altresì nel contenuto di D01001, che può essere utilizzato per confermare che i risultati di AVG(195) hanno subito modifiche. A partire dal terzo ciclo, AVG(195) calcola il valore medio del contenuto dei canali compresi tra D01002 e D01004 e scrive tale valore in D01000.



| | 1° ciclo | 2° ciclo | 3° ciclo | 4° ciclo |
|----------|----------|----------|----------|----------|
| CIO 0040 | 0000 | 0001 | 0002 | 0003 |

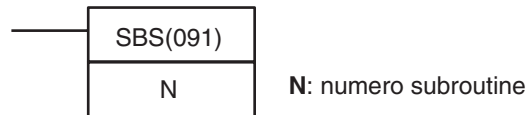
| | | | | | |
|--------|------|------|------|------|--|
| D01000 | 0000 | 0001 | 0001 | 0002 | Media Puntatore 3 precedenti valori di IR 40 |
| D01001 | 0001 | 0002 | 8000 | 8001 | |
| D01002 | 0000 | 0000 | 0000 | 0003 | |
| D01003 | --- | 0001 | 0001 | 0001 | |
| D01004 | --- | --- | 0002 | 0002 | |

3-19 Subroutine

3-19-1 SUBROUTINE CALL: SBS(091)

Scopo Richiama la subroutine con il numero di subroutine specificato ed esegue il programma.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | SBS(091) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SBS(091) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero subroutine

Specifica il numero di subroutine tra 0 e 1023 decimale.

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di subroutine deve essere compreso tra &0 e &255 decimale.

Caratteristiche operando

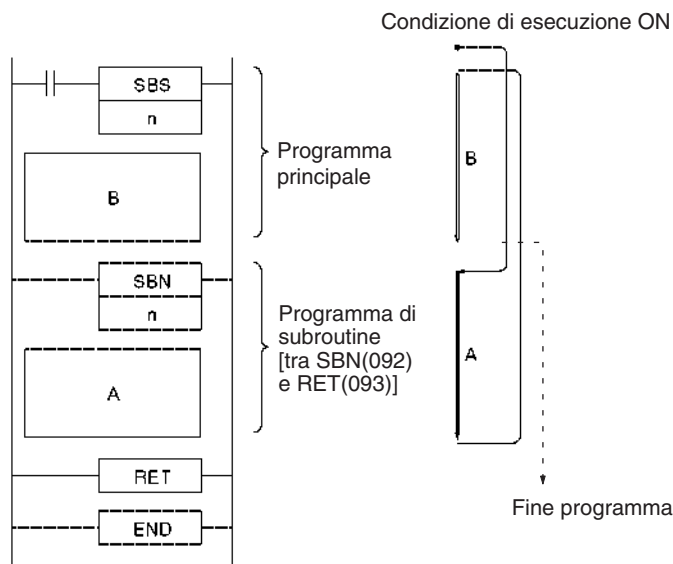
| Area | N |
|--|------------------------------------|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 0 a 1023 decimale (vedere nota) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, l'intervallo è compreso tra &0 e &255 decimale.

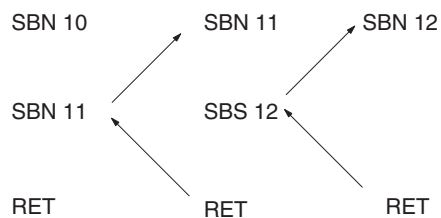
Descrizione

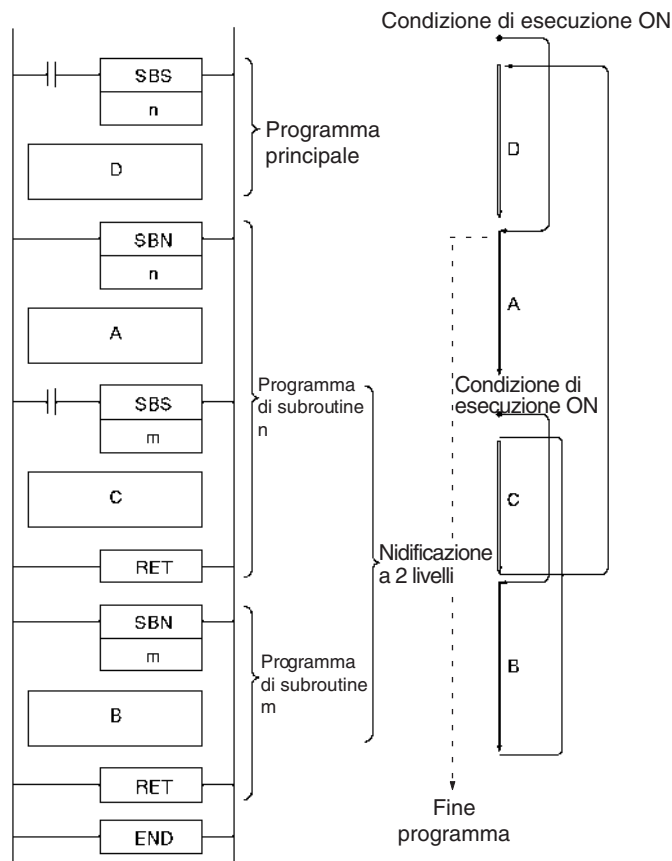
SBS(091) richiama la subroutine con il numero di subroutine specificato. La subroutine è la sezione del programma compresa tra SBN(092) e RET(093).

Quando la subroutine viene completata, l'esecuzione del programma continua con l'istruzione successiva a SBS(091).



Le subroutine possono essere nidificate fino a un massimo di 16 livelli. La nidificazione avviene quando un'altra subroutine viene richiamata dall'interno di un programma di subroutine. Osservare l'esempio seguente, nel quale è riportata una nidificazione a 3 livelli.



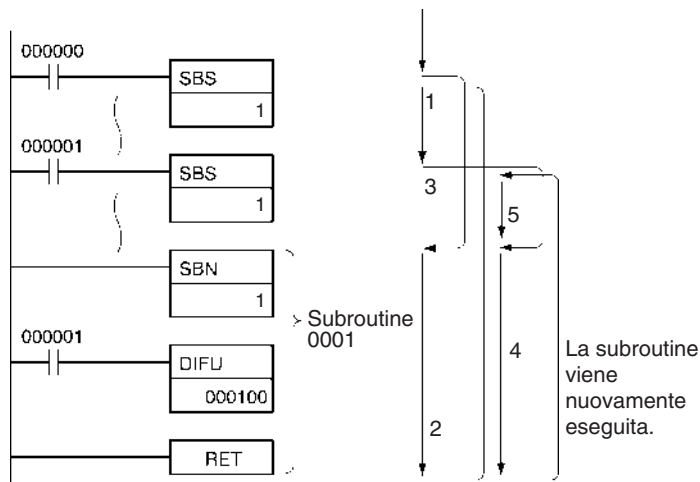


Nota In un programma, una subroutine può essere richiamata più volte.

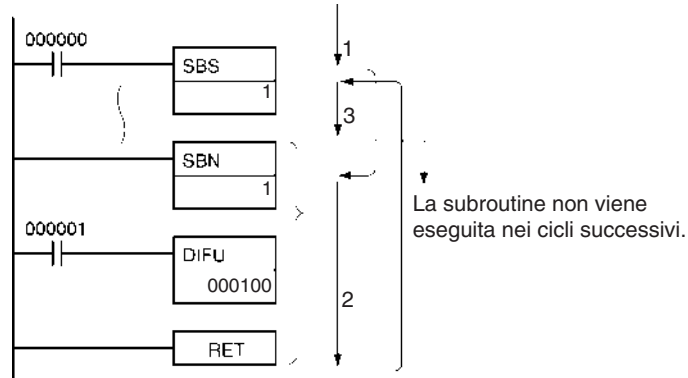
Subroutine e differenziazione

Quando si utilizzano istruzioni differenziate (DIFU(013), DIFU(014) oppure istruzioni differenziate sul fronte di salita o di discesa, è opportuno osservare le seguenti avvertenze.

Non è possibile prevedere il funzionamento di istruzioni differenziate in una subroutine se si esegue tale subroutine più volte nello stesso ciclo. Nell'esempio che segue, la subroutine 0001 viene eseguita quando CIO 000000 è ON e CIO 000100 viene impostato su ON da DIFU(013) una volta che CIO 000001 è passato da OFF a ON. Se CIO 000001 è ON nello stesso ciclo, la subroutine 0001 viene nuovamente eseguita ma questa volta DIFU(013) disattiverà CIO 000100 senza controllare lo stato di CIO 000001.



In compenso, l'uscita di un'istruzione differenziata (DIFU(013) o DIFD(014)) resterebbe impostata su ON se l'istruzione fosse eseguita e l'uscita passasse a ON, ma se la stessa subroutine non fosse richiamata una seconda volta.



Nell'esempio che segue, la subroutine 0001 viene eseguita se CIO 000000 è impostato su ON. Quando CIO 000001 è passato da OFF a ON, l'uscita CIO 000100 viene impostata su ON da DIFU(013). Se CIO 000000 è OFF nel ciclo successivo, la subroutine 0001 non sarà più eseguita e l'uscita CIO 000100 resterà impostata su ON.

Flag

| Nome | Eti-chetta | Funzionamento |
|----------------|------------|--|
| Flag di errore | ER | ON se la nidificazione supera 16 livelli. ON se il numero specificato di subroutine non esiste. ON se una subroutine richiama se stessa. ON se una subroutine in fase di esecuzione viene richiamata. ON se la subroutine specificata non è definita nel task in esecuzione. OFF in tutti gli altri casi. |

Avvertenze

SBS(091) e la corrispondente istruzione SBN(092) devono essere programmate nello stesso task. Se l'SBN(092) corrispondente non è nel task, viene generato un errore.

SBS(091) viene considerata come NOP(000) quando si trova in una sezione del programma interbloccata da IL(002) e ILC(003).

Se SBS(091) viene eseguita in uno dei casi indicati qui di seguito, la subroutine non sarà richiamata e verrà attivato il flag di errore.

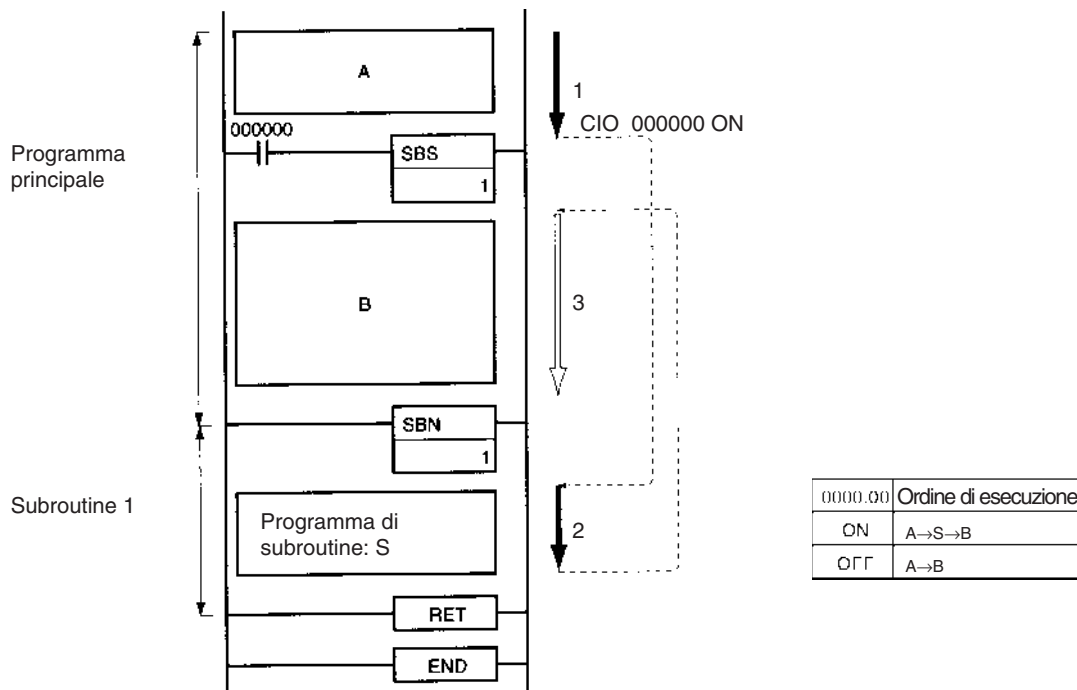
1,2,3...

1. La subroutine specificata non è definita nel task in esecuzione.
2. La subroutine sta richiamando se stessa.
3. La nidificazione della subroutine supera 16 livelli.
4. La subroutine specificata è in fase di esecuzione.

Esempi

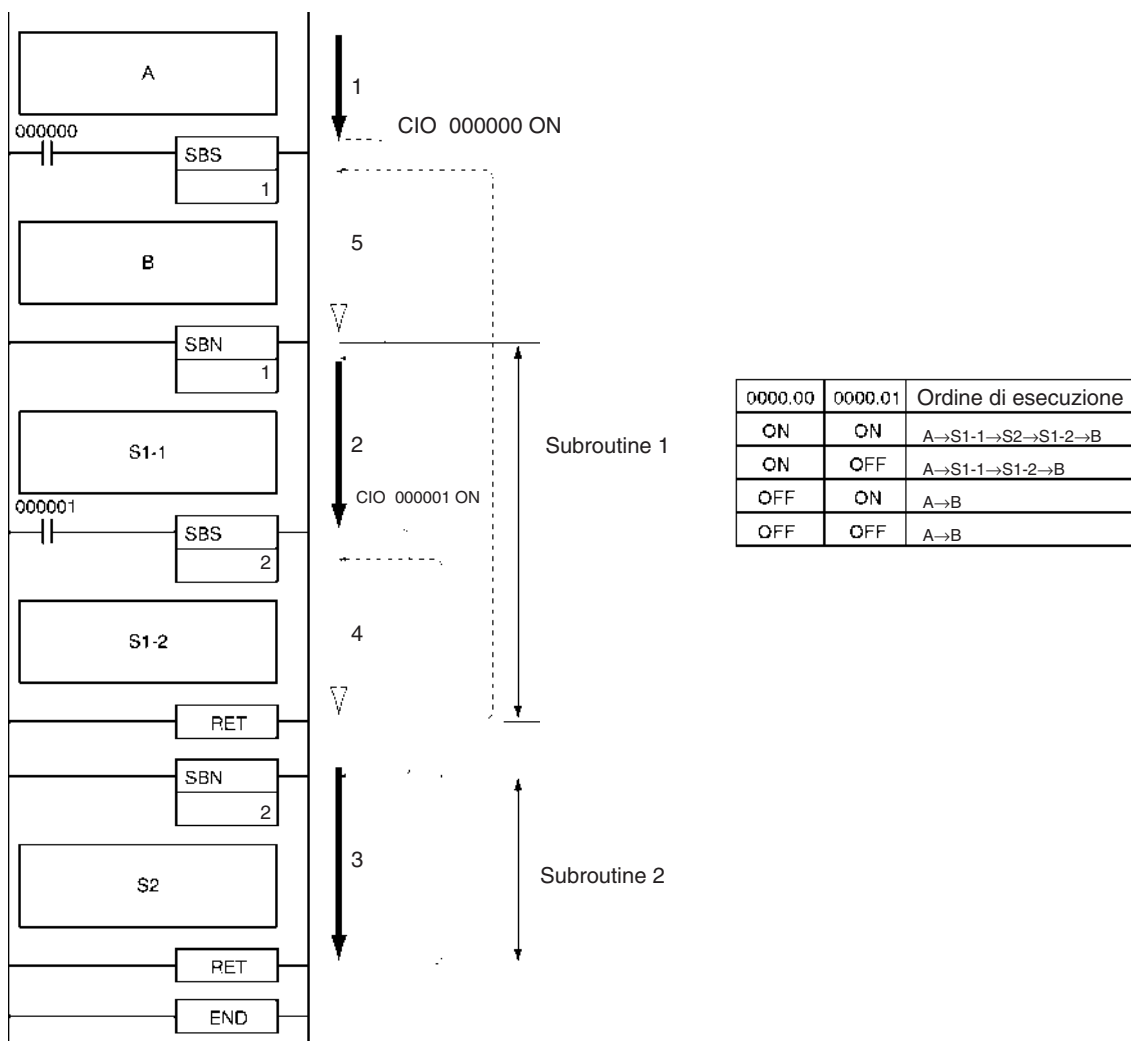
Esempio 1: subroutine sequenziali (non nidificate)

Nell'esempio seguente, quando CIO 000000 è ON, la subroutine 1 viene eseguita e l'esecuzione del programma torna all'istruzione successiva alla SBS(091). La parte restante del programma principale viene quindi eseguita fino all'istruzione che precede immediatamente SBN(092) 1.



Esempio 2: subroutine sequenziali (non nidificate)

Nell'esempio che segue, quando CIO 000000 è ON, la subroutine 1 viene eseguita e l'esecuzione del programma torna all'istruzione successiva a SBS(091) 1. Quando CIO 000001 è ON, la subroutine 2 viene eseguita e l'esecuzione del programma torna all'istruzione successiva a SBS(091) 2.

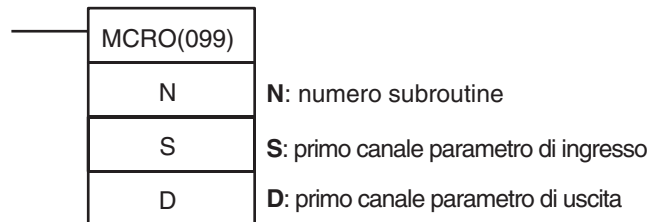


3-19-2 MACRO: MCRO(099)

Scopo

Chiama la subroutine con il numero specificato ed esegue il programma utilizzando i parametri di ingresso specificati da S a S+3 e i parametri di uscita specificati da D a D+3.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | MCRO(099) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MCRO(099) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero subroutine

Specifica il numero di subroutine tra 0 e 1023 decimale.

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di subroutine deve essere compreso tra 0 e 255 decimale.

Caratteristiche operando

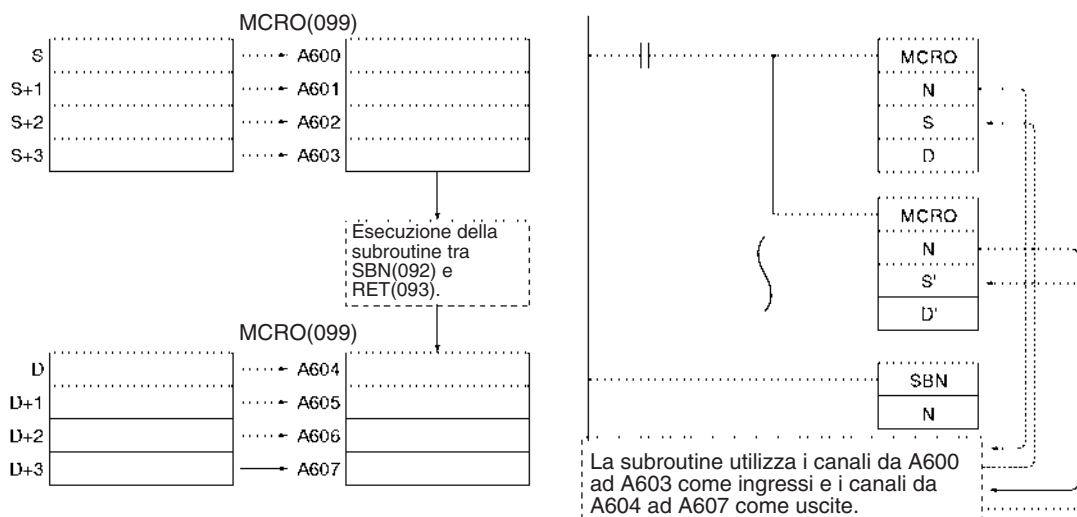
| Area | N | S | D |
|--|------------------------------------|--|-----------------|
| Area CIO | --- | Da CIO 0000 a CIO 6140 | |
| Area di lavoro | --- | Da W000 a W508 | |
| Area bit di ritentività | --- | Da H000 a H508 | |
| Area bit ausiliaria | --- | Da A000 ad A444 Da A448 ad A956 | Da A448 ad A956 |
| Area del temporizzatore | --- | Da T0000 a T4092 | |
| Area del contatore | --- | Da C0000 a C4092 | |
| Area DM | --- | Da D00000 a D32764 | |
| Area EM senza banco | --- | Da E00000 a E32764 | |
| Area EM con banco | --- | Da En_00000 a En_32764 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da 0 a 1023 decimale (vedere nota) | --- | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15, da IR0+(++) a IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, l'intervallo è compreso tra 0 e 255 decimale.

Descrizione

MCRO(099) richiama la subroutine con il numero di subroutine specificato, analogamente a SBS(091). A differenza di SBS(091), gli operandi S e D di MCRO(099) possono essere utilizzati per modificare l'indirizzo di canale e di bit nella subroutine, sebbene la struttura della subroutine sia costante.

All'esecuzione di MCRO(099), il contenuto da S a S+3 viene copiato nell'intervallo compreso tra A600 e A603 (ingressi dell'area delle macro) e la subroutine specificata viene eseguita. Quando la subroutine è completata, il contenuto da A604 a A607 (uscite dell'area delle macro) viene copiato nell'intervallo compreso tra D e D+3 e l'esecuzione del programma prosegue con l'istruzione successiva a MCRO(099).



MCRO(099) può essere utilizzata per consolidare due o più subroutine con la stessa struttura ma differenti indirizzi di ingresso e di uscita in un unico programma di subroutine. All'esecuzione di MCRO(099), i dati di ingresso e uscita specificati vengono trasferiti alla subroutine indicata.

Flag

| Nome | Eti-chetta | Funzionamento |
|----------------|------------|--|
| Flag di errore | ER | ON se la nidificazione supera 16 livelli. ON se il numero specificato di subroutine non esiste. ON se una subroutine richiama se stessa. ON se una subroutine in fase di esecuzione viene richiamata. ON se la subroutine specificata non è definita nel task in esecuzione. OFF in tutti gli altri casi. |

Nella tabella seguente sono riportati i canali relativi nell'area ausiliaria.

| Nome | Indirizzo | Funzionamento |
|--|-----------------|--|
| Canali di ingresso dell'area delle macro | Da A600 ad A603 | All'esecuzione di MCRO(099) i quattro canali da S a S+3 vengono copiati nell'intervallo compreso tra A600 e A603. I canali di ingresso vengono trasferiti alla subroutine. |
| Canali di ingresso dell'area delle macro | Da A604 ad A607 | Al termine dell'esecuzione della subroutine specificata in MCRO(099), i dati di uscita di questi canali di uscita vengono copiati nell'intervallo compreso tra D e D+3. |

Avvertenze

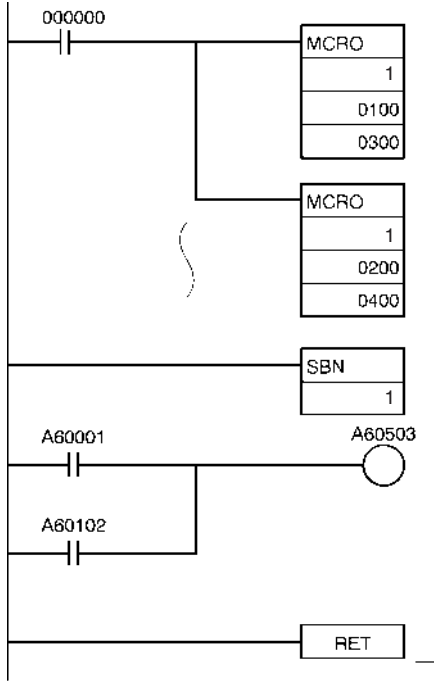
I quattro canali dei dati di ingresso (canali o bit) da A600 ad A603 and e i quattro canali dei dati di uscita (canali o bit) da A604 ad A607 devono essere utilizzati nella subroutine richiamata da MCRO(099). Non è possibile trasferire più di quattro canali di dati.

È possibile nidificare le istruzioni MCRO(099), ma i dati nei canali di ingresso e di uscita dell'area della macro (da A600 ad A607) devono essere salvati prima che venga richiamata un'altra subroutine: tutte le istruzioni MCRO(099), infatti, utilizzano gli stessi 8 canali.

Esempio

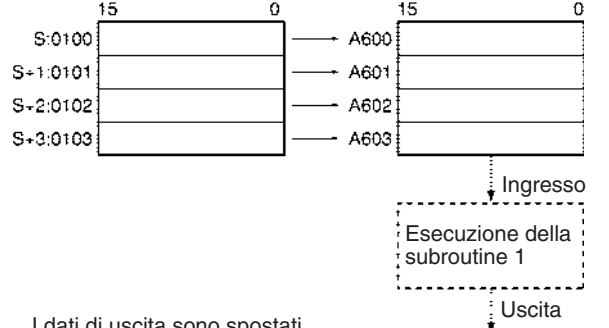
Nell'esempio che segue, quando CIO 000000 è ON, due istruzioni MCRO(099) trasferiscono alla subroutine 1 differenti dati di ingresso e di uscita.

- 1,2,3...**
1. La prima istruzione MCRO(099) passa i dati di ingresso da CIO 0100 a CIO 0103 ed esegue la subroutine. Quando la subroutine è completata, i dati di uscita vengono memorizzati da CIO 0300 a CIO 0303.
 2. La seconda istruzione MCRO(099) sposta i dati di ingresso da CIO 0200 a CIO 0203 ed esegue la subroutine. Quando la subroutine è completata, i dati di uscita vengono memorizzati da CIO 0400 a CIO 0403.

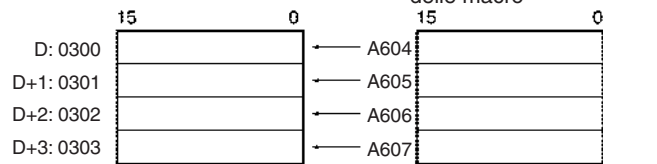


Subroutine 1

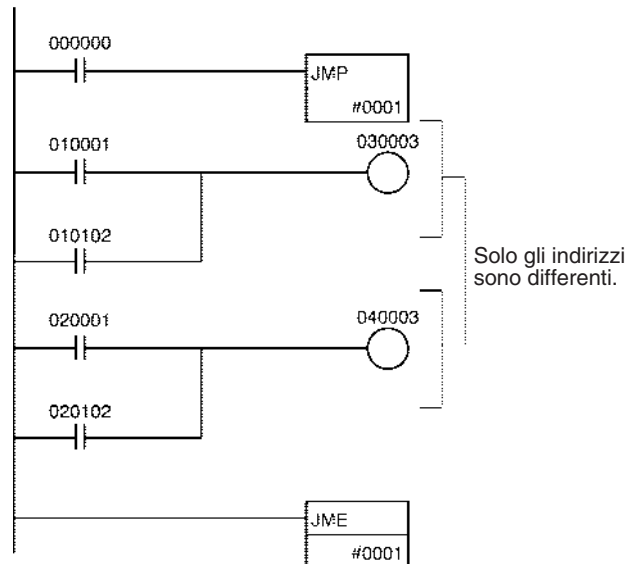
I dati di ingresso sono spostati quando la subroutine viene richiamata.



I dati di uscita sono spostati quando vengono restituiti dalla subroutine.



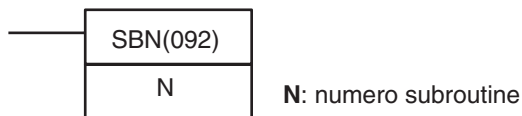
La seconda istruzione MCRO(099) funziona allo stesso modo, ma i dati di ingresso da CIO 0200 a CIO 0203 vengono spostati nell'intervallo compreso tra A600 e A603 e i dati di uscita da A604 ad A607 vengono spostati nell'intervallo compreso tra CIO 0400 e CIO 0403.



3-19-3 SUBROUTINE ENTRY: SBN(092)

Scopo Indica l'inizio del programma di subroutine con il numero di subroutine specificato. Viene utilizzata insieme all'istruzione RET(093) per definire un'area subroutine.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SBN(092) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Non consentita | Non consentita | OK | OK |

Operandi

N: numero subroutine

Specifica il numero di subroutine tra 0 e 1023 decimale.

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di subroutine deve essere compreso tra 0 e 255 decimale.

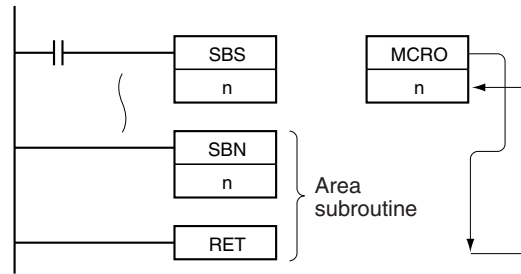
Caratteristiche operando

| Area | N |
|--|------------------------|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 0 a 1023 (decimale) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Descrizione

SBN(092) indica l'inizio della subroutine con il numero di subroutine specificato. La fine della subroutine è indicata da RET(093).

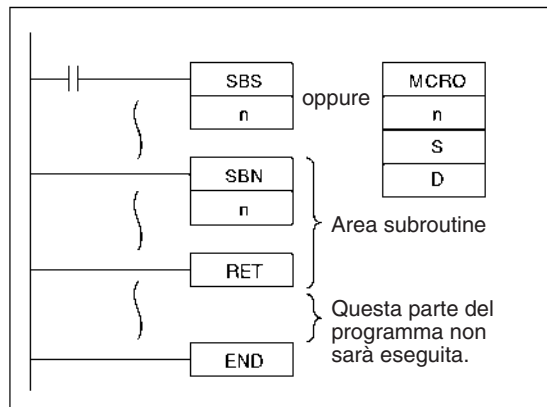
L'area del programma che inizia alla prima istruzione SBN(092) è l'area subroutine. Una subroutine viene eseguita solo quando è stata richiamata da SBS(091) o MCRO(099).



Avvertenze

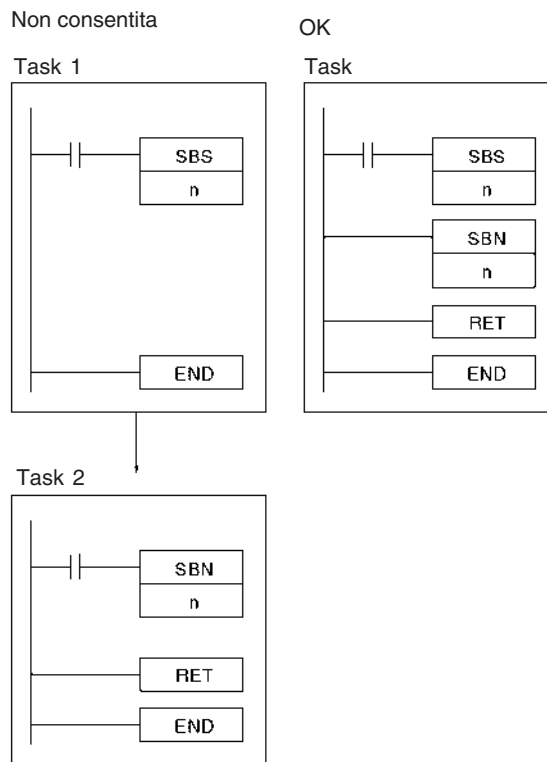
Quando la subroutine non è in corso di esecuzione, le istruzioni vengono considerate come NOP(000).

Inserire le subroutine dopo il programma principale e appena prima dell'istruzione END(001) nel programma di ogni task. Se parte del programma principale è collocata dopo l'area subroutine, tale sezione del programma sarà ignorata.

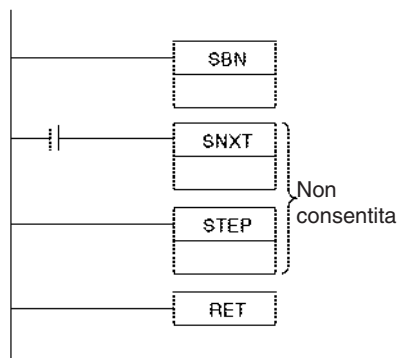


Nota Il metodo di immissione del numero subroutine, N, è differente per CX-Programmer e una console di programmazione. Indicare da #0 a #1023 per CX-Programmer e da 0000 a 1023 su una console di programmazione.

Assicurarsi di specificare ogni subroutine nello stesso programma (task) dell'istruzione SBS(091) o MCRO(099) corrispondente. Non è possibile richiamare una subroutine di un task da un altro task. È possibile programmare una subroutine all'interno di un task ad interrupt.

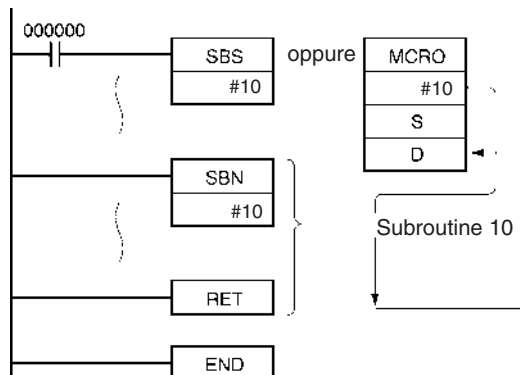


Nelle subroutine non è possibile utilizzare le istruzioni di step, STEP(008) e SNXT(009).



Esempio

Nell'esempio seguente, quando CIO 000000 è ON, la subroutine 10 viene eseguita e l'esecuzione del programma torna all'istruzione successiva all'istruzione SBS(091) o MCRO(099) che ha richiamato la subroutine.



3-19-4 SUBROUTINE RETURN: RET(093)

Scopo Indica la fine del programma di subroutine. Viene utilizzata insieme all'istruzione SBN(092) per definire un'area subroutine.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RET(093) |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | Non consentita | OK | OK |

Descrizione

RET(093) indica la fine di una subroutine mentre SBN(092) ne indica l'inizio. Per ulteriori informazioni sul funzionamento delle subroutine, consultare 3-19-3 *SUBROUTINE ENTRY: SBN(092)*.

Quando l'esecuzione del programma raggiunge RET(093), torna automaticamente all'istruzione successiva all'istruzione SBS(091) o MCRO(099) che ha richiamato la subroutine. Quando la subroutine è richiamata da MCRO(099), i dati di uscita da A604 ad A607 vengono scritti nell'intervallo compreso tra D e D+3 prima che l'esecuzione del programma riprenda.

Avvertenze

Quando la subroutine non è in corso di esecuzione, le istruzioni vengono considerate come NOP(000).

Esempio

Per ulteriori esempi sul funzionamento di RET(093), consultare 3-19-3 *SUBROUTINE ENTRY: SBN(092)*.

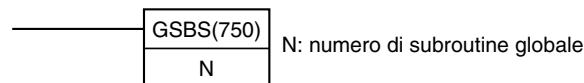
3-19-5 GLOBAL SUBROUTINE CALL: GSBS(750)

Scopo

Richiama la subroutine globale con il numero di subroutine specificato ed esegue il programma (le subroutine globali possono essere richiamate da due o più task).

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D. GSBS(750) viene utilizzata insieme alle istruzioni GSBN(751) e GRET(752) nonché GLOBAL SUBROUTINE ENTRY e GLOBAL SUBROUTINE RETURN.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | GSBS(750) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @GSBS(750) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi**N: numero di subroutine globale**

Specifica il numero di subroutine globale tra 0 e 1023 decimale.

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di subroutine deve essere compreso tra 0 e 255 decimale.

Caratteristiche operando

| Area | N |
|--|------------------------------------|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 0 a 1023 decimale (vedere nota) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, l'intervallo è compreso tra 0 e 255 decimale.

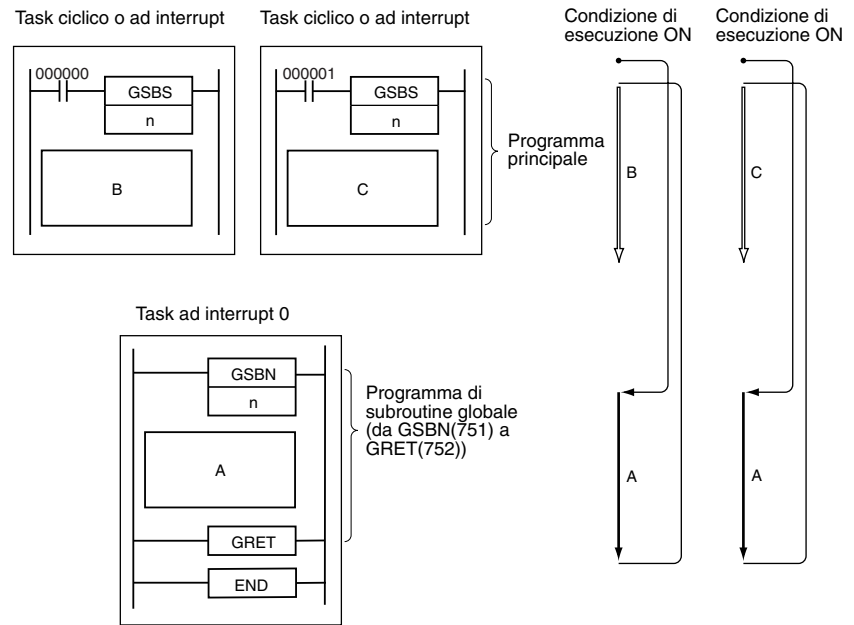
Descrizione

GSBS(750) richiama la subroutine globale con il numero di subroutine globale specificato. La subroutine globale è la sezione del programma compresa tra GSBN(751) e GRET(752). Quando la subroutine globale viene completata, l'esecuzione del programma continua con l'istruzione successiva a GSBS(750).

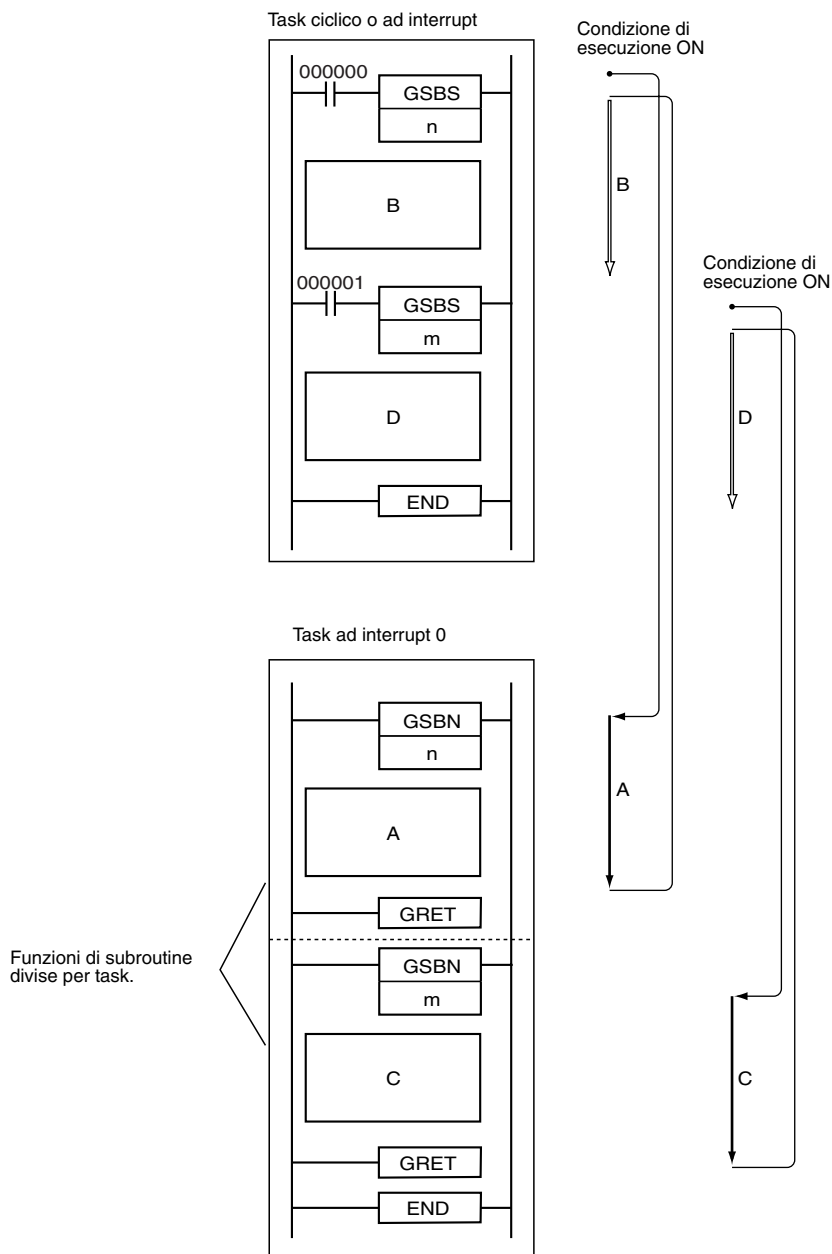
Per richiamare quel programma da task differenti, è possibile scrivere questa istruzione in più task con lo stesso numero di subroutine globale. Il programma può essere modularizzato trasformando le subroutine globali in subroutine standard, comuni a numerosi task.

L'area subroutine globale (tra GSBN(751) e GRET(752)) deve essere definita nel task ad interrupt 0. Se è definita altrove, verrà generato un errore e, all'esecuzione dell'istruzione GSBS(750), verrà attivato il flag di errore.

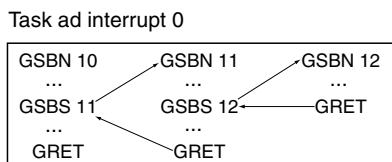
È possibile scrivere l'istruzione GSBS(750) sia nei task ciclici (compresi quelli supplementari) che nei task ad interrupt.



Nel task ad interrupt 0 è possibile definire più aree subroutine globali (da GSBN(751) a GRET(752)).



Per "nidificare" subroutine, è possibile scrivere un'istruzione SBS(091) o GSBS(750) in un'area subroutine (da SBN(092) a RET(093)) o in un'area subroutine globale (da GSBN(751) a GRET(752)). Le subroutine possono essere nidificate fino a un massimo di 16 livelli.

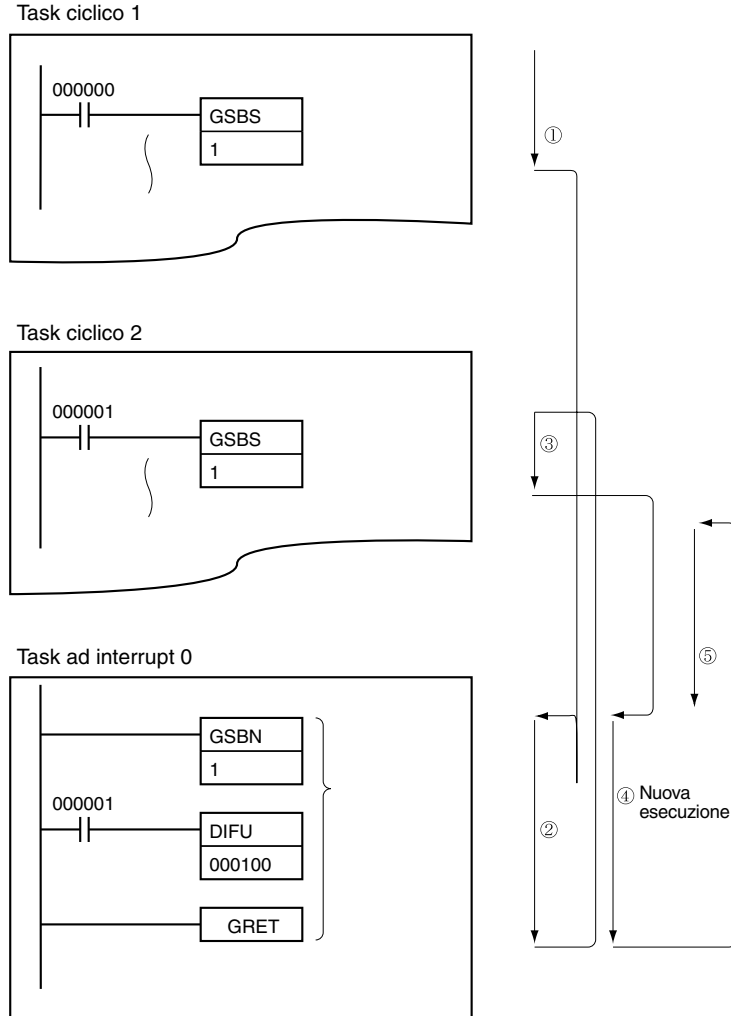


Subroutine globali e differenziazione

Quando si utilizzano istruzioni differenziate (DIFU(013), DIFU(014) oppure istruzioni differenziate sul fronte di salita o di discesa, è opportuno osservare le seguenti avvertenze.

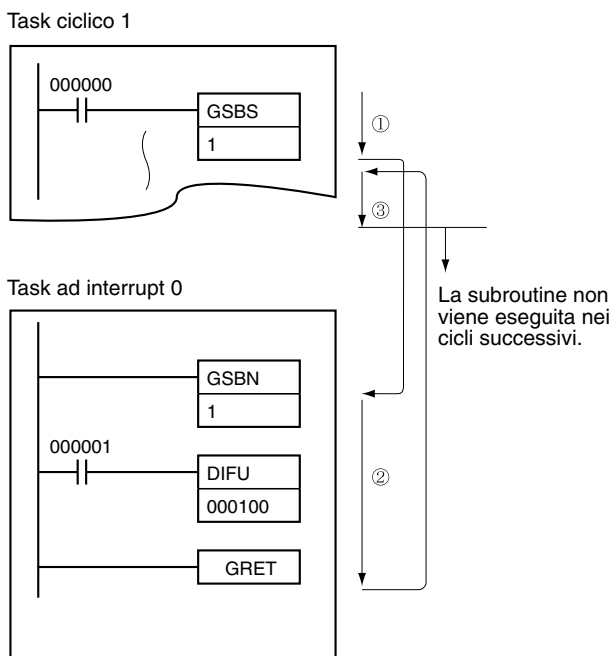
Non è possibile prevedere il funzionamento di istruzioni differenziate in una subroutine globale se si esegue tale subroutine più volte nello stesso ciclo.

Nell'esempio che segue, la subroutine globale 0001 viene eseguita quando CIO 000000 è ON e CIO 000100 viene impostato su ON da DIFU(013) una volta che CIO 000001 è passato da OFF a ON. Se CIO 000001 è ON nello stesso ciclo, la subroutine globale 0001 viene nuovamente eseguita ma questa volta DIFU(013) non rileverà il fronte di salita di CIO 000001 e CIO 000100 passerà a OFF.



In compenso, l'uscita di un'istruzione differenziata (DIFU(013) o DIFD(014)) resterebbe impostata su ON se l'istruzione fosse eseguita e l'uscita passasse a ON, ma la stessa subroutine globale non sarebbe richiamata una seconda volta.

Nell'esempio che segue, la subroutine globale 0001 viene eseguita se CIO 000000 è impostato su ON. Quando CIO 000001 è passato da OFF a ON, l'uscita CIO 000100 viene impostata su ON da DIFU(013). Se CIO 000000 è OFF nel ciclo successivo, la subroutine 0001 non sarà più eseguita e l'uscita CIO 000100 resterà impostata su ON.



Flag

| Nome | Eti-chetta | Funzionamento |
|----------------|------------|---|
| Flag di errore | ER | ON se la nidificazione supera 16 livelli (contando sia le subroutine normali che quelle globali). ON se la subroutine globale specificata non esiste. ON se una subroutine globale richiama se stessa. ON se una subroutine globale in fase di esecuzione viene richiamata. ON se la subroutine specificata non è definita nel task ad interrupt 0. OFF in tutti gli altri casi. |

Avvertenze

L'istruzione GLOBAL SUBROUTINE ENTRY, GSBN(751) e l'istruzione corrispondente GLOBAL SUBROUTINE RETURN GRET(752) devono essere programmate nel task ad interrupt 0. Se l'area subroutine globale è programmata altrove, verrà generato un errore e, all'esecuzione dell'istruzione GSBS(750), sarà attivato il flag di errore.

La normale istruzione SUBROUTINE CALL, SBS(091), non può richiamare un'area subroutine globale (da GSBN(751) a GRET(752)).

GSBS(750) non viene eseguita se programmata in una sezione del programma interbloccata da IL(002) e ILC(003), per cui gli interblocchi non sono consentiti nelle aree subroutine globali.

È possibile richiamare più di una volta la stessa area subroutine globale (da GSBN(751) a GRET(752)).

Se GSBS(750) viene eseguita in uno dei casi indicati qui di seguito, la subroutine globale non sarà richiamata e verrà attivato il flag di errore.

- 1,2,3...**
1. La subroutine globale specificata non è definita.
 2. La nidificazione di subroutine supera 16 livelli (contando sia le subroutine normali che quelle globali).
 3. La subroutine globale sta richiamando se stessa.
 4. La subroutine globale specificata è in fase di esecuzione.
 5. La subroutine globale specificata non è definita nel task ad interrupt 0.

Esempi

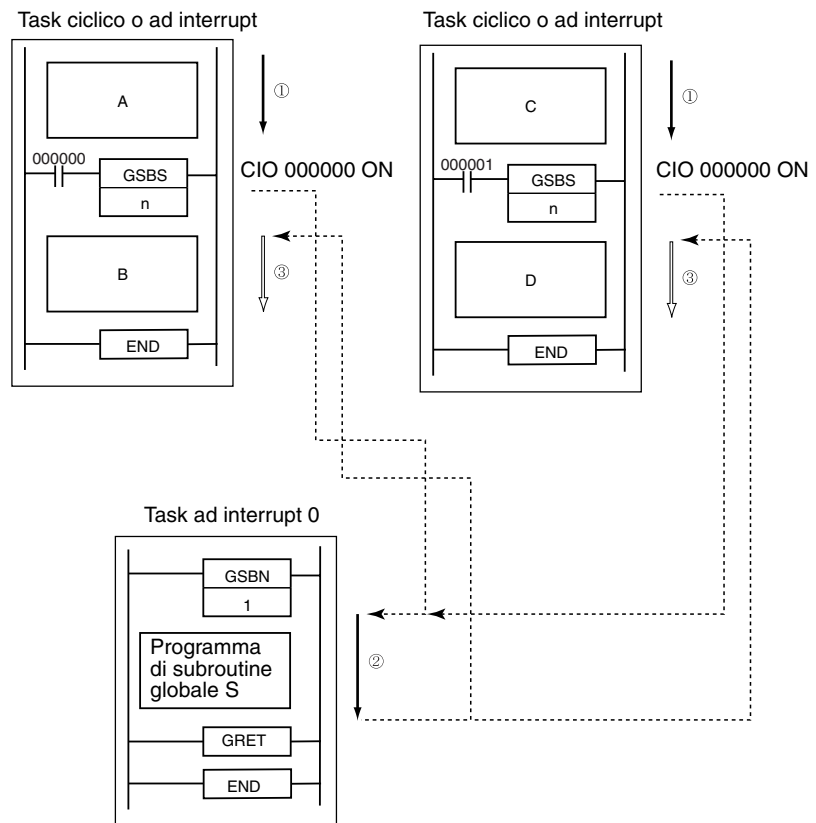
Esempio 1

Nell'esempio seguente, quando CIO 000000 è ON, la subroutine globale 1 viene eseguita e l'esecuzione del programma torna all'istruzione successiva alla GSBS(750).

| Stato di CIO 000000 | Ordine di esecuzione del programma |
|---------------------|------------------------------------|
| ON | A → S → B |
| OFF | A → S → B |

Nell'esempio seguente, quando CIO 000001 è ON, la subroutine globale 1 viene eseguita e l'esecuzione del programma torna all'istruzione successiva alla GSBS(750).

| Stato di CIO 000000 | Ordine di esecuzione del programma |
|---------------------|------------------------------------|
| ON | C → S → D |
| OFF | C → D |



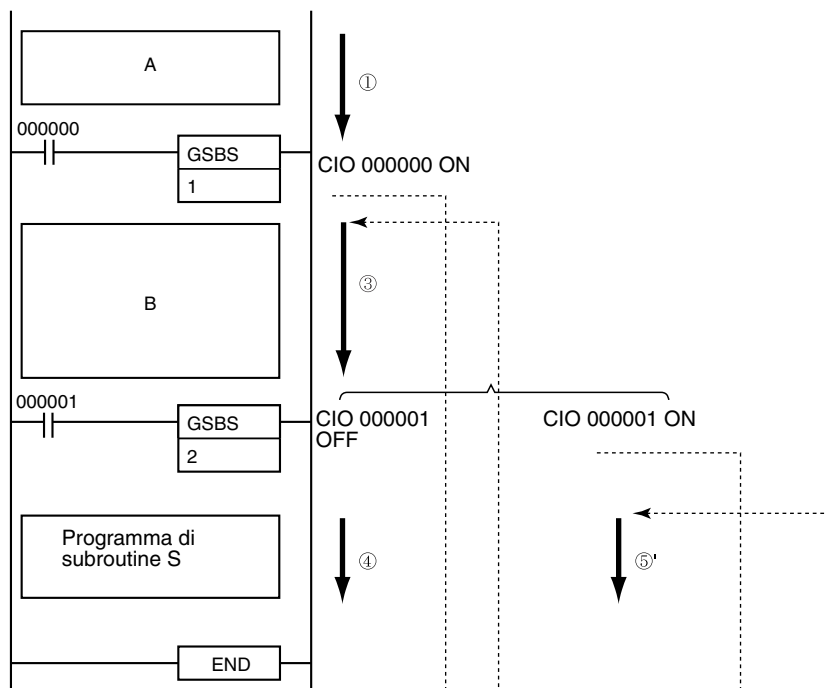
Esempio 2

É possibile programmare due o più programmi di subroutine globale nel task ad interrupt task 0, che in tal caso può essere diviso e utilizzato come task della funzione di subroutine.

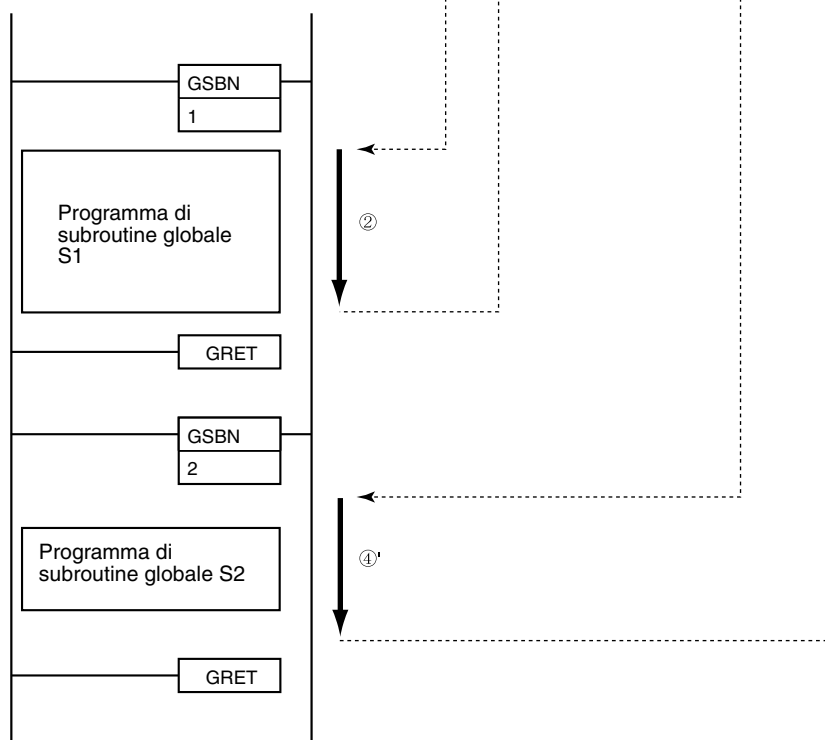
Quando CIO 000000 è ON, viene eseguito il programma di subroutine globale 1.

Quando CIO 000001 è ON, viene eseguito il programma di subroutine globale 2.

Task ciclico o ad interrupt



Task ad interrupt 0



È possibile cercare e correggere gli errori in determinati task utilizzando le subroutine normali del solo task locale o le subroutine condivise con altri task.

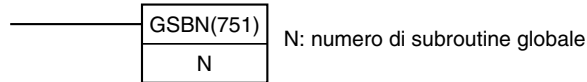
3-19-6 GLOBAL SUBROUTINE ENTRY GSBN(751)

Scopo

Indica l'inizio del programma di subroutine globale con il numero di subroutine specificato. Viene utilizzata insieme all'istruzione GRET(752) per definire un'area subroutine globale.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D. GSBN(751) viene utilizzata insieme alle istruzioni GSBS(750) e GRET(752) nonché GLOBAL SUBROUTINE CALL e GLOBAL SUBROUTINE RETURN.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | GSBN(751) |
|-------------------------|---|----------------|
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | Non consentita | --- | OK |

Operandi

N: numero di subroutine globale

Specifica il numero di subroutine globale tra 0 e 1023 decimale.

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, il numero di subroutine deve essere compreso tra 0 e 255 decimale.

Caratteristiche operando

| Area | N |
|--|------------------------------------|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 0 a 1023 decimale (vedere nota) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Nota Per CPU CJ1M-CPU11 e CJ1M-CPU21, l'intervallo è compreso tra 0 e 255 decimale.

Descrizione

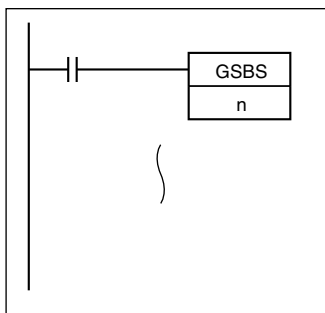
GSBN(751) indica l'inizio della subroutine globale con il numero di subroutine specificato. La conclusione della subroutine è indicata da GRET(752).

L'area del programma che inizia alla prima istruzione GSBN(751) è l'area subroutine. Una subroutine viene eseguita solo quando è stata richiamata da GSBS(750).

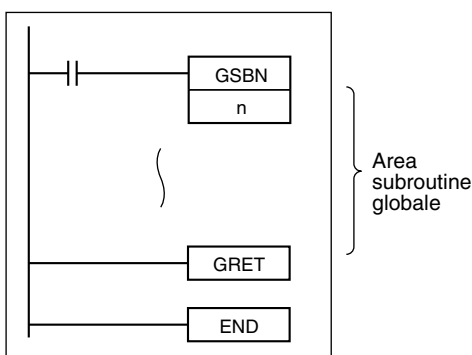
L'area subroutine globale (tra GSBN(751) e GRET(752)) deve essere definita nel task ad interrupt 0. Se è definita altrove, verrà generato un errore e, all'esecuzione dell'istruzione GSBS(750), sarà attivato il flag di errore.

È possibile scrivere l'istruzione GSBS(750) sia nei task ciclici (compresi quelli supplementari) che nei task ad interrupt.

Task ciclico o ad interrupt



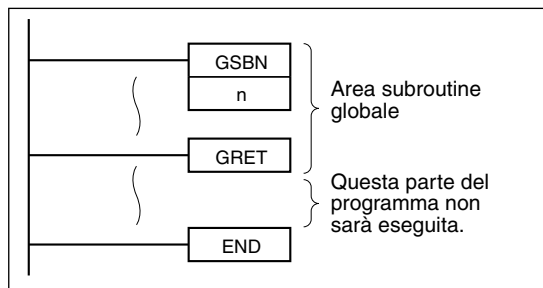
Task ad interrupt 0



Avvertenze

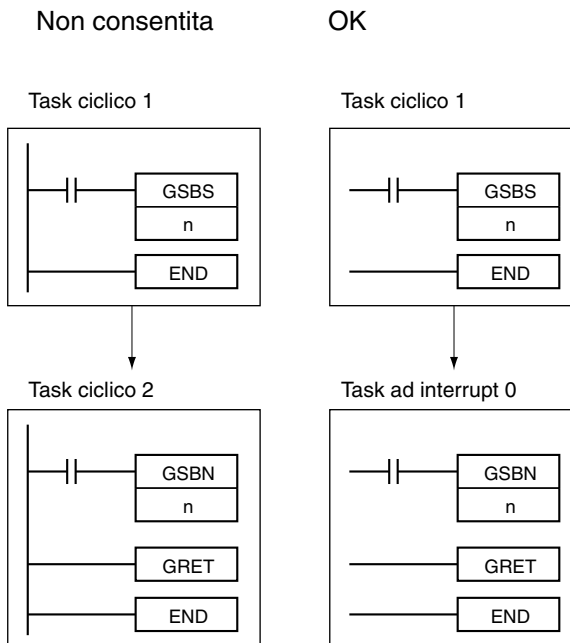
- Quando la subroutine non è in corso di esecuzione, le istruzioni vengono considerate come NOP(000).
- Posizionare l'area subroutine globale (da GSBN(751) a GRET(752)) nel task ad interrupt 0 appena prima dell'istruzione END(001). Quando si utilizzano contemporaneamente due o più subroutine globali, raggrupparle nel task ad interrupt 0 al termine del programma principale. Se parte del programma principale è collocata dopo l'area subroutine globale, tale sezione del programma sarà ignorata.

Task ad interrupt 1

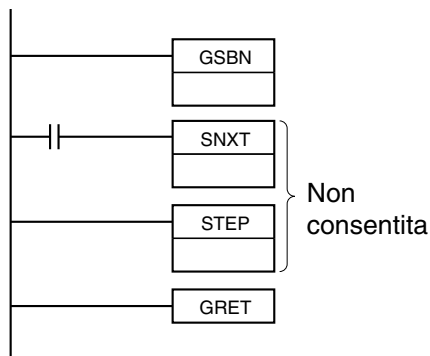


- Il metodo di immissione del numero subroutine globale, N, è differente per CX-Programmer e una console di programmazione. Indicare da #0 a #1023 per CX-Programmer e da 0000 a 1023 su una console di programmazione.

- Posizionare sempre le subroutine globali nel task ad interrupt 0. Se una subroutine globale viene richiamata e non si trova in tale task, verrà generato un errore.

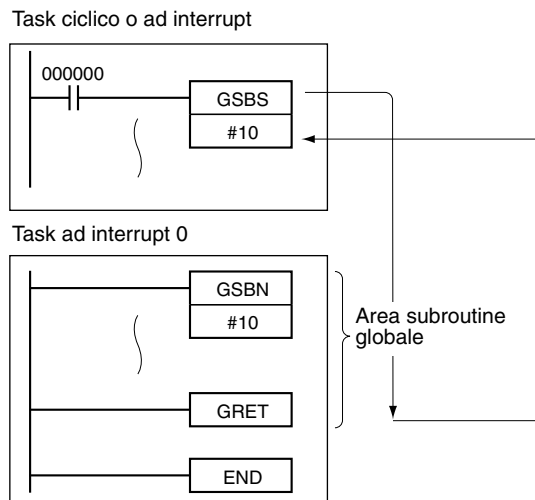


- Nelle subroutine globali non è possibile utilizzare le istruzioni di step STEP(008) e SNXT(009).



Esempio

Nell'esempio seguente, quando CIO 000000 è ON, la subroutine globale 10 viene eseguita e l'esecuzione del programma torna all'istruzione successiva all'istruzione GSBS(750) che ha richiamato la subroutine.



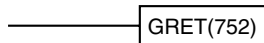
3-19-7 GLOBAL SUBROUTINE RETURN: GRET(752)

Scopo

Indica la conclusione del programma di subroutine. Viene utilizzata insieme all'istruzione GSBN(751) per definire un'area subroutine.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D. GRET(752) viene utilizzata insieme alle istruzioni GSBS(750) e GRET(751) nonché GLOBAL SUBROUTINE CALL e GLOBAL SUBROUTINE ENTRY.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | GRET(752) |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Non consentita | Non consentita | Non consentita | OK |

Descrizione

GRET(752) indica la conclusione di una subroutine globale e GSBN(751) ne indica l'inizio. Per ulteriori informazioni sul funzionamento delle subroutine globali, consultare 3-19-6 GLOBAL SUBROUTINE ENTRY GSBN(751).

Quando l'esecuzione del programma raggiunge GRET(752), torna automaticamente all'istruzione successiva all'istruzione GSBS(750) che ha richiamato la subroutine globale.

Avvertenze

Quando la subroutine non è in corso di esecuzione, le istruzioni vengono considerate come NOP(000).

Esempio

Per ulteriori esempi sul funzionamento di GRET(752), consultare 3-19-6 GLOBAL SUBROUTINE ENTRY GSBN(751).

3-20 Istruzioni di controllo degli interrupt

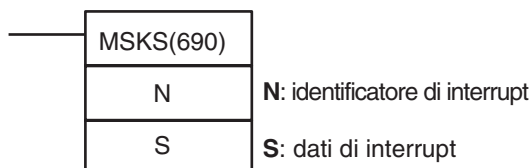
3-20-1 SET INTERRUPT MASK: MSKS(690)

Scopo

Alla prima accensione del PLC, sia i task ad interrupt di I/O che quelli programmati sono mascherati (disattivati). Per smascherare e mascherare gli interrupt di I/O e per impostare gli intervalli di tempo per gli interrupt programmati, è possibile utilizzare l'istruzione MSKS(690).

MSKS(690) non è supportata dalle CPU CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MSKS(690) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MSKS(690) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

CS1W-INT01, CJ1W-INT01

Specifica dell'elaborazione degli interrupt di I/O e della maschera

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 0: numero di modulo 0 1: numero di modulo 1 |
| S | Maschera di interrupt. Impostare un valore compreso tra 0000 ed FFFF esadecimale (16 bit per modulo). I bit singoli indicano quanto segue: 0: abilita interrupt 1: maschera interrupt |

- Nota**
- Non è possibile utilizzare CS1W-INT01 e C200HS-INT01 nello stesso momento.
 - Tutti gli interrupt di ingresso rilevati verranno cancellati alla cancellazione della maschera di interrupt.
 - Il modulo di ingresso ad interrupt CJ1W-INT01 non può venire utilizzato con una CPU CJ1. Inoltre, non è possibile eseguire i task ad interrupt di I/O.

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 0 | Da 100 a 115 | I bit da 00 a 15 di S corrispondono ai task ad interrupt di ingresso. |
| 1 | Da 116 a 131 | |

Specifica dell'identificazione con fronte di salita o di discesa dell'elaborazione degli interrupt di I/O

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 2: numero di modulo 0 3: numero di modulo 1 |
| S | Specificare il fronte di salita o di discesa del segnale di ingresso ad interrupt. Impostare un valore compreso tra 0000 ed FFFF esadecimale (16 bit per modulo). I bit singoli indicano quanto segue: 0: fronte di salita 1: fronte di discesa |

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 2 | Da 100 a 115 | I bit da 00 a 15 di S corrispondono ai task ad interrupt di ingresso. |
| 3 | Da 116 a 131 | |

Nota Tutti gli interrupt di ingresso rilevati verranno cancellati quando viene modificata l'identificazione del fronte di salita o di discesa.

C200HS-INT01

Specifica dell'elaborazione degli interrupt di I/O e della maschera

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 0: numero di modulo 0 1: numero di modulo 1 2: numero di modulo 2 3: numero di modulo 3 |
| S | Maschera di interrupt. Impostare un valore compreso tra 0000 e 00FF esadecimale (8 bit per modulo). I bit singoli indicano quanto segue: 0: abilita interrupt 1: maschera interrupt |

- Nota**
1. Non è possibile utilizzare CS1W-INT01 e C200HS-INT01 nello stesso momento.
 2. Tutti gli interrupt di ingresso rilevati verranno cancellati alla cancellazione della maschera di interrupt.

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 0 | Da 100 a 107 | I bit da 00 a 07 di S corrispondono ai task ad interrupt di ingresso. |
| 1 | Da 108 a 115 | |
| 2 | Da 116 a 123 | |
| 3 | Da 124 a 131 | |

Nota Tutti gli interrupt di ingresso rilevati verranno cancellati quando viene modificata l'identificazione del fronte di salita o di discesa.

Ingressi ad interrupt integrati della CPU CJ1M**Specifica dell'elaborazione degli interrupt di I/O e della maschera**

| Operando | Contenuto |
|----------|--|
| N | Specificare il numero di interrupt di ingresso. 6: interrupt di ingresso 0 7: interrupt di ingresso 1 8: interrupt di ingresso 2 9: interrupt di ingresso 3 |
| S | Maschera di interrupt. 0000 esadecimale: interrupt abilitato (modalità diretta) 0001 esadecimale: interrupt disattivato (modalità diretta) 0002 esadecimale: avvio del contatore decrementale e abilitazione degli interrupt (modalità contatore) 0003 esadecimale: avvio del contatore incrementale e abilitazione degli interrupt (modalità contatore) |

Nota Tutti gli interrupt di ingresso rilevati verranno cancellati alla cancellazione della maschera di interrupt.

La relazione tra i numeri di interrupt di ingresso e ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di interrupt di ingresso | Numeri di task ad interrupt | |
|---------------------------------|-----------------------------|------------|
| Interrupt di ingresso 0 | 140 | CIO 296000 |
| Interrupt di ingresso 1 | 141 | CIO 296001 |
| Interrupt di ingresso 2 | 142 | CIO 296002 |
| Interrupt di ingresso 3 | 143 | CIO 296003 |

Specifica dell'identificazione con fronte di salita o di discesa dell'elaborazione degli interrupt di I/O

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di interrupt di ingresso. 10: Interrupt di ingresso 0 11: Interrupt di ingresso 1 12: Interrupt di ingresso 2 13: Interrupt di ingresso 3 |
| S | Specificare il fronte di salita o di discesa del segnale di ingresso ad interrupt. 0000 esadecimale: fronte di salita 0001 esadecimale: fronte di discesa |

La relazione tra i numeri di interrupt di ingresso e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di interrupt di ingresso | Numeri di task ad interrupt | |
|---------------------------------|-----------------------------|------------|
| Interrupt di ingresso 0 | 140 | CIO 296000 |
| Interrupt di ingresso 1 | 141 | CIO 296001 |
| Interrupt di ingresso 2 | 142 | CIO 296002 |
| Interrupt di ingresso 3 | 143 | CIO 296003 |

Nota Tutti gli interrupt di ingresso rilevati verranno cancellati quando viene modificata l'identificazione del fronte di salita o di discesa.

Specifica degli interrupt programmati

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero dell'interrupt programmato. 4: task ad interrupt 2 5: task ad interrupt 3 |
| S | 0000: disabilita interrupt programmato. Da 0001 a 270F esadecimale: intervallo interrupt a tempo (da 1 a 9999) Nota Nelle definizioni degli interrupt della configurazione del PLC è possibile impostare l'unità per l'intervallo interrupt a tempo su 10 ms o su 1,0 ms. Nelle CPU CJ1M è possibile impostare anche un'unità di 0,1 ms: l'intervallo di impostazione in questo caso sarà compreso tra 0005 e 270F esadecimale (da 5 a 9999). |

Reimpostazione e avvio degli interrupt programmati (solo CJ1M)

| Operando | Contenuto | | |
|----------|---|---|---|
| N | Specificare il numero dell'interrupt programmato. 14: interrupt programmato 0 (task ad interrupt 2) 15: interrupt programmato 1 (task ad interrupt 3) | | |
| S | Disabilitare interrupt programmato. | | 0000 esadecimale |
| | Impostare il tempo dell'intervallo interrupt programmato e avviare tale interrupt | Per un intervallo compreso tra 10 e 99,990 ms o tra 1 e 9,999 ms (cioè quando l'unità è di 10 ms o di 1 ms) | Da 0001 a 270F esadecimale |
| | | Per un intervallo compreso tra 0,5 e 999,9 ms (cioè quando l'unità è di 0,1 ms) | Da 0005 a 270F esadecimale Non è possibile utilizzare impostazioni comprese tra 0001 e 0004 esadecimale; verrà generato un errore di istruzione. |

Caratteristiche operando

| Area | N | S |
|--------------------------------------|-------------------------|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A447 Da A448 ad A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ 32767 Da @ E00000 a @ 32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Solo valori specificati | |
| Registri dati | --- | Da DR0 a DR15 |

| Area | N | S |
|--|-----|--|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0(++) a ,IR15(++) Da ,-(- -)IR0 a ,-(- -)IR15 |

Descrizione

MSKS(690) controlla gli interrupt di I/O e quelli programmati. Il valore di N identifica l'interrupt.

| Modulo | N | Significato |
|--|----------|--|
| CS1W-INT01 o CJ1W-INT01 | 0 o 1 | N corrisponde al task ad interrupt di ingresso. I bit da 0 a 7 di S coincidono con i numeri di interrupt di ingresso nel modulo ad interrupt corrispondente. MSKS(690) maschera (disattiva) l'interrupt di ingresso quando il bit corrispondente è attivato e lo smaschera (attiva) quando tale bit è disattivato. |
| C200HS-INT01 | Da 0 a 3 | |
| Ingressi ad interrupt integrati della CPU CJ1M | Da 6 a 9 | |

| Modulo | N | Significato |
|--|------------|---|
| CS1W-INT01 o CJ1W-INT01 | 2 o 3 | N corrisponde al task ad interrupt di ingresso. S indica il fronte di salita o di discesa come trigger (l'impostazione predefinita utilizza il fronte di salita). |
| Ingressi ad interrupt integrati della CPU CJ1M | Da 10 a 13 | |

- Nota**
- È possibile utilizzare MSKS(690) per abilitare un determinato task ad interrupt di I/O solo in un ciclo particolare e disabilitarlo in altri cicli.
 - I numeri di modulo sono assegnati a moduli di ingresso ad interrupt nello stesso ordine in cui tali moduli vengono installati, da sinistra a destra.

N = 4 o 5

I valori 4 e 5 corrispondono agli interrupt programmati 2 e 3.

Quando N è 4 o 5, il contenuto di S disabilita il task ad interrupt (S = 0000) o lo imposta con l'intervallo di tempo specificato. Nelle impostazioni del PLC è possibile definire le unità per l'intervallo interrupt a tempo su 10 ms, 1,0 ms o 0,1 ms.

N = 14 o 15 (solo CPU CJ1M)

Quando N è 14 o 15, l'unità di tempo degli interrupt programmati specificata in S viene impostata per il task ad interrupt programmato specificato da N e il temporizzatore interno per l'interrupt programmato viene reimpostato. L'unità di tempo per il primo interrupt di avvio del reset viene mantenuta.

- Nota**
- L'unità di tempo dell'interrupt programmato viene impostata nella configurazione del PLC.
 - Assicurarsi che l'intervallo di tempo sia maggiore del tempo necessario per eseguire il task ad interrupt programmato.
 - Per gli interrupt programmati, MSKS(690) viene utilizzata per impostare l'intervallo interrupt a tempo ma non il tempo del primo interrupt programmato. Se si desidera eseguire un controllo accurato del tempo del primo interrupt e dell'intervallo di interrupt, programmare CLI(691) appena prima di MSKS(690) per definire il tempo del primo interrupt programmato. Se MSKS(690) viene utilizzata per riavviare un interrupt programmato per una CPU CJ1M, tuttavia, il tempo del primo interrupt programmato sarà preciso anche senza l'esecuzione di CLI(691).

A440 contiene il tempo di elaborazione maggiore per i task ad interrupt e il byte più a destra di A441 contiene il numero del task ad interrupt del task che ha il tempo di elaborazione più lungo.

Flag

| Nome | Eti-chetta | Funzionamento |
|---------------------|------------|---|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato compreso tra 0 e 5 (da 0 a 15 per gli ingressi ad interrupt integrati della CJ1M). ON se S non rientra nell'intervallo specificato da 0000 a 00FF esadecimale quando N ha un valore compreso tra 0 e 3 (utilizzando un C200HS-INT01 e specificando l'elaborazione degli interrupt di I/O). ON se S non rientra nell'intervallo specificato da 0000 a 0003 esadecimale, utilizzando ingressi ad interrupt integrati della CJ1M e specificando l'elaborazione degli interrupt di I/O. ON se S non rientra nell'intervallo specificato da 0000 a 270F esadecimale quando N è 4 o 5 (da 0005 a 270F esadecimale per un ingresso ad interrupt integrato della CJ1M con un'unità di 0,1 ms). ON se l'istruzione è stata eseguita in un task ad interrupt. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF |
| Flag negativo | N | OFF |

Nella tabella seguente sono riportati i flag relativi nell'area ausiliaria.

| Nome | Indirizzo | Funzionamento |
|--|---------------------|---|
| Flag di errore task ad interrupt | A40213 | Viene attivato nei seguenti casi: 1) durante l'I/O refresh è stato eseguito un task ad interrupt più lungo di 10 ms con un modulo di I/O speciale C200H o un sistema slave di I/O remoti (solo serie CS). 2) IORF(097) è stata eseguita in un task ad interrupt senza disabilitare l'aggiornamento ciclico di I/O speciale. |
| Flag della causa di errore task ad interrupt | A42615 | Indica l'eventuale verificarsi dell' errore task interrupt 1 o 2. |
| Errore task interrupt - Numero task | Da A42600 ad A42611 | Per l'errore 1: indica il numero del task ad interrupt. Per l'errore 2: indica il numero di modulo del modulo di I/O speciale in cui sono avvenuti più refresh I/O. |

Avvertenze

Per i task ad interrupt sono supportati esclusivamente interrupt di ingresso dei moduli di ingresso ad interrupt delle normali serie CS/CJ e C200H. Gli interrupt di ingresso delle schede interne e dei moduli di I/O speciali non sono supportati.

Montare il modulo di ingresso ad interrupt nel rack CPU. Se si utilizza una CPU CJ1-H, montare il modulo negli slot da 0 a 4; nel caso di una CPU CJ1M, montarlo negli slot da 0 a 2. Il modulo di ingresso ad interrupt deve essere montato in uno di tali slot altrimenti non sarà possibile avviare il task ad interrupt di I/O.

I canali sono assegnati a moduli di ingresso ad interrupt nello stesso ordine in cui tali moduli vengono installati e cioè da sinistra a destra.

Gli interrupt presentano diversi gradi di priorità. A un interrupt di spegnimento è assegnata la priorità massima, seguito dagli interrupt di I/O, dagli interrupt esterni e infine dagli interrupt programmati. La priorità assegnata agli interrupt di I/O dotati di numeri bassi è superiore a quella assegnata agli interrupt con numeri alti.

Nel caso di un collegamento con un modulo di I/O speciale C200H o con un sistema slave di I/O remoti SYSMAC BUS, assicurarsi che il task ad interrupt

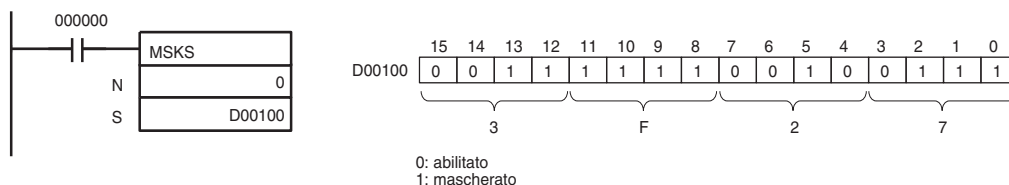
non necessiti di più di 10 ms. Se durante l'I/O refresh è stato eseguito un task ad interrupt più lungo di 10 ms con un modulo di I/O speciale C200H o un sistema slave, viene generato un errore non fatale e attivato il flag di errore task ad interrupt (A40213).

Quando IORF(097) viene eseguita in un task ad interrupt per aggiornare gli I/O in un modulo di I/O speciale, è necessario che nella configurazione del PLC l'aggiornamento ciclico di tale modulo di I/O speciale sia disabilitato. In caso contrario, sarebbe possibile che IORF(097) venisse eseguita durante l'aggiornamento ciclico generando un errore non fatale di duplicazione aggiornamento e attivando il flag di errore task ad interrupt (A40213).

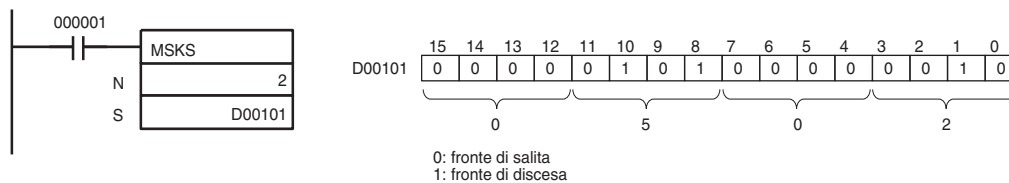
Esempi

Esempi per CS1W-INT01 / CJ1W-INT01

Nell'esempio che segue, quando CIO 000000 passa a ON, MSKS(690) smaschera (attiva) gli interrupt di ingresso nel modulo di ingresso ad interrupt 0.

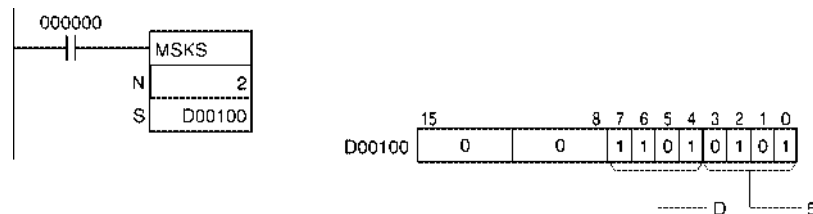


Nell'esempio che segue, quando CIO 000001 passa a ON, MSKS(690) imposta l'identificazione del fronte di salita o di discesa del modulo di ingresso ad interrupt 0.



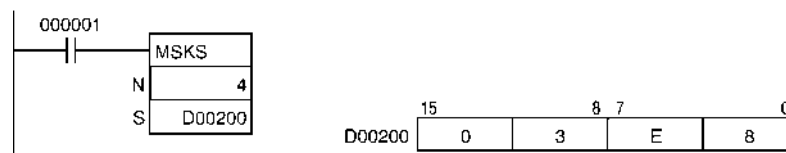
Esempio per C200HS-INT01

Nell'esempio che segue, quando CIO 000000 passa a ON, MSKS(690) smaschera (attiva) gli interrupt di ingresso 1, 3 e 5 nel modulo di ingresso ad interrupt 2.



Esempio per interrupt programmati

Nell'esempio seguente, quando CIO 000001 è ON, MSKS(690) imposta un intervallo di 10 secondi per l'intervallo programmato 2; in questo caso, nella configurazione del PLC le unità per l'intervallo di tempo programmato vengono impostate su 10 ms.

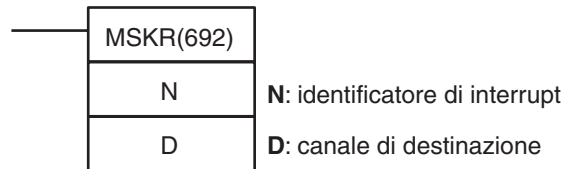


3-20-2 READ INTERRUPT MASK: MSKR(692)

Scopo Legge le impostazioni correnti di elaborazione dell'interrupt definite tramite l'istruzione MSKS(690).

MSKR(692) non è supportata dalle CPU CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MSKR(692) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MSKR(692) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

CS1W-INT01, CJ1W-INT01

Letture delle maschere

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 0: numero di modulo 0 1: numero di modulo 1 |
| D | Stato della maschera di interrupt. Da 0000 a FFFF esadecimale (16 bit per modulo) I bit singoli indicano quanto segue: 0: interrupt abilitato 1: interrupt mascherato |

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 0 | Da 100 a 115 | I bit da 00 a 15 di S corrispondono ai task ad interrupt di ingresso. |
| 1 | Da 116 a 131 | |

Letture dell'identificazione del fronte di salita o di discesa

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 2: numero di modulo 0 3: numero di modulo 1 |
| D | Il fronte di salita o di discesa del segnale di ingresso ad interrupt. Da 0000 a FFFF esadecimale (16 bit per modulo) I bit singoli indicano quanto segue: 0: fronte di salita 1: fronte di discesa |

Nota 1. Non è possibile utilizzare CS1W-INT01 e C200HS-INT01 nello stesso momento.

2. Il modulo di ingresso ad interrupt CJ1W-INT01 non può venire utilizzato con una CPU CJ1. Inoltre, non è possibile eseguire i task ad interrupt di I/O.

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 2 | Da 100 a 115 | I bit da 00 a 15 di S corrispondono ai task ad interrupt di ingresso. |
| 3 | Da 116 a 131 | |

C200HS-INT01

Letture delle maschere

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 0: numero di modulo 0 1: numero di modulo 1 2: numero di modulo 2 3: numero di modulo 3 |
| D | Stato della maschera di interrupt. Da 0000 a 00FF esadecimale (8 bit per modulo). I bit singoli indicano quanto segue: 0: interrupt abilitato 1: interrupt mascherato |

Nota Non è possibile utilizzare CS1W-INT01 e C200HS-INT01 nello stesso momento.

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 0 | Da 100 a 107 | I bit da 00 a 07 di S corrispondono ai task ad interrupt di ingresso. |
| 1 | Da 108 a 115 | |
| 2 | Da 116 a 123 | |
| 3 | Da 124 a 131 | |

Ingressi ad interrupt integrati della CPU CJ1M

Letture delle maschere

| Operando | Contenuto |
|----------|--|
| N | Specificare il numero di interrupt di ingresso. 6: Interrupt di ingresso 0 7: Interrupt di ingresso 1 8: Interrupt di ingresso 2 9: Interrupt di ingresso 3 |
| D | Maschera di interrupt. 0000 esadecimale: interrupt abilitato (modalità diretta) 0001 esadecimale: interrupt disattivato (modalità diretta) 0002 esadecimale: avvio del contatore decrementale e abilitazione degli interrupt (modalità contatore) 0003 esadecimale: avvio del contatore incrementale e abilitazione degli interrupt (modalità contatore) |

La relazione tra i numeri di interrupt di ingresso e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di interrupt di ingresso | Numeri di task ad interrupt | |
|---------------------------------|-----------------------------|------------|
| Interrupt di ingresso 0 | 140 | CIO 296000 |
| Interrupt di ingresso 1 | 141 | CIO 296001 |
| Interrupt di ingresso 2 | 142 | CIO 296002 |
| Interrupt di ingresso 3 | 143 | CIO 296003 |

Lettura dell'identificazione con fronte di salita o di discesa dell'elaborazione degli interrupt di I/O

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di interrupt di ingresso. 10: Interrupt di ingresso 0 11: Interrupt di ingresso 1 12: Interrupt di ingresso 2 13: Interrupt di ingresso 3 |
| D | Specificare il fronte di salita o di discesa del segnale di ingresso ad interrupt. 0000 esadecimale: fronte di salita 0001 esadecimale: fronte di discesa |

La relazione tra i numeri di interrupt di ingresso e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di interrupt di ingresso | Numeri di task ad interrupt | |
|---------------------------------|-----------------------------|------------|
| Interrupt di ingresso 0 | 140 | CIO 296000 |
| Interrupt di ingresso 1 | 141 | CIO 296001 |
| Interrupt di ingresso 2 | 142 | CIO 296002 |
| Interrupt di ingresso 3 | 143 | CIO 296003 |

Lettura dell'intervallo interrupt a tempo

| Operando | Contenuto |
|----------|--|
| N | Specificare il numero dell'intervallo programmato. 4: interrupt programmato 0 (task ad interrupt 2) 5: interrupt programmato 1 (task ad interrupt 3) |
| D | 0000: interrupt programmato disabilitato Da 0001 a 270F esadecimale: intervallo interrupt a tempo (da 1 a 9999) Nota Nelle definizioni degli interrupt della configurazione del PLC è possibile impostare l'unità per l'intervallo interrupt a tempo su 10 ms o su 1,0 ms. Per CJ1M è possibile impostare anche 0,1 ms per cui il tempo sarà da 0005 a 270F esadecimale (da 5 a 9999). |

Lettura del valore attuale (PV) dell'intervallo interrupt programmato

| Operando | Contenuto | |
|----------|---|--|
| N | Specificare il numero dell'intervallo programmato. 14: interrupt programmato 0 (task ad interrupt 2) 15: interrupt programmato 1 (task ad interrupt 3) | |
| D | Tempo trascorso dall'avvio dell'elaborazione dell'intervallo programmato o tempo trascorso dall'intervallo programmato precedente. Unità di 10 ms: da 0 a 99.990 Unità di 1 ms: da 0 a 9.999 Unità di 0,1 ms: da 0,0 a 999,9 (solo CJ1M) | Da 0000 a 270F esadecimale Nota: anche se l'intervallo programmato viene interrotto, è possibile leggere il tempo trascorso prima dell'interruzione. Se l'intervallo programmato non è stato avviato affatto, verrà restituito 0000 esadecimale. |

Caratteristiche operando

| Area | N | D |
|-------------------------|-----|------------------------|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A448 ad A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |

| Area | N | D |
|--|-------------------------|--|
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Solo valori specificati | --- |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

MSKR(692) legge le impostazioni dell'interrupt definite con MSKS(690).

N = 0 o 1 (da 0 a 3 per C200HS-INT01)

I valori 0 e 1 (da 0 a 3) corrispondono ai moduli di ingresso ad interrupt 0 e 1 (da 0 a 3).

I bit da 0 a 7 di D coincidono con i numeri di ingresso ad interrupt da 0 a 7 del modulo specificato. Se un bit è ON, l'interrupt di ingresso corrispondente viene mascherato (disabilitato); se un bit è OFF, l'interrupt di ingresso corrispondente viene smascherato (abilitato).

N = 2 o 3 (solo ingressi ad interrupt integrati di CS1W-INT01/CJ1W-INT01/CJ1M)

I valori 2 e 3 corrispondono ai moduli di ingresso ad interrupt 0 e 1. Le identificazioni del fronte di salita o di discesa per gli interrupt di ingresso del modulo di ingresso ad interrupt specificato con N vengono inviate a D.

N = 4 o 5

I valori 4 e 5 corrispondono agli interrupt programmati 2 e 3.

Quando N è 4 o 5, il contenuto di D indica l'intervallo di tempo impostato per tale interrupt. Un'impostazione di 0000 significa che l'interrupt è stato disabilitato. Nelle impostazioni del PLC (00: 10 ms, 01: 1,0 ms) è possibile definire le unità per l'intervallo interrupt a tempo, per cui tale intervallo può essere compreso da 10 ms a 99,99 s o da 1 ms a 9,999 s.

N = 14 o 15

Quando N è 14 o 15, il valore attuale (PV) del temporizzatore dell'interrupt programmato per il task ad interrupt programmato specificato da N viene memorizzato in D.

Flag

| Nome | Eti-chetta | Funzionamento |
|----------------|------------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato compreso tra 0 e 5 (da 0 a 15 per CJ1M). OFF in tutti gli altri casi. |

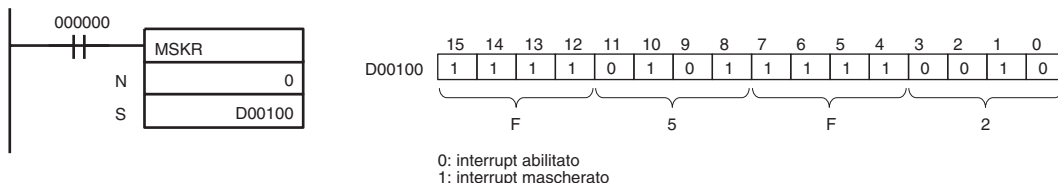
Avvertenze

MSKR(692) può essere eseguita nel programma principale o nei task ad interrupt.

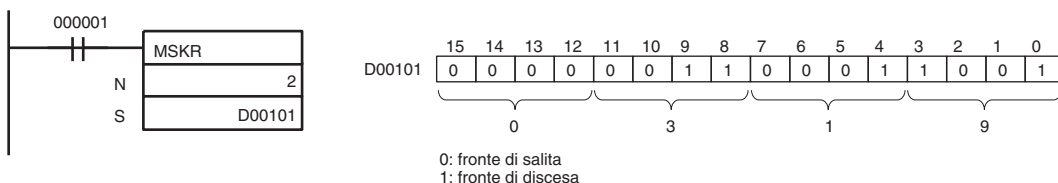
Esempi

Esempio per CS1W-INT01 / CJ1W-INT01

Nell'esempio che segue, quando CIO 000000 passa a ON, MSKR(692) legge lo stato attuale della maschera del modulo di ingresso ad interrupt 2 e lo memorizza in D00100.

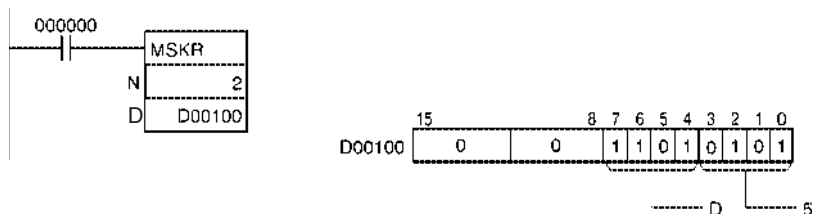


Nell'esempio che segue, quando CIO 000001 passa a ON, MSKS(690) legge l'identificazione del fronte di salita o di discesa del modulo di ingresso ad interrupt 0 e lo memorizza in D00101.



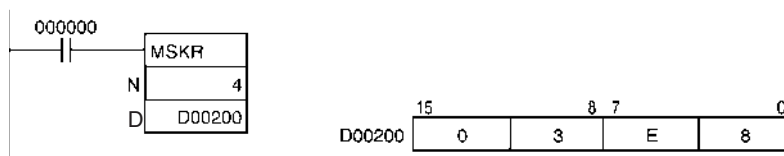
Esempio per C200H-INT01

Nell'esempio che segue, quando CIO 000000 passa a ON, MSKR(692) legge lo stato attuale della maschera del modulo di ingresso ad interrupt 2. In questo caso, gli interrupt di ingresso 1, 3 e 5 vengono abilitati.



Esempio per interrupt programmati

Nell'esempio seguente, quando CIO 000100 passa a ON, MSKR(692) legge le impostazioni dell'intervallo di tempo programmato 2. In questo caso, l'intervallo di tempo viene impostato su 1.000 (3E8 esadecimale) che equivale a 10 s, se nella configurazione del PLC le unità per l'intervallo di tempo programmato sono impostate su 10 ms.



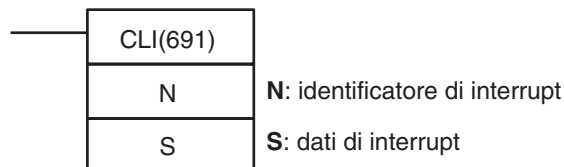
3-20-3 CLEAR INTERRUPT: CLI(691)

Scopo

Cancella o mantiene gli ingressi di interrupt registrati per gli interrupt di I/O o imposta il tempo del primo interrupt per gli interrupt programmati.

CLI(691) non è supportata dalle CPU CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CLI(691) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CLI(691) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Interrupt di I/O per CS1W-INT01 / CJ1W-INT01

| Operando | Contenuto |
|----------|---|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 0: numero di modulo 0 1: numero di modulo 1 |
| S | Specifica di eliminazione della maschera di interrupt (16 bit per modulo) Da 0000 a FFFF esadecimale Significato di ciascun bit 0: mantenuto ingresso ad interrupt registrato 1: eliminato ingresso ad interrupt registrato |

- Nota**
- Non è possibile utilizzare CS1W-INT01 e C200HS-INT01 nello stesso momento.
 - Il modulo di ingresso ad interrupt CJ1W-INT01 non può venire utilizzato con una CPU CJ1. Inoltre, non è possibile eseguire i task ad interrupt di I/O.

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 0 | Da 100 a 115 | I bit da 00 a 15 di S corrispondono ai task ad interrupt di ingresso. |
| 1 | Da 116 a 131 | |

Interrupt di I/O per C200H-INT01

| Operando | Contenuto |
|----------|--|
| N | Specificare il numero di modulo del modulo di ingresso ad interrupt. 0: numero di modulo 0 1: numero di modulo 1 2: numero di modulo 2 3: numero di modulo 3 |
| S | Specifica di eliminazione della maschera di interrupt (8 bit per modulo) Da 0000 a 00FF esadecimale Significato di ciascun bit 0: mantenuto ingresso ad interrupt registrato 1: eliminato ingresso ad interrupt registrato |

Nota Non è possibile utilizzare CS1W-INT01 e C200HS-INT01 nello stesso momento.

La relazione tra i numeri di modulo del modulo di ingresso ad interrupt e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di modulo | Numeri di task ad interrupt | |
|------------------|-----------------------------|---|
| 0 | Da 100 a 107 | I bit da 00 a 07 di S corrispondono ai task ad interrupt di ingresso. |
| 1 | Da 108 a 115 | |
| 2 | Da 116 a 123 | |
| 3 | Da 124 a 131 | |

Ingressi ad interrupt integrati della CPU CJ1M

| Operando | Contenuto |
|----------|--|
| N | Specificare il numero di interrupt di ingresso. 6: interrupt di ingresso 0 7: interrupt di ingresso 1 8: interrupt di ingresso 2 9: interrupt di ingresso 3 |
| S | Specifica di eliminazione della maschera di interrupt. 0000 esadecimale: mantenuto ingresso ad interrupt registrato 0001 esadecimale: eliminato ingresso ad interrupt registrato |

La relazione tra i numeri di interrupt di ingresso e i numeri di task ad interrupt è illustrata nella tabella che segue.

| Numero di interrupt di ingresso | Numeri di task ad interrupt | |
|---------------------------------|-----------------------------|------------|
| Interrupt di ingresso 0 | 140 | CIO 296000 |
| Interrupt di ingresso 1 | 141 | CIO 296001 |
| Interrupt di ingresso 2 | 142 | CIO 296002 |
| Interrupt di ingresso 3 | 143 | CIO 296003 |

Eliminazione degli interrupt dei contatori veloci (solo CJ1M)

| Operando | Contenuto |
|----------|--|
| N | Specificare l'ingresso del contatore veloce 10: ingresso contatore veloce 0 11: ingresso contatore veloce 1 |
| S | Specifica di eliminazione della maschera di interrupt. 0000 esadecimale: mantenuto ingresso ad interrupt registrato 0001 esadecimale: eliminato ingresso ad interrupt registrato |

Impostazione del tempo per i primi interrupt programmati

| Operando | Contenuto |
|----------|--|
| N | Specificare il numero dell'interrupt programmato. 4: interrupt programmato 0 (task ad interrupt 2) 5: interrupt programmato 1 (task ad interrupt 3) |
| S | Da 0000 a 270F esadecimale: tempo fino al primo interrupt programmato (da 0 a 9999) Nota Nelle definizioni degli interrupt della configurazione del PLC è possibile impostare l'unità per l'intervallo interrupt a tempo su 10 ms o su 1,0 ms. Per CJ1M, è possibile impostare l'unità anche su 0,1 ms. |

Caratteristiche operando

| Area | N | S |
|------------------------|-----|------------------------|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |

| Area | N | S |
|--|-------------------------|--|
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- | Da DR0 a DR15 |
| Registri dati | Solo valori specificati | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 |

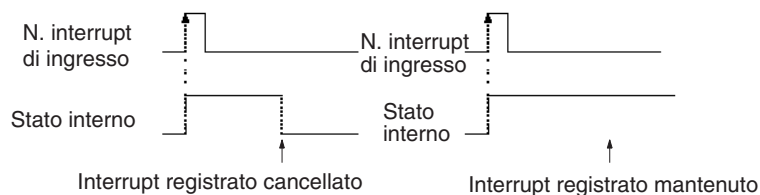
Descrizione

In base al valore di N, CLI(691) elimina gli interrupt di I/O registrati specificati oppure imposta il tempo che precede l'esecuzione del primo interrupt programmato. Con CJ1M, è possibile utilizzarla anche al fine di eliminare gli interrupt per i contatori veloci.

N = 0 o 1 (da 0 a 3 per C200HS-INT01 o da 6 a 9 per ingressi ad interrupt integrati della CPU CJ1M)

I valori 0 e 1 (da 0 a 3) corrispondono ai moduli di ingresso ad interrupt 0 e 1 (da 0 a 3).

I bit da 0 a 7 di S coincidono con i numeri di ingresso ad interrupt da 0 a 7 del modulo specificato. CLI(691) elimina un ingresso ad interrupt registrato quando il bit corrispondente di S è ON e mantiene l'ingresso ad interrupt registrato quando il bit corrispondente è OFF.



Se un task ad interrupt di I/O è in esecuzione e viene ricevuto un ingresso ad interrupt con un numero di interrupt differente, tale numero di interrupt viene registrato internamente. Gli interrupt di I/O registrati vengono eseguiti successivamente in base alla loro priorità (dal numero più basso a quello più alto). È possibile utilizzare CLI(691) per eliminare tali interrupt registrati prima della loro esecuzione.

- Nota**
1. È possibile utilizzare MSKS(690) per abilitare un determinato task ad interrupt di I/O in un ciclo particolare e disabilitarlo in altri cicli.

2. I numeri di modulo sono assegnati a moduli di ingresso ad interrupt nello stesso ordine in cui tali moduli vengono installati e cioè da sinistra a destra.

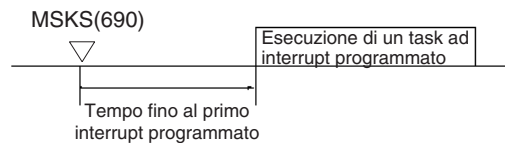
N = 4 o 5

I valori 4 e 5 corrispondono agli interrupt programmati 2 e 3.

Quando N è 4 o 5, il contenuto di S indica l'intervallo di tempo fino al primo task ad interrupt programmato dopo l'esecuzione di MSKS(690).

Tale intervallo può essere impostato da 0000 a 270F (tra 0 e 9.999). Nelle impostazioni del PLC (00: 10 ms, 01: 1,0 ms) è possibile definire le unità per l'intervallo interrupt a tempo, per cui l'intervallo effettivo può essere compreso da 10 ms a 99,99 s o da 1 ms a 9,999 s.

Nota Impostare l'intervallo fino al primo interrupt programmato su 10 ms o su un valore superiore.



N = 10 o 11 (solo CPU CJ1M)

I valori 10 e 11 corrispondono agli interrupt per i contatori veloci e possono essere utilizzati per eliminare o mantenere gli interrupt dei contatori (sia per il confronto a intervalli che per il confronto con valori di riferimento).

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | <p>ON se N non rientra nell'intervallo specificato compreso tra 0 e 5 (0, 1 o da 4 a 11 per CJ1M).</p> <p>ON se S non rientra nell'intervallo specificato da 0000 a 00FF esadecimale quando N ha un valore compreso tra 0 e 3 (solo per interrupt di I/O e C200HS-INT).</p> <p>ON se S è diverso da 0000 o 0001 esadecimale (solo per interrupt di contatori veloci e per ingressi ad interrupt integrati della CJ1M).</p> <p>ON se S non rientra nell'intervallo specificato da 0000 a 270F esadecimale per interrupt programmati.</p> <p>OFF in tutti gli altri casi.</p> |

Avvertenze

Il numero di ingressi ad interrupt registrabili è illimitato. È possibile registrare tutti gli ingressi ad interrupt di I/O, ma un nuovo ingresso ad interrupt viene ignorato se l'ingresso è già stato registrato. Inoltre, l'interrupt registrato non viene eliminato fino al completamento del proprio task ad interrupt; di conseguenza, un nuovo interrupt di ingresso è ignorato se viene ricevuto nel corso dell'esecuzione del proprio task ad interrupt.

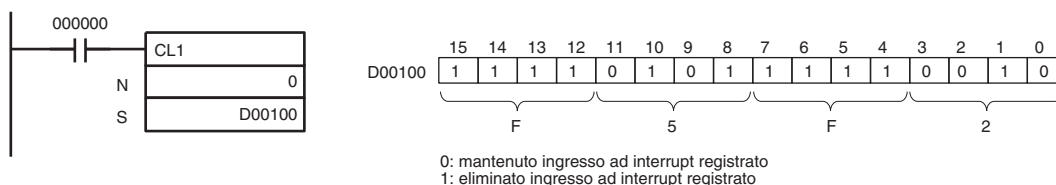
Per i task ad interrupt sono supportati esclusivamente interrupt di ingresso dei moduli di ingresso ad interrupt delle normali serie CS/CJ e C200H. Gli interrupt di ingresso delle schede interne e dei moduli di I/O speciali non sono supportati.

Gli interrupt presentano diversi gradi di priorità. A un interrupt di spegnimento è assegnata la priorità massima, seguito dagli interrupt di I/O, dagli interrupt esterni e infine dagli interrupt programmati. La priorità assegnata agli interrupt di I/O dotati di numeri bassi è superiore a quella assegnata agli interrupt con numeri alti.

Esempi

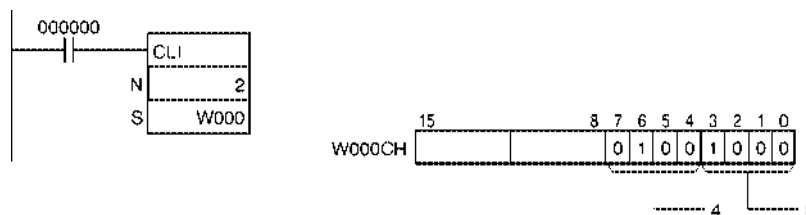
Esempi per CS1W-INT01 / CJ1W-INT01

Nell'esempio che segue, quando CIO 000000 passa a ON, CLI(691) elimina gli interrupt registrati per gli interrupt di ingresso 1, 4-8 e 12-15 nel modulo di ingresso ad interrupt 0.



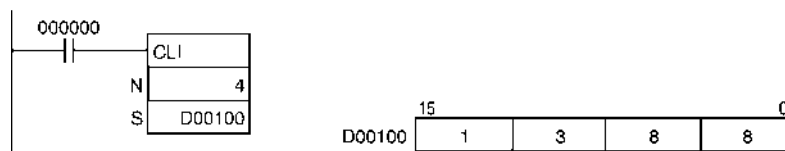
Esempio per C200HS-INT01

Nell'esempio che segue, quando CIO 000000 passa a ON, CLI(691) elimina gli interrupt registrati per gli interrupt di ingresso 3 e 6 nel modulo di ingresso ad interrupt 2.



Esempio di impostazione del tempo fino al primo interrupt programmato

Nell'esempio che segue, quando CIO 000000 passa a ON, CLI(691) imposta il tempo per la prima esecuzione dell'intervallo di tempo programmato 2 su 50 secondi. In tal caso, nella configurazione del PLC, le unità per l'intervallo di tempo programmato sono impostate su 10 ms.



3-20-4 DISABLE INTERRUPTS: DI(693)

Scopo

Disabilita l'esecuzione dei task ad interrupt eccetto quello di spegnimento. Quando si utilizza una CPU CS1D per sistemi a singola CPU oppure CPU CS1-H, CJ1-H o CJ1M e il task ad interrupt di spegnimento è disattivato, è possibile disabilitare contemporaneamente l'elaborazione dell'intervallo di spegnimento.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | DI(693) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DI(693) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Descrizione

DI(693) viene eseguita dal programma principale per disattivare temporaneamente tutti i task ad interrupt ad eccezione dell'intervallo di spegnimento (interrupt di I/O, interrupt programmati e interrupt esterni).

Tutti i task ad interrupt saranno disattivati fino alla successiva abilitazione prodotta dall'esecuzione di EI(694).

CPU CS1-H, CJ1-H e CJ1M e interrupt di spegnimento

Nel caso di CPU CS1-H, CJ1-H e CJ1M, è possibile disattivare l'elaborazione dell'interrupt di spegnimento nello stesso istante in cui A503 (la disabilitazione dell'impostazione per gli interrupt di spegnimento) viene impostata su A5A5 esadecimale. Anche se viene rilevata una caduta di tensione dopo l'esecuzione di DI(693), la CPU sarà reimpostata quando le istruzioni del programma sono state eseguite in sequenza fino all'istruzione EI(694) o END(001) dell'ultimo task.

Se il task ad interrupt di spegnimento è abilitato, la CPU viene reimpostata al termine dell'esecuzione di tale task. Per informazioni dettagliate sul task ad interrupt di spegnimento, consultare il *Manuale di programmazione dei PLC della serie CS/CJ*.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se DI(693) viene eseguita da un task ad interrupt. OFF in tutti gli altri casi. |

Flag e canali correlati

Il canale seguente è nell'area ausiliaria.

| Nome | Indirizzo | Contenuto |
|--|-----------|--|
| Disabilitazione dell'impostazione per gli interrupt di spegnimento | A530 | A5A5 esadecimale: attiva la disabilitazione dell'impostazione per gli interrupt di spegnimento. La procedura di spegnimento (esclusa l'esecuzione del task ad interrupt di spegnimento) è mascherata tra le istruzioni DI(694) ed EI(694); di conseguenza, le istruzioni fino a quest'ultima vengono eseguite. |

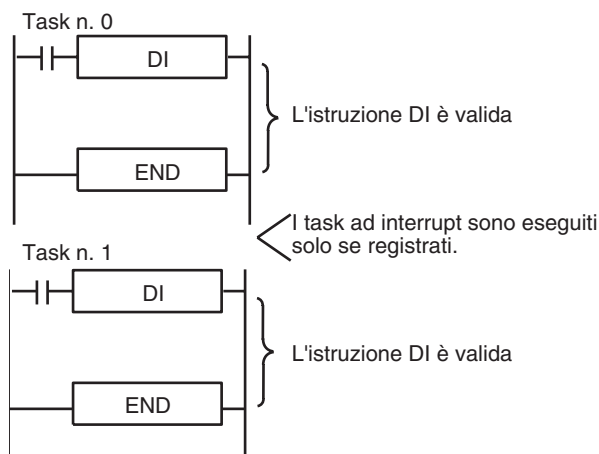
Avvertenze

Tutti i task ad interrupt restano disattivati fino all'esecuzione di EI(694).

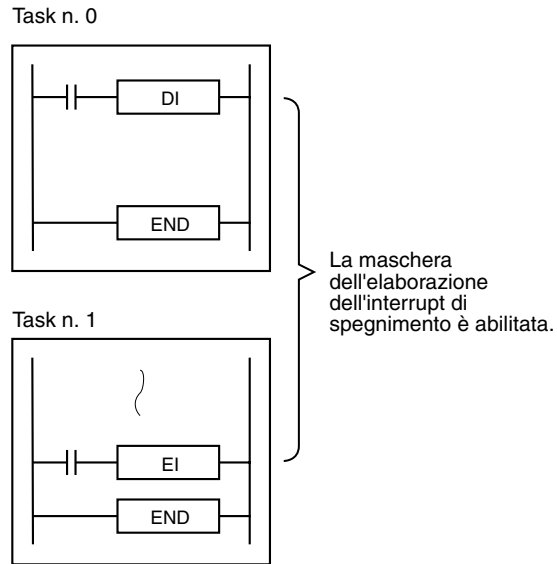
Non è possibile eseguire DI(693) da un task ad interrupt

Non è possibile eseguire DI(693) per più di un task ciclico Per disattivare più task di esecuzione del ciclo, inserire DI(693) in ciascun task ciclico. Tutti gli interrupt che si verificano durante l'esecuzione di un task di esecuzione del ciclo vengono eseguiti al termine dell'esecuzione di tale task, a meno che non siano disattivati da CLI(691). Osservare l'esempio seguente.

Se si utilizza DI(693) per disattivare l'elaborazione dell'interrupt di spegnimento in CPU CS1-H, CJ1-H e CJ1M, è possibile disattivare l'elaborazione mediante i task ciclici. La condizione disattivata viene rilasciata al completamento di tutti i task avviati.

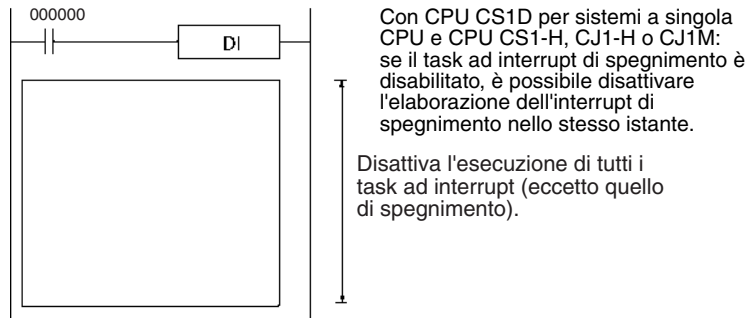


Nel caso di CPU CS1D per sistemi a singola CPU oppure di CPU CS1-H, CJ1-H o CJ1M, il task ad interrupt di spegnimento è disattivato e A530 viene impostata su A5A5 esadecimale; la CPU sarà nuovamente impostata al termine dell'esecuzione di EI(694) nel caso in cui venga rilevata una caduta di tensione durante l'esecuzione delle istruzioni comprese tra DI(693) e EI(694).



Esempi

Nell'esempio che segue, quando CIO 000000 è ON, DI(693) disattiva tutti i task ad interrupt tranne quello di spegnimento.



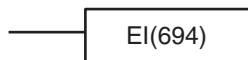
3-20-5 ENABLE INTERRUPTS: EI(694)

Scopo

Abilita l'esecuzione di tutti i task ad interrupt che erano stati disabilitati tramite l'istruzione DI(693).

Nel caso di CPU CS1D per sistemi a singola CPU oppure di CPU CS1-H, CJ1-H o CJ1M e se il task ad interrupt di spegnimento è disattivato, EI(694) rilascia contemporaneamente l'elaborazione disattivata dell'interrupt di spegnimento.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione normalmente ON | EI(694) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Descrizione

EI(694) viene eseguita dal programma principale per abilitare temporaneamente tutti i task ad interrupt disattivati da DI(693). Ad eccezione di quello di spegnimento, DI(693) disattiva tutti gli interrupt: di I/O, programmati ed esterni.

CPU CS1-H, CJ1-H e CJ1M e interrupt di spegnimento

Nel caso di CPU CS1-H, CJ1-H e CJ1M, se l'elaborazione dell'interrupt di spegnimento è stata disattivata con DI(693), EI(694) rilascerà anche il blocco sull'elaborazione dell'interrupt di spegnimento. Terminata l'esecuzione di DI(593), la CPU non sarà reimpostata nemmeno se viene rilevata una caduta di tensione. La CPU viene impostata nuovamente al termine dell'esecuzione di tutte le istruzioni comprese tra DI(693) ed EI(694). Per ulteriori informazioni sull'utilizzo dell'istruzione DI(693) per la disabilitazione dell'elaborazione dell'interrupt di spegnimento, consultare 3-20-4 *DISABLE INTERRUPTS: DI(693)*:

Flag

| Nome | Eti-chetta | Funzionamento |
|----------------|------------|---|
| Flag di errore | ER | ON se EI(694) viene eseguita da un task ad interrupt. OFF in tutti gli altri casi. |

Flag e canali correlati

Il canale seguente è nell'area ausiliaria.

| Nome | Indirizzo | Contenuto |
|--|-----------|---|
| Disabilitazione dell'impostazione per gli interrupt di spegnimento | A530 | A5A5 esadecimale: attiva la disabilitazione dell'impostazione per gli interrupt di spegnimento. La procedura di spegnimento (esclusa l'esecuzione del task ad interrupt di spegnimento) è mascherata tra le istruzioni DI(694) ed EI(694); di conseguenza, le istruzioni fino a quest'ultima vengono eseguite. Qualsiasi altro valore: disattiva la maschera della procedura di spegnimento. |

Avvertenze

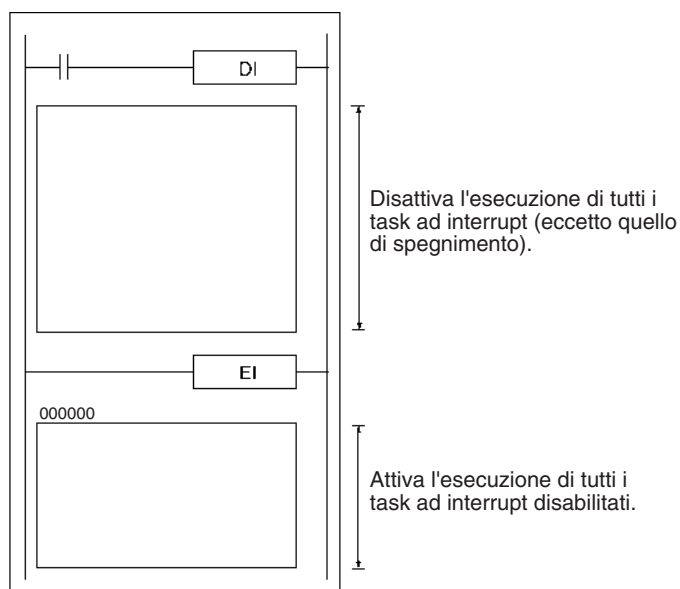
L'istruzione EI(694) non richiede alcuna condizione di esecuzione. Viene eseguita sempre con condizione di esecuzione impostata su ON. EI(694) attiva i task ad interrupt che erano stati disabilitati da DI(693).

Non è in grado di attivare gli interrupt di I/O che non sono stati smascherati da MSKS(690) o di impostare interrupt programmati che non sono stati definiti da MSKS(690).

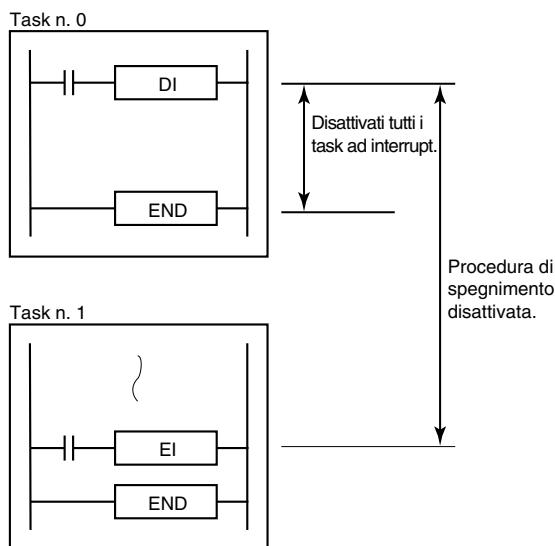
Non è possibile eseguire EI(694) in un task ad interrupt

Esempi

Nell'esempio seguente, EI(694) attiva tutti i task ad interrupt che erano stati disabilitati da DI(693).



Nota Quando il task ad interrupt di spegnimento è disattivato per CPU CS1-H, CJ1-H e CJ1M o per CPU CS1D per sistemi a singola CPU, la procedura di spegnimento verrà a sua volta abilitata nello stesso istante.



3-20-6 Riepilogo del controllo degli interrupt

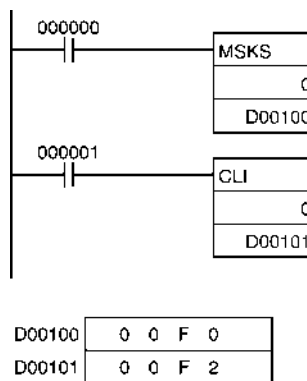
Le istruzioni di controllo degli interrupt verificano o leggono le impostazioni degli interrupt di I/O e degli interrupt programmati. DI(693) e EI(694) controllano il funzionamento degli interrupt esterni, di quelli programmati e di I/O.

Le istruzioni che intervengono su interrupt singoli hanno un operando, N, che identifica l'origine dell'interrupt. I numeri da 0 a 3 indicano i moduli di interrupt di ingresso compresi tra 0 e 3 e i numeri da 4 e 5 indicano gli interrupt programmati 2 e 3.

Elaborazione degli interrupt di I/O (N = da 0 a 3)

Un interrupt di I/O è causato da un segnale di ingresso proveniente da un modulo di ingresso ad interrupt. È possibile connettere al PLC fino a quattro moduli di ingresso ad interrupt. I numeri di modulo da 0 a 3 sono assegnati ai moduli in base loro posizione nel PLC da sinistra a destra.

Il seguente esempio di programma illustra il funzionamento di MSKS(690) e CLI(691) quando sono utilizzate per controllare gli interrupt di I/O.



Funzionamento di MSKS(690)

Alla prima accensione del PLC, sia i task ad interrupt di I/O che i task programmati sono mascherati (disattivati). Per smascherare e mascherare gli interrupt di I/O e per impostare gli intervalli di tempo per gli interrupt programmati, è possibile utilizzare l'istruzione MSKS(690).

Nell'esempio, MSKS(690) utilizza il contenuto di D00100 per smascherare gli interrupt di ingresso da 0 a 3 e mascherare gli interrupt di ingresso da 4 a 7 dal modulo di ingresso ad interrupt 0.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | F | | | 0 | | | |
| Interrupt di ingresso dal modulo 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Impostazioni della maschera di interrupt. | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

1 = maschera (disattiva) 0 = smaschera (attiva)

Quando l'interrupt di ingresso 3 passa da OFF a ON, l'esecuzione del programma principale viene interrotta e il task ad interrupt di I/O 3 (task ad interrupt 103) sarà eseguito. Al completamento del task ad interrupt di I/O 3, l'esecuzione del programma principale riprende dal punto di interruzione.

Task ad interrupt di I/O Livelli di priorità

Quando due o più interrupt di ingresso vengono ricevuti contemporaneamente, sono eseguiti in base al loro numero di interrupt, dal più basso al più alto (da 100 a 131).

| Modulo | Task ad interrupt |
|-----------------------------------|---|
| Modulo di ingresso ad interrupt 0 | Gli ingressi da 0 a 7 corrispondono ai task ad interrupt di I/O compresi tra 100 e 107. |
| Modulo di ingresso ad interrupt 1 | Gli ingressi da 0 a 7 corrispondono ai task ad interrupt di I/O compresi tra 108 e 115. |
| Modulo di ingresso ad interrupt 2 | Gli ingressi da 0 a 7 corrispondono ai task ad interrupt di I/O compresi tra 116 e 123. |
| Modulo di ingresso ad interrupt 3 | Gli ingressi da 0 a 7 corrispondono ai task ad interrupt di I/O compresi tra 124 e 131. |

Quando più interrupt di ingresso vengono ricevuti mentre un task ad interrupt è in fase di esecuzione, gli interrupt registrati sono eseguiti in base alla loro priorità, una volta che il task ad interrupt corrente è completato.

Se si verifica un interrupt programmato, il task ad interrupt programmato avrà la priorità rispetto ai task ad interrupt di I/O.

Funzionamento di CLI(691)

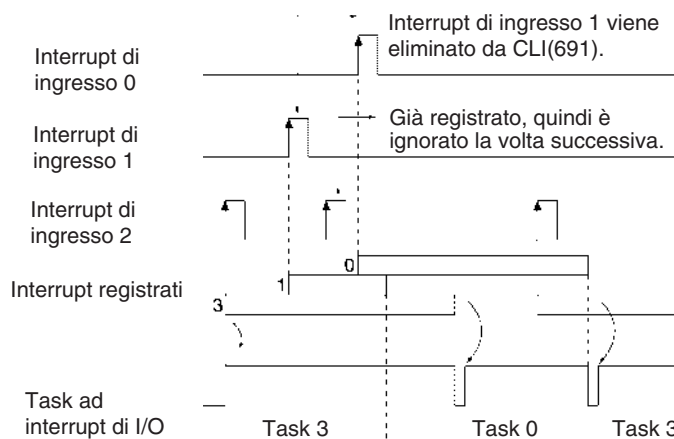
Se un interrupt di ingresso viene ricevuto mentre un altro task ad interrupt di I/O è in fase di esecuzione, il numero di interrupt di ingresso viene registrato internamente fino al completamento del task corrente e di tutti i task con priorità più elevata. È possibile utilizzare CLI(691) per eliminare interrupt registrati prima dell'esecuzione, ma non i task ad interrupt in fase di esecuzione.

Nell'esempio che segue, CLI(691) utilizza il contenuto di D00101 per eliminare dal modulo di ingresso ad interrupt 0 tutti gli interrupt di ingresso registrati, con eccezione degli ingressi 0, 2 e 3.

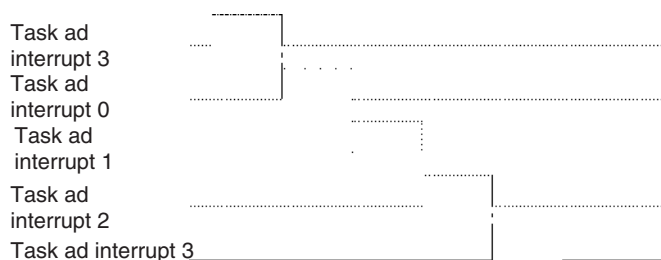
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | F | | | 2 | | | | |
| Interrupt di ingresso dal modulo 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Impostazioni di eliminazione/mantenimento per interrupt | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

1 = elimina ingresso registrato 0 = mantieni ingresso registrato

Al completamento del task ad interrupt 3, gli interrupt registrati vengono eseguiti in base alla loro priorità. Poiché è stato registrato un ingresso dall'intervallo di ingresso 0, il task ad interrupt di I/O 0 (task ad interrupt 100) viene eseguito al completamento del task 3. L'intervallo di ingresso 1 non è mantenuto da CLI(691), per cui tale ingresso viene eliminato.



Se tutti gli interrupt di ingresso da 0 a 3 passano a ON e CLI(691) non è eseguita, tutti gli ingressi vengono registrati e i task ad interrupt saranno eseguiti in sequenza al completamento del task ad interrupt 3. I task ad interrupt vengono eseguiti in base alla loro priorità (dal numero più basso a quello più alto).



- Nota**
1. Non sempre è necessario utilizzare CLI(691).
 2. Quando CLI(691) non viene eseguita, tutti gli interrupt di ingresso di I/O ricevuti durante l'esecuzione di un task ad interrupt saranno registrati. Se un ingresso registrato viene ricevuto nuovamente, alla seconda occasione verrà ignorato.
 3. Quando uno o più interrupt di ingresso di I/O sono registrati, vengono eseguiti in base alla loro priorità. L'ordine secondo il quale gli ingressi registrati vengono ricevuti è irrilevante.

Elaborazione degli interrupt programmati (N = 4 o 5)

Un interrupt programmato viene ripetuto a intervalli regolari definiti con MSKS(690) e indipendenti dal tempo del ciclo del PLC. I numeri N 4 e 5 corrispondono rispettivamente ai numeri di interrupt programmati 2 e 3.

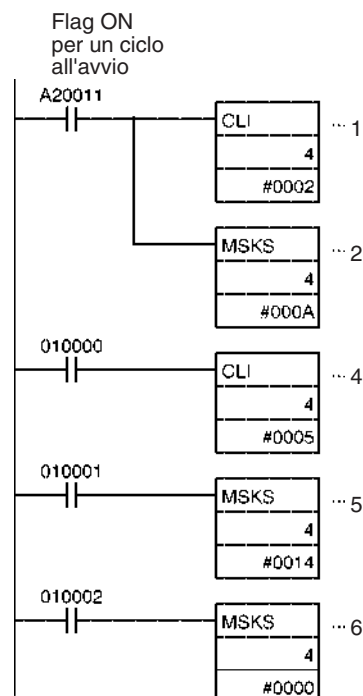
Elaborazione degli interrupt programmati

Le caratteristiche principali dell'elaborazione degli interrupt programmati sono elencate qui di seguito.

- 1,2,3...**
1. Gli interrupt programmati sono mascherati (disattivati) all'accensione iniziale del PLC.
 2. Impostare il tempo fino al primo interrupt programmato (in seguito all'esecuzione di MSKS(690)) con CLI(691). Se il tempo fino al primo interrupt programmato non viene impostato con CLI(691), non sarà possibile prevederlo.
 3. Impostazione dell'intervallo programmato ed elaborazione dell'intervallo
 - Impostare l'intervallo programmato tramite l'istruzione MSKS(690).
 - Una volta eseguita MSKS(690) e trascorso l'intervallo fino al primo interrupt programmato impostato con CLI(691), il task in fase di elaborazione è interrotto e il task ad interrupt programmato viene eseguito.
 - Quando l'esecuzione del task ad interrupt programmato raggiunge un'istruzione END(001), l'esecuzione del programma riprende dal punto in cui si era verificato l'intervallo programmato.
 - L'esecuzione del programma viene interrotta e il task ad interrupt programmato è nuovamente eseguito una volta trascorso l'intervallo programmato. Il task ad interrupt programmato viene eseguito ripetutamente fino alla propria disattivazione.
 4. Disattivazione di un interrupt programmato
 - È possibile disattivare un task ad interrupt programmato utilizzando l'istruzione MSKS(690) per impostare l'intervallo programmato su 0000.
 - Alla successiva attivazione del task ad interrupt programmato, assicurarsi di impostare il tempo fino al primo interrupt programmato con CLI(691) prima di ridefinire l'intervallo programmato con MSKS(690).

Funzionamento dell'intervallo programmato

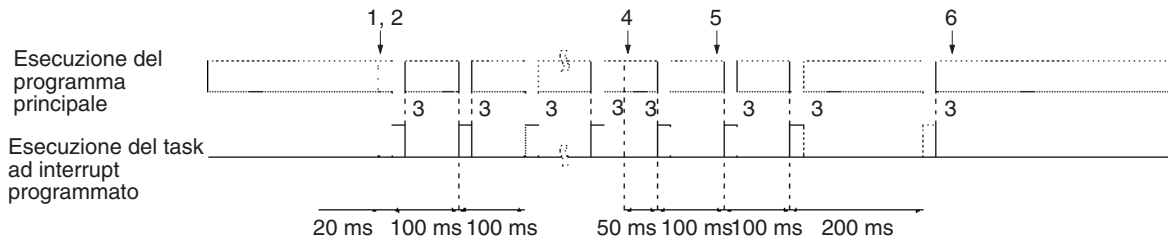
Nell'esempio che segue, le unità per l'intervallo programmato sono impostate su 10 ms nella configurazione del PLC.



- 1,2,3...**
1. Il tempo fino al primo interrupt programmato è impostato su 20 ms con CLI(691).

2. L'intervallo programmato è impostato su 100 ms e l'esecuzione dell'intervallo programmato 2 è attivata con MSKS(690).
3. L'intervallo programmato 2 viene eseguito 20 ms dopo l'esecuzione di MSKS(690) e successivamente ogni 100 ms.
4. Una volta avviata l'elaborazione dell'intervallo programmato, il tempo fino all'intervallo programmato successivo può essere stato modificato con CLI(690), ma l'impostazione resta effettiva una volta soltanto.
5. Una volta avviata l'elaborazione dell'intervallo programmato, l'intervallo programmato può essere modificato da MSKS(690). In tal caso, l'intervallo è modificato da 100 ms a 200 ms.
6. È possibile disattivare l'elaborazione di un intervallo programmato eseguendo MSKS(690) con un intervallo di 0000.

Il seguente diagramma temporale illustra l'esempio appena citato.

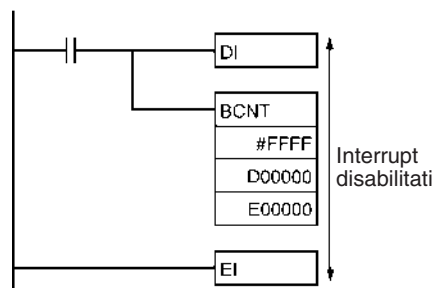
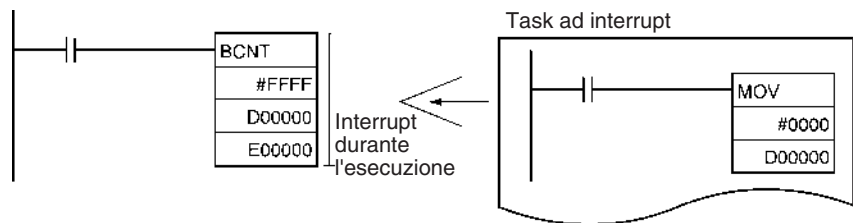


Avvertenze

Assicurarsi che l'intervallo programmato sia maggiore del tempo necessario per eseguire il task ad interrupt programmato. Se l'intervallo è troppo breve, il task ad interrupt sarà eseguito ininterrottamente e verrà generato un errore di tempo di ciclo troppo lungo (un task ad interrupt programmato troppo lungo può incidere seriamente sul tempo complessivo di esecuzione del programma principale).

L'intervallo programmato è eseguito al termine dell'intervallo specificato e del tempo di esecuzione di un'istruzione. In linea generale, il tempo necessario per eseguire un'istruzione è irrisorio; se tuttavia un'istruzione richiede tempi lunghi di esecuzione, è possibile che si verifichino errori eventualmente anche nei temporizzatori (TIM e TIMH) e nella registrazione dei dati. Se nella configurazione del PLC le unità per l'intervallo di tempo programmato sono impostate su 0,5 ms o su 1, è necessario prestare particolare attenzione.

Gli interrupt vengono accettati anche quando un'istruzione è in fase di esecuzione. Di conseguenza, se un interrupt viene accettato mentre è in esecuzione un'istruzione che richiede tempi lunghi di elaborazione, i risultati dell'elaborazione potrebbero non essere corretti, in quanto il task ad interrupt e l'istruzione hanno accesso agli stessi dati. In tal caso, utilizzare DI(693) ed EI(694) per disattivare e attivare l'intervallo.



3-21 Istruzioni per contatore veloce e uscita a impulsi

In questa sezione vengono descritte le istruzioni utilizzate per controllare contatori veloci e uscite a treno di impulsi.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|---------------------------------|-----------|-----------------|--------|
| MODE CONTROL | INI | 880 | 823 |
| HIGH-SPEED COUNTER PV READ | PRV | 881 | 827 |
| COUNTER FREQUENCY CONVERT | PRV2 | 881 | 833 |
| REGISTER COMPARISON TABLE | CTBL | 882 | 837 |
| SPEED OUTPUT | SPED | 885 | 841 |
| SET PULSES | PULS | 886 | 846 |
| PULSE OUTPUT | PLS2 | 887 | 849 |
| ACCELERATION CONTROL | ACC | 888 | 855 |
| ORIGIN SEARCH | ORG | 889 | 862 |
| PULSE WITH VARIABLE DUTY FACTOR | PWM | 891 | 865 |

3-21-1 MODE CONTROL: INI(880) (solo CJ1M-CPU21/22/23)

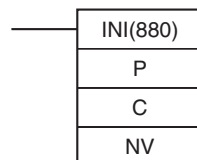
Scopo

È possibile utilizzare l'istruzione INI(880) per eseguire le seguenti operazioni con gli I/O integrati delle CPU CJ1M:

- Iniziare un confronto con la tabella di confronto del contatore veloce.
- Interrompere un confronto con la tabella di confronto del contatore veloce.
- Modificare il valore attuale del contatore veloce.
- Modificare il valore attuale degli ingressi ad interrupt in modalità contatore.
- Modificare il valore attuale dell'uscita a treno di impulsi (origine fissata a 0).
- Interrompere l'uscita a treno di impulsi.

Questa istruzione è supportata solo dalle CPU CJ1M-CPU21/22/23.

Simbolo programmazione ladder



P: identificatore della porta
C : dati di controllo
NV: primo canale contenente il nuovo valore attuale

Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | INI(880) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @INI(880) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

P: identificatore della porta

P specifica la porta a cui deve essere applicata l'operazione.

| P | Porta |
|------------------|-----------------------------|
| 0000 esadecimale | Uscita a treno di impulsi 0 |
| 0001 esadecimale | Uscita a treno di impulsi 1 |
| 0010 esadecimale | Contatore veloce 0 |
| 0011 esadecimale | Contatore veloce 1 |

| P | Porta |
|------------------|---|
| 0100 esadecimale | Ingresso ad interrupt 0 in modalità contatore |
| 0101 esadecimale | Ingresso ad interrupt 1 in modalità contatore |
| 0102 esadecimale | Ingresso ad interrupt 2 in modalità contatore |
| 0103 esadecimale | Ingresso ad interrupt 3 in modalità contatore |
| 1000 esadecimale | Uscita PWM(891) 0 |
| 1001 esadecimale | Uscita PWM(891) 1 |

C: dati di controllo

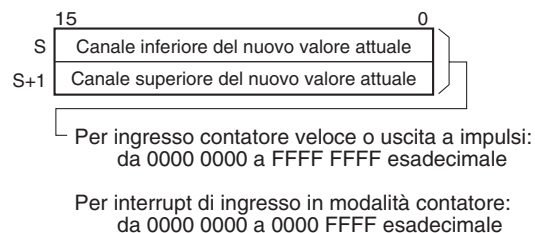
La funzione dell'istruzione INI(880) è determinata dai dati di controllo C.

| C | Funzione di INI(880) |
|------------------|---|
| 0000 esadecimale | Inizia il confronto. |
| 0001 esadecimale | Interrompe il confronto. |
| 0002 esadecimale | Modifica il valore attuale. |
| 0003 esadecimale | Interrompe l'uscita a treno di impulsi. |

NV: primo canale contenente il nuovo valore attuale

Quando si modifica il valore attuale, NV e NV+1 contengono il nuovo valore attuale.

Se C è 0002 esadecimale (ovvero si sta modificando un valore attuale), NV e NV+1 contengono il nuovo valore attuale. Se C non è 0002 esadecimale, i valori presenti in NV e NV+1 vengono ignorati.

**Caratteristiche operando**

| Area | P | C | NV |
|--------------------------------------|--------------------------------------|--------------------------------------|------------------------|
| Area CIO | --- | --- | Da CIO 0000 a CIO 6142 |
| Area di lavoro | --- | --- | Da W000 a W510 |
| Area bit di ritentività | --- | --- | Da H000 a H510 |
| Area bit ausiliaria | --- | --- | Da A448 ad A958 |
| Area del temporizzatore | --- | --- | Da T0000 a T4094 |
| Area del contatore | --- | --- | Da C0000 a C4094 |
| Area DM | --- | --- | Da D00000 a D32766 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | --- | Da *D00000 a *D32767 |
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. | --- |
| Registri dati | --- | --- | --- |

| Area | P | C | NV |
|--|-----|-----|--|
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047 ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione INI(880) esegue l'operazione specificata in C per la porta specificata in P. Nella seguente tabella sono riportate le possibili combinazioni di operazioni e porte.

| P: identificatore della porta | C: dati di controllo | | | |
|---|---|---|---|---|
| | 0000 esadecimale: inizio del confronto | 0001 esadecimale: interruzione del confronto | 0002 esadecimale: modifica del valore attuale | 0003 esadecimale: interruzione uscita a treno di impulsi |
| 0000 o 0001 esadecimale: uscita a treno di impulsi | Non consentita | Non consentita | OK | OK |
| 0010 o 0011 esadecimale: ingresso contatore veloce | OK | OK | OK | Non consentita |
| 0100, 0101, 0102 o 0103 esadecimale: ingresso ad interrupt in modalità contatore | Non consentita | Non consentita | OK | Non consentita |
| 1000 o 1001 esadecimale: uscita PWM(891) | Non consentita | Non consentita | Non consentita | OK |

■ **Inizio del confronto (C = 0000 esadecimale)**

Se C è 0000 esadecimale, l'istruzione INI(880) inizia il confronto del valore attuale di un contatore veloce con la tabella di confronto registrata utilizzando l'istruzione CTBL(882).

Nota È necessario che sia stata già registrata una tabella di confronto contenente i valori di riferimento utilizzando l'istruzione CTBL(882). Se l'istruzione INI(880) viene eseguita senza che sia stata precedentemente registrata una tabella di confronto, verrà attivato il flag di errore.

■ **Interruzione del confronto (C = 0001 esadecimale)**

Se C è 0001 esadecimale, l'istruzione INI(880) interrompe il confronto del valore attuale di un contatore veloce con la tabella di confronto registrata utilizzando l'istruzione CTBL(882).

■ Modifica di un valore attuale (C = 0002 esadecimale)

Se C è 0002 esadecimale, l'istruzione INI(880) modifica un valore attuale come descritto nella seguente tabella.

| Porta e modalità | | | Funzionamento | Intervallo di impostazione |
|---|--------------------|--|---|--|
| Uscita a treno di impulsi (P = 0000 o 0001 esadecimale) | | | Il valore attuale dell'uscita a treno di impulsi viene modificato con il nuovo valore specificato in NV e NV+1. Nota: è possibile eseguire questa istruzione solo se l'uscita a treno di impulsi è stata interrotta. Se viene eseguita durante l'uscita a treno di impulsi, si verificherà un errore. | Da 8000 0000 a 7FFF FFFF esadecimale (-2.147.483.648 ... 2.147.483.647) |
| Ingresso contatore veloce (P = 0010 o 0011 esadecimale) | Modalità lineare | Ingressi differenziali, impulsi incrementali/decrementali o ingressi impulsi + direzione | Il valore attuale del contatore veloce viene modificato con il nuovo valore specificato in NV e NV+1. Nota: se la porta specificata non è impostata come contatore veloce, l'istruzione genererà un errore. | Da 8000 0000 a 7FFF FFFF esadecimale (-2.147.483.648 ... 2.147.483.647) |
| | | Ingresso a impulsi incrementale | | 0000 0000 ... FFFF FFFF esadecimale (0 ... 4.294.967.295) |
| | Modalità circolare | 0000 0000 ... FFFF FFFF esadecimale (0 ... 4.294.967.295) | | |
| Ingresso ad interrupt in modalità contatore (P = 0100, 0101, 0102 o 0103 esadecimale) | | | Il valore attuale dell'ingresso ad interrupt viene modificato con il nuovo valore specificato in NV e NV+1. | da 0000 0000 a 0000 FFFF esadecimale (0 ... 65.535) Nota: se il valore non è compreso nell'intervallo specificato, si verificherà un errore. |

■ Interruzione dell'uscita a treno di impulsi (P = 1000 o 1001 esadecimale e C = 0003 esadecimale)

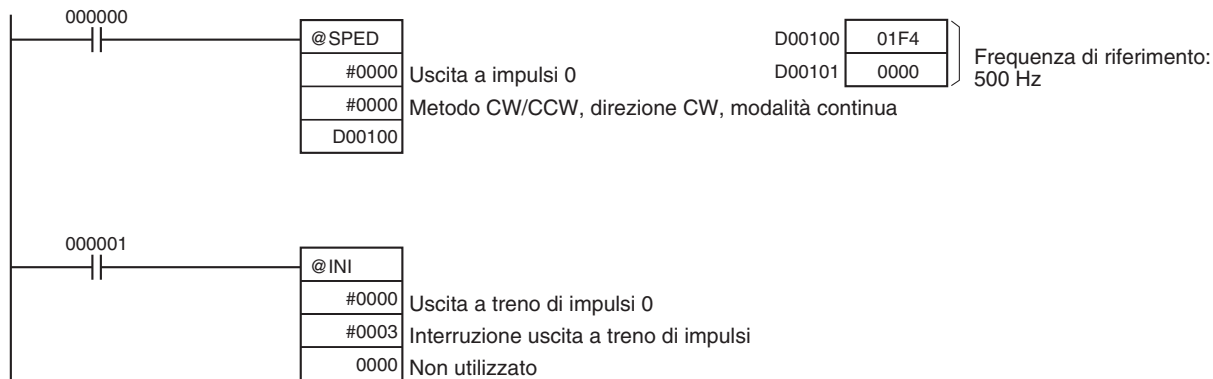
Se C è 0003 esadecimale, l'istruzione INI(880) interrompe immediatamente l'uscita a treno di impulsi per la porta specificata. Se l'istruzione viene eseguita quando l'uscita a treno di impulsi è già stata interrotta, l'impostazione della quantità di impulsi verrà cancellata.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se viene superata la gamma specificata per P, C o NV.</p> <p>ON se la combinazione di P e C non è consentita.</p> <p>ON se è specificato l'inizio di un confronto ma non è stata registrata alcuna tabella di confronto.</p> <p>ON se viene specificato un nuovo valore attuale per una porta da cui è in corso un'uscita di impulsi.</p> <p>ON se viene specificata una modifica del valore attuale di un contatore veloce per una porta non impostata come contatore veloce.</p> <p>ON se come valore attuale per un ingresso ad interrupt in modalità contatore viene specificato un valore non compreso nell'intervallo.</p> <p>ON se viene eseguita l'istruzione INI(880) in un task ad interrupt per un contatore veloce e si verifica un interrupt durante l'esecuzione di CTBL(882).</p> <p>ON se l'istruzione viene eseguita per una porta non impostata come ingresso a interrupt in modalità contatore.</p> |

Esempio

Nell'esempio seguente, quando CIO 000000 viene impostato su ON, SPED(885) inizia a inviare impulsi dall'uscita a treno di impulsi 0 in modalità continua a 500 Hz. Quando CIO 000001 passa a ON, l'uscita a treno di impulsi viene interrotta dall'istruzione INI(880).



3-21-2 HIGH-SPEED COUNTER PV READ: PRV(881) (solo CJ1M-CPU21/22/23)

Scopo

È possibile utilizzare l'istruzione PRV(881) per leggere i seguenti dati dagli I/O integrati delle CPU CJ1M.

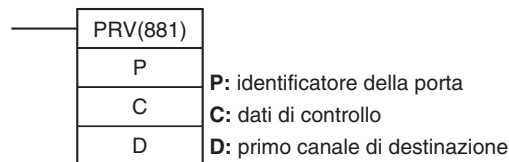
- Valori attuali: valore attuale di un contatore veloce, di un'uscita a treno di impulsi, di un ingresso ad interrupt in modalità contatore.
- Di seguito sono riportate le informazioni sullo stato.

| Tipo di stato | Contenuto |
|--------------------------------------|--|
| Stato dell'uscita a treno di impulsi | Flag di stato dell'uscita a treno di impulsi Flag di overflow/underflow per il valore attuale Flag di valore impostato per l'uscita a treno di impulsi Flag di uscita a treno di impulsi completata Flag dell'uscita a treno di impulsi Flag di nessuna origine Flag di posizione su origine Flag di errore di uscita a treno di impulsi interrotta |
| Stato ingresso contatore veloce | Flag di confronto in corso Flag di overflow/underflow per il valore attuale |
| Stato uscita PWM(891) | Flag uscita a treno di impulsi in corso |

- Risultati dei confronti a intervalli
- Frequenza dell'uscita a treno di impulsi dell'uscita a impulsi 0 o 1 (supportata solo dalle CPU CJ1M versione 2.0 o successive).
- Frequenza contatore veloce per ingresso del contatore veloce 0.

Questa istruzione è supportata solo dalle CPU CJ1M-CPU21/22/23.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | PRV(881) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PRV(881) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

P: identificatore della porta

P specifica la porta a cui deve essere applicata l'operazione.

| P | Porta |
|------------------|---|
| 0000 esadecimale | Uscita a treno di impulsi 0 |
| 0001 esadecimale | Uscita a treno di impulsi 1 |
| 0010 esadecimale | Contatore veloce 0 |
| 0011 esadecimale | Contatore veloce 1 |
| 0100 esadecimale | Ingresso ad interrupt 0 in modalità contatore |
| 0101 esadecimale | Ingresso ad interrupt 1 in modalità contatore |
| 0102 esadecimale | Ingresso ad interrupt 2 in modalità contatore |
| 0103 esadecimale | Ingresso ad interrupt 3 in modalità contatore |
| 1000 esadecimale | Uscita PWM(891) 0 |
| 1001 esadecimale | Uscita PWM(891) 1 |

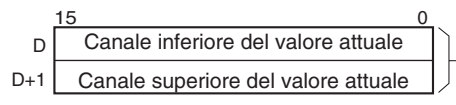
C: dati di controllo

La funzione dell'istruzione INI(880) è determinata dai dati di controllo C.

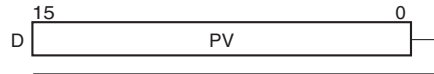
| C | Funzione di PRV(881) | Variazioni |
|------------------|--|--|
| 0000 esadecimale | Legge il valore attuale. | --- |
| 0001 esadecimale | Legge lo stato. | --- |
| 0002 esadecimale | Legge i risultati dei confronti a intervalli. | --- |
| 00□3 esadecimale | P = 0000 o 0001: legge la frequenza dell'uscita a treno di impulsi dell'uscita a impulsi 0 o 1 P = 0010: legge la frequenza dell'ingresso del contatore veloce 0. | C = 0003 esadecimale: funzionamento normale C = 0013 esadecimale: metodo con campionamento ogni 10 ms per alte frequenze (supportato solo dalle CPU CJ1M versione 3 o successive) C = 0023 esadecimale: metodo con campionamento ogni 100 ms per alte frequenze (supportato solo dalle CPU CJ1M versione 3 o successive) C = 0033 esadecimale: metodo con campionamento ogni singolo ms per alte frequenze (supportato solo dalle CPU CJ1M versione 3 o successive) |

D: primo canale di destinazione

Il valore attuale viene inviato a D o a D e D+1.



PV in due canali
 Valore attuale dell'uscita a impulsi, valore attuale dell'ingresso del contatore veloce, frequenza di ingresso del contatore veloce 0



PV in un canale
 Valore attuale dell'interrupt di ingresso in modalità contatore, stato, risultati del confronto a intervalli

Caratteristiche operando

| Area | P | C | D |
|--|--------------------------------------|--------------------------------------|--|
| Area CIO | --- | --- | Da CIO 0000 a CIO 6142 |
| Area di lavoro | --- | --- | Da W000 a W510 |
| Area bit di ritenività | --- | --- | Da H000 a H510 |
| Area bit ausiliaria | --- | --- | Da A448 ad A958 |
| Area del temporizzatore | --- | --- | Da T0000 a T4094 |
| Area del contatore | --- | --- | Da C0000 a C4094 |
| Area DM | --- | --- | Da D00000 a D32766 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- | Da @ D00000 a @ D32766 |
| Indirizzi DM/EM indiretti in BCD | --- | --- | Da *D00000 a *D32766 |
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. | --- |
| Registri dati | --- | --- | --- |
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047 ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

L'istruzione PRV(881) legge i dati specificati in C per la porta specificata in P. Nella seguente tabella sono riportate le possibili combinazioni di dati e porte.

| P: identificatore della porta | C: dati di controllo | | | |
|--|--|---------------------------------------|--|--|
| | 0000 esadecimale: lettura del valore attuale | 0001 esadecimale: lettura dello stato | 0002 esadecimale: lettura dei risultati dei confronti a intervalli | 0003 esadecimale: frequenza del contatore veloce o dell'uscita a impulsi |
| 0000 o 0001 esadecimale: uscita a treno di impulsi | OK | OK | Non consentita | OK (solo CPU CJ1M versione 3.0 o successive) |
| 0010 o 0011 esadecimale: ingresso contatore veloce | OK | OK | OK | OK (solo contatore veloce 0) |
| 0100, 0101, 0102 o 0103 esadecimale: ingresso ad interrupt in modalità contatore | OK | Non consentita | Non consentita | Non consentita |
| 1000 o 1001 esadecimale: uscita PWM(891) | Non consentita | OK | Non consentita | Non consentita |

■ **Letture di un valore attuale (C = 0000 esadecimale)**

Se C è 0000 esadecimale, l'istruzione PRV(881) legge un valore attuale come descritto nella seguente tabella.

| Porta e modalità | | Funzionamento | Intervallo di impostazione |
|---|--------------------|--|---|
| Uscita a treno di impulsi (P = 0000 o 0001 esadecimale) | | Il valore attuale dell'uscita a treno di impulsi viene memorizzato in D e D+1. | Da 8000 0000 a 7FFF FFFF esadecimale (-2.147.483.648 ... 2.147.483.647) |
| Ingresso contatore veloce (P = 0010 o 0011 esadecimale) | Modalità lineare | Il valore attuale del contatore veloce viene memorizzato in D e D+1. | Da 8000 0000 a 7FFF FFFF esadecimale (-2.147.483.648 ... 2.147.483.647) |
| | Modalità circolare | | 0000 0000 ... FFFF FFFF esadecimale (0 ... 4.294.967.295) |
| Ingressi ad interrupt in modalità contatore (P = 0100, 0101, 0102 o 0103 esadecimale) | | Il valore attuale dell'ingresso ad interrupt viene memorizzato in D. | Da 0000 a FFFF esadecimale (0 ... 65.535) |

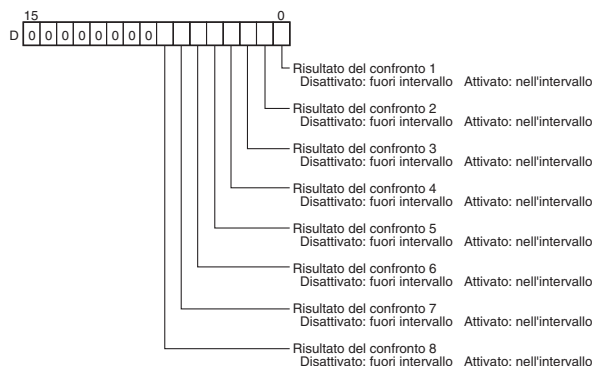
■ **Letture dello stato (C = 0001 esadecimale)**

Se C è 0001 esadecimale, l'istruzione PRV(881) legge lo stato come descritto nella seguente tabella.

| Porta e modalità | Funzionamento | Risultati della lettura |
|---------------------------|---|---|
| Uscita a impulsi | Lo stato dell'uscita a treno di impulsi viene memorizzato in D. | <ul style="list-style-type: none"> Flag di stato dell'uscita a treno di impulsi Disattivato: velocità costante Attivato: accelerazione/decelerazione Flag di overflow/underflow per il valore attuale Disattivato: normale Attivato: errore Flag di valore impostato per l'uscita a treno di impulsi Disattivato: non impostato Attivato: impostato Flag di uscita a treno di impulsi completata Disattivato: invio non completato Attivato: invio completato Flag uscita a treno di impulsi in corso Disattivato: interrotto ON: invio in corso Flag di nessuna origine Disattivato: origine determinata ON: origine non determinata Flag di posizione su origine Disattivato: non posizionato sull'origine ON: posizionato sull'origine Flag di errore di uscita a treno di impulsi interrotta Disattivato: nessun errore Attivato: emissione impulsi interrotta a causa di un errore |
| Ingresso contatore veloce | Lo stato del contatore veloce viene memorizzato in D. | <ul style="list-style-type: none"> Flag di confronto in corso Disattivato: interrotto Attivato: confronto Flag di overflow/underflow per il valore attuale Disattivato: normale Attivato: errore |
| Uscita PWM(891) | L'uscita di PWM(891) viene memorizzata in D. | <ul style="list-style-type: none"> Flag uscita a treno di impulsi in corso Disattivato: interrotto Attivato: invio in corso |

■ **Letture dei risultati del confronto a intervalli (C = 0002 esadecimale)**

Se C è 0002 esadecimale, l'istruzione PRV(881) legge i risultati del confronto a intervalli e li memorizza in D come descritto nella seguente tabella.



■ **Letture della frequenza dell'uscita a treno di impulsi o del contatore veloce (C = 00□3 esadecimale)**

Se C è pari a 00□3 esadecimale, tramite l'istruzione PRV(881) viene letta la frequenza emessa dall'uscita a treno di impulsi 0 o 1 oppure la frequenza emessa al contatore veloce 0 e tale valore viene memorizzato in D e D+1.

Intervalli di frequenza

| Valore di P | Risultato della conversione |
|--|--|
| 0000 o 0001 esadecimale (lettura della frequenza dell'uscita a treno di impulsi 0 o 1) | 0000 0000 ... 0001 86A0 esadecimale (0 ... 100.000) |
| 0010 esadecimale (lettura della frequenza del contatore veloce 0) | Metodo dell'ingresso del contatore: qualsiasi metodo di ingresso eccetto la modalità a fasi differenziali 4× Risultato = 00000000 ... 000186A0 esadecimale (0 ... 100.000) Nota Se viene immessa una frequenza superiore a 100 kHz, il valore dell'uscita rimarrà quello massimo, ossia 000186A0 esadecimale. |
| | Metodo dell'ingresso del contatore: 4modalità a fasi differenziali 4× Risultato = 00000000 ... 00030D40 esadecimale (0 ... 200.000) Nota Se viene immessa una frequenza superiore a 200 kHz, il valore dell'uscita rimarrà quello massimo, ossia 00030D40 esadecimale. |

Metodi di calcolo della frequenza di impulsi

Se si utilizza una CPU CJ1M versione 3.0 o successiva, sono disponibili due metodi per calcolare la frequenza di impulsi in uscita dell'uscita a treno di impulsi 0 o 1 o la frequenza di impulsi in ingresso del contatore veloce 0.

1. Metodo di calcolo standard (metodo precedente)

Il conteggio viene determinato contando ogni impulso, indipendentemente dalla frequenza. La corruzione dei fronti di salita o di discesa di alcuni impulsi ad alte frequenze provoca tuttavia errori (circa 1% massimo a 100 kHz).

2. Metodo di calcolo ad alta frequenza

In questo caso il metodo utilizzato cambia in base alla frequenza (alta o bassa).

• Conteggio ad alta frequenza

Ad alte frequenze (oltre 1 kHz) viene contato il numero di impulsi all'interno di un intervallo fisso (tempo di campionamento) e la frequenza è determinata in base a tale conteggio. È possibile selezionare uno dei tre seguenti tempi di campionamento impostando le due cifre all'estrema destra di C.

| Tempo di campionamento | Valore di C | Descrizione |
|------------------------|---------------------|---|
| 10 ms | 0013 esadecimale | Il numero di impulsi viene calcolato ogni 10 ms. L'errore massimo è il 10% a 1 kHz. |
| 100 ms | 0023 esadecimale | Il numero di impulsi viene calcolato ogni 100 ms. L'errore massimo è l'1% a 1 kHz. |
| 1 s | 0033 esadecimale | Il numero di impulsi viene calcolato ogni 1 s. L'errore massimo è lo 0,1% a 1 kHz. |

• Conteggio a bassa frequenza

A frequenze inferiori a 1 kHz viene utilizzato il metodo di calcolo standard, indipendentemente dall'impostazione del tempo di campionamento.

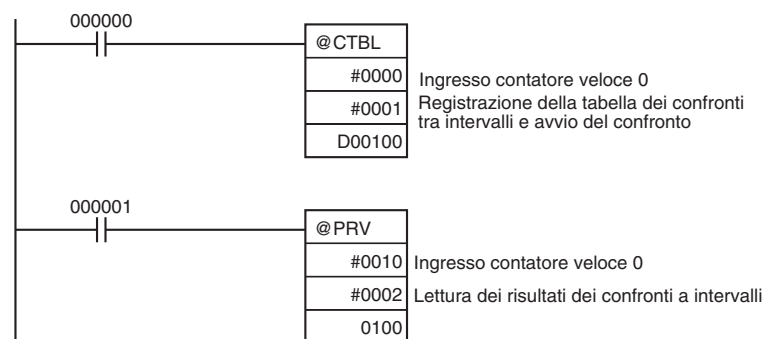
Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | <p>ON se viene superata la gamma specificata per P o C.</p> <p>ON se la combinazione di P e C non è consentita.</p> <p>ON se viene specificata la lettura dei risultati del confronto a intervalli quando il confronto a intervalli non è in esecuzione.</p> <p>ON se viene specificata la lettura della frequenza di uscita per un elemento diverso dal contatore veloce 0.</p> <p>ON se viene specificata la lettura per una porta non impostata come contatore veloce.</p> <p>ON se l'istruzione viene eseguita per una porta non impostata come ingresso a interrupt in modalità contatore.</p> |

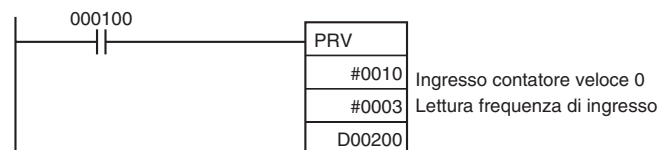
Esempi

■ **Esempio 1**

Nel seguente esempio di programmazione, quando CIO 000000 è ON, l'istruzione CTBL(882) registra una tabella di confronto a intervalli per il contatore veloce 0 e inizia il confronto. Quando CIO 000001 è ON, l'istruzione PRV(881) legge i risultati correnti del confronto a intervalli e li memorizza in CIO 0100.

■ **Esempio 2**

Nel seguente esempio di programmazione, quando CIO 000100 è ON, l'istruzione PRV(881) legge la frequenza corrente dell'impulso immesso nel contatore veloce 0 e la memorizza come valore esadecimale in D00200 e D00201.



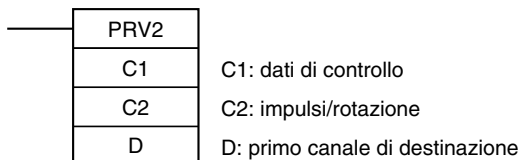
3-21-3 COUNTER FREQUENCY CONVERT: PRV2(883)

Scopo

PRV2(883) legge l'ingresso della frequenza di impulsi di un contatore veloce e converte la frequenza in velocità di rotazione oppure converte il valore attuale del contatore nel numero totale di rotazioni. Il risultato viene inviato ai canali di destinazione in formato esadecimale a 8 cifre. Gli impulsi possono essere immessi esclusivamente dal contatore veloce 0.

Questa istruzione è supportata solo dalle CPU CJ1M-CPU21/22/23 versione 2.0 o successive.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | PRV2(883) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PRV2(883) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Arete di programma applicabili

| Arete di programma a blocchi | Arete di programma step | Subroutine | Task ad interrupt |
|------------------------------|-------------------------|------------|-------------------|
| OK | OK | OK | OK |

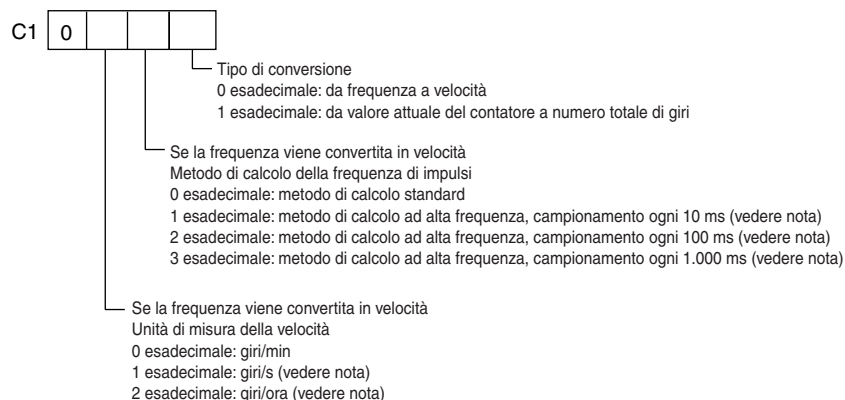
Operandi

C1: dati di controllo

La funzione dell'istruzione PRV2(883) è determinata dai dati di controllo C1.

| C1 | Funzione di PRV2(883) |
|---------------------------------|--|
| 0□*0 esadecimale (vedere nota). | Converte la frequenza in velocità di rotazione. |
| 0001 esadecimale | Converte il valore attuale del contatore nel numero totale di rotazioni. |

Nota La seconda cifra di C (□) indica l'unità di misura e la terza cifra (*) indica il metodo di calcolo della frequenza.

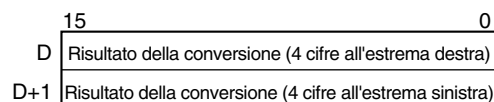


C2: impulsi/rotazione

Specifica il numero di impulsi per rotazione (da 0001 a FFFF esadecimale).

D: primo canale di destinazione

Il valore attuale viene inviato a D o a D e D+1.



Caratteristiche operando

| Area | C1 | C2 | D |
|-------------------------|-----|------------------------|------------------------|
| Area CIO | --- | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | --- | Da W000 a W511 | Da W000 a W510 |
| Area bit di ritentività | --- | Da H000 a H511 | Da H000 a H510 |
| Area bit ausiliaria | --- | Da A448 a A959 | Da A448 ad A958 |

| Area | C1 | C2 | D |
|--|--------------------------------------|---|------------------------|
| Area del temporizzatore | --- | Da T0000 a T4095 | Da T0000 a T4094 |
| Area del contatore | --- | Da C0000 a C4095 | Da C0000 a C4094 |
| Area DM | --- | Da D00000 a D32767 | Da D00000 a D32766 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ D32767 | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 | Da *D00000 a *D32767 |
| Costanti | Vedere la descrizione dell'operando. | --- | --- |
| Registri dati | --- | --- | --- |
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047 ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

PRV2(883) converte l'ingresso della frequenza a impulsi del contatore ad alta velocità 0 in base al metodo di conversione specificato in C1 e al coefficiente impulsi/rotazioni specificato in C2 e invia il risultato a D e D+1.

Selezionare uno dei metodi di conversione riportati di seguito impostando C1 su 0000 o 0001 esadecimale.

Conversione della frequenza in velocità di rotazione (C1 = 0□*0 esadecimale)

Se C1 è 0□*0 esadecimale, l'istruzione PRV2(883) calcola la velocità di rotazione (giri/min) in base ai dati della frequenza e all'impostazione di impulsi/rotazione. La seconda cifra di C (□) indica l'unità di misura e la terza cifra (*) indica il metodo di calcolo della frequenza.

1. Unità di misura della velocità di rotazione

- Unità di misura della velocità di rotazione = giri/min

Se la seconda cifra di C (□) è 0, l'istruzione PRV2(883) calcola la velocità di rotazione (giri/min) in base ai dati della frequenza e all'impostazione di impulsi/rotazione.

$$\text{Velocità di rotazione (giri/min)} = (\text{frequenza} \div \text{impulsi/rotazione}) \times 60$$

- Unità di misura della velocità di rotazione = giri/s (solo CPU CJM1 versione 3.0 o successive)

Se la seconda cifra di C (□) è 1, l'istruzione PRV2(883) calcola la velocità di rotazione (giri/s) in base ai dati della frequenza e all'impostazione di impulsi/rotazione.

$$\text{Velocità di rotazione (giri/s)} = (\text{frequenza} \div \text{impulsi/rotazione})$$

- Unità di misura della velocità di rotazione = giri/ora (solo CPU CJM1 versione 3.0 o successive)

Se la seconda cifra di C (□) è 2, l'istruzione PRV2(883) calcola la velocità di rotazione (giri/ora) in base ai dati della frequenza e all'impostazione di impulsi/rotazione.

$$\text{Velocità di rotazione (giri/ora)} = (\text{frequenza} \div \text{impulsi/rotazione}) \times 60 \times 60$$

- Intervalli dei risultati di conversione

- Metodo dell'ingresso del contatore: qualsiasi metodo eccetto la modalità a fasi differenziali 4x
Risultato della conversione = 00000000 ... 000186A0 esadecimale (0 ... 100.000)

Se viene immessa una frequenza superiore a 100 kHz, il valore dell'uscita rimarrà quello massimo, ossia 000186A0 esadecimale.

- Metodo dell'ingresso del contatore: 4modalità a fasi differenziali 4×
Risultato della conversione = 00000000 ... 00030D40 esadecimale (0 ... 200.000)

Se viene immessa una frequenza superiore a 200 kHz, il valore dell'uscita rimarrà quello massimo, ossia 00030D40 esadecimale.

2. Metodo di calcolo della frequenza

Se si utilizza una CPU CJ1M versione 3.0 o successiva, sono disponibili due metodi per calcolare la frequenza di impulsi in ingresso del contatore veloce 0.

a) Metodo di calcolo standard (C1 = 0□□00)

Il conteggio viene determinato contando ogni impulso, indipendentemente dalla frequenza. La corruzione dei fronti di salita o di discesa di alcuni impulsi ad alte frequenze provoca tuttavia errori (circa 1% massimo a 100 kHz).

b) Metodo di calcolo ad alta frequenza

In questo caso il metodo utilizzato cambia in base alla frequenza (alta o bassa) (solo CPU CJ1M versione 3.0 o successive).

- Conteggio ad alta frequenza (C1 = 0□□10, 0□□20, or 0□□30)

Ad alte frequenze (oltre 1 kHz) viene contato il numero di impulsi all'interno di un intervallo fisso (tempo di campionamento) e la frequenza è determinata in base a tale conteggio. È possibile selezionare uno dei tre seguenti tempi di campionamento impostando la terza cifra di C1.

| Tempo di campionamento | Valore di C1 | Descrizione |
|------------------------|-------------------|---|
| 10 ms | 0□□10 esadecimale | Il numero di impulsi viene calcolato ogni 10 ms. L'errore massimo è il 10% a 1 kHz. |
| 100 ms | 0□□20 esadecimale | Il numero di impulsi viene calcolato ogni 100 ms. L'errore massimo è l'1% a 1 kHz. |
| 1 s | 0□□30 esadecimale | Il numero di impulsi viene calcolato ogni 1 s. L'errore massimo è lo 0,1% a 1 kHz. |

- Conteggio a bassa frequenza

A frequenze inferiori a 1 kHz viene utilizzato il metodo di calcolo standard, indipendentemente dall'impostazione del tempo di campionamento.

Conversione del valore attuale del contatore nel numero totale di rotazioni (C1 = 0001 esadecimale)

Se C1 è 0001 esadecimale, l'istruzione PRV2(883) calcola il numero cumulativo di rotazioni in base al valore attuale del contatore e all'impostazione di impulsi/rotazione.

Risultato della conversione = valore attuale contatore ÷ impulsi/rotazione

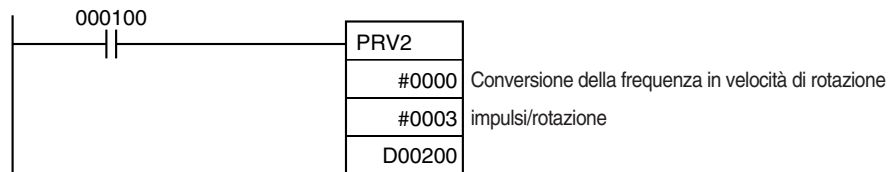
Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se il contatore veloce 0 è disabilitato nelle impostazioni. ON se il valore di C1 non è compreso nell'intervallo specificato (0000 o 0001). ON se l'impostazione di impulsi/rotazione in C2 è 0000. |

Esempi

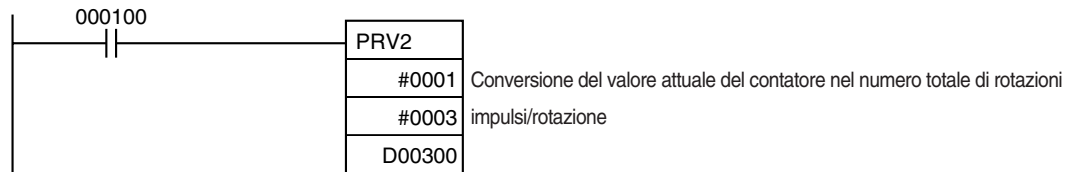
■ Esempio 1

Nel seguente esempio di programmazione, quando CIO 000100 è ON, l'istruzione PRV2(883) legge la frequenza corrente degli impulsi del contatore veloce 0, converte tale valore in velocità di rotazione (giri/min) e invia il risultato in formato esadecimale a D00201 e D00200.



Esempio 2

Nel seguente esempio di programmazione, quando CIO 000100 è ON, l'istruzione PRV2(883) legge il valore attuale del contatore, converte tale valore nel numero di rotazioni e invia il risultato in formato esadecimale a D00301 e D00300.



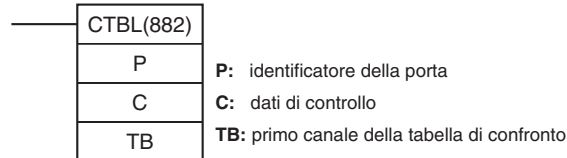
3-21-4 REGISTER COMPARISON TABLE: CTBL(882) (solo CJ1M-CPU21/22/23)

Scopo

È possibile utilizzare l'istruzione CTBL(882) per registrare una tabella di confronto e confrontare il valore attuale di un contatore veloce con un valore di riferimento o un intervallo. Quando la condizione specificata è soddisfatta, viene eseguito un task ad interrupt.

Questa istruzione è supportata solo dalle CPU CJ1M-CPU21/22/23.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CTBL(882) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CTBL(882) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

P: identificatore della porta

P specifica la porta di cui devono essere contati gli impulsi, come indicato nella seguente tabella.

| | |
|------------------|--------------------|
| P | Porta |
| 0000 esadecimale | Contatore veloce 0 |
| 0001 esadecimale | Contatore veloce 1 |

C: dati di controllo

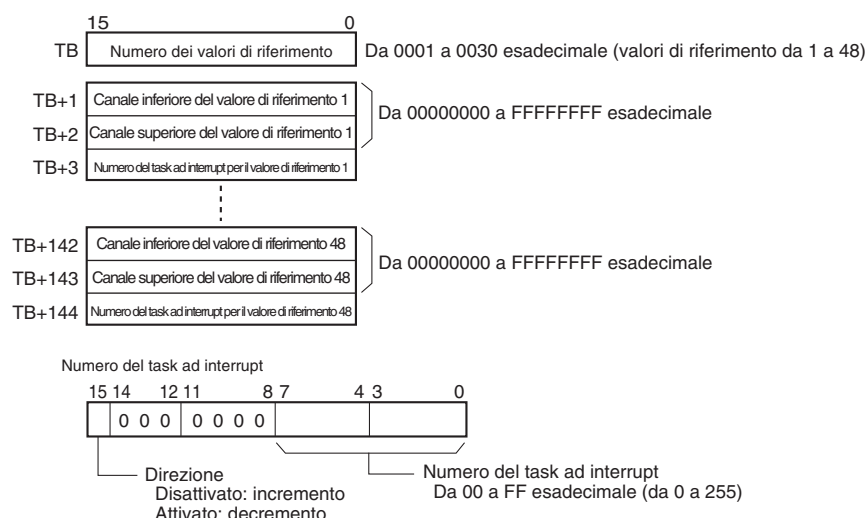
La funzione dell'istruzione CTBL(882) è determinata dai dati di controllo C, come descritto nella seguente tabella.

| C | Funzione di CTBL(882) |
|---------------------|--|
| 0000 esadecimale | Registra una tabella di confronto contenente valori di riferimento e inizia il confronto. |
| 0001 esadecimale | Registra una tabella di confronto contenente intervalli ed esegue un confronto. |
| 0002 esadecimale | Registra una tabella di confronto contenente valori di riferimento. Il confronto verrà iniziato tramite l'istruzione INI(880). |
| 0003 esadecimale | Registra una tabella di confronto contenente intervalli. Il confronto verrà iniziato tramite l'istruzione INI(880). |

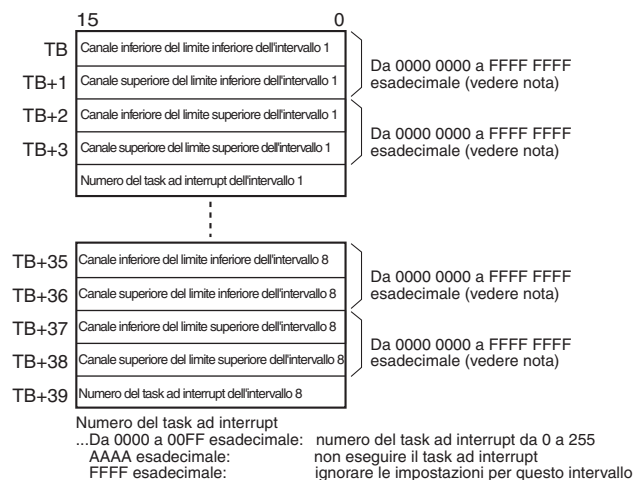
TB: primo canale della tabella di confronto

TB è il primo canale della tabella di confronto. La struttura della tabella di confronto dipende dal tipo di confronto eseguito.

Per il confronto con valori di riferimento, la lunghezza della tabella di confronto è determinata dal numero di valori di riferimento specificati in TB. La lunghezza della tabella può variare da 4 a 145 canali, come illustrato di seguito.



Per il confronto a intervalli, la tabella di confronto contiene sempre otto intervalli. La lunghezza della tabella è di 40 canali, come illustrato di seguito. Se non è necessario definire otto intervalli, impostare il numero di task ad interrupt su FFFF esadecimale per tutti gli intervalli non utilizzati.



Nota Per qualsiasi intervallo, impostare sempre un limite superiore che sia maggiore o uguale al limite inferiore.

Caratteristiche operando

| Area | P | C | TB |
|--|--------------------------------------|--------------------------------------|--|
| Area CIO | --- | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | --- | Da W000 a W511 |
| Area bit di ritenività | --- | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | --- | Da A448 ad A959 |
| Area del temporizzatore | --- | --- | Da T0000 a T4095 |
| Area del contatore | --- | --- | Da C0000 a C4095 |
| Area DM | --- | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | --- | Da *D00000 a *D32767 |
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. | --- |
| Registri dati | --- | --- | --- |
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047 ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

CTBL(882) registra una tabella di confronto e inizia il confronto per la porta specificata in P utilizzando il metodo specificato in C. Una volta registrata, la tabella di confronto resta valida finché non viene registrata una nuova tabella o attivata la modalità PROGRAM per la CPU.

Ad ogni esecuzione dell'istruzione CTBL(882), il confronto viene avviato in base alle condizioni specificate. Quando si utilizza CTBL(882) per iniziare un confronto, in genere è sufficiente utilizzare la versione differenziata dell'istruzione [@CTBL(882)] o una condizione di esecuzione che viene impostata su ON per una sola scansione.

Nota Se si specifica un task ad interrupt non registrato, si verificherà un errore di programma fatale alla prima generazione dell'interrupt.

■ **Registrazione di una tabella di confronto (C = 0002 o 0003 esadecimale)**

Se C viene impostato su 0002 o 0003 esadecimale, verrà registrata una tabella di confronto, ma il confronto non verrà avviato. Il confronto viene avviato tramite l'istruzione INI(880).

■ **Registrazione di una tabella di confronto e avvio del confronto (C = 0000 o 0001 esadecimale)**

Se C viene impostato su 0000 o 0001 esadecimale, verrà registrata una tabella di confronto e il confronto verrà iniziato.

■ **Interruzione del confronto**

Il confronto viene sempre interrotto con l'istruzione INI(880), indipendentemente dall'istruzione utilizzata per iniziarlo.

■ Confronto con il valore di riferimento

Il task ad interrupt corrispondente è richiamato ed eseguito quando il valore attuale coincide con uno dei valori di riferimento.

- Lo stesso numero di task ad interrupt può essere specificato per più valori di riferimento.
- Quando il valore attuale viene incrementato o decrementato, è possibile impostare la direzione per specificare se il valore di riferimento è valido. Se il bit 15 del canale utilizzato per specificare il numero di task ad interrupt per l'intervallo è OFF, il valore attuale viene confrontato con il valore di riferimento solo quando subisce incrementi, mentre se il bit 00 è ON, verrà confrontato solo quando subisce decrementi.
- La tabella di confronto contiene il numero di valori di riferimento specificato in TB, fino a un massimo di 48; la lunghezza della tabella dipende quindi dal numero di valori di riferimento specificato.
- I confronti vengono eseguiti per tutti i valori di riferimento registrati nella tabella.

- Nota**
1. Se lo stesso valore di riferimento con identica direzione del confronto viene registrato più di una volta in una tabella, si verificherà un errore.
 2. Se il contatore veloce viene impostato per la modalità a impulsi incrementale e nella tabella si imposta il decremento come direzione del confronto, si verificherà un errore.
 3. Se la direzione di conteggio viene modificata quando il valore attuale è uguale a un valore di riferimento raggiunto nella direzione opposta a quella impostata come direzione del confronto, la condizione del confronto per il valore di riferimento non verrà soddisfatta. Non impostare come riferimento i valori massimo e minimo del valore di conteggio.

Confronto a intervalli

Il task ad interrupt corrispondente è richiamato ed eseguito quando il valore attuale è all'interno di un intervallo impostato.

- Lo stesso numero di task ad interrupt può essere specificato per più valori di riferimento.
- Nella tabella di confronto a intervalli sono contenuti 8 intervalli, ciascuno dei quali è definito da un limite inferiore e un limite superiore. Se un intervallo non viene utilizzato, impostare il numero di task ad interrupt su FFFF esadecimale per disabilitarlo.
- Il task ad interrupt viene eseguito una sola volta quando il valore attuale è compreso nell'intervallo.
- Se il valore attuale è compreso in più di un intervallo al momento del confronto, verrà attribuita la priorità al task ad interrupt dell'intervallo più prossimo all'inizio della tabella, mentre gli altri task ad interrupt verranno eseguiti nei cicli seguenti.
- Se non è richiesta l'esecuzione di un task ad interrupt, specificare AAAA esadecimale come numero del task ad interrupt. È possibile leggere il risultato del confronto a intervalli utilizzando l'istruzione PRV(881) o i flag di confronto a intervalli in corso.

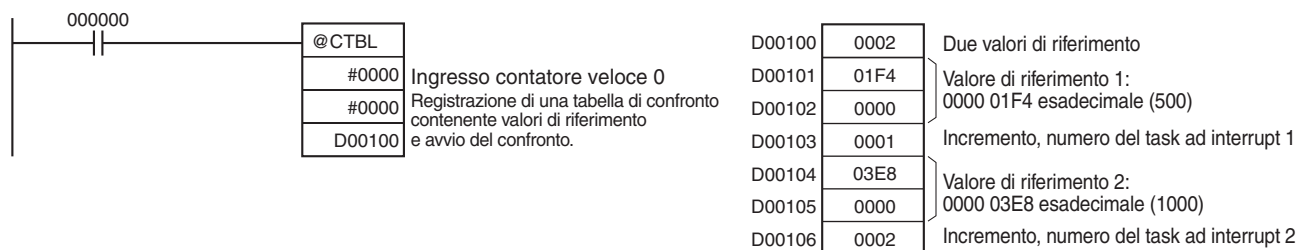
- Nota** Se in uno degli intervalli il limite superiore risulta minore del limite inferiore, si verificherà un errore.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | <p>ON se viene superata la gamma specificata per P o C.</p> <p>ON se il numero di valori di riferimento specificati per il confronto con valori di riferimento è impostato su 0.</p> <p>ON se il numero di valori di riferimento specificati per il confronto con valori di riferimento è superiore a 48.</p> <p>ON se viene specificato più di una volta lo stesso valore di riferimento nella medesima direzione del confronto per il confronto con valori di riferimento.</p> <p>ON se in un intervallo il limite superiore risulta minore del limite inferiore.</p> <p>ON se vengono disabilitati i valori impostati per tutti gli intervalli durante un confronto a intervalli.</p> <p>ON se il contatore veloce viene impostato per la modalità a impulsi incrementale e nella tabella si imposta il decremento come direzione del confronto.</p> <p>ON se viene eseguita un'istruzione quando il contatore veloce è impostato in modalità circolare e il valore specificato supera il valore massimo del ciclo.</p> <p>ON se viene specificata la lettura per una porta non impostata come contatore veloce.</p> <p>ON se l'istruzione viene eseguita per un metodo di confronto diverso mentre è già in corso un confronto.</p> |

Esempio

Nel seguente esempio di programmazione, quando CIO 000000 è ON, l'istruzione CTBL(882) registra una tabella di confronto con valori di riferimento e avvia il confronto per il contatore veloce 0. Il valore attuale del contatore veloce viene incrementato e quando raggiunge 500, eguagliando il valore di riferimento 1, viene eseguito il task ad interrupt 1. Quando il valore attuale raggiunge 1000, eguagliando il valore di riferimento 2, viene eseguito il task ad interrupt 2.



3-21-5 SPEED OUTPUT: SPED(885) (solo CJ1M-CPU21/22/23)

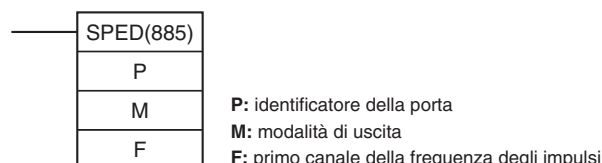
Scopo

L'istruzione SPED(885) viene utilizzata per impostare la frequenza dell'uscita a treno di impulsi per una porta specifica e avviare l'uscita a treno di impulsi senza accelerazione o decelerazione. Sono possibili il posizionamento in modalità indipendente o il controllo della velocità in modalità continua. Per il posizionamento in modalità indipendente il numero di impulsi viene impostato tramite l'istruzione PULS(886).

È possibile eseguire SPED(885) anche durante l'uscita a treno di impulsi per modificare la frequenza di uscita, creando modifiche in fasi della velocità.

Questa istruzione è supportata solo dalle CPU CJ1M-CPU21/22/23.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SPED(885) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SPED(885) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

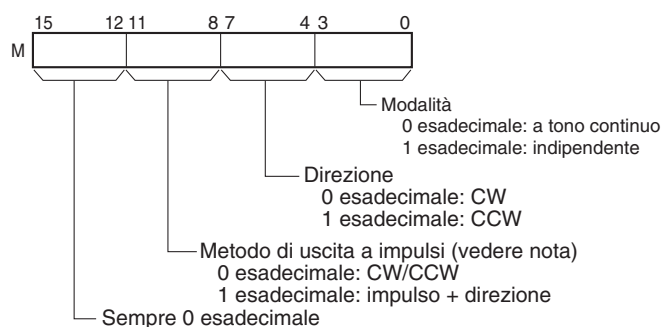
P: identificatore della porta

L'identificatore della porta specifica la porta da dove vengono emessi gli impulsi.

| P | Porta |
|------------------|-----------------------------|
| 0000 esadecimale | Uscita a treno di impulsi 0 |
| 0001 esadecimale | Uscita a treno di impulsi 1 |

M: modalità di uscita

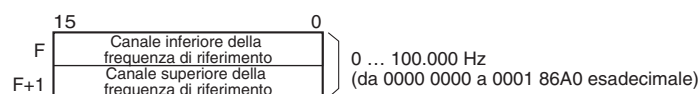
Il valore di M determina la modalità di uscita.



Nota: utilizzare lo stesso metodo di uscita a impulsi per entrambe le uscite a impulsi 0 e 1.

F: primo canale della frequenza degli impulsi

Il valore di F e F+1 definisce la frequenza degli impulsi in Hz.



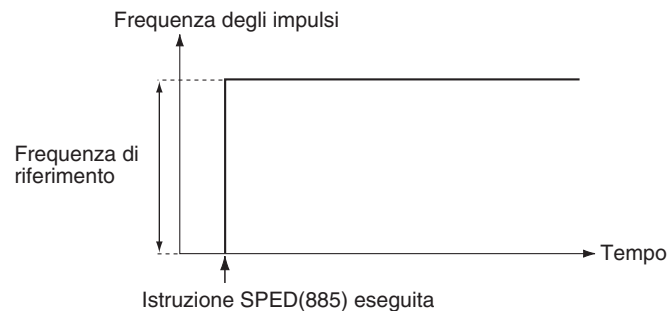
Caratteristiche operando

| Area | P | M | F |
|--------------------------------------|-----|-----|------------------------|
| Area CIO | --- | --- | Da CIO 0000 a CIO 6142 |
| Area di lavoro | --- | --- | Da W000 a W510 |
| Area bit di ritentività | --- | --- | Da H000 a H510 |
| Area bit ausiliaria | --- | --- | Da A448 ad A958 |
| Area del temporizzatore | --- | --- | Da T0000 a T4094 |
| Area del contatore | --- | --- | Da C0000 a C4094 |
| Area DM | --- | --- | Da D00000 a D32766 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | --- | Da *D00000 a *D32767 |

| Area | P | M | F |
|--|--------------------------------------|--------------------------------------|--|
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. |
| Registri dati | --- | --- | --- |
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

SPED(885) avvia l'uscita a treno di impulsi sulla porta specificata in P, utilizzando il metodo specificato in M, alla frequenza specificata in F. L'uscita a treno di impulsi verrà attivata a ogni esecuzione di SPED(885). In genere è sufficiente utilizzare la versione differenziata dell'istruzione [`@SPED(885)`] o una condizione di esecuzione che viene impostata su ON per una sola scansione.



In modalità indipendente l'uscita a treno di impulsi si interrompe in modo automatico quando è stato emesso il numero di impulsi impostato in precedenza con PULS(886). In modalità continua l'uscita a treno di impulsi prosegue finché non viene interrotta dal programma.

Se la modalità cambia da indipendente a continua o viceversa durante l'uscita a treno di impulsi, si verificherà un errore.

■ Controllo della velocità in modalità continua

Quando si avvia il funzionamento in modalità continua, l'uscita a treno di impulsi continua finché non viene interrotta dal programma.

Nota L'uscita a treno di impulsi si interromperà immediatamente se viene attivata la modalità PROGRAM della CPU.

| Funzionamento | Scopo | Applicazione | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|---|--|--|---|---|---|
| Avvio dell'uscita a treno di impulsi | Avviare l'uscita alla velocità specificata | Modifica della velocità (frequenza) in una sola fase | <p>Frequenza degli impulsi</p> <p>Esecuzione di SPED(885)</p> | Emette gli impulsi a una frequenza specificata. | SPED(885) (continua) |
| Modifica delle impostazioni | Modificare la velocità in una sola fase | Modifica della velocità durante il funzionamento | <p>Frequenza degli impulsi</p> <p>Esecuzione di SPED(885)</p> | Modifica la frequenza dell'uscita a treno di impulsi (più alta o più bassa) in una sola fase. | SPED(885) (continua) ↓ SPED(885) (continua) |
| Interruzione dell'uscita a treno di impulsi | Interrompere l'uscita a treno di impulsi | Interruzione immediata | <p>Frequenza degli impulsi</p> <p>Esecuzione di INI(880)</p> | Interrompe immediatamente l'emissione degli impulsi. | SPED(885) (continua) ↓ INI(880) |
| | Interrompere l'uscita a treno di impulsi | Interruzione immediata | <p>Frequenza degli impulsi</p> <p>Esecuzione di SPED(885)</p> | Interrompe immediatamente l'emissione degli impulsi. | SPED(885) (continua) ↓ SPED(885) (continua, frequenza di riferimento di 0 Hz) |

■ Posizionamento in modalità indipendente

Quando si avvia il funzionamento in modalità indipendente, l'uscita a treno di impulsi continua finché non è stato emesso il numero di impulsi specificato.

- Nota**
1. L'uscita a treno di impulsi si interromperà immediatamente se viene attivata la modalità PROGRAM della CPU.
 2. Il numero di impulsi deve essere impostato ogni volta che viene riavviata l'uscita.
 3. Il numero di impulsi in uscita deve essere impostato con PULS(881) prima di eseguire SPED(885), altrimenti non verrà emesso alcun impulso.

4. Se il numero di impulsi impostato con PULS(881) è un valore assoluto, la direzione impostata nell'operando di SPED(885) verrà ignorata.

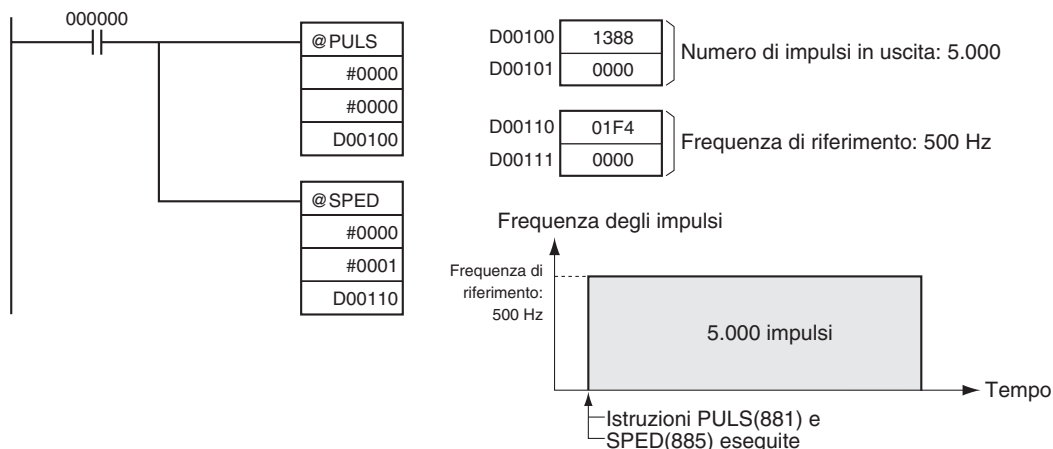
| Funzionamento | Scopo | Applicazione | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|---|--|---|---|---|--|
| Avvio dell'uscita a treno di impulsi | Avviare l'uscita alla velocità specificata | Posizionamento senza accelerazione o decelerazione | <p>Frequenza degli impulsi</p> <p>Numero di impulsi specificato tramite PULS(886)</p> <p>Frequenza di riferimento</p> <p>Tempo</p> <p>Esecuzione di SPED(885)</p> <p>Emette il numero di impulsi specificato quindi interrompe l'emissione.</p> | Avvia l'emissione degli impulsi alla frequenza specificata, interrompendola immediatamente quando è stato emesso il numero di impulsi specificato. | PULS (886) ↓ SPED(885) (indipendente) |
| Modifica delle impostazioni | Modificare la velocità in una sola fase | Modifica della velocità in una sola fase durante il funzionamento | <p>Frequenza degli impulsi</p> <p>Numero di impulsi specificato tramite PULS(886)</p> <p>Nuova frequenza di riferimento</p> <p>Frequenza di riferimento originaria</p> <p>Tempo</p> <p>Esecuzione di SPED(885) (modalità indipendente)</p> <p>Nuova esecuzione di SPED(885) (modalità indipendente) per modificare la frequenza di riferimento. La posizione di riferimento rimane invariata.</p> <p>Il numero di impulsi specificato tramite PULS(886) rimane invariato.</p> | Durante il posizionamento è possibile eseguire l'istruzione SPED(885) per aumentare o diminuire la frequenza dell'uscita a treno di impulsi in un'unica fase. La posizione di riferimento (numero di impulsi specificato) non viene modificata. | PULS (886) ↓ SPED(885) (indipendente) ↓ SPED(885) (indipendente) |
| Interruzione dell'uscita a treno di impulsi | Interrompere l'uscita a treno di impulsi (l'impostazione del numero di impulsi non viene conservata) | Interruzione immediata | <p>Frequenza degli impulsi</p> <p>Frequenza attuale</p> <p>Tempo</p> <p>Esecuzione di SPED(885)</p> <p>Esecuzione di INI(880)</p> | Interrompe immediatamente l'emissione degli impulsi e cancella l'impostazione relativa al numero di impulsi in uscita. | PULS (886) ↓ SPED(885) (indipendente) ↓ INI(880) ↓ PLS2(887) ↓ INI(880) |
| Interruzione dell'uscita a treno di impulsi (l'impostazione del numero di impulsi non viene conservata) | Interruzione immediata | Interruzione immediata | <p>Frequenza degli impulsi</p> <p>Frequenza attuale</p> <p>Tempo</p> <p>Esecuzione di SPED(885)</p> <p>Esecuzione di SPED(885)</p> | Interrompe immediatamente l'emissione degli impulsi e cancella l'impostazione relativa al numero di impulsi in uscita. | PULS (886) ↓ SPED(885) (indipendente) ↓ SPED(885) (indipendente, frequenza di riferimento di 0 Hz) |

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se viene superata l'intervallo specificato per P, M o F.</p> <p>ON se PLS2(887) o ORG(889) sono già in esecuzione per il controllo dell'uscita a treno di impulsi sulla porta specificata.</p> <p>ON se si utilizza SPED(885) o INI(880) per cambiare la modalità tra uscita continua e indipendente durante l'uscita a treno di impulsi.</p> <p>ON se si esegue SPED(885) in un task ad interrupt quando è in esecuzione un'istruzione che controlla l'uscita a treno di impulsi in un task ciclico.</p> <p>ON se si esegue SPEC(885) in modalità indipendente con un numero di impulsi assoluto senza che sia stata stabilita l'origine.</p> |

Esempio

Nel seguente esempio di programmazione, quando CIO 000000 viene impostato su ON, l'istruzione PULS(886) imposta il numero di impulsi per l'uscita a treno di impulsi 0 (viene definito un valore assoluto di 5.000 impulsi). SPED(885) viene eseguita dopo l'avvio dell'uscita a treno di impulsi, utilizzando il metodo CW/CCW in senso orario in modalità indipendente a una frequenza di riferimento di 500 Hz.



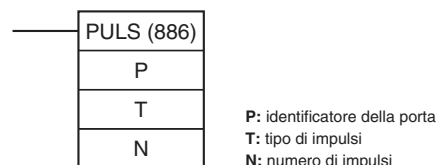
3-21-6 SET PULSES: PULS(886) (solo CJ1M-CPU21/22/23)

Scopo

È possibile utilizzare PULS(886) per impostare il valore (numero di impulsi) per le uscite a treno di impulsi che verranno avviate in seguito dal programma tramite l'istruzione SPED(885) o ACC(888) in modalità indipendente.

Questa istruzione è supportata solo dalle CPU CJ1M-CPU21/22/23.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | PULS (886) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PULS(886) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

P: identificatore della porta

L'identificatore della porta indica la porta. I parametri impostati in D ed N verranno applicati alla successiva istruzione SPED(885) o ACC(888) in cui è specificata tale porta come uscita.

| P | Porta |
|------------------|-----------------------------|
| 0000 esadecimale | Uscita a treno di impulsi 0 |
| 0001 esadecimale | Uscita a treno di impulsi 1 |

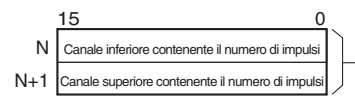
T: tipo di impulsi

T specifica il tipo di impulsi inviati nel modo seguente:

| T | Tipo di impulsi |
|------------------|-----------------|
| 0000 esadecimale | Relativi |
| 0001 esadecimale | Assoluti |

N ed N+1: numero di impulsi

N ed N+1 indicano il numero di impulsi per l'uscita a treno di impulsi relativi o la posizione di riferimento assoluta per gli impulsi assoluti in formato esadecimale a 8 cifre.



Uscita a impulsi relativi:
da 0 a 2.147.483.647 (da 0000 0000 a 7FFF FFFF esadecimale)

Uscita a impulsi assoluti:
-2.147.483.648 ... 2.147.483.647 (da 8000 0000 a 7FFF FFFF esadecimale)

Il numero effettivo di impulsi di movimento inviati viene determinato come indicato qui di seguito.

Per l'uscita a treno di impulsi relativi, numero di impulsi di movimento = numero impostato di impulsi. Per l'uscita a treno di impulsi assoluti, numero di impulsi di movimento = numero impostato di impulsi – valore attuale.

Caratteristiche operando

| Area | P | T | N |
|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| Area CIO | --- | --- | Da CIO 0000 a CIO 6142 |
| Area di lavoro | --- | --- | Da W000 a W510 |
| Area bit di ritenività | --- | --- | Da H000 a H510 |
| Area bit ausiliaria | --- | --- | Da A448 ad A958 |
| Area del temporizzatore | --- | --- | Da T0000 a T4094 |
| Area del contatore | --- | --- | Da C0000 a C4094 |
| Area DM | --- | --- | Da D00000 a D32766 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | --- | Da *D00000 a *D32767 |
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. |
| Registri dati | --- | --- | --- |

| Area | P | T | N |
|--|-----|-----|--|
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

È possibile utilizzare PULS(886) per impostare tipo e numero degli impulsi specificati in T e N per la porta indicata in P. L'uscita effettiva degli impulsi viene avviata in seguito dal programma tramite l'istruzione SPED(885) o ACC(888) in modalità indipendente.

Flag

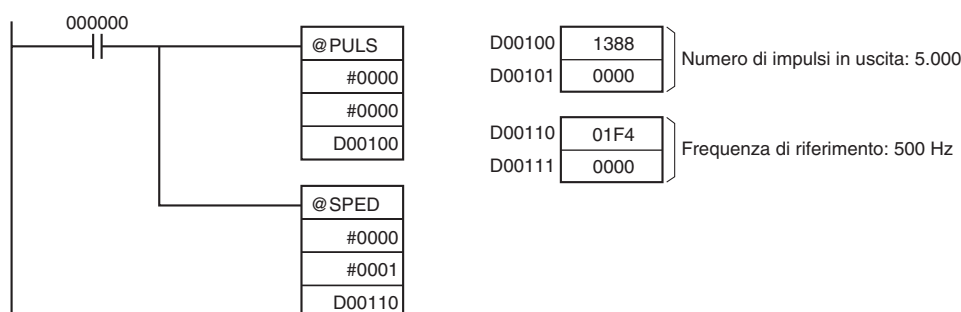
| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se viene superato l'intervallo specificato per P, T o N. ON se si esegue PULS(886) per una porta da cui è già in corso un'uscita a treno di impulsi. ON se si esegue PULS(886) in un task ad interrupt quando è in esecuzione un'istruzione che controlla l'uscita a treno di impulsi in un task ciclico. |

Avvertenze

- Se l'istruzione PULS(886) viene eseguita quando gli impulsi sono già in uscita, si verificherà un errore. Per prevenire questo errore, utilizzare la versione differenziata dell'istruzione [@PULS(886)] o una condizione di esecuzione impostata su ON per una sola scansione.
- Il numero di impulsi in uscita calcolato per PULS(886) non verrà modificato nemmeno se si utilizza INI(880) per modificare il valore attuale dell'uscita a treno di impulsi.
- Se il numero di impulsi definito con PULS(881) è un valore assoluto, la direzione impostata per SPED(885) o ACC(888) verrà ignorata.
- È possibile spostarsi all'esterno dell'intervallo del valore attuale del numero di impulsi per l'uscita a treno di impulsi (da -2.147.483.648 a 2.147.483.647).

Esempio

Nel seguente esempio di programmazione, quando CIO 000000 è ON, l'istruzione PULS(886) imposta il numero di impulsi per l'uscita a treno di impulsi 0 (viene definito un valore assoluto di 5.000 impulsi). SPED(885) viene eseguita dopo l'avvio dell'uscita a treno di impulsi, utilizzando il metodo CW/CCW in senso orario in modalità indipendente a una frequenza di riferimento di 500 Hz.



3-21-7 PULSE OUTPUT: PLS2(887) (solo CPU CJ1M 21/22/23)

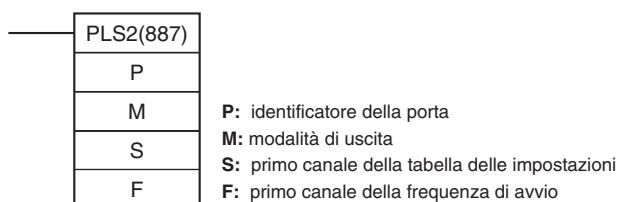
Scopo

PLS2(887) emette un numero specificato di impulsi sulla porta specificata. L'uscita a impulsi viene avviata a una frequenza di avvio specificata, accelera fino alla frequenza di riferimento in base a un valore di accelerazione specificato, decelera in base a un valore di decelerazione specificato e si interrompe approssimativamente alla stessa frequenza di quella di avvio. È supportata solo la modalità di posizionamento indipendente.

È possibile eseguire l'istruzione PLS2(887) anche durante l'uscita a impulsi per modificare il numero di impulsi, la frequenza di riferimento, il valore di accelerazione e il valore di decelerazione. L'istruzione PLS2(887) può quindi essere utilizzata per cambi progressivi di velocità con diversi valori di accelerazione e decelerazione e per modificare posizioni di riferimento, frequenza, velocità o direzione.

Questa istruzione è supportata solo dalle CPU CJ1M 21/22/23.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | PLS2(887) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PLS2 (887) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

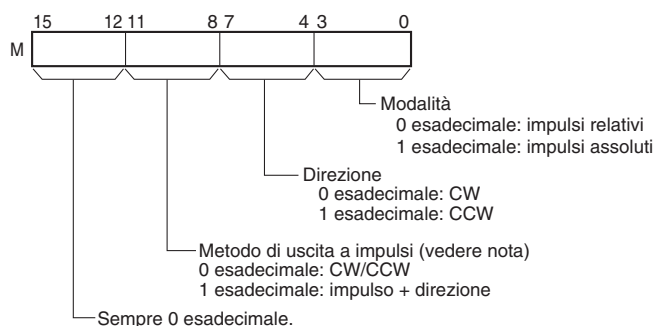
P: identificatore della porta

L'identificatore della porta indica la porta.

| P | Porta |
|------------------|--------------------|
| 0000 esadecimale | Uscita a impulsi 0 |
| 0001 esadecimale | Uscita a impulsi 1 |

M: modalità di uscita

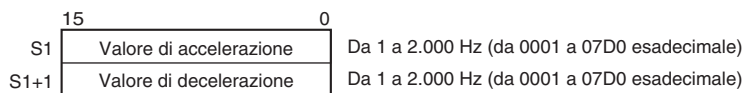
Il contenuto di M definisce i parametri per l'uscita a impulsi come descritto di seguito:



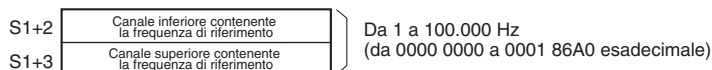
Nota: utilizzare lo stesso metodo di uscita a impulsi per entrambe le uscite a impulsi 0 e 1.

S: primo canale della tabella delle impostazioni

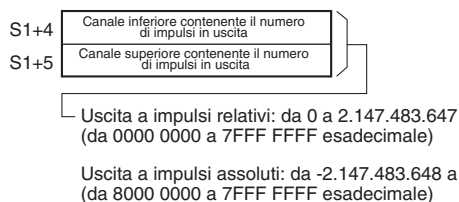
Il contenuto da S a S+5 controlla l'uscita a impulsi come illustrato nei seguenti schemi.



Specificare l'incremento o il decremento della frequenza a ogni periodo di controllo dell'impulso (4 ms).



Specificare l'accelerazione dopo la frequenza in Hz.

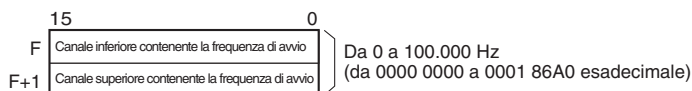


Il numero effettivo di impulsi di movimento inviati viene determinato come segue:

Per l'uscita a impulsi relativi, numero di impulsi di movimento = numero impostato di impulsi. Per l'uscita a impulsi assoluti, numero di impulsi di movimento = numero impostato di impulsi – valore attuale.

F: primo canale della frequenza di avvio

La frequenza di avvio viene specificata in F e F+1.



Specificare la frequenza di avvio in Hz.

Caratteristiche operando

| Area | P | M | S | F |
|--------------------------------------|--------------------------------------|--------------------------------------|------------------------|--------------------------------------|
| Area CIO | --- | --- | Da CIO 0000 a CIO 6138 | Da CIO 0000 a CIO 6142 |
| Area di lavoro | --- | --- | Da W000 a W506 | Da W000 a W510 |
| Area bit di ritentività | --- | --- | Da H000 a H506 | Da H000 a H510 |
| Area bit ausiliaria | --- | --- | Da A448 a A954 | Da A448 ad A958 |
| Area del temporizzatore | --- | --- | Da T0000 a T4090 | Da T0000 a T4094 |
| Area del contatore | --- | --- | Da C0000 a C4090 | Da C0000 a C4094 |
| Area DM | --- | --- | Da D00000 a D32762 | Da D00000 a D32766 |
| Area EM senza banco | --- | --- | --- | --- |
| Area EM con banco | --- | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- | Da @ D00000 a @ D32767 | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | --- | Da *D00000 a *D32767 | Da *D00000 a *D32767 |
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. | --- | Vedere la descrizione dell'operando. |
| Registri dati | --- | --- | --- | --- |

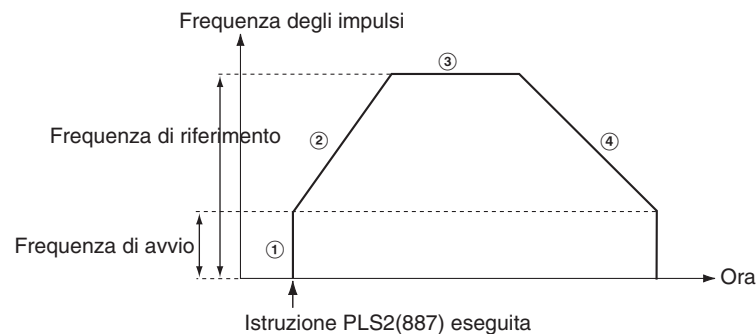
| Area | P | M | S | F |
|--|-----|-----|---|---|
| Registri indice | --- | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

PLS2(887) avvia l'uscita a impulsi sulla porta specificata in P, utilizzando la modalità specificata in M, alla frequenza di avvio specificata in F (1 nello schema). La frequenza viene aumentata a ogni periodo di controllo dell'impulso (4 ms) in base al valore di accelerazione specificato in S fino a raggiungere la frequenza di riferimento specificata in S (2 nello schema). Una volta raggiunta la frequenza di riferimento, l'accelerazione si arresta e l'uscita a impulsi continua a velocità costante (3 nello schema).

Il punto di decelerazione viene calcolato in base al numero di impulsi in uscita e al valore di decelerazione impostato in S. Quando tale punto viene raggiunto, la frequenza decresce a ogni periodo di controllo dell'impulso (4 ms) in base al valore di decelerazione specificato in S, fino a raggiungere la frequenza di avvio specificata in S. A questo punto l'uscita a impulsi viene interrotta (4 nello schema).

L'uscita a impulsi viene avviata a ogni esecuzione di PLS2(887). In genere è sufficiente utilizzare la versione differenziata dell'istruzione [`@PLS2(887)`] o una condizione di esecuzione che viene impostata su ON per una sola scansione.



È possibile utilizzare PLS2(887) solo per il posizionamento.

Con le CPU CJ1M, è possibile eseguire l'istruzione PLS2(887) durante l'uscita a impulsi per ACC(888) in modalità indipendente o continua e durante accelerazione, velocità costante o decelerazione (vedere nota). È possibile eseguire ACC(888) anche durante l'uscita a impulsi per PLS2(887) durante accelerazione, velocità costante o decelerazione.

Nota L'esecuzione di PLS2(887) durante il controllo della velocità con ACC(888) (modalità continua), con la stessa frequenza di riferimento di ACC(888), può essere utilizzata per ottenere l'avanzamento ad interrupt per una determinata distanza. L'accelerazione non verrà eseguita da PLS2(887) per questa applicazione, ma se il valore di accelerazione viene impostato su 0, il flag di errore verrà attivato e l'istruzione PLS2(887) non verrà eseguita. Impostare sempre il valore di accelerazione su un valore diverso da 0.

■ Posizionamento in modalità indipendente

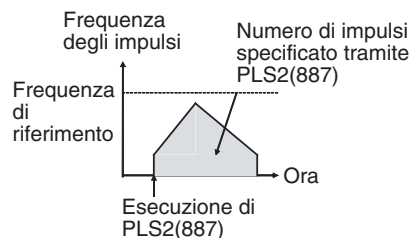
Nota L'uscita a impulsi si interromperà immediatamente se viene attivata la modalità PROGRAM della CPU.

| Operazione | Scopo | Applicazione | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|-----------------------------|--|--|-------------------------|---|-----------------------------|
| Avvio dell'uscita a impulsi | Controllo trapezoidale complesso | Posizionamento con accelerazione e decelerazione trapezoidale (valori indipendenti utilizzati per l'accelerazione e la decelerazione; velocità iniziale specificata). È possibile modificare il numero di impulsi durante il posizionamento. | | <p>Accelera e decelera a velocità fisse. L'uscita a impulsi viene interrotta quando è stato emesso il numero di impulsi specificato (vedere nota).</p> <p>Nota La posizione di riferimento (numero di impulsi specificato) può essere modificata durante il posizionamento.</p> | PLS2(887) |
| Modifica delle impostazioni | Modificare gradualmente e la velocità (con valori di accelerazione e decelerazione differenti) | Modifica della velocità di riferimento (frequenza) durante il posizionamento (valori di accelerazione e decelerazione differenti) | | <p>Durante il posizionamento è possibile eseguire l'istruzione PLS2(887) per modificare il valore di accelerazione, il valore di decelerazione e la frequenza di riferimento.</p> <p>Nota Per impedire la modifica intenzionale della posizione di riferimento, la posizione di riferimento originale deve essere specificata in coordinate assolute.</p> | PLS2(887) ↓ PLS2(887) |
| | | | | <p>Durante il posizionamento è possibile eseguire l'istruzione PLS2(887) per modificare la posizione di riferimento (numero di impulsi), il valore di accelerazione, il valore di decelerazione e la frequenza di riferimento.</p> <p>Nota Se dopo la modifica delle impostazioni risulta impossibile mantenere una velocità costante, si verificherà un errore e l'operazione originale proseguirà fino al raggiungimento della posizione di riferimento originale.</p> | PLS2(887) ↓ PLS2(887) |
| | Modificare la posizione di riferimento | Modifica della posizione di riferimento durante il posizionamento (funzione di avvio multiplo) | | <p>Durante il posizionamento è possibile eseguire l'istruzione PLS2(887) per modificare la posizione di riferimento (numero di impulsi), il valore di accelerazione, il valore di decelerazione e la frequenza di riferimento.</p> <p>Nota Se dopo la modifica delle impostazioni risulta impossibile mantenere una velocità costante, si verificherà un errore e l'operazione originale proseguirà fino al raggiungimento della posizione di riferimento originale.</p> | PLS2(887) ↓ PLS2(887) |

| Operazione | Scopo | Applicazione | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|---------------------------------------|--|---|--|---|--|
| Modifica delle impostazioni, continua | Modificare gradualmente e posizione e velocità di riferimento | Modifica della posizione e della velocità di riferimento (frequenza) durante il posizionamento (funzione di avvio multiplo) | | <p>Durante il posizionamento è possibile eseguire l'istruzione PLS2(887) per modificare la posizione di riferimento (numero di impulsi), il valore di accelerazione, il valore di decelerazione e la frequenza di riferimento.</p> <p>Nota Se dopo la modifica delle impostazioni risulta impossibile mantenere una velocità costante, si verificherà un errore e l'operazione originale proseguirà fino al raggiungimento della posizione di riferimento originale.</p> | PLS2(886) ↓ ACC(888) (indipendente) ↓ PLS2(887) |
| | | Modifica dei valori di accelerazione e decelerazione durante il posizionamento (funzione di avvio multiplo) | | <p>Durante il posizionamento (accelerazione o decelerazione) è possibile eseguire l'istruzione PLS2(887) per modificare il valore di accelerazione o il valore di decelerazione.</p> | PLS2(887) ↓ PLS2(887) ↓ PLS2(886) ↓ ACC(888) (indipendente) ↓ PLS2(887) |
| | | Modificare la direzione | Modifica della direzione durante il posizionamento | | <p>Durante il posizionamento con specifica relativa degli impulsi, è possibile eseguire l'istruzione PLS2(887) per passare alla specifica assoluta e invertire la direzione.</p> |
| Interruzione dell'uscita a impulsi | Interruzione dell'uscita a impulsi (l'impostazione del numero di impulsi non viene conservata) | Interruzione immediata | | <p>Interrompe immediatamente l'emissione degli impulsi e cancella il numero di impulsi in uscita.</p> | PLS2(887) ↓ INI(880) |
| | | Decelerazione fino ad arresto | | <p>Decelera l'uscita a impulsi fino ad arresto.</p> | PLS2(887) ↓ ACC(888) (indipendente, frequenza di riferimento di 0 Hz) |

Nota Controllo triangolare

Se il numero di impulsi specificato è inferiore al numero necessario per raggiungere la frequenza di riferimento e tornare a zero, la funzione ridurrà automaticamente il tempo di accelerazione/decelerazione ed effettuerà un controllo triangolare (solo accelerazione e decelerazione). In tal caso, non si verificherà nessun errore.



■ Passaggio dalla modalità continua (controllo della velocità) alla modalità di posizionamento indipendente

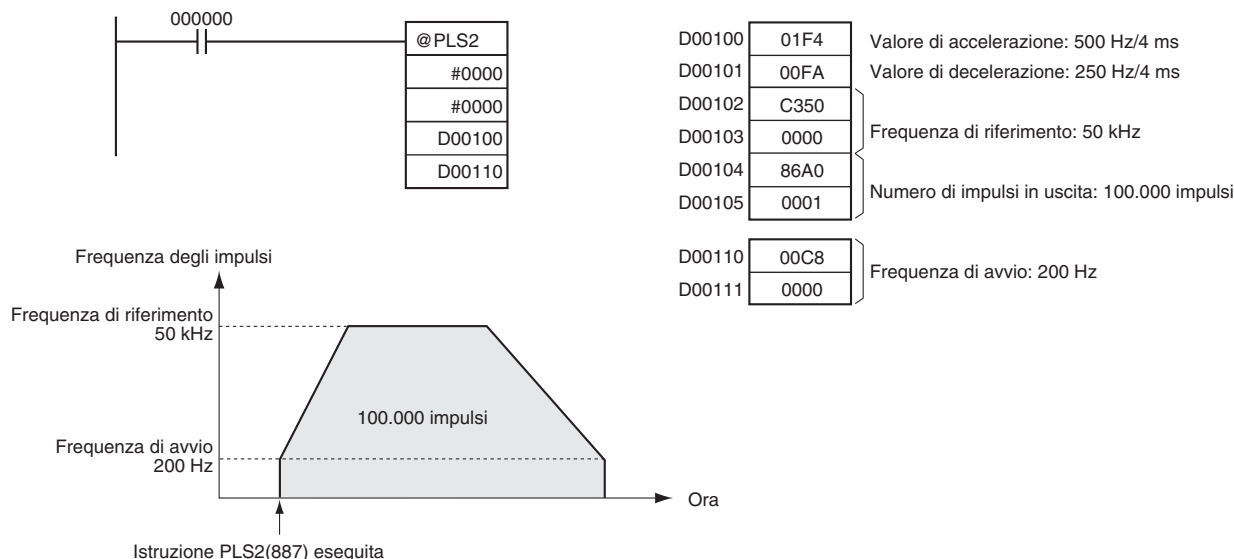
| Applicazione di esempio | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|--|--|--|---------------------------------------|
| Passaggio dal controllo della velocità al posizionamento a distanza fissa durante il funzionamento | <p>Viene emesso il numero di impulsi specificato in PLS2(887). È possibile specificare gli impulsi in modo sia relativo che assoluto.</p> | Durante un'operazione di controllo della velocità avviata tramite ACC(888), è possibile eseguire l'istruzione PLS2(887) per passare alla modalità di posizionamento. | ACC(888) (continua) ↓ PLS2(887) |
| Interrupt di avanzamento fisso | <p>Esecuzione di PLS2(887) con le seguenti impostazioni:</p> <ul style="list-style-type: none"> • Numero di impulsi = numero di impulsi fino ad arresto • Specifica relativa degli impulsi • Frequenza di riferimento = frequenza attuale • Valore di accelerazione = da 0001 a 07D0 esadecimale • Valore di decelerazione = valore di decelerazione di riferimento | | |

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se viene superata la gamma specificata per P, M, S o F. ON se si esegue PLS2(887) per una porta da cui è già in corso un'uscita a impulsi per SPED(885) o ORG(889). ON se si esegue PLS2(887) in un task ad interrupt quando è in esecuzione un'istruzione che controlla l'uscita a impulsi in un task ciclico. ON se si esegue PLS2(887) per un'uscita a impulsi assoluti senza che sia stata stabilita l'origine. |

Esempio

Nel seguente esempio di programmazione, quando CIO 000000 viene impostato su ON, l'istruzione PLS2(887) avvia l'uscita a impulsi 0 con un valore assoluto di 100.000 impulsi. L'uscita a impulsi subisce un'accelerazione di 500 Hz ogni 4 ms, a partire da 200 Hz e fino a raggiungere la velocità di riferimento di 50 kHz. Dal punto di decelerazione, la velocità dell'uscita a impulsi viene ridotta di 250 Hz ogni 4 ms, finché non viene raggiunta la velocità di avvio di 200 Hz. A questo punto l'uscita a impulsi viene interrotta.

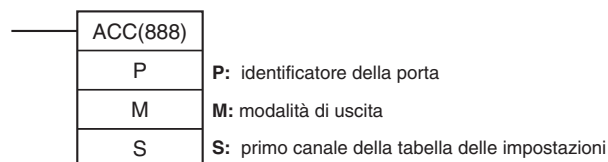


3-21-8 ACCELERATION CONTROL: ACC(888) (solo CPU CJ1M 21/22/23)

Scopo

ACC(888) invia gli impulsi alla porta di uscita specificata, alla frequenza specificata, utilizzando i valori di accelerazione e decelerazione impostati (il valore di accelerazione è uguale al valore di decelerazione). Sono possibili il posizionamento in modalità indipendente o il controllo della velocità in modalità continua. Per il posizionamento, è possibile utilizzare le istruzioni ACC(888) e PULS(886) in combinazione. È possibile eseguire ACC(888) anche durante l'uscita a impulsi per modificare la frequenza di riferimento o i valori di accelerazione/decelerazione e ottenere cambi di velocità progressivi. Questa istruzione è supportata solo dalle CPU CJ1M 21/22/23.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ACC(888) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ACC(888) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

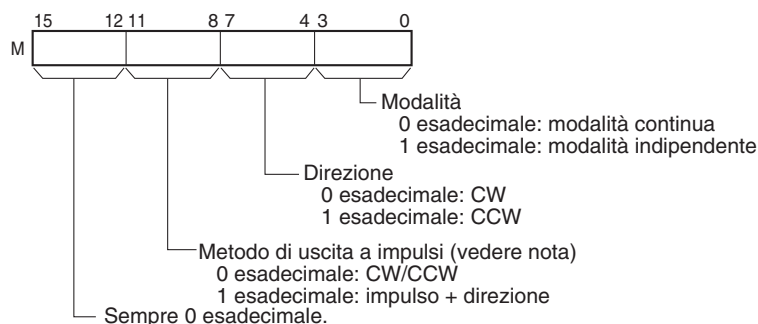
P: identificatore della porta

L'identificatore della porta specifica la porta da dove vengono emessi gli impulsi.

| P | Porta |
|------------------|--------------------|
| 0000 esadecimale | Uscita a impulsi 0 |
| 0001 esadecimale | Uscita a impulsi 1 |

M: modalità di uscita

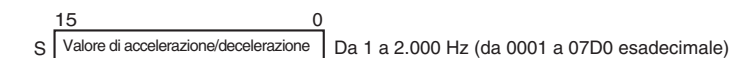
Il contenuto di M definisce i parametri per l'uscita a impulsi come descritto di seguito:



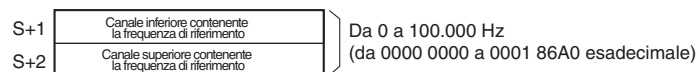
Nota: utilizzare lo stesso metodo di uscita a impulsi per entrambe le uscite a impulsi 0 e 1.

S: primo canale della tabella delle impostazioni

Il contenuto da S a S+2 controlla l'uscita a impulsi come illustrato nei seguenti schemi.



Specificare l'incremento o il decremento della frequenza a ogni periodo di controllo dell'impulso (4 ms).



Specificare l'accelerazione dopo la frequenza in Hz.

Caratteristiche operando

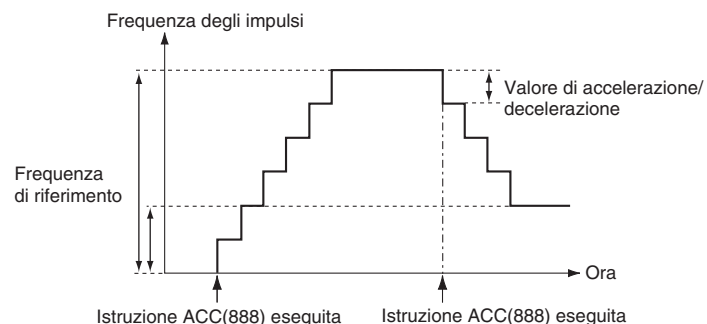
| Area | P | M | S |
|-------------------------|-----|-----|------------------------|
| Area CIO | --- | --- | Da CIO 0000 a CIO 6141 |
| Area di lavoro | --- | --- | Da W000 a W509 |
| Area bit di ritentività | --- | --- | Da H000 a H509 |
| Area bit ausiliaria | --- | --- | Da A448 ad A957 |
| Area del temporizzatore | --- | --- | Da T0000 a T4093 |
| Area del contatore | --- | --- | Da C0000 a C4093 |
| Area DM | --- | --- | Da D00000 a D32765 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |

| Area | P | M | S |
|--|--------------------------------------|--------------------------------------|--|
| Indirizzi DM/EM indiretti in binario | --- | --- | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | --- | Da *D00000 a *D32767 |
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. | --- |
| Registri dati | --- | --- | --- |
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

ACC(888) avvia l'uscita a impulsi sulla porta specificata in P, utilizzando la modalità specificata in M, la frequenza di riferimento e i valori di accelerazione/decelerazione specificati in S. La frequenza aumenta, a ogni periodo di controllo dell'impulso (4 ms), del valore di accelerazione specificato in S fino a raggiungere la frequenza di riferimento specificata in S.

L'uscita a impulsi viene avviata a ogni esecuzione di ACC(888). In genere è sufficiente utilizzare la versione differenziata dell'istruzione [`@ACC(888)`] o una condizione di esecuzione che viene impostata su ON per una sola scansione.



In modalità indipendente l'uscita a impulsi si interrompe automaticamente quando è stato emesso il numero di impulsi specificato. In modalità continua l'uscita a impulsi continua finché non viene interrotta dal programma.

Se si tenta di passare dalla modalità indipendente a quella continua o viceversa durante l'uscita a impulsi, si verificherà un errore.

Con le CPU CJ1M, è possibile eseguire l'istruzione PLS2(887) durante l'uscita a impulsi per ACC(888) in modalità indipendente o continua e durante accelerazione, velocità costante o decelerazione (vedere nota). È possibile eseguire ACC(888) anche durante l'uscita a impulsi per PLS2(887) durante accelerazione, velocità costante o decelerazione.

Nota L'esecuzione di PLS2(887) durante il controllo della velocità con ACC(888) (modalità continua), con la stessa frequenza di riferimento di ACC(888), può essere utilizzata per ottenere l'avanzamento ad interrupt per una determinata distanza. L'accelerazione non verrà eseguita da PLS2(887) per questa applicazione, ma se il valore di accelerazione viene impostato su 0, il flag di errore verrà attivato e l'istruzione PLS2(887) non verrà eseguita. Impostare sempre il valore di accelerazione su un valore diverso da 0.

■ Controllo della velocità in modalità continua

L'uscita a impulsi continua finché non viene interrotta dal programma.

Nota L'uscita a impulsi si interromperà immediatamente se viene attivata la modalità PROGRAM della CPU.

| Operazione | Scopo | Applicazione | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|-----------------------------|---|---|---|--|---|
| Avvio dell'uscita a impulsi | Emettere impulsi con velocità e accelerazione specificate | Accelerazione costante della velocità (frequenza) | <p>Frequenza degli impulsi</p> <p>Frequenza di riferimento</p> <p>Frequenza attuale</p> <p>Valore di accelerazione/decelerazione</p> <p>Ora</p> <p>Esecuzione di ACC(888)</p> | Emette gli impulsi e modifica la frequenza aumentandola in modo costante. | ACC(888) (continua) |
| Modifica delle impostazioni | Modificare gradualmente la velocità | Modifica graduale della velocità durante il funzionamento | <p>Frequenza degli impulsi</p> <p>Frequenza di riferimento</p> <p>Frequenza attuale</p> <p>Valore di accelerazione/decelerazione</p> <p>Ora</p> <p>Esecuzione di ACC(888)</p> | Modifica la frequenza attuale aumentandola o diminuendola in modo costante. La frequenza può essere accelerata o decelerata. | ACC(888) o SPED(885) (continua) ↓ ACC(888) (continua) |
| | | Modifica della velocità in più fasi (una curva spezzata) durante il funzionamento | <p>Frequenza degli impulsi</p> <p>Frequenza di riferimento</p> <p>Frequenza attuale</p> <p>Valore di accelerazione n</p> <p>Valore di accelerazione</p> <p>Valore di decelerazione</p> <p>Ora</p> <p>Esecuzione di ACC(888)</p> <p>Esecuzione di ACC(888)</p> <p>Esecuzione di ACC(888)</p> | Modifica il valore di accelerazione o decelerazione durante l'accelerazione o la decelerazione. | ACC(888) (continua) ↓ ACC(888) (continua) |

| Operazione | Scopo | Applicazione | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|--|---------------------------------|-------------------------------|--|--|--|
| Interruzione dell'uscita a impulsi | Interrompere l'uscita a impulsi | Interruzione immediata | <p>Frequenza degli impulsi</p> <p>Frequenza attuale</p> <p>Esecuzione di ACC(888) Esecuzione di INI(880)</p> | Interrompe immediatamente l'uscita a impulsi. | ACC(888) (continua) ↓ INI(880) (continua) |
| Interrompere l'uscita a impulsi | Interrompere l'uscita a impulsi | Interruzione immediata | <p>Frequenza degli impulsi</p> <p>Frequenza attuale</p> <p>Esecuzione di ACC(888) Esecuzione di SPED(885)</p> | Interrompe immediatamente l'uscita a impulsi. | ACC(888) (continua) ↓ SPED(885) (continua, frequenza di riferimento di 0 Hz) |
| Interrompere gradualmente l'uscita a impulsi | Decelerazione fino ad arresto | Decelerazione fino ad arresto | <p>Frequenza degli impulsi</p> <p>Frequenza attuale</p> <p>Frequenza di riferimento = 0</p> <p>Valore di accelerazione/decelerazione (valore impostato all'avvio)</p> <p>Esecuzione di ACC(888) Esecuzione di ACC(888)</p> | Decelera l'uscita a impulsi fino all'arresto. Nota Se il funzionamento è stato avviato tramite ACC(888), il valore originale di accelerazione/decelerazione resterà valido. Se il funzionamento è stato avviato tramite SPED(885), il valore di accelerazione/decelerazione non sarà più valido e l'uscita a impulsi si interromperà immediatamente. | ACC(888) (continua) ↓ ACC(888) (continua, frequenza di riferimento di 0 Hz) |

■ Posizionamento in modalità indipendente

Quando si avvia il funzionamento in modalità indipendente, l'uscita a impulsi continua finché non è stato emesso il numero di impulsi specificato.

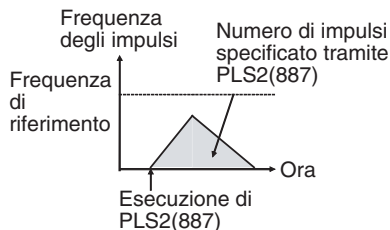
Il punto di decelerazione viene calcolato in base al numero di impulsi in uscita e al valore di decelerazione impostato in S. Quando tale punto viene raggiunto, la frequenza decresce a ogni periodo di controllo dell'impulso (4 ms) in base al valore di decelerazione specificato in S, finché non viene emesso il numero di impulsi specificato. A questo punto l'uscita a impulsi viene interrotta.

- Nota**
1. L'uscita a impulsi si interromperà immediatamente se viene attivata la modalità PROGRAM della CPU.
 2. Il numero di impulsi deve essere impostato ogni volta che viene riavviata l'uscita.
 3. Il numero di impulsi in uscita deve essere impostato utilizzando l'istruzione PULS(881) prima di eseguire l'istruzione ACC(888), altrimenti non verrà emesso alcun impulso.
 4. Se il numero di impulsi impostato con PULS(881) è un valore assoluto, la direzione impostata nell'operando di ACC(888) verrà ignorata.

| Opera-zione | Scopo | Applicazione | Variazioni di frequenza | Descrizione | Procedura/Istruzione |
|---|--|--|-------------------------|--|---|
| Avvio dell'uscita a impulsi | Controllo trapezoidale semplice | Posizionamento con accelerazione e decelerazione trapezoidale (stesso valore utilizzato per l'accelerazione e la decelerazione; nessuna velocità iniziale). Non è possibile modificare il numero di impulsi durante il posizionamento. | | <p>Accelera e decelera in modo costante e interrompe immediatamente l'emissione degli impulsi quando è stato emesso il numero di impulsi specificato (vedere nota).</p> <p>Nota La posizione di riferimento (numero di impulsi specificato) non può essere modificata durante il posizionamento.</p> | <p>PULS (886) ↓ ACC(888) (indipendente)</p> |
| Modifica delle impostazioni | Modificare gradualmente la velocità (con valori di accelerazione e decelerazione identici) | Modifica della velocità di riferimento (frequenza) durante il posizionamento (valore di accelerazione = valore di decelerazione) | | <p>Durante il posizionamento è possibile eseguire l'istruzione ACC(888) per modificare il valore di accelerazione/decelerazione e la frequenza di riferimento. La posizione di riferimento (numero di impulsi specificato) non viene modificata.</p> | <p>PULS (886) ↓ ACC(888) o SPED(885) (indipendente) ↓ ACC(888) (indipendente)</p> |
| Interruzione dell'uscita a impulsi | Interrompere l'uscita a impulsi (l'impostazione del numero di impulsi non viene conservata). | Interruzione immediata | | <p>L'uscita a impulsi viene interrotta immediatamente e gli impulsi rimanenti vengono cancellati.</p> | <p>PULS (886) ↓ ACC(888) (indipendente) ↓ INI(880)</p> |
| Interrompere gradualmente l'uscita a impulsi (l'impostazione del numero di impulsi non viene conservata). | Decelerazione fino ad arresto | | | <p>Decelera l'uscita a impulsi fino ad arresto.</p> <p>Nota Se il funzionamento è stato avviato tramite ACC(888), il valore originale di accelerazione/decelerazione resterà valido. Se il funzionamento è stato avviato tramite SPED(885), il valore di accelerazione/decelerazione non sarà più valido e l'uscita a impulsi si interromperà immediatamente.</p> | <p>PULS (886) ↓ ACC(888) o SPED(885) (indipendente) ↓ ACC(888) (indipendente, frequenza di riferimento di 0 Hz) ↓ PLS2(887) ↓ ACC(888) (indipendente, frequenza di riferimento di 0 Hz)</p> |

Nota Controllo triangolare

Se il numero di impulsi specificato è inferiore al numero necessario per raggiungere la frequenza di riferimento e tornare a zero, la funzione ridurrà automaticamente il tempo di accelerazione/decelerazione ed effettuerà un controllo triangolare (solo accelerazione e decelerazione). In tal caso, non si verificherà nessun errore.

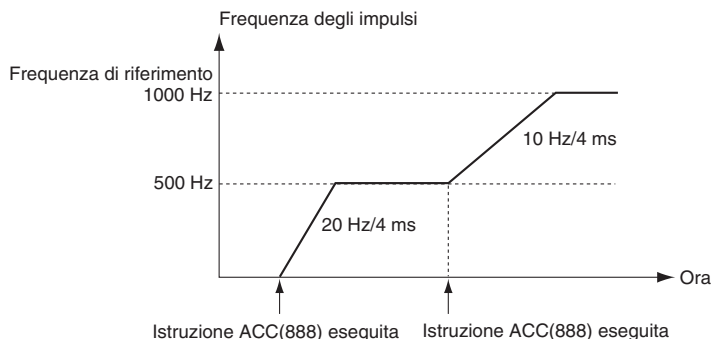
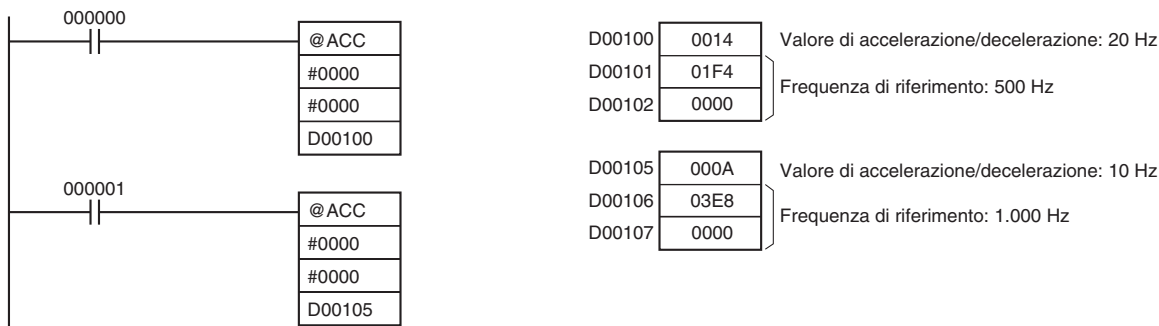


Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | <p>Viene attivato se viene superata la gamma specificata per P, M o S.</p> <p>Viene attivato se gli impulsi vengono emessi utilizzando l'istruzione ORG(889) per la porta specificata.</p> <p>Viene attivato se si esegue ACC(888) per passare dalla modalità indipendente a quella continua o viceversa per una porta da cui è già in corso un'uscita a impulsi per SPED(885), ACC(888) o PLS2(887).</p> <p>Viene attivato se si esegue ACC(888) in un task ad interrupt quando è in esecuzione un'istruzione che controlla l'uscita a impulsi in un task ciclico.</p> <p>Viene attivato se si esegue ACC(888) per un'uscita a impulsi assoluti in modalità indipendente senza che sia stata stabilita l'origine.</p> |

Esempio

Nel seguente esempio di programmazione, quando CIO 000000 viene impostato su ON, l'istruzione ACC(888) avvia l'uscita a impulsi 0 in modalità continua in senso orario, utilizzando il metodo CW/CCW. L'uscita a impulsi subisce un'accelerazione di 20 Hz ogni 4 ms, fino a raggiungere la frequenza di riferimento di 500 Hz. Quando CIO 000001 viene impostato su ON, l'istruzione ACC(888) imposta il valore di accelerazione su 10 Hz ogni 4 ms, fino a raggiungere la frequenza di riferimento di 1.000 Hz.



3-21-9 ORIGIN SEARCH: ORG(889) (solo CPU CJ1M 21/22/23)

Scopo Utilizzare ORG(889) per eseguire operazioni di ricerca dell'origine o di ritorno all'origine.
Questa istruzione è supportata solo dalle CPU CJ1M 21/22/23.

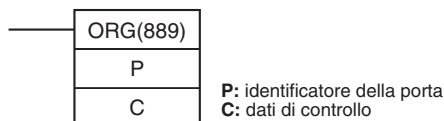
■ Ricerca dell'origine

Gli impulsi vengono emessi utilizzando il metodo specificato per azionare il motore e stabilire l'origine in base ai segnali di ingresso dell'origine e di prossimità dell'origine.

■ Ritorno origine

Il sistema di posizionamento viene riportato nella posizione di origine prestabilita.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ORG(889) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ORG(889) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

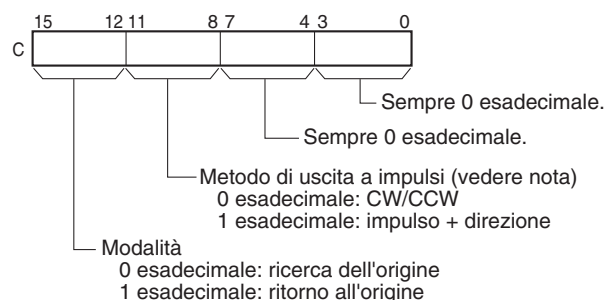
P: identificatore della porta

L'identificatore della porta specifica la porta da dove vengono emessi gli impulsi.

| P | Porta |
|------------------|--------------------|
| 0000 esadecimale | Uscita a impulsi 0 |
| 0001 esadecimale | Uscita a impulsi 1 |

C: dati di controllo

Il valore di C determina il metodo di ricerca dell'origine.



Nota: utilizzare lo stesso metodo di uscita a impulsi per entrambe le uscite a impulsi 0 e 1.

Caratteristiche operando

| Area | P | C |
|-------------------------|-----|-----|
| Area CIO | --- | --- |
| Area di lavoro | --- | --- |
| Area bit di ritentività | --- | --- |
| Area bit ausiliaria | --- | --- |
| Area del temporizzatore | --- | --- |
| Area del contatore | --- | --- |
| Area DM | --- | --- |

| Area | P | C |
|--|--------------------------------------|--------------------------------------|
| Area EM senza banco | --- | --- |
| Area EM con banco | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | --- |
| Indirizzi DM/EM indiretti in BCD | --- | --- |
| Costanti | Vedere la descrizione dell'operando. | Vedere la descrizione dell'operando. |
| Registri dati | --- | --- |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | --- |

Descrizione

ORG(889) esegue un'operazione di ricerca dell'origine o di ritorno all'origine per la porta specificata in P, utilizzando il metodo specificato in C.

Prima di eseguire l'istruzione ORG(889), è necessario impostare i parametri riportati di seguito nella configurazione del PLC. Per ulteriori informazioni, fare riferimento al *Manuale dell'operatore degli I/O integrati della serie CJ*.

| Ricerca dell'origine | Ritorno all'origine |
|---|--|
| Abilitazione/Disabilitazione della funzione di ricerca dell'origine | Velocità iniziale di ricerca dell'origine e di ritorno all'origine |
| Modalità operativa della ricerca dell'origine | Velocità di riferimento del ritorno all'origine |
| Impostazione di funzionamento della ricerca dell'origine | Valore di accelerazione del ritorno all'origine |
| Metodo di rilevamento dell'origine | Valore di decelerazione del ritorno all'origine |
| Impostazione della direzione di ricerca dell'origine | |
| Velocità iniziale di ricerca dell'origine e di ritorno all'origine | |
| Alta velocità di ricerca dell'origine | |
| Velocità di prossimità della ricerca dell'origine | |
| Compensazione dell'origine | |
| Valore di accelerazione della ricerca dell'origine | |
| Valore di decelerazione della ricerca dell'origine | |
| Tipo di segnale di ingresso limite | |
| Tipo di segnale di ingresso di prossimità dell'origine | |
| Tipo di segnale di ingresso dell'origine | |

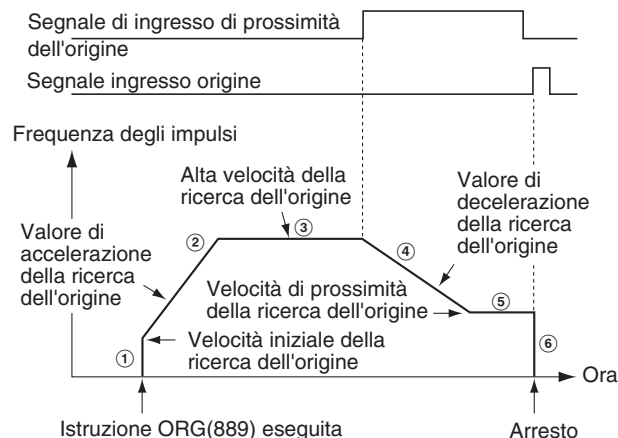
La ricerca dell'origine o il ritorno all'origine viene avviato ogni volta che si esegue l'istruzione ORG(889). In genere è sufficiente utilizzare la versione differenziata dell'istruzione [`@ORG(889)`] o una condizione di esecuzione che viene impostata su ON per una sola scansione.

■ **Ricerca dell'origine (bit da 12 a 15 di C = 0 esadecimale)**

ORG(889) inizia a emettere impulsi utilizzando il metodo specificato alla velocità iniziale della ricerca dell'origine (1 nello schema). L'uscita a impulsi viene accelerata fino alla velocità massima di ricerca dell'origine utilizzando il valore di accelerazione della ricerca dell'origine (2 nello schema). L'uscita a impulsi quindi continua a velocità costante finché non si attiva il segnale di ingresso di prossimità dell'origine (3 nello schema). Da questo punto in poi l'uscita a impulsi decelera fino alla velocità di prossimità di ricerca dell'origine utilizzando il valore di decelerazione della ricerca dell'origine (4 nello schema). Gli impulsi vengono quindi emessi a velocità costante finché non si attiva il segnale di ingresso dell'origine (5 nello schema). L'uscita a impulsi si interrompe quando si attiva il segnale di ingresso dell'origine (6 nello schema).

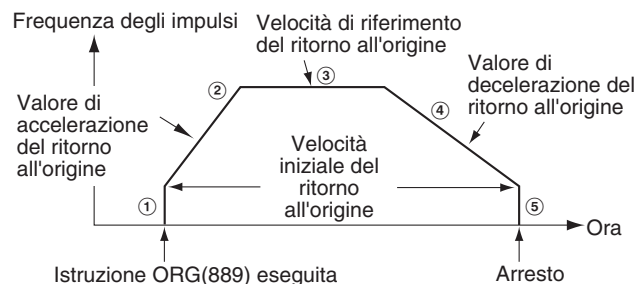
Al termine dell'operazione di ricerca dell'origine, viene attivata l'uscita di reset del contatore di errori. L'esecuzione dell'operazione precedente dipende,

tuttavia, da modalità operativa, metodo di rilevamento dell'origine e altri parametri. Per ulteriori informazioni, fare riferimento al *Manuale dell'operatore degli I/O integrati della serie CJ*.



■ Ritorno all'origine (bit da 12 a 15 di C = 1 esadecimale)

ORG(889) inizia a emettere impulsi utilizzando il metodo specificato alla velocità iniziale del ritorno all'origine (1 nello schema). L'uscita a impulsi viene accelerata fino alla velocità di riferimento di ritorno all'origine utilizzando il valore di accelerazione del ritorno all'origine (2 nello schema), quindi l'uscita a impulsi continua a velocità costante (3 nello schema). Il punto di decelerazione viene calcolato in base al numero di impulsi rimanenti per raggiungere l'origine e al valore di decelerazione. Quando tale punto viene raggiunto, l'uscita a impulsi decelera (4 nello schema) in base al valore di decelerazione di ritorno all'origine, fino a raggiungere la velocità iniziale di ritorno all'origine. A questo punto l'uscita a impulsi viene interrotta in corrispondenza dell'origine (5 nello schema).

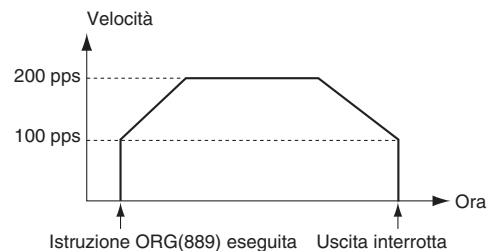
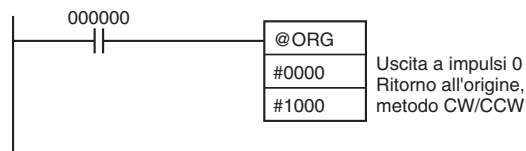


Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se viene superata la gamma specificata per P o C.</p> <p>ON se si specifica ORG(889) per una porta da cui è già in corso un'uscita a impulsi per SPED(885), ACC(888) o PLS2(887).</p> <p>ON se si esegue ORG(889) in un task ad interrupt quando è in esecuzione un'istruzione che controlla l'uscita a impulsi in un task ciclico.</p> <p>ON se i parametri di ricerca dell'origine o di ritorno all'origine impostati nella configurazione del PLC non sono compresi nella gamma.</p> <p>ON se la velocità massima di ricerca dell'origine è minore o uguale alla velocità di prossimità di ricerca dell'origine o se la velocità di prossimità di ricerca dell'origine è minore o uguale alla velocità iniziale della ricerca dell'origine.</p> <p>ON se la velocità di riferimento di ritorno all'origine è minore o uguale alla velocità iniziale del ritorno all'origine.</p> <p>ON se si tenta di eseguire un'operazione di ritorno all'origine quando l'origine non è stata stabilita.</p> |

Esempio

Nel seguente esempio di programmazione, quando CIO 000000 viene impostato su ON, l'istruzione ORG(889) avvia un'operazione di ritorno all'origine per l'uscita a impulsi 0, emettendo impulsi con il metodo CW/CCW. In base alla configurazione del PLC, la velocità iniziale è di 100 pps, la velocità di riferimento è di 200 pps e i valori di accelerazione e decelerazione sono di 50 Hz/4 ms.



I parametri della configurazione del PLC sono impostati come riportato nella seguente tabella.

| Parametro | Impostazione |
|--|--|
| Velocità iniziale dell'uscita a impulsi 0 per la ricerca dell'origine e il ritorno all'origine | 0000 0064 esadecimale: 100 pps |
| Velocità di riferimento di ritorno all'origine per l'uscita a impulsi 0 | 0000 00C8 esadecimale: 200 pps |
| Valore di accelerazione del ritorno all'origine per l'uscita a impulsi 0 | 0032 esadecimale: 50 esadecimale/4 ms |
| Valore di decelerazione del ritorno all'origine per l'uscita a impulsi 0 | 0032 esadecimale: 50 esadecimale/4 ms |

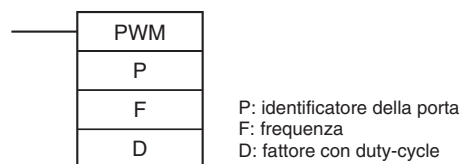
3-21-10 PULSE WITH VARIABLE DUTY FACTOR: PWM(891) (solo CPU CJ1M 21/22/23)

Scopo

L'istruzione PWM(891) viene utilizzata per l'emissione di impulsi con duty-cycle specificato dalla porta impostata.

Questa istruzione è supportata solo dalle CPU CJ1M 21/22/23.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | PWM(891) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PWM(891) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

P: identificatore della porta

L'identificatore della porta specifica la porta da dove vengono emessi gli impulsi.

| P | Porta |
|------------------|--|
| 0000 esadecimale | Uscita a impulsi 0 (duty-cycle: in incrementi dell'1%) |
| 0001 esadecimale | Uscita a impulsi 1 (duty-cycle: in incrementi dell'1%) |

| P | Porta |
|--|--|
| 1000 esadecimale (Solo CPU CJ1M versione 2.0) | Uscita a impulsi 0 (duty-cycle: in incrementi dell'0,1%) |
| 1001 esadecimale (Solo CPU CJ1M versione 2.0) | Uscita a impulsi 1 (duty-cycle: in incrementi dell'0,1%) |

F: frequenza

F specifica la frequenza dell'uscita a impulsi come valore compreso tra 0,1 e 6.553,5 Hz (in unità di 0,1 Hz, da 0001 a FFFF esadecimale). La precisione della forma d'onda di PWM(891) effettivamente emessa (duty-cycle +5%/–0%) è applicabile solo alle frequenze comprese tra 0,1 e 1.000,0 Hz a causa delle limitazioni implicite dei circuiti di uscita.

D: duty-cycle

D specifica il duty-cycle dell'uscita a impulsi, vale a dire la percentuale di tempo in cui l'uscita è attiva. Il valore di D deve essere compreso nell'intervallo indicato di seguito.

- CPU CJ1m precedenti alla versione 2.0
0% e 100% (unità da 1%, da 0000 a 0064 esadecimale)
- CPU CJ1m versione 2.0
0,0% e 100,0% (unità da 0,1%, da 0000 a 03E8 esadecimale)

Caratteristiche operando

| Area | P | F | D |
|--|--------------------------------------|--|----------------------------|
| Area CIO | --- | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 | Da W000 a W511 |
| Area bit di ritentività | --- | Da H000 a H511 | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A448 a A959 | Da A448 a A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 | Da D00000 a D32767 |
| Area EM senza banco | --- | --- | --- |
| Area EM con banco | --- | --- | --- |
| Indirizzi DM/EM indiretti in binario | --- | Da @ D00000 a @ D32767 | Da @ D00000 a @ D32767 |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 | Da *D00000 a *D32767 |
| Costanti | Vedere la descrizione dell'operando. | 0000 ... FFFF esadecimale | Da 0000 a 0064 esadecimale |
| Registri dati | --- | Da DR0 a DR15 | Da DR0 a DR15 |
| Registri indice | --- | --- | --- |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da –2048 a +2047, da ,IR0 a da –2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(–)IR0 a ,-(–)IR15 | |

Descrizione

PWM(891) emette la frequenza specificata in F, al duty-cycle specificato in D, dalla porta specificata in P. È possibile eseguire PWM(891) durante l'uscita a impulsi per modificare il duty-cycle senza interrompere l'uscita a impulsi. Qualsiasi tentativo di modificare la frequenza verrà ignorato.

L'uscita a impulsi viene avviata a ogni esecuzione di PWM(891). In genere è sufficiente utilizzare la versione differenziata dell'istruzione [@PWM(891)] o una condizione di esecuzione che viene impostata su ON per una sola scansione.

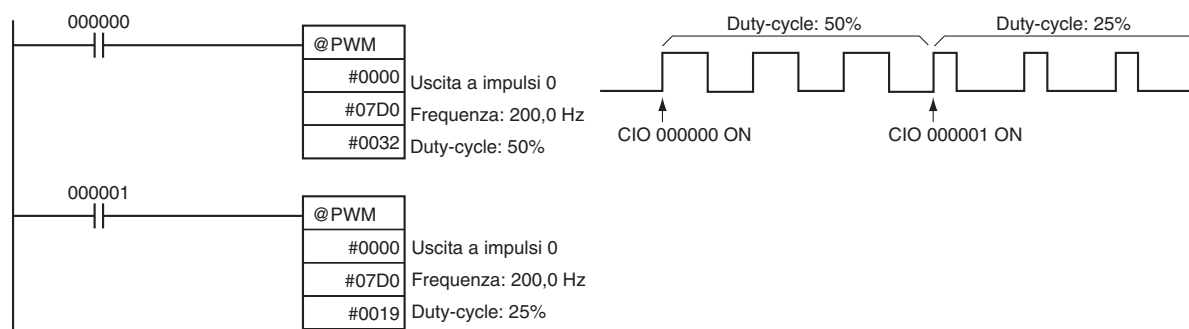
L'uscita a impulsi continuerà finché non viene eseguita l'istruzione INI(880) per interromperla (C = 0003 esadecimale: interruzione dell'uscita a impulsi) oppure finché non viene attivata la modalità PROGRAM della CPU.

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se viene superato l'intervallo specificato per P, F o D. ON se gli impulsi vengono emessi utilizzando l'istruzione ORG(889) per la porta specificata. ON se si esegue PWM(891) in un task ad interrupt quando è in esecuzione un'istruzione che controlla l'uscita a impulsi in un task ciclico. |

Esempio

Nel seguente esempio di programmazione, quando CIO 000000 viene impostato su ON, l'istruzione PWM(891) avvia l'uscita a impulsi 0 a 200 Hz, con un duty-cycle del 50%. Quando CIO 000001 viene impostato su ON, il duty-cycle viene impostato sul 25%.



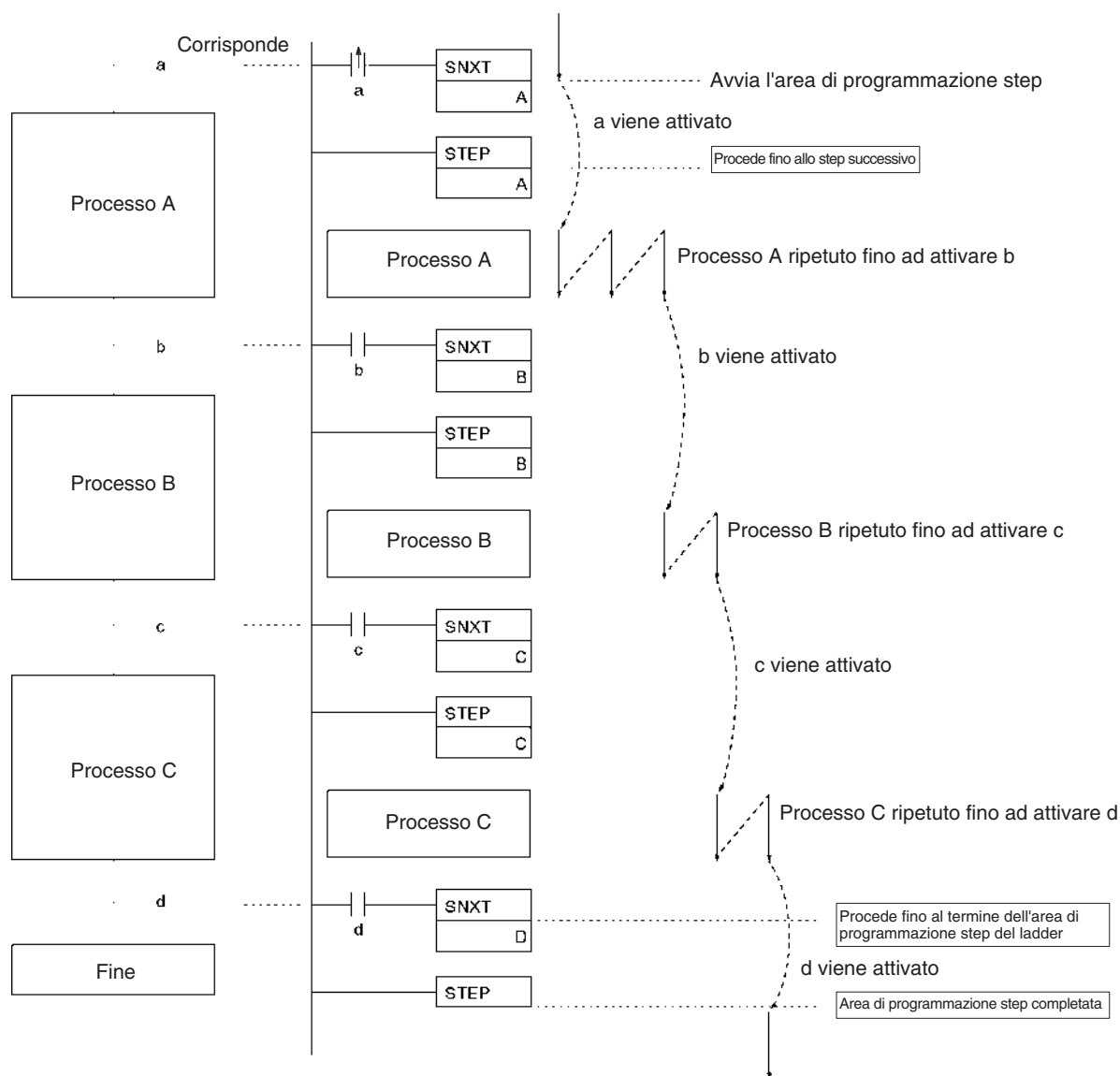
3-22 Istruzioni di step

In questa sezione vengono descritte le istruzioni di step, utilizzate per definire dei punti di interruzione tra le sezioni in un programma di grandi dimensioni, che permettono di eseguire le sezioni come unità e reimpostarle dopo il completamento.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|-------------|-----------|-----------------|--------|
| STEP DEFINE | STEP | 008 | 868 |
| STEP START | SNXT | 009 | 868 |

Nei PLC della serie CS/CJ è possibile utilizzare insieme le istruzioni STEP(008)/SNXT(009) per creare programmi con struttura a step.

| Istruzione | Operazione | Diagramma |
|------------------------|---|-------------|
| SNXT(009): STEP START | Controlla la progressione fino allo step successivo del programma. | Corrisponde |
| STEP(008): STEP DEFINE | Indica l'inizio di uno step. Ripete lo stesso programma step fino a stabilire le condizioni per la progressione allo step successivo. | Corrisponde |



Nota I bit di lavoro vengono usati come bit di controllo per A, B, C e D.

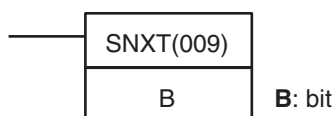
3-22-1 STEP DEFINE e STEP START: STEP(008)/SNXT(009)

Scopo

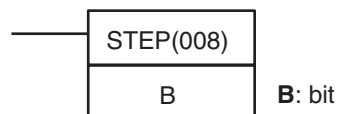
L'istruzione SNXT(009) viene inserita subito prima dell'istruzione STEP(008) e controlla l'esecuzione degli step attivando il bit di controllo specificato. Se subito prima di SNXT(009) è presente un altro step, disattiva anche il bit di controllo di tale processo.

STEP(008) viene inserita subito dopo l'istruzione SNXT(009) e prima di ogni processo. Definisce l'inizio di ogni processo e specifica il bit di controllo per tale processo. Viene inoltre inserita alla fine dell'area di programmazione step, dopo l'ultima istruzione SNXT(009), ad indicare il termine dell'area di programmazione step. Quando si trova alla fine dell'area di programmazione step, STEP(008) non attiva un bit di controllo.

Simboli programmazione ladder



Quando si definisce l'inizio di uno step, un bit di controllo viene specificato nel modo seguente:



Quando si definisce la fine di uno step, un bit di controllo non viene specificato nel modo seguente:



Variazioni

| | | |
|-------------------------|--|-------------------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | STEP(008)/ SNXT(009) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|----------------|-------------------|
| Non consentita | OK | Non consentita | Non consentita |

Caratteristiche operando

| Area | B |
|--|--|
| Area CIO | --- |
| Area di lavoro | Da W00000 a W51115 |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 |

Descrizione

SNXT(009)

L'istruzione SNXT(009) viene utilizzata nei seguenti tre modi:

1,2,3...

1. Per avviare l'esecuzione della programmazione step.
2. Per passare al bit di controllo dello step successivo.
3. Per terminare l'esecuzione della programmazione step.

L'area di programmazione step è compresa tra la prima istruzione STEP(008) (che attiva sempre un bit di controllo) e l'ultima istruzione STEP(008) (che non attiva mai un bit di controllo).

Avvio dell'esecuzione step

SNXT(009) viene inserita all'inizio di un'area di programmazione step per avviare l'esecuzione step. Attiva il bit di controllo specificato per B per l'istruzione STEP(008) successiva e passa allo step B (tutte le istruzioni dopo lo STEP(008) B). È necessario utilizzare una condizione di esecuzione differenziata per l'istruzione SNXT(009) che avvia l'esecuzione dell'area di programmazione step, altrimenti l'esecuzione step avrà la durata di un ciclo soltanto.

Avanzamento fino allo step successivo

Quando SNXT(009) viene eseguita all'interno di un'area di programmazione step, viene utilizzata per passare allo step successivo. L'istruzione disattiva il bit di controllo precedente e attiva il bit di controllo B successivo, per lo step successivo, avviando di fatto lo step B (tutte le istruzioni dopo lo STEP(008) B).

Fine dell'area di programmazione step

Quando SNXT(009) viene inserita alla fine dell'area di programmazione step, termina l'esecuzione step e disattiva il bit di controllo precedente. Il bit di controllo specificato per B è un bit fittizio. Tuttavia, questo bit sarà attivato, in modo da essere sicuri di selezionare un bit che non causerà problemi.

STEP(008)

STEP(008) opera nei due modi indicati di seguito, a seconda della posizione e della presenza o meno di un bit di controllo specificato.

1,2,3...

1. Avvia uno step specifico.
2. Termina l'area di programmazione step (ovvero l'esecuzione step).

Avvio di uno step

STEP(008) viene inserita all'inizio di ogni step con un operando (B) che serve come bit di controllo per lo step.

Il bit di controllo B verrà attivato da SNXT(009) e l'istruzione nello step verrà eseguita da quello immediatamente successivo a STEP(008). A20012 (flag di step) verrà inoltre attivato quando inizia l'esecuzione di uno step.

Dopo il primo ciclo, l'esecuzione step proseguirà finché non saranno soddisfatte le condizioni per cambiare step, ossia finché l'istruzione SNXT(009) non attiverà il bit di controllo nell'istruzione STEP(008) successiva.

Quando SNXT (009) attiva il bit di controllo per uno step, il bit di controllo B dell'istruzione corrente verrà reimpostato (su OFF) e lo step controllato dal bit B diventerà interbloccato.

La gestione delle uscite e delle istruzioni in uno step varia in base allo stato ON/OFF del bit di controllo B (lo stato del bit di controllo è definito dall'istruzione SNXT(009)). Quando il bit di controllo B è OFF, le istruzioni nello step vengono reimpostate e sono interbloccate. Fare riferimento alle seguenti tabelle.

| Stato del bit di controllo | Gestione |
|----------------------------|--|
| ON | Le istruzioni nello step vengono eseguite regolarmente. |
| ON→OFF | I bit e le istruzioni nello step sono interbloccati, come mostrato nella tabella di seguito. |
| OFF | Tutte le istruzioni nello step vengono elaborate come istruzioni NOP. |

Stato di interblocco (IL)

| Uscita istruzione | | Stato |
|---|-----------------------|--|
| Bit specificati per OUT, OUT NOT | | Tutti OFF |
| Le seguenti istruzioni di temporizzatore: TIM, TIMX(551), TIMH(015), TIMHX(551), TMHH(540), TIMHHX(552), TIML(542) e TIMLX(553) | PV | 0000 esadecimale (reset) |
| | Flag di completamento | OFF (reset) |
| Bit o canali specificati per altre istruzioni (vedere nota) | | Mantiene lo stato precedente, ma le istruzioni non vengono eseguite. |

Nota Indica tutte le istruzioni come TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, REST, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT(010) e KEEP(011).

L'istruzione STEP(008) deve essere inserita all'inizio di ciascuno step. STEP(008) viene posizionata all'inizio di un'area di programmazione step per definire l'inizio dello step.

Fine dell'area di programmazione step

STEP(008) viene posizionata alla fine dell'area di programmazione step senza un operando per definire il termine dello step. Quando il bit di controllo che precede un'istruzione SNXT(009) viene disattivato, l'esecuzione step viene interrotta da SNXT(009).

Flag: STEP(008)

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se il bit B specificato non è nell'area WR. ON se STEP(008) viene utilizzata come programma di interrupt. OFF in tutti gli altri casi. |

Flag: SNXT(009)

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se il bit B specificato non è nell'area WR. ON se SNXT(009) viene utilizzata come programma di interrupt. OFF in tutti gli altri casi. |

Avvertenze

Il bit di controllo B deve essere nell'area di lavoro per STEP(008)/SNXT(009).

Un bit di controllo per STEP(008)/SNXT(009) non può essere utilizzato in un'altra posizione nel diagramma ladder. Se lo stesso bit viene usato due volte, verrà generato un errore di duplicazione bit.

Se SBS(091) è utilizzata per chiamare una subroutine dall'interno di uno step, le uscite e le istruzioni della subroutine non verranno interbloccate quando il bit di controllo viene disattivato.

I bit di controllo all'interno di una sezione di programmazione step devono essere sequenziali e provenienti dallo stesso canale.

SNXT(009) verrà eseguita una sola volta, ad esempio sul fronte di salita della condizione di esecuzione.

Immettere SNXT(009) alla fine dell'area di programmazione step e verificare che il bit di controllo nell'area di lavoro sia un bit fittizio. Se un bit di controllo per uno step viene utilizzato nell'ultima istruzione SNXT(009) nell'area di programmazione step, lo step corrispondente verrà avviato quando viene eseguita SNXT(009).

Verrà generato un errore e il flag di errore verrà attivato se l'operando B specificato per SNXT(009) o per STEP(008) non è nell'area di lavoro o se il programma step è stato inserito in una posizione diversa da un task ciclico.

A20012 (flag di step) verrà attivato per un ciclo quando viene eseguita l'istruzione STEP(008). È possibile utilizzare questo flag per eseguire l'inizializzazione dopo che l'esecuzione dello step è stata avviata.

Condizioni per l'inserimento di aree di programmazione step (da STEP B a STEP)

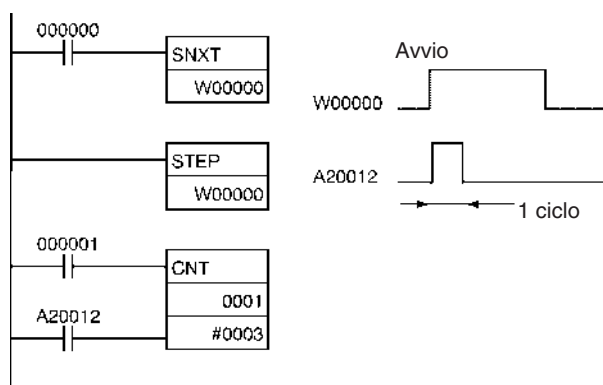
Non è possibile utilizzare STEP(008) e SNXT(009) all'interno di subroutine, programmi di interrupt o programmi a blocchi.

Assicurarsi che due step non vengano eseguiti durante lo stesso ciclo.

Istruzioni che non possono essere utilizzate all'interno di programmi step

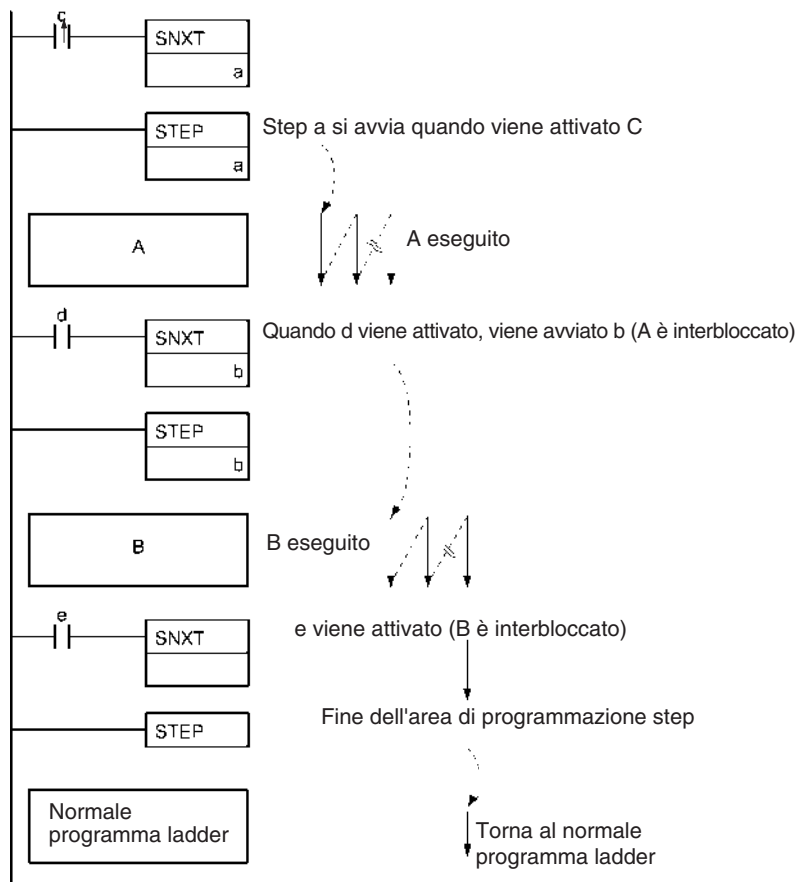
Le istruzioni che non possono essere utilizzate all'interno di programmi step sono elencate nella tabella seguente.

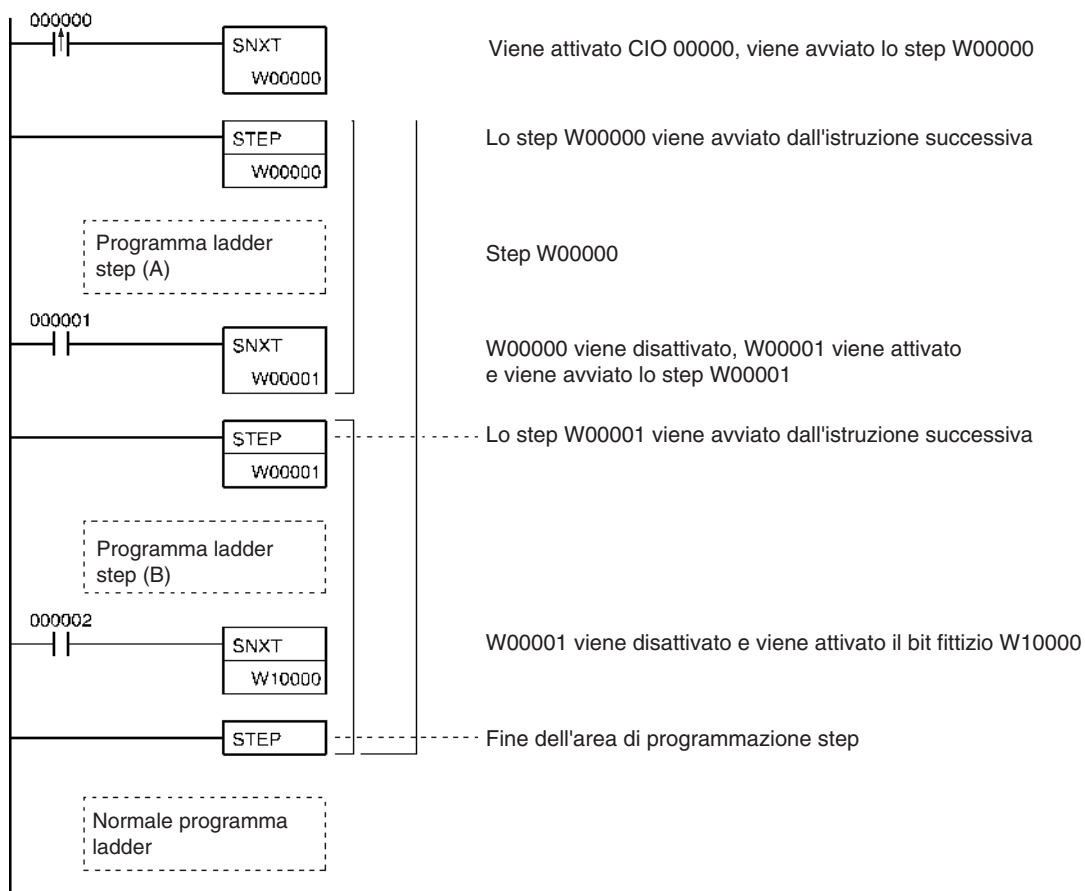
| Funzione | Mnemonico | Nome |
|----------------------------------|-------------------|----------------------|
| Istruzioni di controllo sequenza | END(001) | END |
| | IL(002) | INTERLOCK |
| | ILC(003) | INTERLOCK CLEAR |
| | JMP(004) | JUMP |
| | JME(005) | JUMP END |
| | CJP(510) | CONDITIONAL JUMP |
| | CJPN(511) | CONDITIONAL JUMP NOT |
| | JMP0(515) | MULTIPLE JUMP |
| JME0(516) | MULTIPLE JUMP END | |
| Istruzioni di subroutine | SBN(092) | SUBROUTINE ENTRY |
| | RET(093) | SUBROUTINE RETURN |



Bit correlati

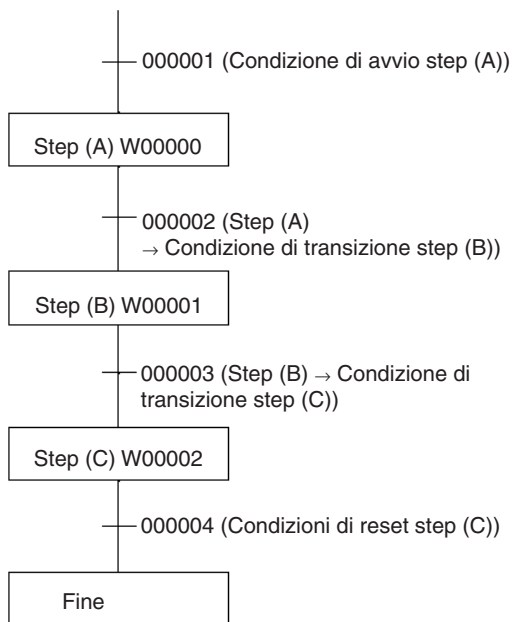
| Nome | Indirizzo | Dettagli |
|--------------|-----------|--|
| Flag di step | A20012 | Viene attivato per un ciclo quando viene avviato un programma step tramite l'istruzione STEP(008). Può essere utilizzato per reimpostare i timer ed eseguire altre elaborazioni quando si avvia un nuovo step. |

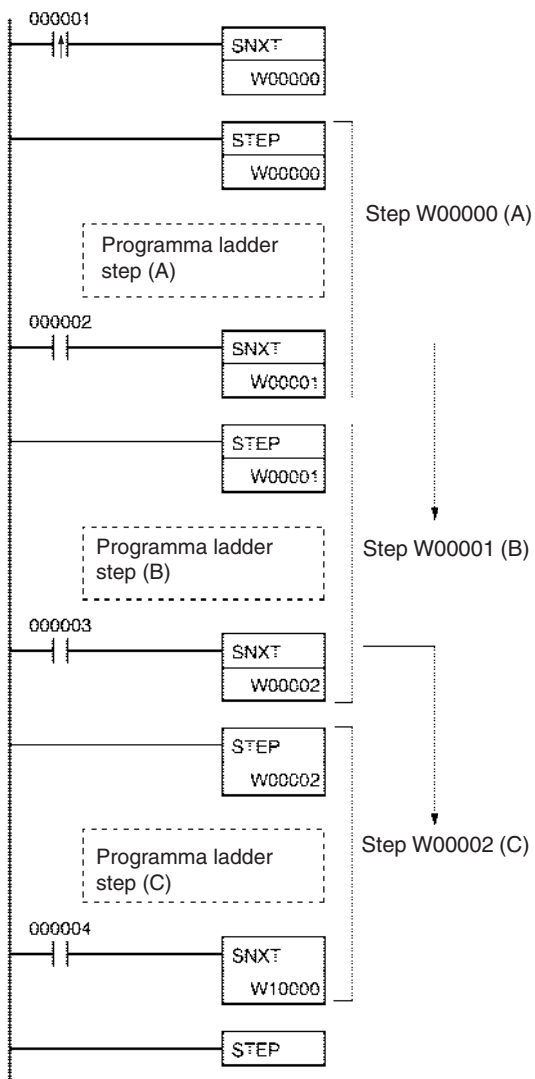




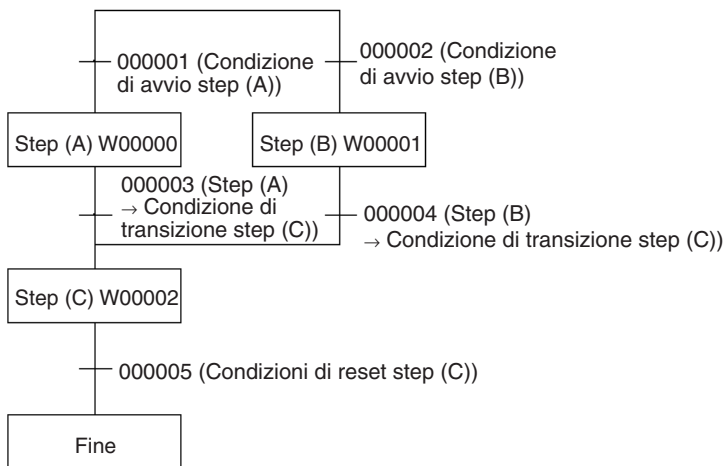
Esempi

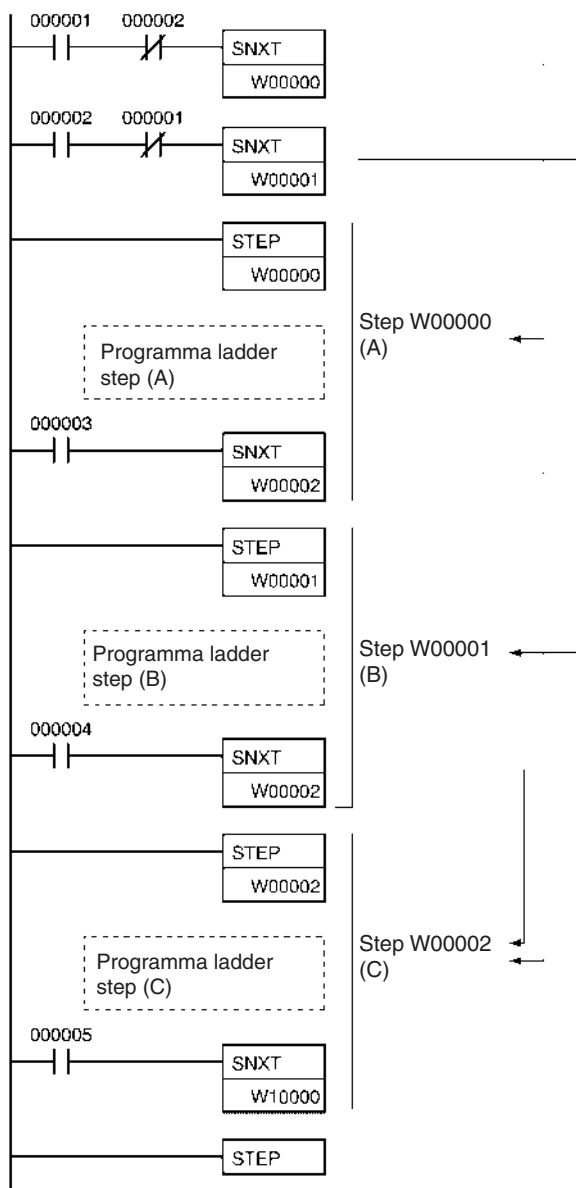
Controllo sequenziale



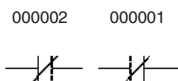


Controllo della diramazione



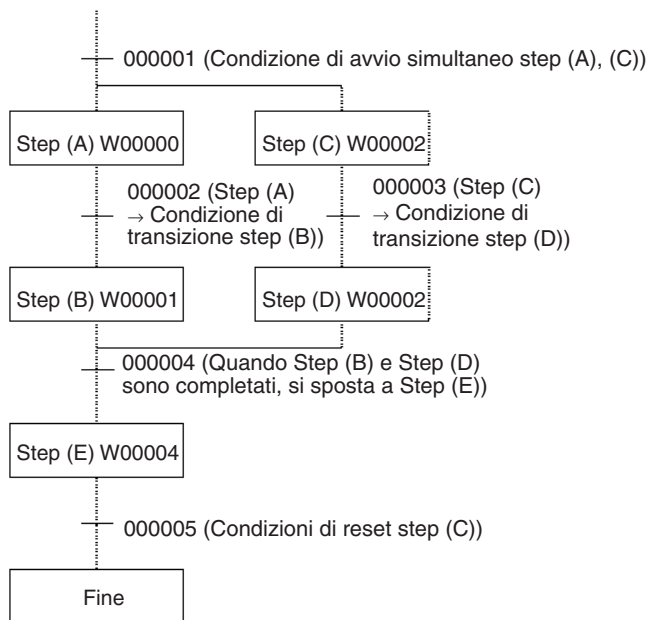


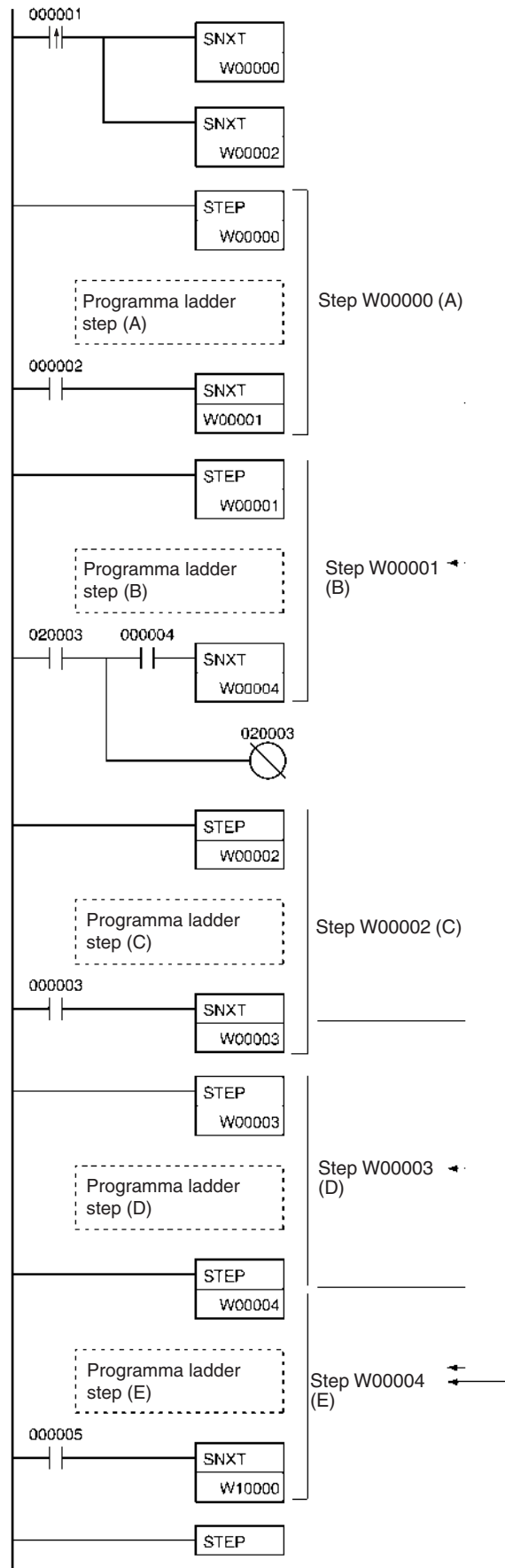
La programmazione precedente viene utilizzata quando gli step A e B non possono essere eseguiti simultaneamente. Per l'esecuzione simultanea di A e B, è necessario eliminare le condizioni di esecuzione illustrate di seguito.



Nota Nell'esempio precedente, in cui l'istruzione SNXT(009) viene eseguita per W00002, la diramazione si sposta agli step successivi anche se viene utilizzato due volte lo stesso bit di controllo. Ciò non viene rilevato come errore nel controllo del programma quando si utilizza CX-Programmer. Un errore per bit duplicato si verifica soltanto in un programma ladder a step quando un bit di controllo in un'istruzione step viene utilizzato anche nel normale diagramma ladder.

Controllo parallelo



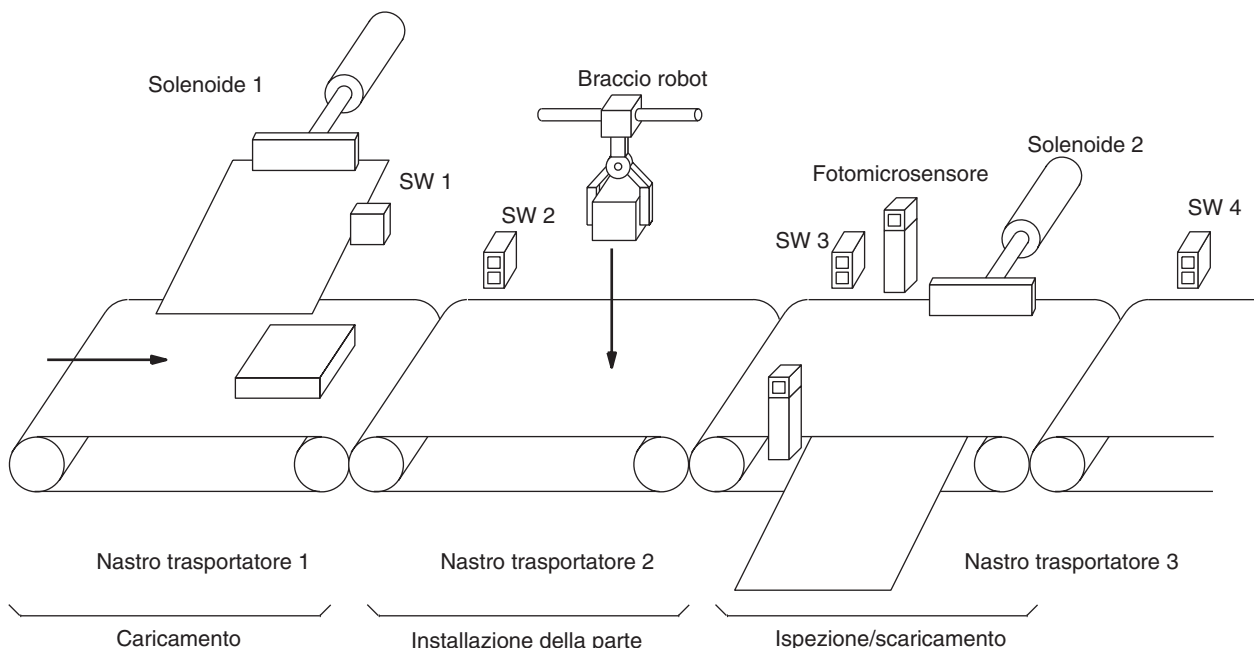


Esempi di applicazione

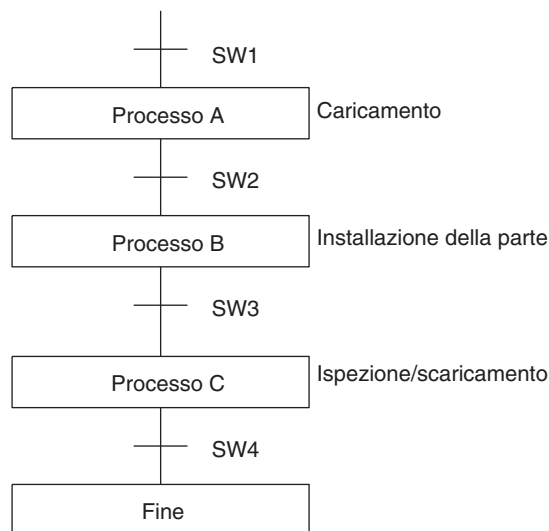
Nei seguenti tre esempi vengono mostrati i tre tipi di controllo di esecuzione possibili con la programmazione step. Nell'*Esempio 1* viene mostrata l'esecuzione sequenziale, nell'*Esempio 2* l'esecuzione di diramazione e nell'*Esempio 3* l'esecuzione parallela.

Esempio 1: Esecuzione sequenziale

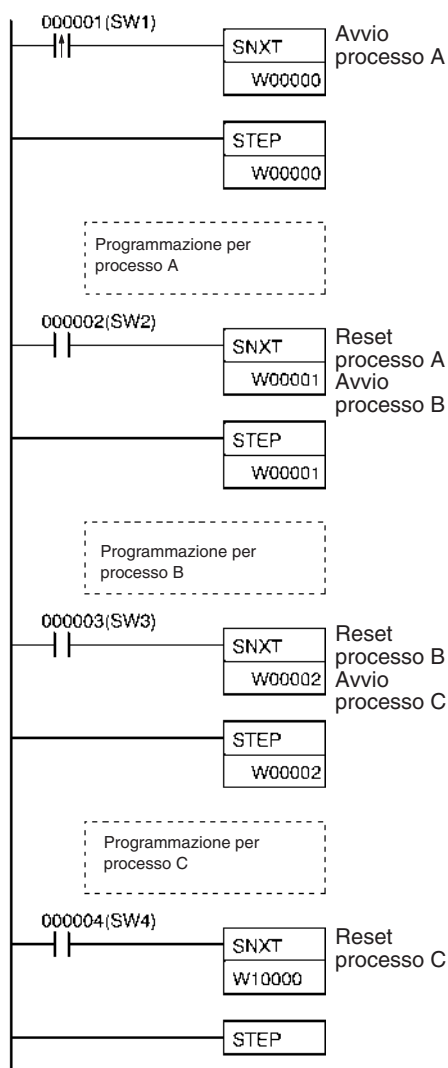
Il processo seguente richiede l'esecuzione in sequenza di tre processi (caricamento, installazione della parte e ispezione/scaricamento), ciascuno dei quali viene azzerato prima di proseguire con il processo successivo. Vari sensori (SW1, SW2, SW3 e SW4) sono posizionati sui segnali dove i processi devono iniziare e terminare.



Nel diagramma seguente vengono mostrati il flusso di elaborazione e i selettori utilizzati per il controllo dell'esecuzione.



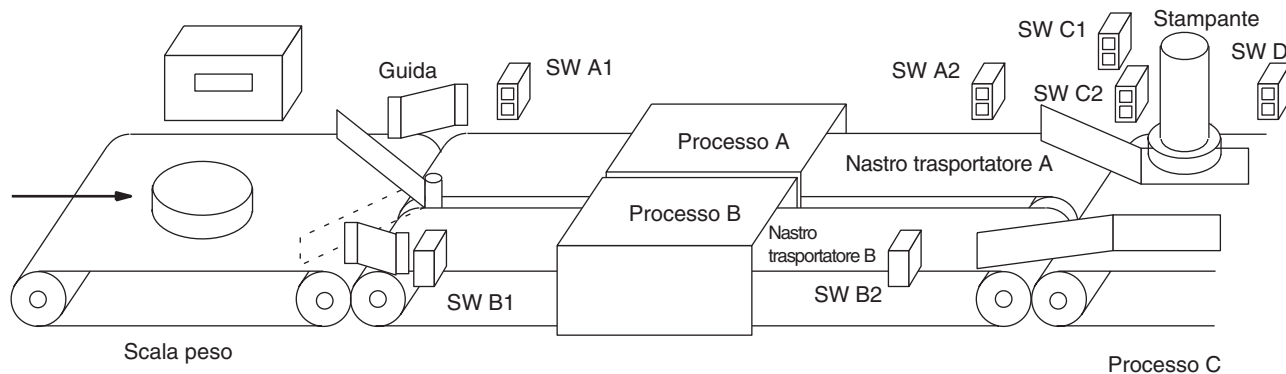
Il programma per questo processo, illustrato di seguito, utilizza il tipo base di programmazione step: ogni step viene completato da un'istruzione SNXT(009) univoca che avvia lo step successivo. Ogni step inizia quando viene attivato il selettore che indica che lo step precedente è stato completato.



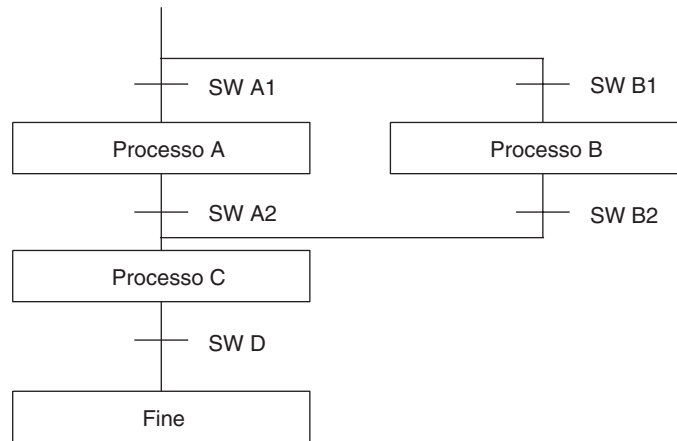
| Indirizzo | Istruzione | Operandi |
|------------|------------|----------|
| 000000 | @LD | 000001 |
| 000001 | SNXT(009) | W00000 |
| 000002 | STEP(008) | W00000 |
| Processo A | | |
| 000100 | LD | 000002 |
| 000101 | SNXT(009) | W00001 |
| 000102 | STEP(008) | W00001 |
| Processo B | | |
| 000100 | LD | 000003 |
| 000101 | SNXT(009) | W00002 |
| 000102 | STEP(008) | W00002 |
| Processo C | | |
| 000200 | LD | 000004 |
| 000201 | SNXT(009) | W00003 |
| 000202 | STEP(008) | W00003 |

**Esempio 2:
Esecuzione di
diramazione**

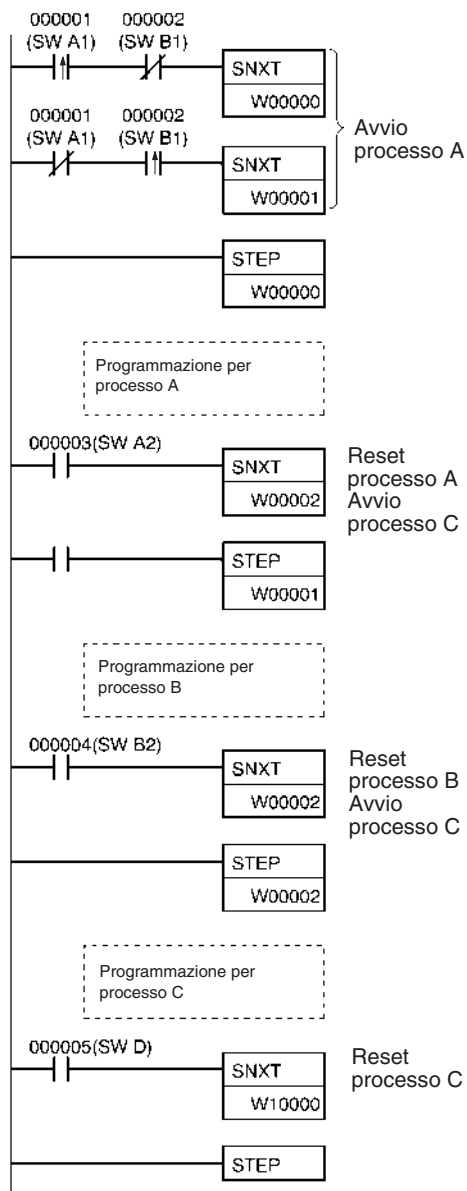
Il processo seguente richiede che un prodotto venga elaborato in uno di due modi, in base al peso, prima di essere stampato. Il processo di stampa è lo stesso indipendentemente dal primo processo utilizzato. Vari sensori sono posizionati sui segnali dove i processi devono iniziare e terminare.



Nel diagramma seguente vengono mostrati il flusso di elaborazione e i selettori utilizzati per il controllo dell'esecuzione. Qui viene utilizzato il processo A o il processo B, in base allo stato di SW A1 e di SW B1.



Il programma per questo processo, illustrato di seguito, inizia con due istruzioni SNXT(009) che avviano i processi A e B. In seguito al modo in cui CIO 000001 (SW A1) e CIO 000002 (SW B1) sono stati programmati, soltanto uno di questi verrà eseguito con una condizione di esecuzione ON per avviare il processo A o il processo B. Entrambi gli step per questi processi terminano con un'istruzione SNXT(009) che avvia lo step (processo C).

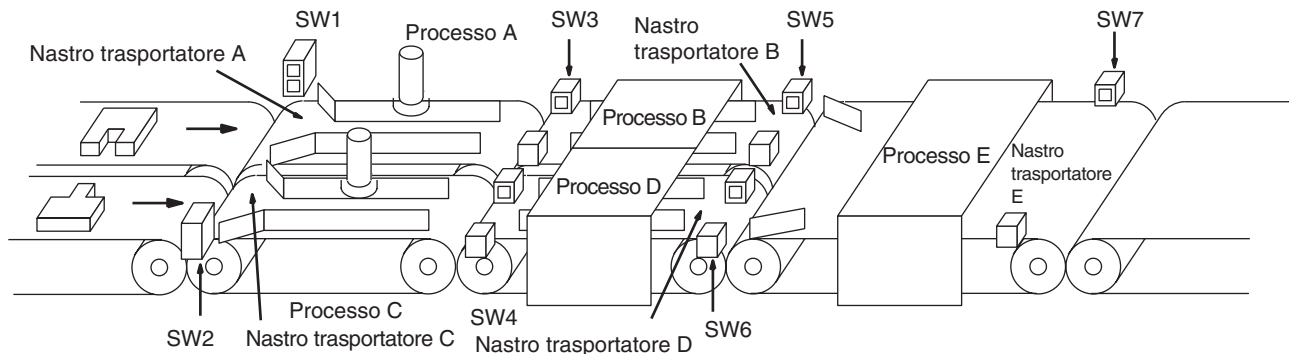


| Indirizzo | Istruzione | Operandi |
|------------|------------|----------|
| 000000 | @LD | 000001 |
| 000001 | AND NOT | 000002 |
| 000002 | SNXT(009) | 010000 |
| 000003 | LD NOT | 000001 |
| 000004 | @AND | 000002 |
| 000005 | SNXT(009) | 010001 |
| 000006 | STEP(008) | 010000 |
| Processo A | | |
| 000100 | LD | 000003 |
| 000101 | SNXT(009) | 010002 |
| 000102 | STEP(008) | 010001 |
| Processo B | | |
| 000100 | LD | 000004 |
| 000101 | SNXT(009) | 010002 |
| 000102 | STEP(008) | 010002 |
| Processo C | | |
| 000200 | LD | 000005 |
| 000201 | SNXT(009) | 024614 |
| 000202 | STEP(008) | --- |

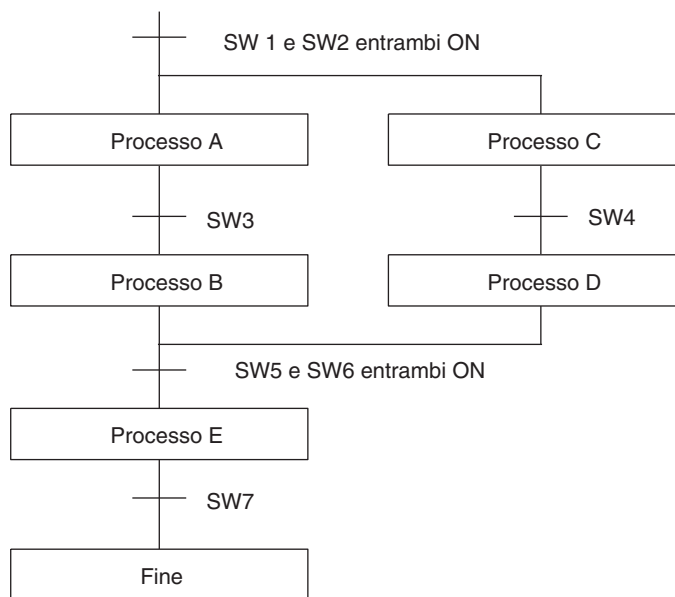
Nota Nella programmazione precedente, CIO 010002 viene utilizzato in due istruzioni SNXT(009). Ciò non produce un errore di duplicazione durante il controllo del programma.

Esempio 3:
Esecuzione parallela

Il seguente processo richiede che due parti di un prodotto passino simultaneamente attraverso due processi ciascuna, prima di essere riunite insieme in un quinto processo. Vari sensori sono posizionati sui segnali dove i processi devono iniziare e terminare.

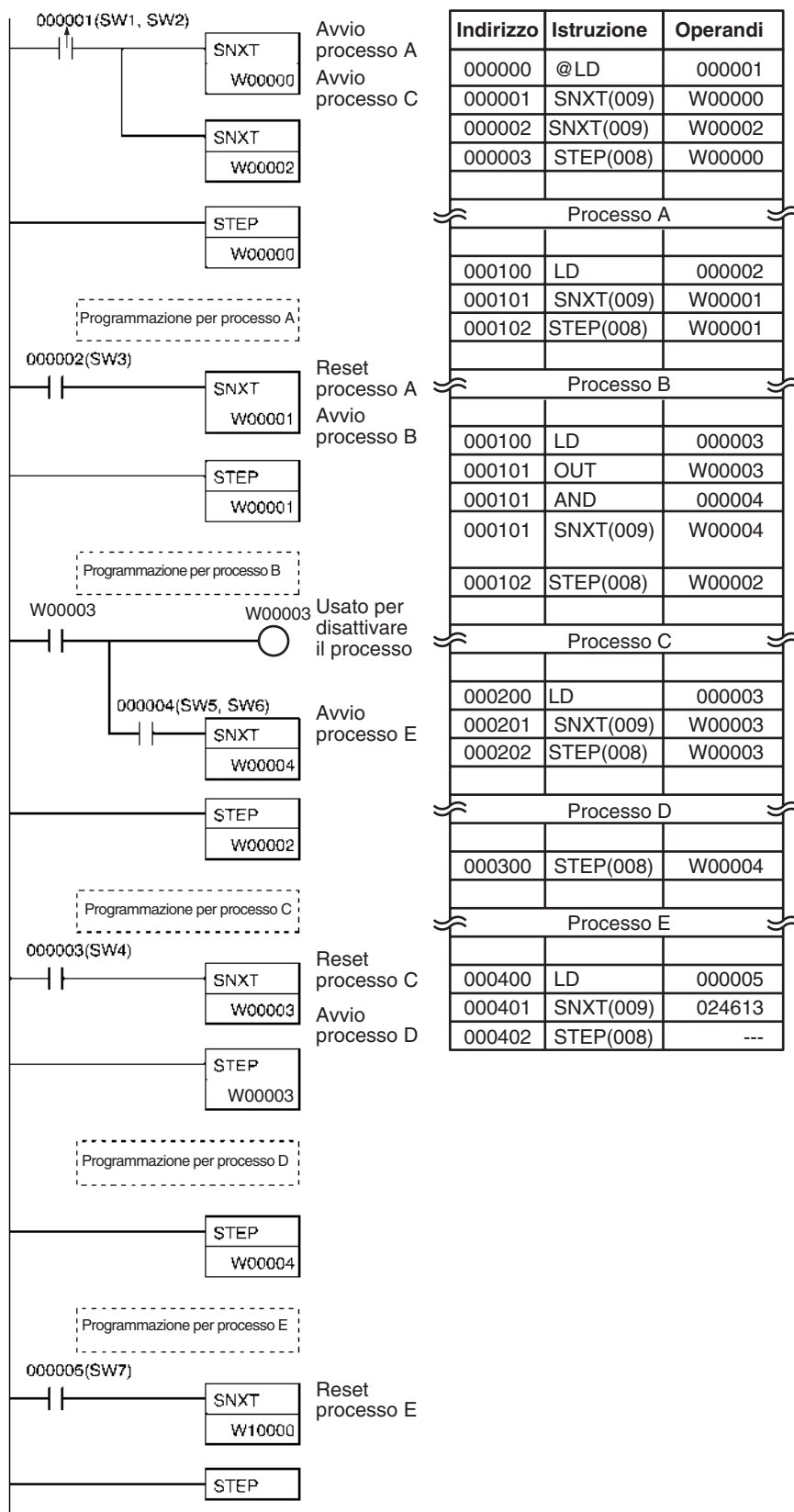


Nel diagramma seguente vengono mostrati il flusso di elaborazione e i selettori utilizzati per il controllo dell'esecuzione. In questo caso, il processo A e il processo C vengono avviati insieme. Al termine del processo A viene avviato il processo B; al termine del processo C viene avviato il processo D. Quando entrambi i processi B e D sono terminati, si avvia il processo E.



Il programma per questa operazione, illustrato di seguito, inizia con due istruzioni SNXT(009) che avviano i processi A e C. Queste istruzioni derivano dalla stessa riga di istruzione e vengono sempre eseguite insieme, avviando gli step sia per A che per C. Quando entrambi gli step A e C sono terminati, si avviano immediatamente gli step per i processi B e D.

Quando entrambi i processi B e D sono terminati, ovvero SW5 e SW6 sono impostati su ON, i processi B e D vengono reimpostati insieme dall'istruzione SNXT(009) al termine della programmazione per il processo B. Nonostante non vi sia alcuna istruzione SNXT(009) alla fine del processo D, il rispettivo bit di controllo viene disattivato dall'esecuzione di SNXT(009) W00004. Questo perché l'istruzione OUT per il bit W00003 si trova nello step reimpostato da SNXT(009) W00004, ossia W00003 viene disattivato (OFF) quando viene eseguita SNXT(009) W00004. Quindi il processo B viene reimpostato direttamente e il processo D viene reimpostato indirettamente prima dell'esecuzione dello step per il processo E.



3-23 Istruzioni per moduli di I/O di base

In questa sezione vengono descritte le istruzioni utilizzate per i moduli di I/O.

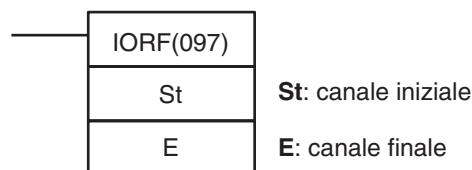
| Istruzione | Mnemonico | Codice funzione | Pagina |
|--------------------------|-----------|-----------------|--------|
| I/O REFRESH | IORF | 097 | 885 |
| 7-SEGMENT DECODER | SDEC | 078 | 888 |
| INTELLIGENT I/O READ | IORF | 222 | 913 |
| INTELLIGENT I/O WRITE | IOWR | 223 | 917 |
| DIGITAL SWITCH INPUT | DSW | 210 | 890 |
| TEN KEY INPUT | TKY | 211 | 896 |
| HEXADECIMAL KEY INPUT | HKY | 212 | 899 |
| MATRIX INPUT | MTR | 213 | 904 |
| 7-SEGMENT DISPLAY OUTPUT | 7SEG | 214 | 908 |

3-23-1 I/O REFRESH: IORF(097)

Scopo

Aggiorna i canali di I/O specificati.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | IORF(097) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @IORF(097) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

St: canale iniziale

CIO 0000 a CIO 0999 (area dei bit di I/O) o Da

CIO 2000 a CIO 2959 (area dei bit del modulo di I/O speciale)

E: canale finale

CIO 0000 a CIO 0999 (area dei bit di I/O) o Da

CIO 2000 a CIO 2959 (area dei bit del modulo di I/O speciale)

Nota St e E devono essere nella stessa area di memoria.

Caratteristiche operando

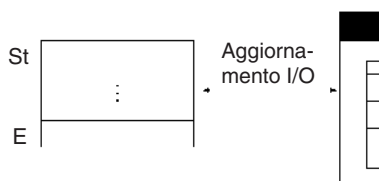
| Area | St | E |
|-------------------------|--|---|
| Area CIO | Da CIO 0000 a CIO 0999 Da CIO 2000 a CIO 2959 | |
| Area ausiliaria | --- | |
| Area bit di ritenività | --- | |
| Area bit speciale | --- | |
| Area del temporizzatore | --- | |
| Area del contatore | --- | |
| Area DM | --- | |

| Area | St | E |
|--|---|---|
| Area EM senza banco | --- | |
| Area EM con banco | --- | |
| Indirizzi DM/EM indiretti in binario | --- | |
| Indirizzi DM/EM indiretti in BCD | --- | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a IR15 Da -2048 a +2047, da IR0 a IR15, Da DR0 a DR15, da IR0 a IR15, Da IR0 a IR15+(++) Da ,-(-)IR0 a IR15 | |

Descrizione

L'istruzione IORF(097) aggiorna in modo globale i canali di I/O tra St e E. IORF(097) viene utilizzata per aggiornare i canali assegnati ai moduli di I/O di base e ai moduli di I/O speciali montati sul rack CPU o sui rack di espansione. Non è possibile utilizzare l'istruzione IORF(097) per aggiornare contemporaneamente i canali in entrambe le aree, ossia con la stessa istruzione. Ai moduli di I/O di base sono assegnati i canali compresi tra CIO 0000 e CIO 0999, mentre ai moduli di I/O speciali sono assegnati i canali compresi tra CIO 2000 e CIO 2959. Quando è specificato l'aggiornamento per i canali nell'area dei bit del modulo di I/O speciale, tutti i 10 canali assegnati al modulo verranno aggiornati fin tanto che il primo canale dei 10 canali assegnati al modulo è incluso nell'intervallo di canali specificato.

Area dei bit del modulo di I/O Modulo di I/O o
o del modulo di I/O speciale modulo di I/O speciale



Se tra St ed E vi sono dei canali per i quali non vi è alcun modulo montato, non verrà eseguito alcun aggiornamento e verranno aggiornati soltanto i canali assegnati ai moduli.

È possibile aggiornare con la stessa istruzione sia i moduli di I/O speciali C200H, sia i moduli di I/O speciali CS (solo serie CS).

Tutti i canali assegnati ai moduli di I/O ad alta densità C200H gruppo 2 devono essere aggiornati contemporaneamente. I canali di I/O del modulo verranno aggiornati se il primo canale assegnato al modulo è incluso nell'intervallo dei canali di I/O specificato. I canali del modulo non verranno aggiornati se il canale iniziale si trova dopo il primo canale assegnato al modulo. Tuttavia, essi verranno aggiornati anche quando il canale finale è prima dell'ultimo canale assegnato al modulo (solo serie CS).

È possibile utilizzare l'istruzione IORF(097) nei task ad interrupt, consentendo tempi di risposta rapidi per i canali di I/O specifici aggiornati nel task ad interrupt (vedere Avvertenze).

Moduli applicabili

Tramite l'istruzione IORF(097) è possibile aggiornare i moduli indicati di seguito, ma solo se montati su rack CPU o su rack di espansione, non su sistemi slave.

Moduli di I/O di base della serie CS, moduli di I/O di base C200H (solo serie CS), moduli di I/O ad alta densità C200H gruppo 2 (solo serie CS), moduli di I/O di base della serie CJ e moduli di I/O speciali (inclusi moduli ad alta densità). Tutti i canali assegnati ai moduli possono essere aggiornati.

Nota I moduli che possono essere aggiornati tramite l'istruzione IORF(097) non sono necessariamente gli stessi moduli che possono essere aggiornati tramite aggiornamento immediato (!).

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se St è maggiore di E. ON se St ed E si trovano in aree di memoria diverse. Con le CPU CS1D: ON se non è stato possibile sincronizzare le CPU attiva e in attesa. OFF in tutti gli altri casi. |

Avvertenze

Si verificherà un errore se per la stessa istruzione vengono specificati sia i canali dell'area dei bit di I/O (da CIO 0000 a CIO 0999) che i canali dell'area dei bit del modulo di I/O speciale (da CIO 2000 a CIO 2959).

L'aggiornamento I/O non verrà eseguito per i moduli nei quali si è verificato un errore della tabella I/O (solo serie CS).

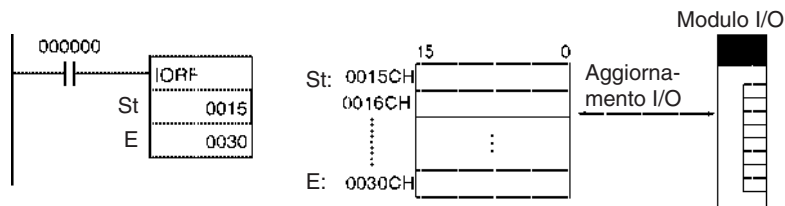
L'aggiornamento I/O avviato da IORF(097) verrà interrotto a metà se si verifica un errore del bus di I/O durante l'aggiornamento I/O.

Quando IORF(097) viene utilizzata per un task ad interrupt, controllare che nella configurazione del PLC sia stato disabilitato l'aggiornamento ciclico per i moduli di I/O speciali. Se l'aggiornamento ciclico per i moduli di I/O speciali è abilitato e si esegue di nuovo un aggiornamento I/O tramite l'istruzione IORF(097), verrà generato un errore non fatale di aggiornamento duplicato e verrà attivato il flag di errore task ad interrupt (A40213).

Esempi

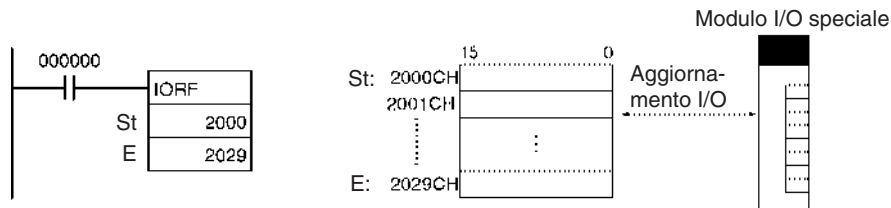
Aggiornamento dei canali nell'area dei bit di I/O

Nell'esempio seguente viene mostrato come aggiornare 16 canali da CIO 0015 a CIO 0030 quando CIO 000000 viene impostato su ON.



Aggiornamento dei canali nell'area dei bit del modulo di I/O speciale

Nell'esempio seguente viene mostrato come aggiornare 30 canali da CIO 2000 a CIO 2029 quando CIO 000000 viene impostato su ON.

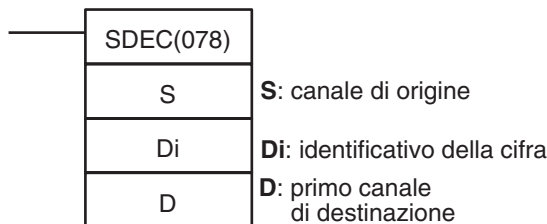


3-23-2 7-SEGMENT DECODER SDEC(078)

Scopo

Converte il contenuto esadecimale delle cifre specificate in codici display a 7 segmenti e li memorizza negli 8 bit superiori o inferiori dei canali di destinazione specificati.

Simbolo programmazione ladder



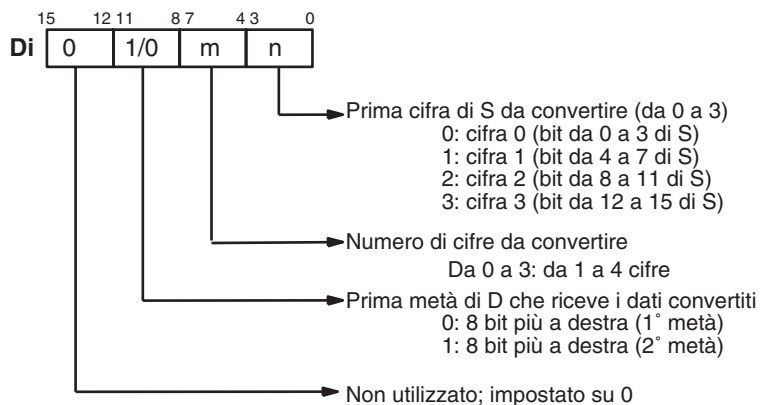
Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SDEC(078) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SDEC(078) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi: identificativo della cifra



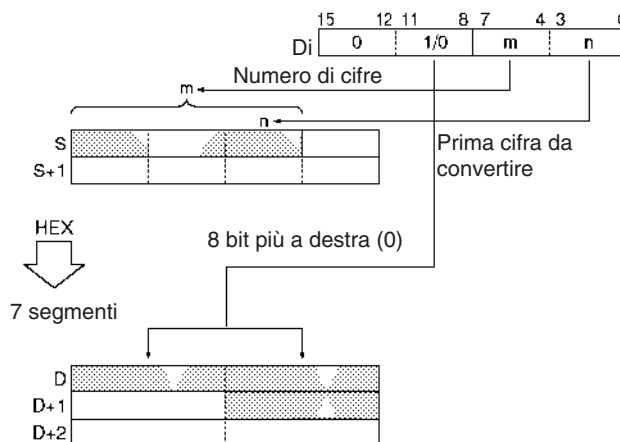
Caratteristiche operando

| Area | S | Di | D |
|-------------------------|--|----|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 a A959 | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |

| Area | S | Di | D |
|--|--|-------------------------|-----|
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | --- |
| Registri dati | Da DR0 a DR15 | | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SDEC(078) considera i dati specificati da S come dati esadecimali a 4 cifre, converte le cifre specificate in S da Di (prima cifra e il numero di cifre) in dati a 7 segmenti e invia i risultati a D nei bit specificati in Di.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se le impostazioni in Di non sono incluse negli intervalli specificati. OFF in tutti gli altri casi. |

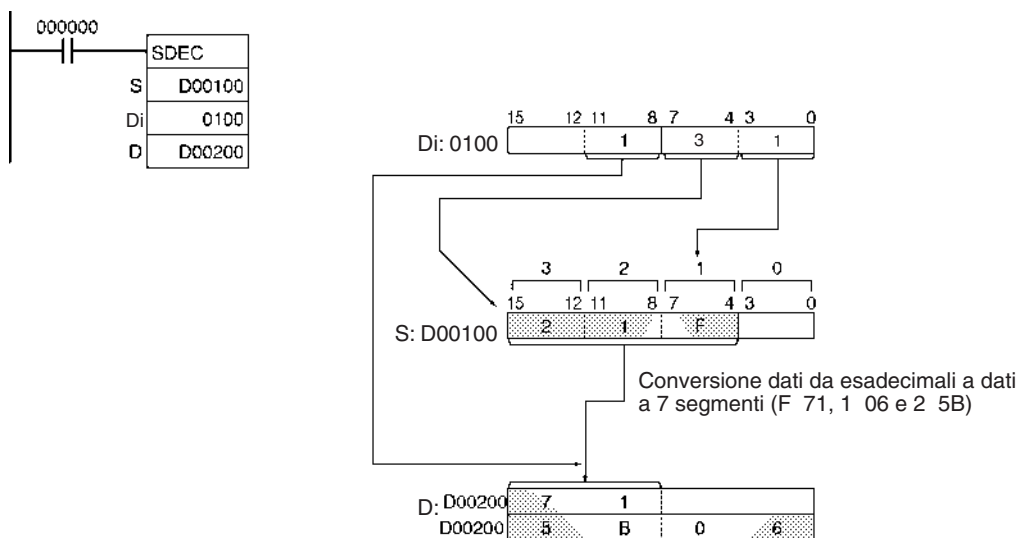
Avvertenze

Se per la conversione in Di sono state specificate più cifre, le cifre vengono convertite in ordine verso la cifra più significativa. La cifra 0 è la cifra che segue la cifra 3.

I risultati vengono memorizzati in D in ordine, partendo dalla parte specificata verso i canali con indirizzo più alto. Se soltanto uno dei byte in un canale di destinazione riceve i dati convertiti, l'altro byte resta invariato.

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i contenuti delle 3 cifre verranno convertiti da dati esadecimali a dati a 7 segmenti, cominciando dalla cifra 1 in D00100, e il risultato verrà inviato al byte superiore di D00200 e a entrambi i byte di D00201. Le specifiche dei byte da convertire e la posizione dei byte di uscita vengono definite in CIO 0100.



Dati a 7 segmenti

Nella tabella seguente sono riportate le conversioni dei dati da una cifra esadecimale (4 bit) a un codice a 7 segmenti (8 bit).

| Dati originali | | Codice convertito (segmenti) | | | | | | | | | | Visualizzazione dati originali | |
|----------------|---------|------------------------------|---|---|---|---|---|---|---|-----|---|--------------------------------|--|
| Cifra | Bit | - | g | f | e | d | c | b | a | Esa | | | |
| 0 | 0 0 0 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3F | 0 | | |
| 1 | 0 0 0 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 06 | 1 | | |
| 2 | 0 0 1 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5B | 2 | | |
| 3 | 0 0 1 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 4F | 3 | | |
| 4 | 0 1 0 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 66 | 4 | | |
| 5 | 0 1 0 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 6D | 5 | | |
| 6 | 0 1 1 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 7D | 6 | | |
| 7 | 0 1 1 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 27 | 7 | | |
| 8 | 1 0 0 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7F | 8 | | |
| 9 | 1 0 0 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 6F | 9 | | |
| A | 1 0 1 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 77 | A | | |
| B | 1 0 1 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 7C | b | | |
| C | 1 1 0 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 39 | c | | |
| D | 1 1 0 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 5E | d | | |
| E | 1 1 1 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 79 | e | | |
| F | 1 1 1 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 71 | F | | |

LSB

→

a

→

b

→

c

→

d

→

e

→

f

→

g

→

0

MSB

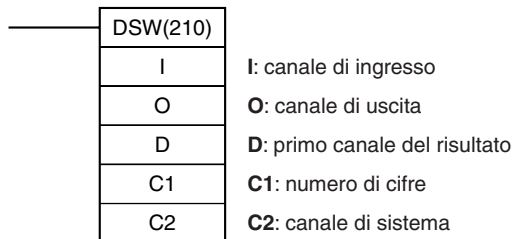
3-23-3 DIGITAL SWITCH INPUT – DSW(210)

Scopo

Legge il valore impostato su un selettore digitale esterno (o selettore rotativo) collegato a un modulo di I/O e memorizza il valore a 4 o 8 cifre nei canali specificati.

Questa istruzione è supportata solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | DSW(210) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

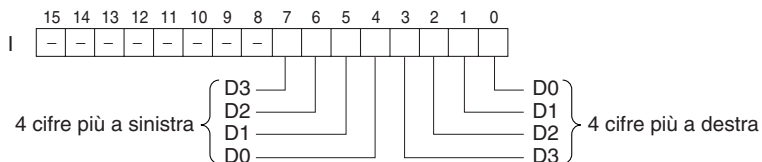
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

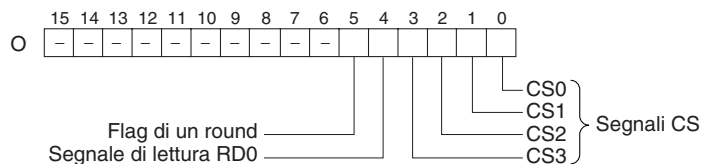
I: canale di ingresso (ingressi da D0 a D3 della riga dati)

Specificare il canale di ingresso assegnato al modulo di ingresso e collegare le righe dati da D0 a D3 del selettore digitale al modulo di ingresso come illustrato nel diagramma seguente.



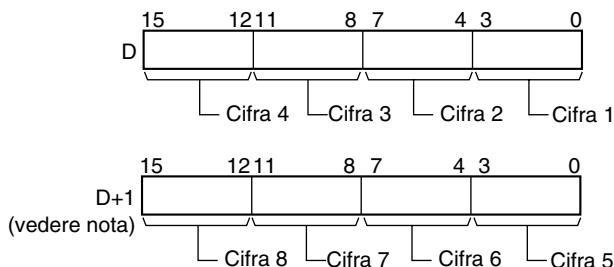
O: canale di uscita (uscite del segnale di controllo CS/RD)

Specificare il canale di uscita assegnato al modulo di uscita e collegare i segnali di controllo (segnali CS ed RD) del selettore digitale al modulo di uscita come illustrato nel diagramma seguente.



D: primo canale del risultato

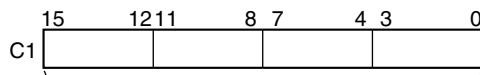
Specifica l'indirizzo del canale iniziale in cui saranno memorizzati i valori impostati per il selettore digitale esterno.



Nota: solo se C1 = 0001 esadecimale per leggere 8 cifre.

C1: numero di cifre

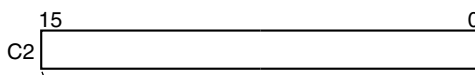
Specifica il numero di cifre che saranno lette dal selettore digitale esterno. Per leggere 4 cifre, impostare C1 su 0000 esadecimale e per leggerne 8, impostarlo su 0001 esadecimale.



Numero di cifre
0000 esadecimale: 4 cifre
0001 esadecimale: 8 cifre

C2: canale di sistema

Specifica un canale di lavoro utilizzato dall'istruzione. Questo canale non può venire utilizzato in altre applicazioni.



Canale di sistema
(l'utente non può averne accesso)

Caratteristiche operando

| Area | I | O | D | C1 | C2 |
|--------------------------------------|--|-----------------|---|-------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | | | --- | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | | | --- | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A953 | | --- | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | | --- | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | | | --- | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | | | --- | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | | | --- | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | --- | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | --- | --- |
| Costanti | --- | | | 0000 o 0001 esadecimale | --- |
| Registri dati | Da DR0 a DR15 | | | | Da DR0 a DR15 |

| Area | I | O | D | C1 | C2 |
|--|---|---|---|----|--|
| Registri indice | --- | | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

DSW(210) invia segnali di controllo ai bit da 00 a 04 di O, legge da I il numero di cifre specificato (4 o 8 cifre, indicato in C1) dei dati della riga dati del selettore digitale e memorizza il risultato in D e D+1. Se vengono lette 4 cifre, il risultato è memorizzato in D, mentre se ne vengono lette 8 il risultato sarà memorizzato in D e D+1.

DSW(210) legge i dati a 4 od 8 cifre del selettore digitale una volta ogni 16 cicli, quindi ricomincia e continua a leggere i dati. Il flag di un round (bit da 05 a O) viene attivato una volta ogni 16 cicli della CPU.

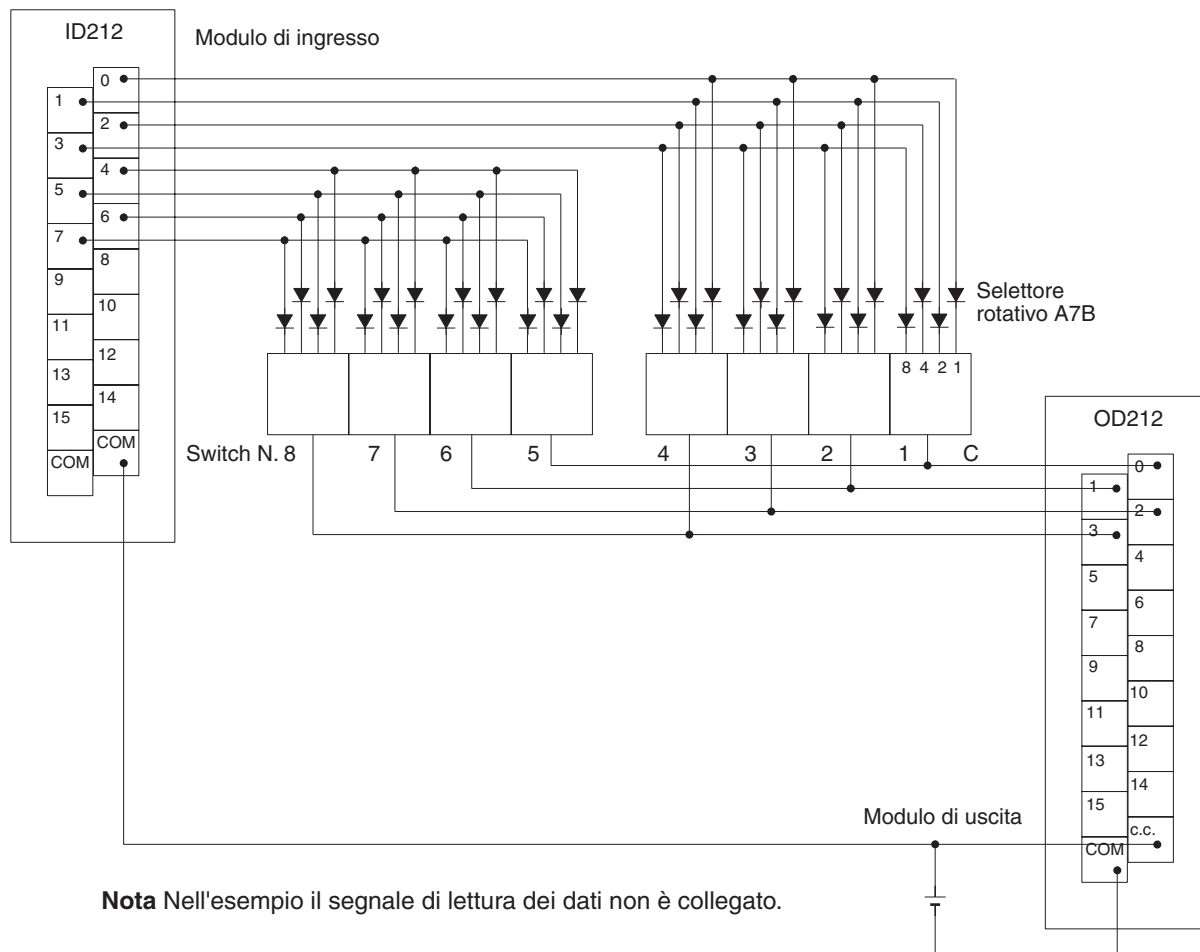
DSW(210) legge i dati a 4 od 8 cifre una volta ogni 16 cicli, quindi ricomincia e rilegge i dati nei successivi 16 cicli.

Quando viene eseguita, DSW(210) inizia a leggere i dati del selettore dal primo dei sedici cicli, indipendentemente dal punto in cui era stata interrotta l'ultima istruzione.

A differenza delle serie C200HX/HG/HE e CQM1H, il numero di volte in cui è possibile specificare l'istruzione DSW(210) nel programma è illimitato.

Collegamenti esterni

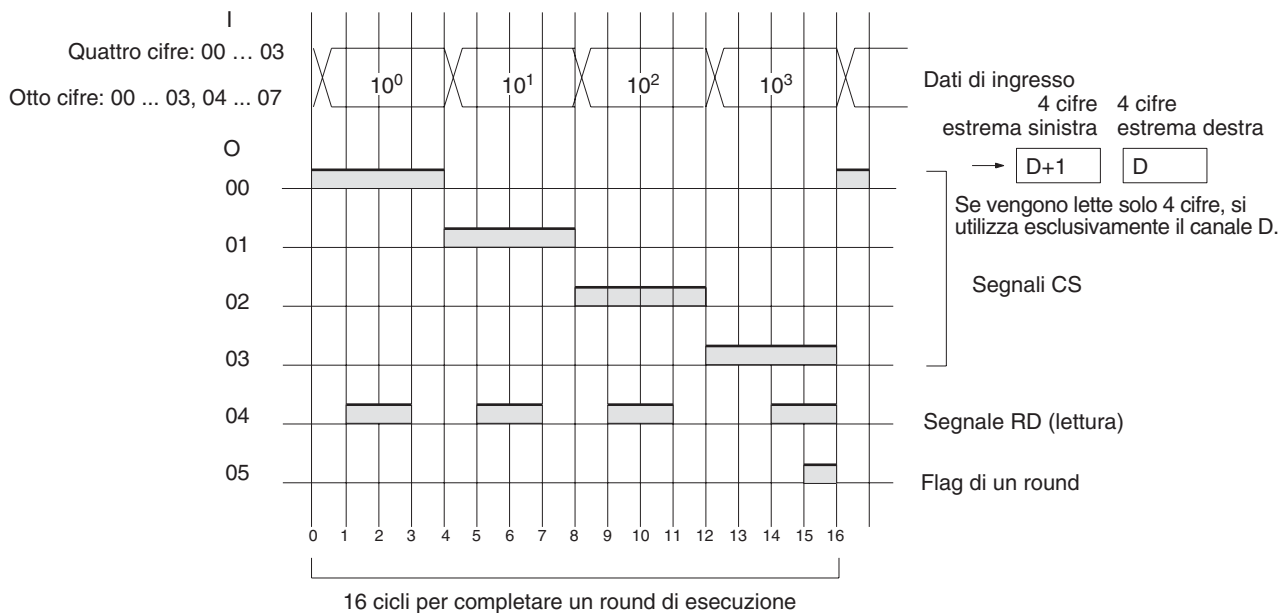
Come illustrato nel seguente diagramma, collegare il selettore digitale o rotativo ai contatti del modulo di ingresso compresi tra 0 e 7 e ai contatti del modulo di uscita compresi tra 0 e 4. L'esempio indica i collegamenti per un selettore rotativo A7B.



Gli ingressi e le uscite possono essere collegati ai seguenti tipi di moduli di I/O di base e moduli di I/O ad alta densità, purché non siano installati su un sistema slave di I/O remoti SYSMAC BUS.

- Moduli di ingresso c.c. con 8 o più punti di ingresso
- Moduli di uscita a transistor con 8 o più punti di uscita

Diagramma di funzionamento



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---------------|
| Flag di errore | ER | OFF |

Avvertenze

Non leggere o scrivere il canale di sistema C2 da altre istruzioni. Se il canale di sistema viene utilizzato da altre istruzioni, il funzionamento di DSW(210) non sarà corretto. All'avvio dell'esecuzione del programma, il canale di sistema non viene inizializzato da DSW(210) durante il primo ciclo. Se DSW(210) viene utilizzata dal primo ciclo, eliminare il canale di sistema dal programma.

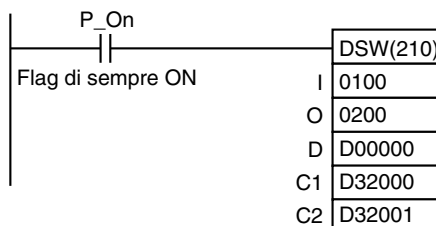
Se al termine dell'esecuzione di DSW(210) non viene eseguito il refresh I/O con i moduli di ingresso e di uscita collegati al selettore digitale o rotativo, il funzionamento dell'istruzione non sarà corretto. Pertanto non collegare il selettore digitale o rotativo ai seguenti moduli.

- Moduli di I/O di base o moduli di I/O ad alta densità installati su un sistema slave di I/O remoti SYSMAC BUS
- Slave per le comunicazioni (slave DeviceNet o CompoBus/S)

Esempio

Nell'esempio, DSW(210) viene utilizzata per leggere un numero di 8 cifre da un selettore digitale e inviare il valore risultante in D00000 e D00001 in modo costante. Il selettore digitale è collegato tramite CIO 0100 (assegnato a un modulo di ingresso c.c. a 16 punti CS1W-ID211) e CIO 0200 (assegnato a un modulo di uscita a transistor a 16 punti CS1W-OD211).

Poiché vengono lette 8 cifre di dati, C1 (in questo caso, D32000) è impostato su 0001 esadecimale. D32001 viene utilizzato come canale di sistema.



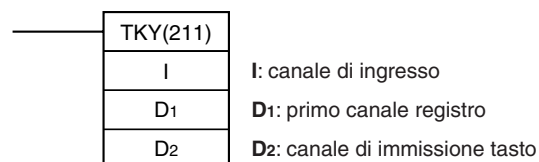
3-23-4 TEN KEY INPUT – TKY(211)

Scopo

Legge i dati numerici da un tastierino a dieci tasti collegato a un modulo di ingresso e memorizza fino a 8 cifre di dati in formato BCD nei canali specificati.

Questa istruzione è supportata solo dalla CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TKY(211) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TKY(211) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

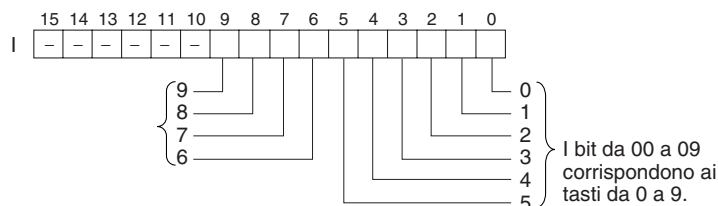
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

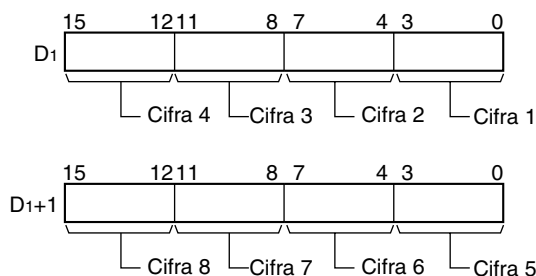
I: canale di ingresso (ingressi della riga dati)

Specificare il canale di ingresso assegnato al modulo di ingresso e collegare le righe dati da 0 a 9 del tastierino a dieci tasti al modulo di ingresso come illustrato nel diagramma seguente.



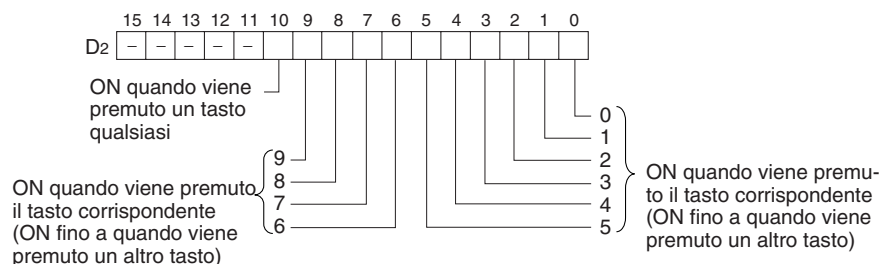
D₁: primo canale registro

Specifica l'indirizzo del canale iniziale in cui sarà memorizzato l'ingresso numerico (fino a 8 cifre) del tastierino a dieci tasti.



D₂: canale di immissione tasto

I bit da 00 a 10 di D₂ indicano le immissioni tasto. Quando viene premuto uno dei tasti del tastierino (0 ... 9), il bit corrispondente di D₂ (0 ... 9) viene impostato su ON. Il bit 10 di D₂ è impostato su ON se viene premuto un tasto qualsiasi.



Nota A differenza di altre istruzioni di I/O come HKY(212), TKY(211) non necessita di un canale di sistema.

Caratteristiche operando

| Area | I | D ₁ | D ₂ |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A958 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

TKY(211) legge i dati numerici dal canale di ingresso I, che è assegnato a un tastierino a dieci tasti collegato a un modulo di ingresso, e memorizza fino a 8 cifre di dati in formato BCD nei canali registro D₁ e D₁+1. Inoltre, ogni volta che viene premuto un tasto, il bit corrispondente in D₂ (0 ... 9) passa a ON e resta attivato fino a quando non viene premuto un altro tasto. Il bit 10 di D₂ è impostato su ON se viene premuto un tasto qualsiasi e su OFF se non viene premuto alcuno.

Il registro a doppio canale in D_1 e D_1+1 funziona analogamente a un registro di shift a 8 cifre. Quando viene premuto un tasto sul tastierino a dieci tasti, la cifra BCD corrispondente viene fatta scorrere nella cifra meno significativa di D_1 . Le altre cifre di D_1 , D_1+1 vengono fatte scorrere a sinistra e la cifra più significativa di D_1+1 è persa.

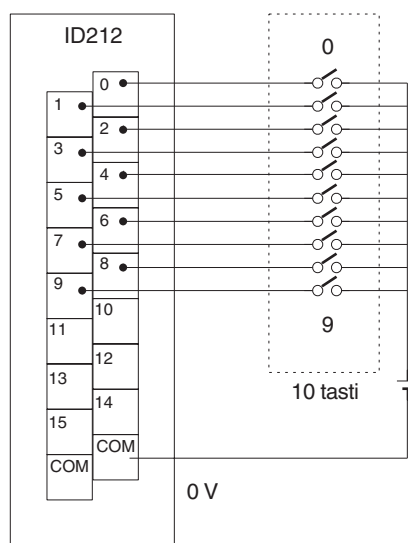
Quando viene eseguita, TKY (211) inizia a leggere i dati di immissione tasto dal primo ciclo, indipendentemente dal punto in cui era stata interrotta l'ultima istruzione.

Quando viene premuto uno dei tasti del tastierino, l'immissione dagli altri tasti è disabilitata.

A differenza delle serie C200HX/HG/HE e CQM1H, il numero di volte in cui è possibile specificare l'istruzione TKY(211) nel programma è illimitato.

Collegamenti esterni

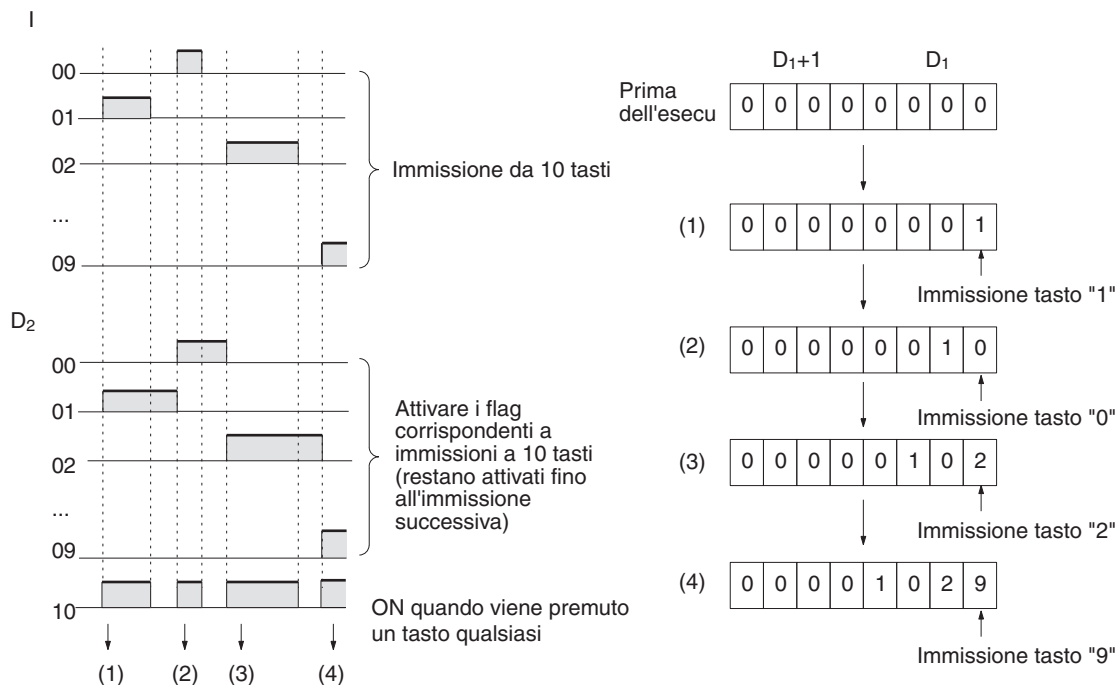
Collegare il tastierino a dieci tasti in modo tale che i selettori per i tasti da 0 a 9 siano collegati ai contatti da 0 a 9 del modulo di ingresso come illustrato nel diagramma che segue.



Modulo di ingresso c.c.

È necessario che il modulo di ingresso sia c.c. o ad alta densità con almeno 16 ingressi e che non sia installato su un sistema slave di I/O remoti SYSMAC BUS.

Diagramma di funzionamento



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---------------|
| Flag di errore | ER | OFF |

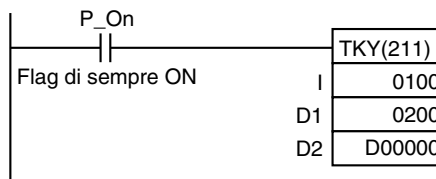
Avvertenze

Se al termine dell'esecuzione di TKY(211) non viene eseguito il refresh I/O con il modulo di ingresso collegato al tastierino a 10 tasti, il funzionamento dell'istruzione non sarà corretto. Pertanto non collegare il tastierino a 10 tasti ai seguenti moduli.

- Moduli di I/O di base o moduli di I/O ad alta densità installati su un sistema slave di I/O remoti SYSMAC BUS
- Slave per le comunicazioni (slave DeviceNet o CompoBus/S)

Esempio

In questo esempio, TKY(211) legge le immissioni tasto da un tastierino a dieci tasti e le memorizza in D00000 e D00001. Il tastierino a dieci tasti è collegato a CIO 0100 (assegnato a un modulo di ingresso c.c. a 16 punti CS1W-ID211).



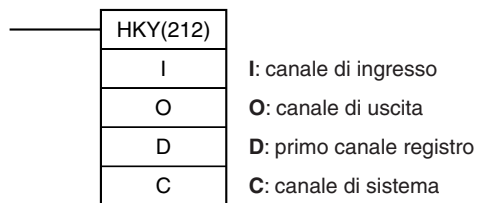
3-23-5 HEXADECIMAL KEY INPUT – HKY(212)

Scopo

Legge i dati numerici da un tastierino esadecimale collegato a un modulo di ingresso e a un modulo di uscita e memorizza fino a 8 cifre di dati in formato esadecimale nei canali specificati.

Questa istruzione è supportata solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | HKY(212) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

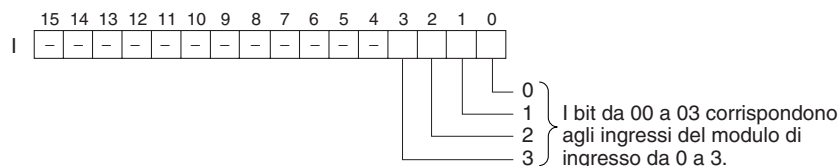
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

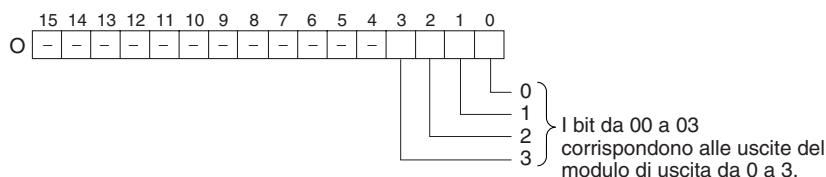
I: canale di ingresso (ingressi da D0 a D3 della riga dati)

Specificare il canale di ingresso assegnato al modulo di ingresso e collegare le righe dati da D0 a D3 del tastierino esadecimale al modulo di ingresso come illustrato nel diagramma seguente.



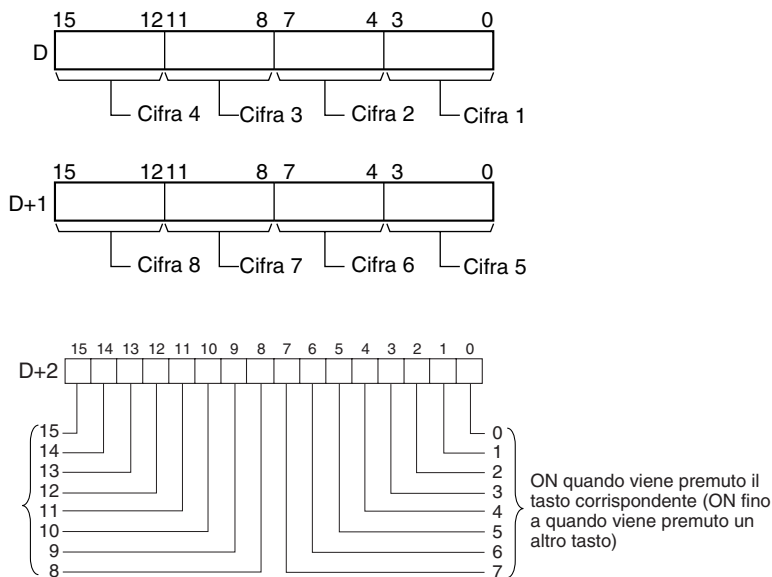
O: canale di uscita (uscite del segnale di selezione)

Specificare il canale di uscita assegnato al modulo di uscita e collegare i segnali di selezione del tastierino esadecimale al modulo di uscita come illustrato nel diagramma seguente.



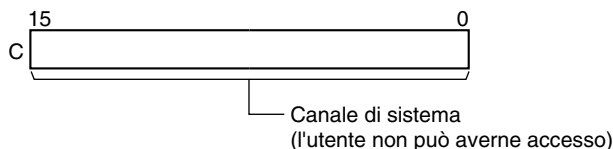
D: primo canale registro

Specifica l'indirizzo del canale iniziale in cui sarà memorizzato l'ingresso numerico (fino a 8 cifre) del tastierino esadecimale. Inoltre, ogni volta che viene premuto un tasto, viene attivato il bit corrispondente in D+2 (da 0 a F) che mantiene l'impostazione fino a quando non viene premuto un altro tasto.



C: canale di sistema

Specifica un canale di lavoro utilizzato dall'istruzione. Questo canale non può venire utilizzato in altre applicazioni.



Caratteristiche operando

| Area | I | O | D | C |
|--------------------------------------|--|-----------------|---|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6141 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W509 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H509 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A957 | Da A448 ad A959 | Da A448 ad A957 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4093 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4093 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32765 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32765 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | En_00000 ... En_32765 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | |

| Area | I | O | D | C |
|--|--|-----|---|---------------|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | |
| Costanti | --- | | | |
| Registri dati | Da DR0 a DR15 | --- | | Da DR0 a DR15 |
| Registri indice | --- | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 | | | |

Descrizione

HKY(212) invia i segnali di selezione ai bit da 00 a 03 di O, legge i dati per ordine dai bit compresi tra 00 e 03 di I e memorizza fino a 8 cifre dei dati esadecimali nei canali registro D e D+1.

HKY(212) immette ogni cifra in un numero di cicli compreso tra 3 e 12, quindi ricomincia e procede con l'immissione. Inoltre, ogni volta che viene premuto un tasto, viene attivato il bit corrispondente in D+2 (da 0 a F) che mantiene l'impostazione fino a quando non viene premuto un altro tasto.

HKY(212) determina il tasto premuto identificando l'ingresso attivo nel momento in cui è attivato un determinato segnale di selezione; di conseguenza possono essere comunque necessari da 3 a 12 cicli per leggere una cifra esadecimale. Una volta letta l'immissione tasti, HKY(212) ricomincia e legge un'altra cifra nei cicli successivi (da 3 a 12).

Quando viene eseguita, HKY (212) inizia a leggere i dati di immissione tasto dal primo segnale di selezione, indipendentemente dal punto in cui era stata interrotta l'ultima istruzione.

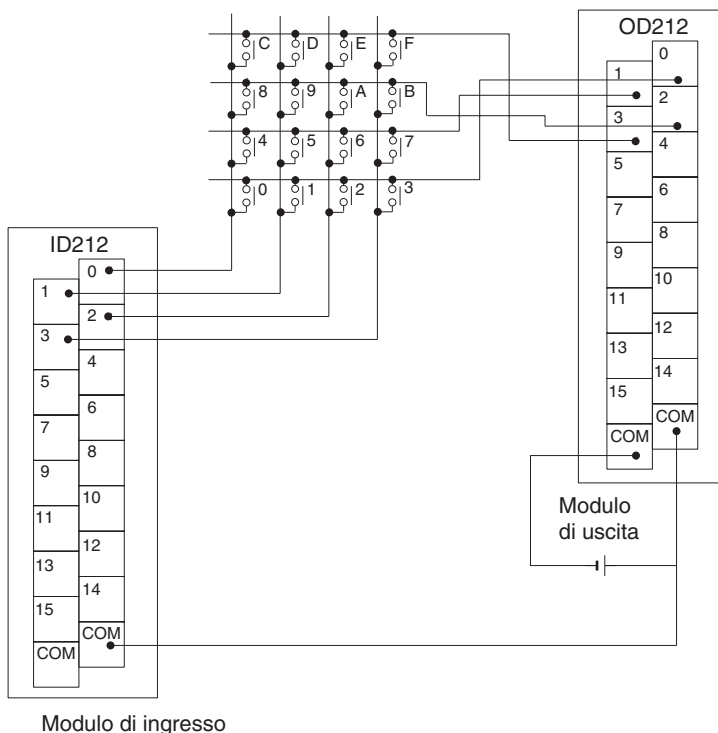
Il registro a doppio canale in D_1 e D_1+1 funziona analogamente a un registro di shift a 8 cifre. Quando viene premuto un tasto sul tastierino a dieci tasti, la cifra esadecimale corrispondente viene fatta scorrere nella cifra meno significativa di D_1 . Le altre cifre di D_1 , D_1+1 vengono fatte scorrere a sinistra e la cifra più significativa di D_1+1 è persa.

Quando viene premuto uno dei tasti del tastierino, l'immissione dagli altri tasti è disabilitata.

A differenza delle serie CQM1H, il numero di volte in cui è possibile specificare l'istruzione HKY(212) nel programma è illimitato.

Collegamenti esterni

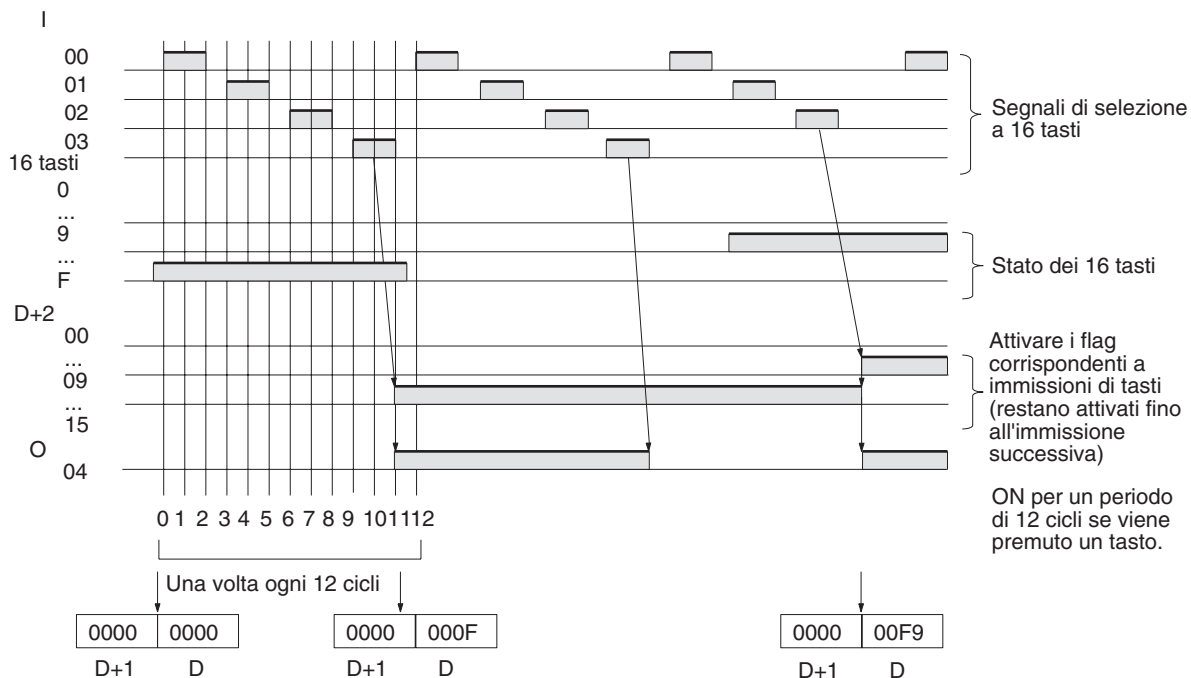
Come illustrato nel seguente diagramma, collegare il tastierino esadecimale ai contatti dei moduli di ingresso e di uscita compresi tra 0 e 3.



Gli ingressi e le uscite possono essere collegati ai seguenti tipi di moduli di I/O di base e moduli di I/O ad alta densità, purché non siano installati su un sistema slave di I/O remoti SYSMAC BUS.

- Moduli di ingresso c.c. con 8 o più punti di ingresso
- Moduli di uscita a transistor con 8 o più punti di uscita

Diagramma di funzionamento



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---------------|
| Flag di errore | ER | OFF |

Avvertenze

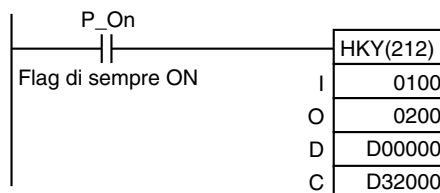
Non leggere o scrivere il canale di sistema C da altre istruzioni. Se il canale di sistema viene utilizzato da altre istruzioni, il funzionamento di HKY(212) non sarà corretto. All'avvio dell'esecuzione del programma, il canale di sistema non viene inizializzato da HKY(212) durante il primo ciclo. Se HKY(212) viene utilizzata dal primo ciclo, eliminare il canale di sistema dal programma.

Se al termine dell'esecuzione di HKY(212) non viene eseguito il refresh I/O con i moduli di ingresso e di uscita collegati al tastierino esadecimale, il funzionamento dell'istruzione non sarà corretto. Pertanto non collegare il tastierino esadecimale ai seguenti moduli.

- Moduli di I/O di base o moduli di I/O ad alta densità installati su un sistema slave di I/O remoti SYSMAC BUS
- Slave per le comunicazioni (slave DeviceNet o CompoBus/S)

Esempio

In questo esempio, HKY(212) legge fino a 8 cifre dei dati esadecimali da un tastierino esadecimale e li memorizza in D00000 e D00001. Il tastierino esadecimale è collegato tramite CIO 0100 (assegnato a un modulo di ingresso c.c. a 16 punti CS1W-ID211) e CIO 0200 (assegnato a un modulo di uscita a transistor a 16 punti CS1W-OD211). D32000 viene utilizzato come canale di sistema.



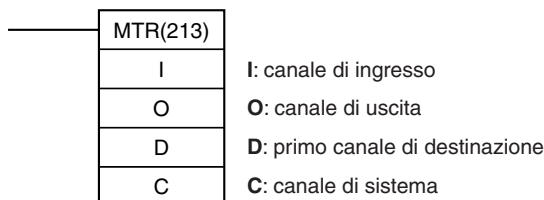
3-23-6 MATRIX INPUT: MTR(213)

Scopo

Accetta fino a 64 segnali da una matrice 8×8 collegata a un modulo di ingresso e a un modulo di uscita (utilizzando 8 punti di ingresso e 8 punti di uscita) e memorizza tali dati a 64 bit nei 4 canali di destinazione.

Questa istruzione è supportata solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | MTR(213) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

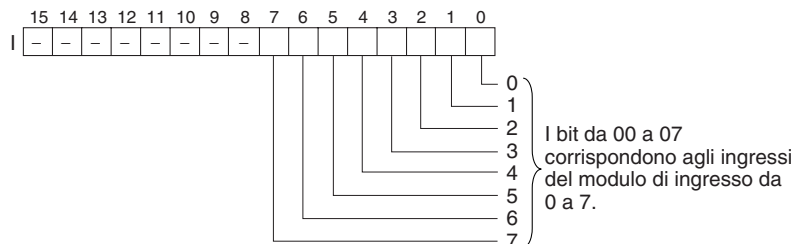
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

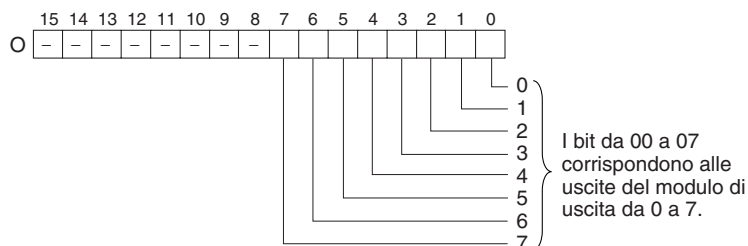
I: canale di ingresso

Specificare il canale di ingresso assegnato al modulo di ingresso e collegare le righe del segnale a 8 ingressi al modulo di ingresso come illustrato nel diagramma seguente.



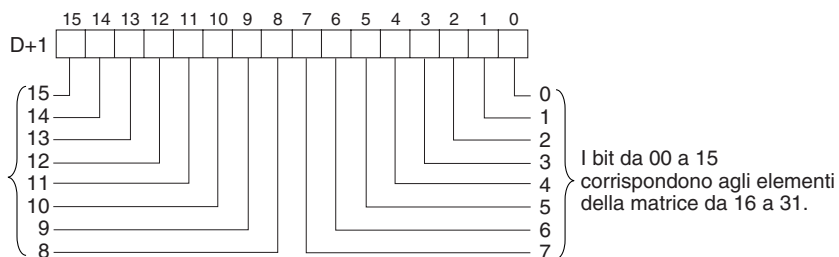
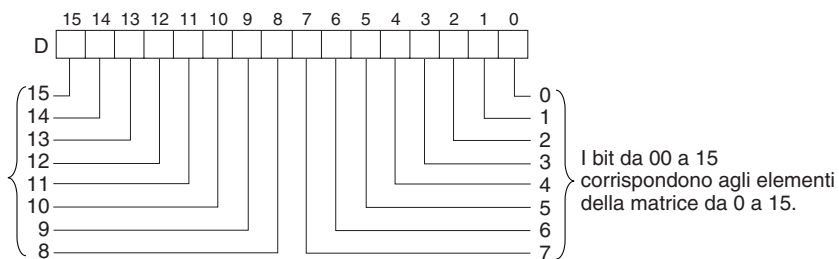
O: canale di uscita (uscite del segnale di selezione)

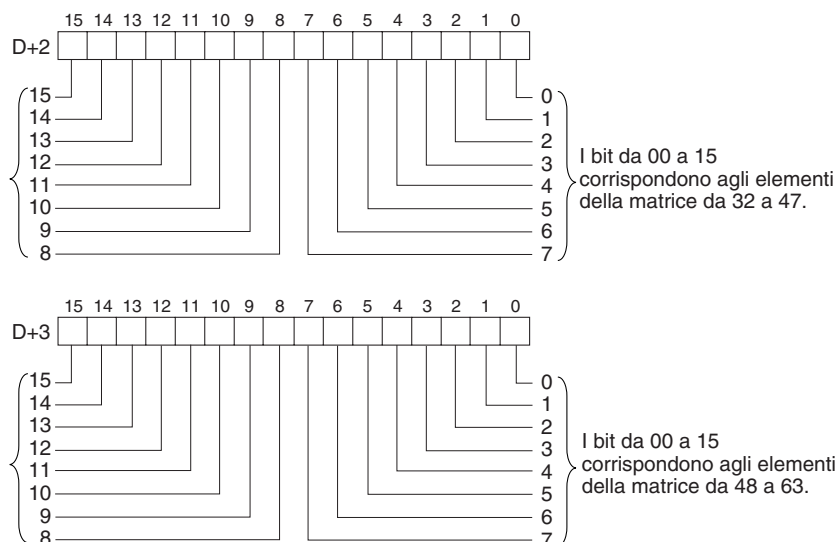
Specificare il canale di uscita assegnato al modulo di uscita e collegare i segnali di selezione a 8 punti al modulo di uscita come illustrato nel diagramma seguente.



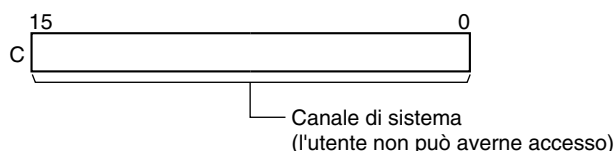
D: primo canale registro

Specifica l'indirizzo del canale iniziale dei 4 canali che contengono i dati della matrice 8 x 8.



**C: canale di sistema**

Specifica un canale di lavoro utilizzato dall'istruzione. Questo canale non può venire utilizzato in altre applicazioni.

**Caratteristiche operando**

| Area | I | O | D | C |
|--------------------------------------|--|-----------------|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 614 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W508 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H508 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | Da A448 ad A956 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4092 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4092 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32764 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32764 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da A448 ad A959 | Da En_00000 a En_32764 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | |
| Costanti | --- | | | |
| Registri dati | Da DR0 a DR15 | --- | --- | Da DR0 a DR15 |

| Area | I | O | D | C |
|--|---|---|---|---|
| Registri indice | --- | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 | | | |
| | Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 | | | |
| | Da DR0 a DR15, da IR0 a IR15 | | | |
| | Da ,IR0+(++) a ,IR15+(++) | | | |
| | Da ,-(--)IR0 a ,-(--)IR15 | | | |

Descrizione

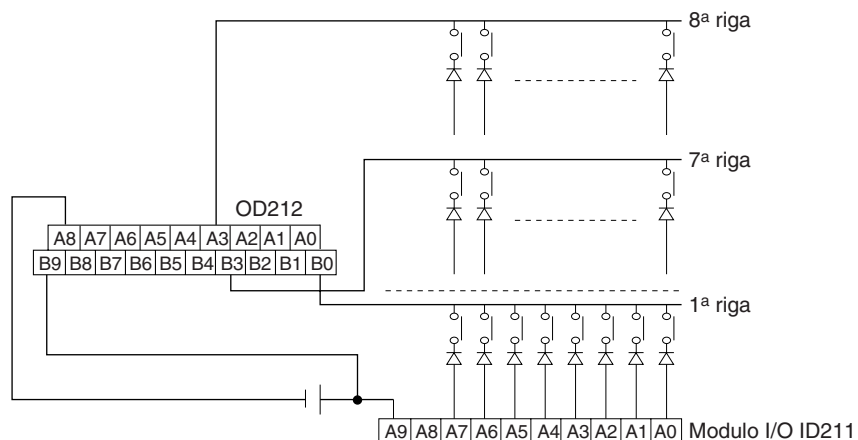
MTR(213) invia i segnali di selezione ai bit da 00 a 07 di O, legge i dati per ordine dai bit compresi tra 00 e 07 di I e memorizza i 64 bit di dati nei 4 canali da D a D+3. MTR(213) legge lo stato della matrice a 64 bit ogni 24 cicli della CPU. Il flag di un round (bit 08 di O) viene attivato una sola volta ogni 24 cicli, dopo che ciascuno dei segnali di selezione è stato impostato su ON.

Quando viene eseguita, MTR(213) inizia a leggere lo stato della matrice dall'inizio della matrice stessa, indipendentemente dal punto in cui era stata interrotta l'ultima istruzione.

A differenza delle serie C200HX/HG/HE e CQM1H, il numero di volte in cui è possibile specificare l'istruzione MTR(213) nel programma è illimitato.

Collegamenti esterni

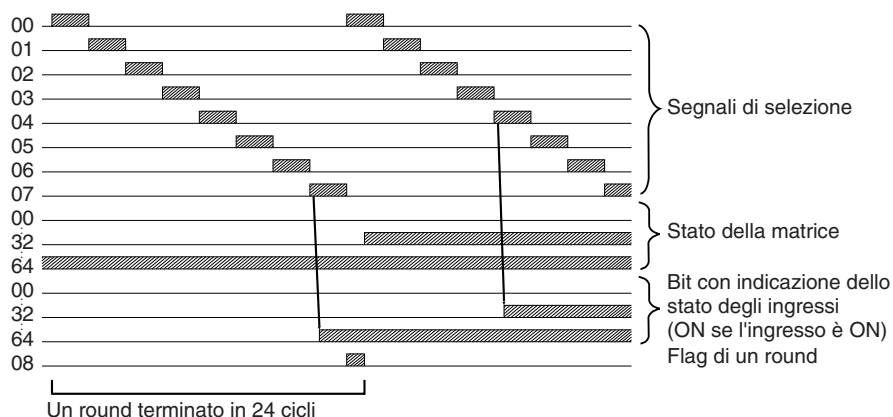
Come illustrato nel seguente diagramma, collegare il tastierino esadecimale ai contatti dei moduli di ingresso e di uscita compresi tra 0 e 3.



Gli ingressi e le uscite possono essere collegati ai seguenti tipi di moduli di I/O di base e moduli di I/O ad alta densità, purché non siano installati su un sistema slave di I/O remoti SYSMAC BUS.

- Moduli di ingresso c.c. con 8 o più punti di ingresso
- Moduli di uscita a transistor con 8 o più punti di uscita

Diagramma di funzionamento



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---------------|
| Flag di errore | ER | OFF |

Avvertenze

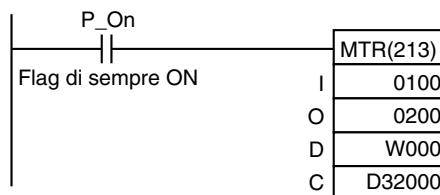
Non leggere o scrivere il canale di sistema C da altre istruzioni. Se il canale di sistema viene utilizzato da altre istruzioni, il funzionamento di MTR(213) non sarà corretto. All'avvio dell'esecuzione del programma, il canale di sistema non viene inizializzato da MTR(213) durante il primo ciclo. Se MTR(213) viene utilizzata dal primo ciclo, eliminare il canale di sistema dal programma.

Se al termine dell'esecuzione di MTR(213) non viene eseguito il refresh I/O con i moduli di ingresso e di uscita collegati alla matrice esterna, il funzionamento dell'istruzione non sarà corretto. Pertanto non collegare la matrice esterna ai seguenti moduli.

- Moduli di I/O di base o moduli di I/O ad alta densità installati su un sistema slave di I/O remoti SYSMAC BUS
- Slave per le comunicazioni (slave DeviceNet o CompoBus/S)

Esempio

Nell'esempio, MTR(213) legge i 64 bit di dati dalla matrice 8×8 e memorizza i dati nell'intervallo compreso tra W000 e W003. La matrice 8×8 è collegata tramite CIO 0100 (assegnato a un modulo di ingresso c.c. a 16 punti CS1W-ID211) e CIO 0200 (assegnato a un modulo di uscita a transistor a 16 punti CS1W-OD211). D32000 viene utilizzato come canale di sistema.



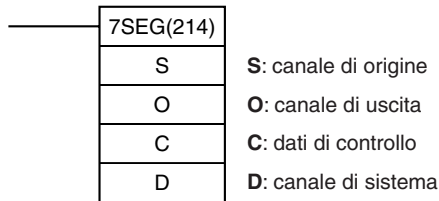
3-23-7 7-SEGMENT DISPLAY OUTPUT – 7SEG(214)

Scopo

Converte i dati di origine (a 4 o 8 cifre in formato BCD) in dati di visualizzazione a 7 segmenti e li invia al canale di uscita specificato.

Questa istruzione è supportata solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | 7SEG(214) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

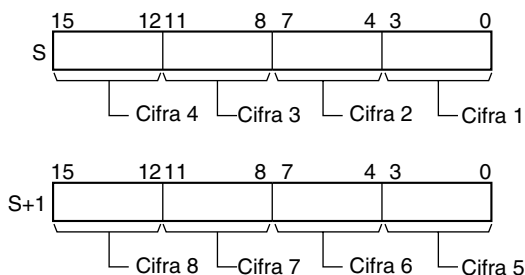
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

S: canale di origine

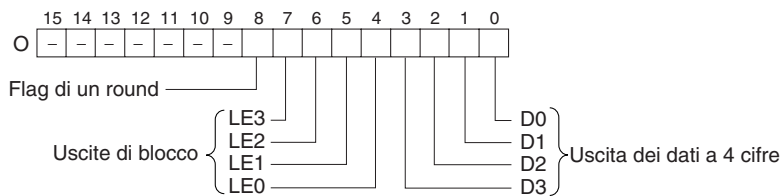
Specificare il primo canale di origine che contiene i dati da convertire in dati di visualizzazione a 7 segmenti.



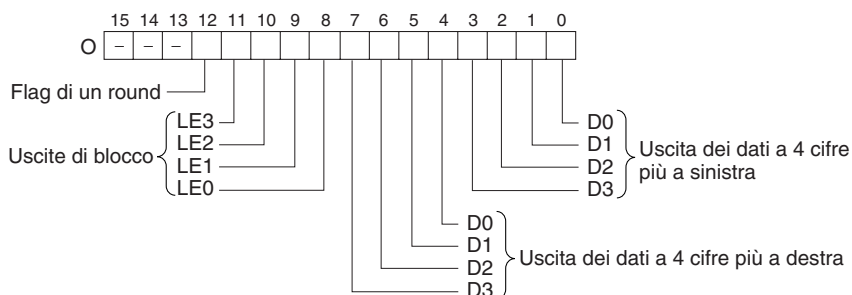
O: canale di uscita (uscite di dati e di blocco)

Specificare il canale di uscita assegnato al modulo di uscita e collegare il display a 7 segmenti al modulo di uscita come illustrato nel diagramma seguente.

- Conversione di 4 cifre



- Conversione di 8 cifre



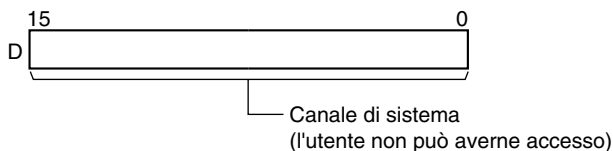
C: dati di controllo

Il valore di C indica il numero di cifre dei dati di origine e la logica per i moduli di I/O, come illustrato nella seguente tabella (per logica si intende quella dell'NPN o PNP dell'uscita a transistor).

| Dati di origine | Logica del display per i dati in ingresso | Logica del display per i dati di uscita | C |
|------------------|---|---|------|
| 4 cifre (S) | Uguale al modulo di uscita | Uguale al modulo di uscita | 0000 |
| | | Diversa dal modulo di uscita | 0001 |
| | Diversa dal modulo di uscita | Uguale al modulo di uscita | 0002 |
| | | Diversa dal modulo di uscita | 0003 |
| 8 cifre (S, S+1) | Uguale al modulo di uscita | Uguale al modulo di uscita | 0004 |
| | | Diversa dal modulo di uscita | 0005 |
| | Diversa dal modulo di uscita | Uguale al modulo di uscita | 0006 |
| | | Diversa dal modulo di uscita | 0007 |

D: canale di sistema

Specifica un canale di lavoro utilizzato dall'istruzione. Questo canale non può venire utilizzato in altre applicazioni.

**Caratteristiche operando**

| Area | S | O | C | D |
|--------------------------------------|--|-----------------|------------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | | --- | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | | --- | Da H000 a H511 |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | --- | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | --- | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | | --- | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | | --- | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | | --- | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | --- | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | |
| Costanti | --- | --- | 0000 ... 0007 | --- |

| Area | S | O | C | D |
|--|--|---------------|-----|--|
| Registri dati | --- | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | | |
| Indirizzamento indiretto con i registri indice | Da IR0 a IR15, da -2048 a +2047, da IR0 a IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | --- | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

7SEG(214) legge i dati di origine, li converte nel formato per il display a 7 segmenti e invia tali dati (come 4 cifre più a sinistra da D0 a D3, 4 cifre più a destra da D0 a D3, segnali di blocco in uscita da LE0 a LE3) ai dati a 7 segmenti per il display collegati all'uscita indicata da O. Il valore di C indica il numero di cifre dei dati di origine (4 o 8 cifre) e la logica per i moduli di ingresso e uscita.

7SEG(214) visualizza i dati a 4 o 8 cifre in 12 cicli, quindi ricomincia e continua a visualizzare i dati.

Il flag di un round (bit 08 di O quando si convertono 4 cifre, bit 12 di O quando si convertono 8 cifre) viene attivato per un ciclo ogni 12 cicli dopo che 7SEG(214) ha attivato ciascuno dei segnali di blocco in uscita. Dopo che i dati a 7 segmenti sono stati emessi in 12 cicli, l'istruzione 7SEG(214) si avvia e converte il contenuto attuale del canale o dei canali di origine dei 12 cicli successivi.

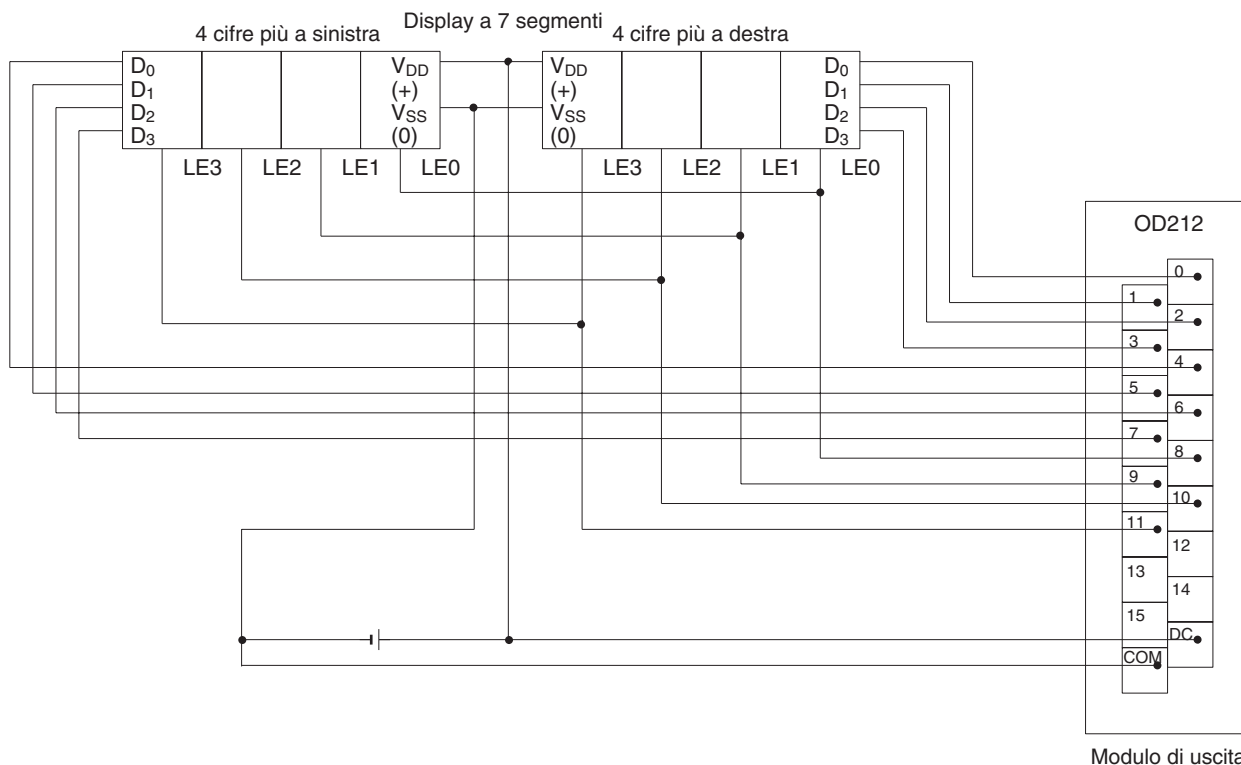
Quando viene eseguita, l'istruzione 7SEG(214) comincia sull'uscita di blocco 0 all'inizio del round, indipendentemente dal punto in cui è stata interrotta l'ultima istruzione.

Anche se il display a 7 segmenti collegato ha meno di 4 o 8 cifre visualizzate, 7SEG(214) invierà ancora 4 o 8 cifre di dati.

Collegamenti esterni

Collegare il display a 7 segmenti al modulo di uscita, come mostrato nel diagramma riportato di seguito. Nell'esempio è illustrato un display a 8 cifre. Con un display a 4 cifre, le uscite dei dati (da D0 a D3) verrebbero collegate alle uscite da 0 a 3 e le uscite di blocco (da LE0 a LE3) verrebbero collegate alle uscite da 4 a 7. Quando è stato emesso un round di dati, verrà attivato il punto di uscita 12 (per display a 8 cifre) o il punto di uscita 8 (per display a 4

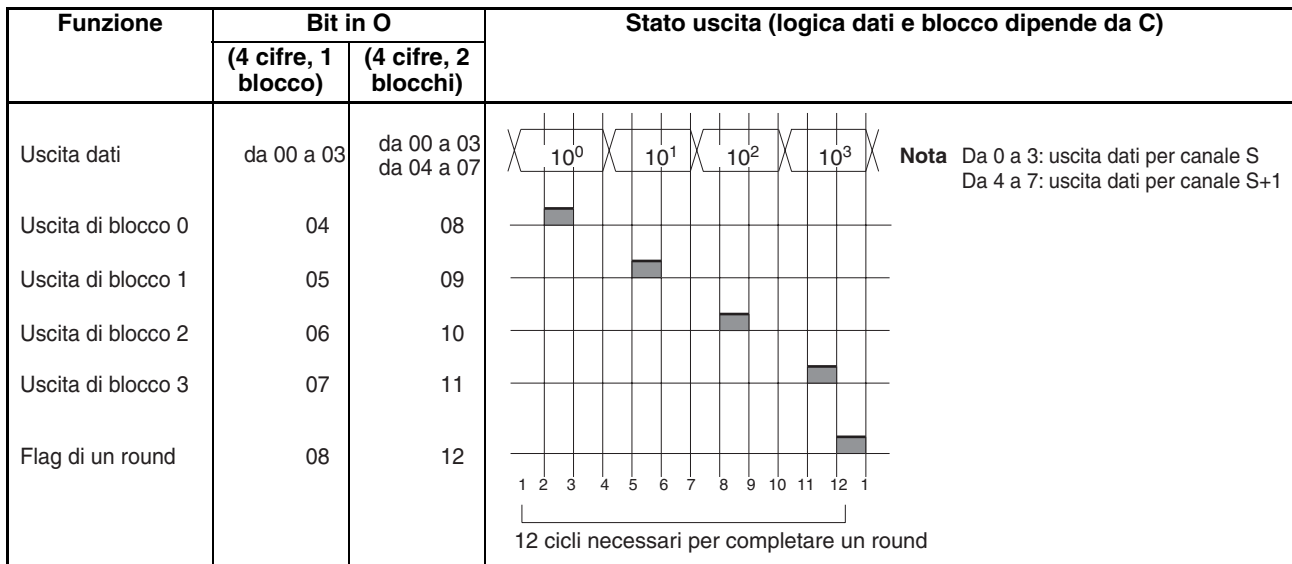
cifre). Tuttavia, non è necessario collegarli se non richiesto espressamente dall'applicazione.



È possibile collegare gli ingressi e le uscite ai seguenti tipi di moduli di I/O di base e ai moduli di I/O ad alta densità, sempre che non siano montati su un rack di I/O remoto SYSMAC BUS.

- display a 4 cifre: moduli di uscita a transistor con 8 o più punti di uscita,
- display a 8 cifre: moduli di uscita a transistor con 16 o più punti di uscita.

Diagramma temporale



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|------------|
| Flag di errore | ER | OFF |

Avvertenze

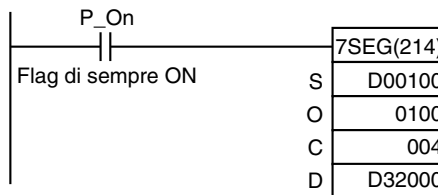
Non leggere o scrivere il canale di sistema (D) da altre istruzioni. 7SEG(214) non funzionerà correttamente se al canale di sistema si accede da un'altra istruzione. Il canale di sistema non viene inizializzato da 7SEG(214) nel primo ciclo quando si avvia l'esecuzione del programma. Se 7SEG(214) deve essere utilizzata dal primo ciclo, eliminare il canale di sistema dal programma. 7SEG(214) non funzionerà correttamente se non viene eseguito l'aggiornamento I/O con il modulo di uscita collegato al display a 7 segmenti dopo l'esecuzione dell'istruzione 7SEG(214). Pertanto, non collegare la matrice esterna ai seguenti moduli:

- moduli di I/O di base o moduli di I/O ad alta densità montati sul rack di un sistema slave di I/O remoto SYSMAC BUS,
- slave di comunicazione (slave DeviceNet o CompoBus/S).

Esempio

In questo esempio, 7SEG(214) converte i dati BCD a 8 cifre in D00100 e D00101 e li invia tramite CIO 0100 a un display a 7 segmenti collegato a un modulo di uscita a transistor CS1W-OD211 con 16 punti di uscita.

Vengono dunque inviate 8 cifre di dati e la logica del display a 7 segmenti è la stessa logica dell'unità di uscita, quindi i dati di controllo (C) vengono impostati su 0004. D32000 viene utilizzato come canale di sistema, D.



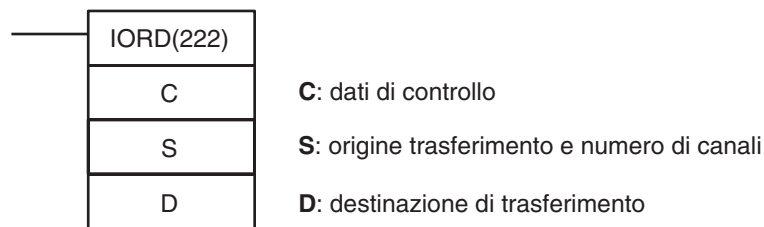
3-23-8 INTELLIGENT I/O READ: IORD(222)

Scopo

Legge il contenuto dell'area di memoria di un modulo di I/O speciale o dell'unità bus CPU (vedere nota).

Nota Vi sono alcune limitazioni nella funzionalità delle unità bus CPU. Consultare a questo proposito il paragrafo *Limitazioni* più avanti in questa sezione.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | IORD(222) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @IORD(222) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

- C:** dipende dal modulo di I/O speciale o dall'unità bus CPU.
S: unità di I/O speciale: da 0000 a 005F esadecimale
(per specificare i numeri di modulo da 0 a 95)
unità bus CPU: da 8000 a 800F esadecimale
(per specificare i numeri di modulo da 0 a F
esadecimale)
S+1: numero di canali da trasferire
(da 0001 a 0080 esadecimale, in base al modulo di I/O speciale o
dall'unità bus CPU)



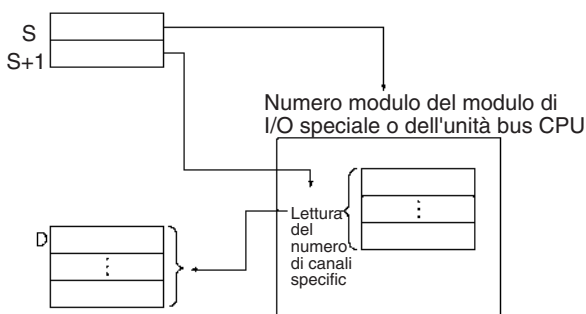
S+1: 4 cifre più a sinistra
S: 4 cifre più a destra

Caratteristiche operando

| Area | C | S | D |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A958 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | Solo valori specificati | --- |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

IORD(222) legge il numero di canali impostati in S+1 dall'area di memoria del modulo di I/O speciale o dall'unità bus CPU il cui numero di modulo è specificato da S e invia i dati a D. È possibile specificare soltanto moduli di I/O speciali o unità bus CPU montate su rack CPU o rack I/O di espansione. Per informazioni specifiche su ogni singola unità, consultare il manuale dell'operatore del relativo modulo di I/O speciale o dell'unità bus CPU da cui vengono letti i dati.



Limitazioni

La lettura di un'unità bus CPU è soggetta alle limitazioni esposte di seguito.

■ Limitazioni del modulo CPU

CPU CS1-H

La lettura di un'unità bus CPU è possibile soltanto con i seguenti modelli di CPU ed esclusivamente con CPU prodotte dal 18 aprile 2003 in poi (numero di lotto 030418 o successivo).

- CS1G-CPU□□H
- CS1H-CPU□□H

Per determinare la data di produzione, verificare il numero di lotto riportato sul lato o sulla parte inferiore della CPU. Nei numeri di lotto, la data di produzione è indicata come segue:

AAMMGG nnnn

AA = le ultime due cifre dell'anno, MM = mese espresso come valore numerico, GG = giorno del mese, nnnn = numero di serie

CPU CJ1-H, CJ1M e CS1D

La lettura di un'unità bus CPU è possibile soltanto con le CPU versione 2.0 o successiva.

Nota Se IORD(222) viene eseguita per un'unità bus CPU in esecuzione su un modulo CPU che non supporta l'utilizzo di tale istruzione con le unità bus CPU, si verificherà un errore e verrà attivato il flag ER.

■ Limitazioni per CX-Programmer

È possibile specificare i numeri di modulo per le unità bus CPU per S con CX-Programmer versione 3.0 o successiva.

Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|--|
| Flag di errore | ER | <p>ON se il numero di canali da trasferire (S) non è incluso nell'intervallo da 0001 a 0080 esadecimale.</p> <p>ON se il numero di modulo (S) non è incluso nell'intervallo da 0000 a 005F esadecimale o da 8000 a 800F esadecimale.</p> <p>ON se è il modulo di I/O speciale specificato è su SYSMAC BUS.</p> <p>ON se è stato specificato un modulo di I/O speciale o un'unità bus CPU non influenzati dall'istruzione IORD(222).</p> <p>ON se è stato specificato un modulo di I/O speciale con un errore del modulo di I/O speciale o con un errore di configurazione del modulo di I/O speciale.</p> <p>ON se è stato specificato un modulo bus CPU con un errore del modulo bus CPU o con un errore di configurazione del modulo bus CPU.</p> <p>Con le CPU CS1D: ON se non è stato possibile sincronizzare le CPU attiva e in attesa.</p> <p>OFF in tutti gli altri casi.</p> |
| Flag di uguaglianza | = | <p>ON se l'operazione di lettura è stata completata regolarmente.</p> <p>OFF se l'operazione di lettura non è stata completata regolarmente.</p> |

Avvertenze

Il flag di uguaglianza verrà attivato se l'operazione di lettura è stata completata regolarmente.

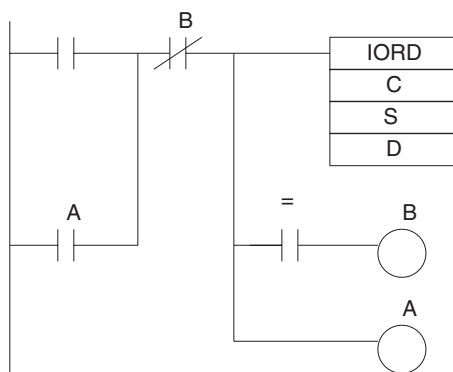
Il flag di uguaglianza verrà disattivato se l'operazione di lettura non è stata completata regolarmente a causa del modulo di I/O speciale occupato o dell'unità bus CPU occupata.

Ogni volta che si verifica una delle seguenti condizioni, verrà generato un errore e verrà attivato il flag di errore.

- Il numero di canali da trasferire (S) non è incluso nell'intervallo da 0001 a 0080 (esadecimale).
- Il numero di modulo (S) non è incluso nell'intervallo da 0000 a 005F esadecimale o da 8000 a 800F esadecimale.
- Il modulo di I/O speciale specificato è su SYSMAC BUS.
- È stato specificato un modulo di I/O speciale o un'unità bus CPU non influenzati dall'istruzione IORD(222).
- Se è stato specificato un modulo di I/O speciale con un errore del modulo di I/O speciale o con un errore di configurazione del modulo di I/O speciale.
- Se è stato specificato un modulo bus CPU con un errore del modulo bus CPU o con un errore di configurazione del modulo bus CPU.

Quando viene eseguita l'istruzione IORD(222), i risultati dell'esecuzione si riflettono sui flag di condizione. In particolare, il flag di uguaglianza viene impostato su ON quando la lettura è stata completata. Immettere i flag di condizione come il flag di uguaglianza con diramazione di uscita dalle medesime condizioni di ingresso dell'istruzione IORD(222).

Se il modulo di I/O speciale o l'unità bus CPU è occupato/a, l'operazione di lettura non verrà eseguita. Utilizzare il flag di uguaglianza per creare un programma di autoritenuta come mostrato di seguito, in modo che l'istruzione IORD(222) venga eseguita ad ogni ciclo fino all'esecuzione dell'operazione di lettura.

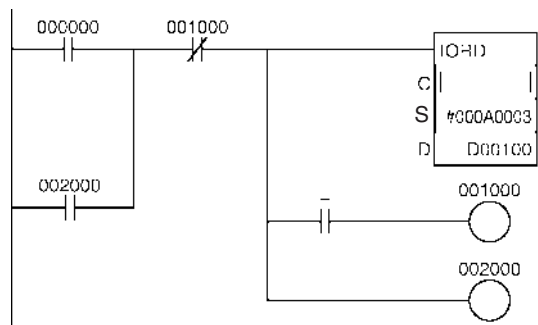


Quando la condizione di ingresso è soddisfatta, l'autoritenuta viene effettuata dall'uscita A e l'istruzione IORD(222) viene eseguita ad ogni ciclo finché viene attivato il flag di uguaglianza. Quando la lettura è stata completata e il flag di uguaglianza è ON, viene attivata l'uscita B e viene cancellata l'autoritenuta.

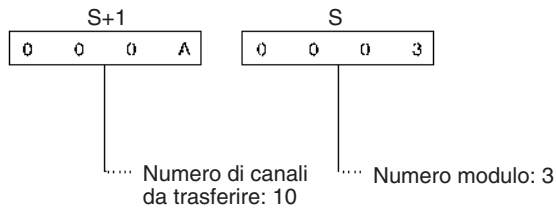
Verificare che la posizione dei flag di condizione sia immediatamente dopo le istruzioni IORD(222) e non dopo istruzioni di altro tipo. Se un flag di condizione viene inserito dopo un'altra istruzione, verrà influenzato dai risultati dell'esecuzione di tale istruzione.

Esempio

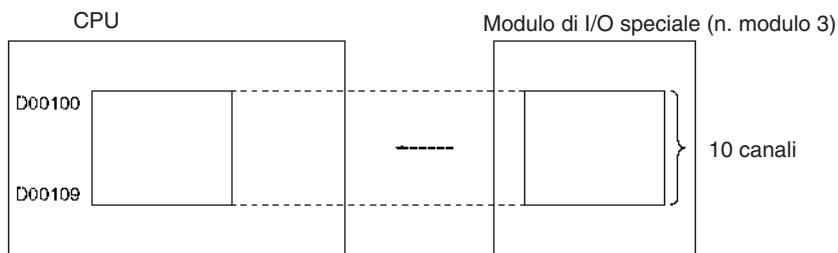
In questo esempio l'istruzione IORD(222) viene utilizzata per leggere i dati.



Quando CIO 000000 è impostato su ON, dall'unità di I/O speciale con numero di unità 3 vengono letti 10 canali, che verranno memorizzati in



Il codice di controllo (C) varia in base all'unità di I/O speciale.

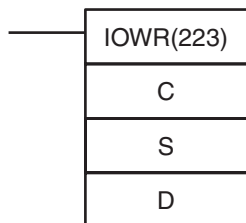


3-23-9 INTELLIGENT I/O WRITE: IOWR(223)

Scopo

Invia il contenuto dell'area di memoria I/O della CPU al modulo di I/O speciale o all'unità bus CPU (vedere nota).

Nota Vi sono alcune limitazioni nella funzionalità delle unità bus CPU. Consultare a questo proposito il paragrafo *Limitazioni* più avanti in questa sezione.

Simbolo programmazione ladder

C: dati di controllo

S: origine trasferimento e numero di canali

D: destinazione trasferimento e numero di canali

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | IOWR(223) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @IOWR(223) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi
C: dipende dal modulo di I/O speciale o dall'unità bus CPU.

D: unità di I/O speciale: da 0000 a 005F esadecimale

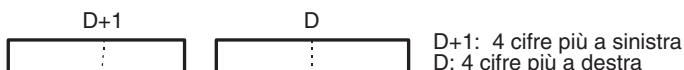
(per specificare i numeri di modulo da 0 a 95)

unità bus CPU: da 8000 a 800F esadecimale

(per specificare i numeri di modulo da 0 a F esadecimale)

D+1: numero di canali da trasferire

(da 0000 a 0080 esadecimale, in base al modulo di I/O speciale o dall'unità bus CPU)

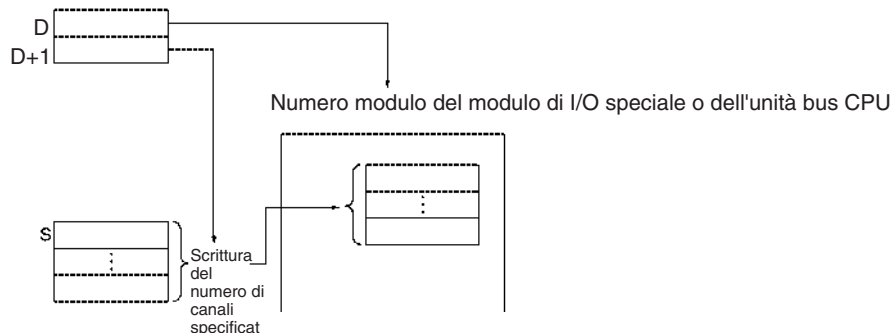

Caratteristiche operando

| Area | C | S | D |
|--------------------------------------|--|---|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6142 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W510 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H510 |
| Area bit ausiliaria | Da A000 a A959 | | Da A000 ad A958 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4094 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4094 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32766 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32766 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32766 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |

| Area | C | S | D |
|--|--|-----|-------------------------|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | | Solo valori specificati |
| Registri dati | Da DR0 a DR15 | --- | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

IOWR(223) scrive il numero di canali impostati (D) dal primo canale di origine (specificato da S) in avanti e li invia al modulo di I/O speciale o all'unità bus CPU avente il numero di modulo indicato da D. È possibile specificare soltanto moduli di I/O speciali o unità bus CPU montate su rack CPU o rack I/O di espansione.

**Limitazioni**

La scrittura di un'unità bus CPU è soggetta alle limitazioni esposte di seguito.

■ Limitazioni del modulo CPU**CPU CS1-H**

La scrittura in un'unità bus CPU è possibile soltanto con i seguenti modelli di CPU ed esclusivamente con CPU prodotte dal 18 aprile 2003 in poi (numero di lotto 030418 o successivo).

- CS1G-CPU□□H
- CS1H-CPU□□H

Per determinare la data di produzione, verificare il numero di lotto riportato sul lato o sulla parte inferiore della CPU. Nei numeri di lotto, la data di produzione è indicata come segue:

AAMMGG nnnn

AA = le ultime due cifre dell'anno, MM = mese espresso come valore numerico, GG = giorno del mese, nnnn = numero di serie

CPU CJ1-H, CJ1M e CS1D

La scrittura in un'unità bus CPU è possibile soltanto con le CPU versione 2.0 o successiva.

Nota Se IOWR(223) viene eseguita per un'unità bus CPU in esecuzione su un modulo CPU che non supporta l'utilizzo di tale istruzione con le unità bus CPU, si verificherà un errore e verrà attivato il flag ER.

■ Limitazioni per CX-Programmer

È possibile specificare i numeri di modulo per le unità bus CPU per S con CX-Programmer versione 3.0 o successiva.

Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|--|
| Flag di errore | ER | <p>ON se il numero di canali da trasferire (D) non è incluso nell'intervallo da 0001 a 0080 esadecimale.</p> <p>ON se il numero di modulo (D) non è incluso nell'intervallo da 0000 a 005F esadecimale o da 8000 a 800F esadecimale.</p> <p>ON se S viene specificato da una costante quando il numero di canali da trasferire (D+1) non è 0001 esadecimale.</p> <p>ON se il modulo di I/O speciale specificato è su SYSMAC BUS.</p> <p>ON se è stato specificato un modulo di I/O speciale o un'unità bus CPU non influenzati dall'istruzione IOWR(223).</p> <p>ON se è stato specificato un modulo di I/O speciale con un errore del modulo di I/O speciale o con un errore di configurazione del modulo di I/O speciale.</p> <p>ON se è stato specificato un modulo bus CPU con un errore del modulo bus CPU o con un errore di configurazione del modulo bus CPU.</p> <p>Con le CPU CS1D: ON se non è stato possibile sincronizzare le CPU attiva e in attesa.</p> <p>OFF in tutti gli altri casi.</p> |
| Flag di uguaglianza | = | <p>ON se l'operazione di scrittura è stata completata regolarmente.</p> <p>OFF se l'operazione di scrittura non è stata completata regolarmente.</p> |

Avvertenze

Quando per il numero di canali da trasferire (D+1) è stato specificato "0001", è possibile definire i dati per S tramite una costante. Se per S viene impostata una costante quando il numero di canali da trasferire non è "0001", si verificherà un errore e verrà attivato il flag di errore.

Il flag di uguaglianza verrà attivato se l'operazione di scrittura è stata completata regolarmente.

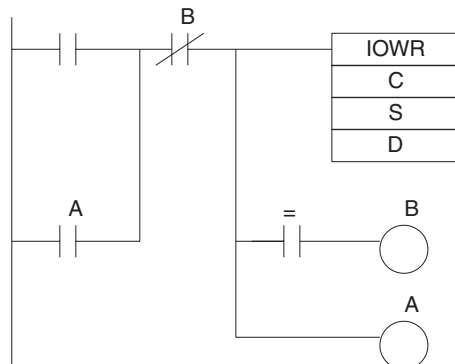
Il flag di uguaglianza verrà disattivato se l'operazione di scrittura non è stata completata regolarmente a causa del modulo di I/O speciale occupato o dell'unità bus CPU occupata.

Ogni volta che si verifica una delle seguenti condizioni, verrà generato un errore e verrà attivato il flag di errore.

- In presenza di un errore di verifica del modulo di I/O, un errore di configurazione del modulo di I/O speciale o un errore di modulo di I/O speciale sul modulo di I/O speciale.
- In presenza di un errore di verifica del modulo di I/O, un errore di configurazione dell'unità bus CPU o un errore di unità bus CPU sull'unità bus CPU.
- Il numero di canali da trasferire (D) non è incluso nell'intervallo da 0001 a 0080 (esadecimale).
- Il numero di modulo (D) non è incluso nell'intervallo da 0000 a 005F esadecimale o da 8000 a 800F esadecimale.
- Il modulo di I/O speciale specificato è su SYSMAC BUS.
- È stato specificato un modulo di I/O speciale o un'unità bus CPU non influenzati dall'istruzione IOWR(223).
- È stato specificato un modulo di I/O speciale con un errore del modulo di I/O speciale o con un errore di configurazione del modulo di I/O speciale.
- È stato specificato un modulo di I/O speciale con un errore del modulo di I/O speciale o con un errore di configurazione del modulo di I/O speciale.

Quando viene eseguita l'istruzione IOWR(223), i risultati dell'esecuzione si riflettono sui flag di condizione. In particolare, il flag di uguaglianza viene impostato su ON quando la lettura è stata completata. Immettere i flag di condizione come il flag di uguaglianza con diramazione di uscita dalle medesime condizioni di ingresso dell'istruzione IOWR(223).

Se il modulo di I/O speciale o l'unità bus CPU è occupato/a, l'operazione di scrittura non verrà eseguita. Utilizzare il flag di uguaglianza per creare un programma di autoritenuta, come mostrato di seguito, in modo che l'istruzione IOWR(223) venga eseguita ad ogni ciclo fino all'esecuzione dell'operazione di scrittura.

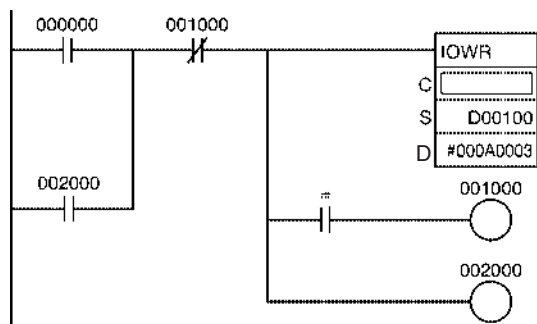


Quando la condizione di ingresso è soddisfatta, l'autoritenuta viene effettuata dall'uscita A e l'istruzione IOWR(223) viene eseguita ad ogni ciclo finché viene attivato il flag di uguaglianza. Quando la scrittura è stata completata e il flag di uguaglianza è ON, viene attivata l'uscita B e viene cancellata l'autoritenuta.

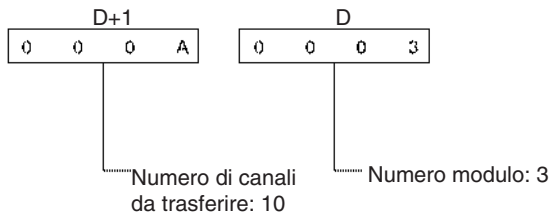
Verificare che la posizione dei flag di condizione sia immediatamente dopo le istruzioni IOWR(223) e non dopo istruzioni di altro tipo. Se un flag di condizione viene inserito dopo un'altra istruzione, verrà influenzato dai risultati dell'esecuzione di tale istruzione.

Esempio

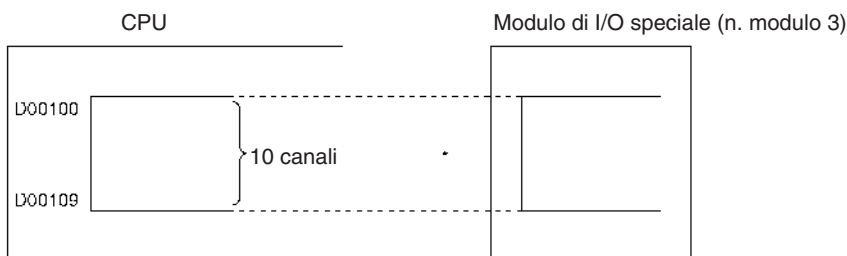
In questo esempio l'istruzione IOWR(223) viene utilizzata per scrivere i dati.



Quando CIO 000000 è impostato su ON, i 10 canali presenti in D00100 - D00109 vengono scritti nel modulo di I/O



Il codice di controllo (C) varia in base all'unità di I/O speciale.



3-23-10 CPU BUS UNIT I/O REFRESH: DLNK(226)

Scopo

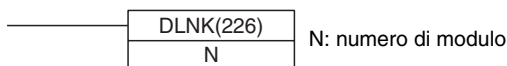
Esegue immediatamente l'aggiornamento degli I/O dell'unità bus CPU con il numero di modulo specificato. Vengono aggiornati i seguenti dati:

- i canali assegnati all'unità bus CPU nelle aree dell'unità bus CPU del PLC (25 canali nell'area CIO e 100 canali nell'area),

- l'aggiornamento specifico dei dati delle unità, ad esempio unità che supportano i data link.

Questa istruzione è supportata solo dalle CPU CS1/H, CJ1/H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | DLNK(226) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ DLNK(226) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero di modulo

Specifica il numero di modulo dell'unità bus CPU (da 0000 a 000F esadecimale o da 0 a 15 decimale).

Caratteristiche operando

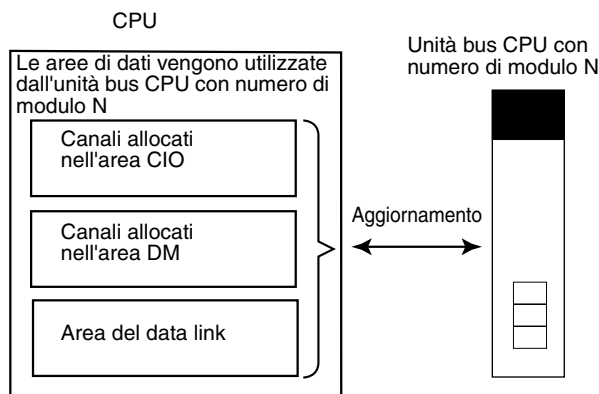
| Area | N |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Da #0000 a 000F (binario) o da 0 a 15 (decimale) |
| Registri dati | Da DR0 a DR15 |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

DLNK(226) esegue l'aggiornamento immediato degli I/O dell'unità bus CPU con il numero di modulo specificato. Vengono aggiornati i dati riportati di seguito. Per le condizioni di esecuzione da utilizzare per l'aggiornamento immediato, consultare più avanti la sezione *Avvertenze*.

1. I canali assegnati all'unità bus CPU nelle aree dell'unità bus CPU del PLC (25 canali nell'area CIO e 100 canali nell'area).
2. Dati specifici dell'unità bus CPU, come i data link o i dati per le comunicazioni con I/O remoti DeviceNet (aggiornati insieme ai dati nelle aree dell'unità bus CPU).

| Unità bus CPU | Aggiornamento dei dati specifici del modulo |
|--|--|
| Modulo Controller Link e modulo SYSMAC Link | Aggiornamento del data link |
| Modulo DeviceNet (non comprende moduli master DeviceNet C200H) | Aggiornamento delle comunicazioni degli I/O remoti |



Nella tabella seguente sono riportate le differenze tra le istruzioni DLNK(226) e IORF(097).

| Istruzione | Operazione |
|------------|--|
| DLNK(226) | <ul style="list-style-type: none"> • Aggiornamento I/O dell'area dell'unità bus CPU CS1 nell'area CIO (25 canali) • Aggiornamento I/O dell'area dell'unità bus CPU CS1 nell'area DM (100 canali) • Aggiornamento dei dati specifici dell'unità bus CPU, come i data link o i dati per le comunicazioni con I/O remoti DeviceNet |
| IORF(097) | <ul style="list-style-type: none"> • Aggiornamento I/O dei canali utilizzati dai moduli di I/O di base • Aggiornamento I/O dei 10 canali CIO assegnati a un modulo di I/O speciale |

DLNK(226) aggiorna i dati tra il modulo CPU e l'unità bus CPU specificata. Quando si utilizza l'istruzione DLNK(226), vi sono due fattori speciali da tenere in considerazione:

- 1,2,3...**
1. Quando si scambiano i dati tramite un data link o una comunicazione con I/O remoti DeviceNet, i dati non vengono scambiati con altri moduli durante il periodo di tempo in cui viene eseguita l'istruzione DLNK(226). Lo scambio di dati avverrà quando il ciclo di comunicazione della rete raggiunge il modulo in questione e i dati verranno scambiati con tale modulo. Pertanto, lo scambio effettivo dei dati potrebbe essere ritardato a seconda del tempo del ciclo di comunicazione della rete.
 2. Con l'istruzione DLNK(226) non è possibile eseguire l'aggiornamento I/O se su tale unità bus CPU è in corso lo scambio di dati. Se DLNK(226) viene eseguita troppo spesso, l'aggiornamento I/O non verrà effettuato. Si consiglia pertanto di consentire un ritardo tra le esecuzioni di DLNK(226) che sia superiore al tempo del ciclo di comunicazione.

Flag

| Nome | Etichetta | Operazione |
|---------------------|-----------|---|
| Flag di errore | ER | ON se il numero di modulo specificato non è compreso tra 0000 e 000F esadecimale (tra 0 e 15 decimale). ON se il PLC non è dotato di un'unità bus CPU con il numero di modulo specificato. Con le CPU CS1D: ON se non è stato possibile sincronizzare le CPU attiva e in attesa. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | OFF se non è stato possibile aggiornare gli I/O poiché sull'unità bus CPU era in corso l'aggiornamento dei dati. OFF in presenza di un errore di unità bus CPU o di un errore di impostazione dell'unità bus CPU nell'unità bus CPU specificata. OFF se DLNK(226) è stata eseguita in un task ad interrupt o si è verificato un conflitto con il normale aggiornamento I/O che ha provocato una sovrapposizione degli aggiornamenti. ON se l'aggiornamento I/O è stato completato normalmente. |

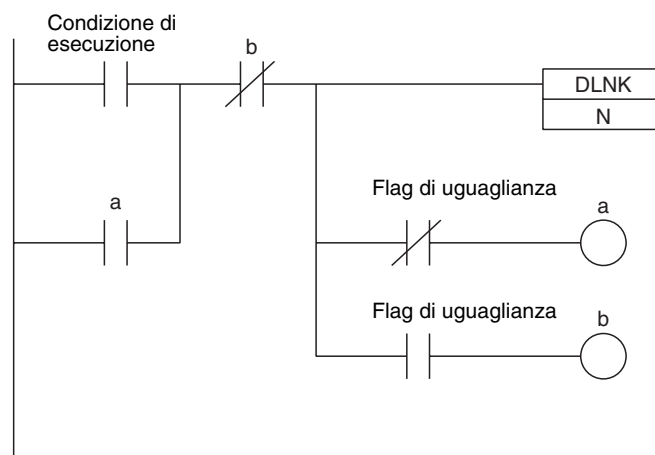
Avvertenze

Se nell'unità bus CPU specificata si è verificato un errore di unità bus CPU (A40207) o un errore di configurazione dell'unità bus CPU (A40203), l'aggiornamento I/O non verrà eseguito.

Se si verifica un errore del bus di I/O mentre l'istruzione DLNK(226) sta eseguendo l'aggiornamento I/O, tale aggiornamento verrà interrotto.

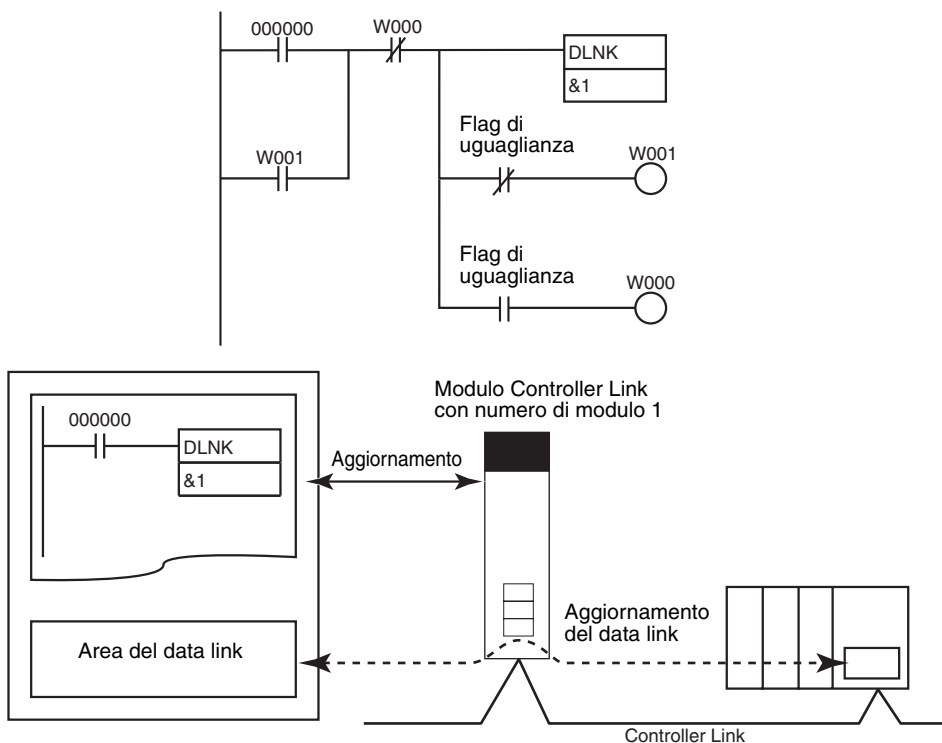
DLNK(226) aggiorna i dati tra il modulo CPU e l'unità bus CPU specificata. Lo scambio dei dati con un'unità bus CPU necessita di un po' di tempo, ad esempio un data link con un modulo Controller Link.

Se l'unità bus CPU sta scambiando i dati, l'istruzione DLNK(226) non verrà eseguita e il flag di uguaglianza verrà disattivato. Si consiglia di programmare le condizioni di esecuzione riportate di seguito in modo che l'esecuzione di DLNK(226) venga ritentata automaticamente.



Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, DLNK(226) esegue l'aggiornamento immediato degli I/O (in questo caso, l'aggiornamento del data link nel PLC) per l'unità bus CPU avente il numero di modulo 1 (in questo caso, un modulo Controller Link). Se non è possibile eseguire l'aggiornamento I/O poiché il modulo Controller Link sta aggiornando i dati, il flag di uguaglianza verrà disattivato provocando l'attivazione di W001 e l'esecuzione dell'istruzione verrà ritentata nel ciclo successivo. Se invece l'aggiornamento I/O viene completato normalmente, il flag di uguaglianza verrà attivato e l'istruzione non verrà ritentata nel ciclo successivo.

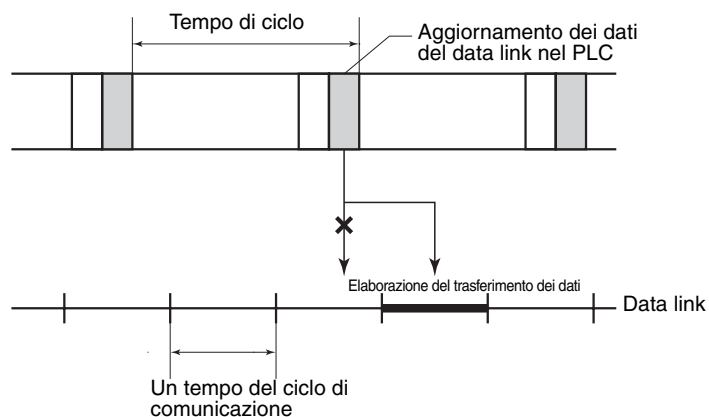


In questo esempio, la tempistica effettiva per l'aggiornamento dell'area del data link è la seguente:

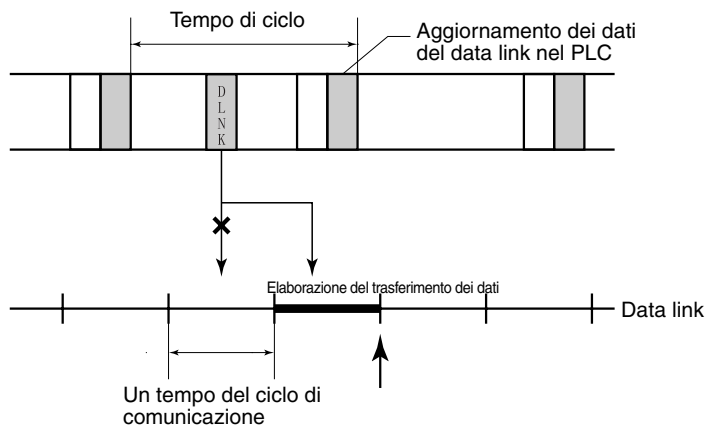
- Nella trasmissione: i dati vengono trasmessi tramite la rete la volta successiva che viene acquisito il diritto di token. La trasmissione dei dati viene ritardata al massimo fino a 1 tempo del ciclo di comunicazione.
- Nella ricezione: i dati immessi sono stati ricevuti dalla rete l'ultima volta che è stato acquisito il diritto di token. La ricezione dei dati viene ritardata al massimo fino a 1 tempo del ciclo di comunicazione.

Esempi di elaborazione del trasferimento di dati:

- Trasferimento dei dati dal precedente aggiornamento I/O



- Trasferimento dei dati con l'esecuzione di DLNK(226)



3-24 Istruzioni per la comunicazione seriale

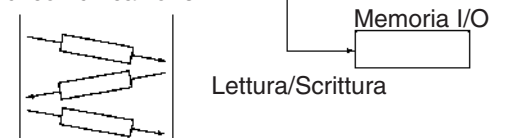
In questa sezione vengono descritte le istruzioni utilizzate per la comunicazione seriale.

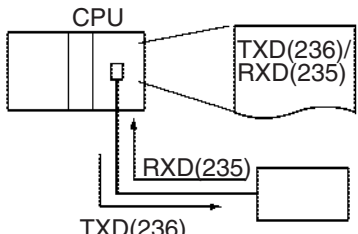
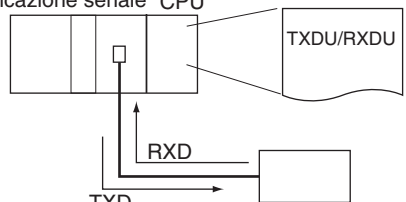
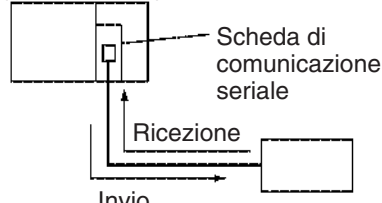
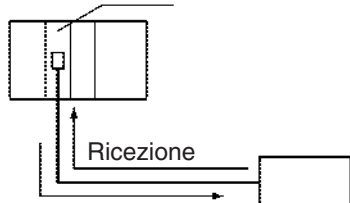
| Istruzione | Mnemonico | Codice funzione | Pagina |
|---|-----------|-----------------|--------|
| PROTOCOL MACRO | PMCR | 260 | 928 |
| TRANSMIT | TXD | 236 | 937 |
| RECEIVE | RXD | 235 | 944 |
| TRANSMIT VIA SERIAL COMMUNICATIONS UNIT | TXDU | 256 | 952 |
| RECEIVE VIA SERIAL COMMUNICATIONS UNIT | RXDU | 255 | 960 |
| CHANGE SERIAL PORT SETUP | STUP | 237 | 968 |

3-24-1 Comunicazione seriale

Sono disponibili due tipi di istruzioni per la comunicazione seriale. Le istruzioni TXD(236), RXD(235), TXDU(256) e RXDU(255) inviano e ricevono i dati tramite una comunicazione senza protocollo (personalizzata) con un dispositivo esterno. L'istruzione PMCR(260) invia e riceve i dati con un dispositivo esterno utilizzando protocolli definiti dall'utente. La differenza è illustrata nelle tabelle riportate di seguito.

- Nota**
1. Le istruzioni TXD(236) e RXD(235) trasferiscono i dati soltanto tramite una porta seriale incorporata nella CPU o una porta seriale di una scheda di comunicazione seriale (versione 1.2 o successiva).
 2. Le istruzioni TXDU(256) e RXDU(255) trasferiscono i dati soltanto tramite un modulo di comunicazione seriale (versione 1.2 o successiva).

| Istruzioni | Frame di comunicazione | Funzione | | | | | | | | | | | | | | | |
|---|--|--------------|------------------|-------------|------------------|-------------|--|------|----|------|----|----|----|------|----|----|--|
| TXD(236), RXD(235), TXDU(256) e RXDU(255) | <p>È possibile utilizzare una qualsiasi delle impostazioni indicate di seguito.</p> <p>Nessun codice di inizio o di fine</p> <table border="1" data-bbox="316 283 563 315"> <tr><td>Dati</td></tr> </table> <p>Solo codice di inizio</p> <table border="1" data-bbox="316 373 563 405"> <tr><td>ST</td><td>Dati</td></tr> </table> <p>Solo codice di fine</p> <table border="1" data-bbox="316 464 563 495"> <tr><td>Dati</td><td>ED</td></tr> </table> <p>Codice di inizio e di fine</p> <table border="1" data-bbox="619 283 866 315"> <tr><td>ST</td><td>Dati</td><td>ED</td></tr> </table> <p>Codice di fine CR+LF</p> <table border="1" data-bbox="619 373 866 405"> <tr><td>Dati</td><td>CR</td><td>LF</td></tr> </table> <p>Codice di inizio e codice di fine CR+LF</p> <table border="1" data-bbox="619 464 866 495"> <tr><td>ST</td><td>Dati</td><td>CR</td><td>LF</td></tr> </table> | Dati | ST | Dati | Dati | ED | ST | Dati | ED | Dati | CR | LF | ST | Dati | CR | LF | <p>Invia o riceve i dati in una sola direzione.</p> <p>È possibile impostare un ritardo per l'invio.</p> |
| Dati | | | | | | | | | | | | | | | | | |
| ST | Dati | | | | | | | | | | | | | | | | |
| Dati | ED | | | | | | | | | | | | | | | | |
| ST | Dati | ED | | | | | | | | | | | | | | | |
| Dati | CR | LF | | | | | | | | | | | | | | | |
| ST | Dati | CR | LF | | | | | | | | | | | | | | |
| PMCR(260) | <p>Per soddisfare i requisiti del dispositivo esterno, è possibile creare i seguenti tipi di frame (messaggi).</p> <table border="1" data-bbox="339 598 858 630"> <tr> <td>Intestazione</td> <td>Indirizzo</td> <td>Dati</td> <td>Controllo errori</td> <td>Terminatore</td> </tr> </table> <p>È possibile creare step di comunicazione.</p>  | Intestazione | Indirizzo | Dati | Controllo errori | Terminatore | <p>Per l'invio e la ricezione è possibile definire fino a 16 step.</p> <p>In base alle risposte, è possibile modificare gli step e riprovare l'elaborazione eseguita.</p> <p>È possibile definire tempi di monitoraggio della comunicazione.</p> <p>È possibile leggere e scrivere i simboli per il PLC.</p> <p>È possibile utilizzare simboli di ripetizione.</p> <p>Altro.</p> | | | | | | | | | | |
| Intestazione | Indirizzo | Dati | Controllo errori | Terminatore | | | | | | | | | | | | | |

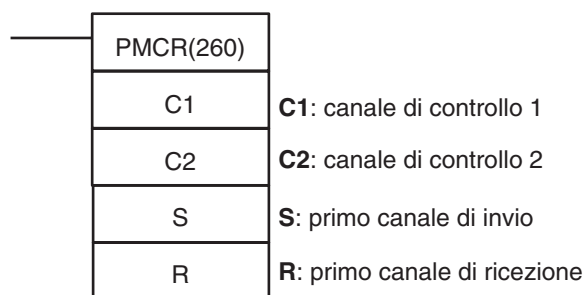
| Istruzioni | Modalità | Porte di comunicazione | |
|--------------------------|-----------------------------------|--|--|
| TXD(236) e RXD(235) | Senza protocollo (personalizzata) | <p>Porta seriale della CPU o della scheda di comunicazione seriale</p>  <p>TXD(236) e RXD(235) utilizzano porte seriali della CPU o delle schede di comunicazione seriale (versione 1.2 o successiva).</p> | |
| TXDU(256) e RXDU(255) | Senza protocollo (personalizzata) | <p>Porta seriale del modulo di comunicazione seriale (versione 1.2 o successiva)</p> <p>Modulo di comunicazione seriale CPU</p>  | |
| PMCR(260) | Protocol macro | <p>Scheda di comunicazione seriale (solo serie CS)</p>  | <p>Modulo di comunicazione seriale</p>  |

3-24-2 PROTOCOL MACRO: PMCR(260)

Scopo

Richiama ed esegue una sequenza di comunicazione registrata in un modulo di comunicazione seriale o in una scheda di comunicazione seriale (solo serie CS).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | PMCR(260) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @PMCR(260) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

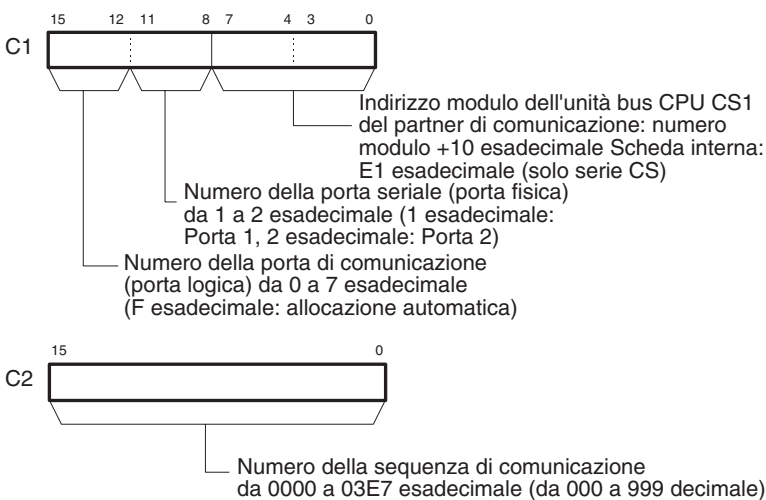
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C1: canale di controllo 1 e C2: canale di controllo 2

I contenuti dei due canali di controllo sono mostrati di seguito.

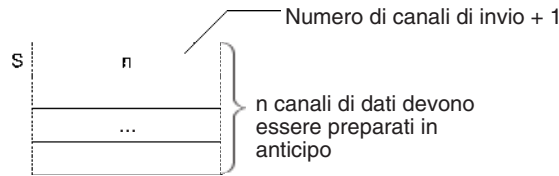


Nota Per informazioni dettagliate sull'utilizzo dell'allocazione automatica del numero della porta di comunicazione (porta logica), consultare la sezione *Allocazione automatica delle porte di comunicazione* a pagina 979.

S: primo canale di invio e area di invio

Viene specificato il primo canale dei canali necessari per l'invio dei dati. S contiene il numero di canali da inviare +1 (incluso il canale S) e i dati di invio iniziano in S+1. È possibile inviare i canali compresi tra 0000 e 00FA esadecimale (tra 0 e 250 decimale).

Se non vi è alcun operando specificato nella sequenza di esecuzione, ad esempio un canale diretto o collegato, specificare per S la costante #0000. Se è specificato l'indirizzo di un canale o un registro, i dati nel canale o nel registro devono sempre essere 0000. Se è stata specificata un'altra costante o un altro indirizzo di canale, si verificherà un errore, verrà attivato il flag di errore e l'istruzione PMCR(260) non verrà eseguita.



R: primo canale di ricezione e area di ricezione

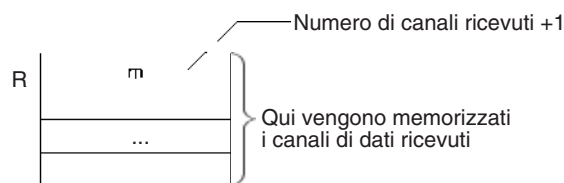
I dati ricevuti vengono memorizzati automaticamente nei canali cominciando da R+1 e il numero di canali ricevuti più R (ossia includendo R) viene scritto automaticamente in R, tra 0000 e 00FA esadecimale (tra 0 e 250 decimale).

Impostazione prima dell'esecuzione di PMCR

Impostare i dati specificati da m (cominciando con D) come dati iniziali per il buffer di ricezione (dati di backup per errori di ricezione). È possibile impostare i dati m con un valore compreso tra 0002 e 00FA (esadecimale) (tra 2 e 255). Se per m è stato specificato il valore 0000 (esadecimale) o 0001 (esadecimale), il valore iniziale del buffer di ricezione verrà azzerato (0).

Impostare sempre un indirizzo di canale per R, anche se non vi sono dati da ricevere. Se è stata definita una costante, si verificherà un errore, verrà attivato il flag di errore e l'istruzione PMCR(260) non verrà eseguita. In assenza di dati da ricevere, R non verrà utilizzato e può essere usato per altri scopi.

Se non vi è alcun operando specificato nella sequenza di esecuzione, ad esempio un canale diretto o collegato, specificare per R la costante #0000. Se è specificato l'indirizzo di un canale o un registro, i dati nel canale o nel registro devono sempre essere 0000.



Caratteristiche operando

| Area | C1 | C2 | S | R |
|--------------------------------------|--|----|---|----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | | |
| Area di lavoro | Da W000 a W511 | | | |
| Area bit di ritentività | Da H000 a H511 | | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 a A959 | | | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | | | |
| Area del contatore | Da C0000 a C4095 | | | |
| Area DM | Da D00000 a D32767 | | | |
| Area EM senza banco | Da E00000 a E32767 | | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | |

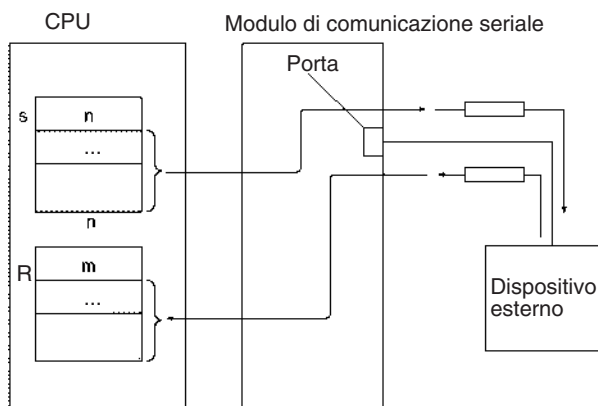
| Area | C1 | C2 | S | R |
|--|--|---|-----------------|---|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | |
| Costanti | Solo valori specificati | Da 0000 a 03E7 esadecimale (da 0 a 999) | #0000 (binario) | |
| Registri dati | Da DR0 a DR15 | | --- | |
| Registri indice | --- | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | |

Descrizione

PMCR(260) eseguirà la sequenza di comunicazione specificata in C2 utilizzando la porta logica definita nei bit da 12 a 15 di C1 e la porta fisica definita nei bit da 8 a 11 di C1 per l'indirizzo del modulo specificato nei bit da 0 a 7 di C1.

Se come operando di un messaggio di invio è stato specificato un simbolo, verrà utilizzata come area di invio il numero di canali di invio impostati in S. Se come operando di un messaggio di ricezione è stato specificato un simbolo, i dati da ricevere verranno inseriti nella memoria cominciando con R+1 e il numero di canali ricevuti verrà scritto automaticamente in R se la trasmissione si è svolta correttamente.

Se la trasmissione fallisce, i dati (da R+1 in avanti) impostati prima dell'esecuzione dell'istruzione PMCR(260) verranno letti dal buffer di ricezione e memorizzati di nuovo da R+1 in avanti.

**Flag**

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | <p>ON se il flag di abilitazione porte di comunicazione è OFF per la porta logica specificata quando viene eseguita l'istruzione PMCR(260).</p> <p>ON se C1 non è incluso negli intervalli specificati. (Il flag di errore non verrà attivato se i dati C2 non sono inclusi negli intervalli specificati. Il codice di fine verrà memorizzato nel codice di completamento per le porte di comunicazione (da A203 a A210) dell'area ausiliaria).</p> <p>ON se il numero di canali di S o R supera 249 (quando vengono specificati i canali).</p> <p>OFF in tutti gli altri casi.</p> |

Avvertenze

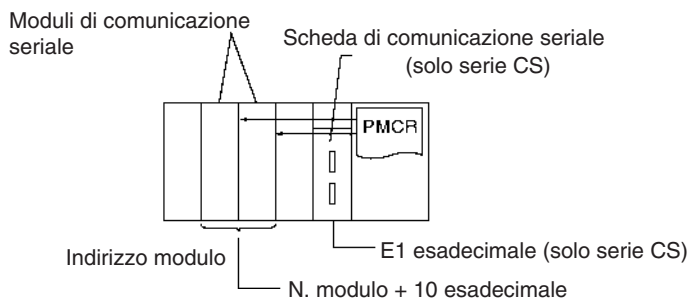
I dati nell'area di invio specificata con S vengono inviati tramite il simbolo opzione di lettura R () in un messaggio di invio.

I dati vengono effettivamente ricevuti nell'area di ricezione specificata da R utilizzando il simbolo opzione di scrittura W () in un messaggio di ricezione.

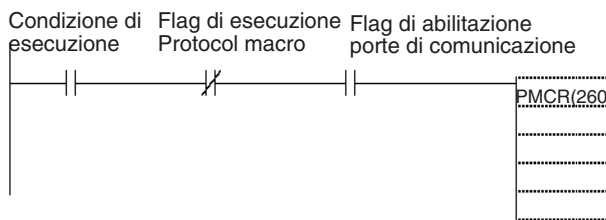
Per informazioni sulle procedure per la designazione dei simboli R () e W (), consultare il *CX-Protocol Operation Manual (W344)*.

È possibile eseguire l'istruzione PMCR(260) per una porta di comunicazione seriale presente su una scheda di comunicazione seriale (solo serie CS) o su un modulo di comunicazione seriale. È possibile montare sul rack CPU o sui rack I/O di espansione fino a 16 moduli di comunicazione seriale. L'indirizzo del modulo del partner di comunicazione deve essere impostato nei bit da 0 a 7 di C1 per specificare il modulo o la scheda da utilizzare e occorre impostare il numero di porta seriale nei bit da 8 a 11. Gli indirizzi dei moduli sono indicati nella tabella riportata di seguito.

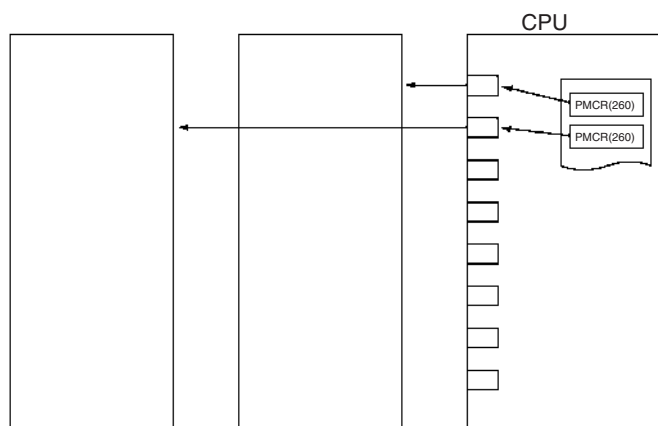
| Modulo/scheda | Indirizzo modulo |
|---|-----------------------------------|
| Scheda di comunicazione seriale (solo serie CS) | E1 esadecimale |
| Modulo di comunicazione seriale | Numero di modulo + 10 esadecimale |



All'avvio dell'esecuzione di PMCR(260) verrà attivato il rispettivo flag di esecuzione Protocol macro, il quale verrà disattivato dopo avere completato la sequenza di comunicazione e la scrittura dei dati nell'area di ricezione specificata. Ogni volta che si esegue un'istruzione PMCR(260), come parte della condizione di esecuzione si dovrebbe utilizzare un ingresso NC per il rispettivo flag di esecuzione Protocol macro, in modo da essere sicuri che verrà eseguita una sola sequenza di comunicazione alla volta per la stessa porta fisica. Di seguito è illustrato un esempio.



Anche le istruzioni SEND(090), RECV(098) e CMND(490) utilizzano le porte logiche da 0 a 7 per l'esecuzione delle sequenze di comunicazione tramite moduli e schede di comunicazione seriale (internamente mediante i comandi FINS). Non è possibile eseguire PMCR(260) per una porta logica già utilizzata da SEND(090), RECV(098), CMND(490) o PMCR(260). Per evitare di eseguire più sequenze di comunicazione per la stessa porta logica, il flag di abilitazione porte di comunicazione (da A20200 a A20207) dovrebbe essere utilizzato come ingresso NO nella condizione di esecuzione per l'istruzione PMCR(260), come mostrato nello schema precedente.



Il flag di errore verrà attivato nei seguenti casi:

- se il flag di abilitazione porte di comunicazione è OFF per la porta logica specificata (da 0 a 7) quando è stata eseguita l'istruzione PMCR(260);
- se C1 non è incluso negli intervalli specificati.

Designazione dell'area di ricezione

Prima di eseguire PMCR(260), è necessario impostare i dati di backup nell'area di ricezione per gli errori di elaborazione ricezione. Dopo l'esecuzione di PMCR(260), i dati verranno automaticamente memorizzati nel buffer di ricezione nell'area di ricezione. Un esempio di applicazione dei dati di backup: un determinato valore (dati di backup) viene impostato anticipatamente in modo che il valore attuale non venga letto come zero quando si verifica un errore di trasmissione mentre viene eseguito il protocollo per la lettura del valore attuale di un controller.

Canali e flag correlati

Nell'esecuzione dell'istruzione PMCR(260) è possibile utilizzare i seguenti flag e canali in base alle esigenze.

Area ausiliaria

| Nome | Indirizzo | Contenuto |
|---|---------------------|---|
| Flag di abilitazione porte di comunicazione | Da A20200 ad A20207 | ON quando sono abilitate le comunicazioni di rete (inclusa PMCR(260)). I bit da 00 a 07 corrispondono rispettivamente alle porte logiche da 0 a 7. All'avvio della comunicazione di rete verrà disattivato un flag di abilitazione porte di comunicazione, il quale verrà poi attivato quando tale comunicazione è stata completata, indipendentemente dal fatto che la comunicazione sia terminata regolarmente o con un errore. |

| Nome | Indirizzo | Contenuto |
|---|---------------------|---|
| Flag di errore porte di comunicazione | Da A21900 ad A21907 | ON quando si verifica un errore nella comunicazione di rete. I bit da 00 a 07 corrispondono rispettivamente alle porte logiche da 0 a 7. Lo stato del flag verrà mantenuto fino all'avvio successivo della comunicazione di rete. Quando si riavvia la comunicazione, il flag verrà disattivato, anche se si è verificato un errore nell'ultima esecuzione. |
| Codici di completamento per le porte di comunicazione | Da A203 ad A210 | Contiene il codice di completamento memorizzato durante l'esecuzione della comunicazione di rete. I canali da A203 ad A210 corrispondono rispettivamente alle porte logiche da 0 a 7. Durante l'esecuzione dell'istruzione di comunicazione, il codice di completamento sarà 00. Quando l'esecuzione è stata completata, verrà memorizzato il nuovo codice di risposta. Il contenuto di questi canali viene azzerato quando inizia l'operazione. |

Risposte di comunicazione

| Codice | Contenuto |
|--------------------|--|
| 1106 (esadecimale) | Nessun numero di programma corrispondente Non è stato registrato il numero della sequenza di invio/ricezione specificato. Modificare il numero della sequenza di invio/ricezione o aggiungere tale numero mediante CX-Programmer. |
| 2201 (esadecimale) | Non utilizzabile a causa dell'esecuzione di una protocol macro Poiché è già stata eseguita una protocol macro, non viene accettata nessuna esecuzione ulteriore. Aggiungere al programma una condizione NC per il flag di esecuzione Protocol macro. |
| 2202 (esadecimale) | Non utilizzabile a causa di un'interruzione Poiché il protocollo è in corso di attivazione, non viene accettata nessuna esecuzione ulteriore. Aggiungere al programma una condizione NC per il flag di modifica impostazione seriale. |
| 2401 (esadecimale) | Nessuna tabella di registrazione Si è verificato un errore nei dati della protocol macro oppure è in corso la trasmissione dei dati. Trasmettere i dati della protocol macro utilizzando CX-Programmer. |
| Altri | Per informazioni sui codici di risposta, fare riferimento al manuale <i>CS/CJ-series Communications Commands Reference Manual (W342)</i> . |

Area delle schede interne (solo serie CS)

| Nome | Indirizzo | Contenuto |
|---|------------|---|
| Flag di esecuzione Protocol macro porta 1 | CIO 190915 | ON quando viene eseguita l'istruzione PMCR(260). Se l'esecuzione fallisce, il flag resta OFF. Il flag verrà disattivato quando la sequenza di comunicazione è stata completata (terminata o interrotta). |
| Flag di esecuzione Protocol macro porta 2 | CIO 191915 | |

Area unità bus CPU

$n = 1500 + 25 \times \text{numero di modulo}$

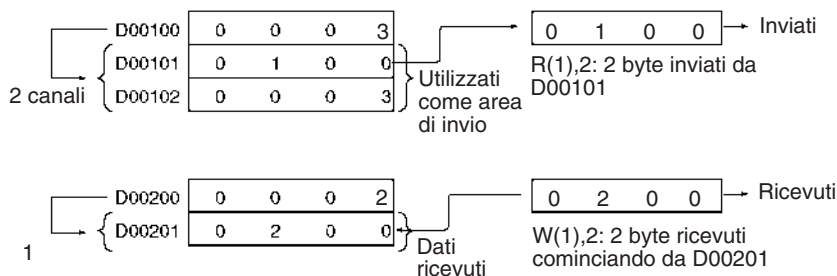
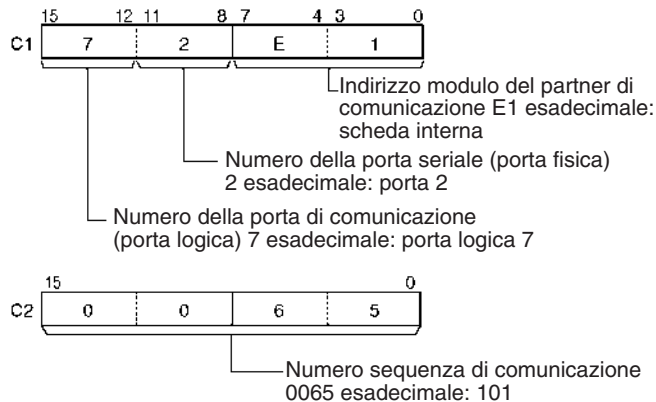
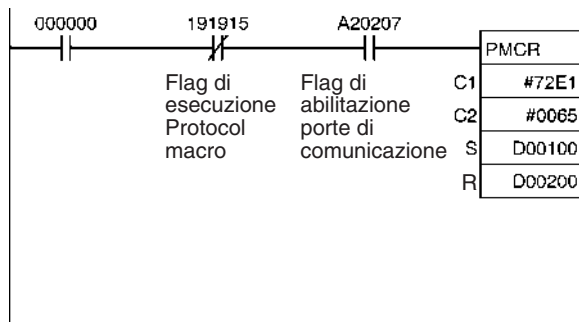
| Nome | Indirizzo | Contenuto |
|---|--------------------|--|
| Flag di esecuzione Protocol macro porta 1 | Bit 15 di CIO n+9 | ON quando viene eseguita l'istruzione PMCR(260). Se l'esecuzione fallisce, il flag resta OFF. Il flag verrà disattivato quando la sequenza di comunicazione è stata completata (terminata o interrotta). |
| Flag di esecuzione Protocol macro porta 2 | Bit 15 di CIO n+19 | |

Esempi

Nell'esempio seguente, quando CIO 0000 è impostato su ON, la sequenza di comunicazione n. 101 (0065 esadecimale) verrà eseguita finché il flag di abilitazione porte di comunicazione resta attivato per la porta 7 (A20207) e il flag di esecuzione porta 1 (CIO 190915) è disattivato.

Se in un messaggio di invio è stato specificato un operando per il simbolo, verranno utilizzati come area di invio 2 canali di dati cominciando da D00101 (poiché il contenuto di D00100 è #0003).

Se in un messaggio di ricezione è stato specificato un operando per il simbolo, verranno memorizzati 2 canali di dati cominciando da D00201 e il numero di canali ricevuti +1 verrà scritto in D00200.



Nota Come mostrato sopra, il simbolo opzione di lettura R() nel messaggio di invio o il simbolo opzione di scrittura W() invia/riceve realmente i dati.

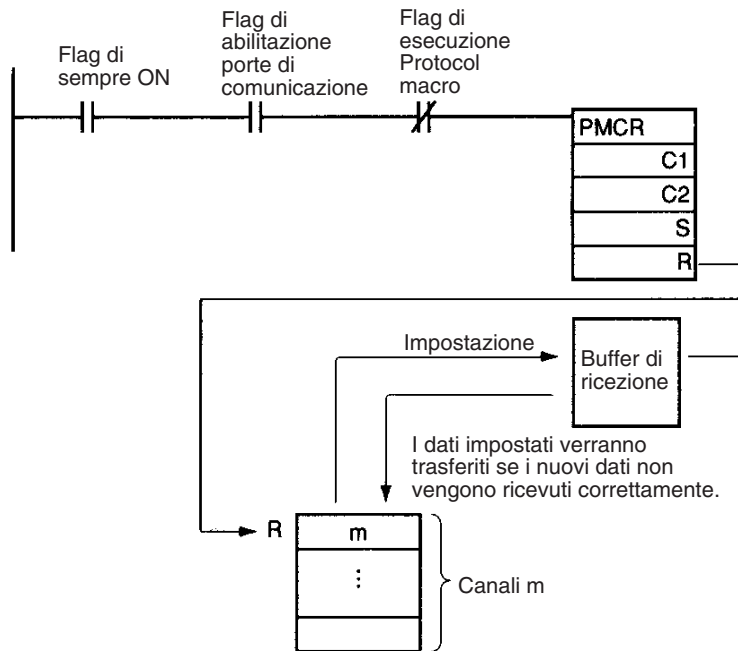
Mantenimento dell'area di ricezione

Subito prima dell'esecuzione di una sequenza di comunicazione per PMCR(260), il buffer di ricezione viene azzerato. Se per leggere periodicamente i dati del valore attuale o altri dati si utilizza una programmazione come quella indicata di seguito, e non è possibile leggere i dati a causa di un errore di ricezione o per altri motivi, i dati in corso di lettura verranno azzerati fino alla successiva lettura riuscita.

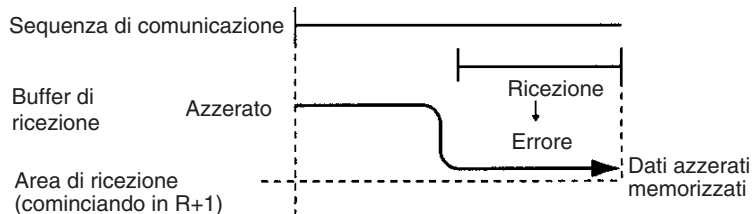
Per conservare i dati nell'area di ricezione anche quando si verifica un errore di ricezione, viene fornita una funzione. Utilizzando questa funzione, i dati verranno trasferiti dai primi canali m dell'area di ricezione al buffer di ricezione, dopo che il buffer è stato azzerato ma prima dell'esecuzione della sequenza di comunicazione. In questo modo si evita che l'area di ricezione venga temporaneamente azzerata scrivendo i dati di ricezione più recenti quando non si riesce ad ottenere i nuovi dati di ricezione.

Specificare il numero di canali dell'area di ricezione da mantenere come valore m. Se viene specificato 0 o 1, la funzione di ritenzione verrà disabilitata e l'area di ricezione verrà completamente azzerata.

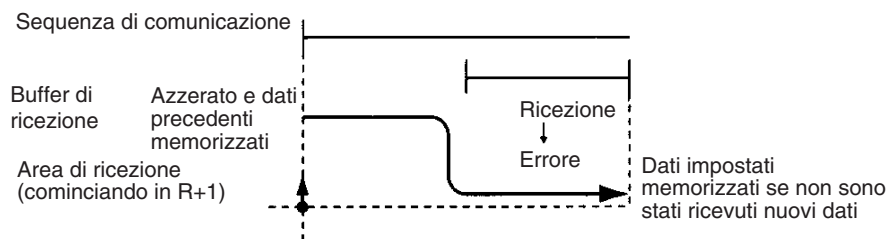
Nell'esempio di programmazione seguente vengono mostrate le istruzioni utilizzate per eseguire costantemente o periodicamente PMCR(260) per la lettura dei dati tramite una singola operazione di ricezione.



Area di ricezione non mantenuta



Area di ricezione mantenuta

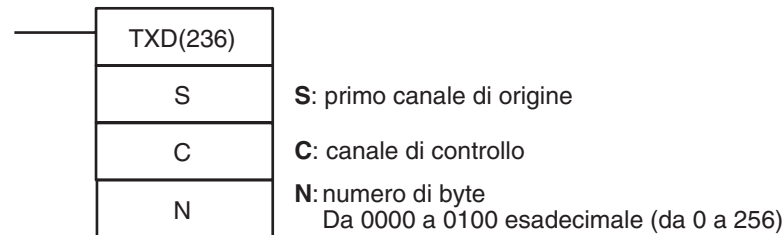


3-24-3 TRANSMIT: TXD(236)

Scopo

Invia il numero di byte di dati specificato dalla porta RS-232C integrata nella CPU o da una delle porte seriali della scheda di comunicazione seriale. (La scheda di comunicazione seriale deve essere della versione 1.2 o successiva).

Simbolo programmazione ladder



Variazioni

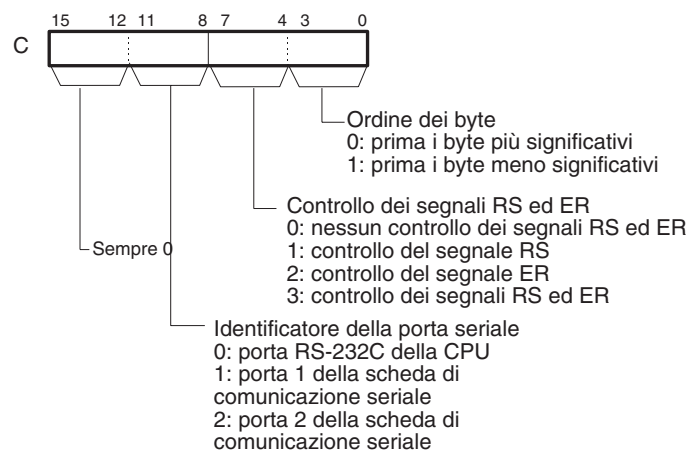
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | TXD(236) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TXD(236) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Il contenuto del canale di controllo C è mostrato di seguito.



Caratteristiche operando

| Area | S | C | N |
|-------------------------|-----------------------------------|---|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 a A959 | | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |

| Area | S | C | N |
|--|--|-------------------------|--|
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | Da #0000 a #0100 (binario) o da &0 a &256 (decimale) |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 | | |

Descrizione

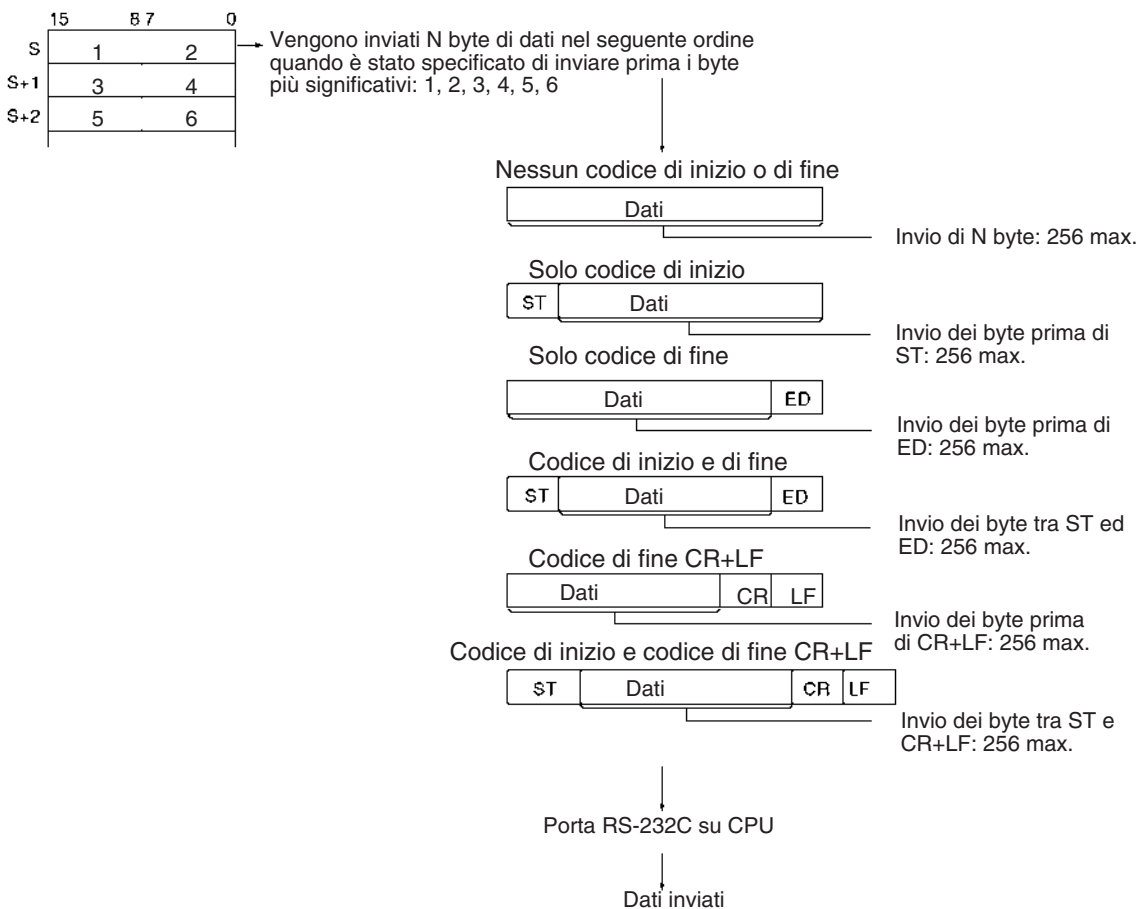
TXD(236) legge N byte di dati dai canali da S a S+(N+2)-1 e invia i dati grezzi in modalità nessun protocollo dalla porta RS-232C integrata nel modulo CPU o da una delle porte seriali della scheda di comunicazione seriale. (La porta di uscita è specificata con i bit 8 - 11 di C).

Prima che i dati vengano emessi, vengono aggiunti i codici di inizio e di fine specificati per la modalità nessun protocollo. I codici di inizio e di fine sono specificati nella configurazione del PLC (per la porta RS-232C della CPU) o nell'area di configurazione DM allocata (per le porte della scheda di comunicazione seriale).

I dati possono essere trasmessi solo quando il flag Pronto per Trasmissione della porta è attivato. Il flag Pronto per Trasmissione è A39205 per la porta RS-232C del modulo CPU, A39605 per la porta 1 della scheda di comunicazione seriale, A39613 per la porta 2 della scheda di comunicazione seriale.

È possibile inviare fino a un massimo di 259 byte, compresi i dati di trasmissione (N = 256 byte max.), il codice di inizio e il codice di fine.

Nello schema seguente è illustrato l'ordine in cui verranno inviati i dati e il contenuto del frame di invio per varie impostazioni del codice di inizio e di fine.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se la porta RS-232C della CPU è specificata come porta di invio, ma nella configurazione del PLC non è stata impostata alcuna modalità nessun protocollo.</p> <p>ON se una delle porte seriali della scheda di comunicazione seriale è specificata come porta di invio, ma nell'area di configurazione DM non è stata impostata alcuna modalità nessun protocollo.</p> <p>ON se il valore di C non è incluso nell'intervallo.</p> <p>ON se il valore di N non è compreso tra 0000 e 0100 esadecimale.</p> <p>ON se viene tentato un invio quando il flag Pronto per Trasmissione è impostato su OFF. Il flag Pronto per Trasmissione è A39205 per la porta RS-232C del modulo CPU, A39605 per la porta 1 della scheda di comunicazione seriale, A39613 per la porta 2 della scheda di comunicazione seriale.</p> <p>ON (flag ER nei task ad interrupt) se nel task ciclico è in corso l'esecuzione di un'istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale, se il task ciclico viene interrotto e viene eseguita un'altra istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale nel task ad interrupt (vedere nota).</p> <p>Nota Il flag di errore (ER) verrà attivato immediatamente dopo un'altra istruzione TXD(236) o RXD(235) nel task ad interrupt.</p> <p>OFF in tutti gli altri casi.</p> |

Avvertenze

È possibile utilizzare TXD(236) soltanto per la porta RS-232C della CPU o per una delle porte seriali della scheda di comunicazione seriale. Inoltre, per la porta deve essere impostata la modalità nessun protocollo.

È possibile impostare il seguente formato di frame per i messaggi di invio nella configurazione del PLC (per la porta RS-232C della CPU) o nell'area di configurazione DM allocata (per le porte della scheda di comunicazione seriale).

- Codice di inizio: nessuno o da 00 a FF esadecimale.
- Codice di fine: nessuno, CR+LF o da 00 a FF esadecimale.

I dati verranno inviati con qualsiasi codice di inizio e/o di fine specificato nella configurazione del PLC o nell'area di configurazione DM allocata. Se sono stati specificati i codici di inizio e fine, tali codici verranno aggiunti ai dati da inviare (N). In questo caso, il numero massimo di byte che è possibile specificare per N è 256 byte.

I dati possono essere trasmessi solo quando il flag Pronto per Trasmissione della porta è attivato. Il flag Pronto per Trasmissione è A39205 per la porta RS-232C del modulo CPU, A39605 per la porta 1 della scheda di comunicazione seriale, A39613 per la porta 2 della scheda di comunicazione seriale.

I dati vengono inviati nell'ordine specificato in C.

Se per N è stato specificato 0, non verrà eseguito alcun invio.

Se in C è stato specificato il controllo del segnale RS, come segnale RS verrà utilizzato il bit 15 di S.

Se in C è stato specificato il controllo del segnale ER, come segnale ER verrà utilizzato il bit 15 di S.

Se in C è stato specificato il controllo dei segnali RS ed ER, come segnale RS verrà utilizzato il bit 15 di S e come segnale ER verrà utilizzato il bit 14 di S.

Se per il controllo dei segnali RS ed ER in C è stato specificato 1, 2 o 3 esadecimale, l'istruzione TXD(236) verrà eseguita senza tener conto dello stato del flag Pronto per Trasmissione (A39205, A39605 o A39613, in base alla porta utilizzata).

Se l'istruzione TXD(236) viene eseguita per una scheda che non supporta la modalità nessun protocollo, ad esempio una scheda di comunicazione seriale senza un numero di versione, verranno attivati i flag di disabilitazione servizio scheda interna (A42404) e il flag di errore.

Si verificherà un errore e il flag di errore verrà attivato nei seguenti casi:

- La porta RS-232C della CPU è stata specificata, ma nella configurazione del PLC non è stata impostata alcuna modalità nessun protocollo.
- Una delle porte seriali della scheda di comunicazione seriale è stata specificata, ma nell'area di configurazione DM allocata non è stata impostata alcuna modalità nessun protocollo per tale porta.
- È stata specificata una delle porte seriali della scheda di comunicazione seriale ma la scheda non supporta la modalità nessun protocollo, ovvero non ha un numero di versione.
- Il valore di C non è incluso nell'intervallo.
- Il valore di N non è compreso tra 0000 e 0100 esadecimale.
- È stato tentato un invio quando il flag Pronto per Trasmissione era impostato su OFF. Il flag Pronto per Trasmissione è A39205 per la porta RS-232C del modulo CPU, A39605 per la porta 1 della scheda di comunicazione seriale, A39613 per la porta 2 della scheda di comunicazione seriale.
- Nel task ciclico era in corso l'esecuzione di un'istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale, il task ciclico è stato interrotto ed è stata eseguita un'altra istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale nel task ad interrupt.

Nota Non si deve programmare TXD(236)/RXD(235) per la porta (porta 1 o 2) di una scheda di comunicazione seriale sia nel task ciclico che nel task ad interrupt. Non è possibile eseguire un'istruzione TXD(236)/RXD(235) per la scheda di comunicazione seriale nel task ad interrupt se è in corso l'esecuzione di un'istruzione TXD(236)/RXD(235) per la scheda di comunicazione seriale nel task ciclico. Se si esegue un'istruzione TXD(236)/RXD(235) per la scheda di comunicazione seriale nel task ad interrupt mentre per tale scheda è in corso l'esecuzione di un'istruzione TXD(236)/RXD(235) nel task ciclico, si verificherà un errore e il flag ER verrà attivato. Non è possibile programmare queste istruzioni sia nel task ciclico che nel task ad interrupt, neppure se vengono eseguite per porte diverse della scheda di comunicazione seriale.

Canali e flag correlati

Quando si esegue l'istruzione TXD(236), è possibile utilizzare le seguenti impostazioni per la configurazione del PLC e per il flag dell'area ausiliaria.

Impostazioni di configurazione del PLC per la porta RS-232C della CPU

| Indirizzo della console di programmazione | | Nome | Impostazioni |
|---|-----------|---|---|
| Canale | Bit | | |
| 162 | Da 0 a 15 | Ritardo invio in modalità nessun protocollo | Da 0000 a 210F esadecimale, da 0 a 99.990 ms decimali (in unità di 10 ms) |
| 164 | Da 8 a 15 | Codice di inizio in modalità nessun protocollo | Da 00 a FF esadecimale |
| | Da 0 a 7 | Codice di fine in modalità nessun protocollo | Da 00 a FF esadecimale |
| 165 | 12 | Identificatore codice di inizio in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di inizio. |
| | 8 e 9 | Identificatore codice di fine in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di fine. 2: utilizzo di CR+LF. |
| | Da 0 a 7 | Numero di byte di dati in modalità nessun protocollo | 00: 256 byte Da 01 a FF: da 1 a 255 byte |

Impostazioni dell'area di configurazione DM per le porte della scheda di comunicazione seriale

| Canale area di configurazione | | Bit | Nome | Impostazioni |
|-------------------------------|---------|------------|---|---|
| Porta 1 | Porta 2 | | | |
| D32002 | D32012 | 15 | Identificatore ritardo invio in modalità nessun protocollo | 0: impostazione predefinita (0 ms) 1: utilizzare il ritardo nei bit da 1 a 14. |
| | | Da 0 a 14 | Tempo di ritardo invio in modalità nessun protocollo | Da 0000 a 7530 esadecimale Da 0 a 300.000 ms decimali (in unità di 10 ms) |
| D32004 | D32014 | Da 8 a 15 | Codice di inizio in modalità nessun protocollo | Da 00 a FF esadecimale |
| | | Da 0 a 7 | Codice di fine in modalità nessun protocollo | Da 00 a FF esadecimale |
| D32005 | D32015 | Da 12 a 15 | Identificatore codice di inizio in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di inizio. |
| | | Da 8 a 11 | Identificatore codice di fine in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di fine. 2: utilizzo di CR+LF. |

Area ausiliaria

Flag Pronto per Trasmissione

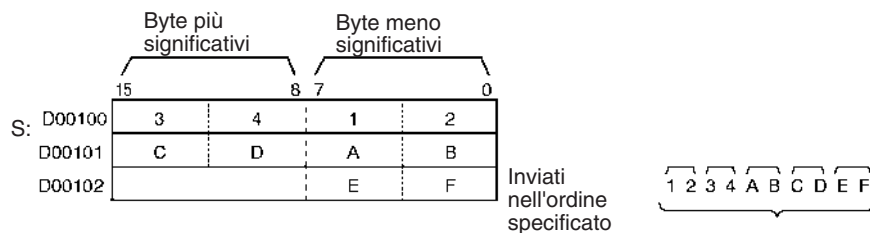
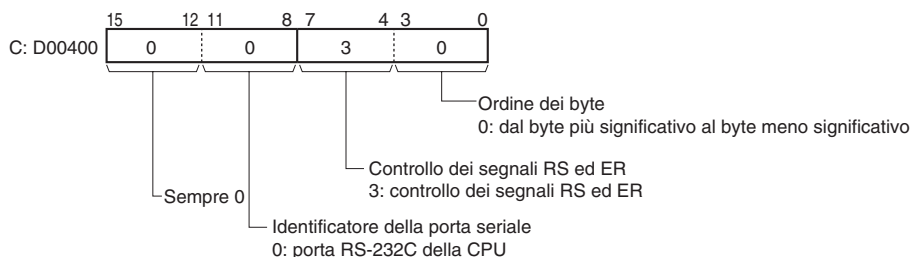
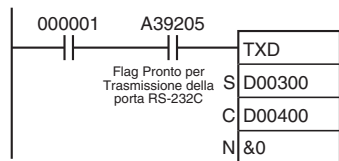
| Porta | Indirizzo | Contenuto |
|---|------------------|---|
| Porta RS-232C integrata nell'unità bus CPU | A39205 | ON quando è possibile inviare i dati in modalità nessun protocollo. |
| Porta 1 della scheda di comunicazione seriale | A39605 | |
| Porta 2 della scheda di comunicazione seriale | A39613 | |

Flag scheda interna per la scheda di comunicazione seriale (porte 1 e 2)

| Nome | Indirizzo | Contenuto |
|---|------------------|---|
| Flag di disabilitazione servizio scheda interna | A42404 | ON quando l'istruzione TXD(236) viene eseguita per una scheda di comunicazione seriale che non supporta la modalità nessun protocollo, ovvero una scheda senza un numero di versione. |

Esempio: invio di dati

Nell'esempio seguente, quando CIO 000001 e il flag Pronto per Trasmissione della porta RS-232C (A39205) sono impostati su ON, il segnale RS viene impostato in base allo stato di D00300 bit 15, mentre il segnale ER viene impostato in base allo stato di D00300 bit 14.



Codici di inizio e fine aggiunti in base all'impostazione nella configurazione del PLC (questo esempio presume che entrambi i codici di inizio e fine siano stati definiti).

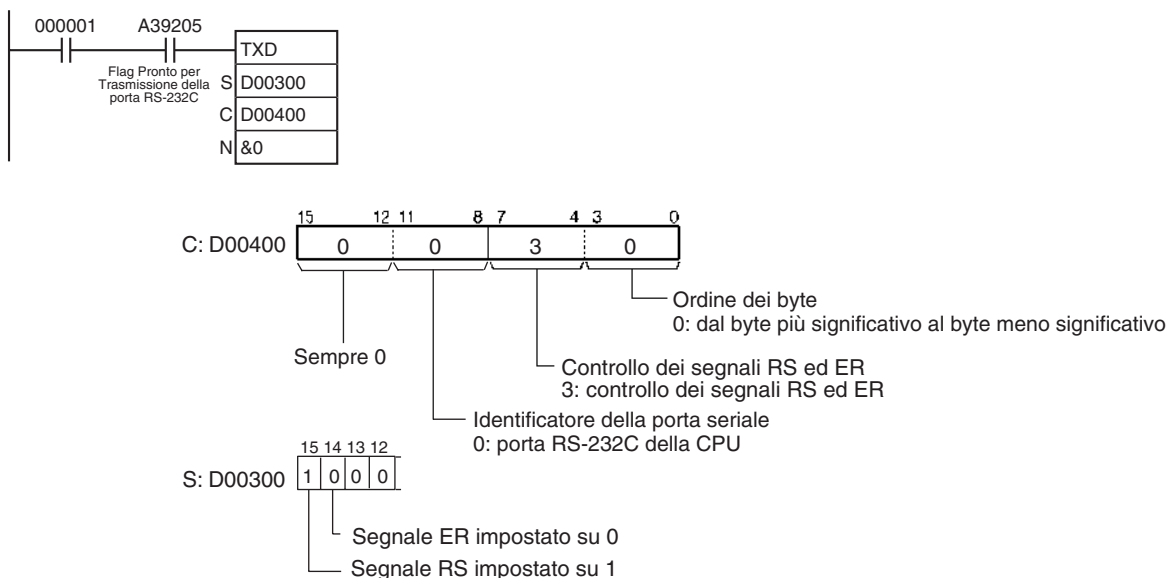


ST: codice di inizio (ad esempio, 02 esadecimale)
ED: codice di fine (ad esempio, 03 esadecimale)

Inviati

Esempio: esecuzione del controllo dei segnali

Nell'esempio seguente, quando CIO 000001 e il flag Pronto per Trasmissione della porta RS-232C (A39205) sono impostati su ON, il segnale RS viene impostato in base allo stato di D00300 bit 15, mentre il segnale ER viene impostato in base allo stato di D00300 bit 14.

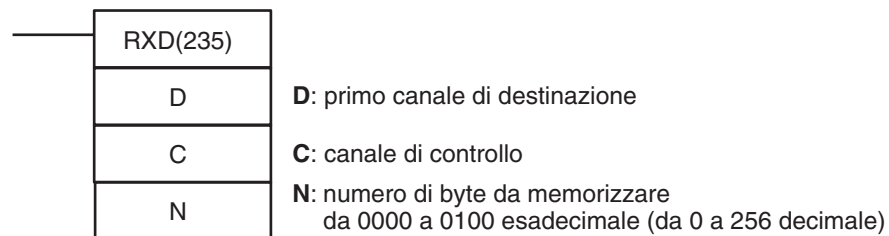


3-24-4 RECEIVE: RXD(235)

Scopo

Legge il numero di byte di dati specificato dalla porta RS-232C integrata nella CPU o da una delle porte seriali della scheda di comunicazione seriale. (La scheda di comunicazione seriale deve essere della versione 1.2 o successiva).

Simbolo programmazione ladder



Variazioni

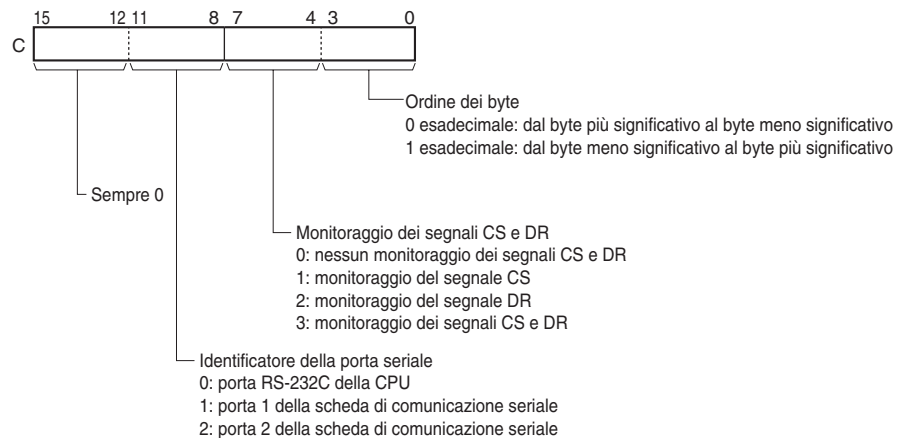
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RXD(235) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RXD(235) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

Il contenuto del canale di controllo C è mostrato di seguito.



Caratteristiche operando

| Area | D | C | N |
|--|--|-----------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A448 a A959 | Da A000 ad A447 Da A448 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | Da #0000 a #0100 (binario) o da &0 a &256 (decimale) |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

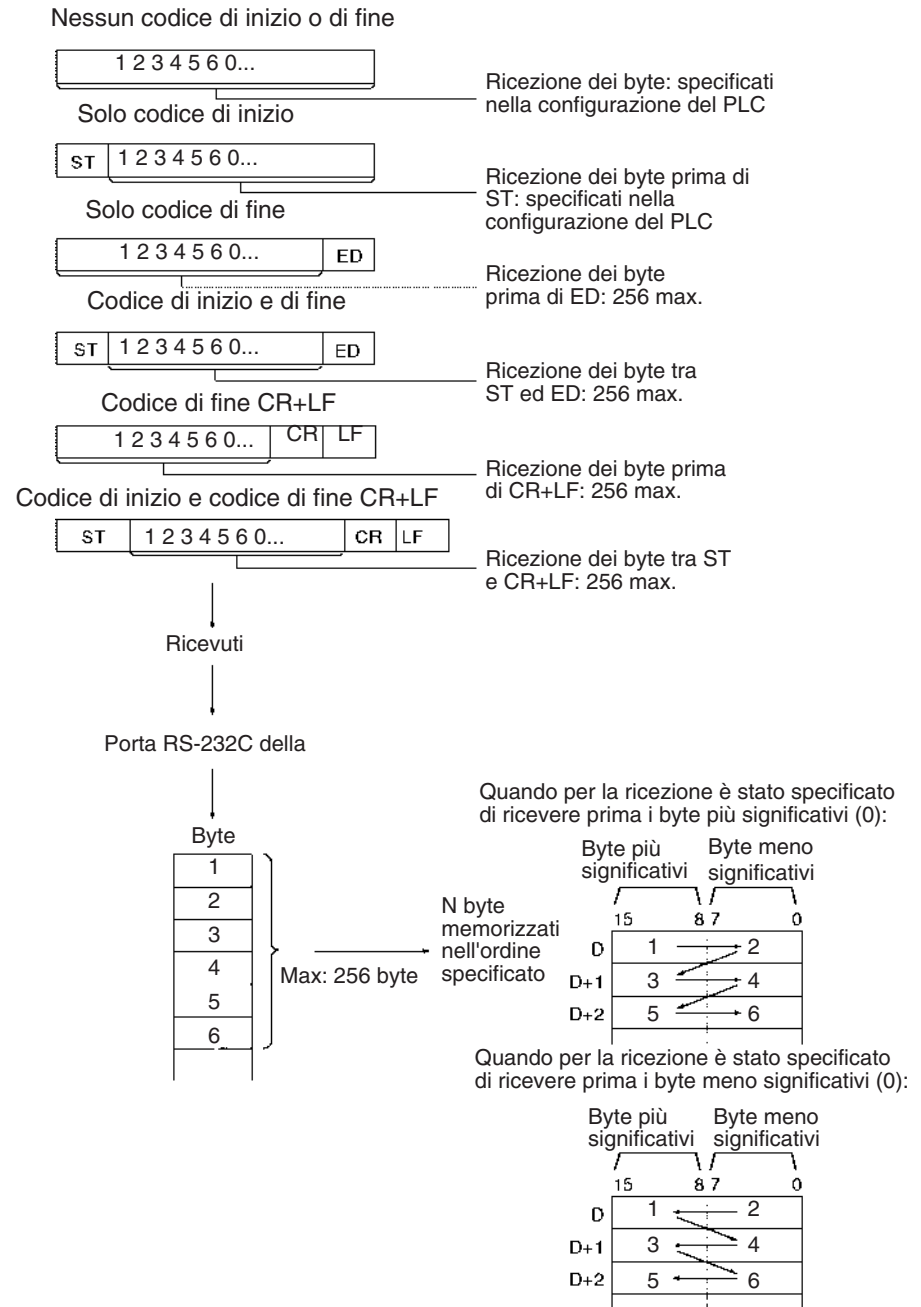
Descrizione

RXD(235) legge i dati ricevuti in modalità nessun protocollo sulla porta RS-232C integrata del modulo CPU o su una delle porte seriali della scheda di comunicazione seriale (la porta è specificata con i bit da 8 a 11 di C), quindi memorizza N byte di dati nei canali D - D+(N÷2)-1. Se N byte di dati non vengono ricevuti sulla porta, saranno memorizzati unicamente i dati effettivamente ricevuti.

I dati possono essere ricevuti solo quando il flag Pronto per Ricezione della porta è attivato. Il flag Pronto per Ricezione è A39206 per la porta RS-232C del modulo CPU, A39606 per la porta 1 della scheda di comunicazione seriale, A39614 per la porta 2 della scheda di comunicazione seriale. Eseguire RXD(235) solo quando il relativo flag Pronto per Ricezione è attivato.

È possibile ricevere fino a un massimo di 259 byte, compresi i dati di ricezione (N = 256 byte max.), il codice di inizio e il codice di fine.

Nello schema seguente è illustrato l'ordine in cui verranno ricevuti i dati e il contenuto del frame di ricezione per varie impostazioni del codice di inizio e di fine.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se la porta RS-232C della CPU è specificata come porta di invio, ma nella configurazione del PLC non è stata impostata alcuna modalità nessun protocollo.</p> <p>ON se una delle porte seriali della scheda di comunicazione seriale è specificata come porta di invio, ma nell'area di configurazione DM non è stata impostata alcuna modalità nessun protocollo.</p> <p>ON se il valore di C non è incluso nell'intervallo.</p> <p>ON se il valore di N non è compreso tra 0000 e 0100 esadecimale.</p> <p>ON se viene tentato un invio quando il flag Pronto per Trasmissione è impostato su OFF. Il flag Pronto per Trasmissione è A39205 per la porta RS-232C del modulo CPU, A39605 per la porta 1 della scheda di comunicazione seriale, A39613 per la porta 2 della scheda di comunicazione seriale.</p> <p>ON (flag ER nei task ad interrupt) se nel task ciclico è in corso l'esecuzione di un'istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale, se il task ciclico viene interrotto e viene eseguita un'altra istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale nel task ad interrupt (vedere nota).</p> <p>Nota Il flag di errore (ER) verrà attivato immediatamente dopo un'altra istruzione TXD(236) o RXD(235) nel task ad interrupt.</p> <p>OFF in tutti gli altri casi.</p> |

Avvertenze

È possibile utilizzare RXD(235) soltanto per la porta RS-232C della CPU o per una delle porte seriali della scheda di comunicazione seriale. Inoltre, per la porta deve essere impostata la modalità nessun protocollo.

È possibile impostare il seguente formato di frame per i messaggi di ricezione nella configurazione del PLC (per la porta RS-232C della CPU) o nell'area di configurazione DM allocata (per le porte della scheda di comunicazione seriale).

- Codice di inizio: nessuno o da 00 a FF esadecimale.
- Codice di fine: nessuno, CR+LF o da 00 a FF esadecimale. Se non è stato impostato nessun codice di fine, il numero di byte da ricevere verrà impostato da 00 a FF esadecimale (da 1 a 256 decimale; 00 specifica 256 byte).

Verrà attivato il flag di ricezione avvenuta (nota 1) quando viene ricevuto il numero di byte specificati nella configurazione del PLC (per la porta RS-232C della CPU) o nell'area di configurazione DM allocata (per le porte della scheda di comunicazione seriale). Quando viene attivato il flag di ricezione avvenuta, il numero di byte nel contatore di ricezione (nota 2) avrà lo stesso valore del numero di byte ricevuti specificato nella configurazione del PLC o nell'area di configurazione DM allocata. Se vengono ricevuti più byte di quelli specificati, verrà attivato il flag di overflow durante ricezione (nota 3).

Se nella configurazione del PLC o nell'area di configurazione DM allocata viene specificato un codice di fine, verrà attivato il flag di ricezione avvenuta (nota 1) quando viene ricevuto il codice di fine o 256 byte di dati. Se dopo l'attivazione del flag di ricezione avvenuta (nota 1) vengono ricevuti più dati del previsto, verrà attivato il flag di overflow durante ricezione (nota 3).

La ricezione verrà interrotta se vengono ricevuti 259 byte di dati. Se in seguito vengono immessi più dati, verranno attivati i flag di errore di overrun (nota 5) e di errore di trasmissione (nota 6).

Quando nella porta seriale della scheda di comunicazione seriale vengono immessi più dati di quelli specificati in N, questi dati verranno scartati quando viene eseguita l'istruzione RXD(235). Invece l'immissione di ulteriori dati nella porta RS-232C della CPU non verrà scartata quando viene eseguita l'istruzione RXD(235).

Quando l'istruzione RXD(235) viene eseguita, i dati vengono memorizzati nella memoria cominciando in D, il flag di ricezione avvenuta (nota 1) verrà disattivato (anche se il flag di overflow durante ricezione (nota 3) è ON) e il contatore di ricezione (nota 2) verrà reimpostato su 0.

Con la porta RS-232C integrata nella CPU, se viene attivato il bit di riavvio della porta RS-232C (nota 4), il flag di ricezione avvenuta (nota 1) verrà disattivato (anche se il flag di overflow durante ricezione è ON) e il contatore di ricezione (nota 2) verrà reimpostato su 0.

I dati verranno archiviati in memoria nell'ordine specificato in C.

Se per N è specificato 0, verrà disattivato il flag di ricezione avvenuta (nota 1), il contatore di ricezione (nota 2) verrà reimpostato su 0 e in memoria non verrà archiviato nulla.

Se in C è stato specificato il monitoraggio del segnale CS, lo stato del segnale CS verrà memorizzato nel bit 15 di D.

Se in C è stato specificato il monitoraggio del segnale DR, lo stato del segnale DR verrà memorizzato nel bit 15 di D.

Se in C è stato specificato il monitoraggio dei segnali CS e DR, lo stato del segnale CS verrà memorizzato nel bit 15 di D e lo stato del segnale DR verrà memorizzato nel bit 14 di D.

Se non è stato specificato il monitoraggio del segnale CS o DR, i dati ricevuti non verranno memorizzati.

Se per il controllo dei segnali RS ed ER in C è stato specificato 1, 2 o 3 esadecimale, l'istruzione RXD(235) verrà eseguita senza tener conto dello stato del flag di ricezione avvenuta (nota 1).

Se l'istruzione RXD(235) viene eseguita per una scheda che non supporta la modalità nessun protocollo, ad esempio una scheda di comunicazione seriale senza un numero di versione, verranno attivati i flag di disabilitazione servizio scheda interna (A42404, errore non fatale) e il flag di errore.

- Nota**
1. Flag di ricezione avvenuta

| | |
|--|--------|
| Porta RS232C integrata | A39206 |
| Porta 1 della scheda di comunicazione seriale: | A35606 |
| Porta 2 della scheda di comunicazione seriale: | A35614 |
 2. Contatori di ricezione

| | |
|--|------|
| Porta RS232C integrata | A393 |
| Porta 1 della scheda di comunicazione seriale: | A357 |
| Porta 2 della scheda di comunicazione seriale: | A358 |
 3. Flag di overflow durante ricezione

| | |
|--|--------|
| Porta RS232C integrata | A39207 |
| Porta 1 della scheda di comunicazione seriale: | A35607 |
| Porta 2 della scheda di comunicazione seriale: | A35615 |
 4. Bit di riavvio della porta RS-232C

| | |
|------------------------|--------|
| Porta RS232C integrata | A52600 |
|------------------------|--------|
 5. Flag di errore di overrun

| | |
|--|------------|
| Porta 1 della scheda di comunicazione seriale: | CIO 190804 |
| Porta 2 della scheda di comunicazione seriale: | CIO 191804 |
 6. Flag di errore di trasmissione

| | |
|--|------------|
| Porta 1 della scheda di comunicazione seriale: | CIO 190815 |
| Porta 2 della scheda di comunicazione seriale: | CIO 191815 |
 7. Flag di disabilitazione servizio scheda interna

| | |
|--|--------|
| Porte 1 e 2 della scheda di comunicazione seriale: | A42404 |
|--|--------|

Si verificherà un errore e il flag di errore verrà attivato nei seguenti casi:

- La porta RS-232C della CPU è stata specificata, ma nella configurazione del PLC non è stata impostata alcuna modalità nessun protocollo.
- Una delle porte seriali della scheda di comunicazione seriale è stata specificata, ma nell'area di configurazione DM allocata non è stata impostata alcuna modalità nessun protocollo per tale porta.
- È stata specificata una delle porte seriali della scheda di comunicazione seriale ma la scheda non supporta la modalità nessun protocollo, ovvero non ha un numero di versione.
- Il valore di C non è incluso nell'intervallo.
- Il valore di N non è compreso tra 0000 e 0100 esadecimale.
- Nel task ciclico era in corso l'esecuzione di un'istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale, il task ciclico è stato interrotto ed è stata eseguita un'altra istruzione TXD(236) o RXD(235) per la scheda di comunicazione seriale nel task ad interrupt.
- Non è possibile ricevere ulteriori dati finché i dati ricevuti non vengono letti dal buffer tramite RXD(235). Quando si attiva il flag di ricezione avvenuta, i dati vengono immediatamente letti con RXD(235), prima che vengano inviati ulteriori dati alla porta.
- Quando l'istruzione RXD(235) viene utilizzata per leggere i dati ricevuti sulla porta RS-232C della CPU, i dati che rimangono nel buffer di ricezione della porta non vengono eliminati, quindi RXD(235) può essere eseguita ripetutamente per leggere in varie parti un blocco di dati. Invece, quando RXD(235) viene utilizzata per leggere i dati ricevuti su una delle porte della scheda di comunicazione seriale (scheda di comunicazione seriale della versione 1.2 o successiva), il buffer di ricezione della porta viene azzerato dopo l'esecuzione dell'istruzione RXD(235). Pertanto, **non** è possibile eseguire più volte l'istruzione RXD(235) per leggere un blocco di dati suddiviso in più parti.

Canali e flag correlati

Quando si esegue l'istruzione RXD(235), è possibile utilizzare le seguenti impostazioni per la configurazione del PLC e per il flag dell'area ausiliaria.

Impostazioni di configurazione del PLC per la porta RS-232C della CPU

| Indirizzo della console di programmazione | | Nome | Impostazioni |
|---|-----------|---|---|
| Canale | Bit | | |
| 162 | Da 0 a 15 | Ritardo invio in modalità nessun protocollo | Da 0000 a 210F esadecimale, Da 0 a 99.990 ms decimali (in unità di 10 ms) |
| 164 | Da 8 a 15 | Codice di inizio in modalità nessun protocollo | Da 00 a FF esadecimale |
| | Da 0 a 7 | Codice di fine in modalità nessun protocollo | Da 00 a FF esadecimale |
| 165 | 12 | Identificatore codice di inizio in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di inizio. |
| | 8 e 9 | Identificatore codice di fine in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di fine. 2: utilizzo di CR+LF. |
| | Da 0 a 7 | Numero di byte di dati in modalità nessun protocollo | 00: 256 byte Da 01 a FF: da 1 a 255 byte |

Impostazioni dell'area di configurazione DM per le porte della scheda di comunicazione seriale

| Canale area di configurazione | | Bit | Nome | Impostazioni |
|-------------------------------|---------|------------|---|--|
| Porta 1 | Porta 2 | | | |
| D32004 | D32014 | Da 8 a 15 | Codice di inizio in modalità nessun protocollo | Da 00 a FF esadecimale |
| | | Da 0 a 7 | Codice di fine in modalità nessun protocollo | Da 00 a FF esadecimale |
| D32005 | D32015 | Da 12 a 15 | Identificatore codice di inizio in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di inizio. |
| | | Da 8 a 11 | Identificatore codice di fine in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di fine. 2: utilizzo di CR+LF. |

Flag dell'area ausiliaria per la porta RS-232C della CPU

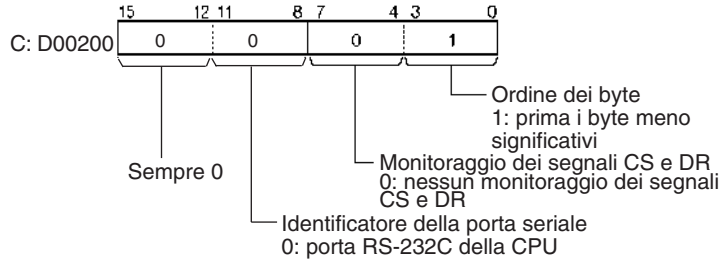
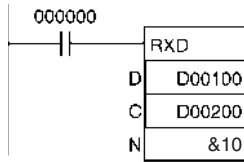
| Nome | Indirizzo | Contenuto |
|---|-----------|--|
| Flag di ricezione avvenuta da parte della porta RS-232C | A39206 | ON quando la ricezione in modalità nessun protocollo è stata completata. Numero di byte ricevuti specificati: il flag viene attivato quando è stato ricevuto il numero di byte specificato. Codice di fine specificato: il flag viene attivato quando sono stati ricevuti il codice di fine o 256 byte. |
| Flag di overflow durante ricezione tramite la porta RS-232C | A39207 | ON quando vengono ricevuti più byte rispetto al numero previsto. Numero di byte ricevuti specificati: il flag viene attivato quando si ricevono ulteriori dati dopo il completamento della ricezione e l'esecuzione della successiva istruzione RXD(235). Codice di fine specificato: il flag viene attivato quando si ricevono ulteriori dati dopo il completamento della ricezione e l'esecuzione della successiva istruzione RXD(235) oppure quando viene ricevuto il 257esimo byte di dati prima che venga ricevuto il codice di fine. |
| Contatore di ricezione della porta RS-232C | A393 | Conta in esadecimale il numero di byte ricevuti in modalità nessun protocollo. |

Flag dell'area ausiliaria per le porte della scheda di comunicazione seriale

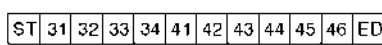
| Porta | Nome | Indirizzo | Contenuto |
|-------------|---|--------------------|---|
| Porta 1 | Flag di ricezione avvenuta | A35606 | ON quando la ricezione in modalità nessun protocollo è stata completata. Numero di byte ricevuti specificati: il flag verrà attivato quando è stato ricevuto il numero di byte specificato. Codice di fine specificato: il flag viene attivato quando sono stati ricevuti il codice di fine o 256 byte. |
| | Flag di overflow durante ricezione | A35607 | ON quando vengono ricevuti più byte rispetto al numero previsto in modalità nessun protocollo. Numero di byte ricevuti specificati: il flag viene attivato quando si ricevono ulteriori dati dopo il completamento della ricezione, ma prima che i dati ricevuti vengano letti dal buffer con RXD(235). Codice di fine specificato: il flag viene attivato quando si ricevono 257 o più byte di dati senza un codice di fine. |
| | Contatore di ricezione | A357 | Conta in esadecimale il numero di byte ricevuti in modalità nessun protocollo (da 0 a 256 decimali). |
| | Flag di errore di overrun | Bit 04 di CIO 1908 | ON quando 260 o più byte di dati vengono ricevuti nel buffer prima di eseguire l'istruzione RXD(235). |
| Porta 2 | Flag di ricezione avvenuta | A35614 | ON quando la ricezione in modalità nessun protocollo è stata completata. Numero di byte ricevuti specificati: il flag viene attivato quando è stato ricevuto il numero di byte specificato. Codice di fine specificato: il flag viene attivato quando sono stati ricevuti il codice di fine o 256 byte. |
| | Flag di overflow durante ricezione | A35615 | ON quando vengono ricevuti più byte rispetto al numero previsto in modalità nessun protocollo. Numero di byte ricevuti specificati: il flag viene attivato quando si ricevono ulteriori dati dopo il completamento della ricezione, ma prima che i dati ricevuti vengano letti dal buffer con RXD(235). Codice di fine specificato: il flag viene attivato quando si ricevono 257 o più byte di dati senza un codice di fine. |
| | Contatore di ricezione | A358 | Conta in esadecimale il numero di byte ricevuti in modalità nessun protocollo (da 0 a 256 decimali). |
| | Flag di errore di overrun | Bit 04 di CIO 1918 | ON quando 260 o più byte di dati vengono ricevuti nel buffer prima di eseguire l'istruzione RXD(235). |
| Porta 1 e 2 | Flag di disabilitazione servizio scheda interna | A42404 | ON quando l'istruzione RXD(235) viene eseguita per una scheda di comunicazione seriale che non supporta la modalità senza protocollo, ovvero una scheda senza un numero di versione. |

Esempi

Nell'esempio seguente, quando CIO 000000 è impostato su ON, i dati vengono ricevuti dalla porta RS-232C e 10 byte di dati vengono memorizzati cominciando in D00100.



Questo esempio presume che nella configurazione del PLC siano stati specificati sia un codice di inizio che un codice di fine.



ST: codice di inizio (ad esempio, 02 esadecimale) ED: codice di fine (ad esempio, 03 esadecimale)

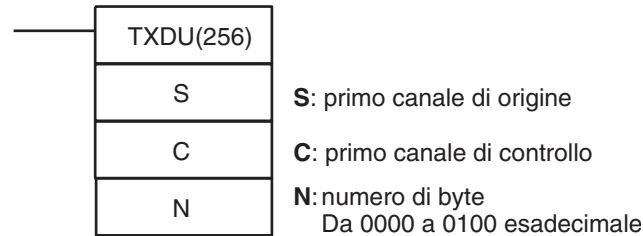
| D | Byte più significativi | | | | Byte meno significativi | | | |
|--------|------------------------|---|---|---|-------------------------|---|---|---|
| | 15 | 8 | 7 | 0 | 15 | 8 | 7 | 0 |
| D00100 | 3 | 2 | 3 | 1 | | | | |
| D00101 | 3 | 4 | 3 | 3 | | | | |
| D00102 | 4 | 2 | 4 | 1 | | | | |
| D00103 | 4 | 4 | 4 | 3 | | | | |
| D00104 | 4 | 6 | 4 | 5 | | | | |

3-24-5 TRANSMIT VIA SERIAL COMMUNICATIONS UNIT: TXDU(256)

Scopo

Invia il numero di byte di dati specificato da una delle porte seriali di un modulo di comunicazione seriale. (Il modulo di comunicazione seriale deve essere della versione 1.2 o successiva).

Simbolo programmazione ladder



Variazioni

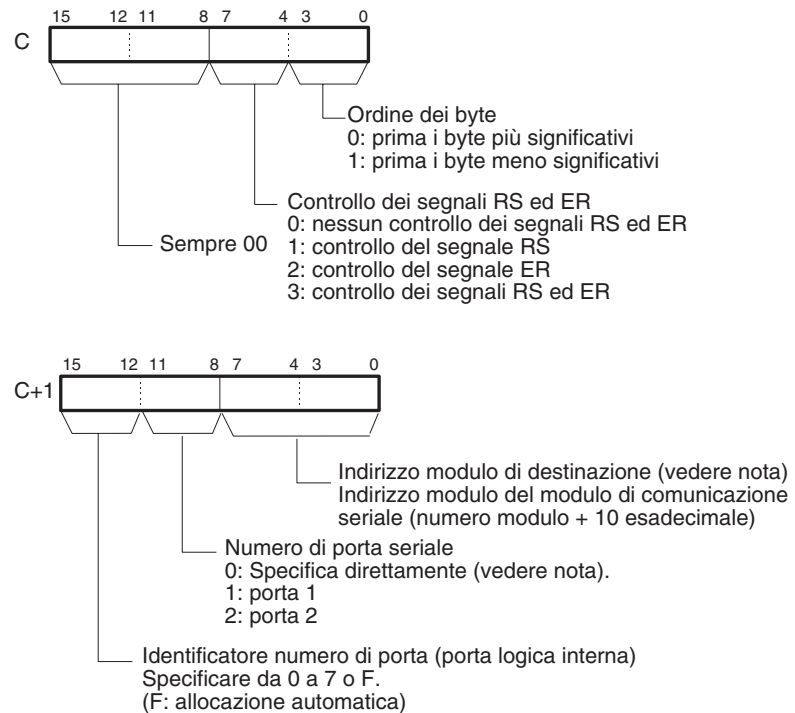
| Variazioni | Eseguita a ogni ciclo con condizione ON | TXDU(256) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TXDU(256) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Il contenuto dei canali di controllo C e C+1 è mostrato di seguito.



Nota È possibile specificare direttamente l'indirizzo del modulo della porta seriale impostando il numero di porta seriale su 0 ed impostando l'indirizzo del modulo di destinazione con l'indirizzo del modulo della porta seriale. (Impostare l'indirizzo del modulo di destinazione su 80 esadecimale + 4 × numero di modulo per la porta 1 o 81 esadecimale + 4 × numero di modulo per la porta 2).

Caratteristiche operando

| Area | S | C | D |
|--------------------------------------|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A958 | Da A000 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |

| Area | S | C | D |
|--|--|-------------------------|--|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | Da #0000 a #0100 (binario) o da &0 a &256 (decimale) |
| Registri dati | --- | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

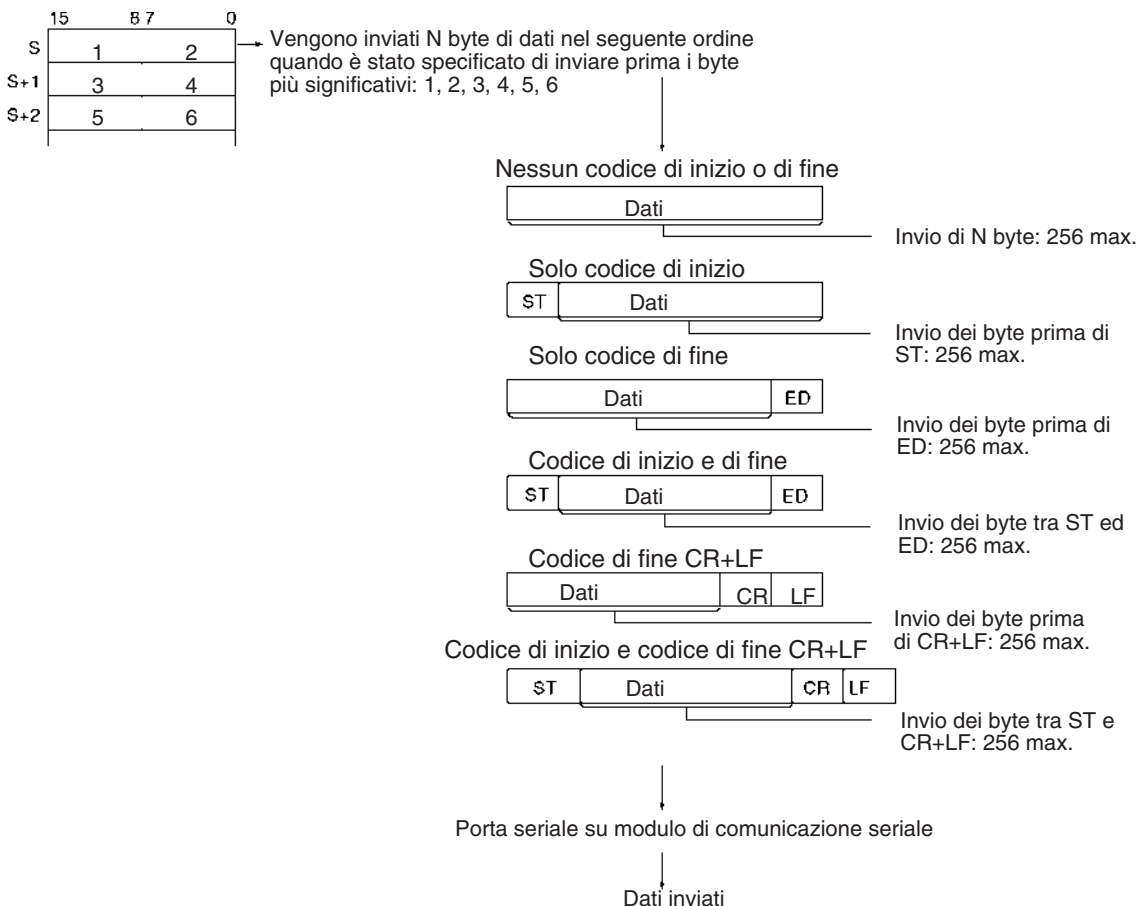
TXDU(256) legge N byte di dati dai canali da S a $S+(N\div 2)-1$ e invia i dati grezzi in modalità nessun protocollo dal modulo di comunicazione seriale avente l'indirizzo di modulo specificato nei bit da 0 a 7 di C+1, tramite la porta specificata con i bit da 8 a 11 di C+1. È possibile impostare il numero di porta logica con un valore qualsiasi compreso tra 0 e 7, specificato con i bit da 12 a 15 di C+1.

Prima che i dati vengano inviati, vengono aggiunti i codici di inizio e di fine specificati per la modalità nessun protocollo nell'area di configurazione DM allocata. È possibile inviare fino a un massimo di 259 byte, compresi i dati di trasmissione (N = 256 byte max.), il codice di inizio e il codice di fine.

È possibile inviare i dati soltanto quando è attivato il flag di abilitazione porte di comunicazione per la porta logica specificata (da A20200 a A20207 per le porte da 0 a 7) e quando è disattivato il flag di esecuzione istruzione TXDU (nell'area di configurazione DM allocata).

Nota Il numero di porta logica può essere assegnato automaticamente impostando i bit da 12 a 15 di C+1 su F. Per informazioni dettagliate, fare riferimento alla sezione *Allocazione automatica delle porte di comunicazione* a pagina 979.

Nello schema seguente è illustrato l'ordine in cui verranno inviati i dati e il contenuto del frame di invio per varie impostazioni del codice di inizio e di fine.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se tutte le porte logiche sono in uso o se il flag di abilitazione porte di comunicazione relativo alla porta logica specificata è disattivato quando viene eseguita l'istruzione. ON se il valore di C non è incluso nell'intervallo. ON se il valore di N non è compreso tra 0000 e 0100 esadecimale. OFF in tutti gli altri casi. |

Avvertenze

È possibile utilizzare l'istruzione TXDU(256) solo per una porta seriale di un modulo di comunicazione seriale impostata in modalità nessun protocollo.

È possibile impostare i seguenti formati di frame per i messaggi di invio nell'area di configurazione DM allocata.

- Codice di inizio: nessuno o da 00 a FF esadecimale.
- Codice di fine: nessuno, CR+LF o da 00 a FF esadecimale.

I dati verranno inviati con qualsiasi combinazione di codici di inizio e/o di fine specificata nell'area di configurazione DM allocata. Se sono stati specificati i codici di inizio e fine, tali codici verranno aggiunti ai dati da inviare (N). In questo caso, il numero massimo di byte che è possibile specificare per N è 256 byte.

I dati possono essere trasmessi solo quando il flag Pronto per Trasmissione della porta è attivato. Il flag Pronto per Trasmissione è A39205 per la porta RS-232C del modulo CPU, A39605 per la porta 1 del modulo di comunicazione seriale, A39613 per la porta 2 del modulo di comunicazione seriale.

I dati vengono inviati nell'ordine specificato in C.

Se per N è stato specificato 0, non verrà eseguito alcun invio.

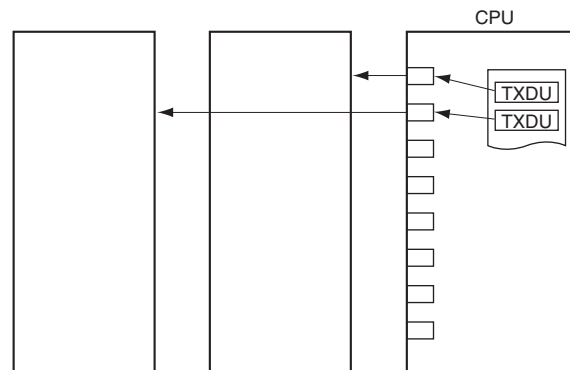
Se in C è stato specificato il controllo del segnale RS, come segnale RS verrà utilizzato il bit 15 di S.

Se in C è stato specificato il controllo del segnale ER, come segnale ER verrà utilizzato il bit 15 di S.

Se in C è stato specificato il controllo dei segnali RS ed ER, come segnale RS verrà utilizzato il bit 15 di S e come segnale ER verrà utilizzato il bit 14 di S.

TXDU(256) utilizza una porta logica (poiché invia un comando interno FINS) per impartire un comando della sequenza di invio al modulo di comunicazione seriale (numero di versione 1.2 o successiva). Poiché anche le istruzioni SEND(090), RECV(098), CMND(490), PMCR(260) e RXDU(255) utilizzano le porte logiche da 0 a 7, non è possibile eseguire l'istruzione TXDU(256) per una porta logica se tale porta è già utilizzata da una di queste istruzioni o da un'altra istruzione TXDU(256).

Per essere sicuri che l'istruzione TXDU(256) non venga eseguita quando la porta logica è occupata, programmare il flag di abilitazione porte di comunicazione (da A20200 a A20207) come condizione normalmente aperta.



Non è possibile eseguire TXDU(256) quando è attivato il flag di esecuzione istruzione TXDU (bit 5 di n+9 o n+19, dove n = CIO 1500 + 25 × numero di modulo). Per essere certi che un'altra istruzione TXDU(256) non venga eseguita per la porta finché la prima istruzione TXDU(256) non è stata completata, programmare il flag di esecuzione istruzione TXDU relativo alla porta come condizione normalmente chiusa.

Si verificherà un errore e il flag di errore verrà attivato nei seguenti casi:

- Il flag di abilitazione porte di comunicazione è OFF per la porta logica specificata quando è stata eseguita l'istruzione TXDU(256).
- Il valore di C non è incluso nell'intervallo.
- Il valore di N non è compreso tra 0000 e 0100 esadecimale.

Nota In base al dispositivo esterno, può essere necessario impostare un ritardo nell'invio quando si inviano i dati con TXDU(256). Se è richiesto un ritardo di invio, impostare o regolare il ritardo nell'area di configurazione DM allocata.

Canali e flag correlati

Quando si esegue l'istruzione TXD(236), è possibile utilizzare le seguenti impostazioni per la configurazione del PLC e per il flag dell'area ausiliaria.

Impostazioni dell'area di configurazione DM

(m = D30000 + 100 × numero modulo)

| Canale area di configurazione | | Bit | Nome | Impostazioni |
|-------------------------------|---------|------------|---|---|
| Porta 1 | Porta 2 | | | |
| m+2 | m+12 | 15 | Identificatore ritardo invio in modalità nessun protocollo | 0: impostazione predefinita (0 ms) 1: utilizzare il ritardo nei bit da 1 a 14. |
| | | Da 0 a 14 | Tempo di ritardo invio in modalità nessun protocollo | Da 0000 a 7530 esadecimale Da 0 a 300.000 ms decimali (in unità di 10 ms) |
| m+4 | m+14 | Da 8 a 15 | Codice di inizio in modalità nessun protocollo | Da 00 a FF esadecimale |
| | | Da 0 a 7 | Codice di fine in modalità nessun protocollo | Da 00 a FF esadecimale |
| m+5 | m+15 | Da 12 a 15 | Identificatore codice di inizio in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di inizio. |
| | | Da 8 a 11 | Identificatore codice di fine in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di fine. 2: utilizzo di CR+LF. |

Area ausiliaria

| Nome | Indirizzo | Descrizione |
|---|---------------------|--|
| Flag di abilitazione porte di comunicazione | Da A20200 ad A20207 | ON quando un'istruzione di comunicazione, inclusa TXDU(256), può essere eseguita con il numero di porta corrispondente. I bit da 00 a 07 corrispondono alle porte di comunicazione da 0 a 7. Il flag è OFF quando è in corso l'esecuzione di un'istruzione di comunicazione ed è ON quando l'esecuzione è stata completata (terminata normalmente o con errori). |
| Codici di completamento per le porte di comunicazione | Da A203 ad A210 | Questi canali contengono i codici di completamento per i numeri delle porte corrispondenti al termine dell'esecuzione di un'istruzione di comunicazione. I canali da A203 ad A210 corrispondono alle porte di comunicazione da 0 a 7. Il codice è 00 durante l'esecuzione dell'istruzione e contiene il relativo codice quando l'esecuzione è stata completata. Questi canali vengono reimpostati su 0000 quando comincia il funzionamento del PLC. |
| Flag di errore porte di comunicazione | A219 | ON quando si verifica un errore durante l'esecuzione di un'istruzione di comunicazione. Quando un flag è ON, controllare il codice di completamento in A203 - A210 per risolvere l'errore. OFF quando l'esecuzione è terminata normalmente. I bit da 00 a 07 corrispondono alle porte di comunicazione da 0 a 7. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di comunicazione successiva. Perfino se si verifica un errore, un flag verrà azzerato la volta successiva che viene eseguita un'istruzione di comunicazione per tale porta. |

Codici di completamento

| Codice | Significato |
|---------------------|---|
| 0205 esadecimale | Timeout risposta. Questo errore può verificarsi quando è impostata la modalità di comunicazione Host Link. |
| 0401 esadecimale | Comando non definito. Questo errore può verificarsi quando è impostata la modalità di comunicazione Protocol macro, NT Link, test echoback o gateway seriale. |
| 1001 esadecimale | Il comando è troppo lungo. |
| 1002 esadecimale | Il comando è troppo corto. |
| 1003 esadecimale | Il numero specificato di elementi di dati non corrisponde alla quantità effettiva di dati inviati. |
| 1004 esadecimale | Il formato del comando non è corretto. |
| 110C esadecimale | Errore altro parametro |
| 2201 esadecimale | Impossibile eseguire l'operazione durante il funzionamento (operazione disabilitata poiché il modulo è occupato nell'invio). |
| 2202 esadecimale | Impossibile eseguire l'operazione al momento dell'interruzione (operazione disabilitata poiché il modulo sta commutando i protocolli). |

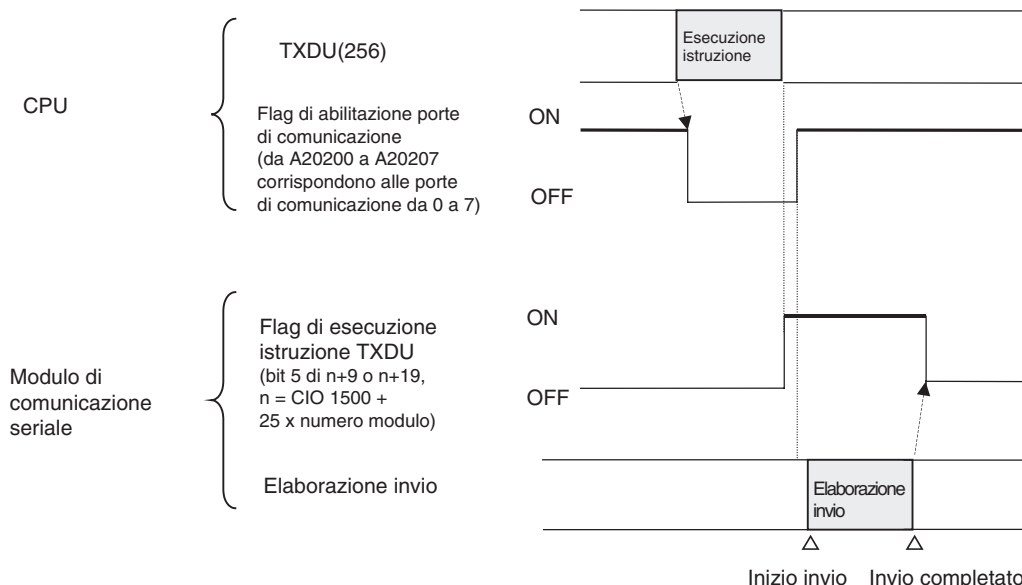
Flag correlati nell'area dell'unità bus CPU

(n = CIO 1500 + 25 × numero modulo)

| Canale | | Bit | Nome | Stato |
|---------|---------|-----|------------------------------------|--|
| Porta 1 | Porta 2 | | | |
| n+9 | n+19 | 05 | Flag di esecuzione istruzione TXDU | 0: TXDU(256) non viene eseguita. 1: TXDU(256) è in corso di esecuzione. |

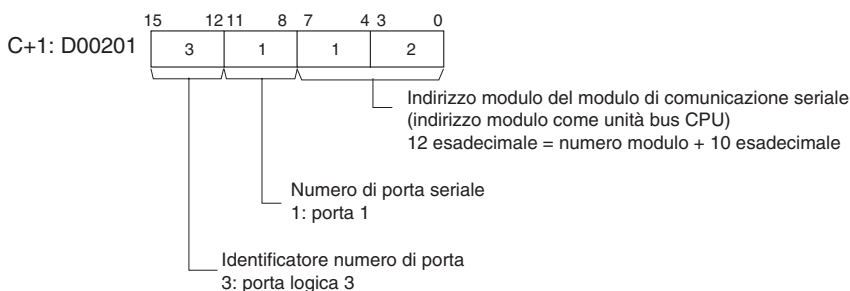
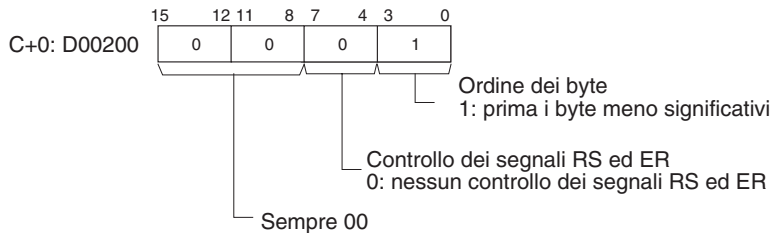
Esempio: funzionamento del flag

Nello schema seguente viene mostrato il funzionamento del flag di abilitazione porte di comunicazione e del flag di esecuzione istruzione TXDU.



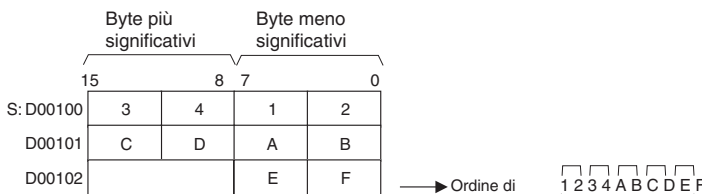
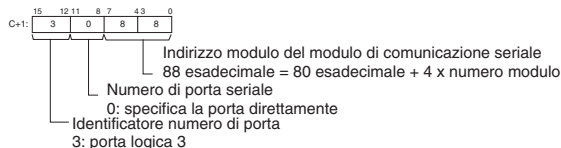
Esempio: invio di dati

Nell'esempio seguente, quando CIO 000000 è impostato su ON, A20203 (il flag di abilitazione porte di comunicazione) è attivato e CIO 155905 (il flag di esecuzione istruzione TXDU per la porta 1) è disattivato, TXDU(256) invia i dati tramite la porta seriale 1 del modulo di comunicazione seriale con numero di modulo 2. I 5 byte dei dati di invio vengono letti dall'area DM cominciando dal byte più a destra di D00100 e inviati tramite la porta logica 3 a un dispositivo generico, ad esempio una stampante.



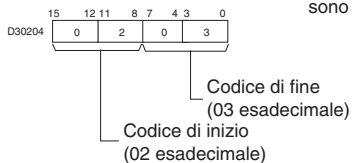
Nota:

È possibile specificare direttamente l'indirizzo del modulo della porta seriale impostando il numero di porta seriale su 0 ed impostando l'indirizzo modulo del modulo di comunicazione seriale con l'indirizzo del modulo della porta seriale. (Impostare l'indirizzo del modulo su 80 esadecimale +4 x numero di modulo per la porta 1 o 81 esadecimale +4 x numero di modulo per la porta 2).

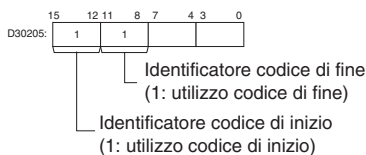


Esempio di impostazioni dell'area di configurazione DM allocata:

Valori codice di inizio e codice di fine



Identificatori codice di inizio e codice di fine



In questo esempio, nell'area di configurazione DM allocata sono stati specificati un codice di inizio e un codice di fine.



ST: codice di inizio (ad esempio, 02 esadecimale)
ED: codice di fine (ad esempio, 03 esadecimale)

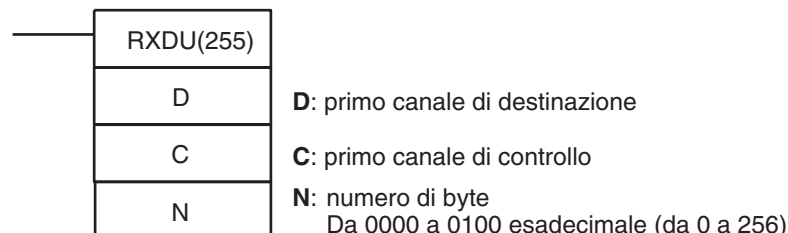
Dati inviati

3-24-6 RECEIVE VIA SERIAL COMMUNICATIONS UNIT: RXDU(255)

Scopo

Legge il numero di byte di dati specificato da una delle porte seriali di un modulo di comunicazione seriale. (Il modulo di comunicazione seriale deve essere della versione 1.2 o successiva).

Simbolo programmazione ladder



Variazioni

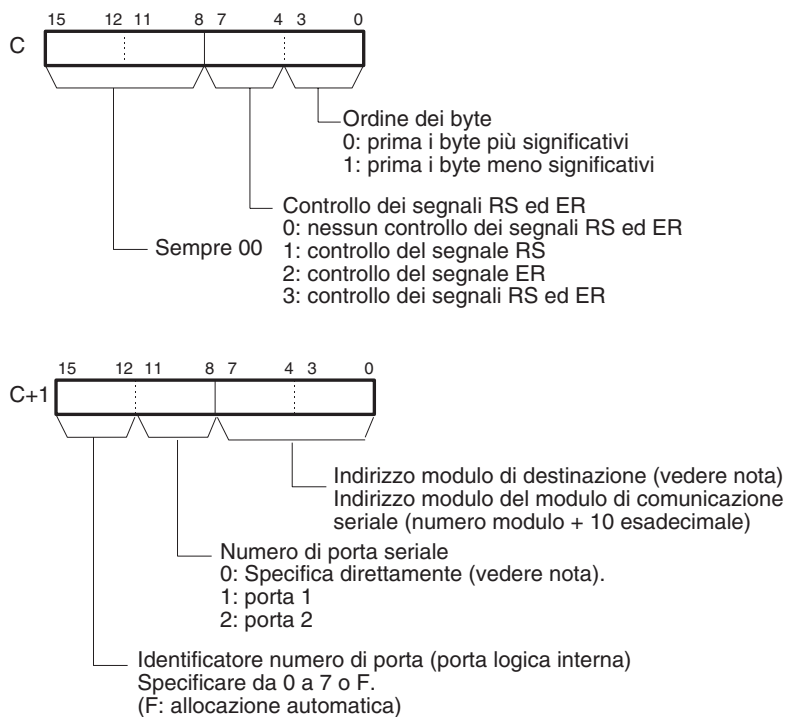
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RXDU(255) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RXDU(255) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Il contenuto dei canali di controllo C e C+1 è mostrato di seguito.



Nota È possibile specificare direttamente l'indirizzo del modulo della porta seriale impostando il numero di porta seriale su 0 ed impostando l'indirizzo del modulo di destinazione con l'indirizzo del modulo della porta seriale. (Impostare l'indirizzo del modulo di destinazione su 80 esadecimale + 4 × numero di modulo per la porta 1 o 81 esadecimale + 4 × numero di modulo per la porta 2).

Caratteristiche operando

| Area | D | C | D |
|--|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 | Da W000 a W510 | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H510 | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A958 | Da A000 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4094 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4094 | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32766 | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32766 | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | Da #0000 a #0100 (binario) o da &0 a &256 (decimale) |
| Registri dati | --- | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

RXDU(255) legge i dati che ha ricevuto in modalità senza protocollo sul modulo di comunicazione seriale con l'indirizzo di modulo specificato nei bit da 0 a 7 di C+1, tramite la porta specificata con i bit da 8 a 11 di C+1, e memorizza a tali dati cominciando da D. Se sulla porta sono stati ricevuti meno byte di dati rispetto a N, verranno memorizzati soltanto i dati ricevuti. È possibile impostare il numero di porta logica con un valore qualsiasi compreso tra 0 e 7, specificato con i bit da 12 a 15 di C+1.

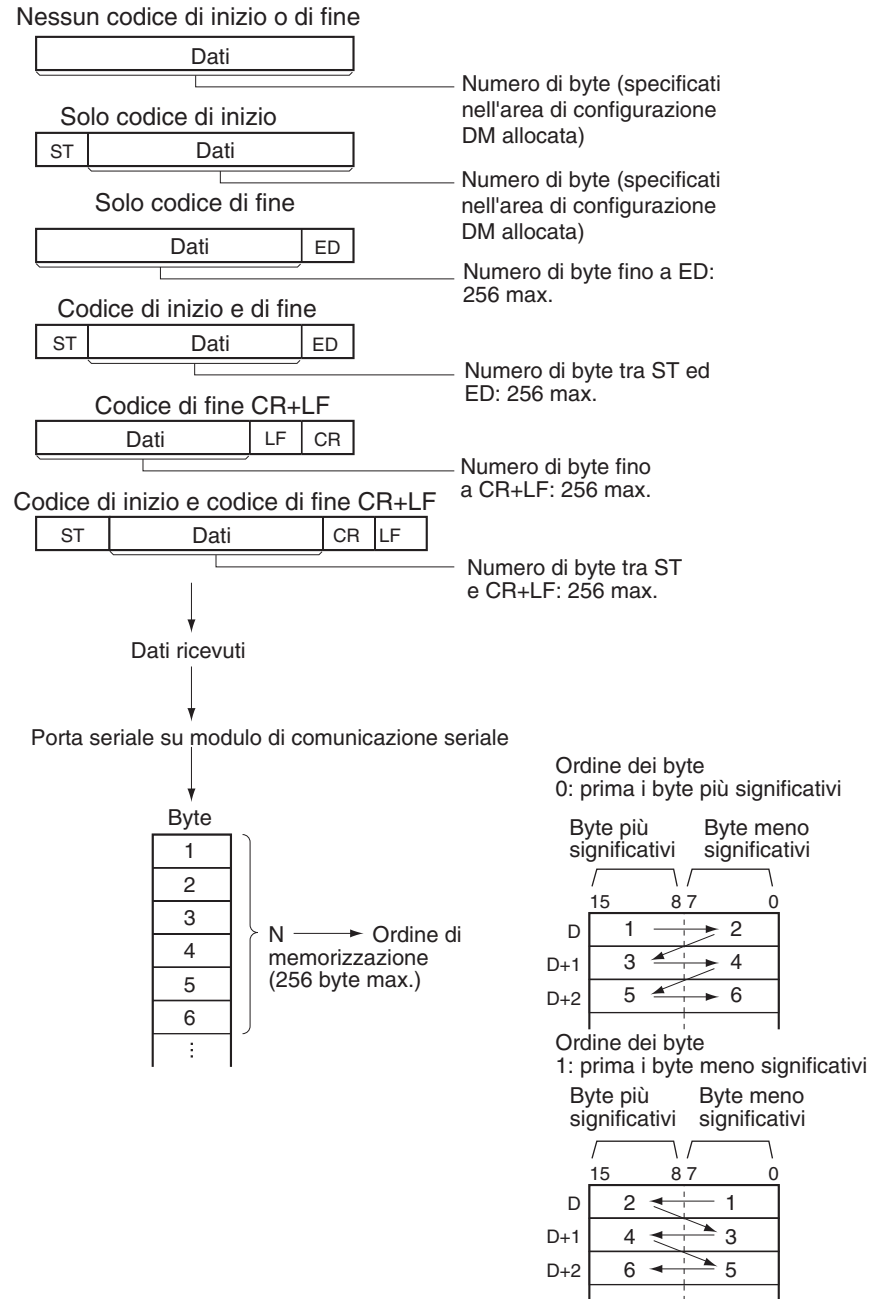
Eseguire l'istruzione RXDU(255) per leggere i dati ricevuti dal buffer quando il flag di ricezione avvenuta (nell'area di configurazione DM) è ON.

È possibile ricevere fino a un massimo di 259 byte, compresi i dati di ricezione (N = 256 byte max.), il codice di inizio e il codice di fine.

Nello schema seguente è illustrato l'ordine in cui verranno ricevuti i dati e il contenuto del frame di ricezione per varie impostazioni del codice di inizio e di fine.

Nota Il numero di porta logica può essere assegnato automaticamente impostando i bit da 12 a 15 di C+1 su F. Per informazioni dettagliate, fare riferimento alla sezione *Allocazione automatica delle porte di comunicazione* a pagina 979.

Nello schema seguente è illustrato l'ordine in cui verranno inviati i dati e il contenuto del frame di invio per varie impostazioni del codice di inizio e di fine.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se tutte le porte logiche sono in uso o se il flag di abilitazione porte di comunicazione relativo alla porta logica specificata è disattivato quando viene eseguita l'istruzione. ON se il valore di C non è incluso nell'intervallo. ON se il valore di N non è compreso tra 0000 e 0100 esadecimale. OFF in tutti gli altri casi. |

Avvertenze

È possibile utilizzare l'istruzione RXDU(255) solo per una porta seriale di un modulo di comunicazione seriale impostata in modalità nessun protocollo.

È possibile impostare i seguenti formati di frame per i messaggi di ricezione nell'area di configurazione DM allocata.

- Codice di inizio: nessuno o da 00 a FF esadecimale.
- Codice di fine: nessuno, CR+LF o da 00 a FF esadecimale. Se non è stato impostato nessun codice di fine, il numero di byte da ricevere verrà impostato da 00 a FF esadecimale (da 1 a 256 decimale; 00 specifica 256 byte).

Il flag di ricezione avvenuta (nota 1) verrà attivato quando è stato ricevuto il numero di byte specificati nell'area di configurazione DM allocata. Quando viene attivato il flag di ricezione avvenuta, il numero di byte nel contatore di ricezione (nota 2) avrà lo stesso valore del numero di byte ricevuti specificato nell'area di configurazione DM allocata. Se vengono ricevuti più byte di quelli specificati, verrà attivato il flag di overflow durante ricezione (nota 3).

Se nell'area di configurazione DM allocata viene specificato un codice di fine, verrà attivato il flag di ricezione avvenuta (nota 1) quando viene ricevuto il codice di fine o 256 byte di dati. Se dopo l'attivazione del flag di ricezione avvenuta (nota 1) vengono ricevuti più dati del previsto, verrà attivato il flag di overflow durante ricezione (nota 3).

La ricezione verrà interrotta se vengono ricevuti 259 byte di dati. Se in seguito vengono immessi più dati, verranno attivati i flag di errore di overrun (nota 4) e di errore di trasmissione (nota 5).

Quando nella porta seriale della scheda di comunicazione seriale vengono immessi più dati di quelli specificati in N, questi dati verranno scartati quando viene eseguita l'istruzione RXDU(255) successiva.

Quando l'istruzione RXDU(255) viene eseguita, i dati vengono memorizzati nella memoria cominciando in D, il flag di ricezione avvenuta (nota 1) verrà disattivato (anche se il flag di overflow durante ricezione (nota 3) è ON) e il contatore di ricezione (nota 2) verrà reimpostato su 0.

I dati verranno archiviati in memoria nell'ordine specificato in C.

Se per N è specificato 0, i flag di ricezione avvenuta (nota 1) e di overflow durante ricezione (nota 3) verranno disattivati, il contatore di ricezione (nota 2) verrà reimpostato su 0 e in memoria non verrà archiviato nulla.

Se in C è stato specificato il monitoraggio del segnale CS, lo stato del segnale CS verrà memorizzato nel bit 15 di D.

Se in C è stato specificato il monitoraggio del segnale DR, lo stato del segnale DR verrà memorizzato nel bit 15 di D.

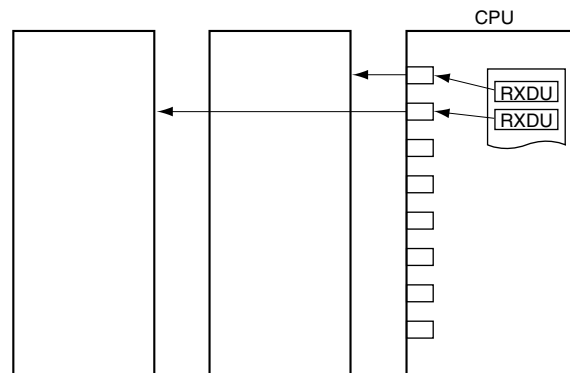
Se in C è stato specificato il monitoraggio dei segnali CS e DR, lo stato del segnale CS verrà memorizzato nel bit 15 di D e lo stato del segnale DR verrà memorizzato nel bit 14 di D.

Se non è stato specificato il monitoraggio del segnale CS o DR, i dati ricevuti non verranno memorizzati.

Se per il controllo dei segnali RS e DR in C è stato specificato 1, 2 o 3 esadecimale, l'istruzione RXDU(255) verrà eseguita senza tener conto dello stato del flag di ricezione avvenuta (nota 1).

RXDU(255) utilizza una porta logica (poiché invia un comando interno FINS) per impartire un comando della sequenza di ricezione a un modulo di comunicazione seriale o a una scheda di comunicazione seriale della serie CS. Poiché anche le istruzioni SEND(090), RECV(098), CMND(490), PMCR(260) e TXDU(255) utilizzano le porte logiche da 0 a 7, non è possibile eseguire l'istruzione RXDU(256) per una porta logica se tale porta è già utilizzata da una di queste istruzioni o da un'altra istruzione RXDU(255).

Per essere sicuri che l'istruzione RXDU(255) non venga eseguita quando la porta logica è occupata, programmare il flag di abilitazione porte di comunicazione (da A20200 a A20207) come condizione normalmente aperta.



Non è possibile eseguire RXDU(255) quando è attivato il flag ricezione avvenuta (bit 6 di n+9 o n+19, dove n = CIO 1500 + 25 × numero di modulo). Programmare il flag di ricezione avvenuta come condizione normalmente aperta di RXDU(255).

Si verificherà un errore e il flag di errore verrà attivato nei seguenti casi:

- Il flag di abilitazione porte di comunicazione è OFF per la porta logica specificata quando viene eseguita l'istruzione RXDU(255).
- Il valore di C non è incluso nell'intervallo.
- Il valore di N non è compreso tra 0000 e 0100 esadecimale.

- Nota**
1. Flag di ricezione avvenuta (n = CIO 1500 + 25 × numero modulo)
 - Porta 1: bit 6 di n+9
 - Porta 2: bit 6 di n+19
 2. Contatori di ricezione (n = CIO 1500 + 25 × numero modulo)
 - Porta 1: n+10
 - Porta 2: n+20
 3. Flag di overflow durante ricezione (n = CIO 1500 + 25 × numero modulo)
 - Porta 1: bit 7 di n+9
 - Porta 2: bit 7 di n+19
 4. Flag di errore di overrun (n = CIO 1500 + 25 × numero modulo)
 - Porta 1: bit 4 di n+8
 - Porta 2: bit 4 di n+18
 5. Flag di errore di trasmissione (n = CIO 1500 + 25 × numero modulo)
 - Porta 1: bit 15 di n+8
 - Porta 2: bit 15 di n+18
 6. Non è possibile ricevere ulteriori dati finché i dati ricevuti non vengono letti dal buffer tramite RXDU(255). Quando si attiva il flag di ricezione avvenuta, i dati vengono immediatamente letti con RXDU(255), prima che vengano inviati ulteriori dati alla porta.
 7. Quando RXDU(255) viene utilizzata per leggere i dati ricevuti su una delle porte del modulo di comunicazione seriale, il buffer di ricezione della porta viene azzerato dopo l'esecuzione dell'istruzione RXDU(255). Pertanto, **non** è possibile eseguire più volte l'istruzione RXDU(255) per leggere un blocco di dati suddiviso in parti.

Canali e flag correlati

Al funzionamento dell'istruzione RXDU(255) sono correlati i seguenti canali.

Impostazioni dell'area di configurazione DM

(m = D30000 + 100 × numero modulo)

| Canale area di configurazione | | Bit | Nome | Impostazioni |
|-------------------------------|---------|------------|---|--|
| Porta 1 | Porta 2 | | | |
| m+4 | m+14 | Da 8 a 15 | Codice di inizio in modalità nessun protocollo | Da 00 a FF esadecimale |
| | | Da 0 a 7 | Codice di fine in modalità nessun protocollo | Da 00 a FF esadecimale |
| m+5 | m+15 | Da 12 a 15 | Identificatore codice di inizio in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di inizio. |
| | | Da 8 a 11 | Identificatore codice di fine in modalità nessun protocollo | 0: nessuna 1: utilizzo codice di fine. 2: utilizzo di CR+LF. |

Area ausiliaria

| Nome | Indirizzo | Descrizione |
|---|---------------------|--|
| Flag di abilitazione porte di comunicazione | Da A20200 ad A20207 | ON quando un'istruzione di comunicazione, inclusa RXDU(255), può essere eseguita con il numero di porta corrispondente. I bit da 00 a 07 corrispondono alle porte di comunicazione da 0 a 7. Il flag è OFF quando è in corso l'esecuzione un'istruzione di comunicazione ed è ON quando l'esecuzione è stata completata (terminata normalmente o con errori). |
| Codici di completamento per le porte di comunicazione | Da A203 ad A210 | Questi canali contengono i codici di completamento per i numeri delle porte corrispondenti al termine dell'esecuzione di un'istruzione di comunicazione. I canali da A203 ad A210 corrispondono alle porte di comunicazione da 0 a 7. Il codice è 00 durante l'esecuzione dell'istruzione e contiene il relativo codice quando l'esecuzione è stata completata. Questi canali vengono reimpostati su 0000 quando comincia il funzionamento del PLC. |
| Flag di errore porte di comunicazione | A219 | ON quando si verifica un errore durante l'esecuzione di un'istruzione di comunicazione. Quando un flag è ON, controllare il codice di completamento in A203 - A210 per risolvere l'errore. OFF quando l'esecuzione è terminata normalmente. I bit da 00 a 07 corrispondono alle porte di comunicazione da 0 a 7. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di comunicazione successiva. Perfino se si verifica un errore, un flag verrà azzerato la volta successiva che viene eseguita un'istruzione di comunicazione per tale porta. |

Codici di completamento

| Codice | Significato |
|------------------|---|
| 0205 esadecimale | Timeout risposta. Questo errore può verificarsi quando è impostata la modalità di comunicazione Host Link. |
| 0401 esadecimale | Comando non definito. Questo errore può verificarsi quando è impostata la modalità di comunicazione Protocol macro, NT Link, test echoback o gateway seriale. |
| 1001 esadecimale | Il comando è troppo lungo. |

| Codice | Significato |
|---------------------|--|
| 1002 esadecimale | Il comando è troppo corto. |
| 1004 esadecimale | Il formato del comando non è corretto. |
| 110C esadecimale | Errore altro parametro |
| 2201 esadecimale | Impossibile eseguire l'operazione durante il funzionamento (operazione disabilitata poiché il modulo è occupato nell'invio). |
| 2202 esadecimale | Impossibile eseguire l'operazione al momento dell'interruzione (operazione disabilitata poiché il modulo sta commutando i protocolli). |

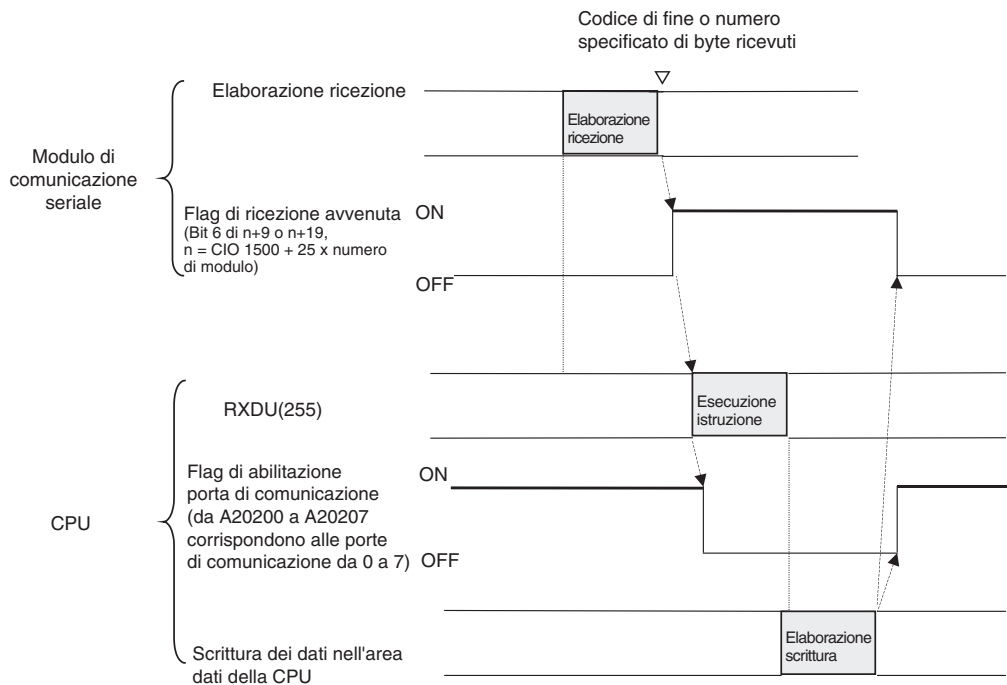
Flag correlati nell'area dell'unità bus CPU

(n = CIO 1500 + 25 × numero modulo)

| Canale | | Bit | Funzione |
|---------|---------|-----|--|
| Porta 1 | Porta 2 | | |
| n+8 | n+18 | 04 | <p>Flag di errore di overrun</p> <p>1: il buffer di ricezione conteneva più di 259 byte prima dell'esecuzione dell'istruzione RXDU(255).</p> <p>Nota: quando questo flag di errore viene attivato, per disattivarlo è necessario spegnere e quindi ripristinare l'alimentazione o riavviare la scheda.</p> |
| n+9 | n+19 | 06 | <p>Flag di ricezione avvenuta</p> <p>0: nessun dato ricevuto o ricezione dei dati in corso 1: ricezione completata</p> <p>0 → 1: la scheda o il modulo ha ricevuto il numero specificato di byte.</p> <p>1 → 0: RXD(235) o RXDU(255) è stata eseguita per scrivere i dati dal buffer nell'area dati di una CPU.</p> |
| n+9 | n+19 | 07 | <p>Flag di overflow durante ricezione</p> <p>0: la scheda o il modulo non ha ricevuto più byte rispetto al numero di byte specificato.</p> <p>1: la scheda o il modulo ha ricevuto più byte rispetto al numero di byte specificato.</p> <p>0 → 1: la scheda o il modulo ha ricevuto ulteriori dati a ricezione già completata.</p> <p>1 → 0: RXD(235) o RXDU(255) è stata eseguita per scrivere i dati dal buffer nell'area dati di una CPU.</p> |
| n+10 | n+20 | 05 | <p>Contatore di ricezione</p> <p>Indica il numero di byte ricevuti in formato esadecimale, tra 0000 e 0100 esadecimale (da 0 a 256 decimali).</p> |

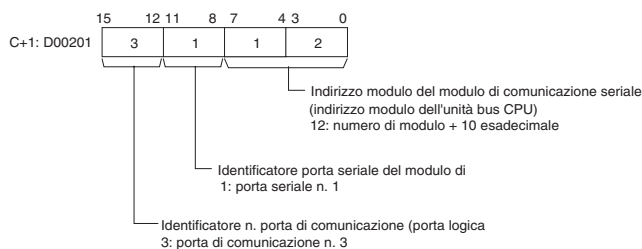
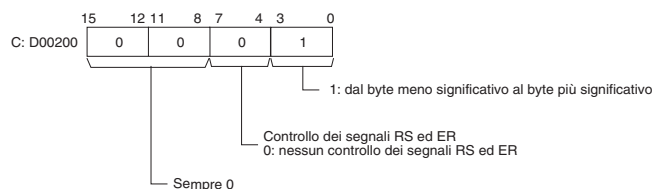
Esempio: funzionamento del flag

Nello schema seguente è illustrato il funzionamento di RXDU(255) e dei flag correlati.

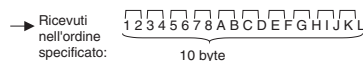
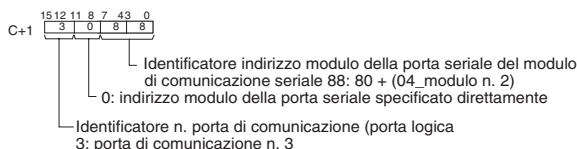


Esempio: ricezione di dati

Nell'esempio seguente, quando CIO 000000 è impostato su ON, A20203 (il flag di abilitazione porte di comunicazione) è attivato e CIO 155906 (il flag di ricezione avvenuta per la porta 1) è disattivato, RXDU(255) legge i dati ricevuti tramite la porta seriale 1 del modulo di comunicazione seriale con numero di modulo 2. (Il numero di porta logica 3 è utilizzato per ricevere i dati da un dispositivo generico, ad esempio un lettore di codici a barre). I 10 byte di dati ricevuti vengono scritti nell'area DM cominciando dal byte più a destra di D00100.



Nota: è possibile inoltre specificare direttamente in C+1 l'indirizzo modulo della porta seriale del modulo di comunicazione seriale.



Codici di inizio e di fine aggiunti in base all'impostazione della configurazione del PLC

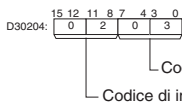
| | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|
| ST | 12 | 34 | 56 | 78 | AB | CD | EF | GH | IJ | KL | ED |
|----|----|----|----|----|----|----|----|----|----|----|----|

ST: codice di inizio (ad esempio, 02 esadecimale)
ED: codice di fine (ad esempio, 03 esadecimale)

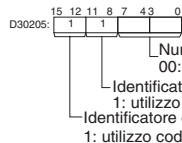
↓
Dati ricevuti

Nota: impostazioni dell'area DM allocata

- Codice di inizio/Codice di fine



- Identificatore codice di inizio/codice di fine

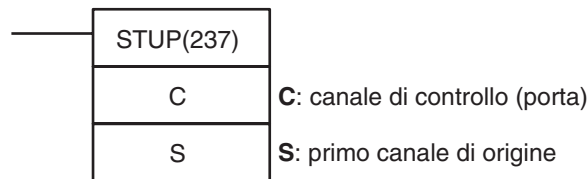


3-24-7 CHANGE SERIAL PORT SETUP: STUP(237)

Scopo

Modifica i parametri di comunicazione di una porta seriale sulla CPU, sulla scheda di comunicazione seriale (solo serie CS) o sul modulo di comunicazione seriale (unità bus CPU). L'istruzione STUP(237) consente pertanto di modificare la modalità di protocollo durante il funzionamento del PLC.

Simbolo programmazione ladder



Variazioni

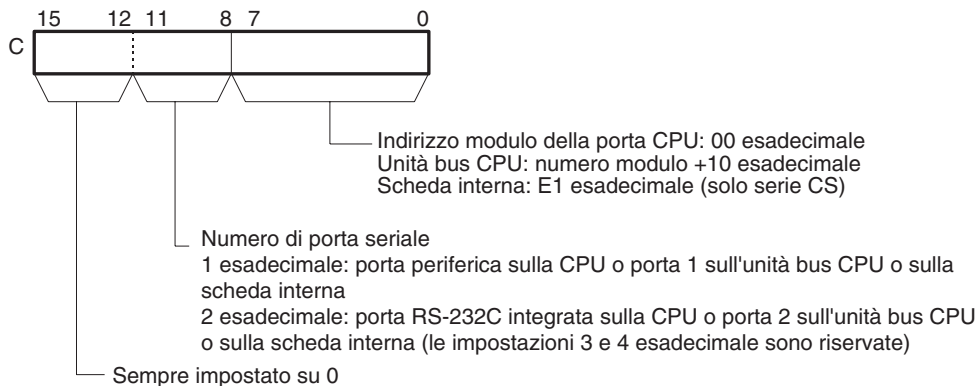
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | STUP(237) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @STUP(237) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Operandi

Il contenuto del canale di controllo C è mostrato di seguito.



Caratteristiche operando

| Area | C | S |
|--------------------------------------|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6134 |
| Area di lavoro | Da W000 a W511 | Da W000 a W502 |
| Area bit di ritenività | Da H000 a H511 | Da H000 a H502 |
| Area bit ausiliaria | Da A000 a A438 Da A448 a A959 | Da A000 a A438 Da A448 a A950 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4086 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4086 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32758 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32758 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32758 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @ D00000 a @ D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |

| Area | C | S |
|--|--|-------|
| Costanti | Solo valori specificati | #0000 |
| Registri dati | Da DR0 a DR15 | --- |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 | |

Descrizione

STUP(237) scrive 10 canali di dati da S a S+9 nell'area di configurazione comunicazione del modulo, con l'indirizzo modulo specificato, come mostrato nella tabella riportata di seguito. Quando la costante #0000 è specificata in S, le impostazioni di comunicazione della porta corrispondente verranno impostate come predefinite.

| Indirizzo modulo | Modulo | N. porta | Porta seriale | Area di configurazione comunicazione della porta seriale |
|-------------------------------|--|---------------|---------------|---|
| 00 esadecimale | CPU | 1 esadecimale | Porta 1 | Parametri di comunicazione per la porta della periferica nella configurazione del PLC |
| | | 2 esadecimale | Porta 2 | Parametri di comunicazione per la porta RS-232C nella configurazione del PLC |
| N. modulo + 10 esadecimale | Modulo di comunicazione seriale (unità bus CPU) | 1 esadecimale | Porta 1 | 10 canali cominciando da D30000 + 100 x n. modulo |
| | | 2 esadecimale | Porta 2 | 10 canali cominciando da D30000 + 100 x n. modulo + 10 |
| E1 esadecimale | Scheda di comunicazione seriale (scheda interna) (solo serie CS) | 1 esadecimale | Porta 1 | 10 canali cominciando da D32000 |
| | | 2 esadecimale | Porta 2 | 10 canali cominciando da D32010 |

Quando viene eseguita l'istruzione STUP(237), viene attivato il relativo flag di modifica parametri porta (A61901, A61902 o da A619 a A636). Il flag resterà ON finché verrà completata la modifica dei parametri.

Utilizzare l'istruzione STUP(237) per modificare i parametri di comunicazione di una porta durante il funzionamento, in base a condizioni specificate. È possibile, ad esempio, utilizzare STUP(237) per modificare le comunicazioni Host Link per operazioni di monitoraggio e programmazione da un computer host, quando vengono soddisfatte determinate condizioni durante l'esecuzione di una sequenza di comunicazione per una connessione via modem.

Differenze tra CPU

Se dopo che è stata utilizzata l'istruzione STUP(237) per modificare i parametri di comunicazione il PLC viene spento e nuovamente acceso, i nuovi parametri verranno mantenuti o verranno ripristinati i parametri precedenti, a seconda della CPU.

| CPU | Stato dei parametri di comunicazione |
|---------------------------|---|
| CS1-H, CJ1-H, CJ1M o CS1D | Se il PLC viene spento e riacceso nuovamente, i parametri di comunicazione vengono ripristinati con le impostazioni esistenti prima delle modifiche effettuate con STUP(237). |
| CS1 | Se il PLC viene spento e riacceso nuovamente, vengono mantenuti i parametri di comunicazione impostati tramite STUP(237). |

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se i valori di C non sono inclusi nell'intervallo. ON se l'istruzione STUP(237) viene eseguita per una porta il cui flag di modifica parametri di comunicazione è già ON. ON se STUP(237) viene eseguita in un task ad interrupt. OFF in tutti gli altri casi. |

Avvertenze

I parametri di comunicazione comprendono la modalità di protocollo, la velocità di trasmissione, il formato dei dati (metodo di trasmissione Protocol macro e lunghezza massima della comunicazione dei dati Protocol macro) e altri parametri. Per informazioni dettagliate sulla porta seriale da impostare, fare riferimento al manuale dell'operatore *Controllori programmabili della serie CS/CJ (W339)* o al *CS/CJ-series Serial Communications Boards and Serial Communications Unit Operation Manual (W336)*.

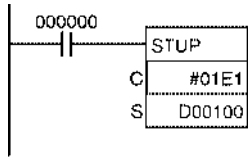
Canali e flag correlati

Nell'esecuzione dell'istruzione STUP(237) è possibile utilizzare i seguenti flag, in base alle esigenze, contenuti nell'area ausiliaria.

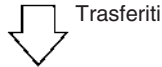
| Nome | Indirizzo | Contenuto |
|---|--|--|
| Flag di modifica parametri porta della periferica | A61901 | ON quando i parametri di comunicazione sono stati modificati per la porta della periferica. |
| Flag di modifica parametri porta RS-232C | A61902 | ON quando i parametri di comunicazione sono stati modificati per la porta RS-232C. |
| Flag di modifica dei parametri della porta per le porte da 1 a 4 sui moduli di comunicazione seriale da 1 a 15. | Da A620 bit 01 - bit 04 ad A635 bit 01 - bit 04 | ON quando i parametri di comunicazione vengono modificati per una porta del modulo di comunicazione seriale. |
| Flag di modifica dei parametri della porta per le porte da 1 a 4 sulla scheda di comunicazione seriale (solo serie CS). | Da A63601 a A63604 | ON quando i parametri di comunicazione vengono modificati per una porta sulla scheda di comunicazione seriale. |

Esempi

Nell'esempio seguente, quando CIO 000000 viene impostato su ON, i parametri di comunicazione per la porta seriale 1 della scheda di comunicazione seriale (scheda interna) vengono modificati con le impostazioni contenute nei 10 canali da D00100 a D00109. In questo esempio viene modificata l'impostazione da modalità protocollo a modalità Protocol macro.



| | | |
|-------------|---------|---|
| S: D00100 | 0 6 0 0 | Impostazione porta: predefinita, modalità protocollo: 6 esadecimale (Protocol macro) |
| S+1: D00101 | 0 0 0 0 | |
| S+2: D00102 | | Velocità di trasmissione: predefinita (9.600 bps) |
| ... | ... | |
| S+9: D00109 | | |



Canali DM allocati alla configurazione di comunicazione della scheda di comunicazione seriale

| | |
|--------|---------|
| D32000 | 0 6 0 0 |
| D32001 | 0 0 0 0 |
| D32002 | |
| ... | ... |
| D32009 | |

3-25 Istruzioni di rete

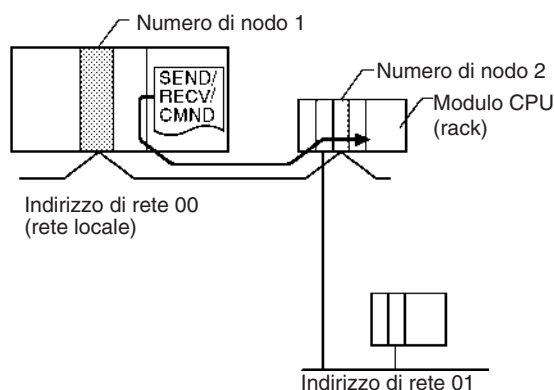
3-25-1 Informazioni sulle operazioni con SYSMAC NET Link/SYSMAC LINK

Le istruzioni di rete possono essere divise in due tipi: SEND(090)/RECV(098) e CMND(490). Queste istruzioni vengono trasmesse tra moduli (moduli CPU, unità bus CPU e computer) di una rete per trasferire dati e controllare le operazioni, ad esempio per cambiare la modalità operativa.

| Istruzione | Contenuto del messaggio | Operazione |
|------------------------|--|--|
| SEND(090) RECV(098) | Comandi per trasmettere / ricevere dati (comando FINS) | <p>CPU: SEND(090) o RECV(098)</p> <p>Invio di dati →</p> <p>Ricezione dei dati ←</p> <p>Altro dispositivo: CPU, unità bus CPU CS1 o computer</p> |
| CMND(490) | Comandi arbitrari (comando FINS) | <p>CPU: CMND(490)</p> <p>Comando inviato →</p> <p>Risposta restituita ←</p> <p>Altro dispositivo: CPU, unità bus CPU CS1 o computer</p> |

I comandi eseguiti dalle istruzioni di rete sono definiti "comandi FINS" e vengono utilizzati per le comunicazioni tra i dispositivi di controllo di automazione industriale. Per informazioni sui comandi FINS, fare riferimento al manuale *CS/CJ-Series Communications Commands Reference Manual*. I comandi FINS permettono di comunicare (in base al formato di comando/risposta) con un modulo in una rete qualsiasi o sullo stesso rack CPU specificando semplicemente l'indirizzo di rete, il numero del nodo e il numero modulo del modulo di destinazione.

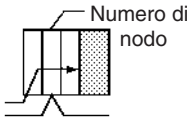
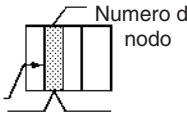
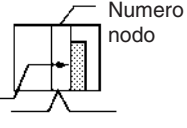
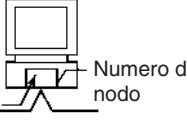
Nell'esempio seguente un comando FINS viene inviato a una CPU tramite il numero di nodo 2 in un indirizzo di rete 00.



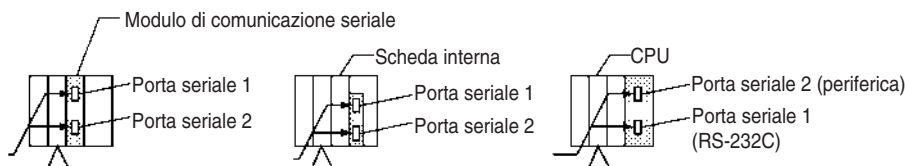
1,2,3...

1. Indirizzo di rete:
Indirizzo della rete (rete locale = 00)
2. Numero di nodo
Indirizzo logico sulla rete
3. Numero modulo
Numero modulo del modulo di destinazione
 - a) CPU: 00
 - b) Unità bus CPU: Numero modulo +10 esadecimale
 - c) Modulo di I/O speciale (ad eccezione dei moduli di I/O speciali della serie C200H):
Numero modulo +20 esadecimale

- d) Scheda interna (solo serie CS):
E1 esadecimale
- e) Computer: 01

| Numero modulo (esadecimale) | Dispositivo di destinazione |
|-----------------------------|--|
| 00 |  |
| Numero modulo +10 |  |
| E1 |  |
| 01 |  |

Nota È possibile anche specificare direttamente una porta seriale (indirizzo modulo) all'interno del dispositivo di destinazione.



Indirizzi del modulo della porta seriale:

- Porte del modulo di comunicazione seriale
Porta 1: 80 esadecimale + 4 × numero modulo

| Numero modulo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Esadecimale | 80 | 84 | 88 | 8C | 90 | 94 | 98 | 9C | A0 | A4 | A8 | AC | B0 | B4 | B8 | BC |
| Decimale | 128 | 132 | 136 | 140 | 144 | 148 | 152 | 156 | 160 | 164 | 168 | 172 | 176 | 180 | 184 | 188 |

Porta 2: 81 esadecimale + 4 × numero modulo

| Numero modulo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Esadecimale | 81 | 85 | 89 | 8D | 91 | 95 | 99 | 9D | A1 | A5 | A9 | AD | B1 | B5 | B9 | BD |
| Decimale | 129 | 133 | 137 | 141 | 145 | 149 | 153 | 157 | 161 | 165 | 169 | 173 | 177 | 181 | 185 | 189 |

- Porte della scheda di comunicazione seriale
Porta 1: E4 esadecimale (228 decimale)
Porta 2: E5 esadecimale (229 decimale)
- Porte della CPU
Porta periferica: FD esadecimale (253 decimale)
Porta RS-232C: FC esadecimale (252 decimale)

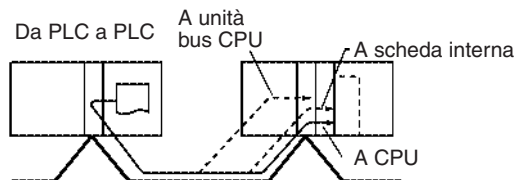
Esempi di comunicazione di rete

Negli esempi riportati di seguito vengono illustrati tre tipi di comunicazione di rete: comunicazione da un PLC a un altro dispositivo collegato in rete,

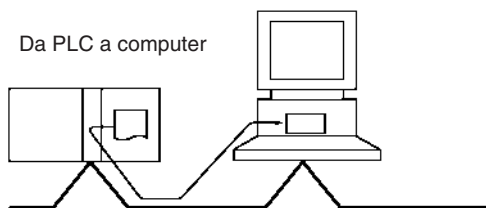
comunicazione da un PLC alle porte seriali di un altro dispositivo collegato in rete e comunicazione a un computer host collegato tramite un Host Link.

Comunicazione a un altro dispositivo collegato in rete

Nell'esempio riportato di seguito viene riprodotta la comunicazione da un PLC ai dispositivi di un altro PLC (CPU, unità bus CPU o scheda interna). Per ulteriori informazioni consultare il manuale dell'operatore della rete in uso (Controller Link o Ethernet).

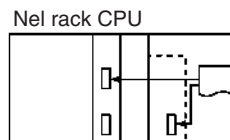
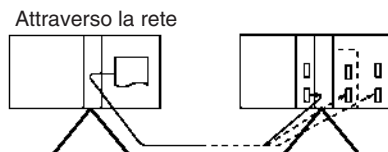


In questo esempio viene riprodotta la comunicazione da un PLC a un personal computer.

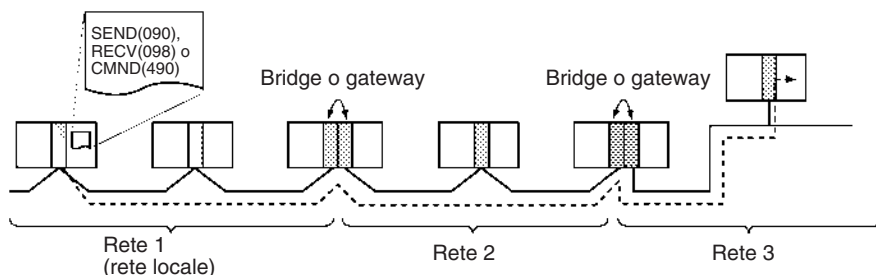


Comunicazione a una porta seriale collegata in rete

Negli esempi riportati di seguito viene illustrata la comunicazione da un PLC alle porte seriali di dispositivi collegati in rete. Nel primo esempio viene mostrata la comunicazione alle porte seriali nei dispositivi di un altro PLC (CPU, unità bus CPU o scheda interna), mentre nel secondo esempio viene mostrata la comunicazione a una porta seriale all'interno dello stesso rack CPU.



Nota È possibile espandere la comunicazione fino a 8 livelli di rete, compresa la rete locale. (La rete locale è la rete in cui ha origine la comunicazione).



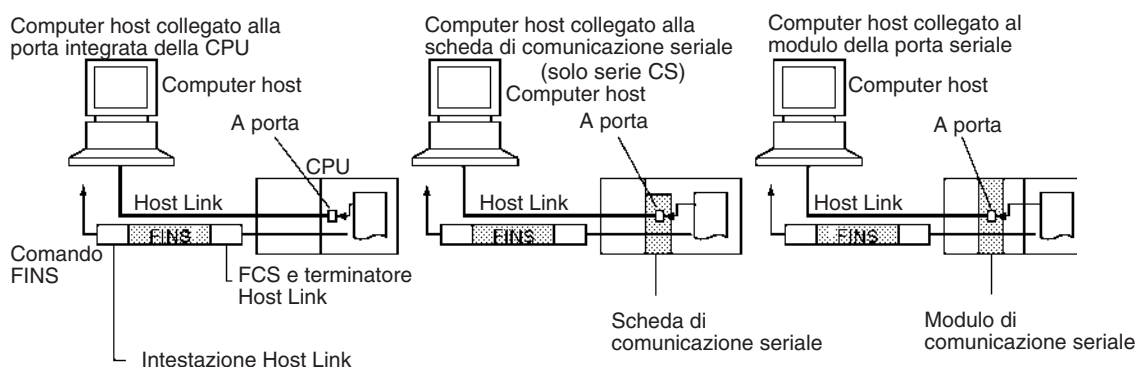
Per comunicare tramite la rete è necessario registrare una tabella di routing nella CPU di ciascun PLC. Questa tabella indica il percorso utilizzato per

trasferire i dati al nodo prescelto. Ciascuna tabella di routing è composta da una tabella della rete locale e da una tabella della rete relè.

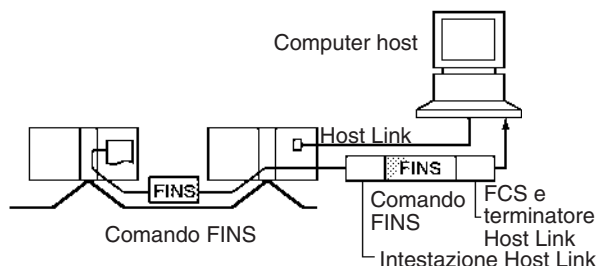
- 1,2,3...**
1. Tabella di rete locale
In questa tabella sono riportati i numeri modulo e gli indirizzi di rete dei nodi collegati al PLC locale.
 2. Tabella di rete relè
In questa tabella sono riportati i numeri di nodo e gli indirizzi di rete dei primi nodi relè verso le reti di destinazione che non sono collegate al PLC locale.

Comunicazione con un computer host (Host Link)

Quando si impartisce un'istruzione SEND(090), RECV(098) o CMND(490) a una porta seriale impostata sulla modalità Host Link, l'intestazione e il terminatore Host Link necessari verranno associati al comando FINS e il comando verrà inviato al computer host.



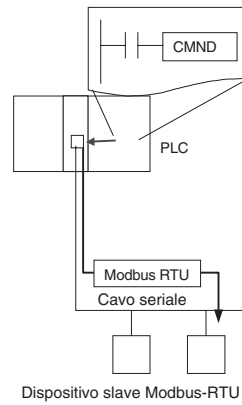
Nota È possibile inviare tramite la rete una comunicazione Host Link. In questo caso, il comando FINS viene inviato normalmente attraverso la rete. Quando il comando raggiunge il sistema Host Link, l'intestazione e il terminatore Host Link necessari vengono associati al comando FINS e il comando viene inviato al computer host.



Comunicazione tramite gateway seriale con un componente o uno slave Host Link

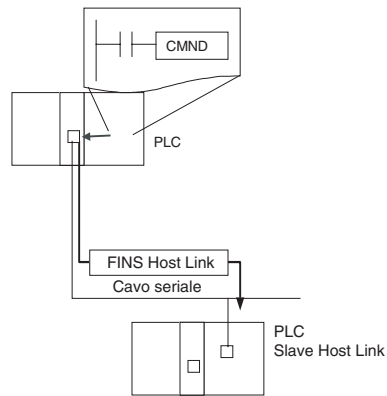
È possibile inviare comandi FINS (o inviare e ricevere dati) a un componente o a uno slave Host Link collegato al PLC tramite la rispettiva porta seriale con la funzione di gateway seriale.

- Invio a un componente
Quando un'istruzione CMND(490) viene eseguita su una porta seriale che supporta la funzione di gateway seriale, tale funzione converte il comando in un comando CompoWay/F, Modbus-RTU o Modbus-ASCII.



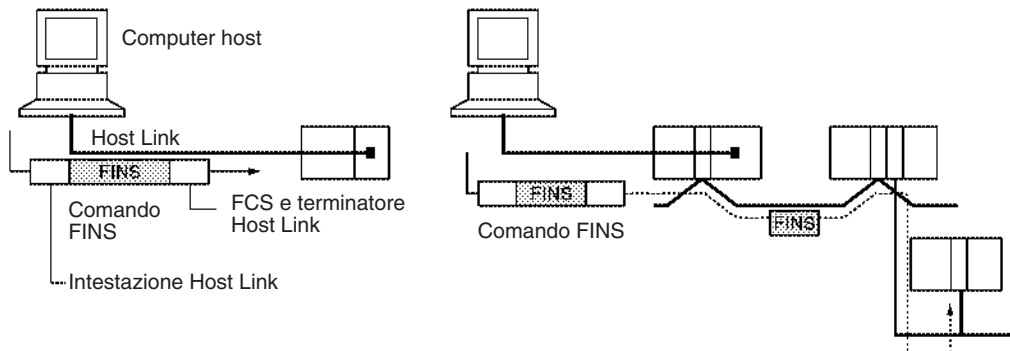
- Invio a un PLC operante come slave Host Link

Quando viene eseguita un'istruzione CMND(490), SEND(090) o RECV(098) su una porta seriale che supporta la funzione di gateway seriale, tale funzione può inviare qualsiasi comando FINS o inviare e ricevere dati.



Comunicazione da un computer host (Host Link)

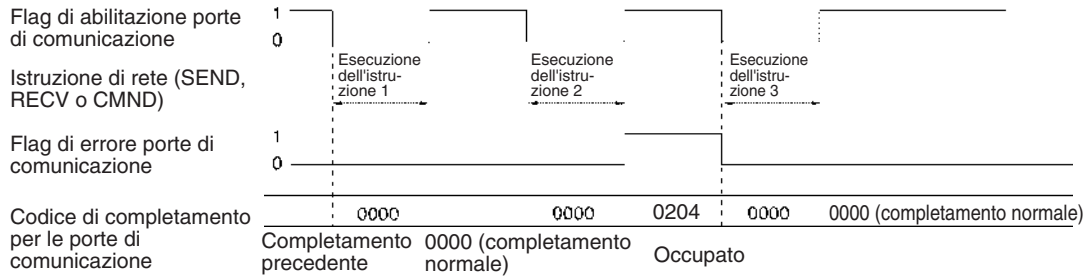
È possibile inviare comandi FINS da un computer host a un PLC con cui è collegato e ad altri dispositivi collegati in rete (CPU, moduli di I/O speciali, computer ecc.). In questo caso, quando si invia il comando FINS occorre associare l'intestazione e il terminatore Host Link necessari.



Flag di comunicazione

Il funzionamento dei flag di comunicazione è descritto qui di seguito.

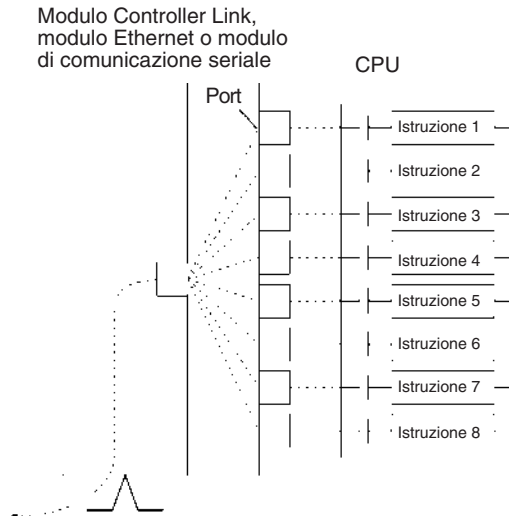
- Il flag di abilitazione porte di comunicazione viene resettato su 0 quando la comunicazione è in corso e impostato su 1 quando la comunicazione è stata completata (normalmente o con errori).
- Lo stato del flag di errore porte di comunicazione viene mantenuto fino alla volta successiva in cui si trasmettono o si ricevono dati.
- Il flag di errore porte di comunicazione viene resettato su 0 la volta successiva che si trasmettono o si ricevono dati, perfino se si verifica un errore nell'operazione precedente.



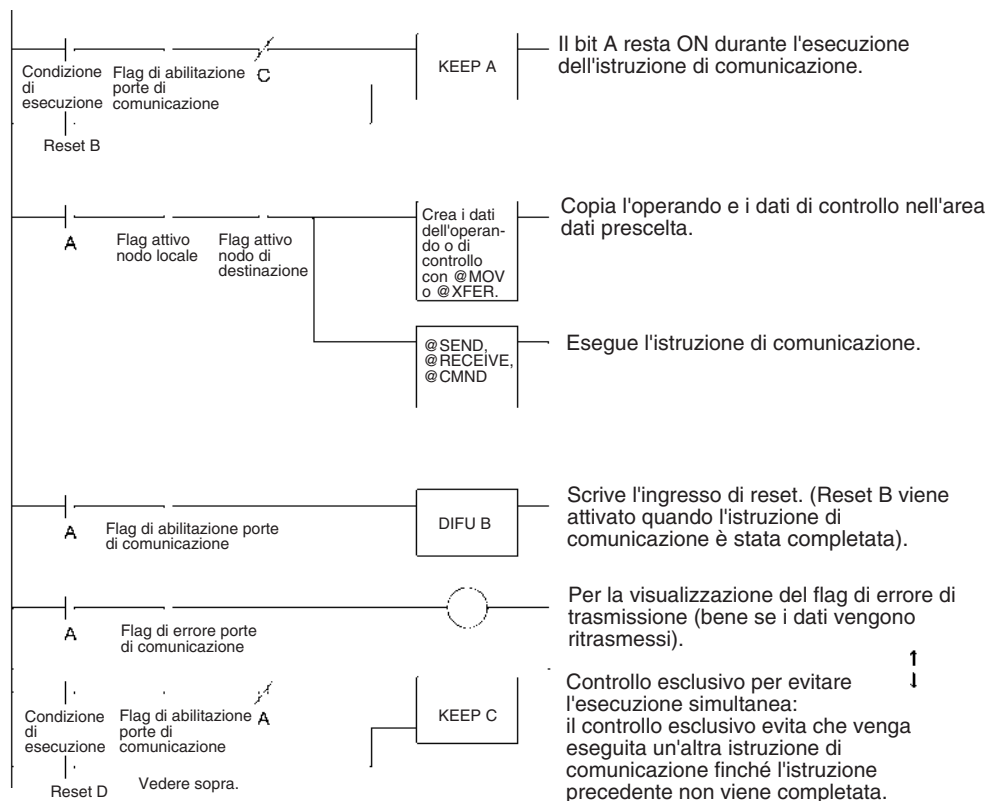
Informazioni sui numeri delle porte di comunicazione

Esistono 8 porte di comunicazione logiche, quindi è possibile eseguire contemporaneamente 8 istruzioni di comunicazione. Per ciascuna porta di comunicazione è possibile eseguire solo un'istruzione per volta. Quando vengono eseguite più di 8 istruzioni è necessario utilizzare un controllo esclusivo.

Questi 8 numeri delle porte di comunicazione vengono condivisi dalle istruzioni di rete (SEND(090), RECV(098) e CMND(490)), dalle istruzioni di comunicazione seriale (TXDU(256) e RXDU(255)) e dall'istruzione PROTOCOL MACRO (PMCR(260)). Verificare di non avere specificato contemporaneamente lo stesso numero di porta in due istruzioni.



Lo schema riportato di seguito illustra un esempio di controllo esclusivo.



Allocazione automatica delle porte di comunicazione

■ Panoramica

Tutte le istruzioni seguenti utilizzano una porta di comunicazione (porta logica) tra le porte da 0 a 7.

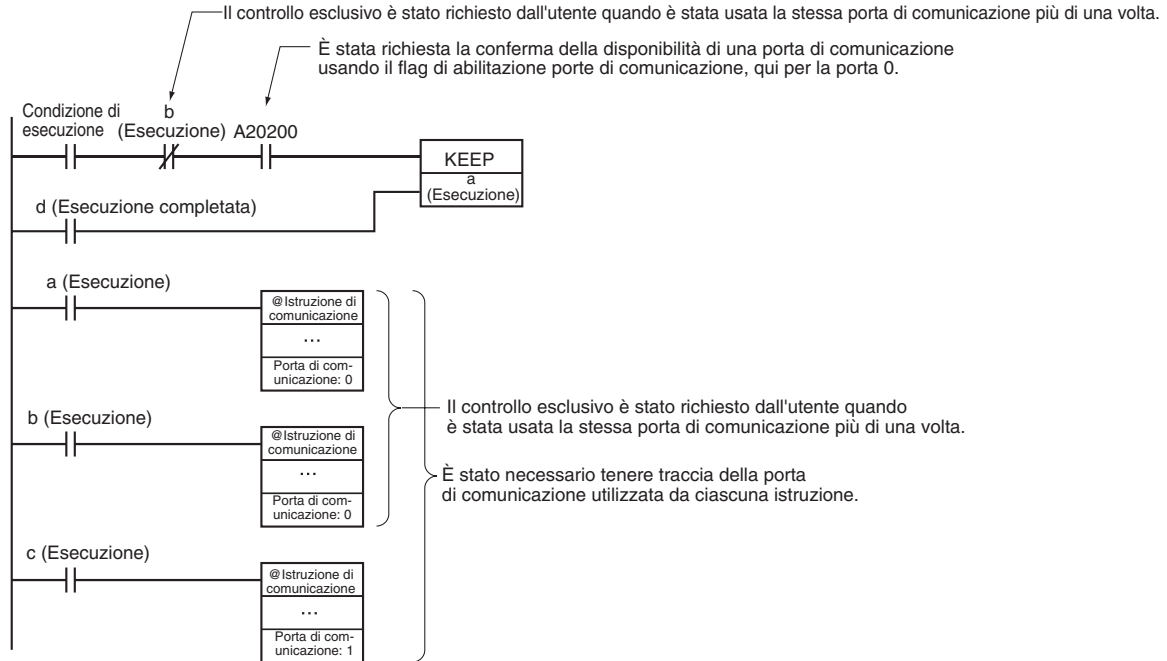
- Istruzioni di comunicazione in rete: SEND(090), RECV(098) e CMND(490)
- Istruzioni per la comunicazione seriale: PMCR(260), TXDU(256) e RXDU(255)

In questa sezione si fa riferimento a queste istruzioni come istruzioni di comunicazione.

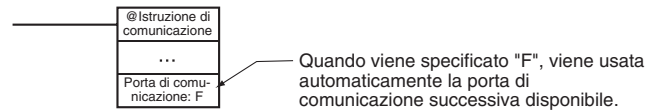
Ogni porta di comunicazione può essere utilizzata solo da una istruzione per volta. Per utilizzare le porte di comunicazione, è necessario che siano stati eseguiti i passi riportati qui di seguito.

- Durante la programmazione, è necessario annotare le porte di comunicazione utilizzate per poterle designare negli operandi.
- Nel programma ladder, deve essere stata confermata la disponibilità delle porte di comunicazione prima di utilizzarle.

Esempio dei requisiti per la precedente programmazione



Per le CPU CS1-H, CJ1-H, CJ1M e CS1D con numero di lotto 020601 o successivo (prodotte il 1° giugno 2002 o in seguito), il numero di porta può essere specificato come "F" anziché con 0 ... 7 per allocare automaticamente la porta di comunicazione, ossia per usare automaticamente la porta di comunicazione successiva aperta.



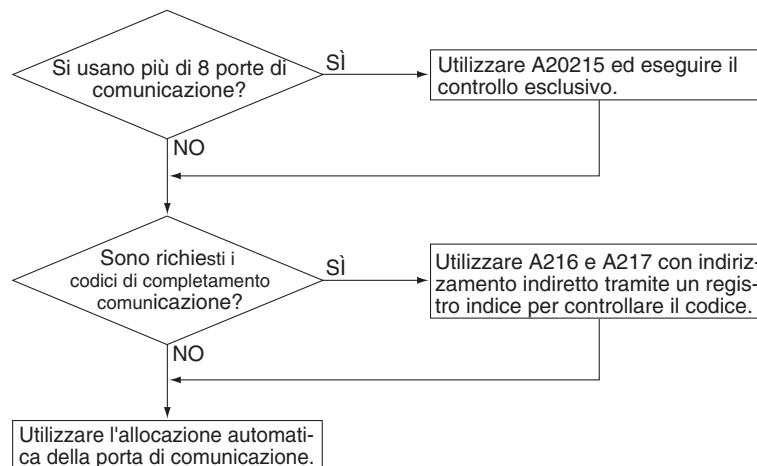
In questo modo il programmatore non deve necessariamente tenere traccia delle porte di comunicazione durante la programmazione. Le differenze tra l'assegnazione di numeri di porta specifici e l'allocazione automatica di numeri di porta sono mostrate nella tabella seguente.

| Elemento | Assegnazione di un numero specifico | Allocazione automatica |
|---|---|---|
| Specifica del numero della porta di comunicazione nei dati di controllo | Da 0 a 7 | F |
| Controllo esclusivo | Richiesta | Non richiesta, a meno che siano necessarie contemporaneamente 8 porte di comunicazione. |
| Applicazioni dei flag | LD o LD NOT utilizzate con il flag corrispondente alla porta di comunicazione specificata. | TST(350) o TSTN(351) utilizzate con A218 (numero della porta di comunicazione utilizzata). |
| Codice di completamento delle comunicazioni di rete | Viene fornito l'accesso al codice di completamento per la porta di comunicazione specificata dall'utente. | Si accede ai codici di completamento utilizzando l'indirizzo della memoria di I/O salvato in A216 e A217 (indirizzo di memorizzazione del codice di completamento delle comunicazioni di rete) e l'indirizzo indiretto del registro l'indice. |

■ **Bit e canali dell'area ausiliaria utilizzati per l'allocazione automatica delle porte di comunicazione**

| Indirizzo | Bit | Nome | Descrizione |
|-------------|------------|---|--|
| A202 | 15 | Flag di abilitazione dell'allocazione delle porte di comunicazione di rete | Viene attivato quando è disponibile una porta di comunicazione per l'allocazione automatica. Questo flag può essere utilizzato per confermare se tutte le otto porte di comunicazione sono già state allocate prima di eseguire le istruzioni di comunicazione. |
| A214 | Da 00 a 07 | Flag del primo ciclo al termine delle comunicazioni di rete | Ogni flag viene attivato solo per un ciclo al completamento delle comunicazioni. I bit da 00 a 07 corrispondono alle porte da 0 a 7. Utilizzare il numero di porta di comunicazione memorizzato in A218 per determinare a quale flag accedere. Nota: Questi flag non sono significativi finché non viene eseguito il ciclo successivo all'istruzione di comunicazione. Non accedervi per almeno un ciclo. |
| | Da 08 a 15 | Non utilizzare | |
| A215 | Da 00 a 07 | Flag del primo ciclo dopo un errore di comunicazione di rete | Ogni flag viene attivato solo per un ciclo dopo che si è verificato un errore di comunicazione. I bit da 00 a 07 corrispondono alle porte da 0 a 7. Utilizzare il numero di porta di comunicazione memorizzato in A218 per determinare a quale flag accedere. Nota: Questi flag non sono significativi finché non viene eseguito il ciclo successivo all'istruzione di comunicazione. Non accedervi per almeno un ciclo. |
| | Da 08 a 15 | Non utilizzare | |
| A216 e A217 | --- | Indirizzo di memorizzazione del codice di completamento delle comunicazioni di rete | Il codice di completamento per un'istruzione di comunicazione viene automaticamente memorizzato nell'indirizzo della memoria di I/O fornito in questi canali. Inserire l'indirizzo in un registro indice e utilizzarlo indirettamente mediante tale registro per raggiungere il codice di completamento delle comunicazioni. |
| A218 | --- | Numero della porta di comunicazione utilizzata | Quando viene eseguita un'istruzione di comunicazione, il numero della porta di comunicazione che è stata usata viene memorizzato in questo canale. I valori da 0000 a 0007 esadecimale corrispondono alle porte di comunicazione da 0 a 7. |

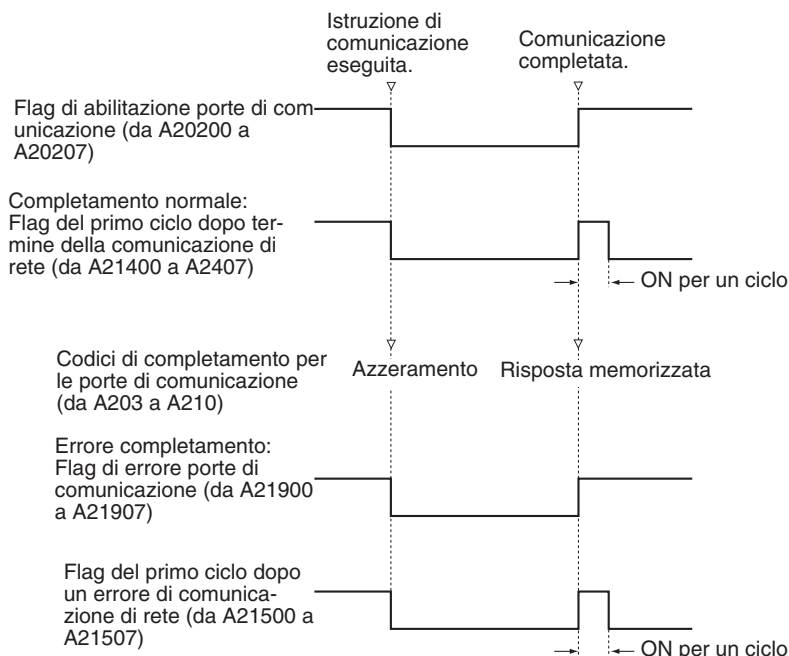
- Nota** 1. Utilizzare il seguente diagramma di flusso per stabilire se utilizzare il Flag di abilitazione dell'allocazione delle porte di comunicazione di rete (A20215) e l'indirizzo di memorizzazione del codice di completamento delle comunicazioni di rete (A216 e A217).



2. Nella tabella seguente sono elencati i bit e i canali dell'area ausiliaria per le porte di comunicazione specificate dall'utente.

| Indirizzo | Bit | Nome | Descrizione |
|-----------------|------------|---|---|
| A202 | Da 00 a 07 | Flag di abilitazione porte di comunicazione | ON quando un'istruzione di comunicazione può essere eseguita con il numero di porta corrispondente. I bit da 00 a 07 corrispondono alle porte di comunicazione da 0 a 7. Il completamento della comunicazione può essere confermato monitorando il momento in cui si attiva un flag. Il flag si disattiva quando viene avviata l'esecuzione di un'istruzione di comunicazione. |
| Da A203 ad A210 | --- | Codici di completamento per le porte di comunicazione | Questi canali contengono i codici di completamento per i numeri delle porte corrispondenti al termine dell'esecuzione di un'istruzione di comunicazione. I canali da A203 ad A210 corrispondono alle porte di comunicazione da 0 a 7. |
| A219 | Da 00 a 07 | Flag di errore porte di comunicazione | ON quando si verifica un errore durante l'esecuzione di un'istruzione di comunicazione. Quando un flag è ON, controllare il codice di completamento in A203 - A210 per risolvere l'errore. Viene disattivato se l'esecuzione viene completata normalmente. I bit da 00 a 07 corrispondono alle porte di comunicazione da 0 a 7. |

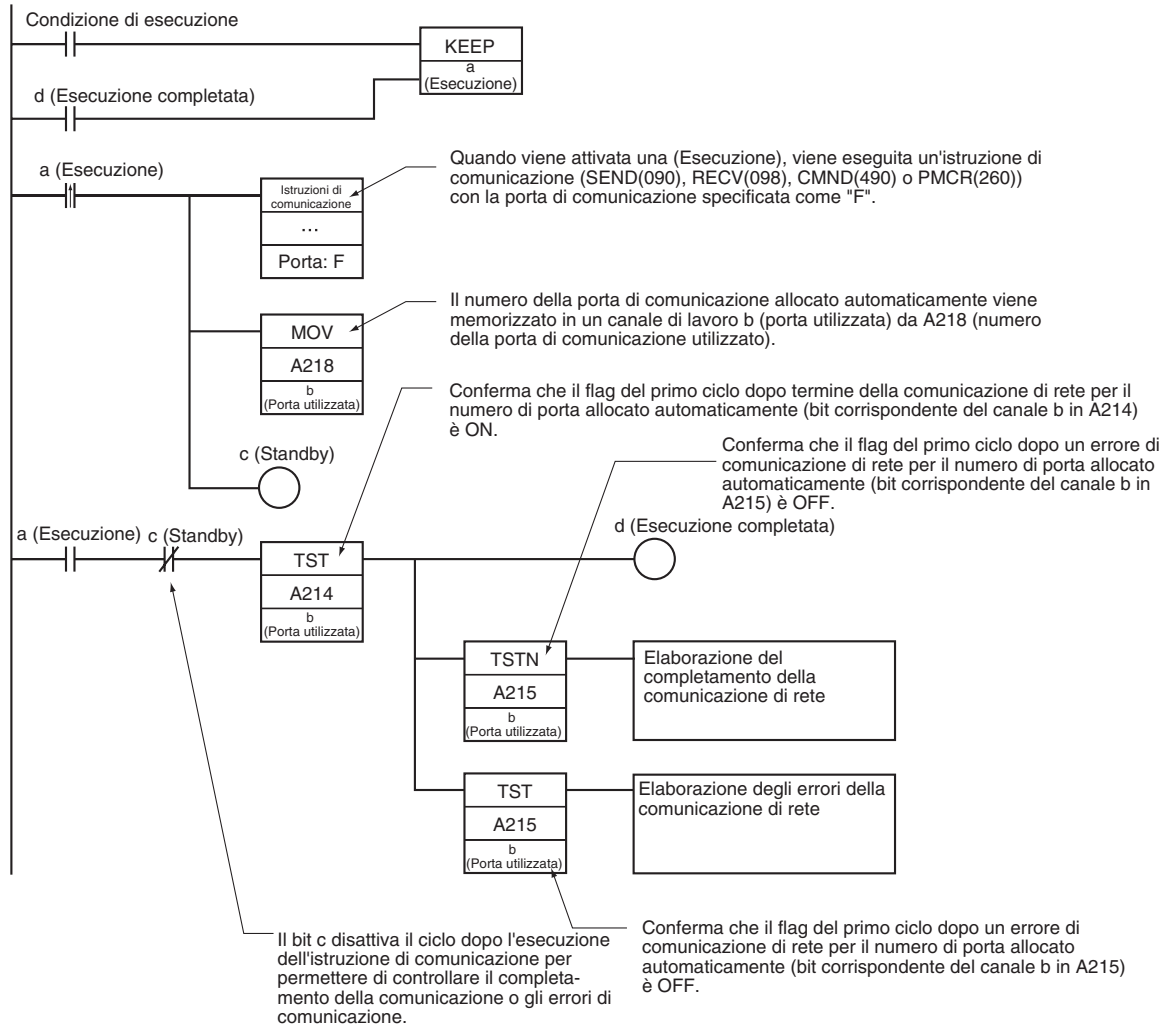
Funzionamento del flag/canale



■ Metodi di applicazione

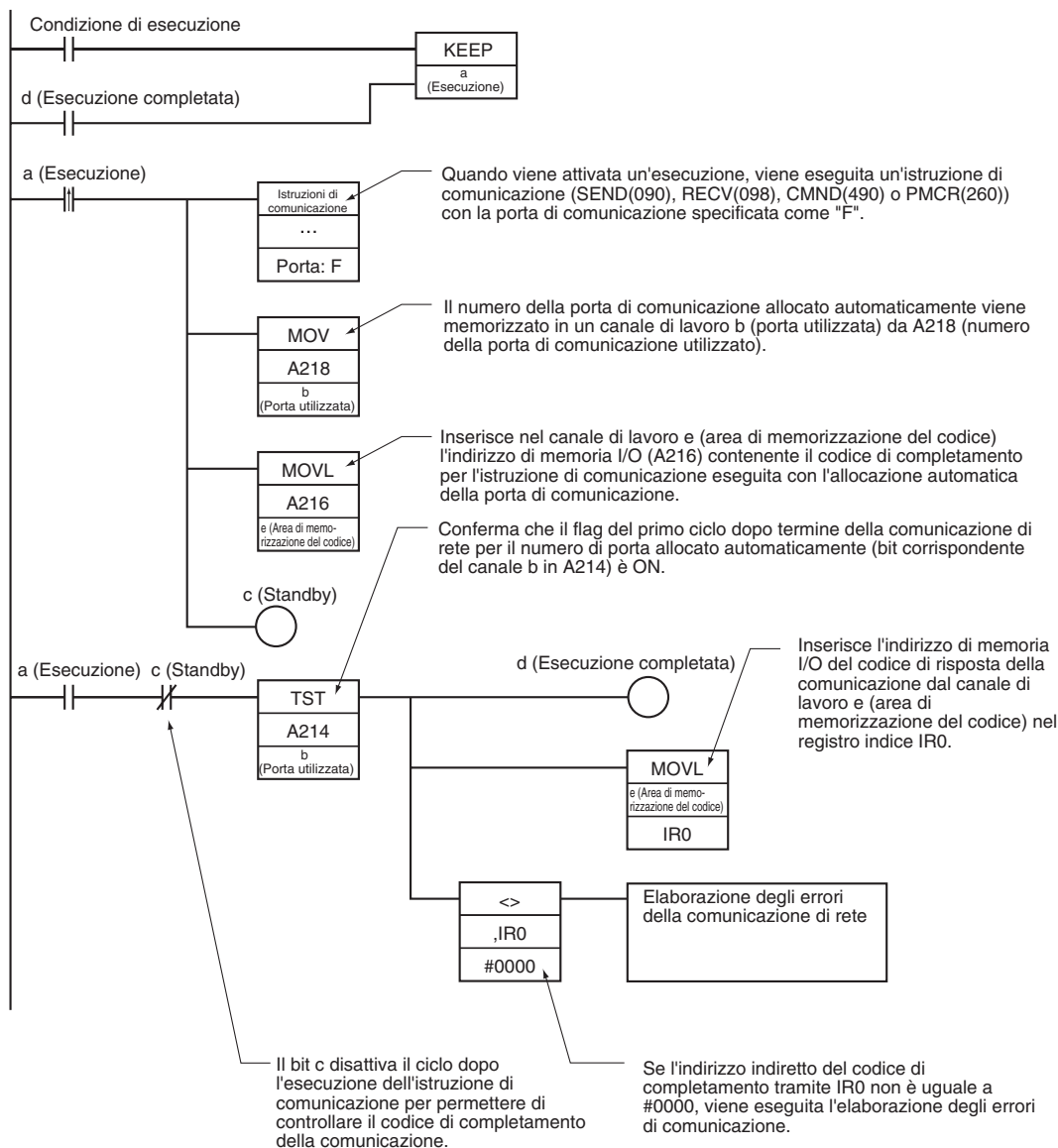
Per utilizzare l'allocazione automatica delle porte di comunicazione, impostare il numero della porta di comunicazione su "F", quindi eseguire la programmazione come mostrato di seguito.

Completamento ed elaborazione degli errori dopo l'esecuzione delle istruzioni di comunicazione



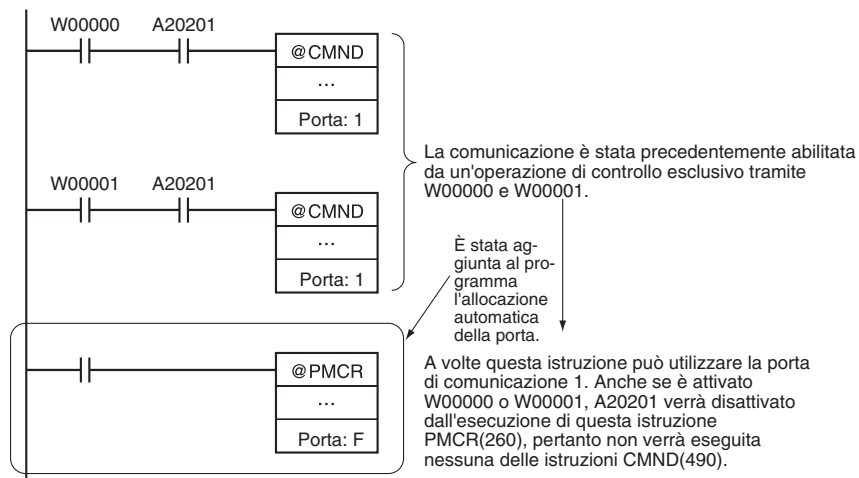
Accesso al codice di completamento dopo l'esecuzione delle istruzioni di comunicazione

I codici di completamento vengono generalmente usati per risolvere gli errori quando questi si verificano. È possibile, tuttavia, utilizzare un codice di completamento 0000 esadecimale per confermare che la comunicazione è stata completata regolarmente.



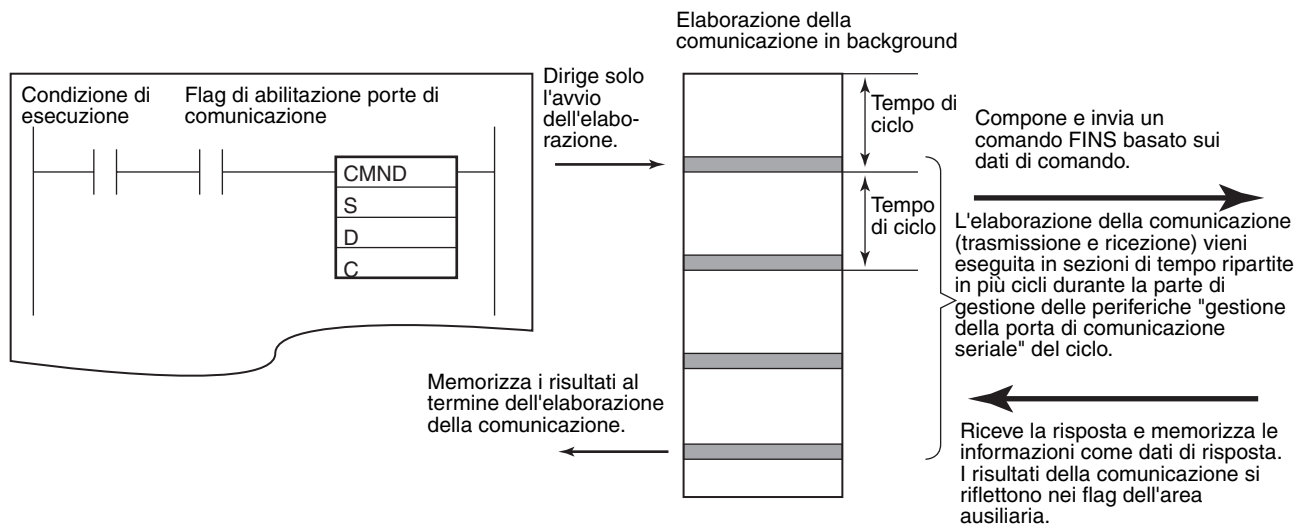
Nota Nello stesso programma è possibile utilizzare sia numeri di porte di comunicazione specificati dall'utente, sia numeri di porte di comunicazione specificati automaticamente. Tuttavia, è possibile che i numeri delle porte di comunicazione specificati dall'utente vengano utilizzati per l'allocazione automatica. È importante, perciò, controllare il programma con attenzione quando si aggiungono istruzioni di comunicazione che utilizzano l'allocazione automatica delle porte di comunicazione in un programma esistente, come mostrato nell'esempio seguente.

Esempio di programmazione



Temporizzazione dell'esecuzione delle istruzioni di rete

Un'istruzione di rete avvia l'elaborazione della comunicazione quando risulta soddisfatta la relativa condizione di esecuzione. L'effettiva elaborazione della comunicazione viene eseguita in background nella parte di gestione delle periferiche che si occupa della "gestione della porta di comunicazione seriale".



L'elaborazione della comunicazione si svolge come descritto di seguito:

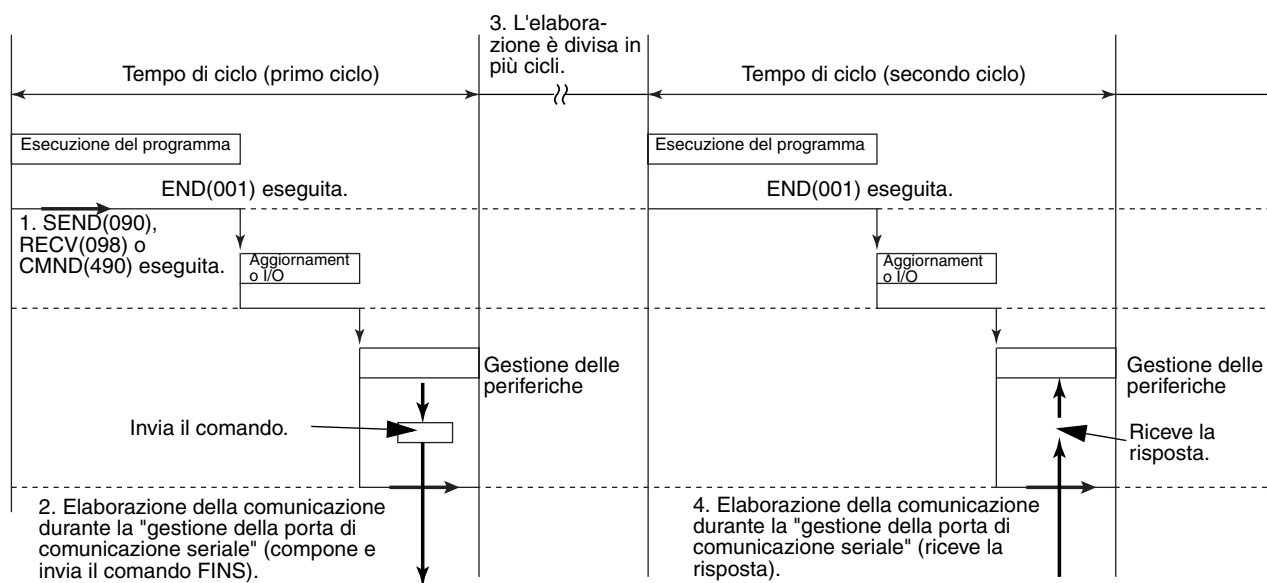
1. Se il flag di abilitazione porte di comunicazione (da A20200 a A20207) è ON quando viene stabilita la condizione di esecuzione, il sistema esegue i processi seguenti:
 - Attiva il flag di abilitazione porte di comunicazione della porta e il flag di errore porte di comunicazione (da A21900 a A21907).
 - Imposta il codice di completamento per le porte di comunicazione della porta (da A203 a A210) su 0000.
 - Legge i canali di controllo (cominciando da C) e avvia l'elaborazione della comunicazione, inviando un comando FINS o ricevendo una risposta.
2. Nella parte del ciclo di gestione delle periferiche relativa alla "gestione della porta di comunicazione seriale", il sistema compone un comando FINS basato sugli operandi (vedere la nota) e invia il comando FINS al modulo di comunicazione o a un altro nodo di destinazione.

Nota Quando viene eseguita l'istruzione SEND(090), vengono letti i contenuti di S e D e viene composto un comando FINS per la trasmissione dei dati.

Quando viene eseguita l'istruzione RECV(098), vengono letti i contenuti di S e D e viene composto un comando FINS per la ricezione dei dati.

Quando viene eseguita l'istruzione CMND(490), vengono letti i contenuti di S e D e viene composto il comando FINS corrispondente.

3. Se l'elaborazione dell'invio non può essere completata nel periodo disponibile in "gestione della porta di comunicazione seriale", l'elaborazione verrà continuata nella gestione della porta di comunicazione seriale del ciclo successivo.
4. Quando viene restituita una risposta, il sistema esegue i seguenti processi:
 - Aggiorna i canali di destinazione specificati nell'istruzione di rete con i dati della risposta.
 - Attiva il flag di abilitazione porte di comunicazione della porta.
 - Aggiorna il flag di errore porte di comunicazione (da A21900 a A21907) e il codice di completamento per le porte di comunicazione (da A203 a A210) relativi alla porta.



3-25-2 Informazioni sulle istruzioni per messaggi espliciti

Metodi per l'utilizzo della comunicazione per messaggi espliciti

Esistono due metodi per inviare messaggi espliciti da un PLC.

- Utilizzare l'istruzione CMND(490) per inviare il codice di un comando FINS di 2801 esadecimale (EXPLICIT MESSAGE SEND).
- Utilizzare le istruzioni per messaggi espliciti riportate di seguito (vedere nota).

Nota Queste istruzioni sono supportate solo da CPU della serie CS/CJ versione 2.0 o successiva.

Istruzioni per messaggi espliciti

Le istruzioni seguenti, utilizzate specialmente per messaggi espliciti, sono definite istruzioni per messaggi espliciti.

| Istruzione | Nome | Profilo |
|------------|------------------------|--|
| EXPLT(720) | EXPLICIT MESSAGE SEND | Invia un messaggio esplicito con qualsiasi codice di servizio. Nota: a livello funzionale, questa istruzione è identica all'invio di un'istruzione CMND(490) con il codice di un comando FINS di 2801 esadecimale. |
| EGATR(721) | EXPLICIT GET ATTRIBUTE | Invia un messaggio esplicito con un codice di servizio di 0E esadecimale (GET ATTRIBUTE SINGLE). |
| ESATR(721) | EXPLICIT SET ATTRIBUTE | Invia un messaggio esplicito con un codice di servizio di 10 esadecimale (SET ATTRIBUTE SINGLE). |
| EGATR(721) | EXPLICIT WORD READ | Utilizza un messaggio esplicito per leggere i dati da una CPU. |
| EGATR(721) | EXPLICIT WORD WRITE | Utilizza un messaggio esplicito per scrivere i dati in una CPU. |

Caratteristiche delle istruzioni per messaggi espliciti

- Le istruzioni per messaggi espliciti non richiedono l'assegnazione di un comando FINS di 2801 esadecimale e sono molto più semplici da programmare rispetto all'istruzione CMND(490).
- Con le istruzioni EXPLICIT GET/SET ATTRIBUTE non occorre inserire il codice di servizio e devono essere immesse soltanto le informazioni dall'ID classe in avanti.
- Con le istruzioni EXPLICIT WORD READ/WRITE, è possibile specificare direttamente l'indirizzo di memoria di I/O nella CPU locale e in quella remota.

Non sono necessarie specifiche del codice per tipi di area e indirizzi di canali esadecimale. Queste informazioni sono necessarie per le istruzioni CMND(490) con il codice di servizio 1E (lettura dati del canale) o 1F esadecimale (scrittura dati del canale).

Ciò permette di eseguire in modo semplice operazioni di lettura e scrittura di dati tra le CPU che utilizzano la comunicazione per messaggi espliciti, come le istruzioni SEND/RECV per i comandi FINS.

Operazione

Viene utilizzato il flag di errore di comunicazione esplicita per stabilire se la comunicazione è terminata normalmente o con errori.

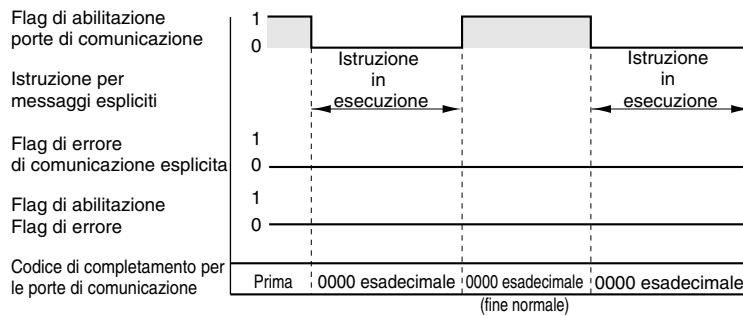
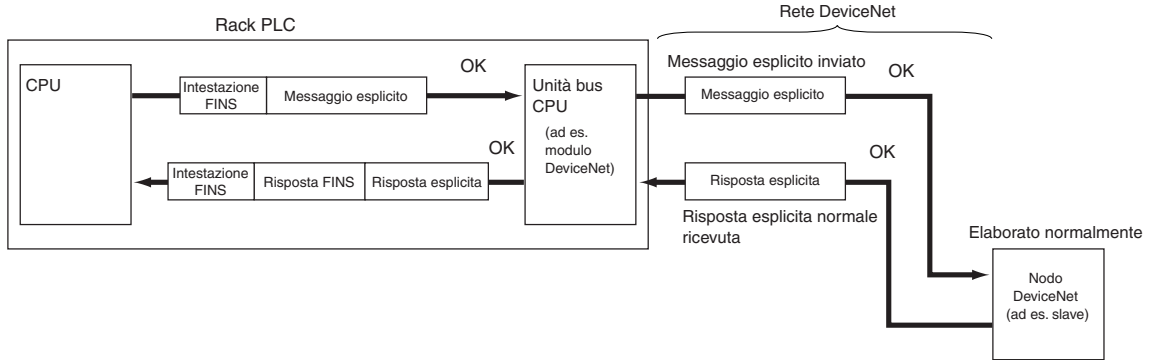
Per il completamento degli errori (ossia quando il flag è ON), viene utilizzato il flag di errore porte di comunicazione per i comandi FINS per stabilire se il messaggio esplicito non è stato mai inviato (ossia quando il flag è ON) oppure se era presente un errore nel messaggio esplicito inviato (ossia quando il flag è OFF).

Il codice di completamento per le porte di comunicazione conterrà 0000 esadecimale se l'invio è terminato regolarmente, un codice di errore relativo al messaggio esplicito se terminato con un errore di comunicazione esplicita e un codice di completamento messaggio FINS se terminato con un errore FINS.

| Condizione | | Flag di errore di comunicazione esplicita (da A21300 a A21307: porta di comunicazione n. 0 ... 7) | Flag di errore porte di comunicazione (da A21900 a A21907: porta di comunicazione n. 0 ... 7) | Codice di completamento per le porte di comunicazione (da A203 a A210: porta di comunicazione n. 0 ... 7) |
|--------------------|--|---|---|---|
| 1) Fine normale | | OFF | OFF | 0000 esadecimale |
| 2) Fine con errore | a) Quando non è stato possibile inviare il messaggio esplicito | ON | ON | Codice di completamento messaggi FINS |
| | b) Quando il messaggio esplicito è stato inviato ma è tornata una risposta di errore esplicito | | OFF | Codice di errore del messaggio esplicito |

1) Fine normale

Viene inviato un messaggio esplicito e viene ricevuta una risposta normale. Il flag di errore di comunicazione esplicita corrispondente (da A21300 a 07: porta di comunicazione n. 0 ... 7) sarà OFF e il codice di risposta della comunicazione di rete (da A203 a A210: porta di comunicazione n. 0 ... 7) conterrà un codice di risposta normale messaggio esplicito di 0000 esadecimale.



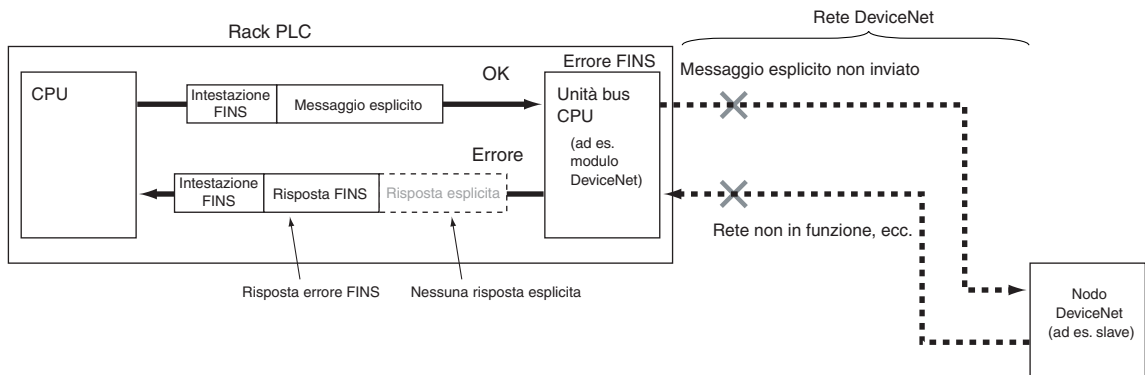
2) Fine con errore

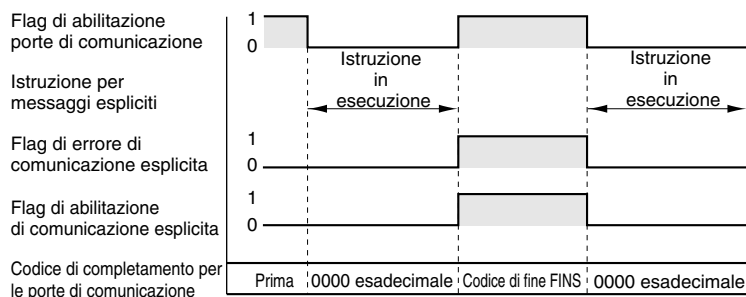
Esistono due possibilità di fine con errore, come descritto nelle due sezioni successive.

a) Quando non è stato possibile inviare il messaggio esplicito

In questo caso, il messaggio esplicito non è mai stato inviato sulla rete, ad esempio perché la rete non era in funzione. Qui, sia il flag di errore di comunicazione esplicita (da A21300 a A21307: porta di comunicazione n. 0 ... 7) che il flag di errore porte di comunicazione (da A21900 a A21907: porta di comunicazione n. 0 ... 7) verranno impostati su ON.

Dopo il completamento, il codice di completamento per le porte di comunicazione (da A203 a A210: porta di comunicazione n. 0 ... 7) conterrà il codice di errore messaggio FINS.

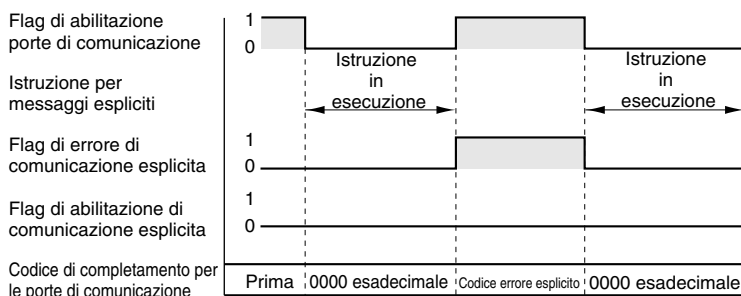
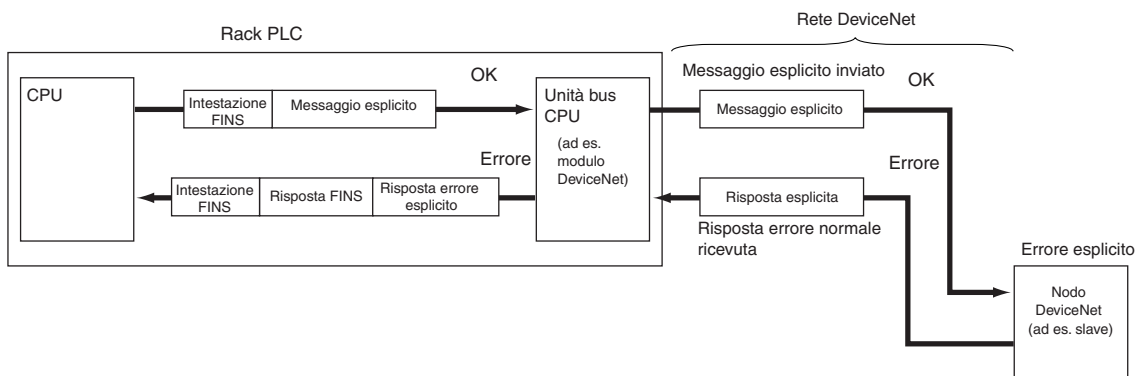




b) Quando il messaggio esplicito è stato inviato ma è tornata una risposta di errore esplicito

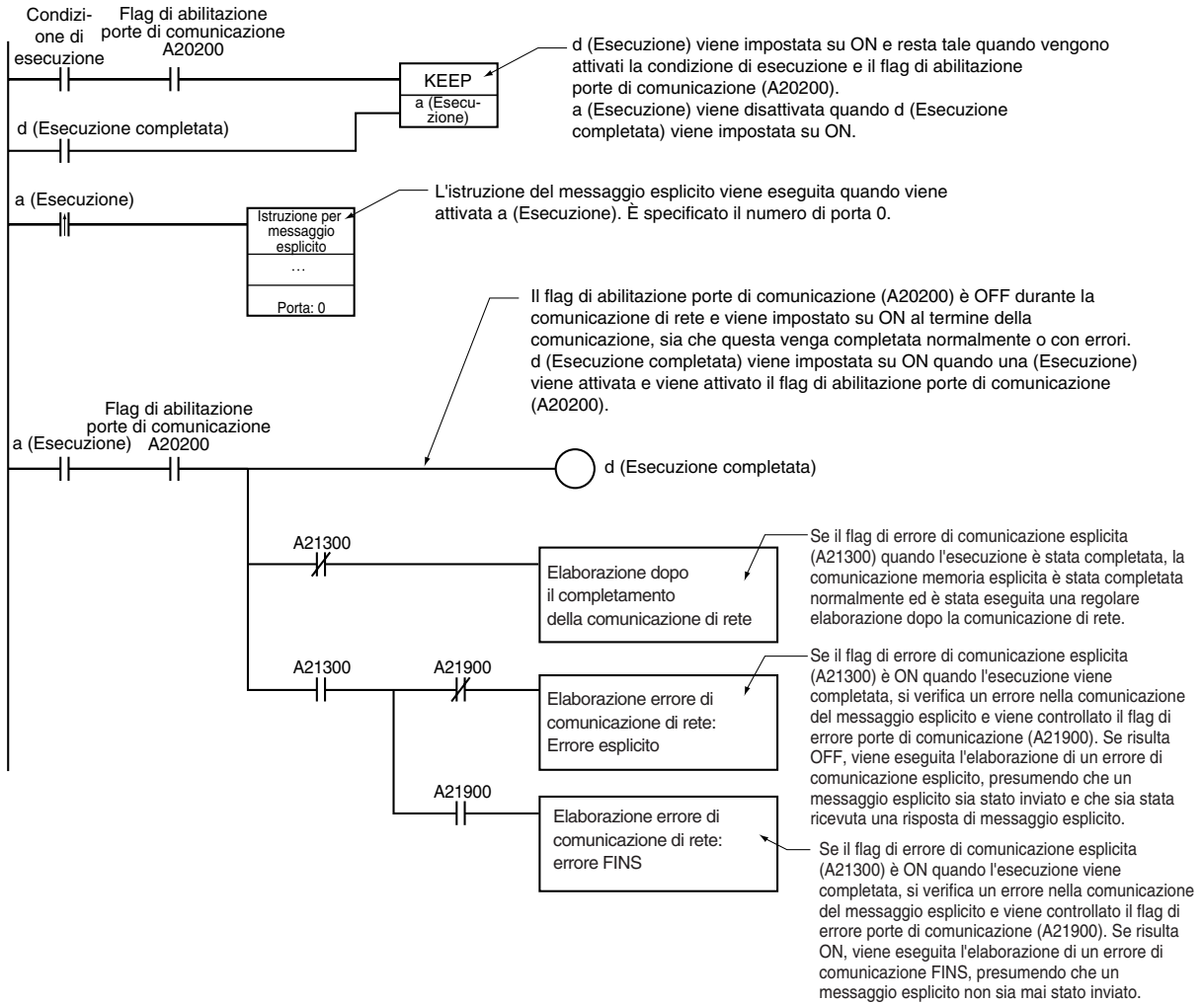
In questo caso, il messaggio esplicito è stato inviato ma esisteva un errore nel frame di comando del messaggio esplicito (codice non supportato, dimensioni non consentite, ecc.). Qui, il flag di errore di comunicazione esplicita (da A21300 a 07: porta di comunicazione n. 0 ... 7) verrà impostato su ON e il flag di errore di comunicazione di rete (da A21900 a 07: porta di comunicazione n. 0 ... 7) resterà OFF.

Dopo il completamento, il codice di risposta della comunicazione di rete (da A203 a A210: porta di comunicazione n. 0 ... 7) conterrà il codice di errore messaggio esplicito.

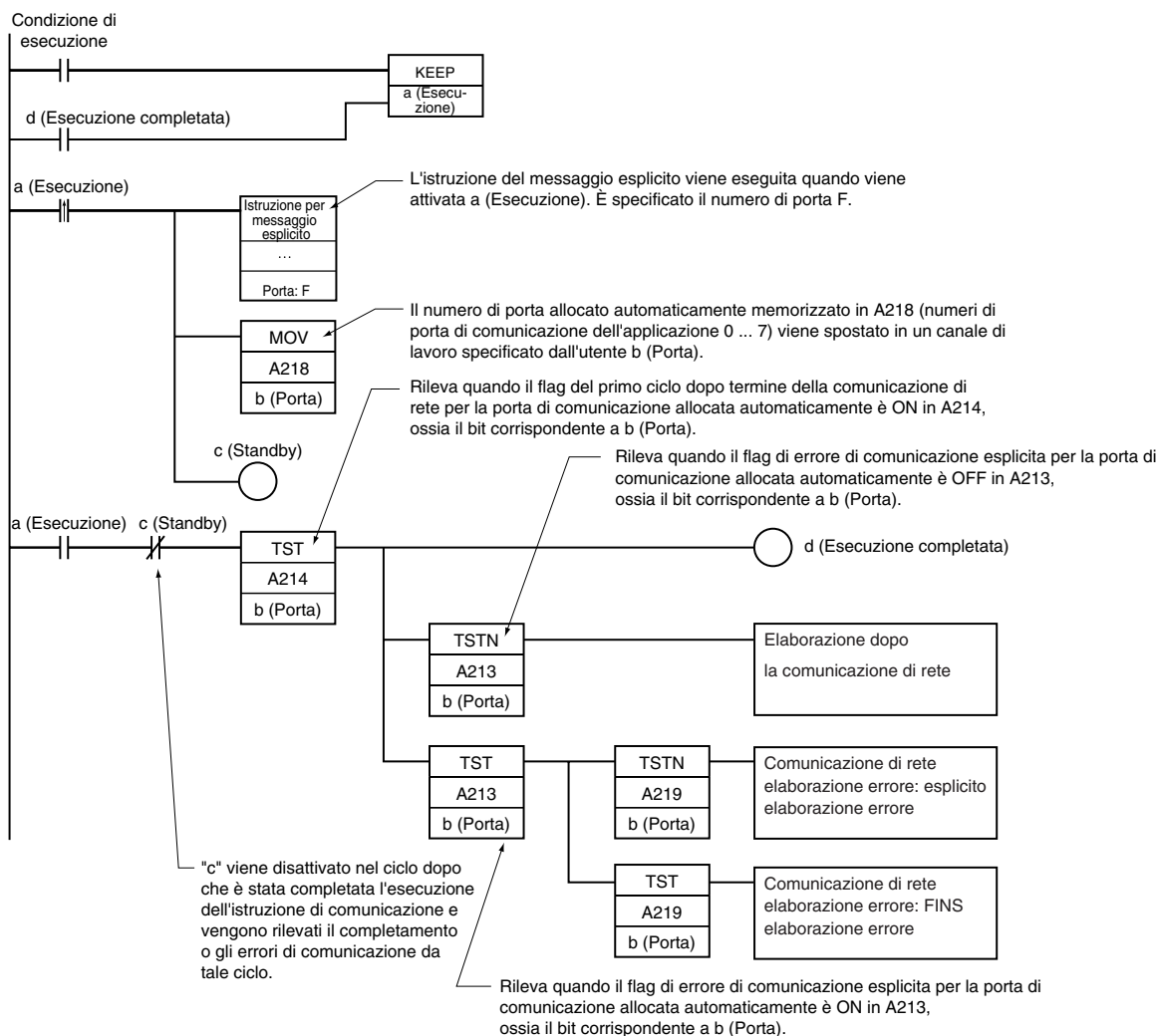


Esempi di programmazione ladder

Esempio 1: numero della porta di comunicazione specificato dall'utente



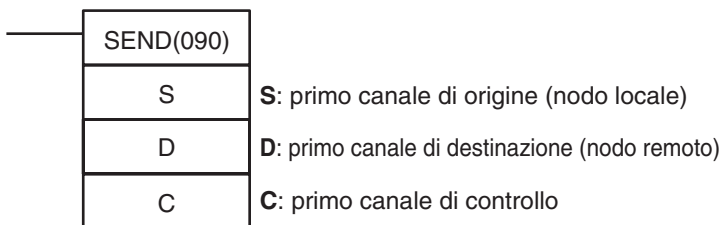
Esempio 2: allocazione automatica del numero della porta di comunicazione



3-25-3 NETWORK SEND: SEND(090)

Scopo Invia i dati a un nodo nella rete.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SEND(090) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SEND(090) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: primo canale di controllo

I cinque canali di controllo da C a C+4 specificano il numero di canali da trasmettere, la destinazione e altre impostazioni riportate nella tabella seguente.

| Canale | Bit da 00 a 07 | Bit da 08 a 15 |
|--------|--|---|
| C | Numero di canali: da 0001 al massimo consentito ¹ (esadecimali a 4 cifre) | |
| C+1 | Indirizzo rete destinazione: da 00 a 7F (da 0 a 127) ^{2, 4} | Bit da 08 a 11: numero di porta seriale ³ (porta fisica) 1 esadecimale: Porta 1 2 esadecimale: Porta 2 (non impostare 0, 3 o 4). Bit da 12 a 15: sempre 0. |
| C+2 | Indirizzo modulo di destinazione: da 00 a FE ⁵ | Indirizzo nodo di destinazione: da 00 al massimo consentito ⁶ |
| C+3 | N. di tentativi: da 00 a 0F (da 0 a 15) | Bit da 08 a 11: Numero porta di comunicazione (porta logica interna): da 0 a 7, allocazione automatica: F ⁷ Bit da 12 a 15: impostazione risposta 0: risposta richiesta. 8: nessuna impostazione richiesta. ⁸ |
| C+4 | Tempo di monitoraggio della risposta: da 0001 a FFFF (da 0,1 a 6553,5 secondi) (L'impostazione predefinita 0000 imposta un tempo di monitoraggio di 2 secondi). | |

Nota

1. Il numero massimo di canali consentiti dipende dalla rete in uso. Per Controller Link l'intervallo ammesso è da 0001 a 03DE (da 1 a 990 canali).
2. Impostare l'indirizzo della rete di destinazione su 00 per trasmettere all'interno della rete locale. Quando sono montate due o più unità bus CPU, l'indirizzo di rete sarà il numero di modulo del modulo avente il numero di modulo più basso.
3. I due metodi riportati di seguito possono essere utilizzati per inviare dati al computer host tramite una porta seriale con host link avviando la comunicazione dal PLC.
 - a) Impostare l'indirizzo del modulo di destinazione (bit da 00 a 07 di C+2) sull'indirizzo del modulo della CPU o della scheda/modulo di comunicazione seriale e impostare il numero della porta seriale (bit da 08 a 11 di C+1) su 1 per la porta 1 o 2 per la porta 2.

| Indirizzo modulo (C+2, bit da 00 a 07) | Modulo | Numero di porta seriale (C+1, bit da 08 a 11) | Porta seriale |
|--|--|---|-------------------------|
| 00 esadecimale | CPU | 1 esadecimale | Porta RS-232C integrata |
| | | 2 esadecimale | Porta periferiche |
| 10 esadecimale + numero modulo | Modulo di comunicazione seriale (unità bus CPU) | 1 esadecimale | Porta 1 |
| | | 2 esadecimale | Porta 2 |
| E1 esadecimale | Scheda di comunicazione seriale (scheda interna) (solo serie CS) | 1 esadecimale | Porta 1 |
| | | 2 esadecimale | Porta 2 |

- b) Impostare l'indirizzo del modulo di destinazione direttamente nei bit da 00 a 07 di C+2. In questo caso, impostare il numero della porta seriale nei bit da 08 a 11 di C+1 su 0 per la specifica diretta.

Porte del modulo di comunicazione seriale

| Porta | Indirizzo del modulo della porta | Esempio: numero di modulo = 1 |
|---------|------------------------------------|--|
| Porta 1 | 80 esadecimale + 4 × numero modulo | 80 + 4 × 1 = 84 esadecimale (132 decimale) |
| Porta 2 | 81 esadecimale + 4 × numero modulo | 81 + 4 × 1 = 85 esadecimale (133 decimale) |

Porte della scheda di comunicazione seriale

| Porta | Indirizzo del modulo della porta |
|---------|----------------------------------|
| Porta 1 | E4 esadecimale (228 decimale) |
| Porta 2 | E5 esadecimale (229 decimale) |

Porte della CPU

| Porta | Indirizzo del modulo della porta |
|------------|----------------------------------|
| Periferica | FD esadecimale (253 decimale) |
| RS-232C | FC esadecimale (252 decimale) |

- Quando si specifica una porta seriale senza una tabella di routing per la funzione di gateway seriale (conversione in FINS Host Link), impostare l'indirizzo del modulo della porta seriale nel byte dell'indirizzo della rete di destinazione.
- L'indirizzo del modulo indica il modulo, come mostrato nella tabella riportata di seguito.

| Modulo | Impostazione dell'indirizzo del modulo |
|--|--|
| CPU | 00 esadecimale |
| Unità bus CPU | 10 esadecimale + numero modulo |
| Modulo di I/O speciale (ad eccezione dei moduli di I/O speciali della serie C200H) | 20 esadecimale + numero modulo |
| Scheda interna (solo serie CS) | E1 esadecimale |
| Computer | 01 esadecimale |
| Modulo collegato alla rete (non necessario specificare il modulo) | FE esadecimale |
| Specifica diretta dell'indirizzo del modulo della porta seriale | Porte del modulo di comunicazione seriale Porta 1: 80 esadecimale + 4 × numero modulo Porta 2: 81 esadecimale + 4 × numero modulo Porte della scheda di comunicazione seriale Porta 1: E4 esadecimale (228 decimale) Porta 2: E5 esadecimale (229 decimale) Porte della CPU Porta periferica: FD esadecimale (253 decimale) Porta RS-232C: FC esadecimale (252 decimale) |

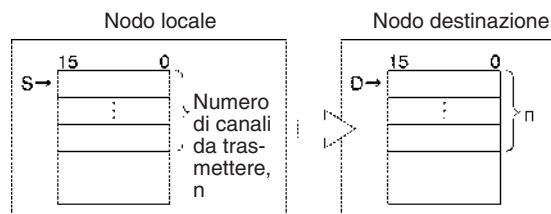
- Il numero massimo di numeri di nodo dipende dalla rete in uso. Per Controller Link l'intervallo ammesso è da 00 a 20 esadecimale (da 0 a 32). Impostare il numero del nodo di destinazione su FF per diffonderlo a tutti i nodi; impostarlo su 00 per trasmetterlo all'interno del nodo locale.
- Per informazioni dettagliate sull'utilizzo dell'allocazione automatica del numero della porta di comunicazione (porta logica), consultare la sezione *Allocazione automatica delle porte di comunicazione* a pagina 979.
- Quando il numero del nodo di destinazione è impostato su FF (trasmissione broadcast), non vi sarà alcuna risposta anche se i bit da 12 a 15 sono impostati su 0.

Caratteristiche operando

| Area | S | D | C |
|--|--|---|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6139 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W507 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H507 |
| Area bit ausiliaria | Da A000 a A959 | | Da A000 a A955 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4091 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4091 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32763 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32763 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32763 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

SEND(090) trasferisce i dati cominciando dal canale S agli indirizzi cominciando in D nel dispositivo designato tramite il bus CPU del PLC o tramite una rete. Il numero di canali da trasmettere è specificato in C.



Se il numero del nodo di destinazione è impostato su FF, i dati verranno diffusi a tutti i nodi della rete designata. Questa operazione è chiamata trasmissione broadcast.

Se è richiesta una risposta (bit da 12 a 15 di C+3 impostati su 0) ma non è stata ricevuta alcuna risposta entro il tempo di monitoraggio della risposta, i dati verranno ritrasmessi fino a 15 volte (tentativi impostati nei bit da 0 a 3 di C+3). Non vi saranno né risposta, né tentativi per le trasmissioni broadcast.

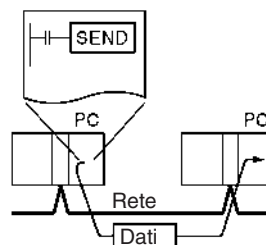
È possibile utilizzare l'istruzione SEND(090) per trasmettere dati a una particolare porta seriale nel dispositivo di destinazione e nel dispositivo stesso.

È possibile trasmettere i dati a un computer host collegato alla porta seriale del PLC (se impostato in modalità Host Link) oltre che a un PLC o a un computer collegato tramite una rete Controller Link o Ethernet.

Se quando viene eseguita l'istruzione SEND(090) il flag di abilitazione porte di comunicazione è ON per la porta di comunicazione specificata in C+3, verranno disattivati il flag di abilitazione porte di comunicazione corrispondente (porte da 00 a 07: da A20200 a A20207) e il flag di errore porte di comunicazione (porte da 00 a 07: da A21900 a A21907) e 0000 verrà scritto nel canale che contiene il codice di completamento (porte da 00 a 07: da A203 a A210). Una volta impostati i flag, i dati verranno trasmessi al nodo di destinazione.

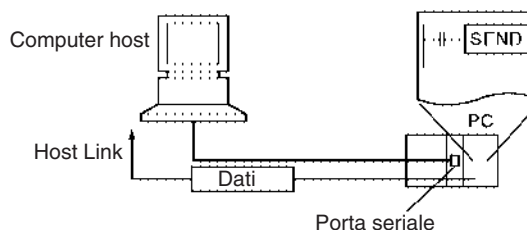
Trasmissione tramite la rete

SEND(090) può essere utilizzata per trasmettere i dati dal PLC all'area dati specificata in un PLC o in un computer collegato tramite una rete Controller Link o un link Ethernet.



Trasmissione tramite Host Link

Quando la porta seriale integrata nella CPU, una scheda di comunicazione seriale (solo serie CS) o un modulo di comunicazione seriale è in modalità Host Link e collegato in modalità uno a uno con un computer host, è possibile eseguire l'istruzione SEND(090) per trasmettere i dati dal PLC al computer host la volta successiva che il PLC possiede i diritti di trasmissione. È possibile inoltre trasmettere ad altri computer host collegati ad altri PLC in rete.



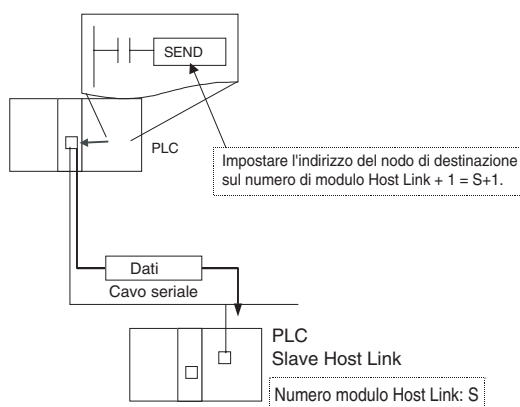
Se l'istruzione SEND(090) viene inviata alla porta seriale della CPU, una scheda di comunicazione seriale (solo serie CS) o un modulo di comunicazione seriale, verrà inviato un comando dalla porta seriale al computer host. Il comando è un messaggio FINS racchiuso tra un'intestazione e un terminatore Host Link. Il comando FINS è un comando MEMORY AREA WRITE (codice di comando 0102) e il codice intestazione Host Link è 0F esadecimale.

È necessario creare un programma nel computer host per elaborare il comando ricevuto (il comando FINS racchiuso tra l'intestazione e il terminatore Host Link).

Se la porta seriale di destinazione è il PLC locale, impostare l'indirizzo di rete su 00 (rete locale) in C+1, impostare l'indirizzo del nodo su 00 (PLC locale) in C+2 e impostare l'indirizzo del modulo su 00 (CPU), E1 (scheda interna (solo serie CS)) o numero di modulo + 10 esadecimale (modulo porta seriale).

Invio di dati a un PLC slave Host Link collegato tramite gateway seriale

È possibile utilizzare la funzione di gateway seriale per inviare i dati a un PLC collegato come slave Host Link a una scheda o a un modulo di comunicazione seriale. In questo caso, l'indirizzo del nodo di destinazione deve essere impostato con il numero del modulo Host Link + 1.



Flag

| Nome | Etichetta | Operazione |
|----------------------------|-----------|---|
| di comunicazione esplicita | ER | ON se il numero di porta seriale specificato in C+1 non è incluso nell'intervallo 00 - 04. ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C+3. OFF in tutti gli altri casi. |

Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

| Nome | Indirizzo | Operazione |
|---|---------------------|---|
| Flag di abilitazione porte di comunicazione | Da A20200 ad A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |
| Flag di errore porte di comunicazione | Da A21900 ad A21907 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di rete. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Codici di completamento per le porte di comunicazione | Da A203 ad A210 | Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete. Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene eseguita un'istruzione. |

Avvertenze

Se il flag di abilitazione porte di comunicazione è OFF per il numero della porta specificata in C+3, l'istruzione verrà considerata come NOP(000) e non verrà eseguita. In questo caso verrà attivato il flag di errore.

Quando per D è stato specificato un indirizzo nel banco corrente dell'area EM, i dati trasmessi verranno scritti nel banco EM corrente del nodo di destinazione.

Quando i dati vengono trasmessi al di fuori della rete locale, l'utente deve registrare le tabelle di routing nei PLC (CPU) di ciascuna rete. (Le tabelle di routing indicano i percorsi ad altre reti in cui sono collegati i nodi di destinazione).

Per ulteriori informazioni sui codici di completamento per la comunicazione di rete, fare riferimento ai codici di risposta del comando FINS nel manuale *CS/CJ Series Communications Commands Reference Manual (W342)*.

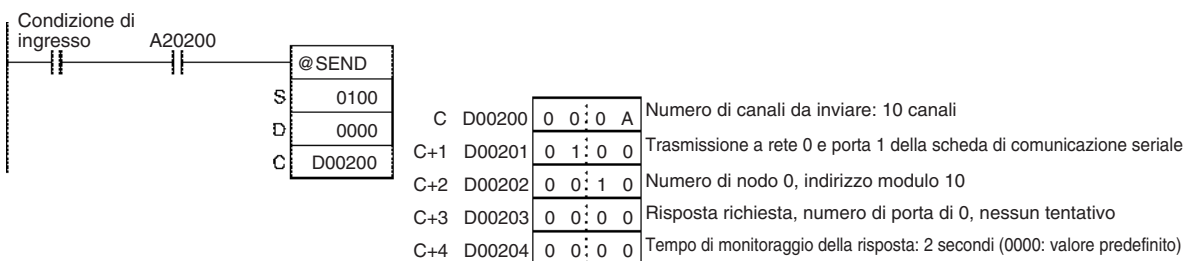
Per ciascuna porta di comunicazione è possibile eseguire solo un'istruzione di rete per volta. Per essere sicuri che l'istruzione SEND(090) non venga eseguita quando una porta è occupata, programmare il flag di abilitazione porte di comunicazione (da A20200 a A20207) come condizione normalmente aperta.

I numeri della porta di comunicazione da 00 a 07 vengono condivisi dalle istruzioni di rete e da PMCR(260), quindi non è possibile eseguire l'istruzione SEND(090) contemporaneamente a PMCR(260) se entrambe le istruzioni stanno usando lo stesso numero di porta.

Il rumore e altri fattori possono deteriorare o provocare la perdita della trasmissione. Pertanto si raccomanda di impostare il numero di tentativi su un valore diverso da zero per provocare la ripetizione dell'esecuzione di SEND(090) se non si riceve risposta entro il tempo di monitoraggio della risposta.

Esempio 1

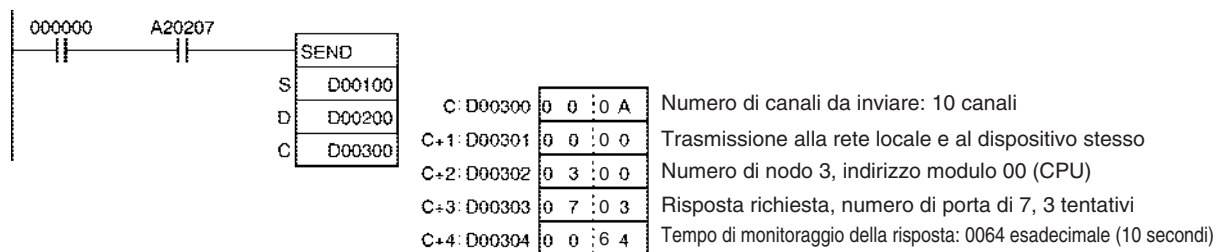
Nell'esempio seguente, quando la condizione di ingresso e A20200 (il flag di abilitazione porte di comunicazione per la porta 0) sono ON, i dieci canali da CIO 100 a CIO 109 vengono trasmessi al computer host collegato alla porta 1 del modulo di comunicazione seriale con indirizzo di modulo 10 (esadecimale) sul numero di nodo 3 nella rete 0.



È necessario creare un programma sul computer host per ricevere i dati e inviare una risposta.

Esempio 2

Nell'esempio seguente, quando CIO 000000 e A20207 (il flag di abilitazione porte di comunicazione per la porta 07) sono ON, i dieci canali da D00100 a D00109 vengono trasmessi al numero di nodo 3 nella rete locale in cui vengono scritti nei dieci canali da D00200 a D00209. Se non si riceve una risposta entro dieci secondi, la trasmissione dei dati verrà ripetuta fino a 3 volte.



3-25-4 NETWORK RECEIVE: RECV(098)

Scopo

Richiede la trasmissione di dati da un nodo nella rete e riceve tali dati.

- a) Impostare l'indirizzo del modulo di origine (bit da 00 a 07 di C+2) sull'indirizzo del modulo della CPU o della scheda/modulo di comunicazione seriale e impostare il numero della porta seriale (bit da 08 a 11 di C+1) su 1 per la porta 1 o 2 per la porta 2.

| Indirizzo modulo (C+2, bit da 00 a 07) | Modulo | Numero di porta seriale (C+1, bit da 08 a 11) | Porta seriale |
|--|--|---|-------------------------|
| 00 esadecimale | CPU | 1 esadecimale | Porta RS-232C integrata |
| | | 2 esadecimale | Porta periferiche |
| 10 esadecimale + numero modulo | Modulo di comunicazione seriale (unità bus CPU) | 1 esadecimale | Porta 1 |
| | | 2 esadecimale | Porta 2 |
| E1 esadecimale | Scheda di comunicazione seriale (scheda interna) (solo serie CS) | 1 esadecimale | Porta 1 |
| | | 2 esadecimale | Porta 2 |

- b) Impostare l'indirizzo del modulo di origine direttamente nei bit da 00 a 07 di C+2. In questo caso, impostare il numero della porta seriale nei bit da 08 a 11 di C+1 su 0 per la specifica diretta.

Porte del modulo di comunicazione seriale

| Porta | Indirizzo del modulo della porta | Esempio: numero di modulo = 1 |
|---------|------------------------------------|--|
| Porta 1 | 80 esadecimale + 4 × numero modulo | 80 + 4 × 1 = 84 esadecimale (132 decimale) |
| Porta 2 | 81 esadecimale + 4 × numero modulo | 81 + 4 × 1 = 85 esadecimale (133 decimale) |

Porte della scheda di comunicazione seriale

| Porta | Indirizzo del modulo della porta |
|---------|----------------------------------|
| Porta 1 | E4 esadecimale (228 decimale) |
| Porta 2 | E5 esadecimale (229 decimale) |

Porte della CPU

| Porta | Indirizzo del modulo della porta |
|------------|----------------------------------|
| Periferica | FD esadecimale (253 decimale) |
| RS-232C | FC esadecimale (252 decimale) |

4. Quando si specifica una porta seriale senza una tabella di routing per la funzione di gateway seriale (conversione in FINS Host Link), impostare l'indirizzo del modulo della porta seriale nel byte dell'indirizzo della rete di origine.
5. L'indirizzo del modulo indica il modulo, come mostrato nella tabella riportata di seguito.

| Modulo | Impostazione dell'indirizzo del modulo |
|--|--|
| CPU | 00 esadecimale |
| Unità bus CPU | 10 esadecimale + numero modulo |
| Modulo di I/O speciale (ad eccezione dei moduli di I/O speciali della serie C200H) | 20 esadecimale + numero modulo |
| Scheda interna (solo serie CS) | E1 esadecimale |

| Modulo | Impostazione dell'indirizzo del modulo |
|---|--|
| Computer | 01 esadecimale |
| Modulo collegato alla rete (non necessario specificare il modulo) | FE esadecimale |
| Specifica diretta dell'indirizzo del modulo della porta seriale | Porte del modulo di comunicazione seriale Porta 1: 80 esadecimale + 4 × numero modulo Porta 2: 81 esadecimale + 4 × numero modulo Porte della scheda di comunicazione seriale Porta 1: E4 esadecimale (228 decimale) Porta 2: E5 esadecimale (229 decimale) Porte della CPU Porta periferica: FD esadecimale (253 decimale) Porta RS-232C: FC esadecimale (252 decimale) |

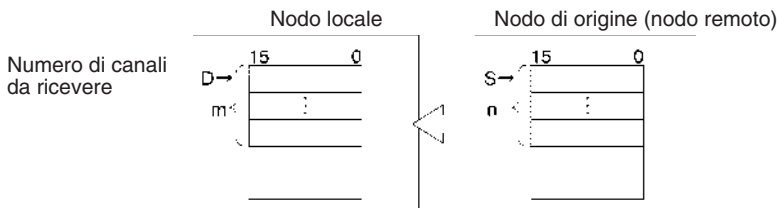
6. Il numero massimo di numeri di nodo dipende dalla rete in uso. Per Controller Link l'intervallo ammesso è da 00 a 20 esadecimale (da 0 a 32). Impostare l'indirizzo del nodo di origine su 00 per trasmettere all'interno del nodo locale.
7. Per informazioni dettagliate sull'utilizzo dell'allocazione automatica del numero della porta di comunicazione (porta logica), consultare la sezione *Allocazione automatica delle porte di comunicazione* a pagina 979.

Caratteristiche operando

| Area | S | D | C |
|--|--|----------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6139 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W507 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H507 |
| Area bit ausiliaria | Da A000 a A447 Da A448 a A959 | Da A448 a A959 | Da A000 a A443 Da A448 a A955 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4091 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4091 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32763 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32763 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32763 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

RECV(098) richiede il numero di canali specificato in C cominciando dal canale S da trasferire dal dispositivo designato al PLC locale. I dati vengono ricevuti tramite l'unità bus CPU del PLC o tramite la rete e scritti nell'area dati del PLC cominciando da D.



Con l'istruzione RECV(098) è necessaria una risposta poiché tale risposta contiene i dati da ricevere. Se non si riceve risposta entro il tempo di monitoraggio risposta impostato in C+4, la richiesta di trasferimento dei dati verrà ritrasmessa fino a 15 volte (tentativi impostati nei bit 0 - 3 di C+3).

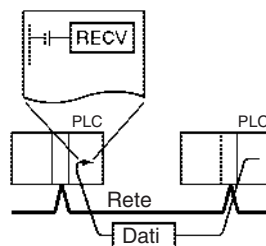
È possibile utilizzare l'istruzione RECV(098) per richiedere una trasmissione di dati da una particolare porta seriale nel dispositivo di origine e nel dispositivo stesso.

È possibile ricevere i dati da un computer host collegato alla porta seriale del PLC (se impostato in modalità Host Link) oltre che da un PLC o da un computer collegato tramite una rete Controller Link o Ethernet.

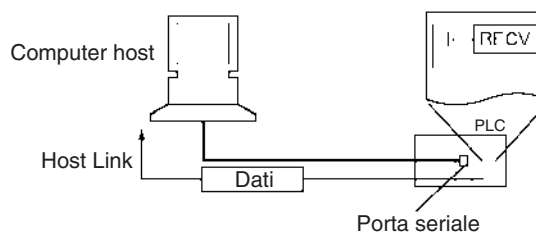
Se quando viene eseguita l'istruzione SEND(090) il flag di abilitazione porte di comunicazione è ON per la porta di comunicazione specificata in C+3, verranno disattivati il flag di abilitazione porte di comunicazione corrispondente (porte da 00 a 07: da A20200 a A20207) e il flag di errore porte di comunicazione (porte da 00 a 07: da A21900 a A21907) e 0000 verrà scritto nel canale che contiene il codice di completamento (porte da 00 a 07: da A203 a A210). Una volta impostati i flag, i dati verranno ricevuti dal nodo di destinazione.

Trasmissione tramite la rete

RECV(098) può essere utilizzata per ricevere i dati trasmessi dall'area dati specificata in un PLC o in un computer collegato tramite una rete Controller Link o un link Ethernet e scrivere tali dati sul PLC locale nell'area dati specificata.

**Trasmissione tramite Host Link**

Quando la porta seriale integrata nella CPU, una scheda di comunicazione seriale (solo serie CS) o un modulo di comunicazione seriale è in modalità Host Link e collegato in modalità uno a uno con un computer host, è possibile eseguire l'istruzione RECV(098) per ricevere i dati dal computer host la volta successiva che il PLC possiede i diritti di trasmissione dei comandi. È possibile inoltre ricevere da altri computer host collegati ad altri PLC in rete.



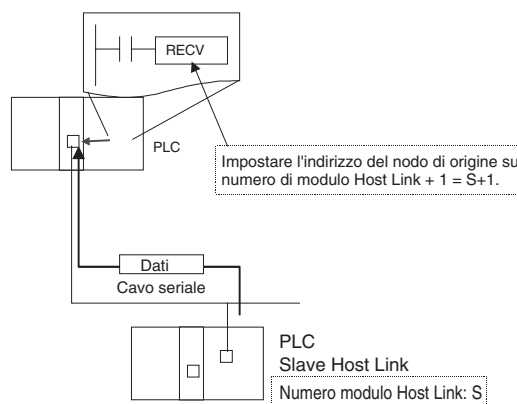
Se l'istruzione RECV(098) viene eseguita per la porta seriale della CPU, di una scheda di comunicazione seriale (solo serie CS) o di un modulo di comunicazione seriale, verrà inviato un comando dalla porta seriale al computer host. Il comando è un messaggio FINS racchiuso tra un'intestazione e un terminatore Host Link. Il comando FINS è un comando MEMORY AREA READ (codice di comando 0101) e il codice intestazione Host Link è 0F esadecimale.

È necessario creare un programma nel computer host per elaborare il comando di invio (il comando FINS racchiuso tra l'intestazione e il terminatore Host Link).

Se la porta seriale di destinazione è il PLC locale, impostare l'indirizzo di rete su 00 (rete locale) in C+1, impostare l'indirizzo del nodo su 00 (PLC locale) in C+2 e impostare l'indirizzo del modulo su 00 (CPU), E1 (scheda interna (solo serie CS)) o numero di modulo + 10 esadecimale (modulo porta seriale).

Ricezione di dati da un PLC slave Host Link collegato tramite gateway seriale

È possibile utilizzare la funzione di gateway seriale per ricevere i dati in un PLC collegato come slave Host Link a una scheda o a un modulo di comunicazione seriale. In questo caso, l'indirizzo del nodo di origine deve essere impostato con il numero del modulo Host Link + 1.



Flag

| Nome | Etichetta | Operazione |
|----------------------------|-----------|---|
| di comunicazione esplicita | ER | ON se il numero di porta seriale specificato in C+1 non è inclusa nell'intervallo 00 - 04. ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C+3. OFF in tutti gli altri casi. |

Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

| Nome | Indirizzo | Operazione |
|---|--------------------|--|
| Flag di abilitazione porte di comunicazione | Da A20200 a A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |

| Nome | Indirizzo | Operazione |
|---|--------------------|---|
| Flag di errore porte di comunicazione | Da A21900 a A21907 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di rete. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Codici di completamento per le porte di comunicazione | Da A203 a A210 | Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete. Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene avviata l'esecuzione del programma. |

Avvertenze

Se il flag di abilitazione porte di comunicazione è OFF per il numero della porta specificata in C+3, l'istruzione verrà considerata come NOP(000) e non verrà eseguita. In questo caso verrà attivato il flag di errore.

Quando per D è stato specificato un indirizzo nel banco corrente dell'area EM, i dati trasmessi verranno scritti nel banco EM corrente del nodo di destinazione.

Quando i dati vengono trasmessi al di fuori della rete locale, l'utente deve registrare le tabelle di routing nei PLC (CPU) di ciascuna rete. (Le tabelle di routing indicano i percorsi ad altre reti in cui sono collegati i nodi di destinazione).

Per ulteriori informazioni sui codici di completamento per la comunicazione di rete, fare riferimento ai codici di risposta del comando FINS nel manuale *CS/CJ Series Communications Commands Reference Manual (W342)*.

Per ciascuna porta di comunicazione è possibile eseguire solo un'istruzione di rete per volta. Per essere sicuri che l'istruzione RECV(098) non venga eseguita quando una porta è occupata, programmare il flag di abilitazione porte di comunicazione (da A20200 a A20207) come condizione normalmente aperta.

I numeri della porta di comunicazione da 00 a 07 vengono condivisi dalle istruzioni di rete e da PMCR(260), quindi non è possibile eseguire l'istruzione RECV(098) contemporaneamente a PMCR(260) se entrambe le istruzioni stanno usando lo stesso numero di porta.

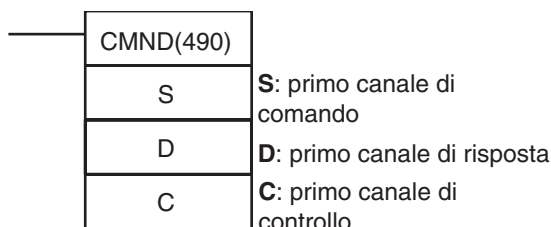
Il rumore e altri fattori possono deteriorare o provocare la perdita della trasmissione. Pertanto si raccomanda di impostare il numero di tentativi su un valore diverso da zero per provocare la ripetizione dell'esecuzione di RECV(098) se non si riceve risposta entro il tempo di monitoraggio della risposta.

3-25-5 DELIVER COMMAND: CMND(490)

Scopo

Invia un comando FINS e riceve la risposta. Per informazioni sui comandi FINS, fare riferimento al manuale *CS/CJ-Series Communications Commands Reference Manual*.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CMND(490) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CMND(490) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: primo canale di controllo

I sei canali di controllo da C a C+5 specificano il numero di byte dei dati di comando e dei dati di risposta, la destinazione e altre impostazioni riportate nella tabella seguente.

| Canale | Bit da 00 a 07 | Bit da 08 a 15 |
|--------|---|---|
| C | Byte dei dati di comando: da 0002 al massimo consentito ¹ (esadecimali a 4 cifre) | |
| C+1 | Byte dei dati di risposta: da 0000 al massimo consentito ¹⁻³ (esadecimali a 4 cifre) | |
| C+2 | Indirizzo rete destinazione: da 00 a 07 ^{4,6} | Bit da 08 a 11: numero di porta seriale (porta fisica) 1 esadecimale: Porta 1 2 esadecimale: Porta 2 (non impostare 0, 3 o 4). Bit da 12 a 15: sempre 0. |
| C+3 | Indirizzo modulo di destinazione: da 00 a FE ^{5,7,9} | Numero nodo di destinazione: da 00 al massimo consentito ⁸ |
| C+4 | N. di tentativi: da 00 a 0F (da 0 a 15) | Bit da 08 a 11: numero della porta (porta logica interna): da 0 a 7 (F: allocazione automatica) ¹⁰ Bit da 12 a 15: impostazione risposta 0: risposta richiesta. 8: nessuna impostazione richiesta. ¹¹ |
| C+5 | Tempo di monitoraggio della risposta: da 0001 a FFFF (da 0,1 a 6553,5 secondi) (L'impostazione predefinita 0000 imposta un tempo di monitoraggio di 2 secondi). | |

Nota

1. Il numero di byte dei dati di comando in C è da 0002 fino alla lunghezza massima dei dati in esadecimale. Ad esempio, per i sistemi Controller Link il numero di byte sarebbe da 0002 a 07C6 esadecimale (da 2 a 1.990 byte). Il numero di byte per la CPU locale è 07C6 esadecimale (1.990 byte). Il numero di byte dei dati di comando dipende dalla rete.
2. Il numero di byte dei dati di risposta in C+1 è da 0000 fino alla lunghezza massima dei dati in esadecimale. Ad esempio, per i sistemi Controller Link il numero di byte sarebbe da 0000 a 07C6 esadecimale (da 0 a 1.990 byte). Il numero di byte per la CPU locale è 07C6 esadecimale (1.990 byte). Il numero di byte dei dati di risposta dipende dalla rete.
3. Per ulteriori informazioni sulla lunghezza massima dei dati per i dati di comando e i dati di risposta, fare riferimento al manuale d'uso della rete specifica. Per tutti i comandi FINS inviati attraverso più reti, la lunghezza massima dei dati per i dati di comando e i dati di risposta viene determinata dalla rete con la lunghezza massima dei dati minore possibile.
4. Impostare l'indirizzo della rete di destinazione su 00 per trasmettere all'interno della rete locale. Quando sono montate due o più unità bus CPU, l'indirizzo di rete sarà il numero di modulo del modulo avente il numero di modulo più basso.

5. Per inviare un comando FINS a un computer host tramite una porta seriale con Host Link, inizializzando la comunicazione dal PLC, o tramite la funzione di gateway seriale (convertito in CompoWay/F, Modbus-RTU o Modbus-ASCII), è possibile utilizzare i due metodi esposti di seguito.
- a) Impostare l'indirizzo del modulo di destinazione (bit da 00 a 07 di C+3) sull'indirizzo del modulo della CPU o della scheda/modulo di comunicazione seriale e impostare il numero della porta seriale (bit da 08 a 11 di C+2) su 1 per la porta 1 o 2 per la porta 2.

| Indirizzo modulo (C+3, bit da 00 a 07) | Modulo | Numero di porta seriale (C+2, bit da 08 a 11) | Porta seriale |
|--|--|---|-------------------------|
| 00 esadecimale | CPU | 1 esadecimale | Porta RS-232C integrata |
| | | 2 esadecimale | Porta periferiche |
| 10 esadecimale + numero modulo | Modulo di comunicazione seriale (unità bus CPU) | 1 esadecimale | Porta 1 |
| | | 2 esadecimale | Porta 2 |
| E1 esadecimale | Scheda di comunicazione seriale (scheda interna) (solo serie CS) | 1 esadecimale | Porta 1 |
| | | 2 esadecimale | Porta 2 |

- b) Impostare l'indirizzo del modulo di destinazione direttamente nei bit da 00 a 07 di C+3. In questo caso, impostare il numero della porta seriale nei bit da 08 a 11 di C+2 su 0 per la specifica diretta.

Porte del modulo di comunicazione seriale

| Porta | Indirizzo del modulo della porta | Esempio: numero di modulo = 1 |
|---------|------------------------------------|---|
| Porta 1 | 80 esadecimale + 4 × numero modulo | $80 + 4 \times 1 = 84$ esadecimale (132 decimale) |
| Porta 2 | 81 esadecimale + 4 × numero modulo | $81 + 4 \times 1 = 85$ esadecimale (133 decimale) |

Porte della scheda di comunicazione seriale

| Porta | Indirizzo del modulo della porta |
|---------|----------------------------------|
| Porta 1 | E4 esadecimale (228 decimale) |
| Porta 2 | E5 esadecimale (229 decimale) |

Porte della CPU

| Porta | Indirizzo del modulo della porta |
|------------|----------------------------------|
| Periferica | FD esadecimale (253 decimale) |
| RS-232C | FC esadecimale (252 decimale) |

6. Quando si specifica una porta seriale senza una tabella di routing per la funzione di gateway seriale (conversione in FINS Host Link), impostare l'indirizzo del modulo della porta seriale nel byte dell'indirizzo della rete di destinazione.
7. L'indirizzo del modulo indica il modulo, come mostrato nella tabella riportata di seguito.

| Modulo | Impostazione dell'indirizzo del modulo |
|--|--|
| CPU | 00 esadecimale |
| Unità bus CPU | 10 esadecimale + numero modulo |
| Modulo di I/O speciale (ad eccezione dei moduli di I/O speciali della serie C200H) | 20 esadecimale + numero modulo |

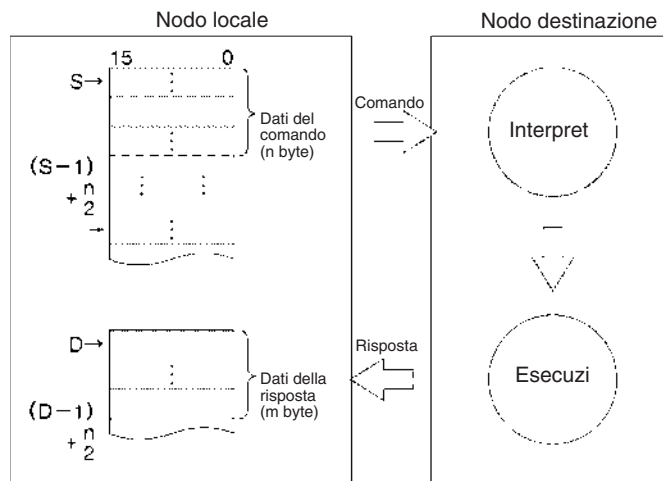
| Modulo | Impostazione dell'indirizzo del modulo |
|---|--|
| Scheda interna (solo serie CS) | E1 esadecimale |
| Computer | 01 esadecimale |
| Modulo collegato alla rete (non necessario specificare il modulo) | FE esadecimale |
| Specifica diretta dell'indirizzo del modulo della porta seriale | Porte del modulo di comunicazione seriale Porta 1: 80 esadecimale + 4 × numero modulo Porta 2: 81 esadecimale + 4 × numero modulo Porte della scheda di comunicazione seriale Porta 1: E4 esadecimale (228 decimale) Porta 2: E5 esadecimale (229 decimale) Porte della CPU Porta periferica: FD esadecimale (253 decimale) Porta RS-232C: FC esadecimale (252 decimale) |

8. Il numero massimo di numeri di nodo dipende dalla rete in uso. Per Controller Link l'intervallo ammesso è da 00 a 20 esadecimale (da 0 a 32). Impostare il numero del nodo di destinazione su FF per diffonderlo a tutti i nodi; impostarlo su 00 per trasmetterlo all'interno del nodo locale.
9. Quando si specifica la porta seriale nella funzione di gateway seriale (conversione in FINS Host Link), impostare l'indirizzo del modulo di destinazione con il numero di modulo Host Link del PLC di destinazione PLC + 1 (intervallo di impostazione: da 1 a 32).
10. Per informazioni dettagliate sull'utilizzo dell'allocazione automatica del numero della porta di comunicazione (porta logica), consultare la sezione *Allocazione automatica delle porte di comunicazione* a pagina 979.
11. Quando il numero del nodo di destinazione è impostato su FF (trasmissione broadcast), non vi sarà alcuna risposta anche se i bit da 12 a 15 sono impostati su 0.

| Area | S | C | D |
|--|--|----------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6138 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W506 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H506 |
| Area bit ausiliaria | Da A000 a A447 Da A448 a A959 | Da A448 a A959 | Da A000 a A442 Da A448 a A954 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4090 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4090 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32762 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32762 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32763 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

L'istruzione CMND(490) trasferisce il numero specificato di byte dei dati del comando FINS cominciando nel canale S al dispositivo designato tramite l'unità bus CPU del PLC o tramite una rete. La risposta viene memorizzata in memoria cominciando nel canale D.



È possibile utilizzare l'istruzione CMND(490) per trasmettere dati di comando a una particolare porta seriale nel dispositivo di destinazione e nel dispositivo stesso. CMND(490) funziona esattamente come SEND(090) se il codice del comando FINS è 0102 (MEMORY AREA WRITE) e come RECV(098) se il codice è 0101 (MEMORY AREA READ).

La CPU che esegue CMND(490) può inviare un comando FINS a se stessa, ad eccezione delle CPU CS1 della serie CS precedenti la versione V1□). Per ottenere ciò, utilizzare le seguenti impostazioni dei dati di controllo.

- Indirizzo della rete di destinazione (bit da 00 a 07 di C+2): 00 esadecimale (rete locale)
- N. di porta seriale (bit da 08 a 11 di C+2): 0 esadecimale (non utilizzato)
- Indirizzo del modulo di destinazione (bit da 00 a 07 di C+3): 00 esadecimale (CPU)
- Indirizzo del nodo di destinazione (bit da 08 a 15 di C+3): 00 esadecimale (nodo locale)
- Numero di tentativi (bit da 00 a 03 di C+4): 0 esadecimale (questa impostazione non è valida; impostarla su 0)
- Tempo di monitoraggio della risposta: (bit da 00 a 15 di C+5): da 0000 a FFFF esadecimale (ma 0000 specificherà 6553,5 s e non 2 s come di solito)

Se il numero del nodo di destinazione è impostato su FF, i dati di comando verranno diffusi a tutti i nodi della rete designata. Questa operazione è chiamata trasmissione broadcast.

Se è richiesta una risposta (bit da 12 a 15 di C+4 impostati su 0) ma non è stata ricevuta alcuna risposta entro il tempo di monitoraggio della risposta, i dati di comando verranno ritrasmessi fino a 15 volte (tentativi impostati nei bit da 0 a 3 di C+3). Non vi saranno né risposta, né tentativi per le trasmissioni broadcast. Per le istruzioni che non richiedono alcuna risposta, impostare l'impostazione della risposta su " non richiesta".

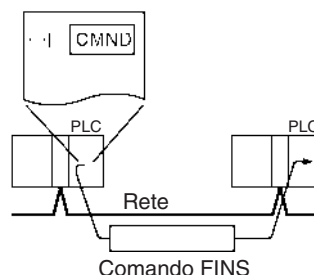
Verrà generato un errore se la quantità di dati della risposta supera il numero di byte specificato per i dati di risposta impostato in C+1.

È possibile trasmettere i dati del comando FINS a un computer host collegato alla porta seriale del PLC (se impostato in modalità Host Link) oltre che a un PLC (CPU, scheda interna (solo serie CS) o unità bus CPU) o a un computer collegato tramite una rete Controller Link o Ethernet.

Se quando viene eseguita l'istruzione CMND(490) il flag di abilitazione porte di comunicazione è ON per la porta di comunicazione specificata in C+3, verranno disattivati il flag di abilitazione porte di comunicazione corrispondente (porte da 00 a 07: da A20200 a A20207) e il flag di errore porte di comunicazione (porte da 00 a 07: da A21900 a A21907) e 0000 verrà scritto nel canale che contiene il codice di completamento (porte da 00 a 07: da A203 a A210). Una volta impostati i flag, i dati del comando verranno trasmessi al nodo di destinazione.

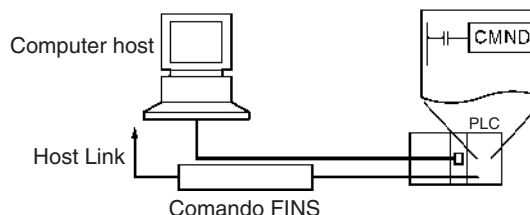
Trasmissione tramite la rete

È possibile utilizzare l'istruzione CMND(490) per trasmettere qualsiasi comando FINS a un personal computer o a un PLC (CPU, scheda interna (solo serie CS) o unità bus CPU) collegato tramite una rete Controller Link o un link Ethernet.



Trasmissione tramite Host Link

Quando la porta seriale integrata nella CPU, una scheda di comunicazione seriale (solo serie CS) o un modulo di comunicazione seriale è in modalità Host Link e collegato in modalità uno a uno con un computer host, è possibile eseguire l'istruzione CMND(490) per trasmettere qualsiasi comando FINS dal PLC al computer host la volta successiva che il PLC possiede i diritti di trasmissione. È possibile inoltre trasmettere ad altri computer host collegati ad altri PLC in rete.



L'istruzione CMND(490) può essere eseguita sia per la porta seriale della CPU, di una scheda di comunicazione seriale (solo serie CS) o di un modulo di comunicazione seriale per inviare un comando al computer host collegato. (Specificare la porta seriale come 1 esadecimale o 2 esadecimale nei bit da 08 a 11 di C+2.) Il comando è un messaggio FINS racchiuso tra un'intestazione e un terminatore Host Link. È possibile inviare qualsiasi comando FINS; il codice dell'intestazione Host Link è 0F esadecimale.

È necessario creare un programma nel computer host per elaborare il comando ricevuto (il comando FINS racchiuso tra l'intestazione e il terminatore Host Link).

Se la porta seriale di destinazione è il PLC locale, impostare l'indirizzo di rete su 00 (rete locale) in C+2, impostare l'indirizzo del nodo su 00 (PLC locale) in C+3 e impostare l'indirizzo del modulo su 00 (CPU), E1 (scheda interna (solo serie CS)) o numero di modulo + 10 esadecimale (modulo porta seriale).

Comunicazione tramite gateway seriale con un componente o uno slave Host Link

È possibile inviare comandi FINS (o inviare e ricevere dati) a un componente o a uno slave Host Link collegato al PLC tramite la rispettiva porta seriale con la funzione di gateway seriale.

- Invio a un componente
(conversione in CompoWay/F, Modbus-RTU o Modbus-ASCII)

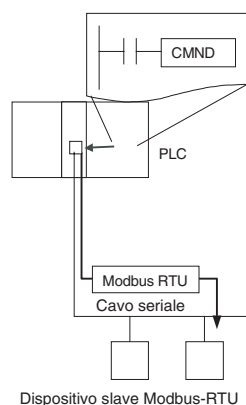
La funzione di gateway seriale può convertire i comandi FINS riportati di seguito in comandi CompoWay/F, Modbus-RTU o Modbus-ASCII quando il comando FINS viene inviato a una scheda di comunicazione seriale, alla porta seriale di un modulo o a una delle porte seriali della CPU (periferica o RS-232C).

Conversione in comando CompoWay/F: 2803 esadecimale

Conversione in comando Modbus-RTU: 2804 esadecimale (vedere nota)

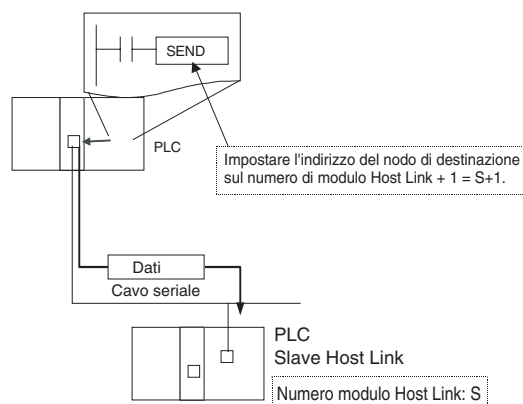
Conversione in comando Modbus-ASCII: 2805 esadecimale (vedere nota)

Nota Non è possibile inviare i comandi Modbus-RTU e Modbus-ASCII alle porte seriali della CPU.



- Invio a un PLC operante come slave Host Link

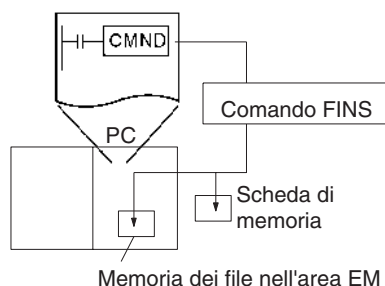
È possibile utilizzare la funzione di gateway seriale per inviare qualsiasi comando FINS a un PLC collegato come slave Host Link e tramite una scheda di comunicazione seriale o una porta seriale del modulo. In questo caso, l'indirizzo del nodo di destinazione deve essere impostato con il numero del modulo Host Link + 1.



Invio di un comando FINS alla CPU che esegue CMND(490) (ad eccezione delle CPU CS1 della serie CS precedenti la versione V1)

La CPU che esegue CMND(490) può inviare un comando FINS a se stessa, ad eccezione delle CPU CS1 della serie CS prive del suffisso -V□). Ad esempio, i comandi per la memoria dei file (codici di comando 22□□ esadecimale) possono essere inviati per formattare la memoria dei file, eliminare e copiare i file ed eseguire altre operazioni. Per ulteriori informazioni, consultare la sezione 5-2 *Manipulating Files* del manuale *CS/CJ-series CPU Unit Programming Manual*.

Verrà attivato il flag di operazione memoria file (A34313) quando un qualsiasi comando FINS viene inviato alla CPU locale (perfino comandi FINS non correlati alla memoria dei file). Utilizzare sempre A34313 in una condizione di ingresso NC per l'istruzione CMND(490) per essere certi che per la CPU venga eseguito solo un comando FINS per volta.



Flag

| Nome | Etichetta | Operazione |
|----------------------------|-----------|---|
| di comunicazione esplicita | ER | ON se il numero di porta seriale specificato in C+2 non è inclusa nell'intervallo 00 - 04. ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C+4. ON se il comando FINS viene inviato alla CPU locale mentre il flag di operazione memoria file (A34313) è ON. OFF in tutti gli altri casi. |

Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

| Nome | Indirizzo | Operazione |
|---|--------------------|---|
| Flag di abilitazione porte di comunicazione | Da A20200 a A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |
| Flag di errore porte di comunicazione | Da A21900 a A21907 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di rete. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Codici di completamento per le porte di comunicazione | Da A203 a A210 | Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete. Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene avviata l'esecuzione del programma. |
| Flag di operazione memoria file | A34313 | ON quando un comando FINS viene inviato alla CPU locale (perfino comandi FINS non correlati alla memoria del file) oppure quando per la memoria dei file viene eseguita una delle istruzioni o operazioni indicate di seguito. FREAD(700) o FWRT(701) Sovrascrittura del programma con bit di controllo in memoria Operazione di backup semplice |

Avvertenze

Se il flag di abilitazione porte di comunicazione è OFF per il numero della porta specificata in C+4, l'istruzione verrà considerata come NOP(000) e non verrà eseguita. In questo caso verrà attivato il flag di errore.

Quando i dati vengono trasmessi al di fuori della rete locale, l'utente deve registrare le tabelle di routing nei PLC (CPU) di ciascuna rete. (Le tabelle di routing indicano i percorsi ad altre reti in cui sono collegati i nodi di destinazione).

Per ulteriori informazioni sui codici di completamento per la comunicazione di rete, fare riferimento ai codici di risposta del comando FINS nel manuale *CS/CJ Series Communications Commands Reference Manual (W342)*.

I numeri della porta di comunicazione da 00 a 07 sono condivisi dalla rete e dalle istruzioni di comunicazione seriale (SEND(090), RECV(098), CMND(490), PMCR(260), TXDU(256) o RXDU(255)). Pertanto, per una porta di comunicazione è possibile eseguire solo una di queste istruzioni per volta.

Per essere sicuri che l'istruzione CMND(490) non venga eseguita quando una porta è occupata, programmare il flag di abilitazione porte di comunicazione (da A20200 a A20207) come condizione normalmente aperta.

Quando si invia un comando FINS alla CPU locale, utilizzare sempre per l'istruzione CMND(490) uno dei flag di abilitazione porte di comunicazione (da A20200 a A20207) in una condizione di ingresso NO e il flag di operazione memoria file (A34313) in una condizione di ingresso NC.

Il rumore e altri fattori possono deteriorare o provocare la perdita della trasmissione. Pertanto si raccomanda di impostare il numero di tentativi su un valore diverso da zero per provocare la ripetizione dell'esecuzione di CMND(490) se non si riceve risposta entro il tempo di monitoraggio della risposta.

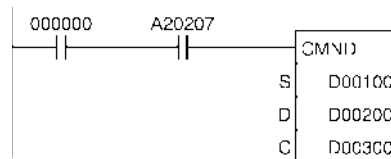
Esempi

Nella sezione di programma riportata di seguito viene illustrato un esempio di invio di un comando FINS a un'altra CPU.

Quando CIO 000000 e A20207 (il flag di abilitazione porte di comunicazione per la porta 07) sono ON, CMND(490) trasmette il comando FINS 0101 (MEMORY AREA READ) al numero di nodo 3. La risposta verrà memorizzata in D00200 - D00211.

Il comando MEMORY AREA READ legge 10 canali da D00010 a D00019. La risposta contiene il codice di comando a 2 byte (0101), il codice di completamento a 2 byte e 10 canali di dati, per un totale di 12 canali o 24 byte.

Se non si riceve una risposta entro dieci secondi, la trasmissione dei dati verrà ripetuta fino a 3 volte.



| | 15 | 8 | 7 | 0 | |
|-------------|----|---|---|---|---|
| S: D00100 | 0 | 1 | 0 | 1 | } D00010 (Area dati = 82 esadecimale, indirizzo = 000A00) Numero di canali da leggere = 0A esadecimale (10 decimale) |
| S+1: D00101 | 8 | 2 | 0 | 0 | |
| S+2: D00102 | 0 | A | 0 | 0 | |
| S+3: D00103 | 0 | 0 | 0 | A | |

| | 15 | 8 | 7 | 0 | |
|-------------|----|---|---|---|---|
| C: D00300 | 0 | 0 | 0 | 8 | Byte dei dati di comando: 0008 (8 decimale) |
| C+1: D00301 | 0 | 0 | 1 | 8 | Byte dei dati di risposta: 0018 (24) |
| C+2: D00302 | 0 | 0 | 0 | 0 | Trasmissione alla rete locale e al dispositivo stesso |
| C+3: D00303 | 0 | 3 | 0 | 0 | Numero di nodo 3, indirizzo modulo 00 (CPU) |
| C+4: D00304 | 0 | 7 | 0 | 3 | Risposta richiesta, numero di porta di 7, 3 tentativi |
| C+5: D00305 | 0 | 0 | 6 | 4 | Tempo di monitoraggio della risposta: 0064 esadecimale (10 secondi) |

Nella sezione di programma riportata di seguito viene illustrato un esempio di invio di un comando FINS a una CPU locale.

Quando CIO 000000 e A20207 (il flag di abilitazione porte di comunicazione per la porta 07) sono ON e A34313 (flag di operazione memoria file) è OFF, CMND(490) trasmette il comando FINS 2215 (CREATE/DELETE DIRECTORY) alla CPU locale. La risposta verrà memorizzata in D00100 - D00101. Qui il comando FINS crea una directory chiamata CS/CJ sotto la directory OMRON. Verranno restituiti e memorizzati come risposta il codice di comando a 2 byte e il codice di fine a 2 byte.



| | 15 | 8 | 7 | 0 | | |
|-------|--------|---|---|---|---|--|
| S: | D00006 | 2 | 2 | 1 | 5 | Codice comando: 2215 esadecimale (CREATE/DELETE DIRECTORY) |
| S+1: | D00007 | 8 | 0 | 0 | 0 | N. disco: 8000 esadecimale (scheda di memoria) |
| S+2: | D00008 | 0 | 0 | 0 | 0 | Parametro: 0000 esadecimale (creazione directory) |
| S+3: | D00009 | 4 | 3 | 5 | 3 | Nome sottodirectory: CS1□□□□□. □□□ (□= spazio) |
| S+4: | D00010 | 3 | 1 | 2 | 0 | |
| S+5: | D00011 | 2 | 0 | 2 | 0 | |
| S+6: | D00012 | 2 | 0 | 2 | 0 | |
| S+7: | D00013 | 2 | E | 2 | 0 | |
| S+8: | D00014 | 2 | 0 | 2 | 0 | |
| S+9: | D00015 | 0 | 0 | 0 | 6 | Lunghezza nome directory: 0006 (6 caratteri) |
| S+10: | D00016 | 5 | C | 4 | F | Percorso assoluto directory: \OMRON |
| S+11: | D00017 | 4 | D | 5 | 2 | |
| S+12: | D00018 | 4 | F | 4 | E | |

| | 15 | 8 | 7 | 0 | | |
|------|--------|---|---|---|---|---|
| S: | D00000 | 0 | 0 | 1 | A | Byte dei dati di comando: 001A (26 decimale) |
| S+1: | D00001 | 0 | 0 | 0 | 4 | Byte dei dati di risposta: 0004 (4) |
| S+2: | D00002 | 0 | 0 | 0 | 0 | Indirizzo rete destinazione: 00 esadecimale (rete locale) |
| S+3: | D00003 | 0 | 0 | 0 | 0 | Indirizzo modulo di destinazione: 00 esadecimale, Numero nodo di destinazione: 00 esadecimale (CPU sul nodo locale) |
| S+4: | D00004 | 0 | 7 | 0 | 0 | Risposta richiesta, numero di porta di 7, 0 tentativi |
| S+5: | D00005 | 0 | 0 | 0 | 0 | Tempo di monitoraggio della risposta: 0000 esadecimale (6553,5 secondi) |

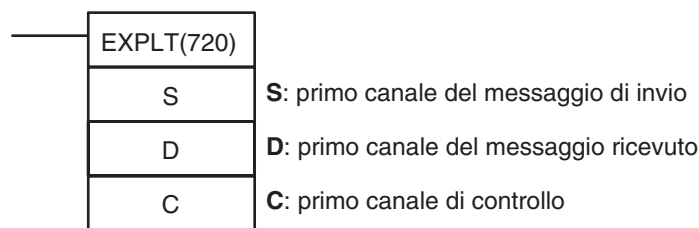
3-25-6 EXPLICIT MESSAGE SEND: EXPLT(720)

Scopo

Invia un messaggio esplicito con qualsiasi codice di servizio.

Questa istruzione è supportata solo dalla CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | EXPLT(720) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @EXPLT(720) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

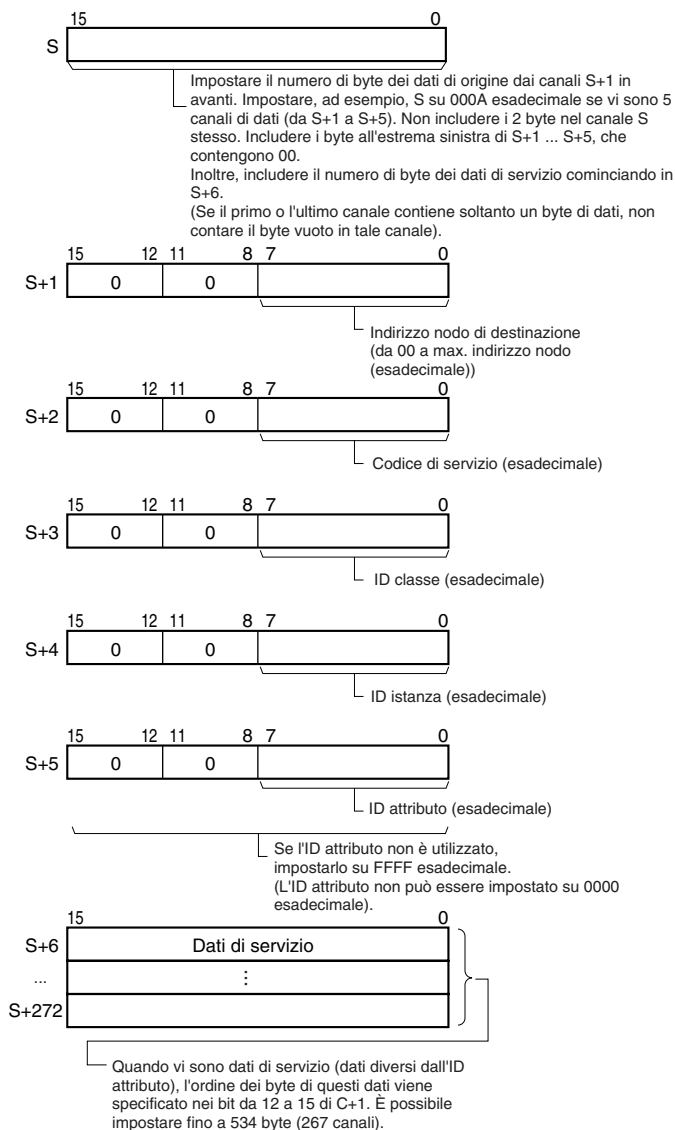
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

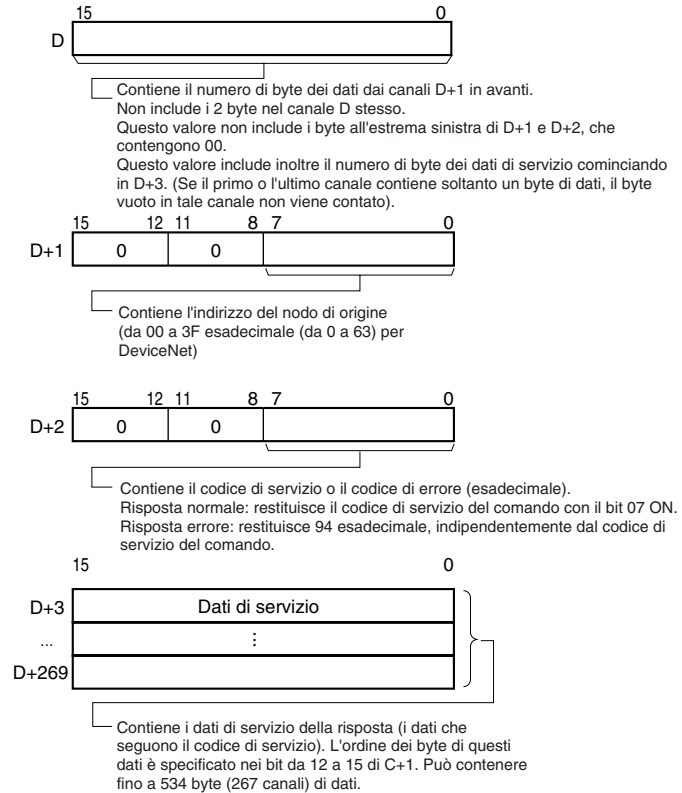
S: primo canale del messaggio di invio

Specifica il primo canale del messaggio di invio (da S a S+272 max.).



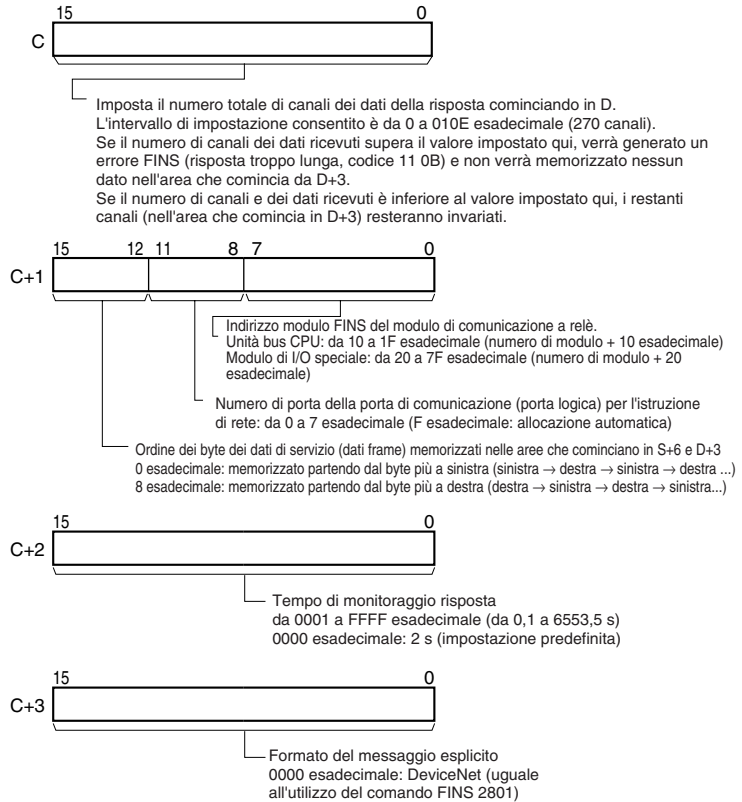
D: primo canale del messaggio ricevuto

Specifica il primo canale del messaggio ricevuto (da D a D+269 max.).



C: primo canale di controllo

Specifica il primo di quattro canali di controllo (da C a C+3).



Caratteristiche operando

| Area | S | D | C |
|--|---|----------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H508 |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A959 | Da A000 a A956 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4092 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Invia un comando di messaggio esplicito (memorizzato nell'intervallo di canali che comincia da S+2) all'indirizzo del nodo specificato in S+1, tramite il modulo di comunicazione con l'indirizzo di modulo FINS specificato nei bit da 00 a 07 di C+1. Quando si riceve la risposta al messaggio esplicito, essa viene memorizzata nell'intervallo di canali che comincia in D+2.

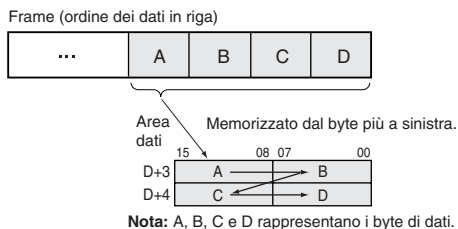
Impostazioni del numero di byte

Il numero di byte dei dati da inviare in S include i 10 byte in S+1 - S+5 e il numero di byte dei dati di servizio cominciando da S+6. Ad esempio, in presenza di 1 byte di dati di servizio, vi sono 11 byte di dati complessivi, quindi è necessario impostare S su 000B esadecimale.

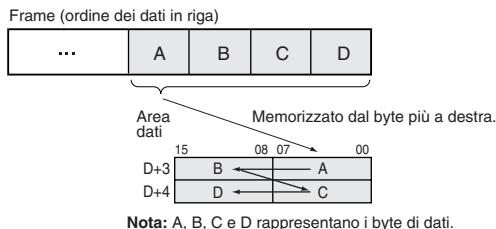
Il numero di byte dei dati ricevuti in D include i 4 byte in D+1 e D+2 e il numero di byte dei dati di servizio cominciando da D+3. Ad esempio, in presenza di 1 byte di dati di servizio, vi sono 5 byte di dati complessivi e D contiene 0005 esadecimale.

L'impostazione nei bit da 12 a 15 di C+1 (0 o 8 esadecimale) determina l'ordine dei byte dei dati di servizio memorizzati in S+6 e D+3.

- Memorizzazione dei dati dal byte più a sinistra
Impostare i bit da 12 a 15 di C+1 su 0 esadecimale.



- Memorizzazione dei dati dal byte più a destra
Impostare i bit da 12 a 15 di C+1 su 8 esadecimale.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C. OFF in tutti gli altri casi. |

Il flag di errore di comunicazione esplicita corrispondente sarà OFF se l'istruzione termina normalmente o ON se si verifica un errore.

Se subentra un errore (flag corrispondente in A213 ON), verrà utilizzato il rispettivo flag di errore porte di comunicazione per determinare se il messaggio esplicito non è stato inviato (flag corrispondente in A219 ON) o se il messaggio è stato inviato ma conteneva un errore (flag corrispondente in A219 OFF).

Il codice di completamento per le porte di comunicazione corrispondente (A203 - A210) sarà 0000 esadecimale se l'istruzione è terminata normalmente, un codice di errore del messaggio esplicito se si è verificato un errore nel messaggio esplicito o un codice di errore FINS se è subentrato un errore FINS.

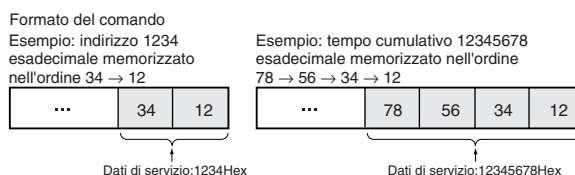
Per ulteriori informazioni sul funzionamento generale delle istruzioni per messaggi espliciti, consultare 3-25-2 *Informazioni sulle istruzioni per messaggi espliciti*.

Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

| Nome | Indirizzo | Operazione |
|---|---------------------|---|
| Flag di abilitazione porte di comunicazione | Da A20200 a A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |
| Flag di errore di comunicazione esplicita | Da A21300 ad A21307 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di una comunicazione di messaggio esplicito. I flag saranno ON se il messaggio esplicito non è stato inviato o se è stato inviato ma è stata restituita una risposta di errore. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di messaggio esplicito successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Flag di errore porte di comunicazione | Da A21900 a A21907 | Questi flag sono ON ad indicare che il messaggio esplicito non è stato inviato dalle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di messaggio esplicito. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Codici di completamento per le porte di comunicazione | Da A203 a A210 | Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete. Il canale corrispondente conterrà 0000 mentre il flag di errore di comunicazione esplicita è OFF. Il canale corrispondente conterrà un codice di errore FINS quando il flag di errore di comunicazione esplicita e il flag di errore porte di comunicazione di tale porta sono entrambi ON. Il canale corrispondente conterrà il codice di errore appropriato relativo al messaggio esplicito quando il flag di errore di comunicazione esplicita di tale porta è ON e il flag di errore porte di comunicazione di tale porta è OFF. Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene avviata l'esecuzione del programma. |

Avvertenze

Assicurarsi che l'ordine dei byte nei dati di origine corrisponda all'ordine nel frame del messaggio esplicito (ordine dei dati nella riga). Ad esempio, quando i dati di servizio sono in unità di 2 o 4 byte, l'ordine dei dati nel frame sarà dal più a sinistra al più a destra in coppie a 2 cifre, come mostrato nello schema seguente.

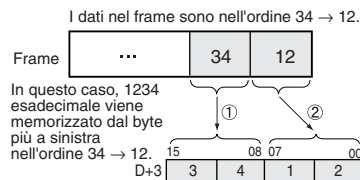


Nello schema seguente viene mostrato come i dati vengono memorizzati nelle aree dati quando i dati di servizio sono in unità da 2 o 4 byte.

1. Dati in unità da 2 byte

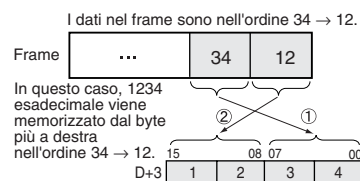
- Memorizzazione dei dati dal byte più a sinistra (bit da 12 a 15 di C = 0 esadecimale)

Esempio: memorizzazione del valore 1234 esadecimale in D+3



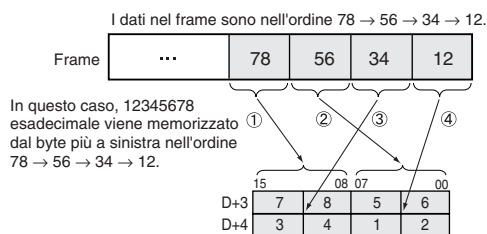
- Memorizzazione dei dati dal byte più a destra (bit da 12 a 15 di C = 8 esadecimale)

Esempio: memorizzazione del valore 1234 esadecimale in D+3

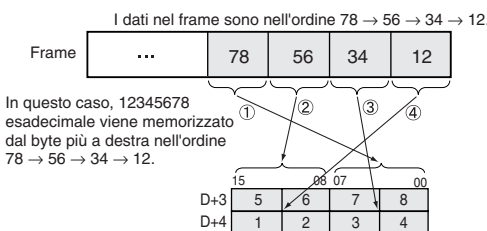


2. Dati in unità da 4 byte

- Memorizzazione dei dati dal byte più a sinistra (bit da 12 a 15 di C = 0 esadecimale) Esempio: memorizzazione del valore 12345678 esadecimale in D+3 e D+4



- Memorizzazione dei dati dal byte più a destra (bit da 12 a 15 di C = 8 esadecimale) Esempio: memorizzazione del valore 12345678 esadecimale in D+3 e D+4



Nota Nell'esempio precedente viene mostrato come i dati ricevuti vengono memorizzati in D+3, ma allo stesso modo vengono memorizzati i dati di invio in S+6.

Esempio

In questo esempio si utilizza EXPLT(720) per leggere il tempo di attivazione totale o il numero di operazione di contatto da uno slave DRT2 (terminale di I/O).

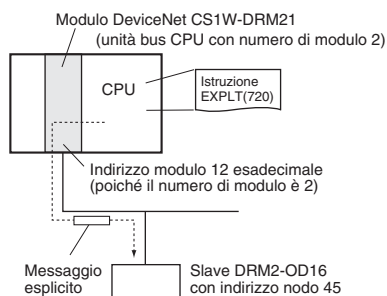
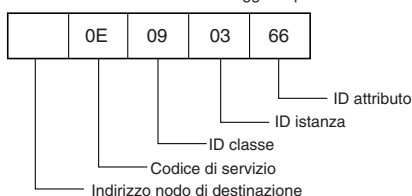


Quando CIO 000000 e A20206 (il flag di abilitazione porte di comunicazione per la porta 06) sono ON, l'istruzione EXPLT(720) legge il tempo di attivazione totale (s) o il numero di operazioni di contatto da uno slave DRT2 (terminale di I/O). In questo caso viene letto il tempo di attivazione totale o il numero di operazioni di contatto per l'ingresso 3.

Codice di servizio = 0E esadecimale, ID classe = 09 esadecimale, ID istanza = 03 esadecimale e ID attributo = 66 esadecimale.

Viene restituito, ad esempio, un valore di 2.752.039 s come risposta sul tempo di attivazione totale.

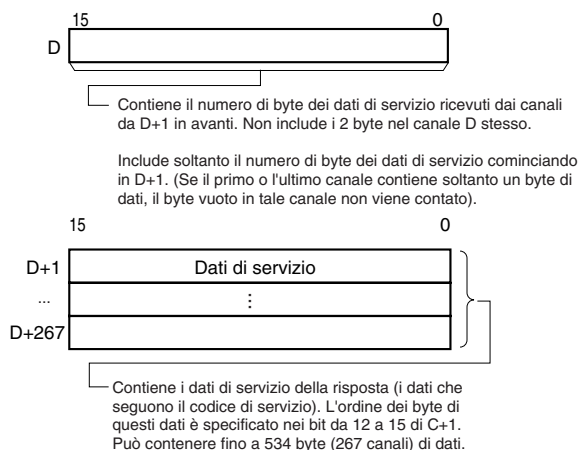
Formato del comando del messaggio esplicito



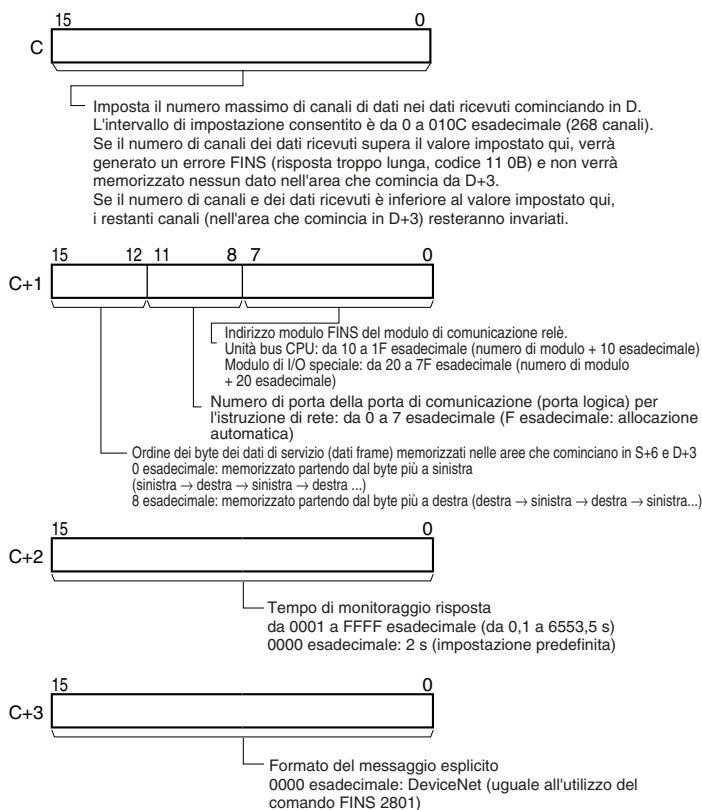
| | | | | | | |
|------|--------|---|---|---|---|--|
| S: | D00000 | 0 | 0 | 0 | A | Numero di byte di dati: S+1 ... S+5 = 5 canali = 10 byte = 0A esadecimale |
| S+1: | D00001 | 0 | 0 | 2 | D | Indirizzo del nodo slave = 45 = 2D esadecimale |
| S+2: | D00002 | 0 | 0 | 0 | E | Codice di servizio = 0E esadecimale |
| S+3: | D00003 | 0 | 0 | 0 | 9 | ID classe = 09 esadecimale |
| S+4: | D00004 | 0 | 0 | 0 | 3 | ID istanza = 03 esadecimale (Ingresso 3) |
| S+5: | D00005 | 0 | 0 | 6 | 6 | ID attributo = 66 esadecimale |
| D: | D00100 | 0 | 0 | 0 | 8 | Contiene 08 esadecimale per 8 byte di dati ricevuti nel frame di risposta. |
| D+1: | D00101 | 0 | 0 | 2 | D | Restituisce l'indirizzo del nodo slave = 45 = 2D esadecimale |
| D+2: | D00102 | 0 | 0 | 8 | E | Codice di servizio = 8E esadecimale (normale completamento) |
| D+3: | D00103 | 2 | 7 | F | E | Dati di servizio = 0029FE27 esadecimale (2.752.039 s decimale) |
| D+4: | D00104 | 2 | 9 | 0 | 0 | |
| C: | D00200 | 0 | 0 | 0 | 4 | Impostare 5 canali = 0005 esadecimale poiché vi sono 5 canali in D ... D+5. |
| C+1: | D00201 | 0 | 6 | 1 | 2 | Ordine dei byte = 0 esadecimale (dal byte più a sinistra), porta di comunicazione = 6 esadecimale (porta 6) e l'indirizzo modulo del modulo DeviceNet = 12 esadecimale |
| C+2: | D00202 | 0 | 0 | 0 | 0 | Tempo di monitoraggio risposta = 0000 esadecimale (2 s) |
| C+3: | D00203 | 0 | 0 | 0 | 0 | Tipo di formato esplicito = 0000 esadecimale (formato DeviceNet) |

D: primo canale del messaggio ricevuto

Specifica il primo canale del messaggio ricevuto (da D a D+267 max.).

**C: primo canale di controllo**

Specifica il primo di quattro canali di controllo (da C a C+3).

**Caratteristiche operando**

| Area | S | D | C |
|-------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6140 | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W508 | Da W000 a W511 | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H508 | Da H000 a H511 | Da H000 a H508 |
| Area bit ausiliaria | Da A000 a A956 | Da A000 a A959 | Da A000 a A956 |
| Area del temporizzatore | Da T0000 a T4092 | Da T0000 a T4095 | Da T0000 a T4092 |

| Area | S | D | C |
|--|--|--|--|
| Area del contatore | Da C0000 a C4092 | Da C0000 a C4095 | Da C0000 a C4092 |
| Area DM | Da D00000 a D32764 | Da D00000 a D32767 | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32764 | Da E00000 a E32767 | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Invia il comando di "lettura informazioni/stato" per il messaggio esplicito (memorizzato nei canali da S+1 a S+3) all'indirizzo del nodo specificato in S, tramite il modulo di comunicazione con l'indirizzo di modulo FINS specificato nei bit da 00 a 07 di C+1.

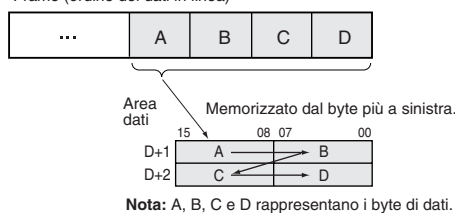
Quando si riceve la risposta al messaggio esplicito, i dati di servizio della risposta (i dati che seguono il codice di servizio) vengono memorizzati nell'intervallo di canali che comincia in D+1.

Il numero di byte dei dati ricevuti indicati in D è il numero di byte dei dati di servizio. Ad esempio, se vi è 1 byte di dati di servizio, D conterrà 0001 esadecimale. D conterrà 0001 esadecimale, indipendentemente dall'impostazione dell'ordine dei byte, ossia se il byte viene memorizzato nel byte più a destra o più a sinistra di D).

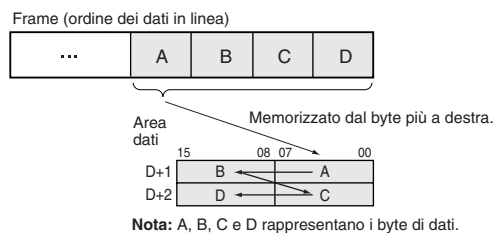
L'impostazione nei bit da 12 a 15 di C+1 (0 o 8 esadecimale) determina l'ordine dei byte dei dati di servizio memorizzati in S+6 e D+3.

- Memorizzazione dei dati dal byte più a sinistra
Impostare i bit da 12 a 15 di C+1 su 0 esadecimale.

Frame (ordine dei dati in linea)



- Memorizzazione dei dati dal byte più a destra
Impostare i bit da 12 a 15 di C+1 su 8 esadecimale.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C. OFF in tutti gli altri casi. |

Il flag di errore di comunicazione esplicita corrispondente sarà OFF se l'istruzione termina normalmente o ON se si verifica un errore.

Se subentra un errore (flag corrispondente in A213 ON), verrà utilizzato il rispettivo flag di errore porte di comunicazione per determinare se il messaggio esplicito non è stato inviato (flag corrispondente in A219 ON) o se il messaggio è stato inviato ma conteneva un errore (flag corrispondente in A219 OFF).

Il codice di completamento per le porte di comunicazione corrispondente (A203 - A210) sarà 0000 esadecimale se l'istruzione è terminata normalmente, un codice di errore del messaggio esplicito se si è verificato un errore nel messaggio esplicito o un codice di errore FINS se è subentrato un errore FINS.

Per ulteriori informazioni sul funzionamento generale delle istruzioni per messaggi espliciti, consultare *3-25-2 Informazioni sulle istruzioni per messaggi espliciti*.

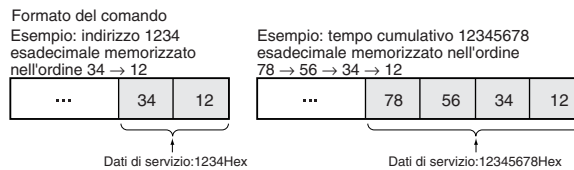
Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

| Nome | Indirizzo | Operazione |
|---|---------------------|--|
| Flag di abilitazione porte di comunicazione | Da A20200 a A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |
| Flag di errore di comunicazione esplicita | Da A21300 ad A21307 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di una comunicazione di messaggio esplicito. I flag saranno ON se il messaggio esplicito non è stato inviato o se è stato inviato ma è stata restituita una risposta di errore. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di messaggio esplicito successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |

| Nome | Indirizzo | Operazione |
|---|--------------------|---|
| Flag di errore porte di comunicazione | Da A21900 a A21907 | Questi flag sono ON ad indicare che il messaggio esplicito non è stato inviato dalle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di messaggio esplicito. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Codici di completamento per le porte di comunicazione | Da A203 a A210 | Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete. Il canale corrispondente conterrà 0000 mentre il flag di errore di comunicazione esplicita è OFF. Il canale corrispondente conterrà un codice di errore FINS quando il flag di errore di comunicazione esplicita e il flag di errore porte di comunicazione di tale porta sono entrambi ON. Il canale corrispondente conterrà il codice di errore appropriato relativo al messaggio esplicito quando il flag di errore di comunicazione esplicita di tale porta è ON e il flag di errore porte di comunicazione di tale porta è OFF. Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene avviata l'esecuzione del programma. |

Avvertenze

Assicurarsi che l'ordine dei byte nei dati di origine corrisponda all'ordine nel frame del messaggio esplicito (ordine dei dati nella linea). Ad esempio, quando i dati di servizio sono in unità di 2 o 4 byte, l'ordine dei dati nel frame sarà dal più a sinistra al più a destra in coppie a 2 cifre, come mostrato nello schema seguente.

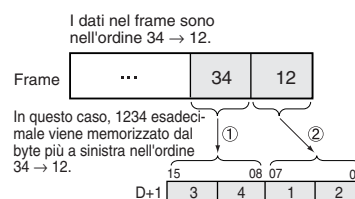


Nello schema seguente viene mostrato come i dati vengono memorizzati nelle aree dati quando i dati di servizio sono in unità da 2 o 4 byte.

1. Dati in unità da 2 byte

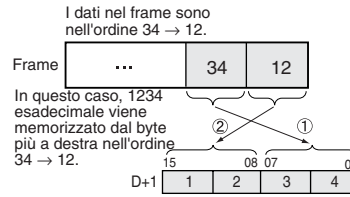
- Memorizzazione dei dati dal byte più a sinistra (bit da 12 a 15 di C = 0 esadecimale)

Esempio: memorizzazione del valore 1234 esadecimale in D+1



- Memorizzazione dei dati dal byte più a destra (bit da 12 a 15 di C = 8 esadecimale)

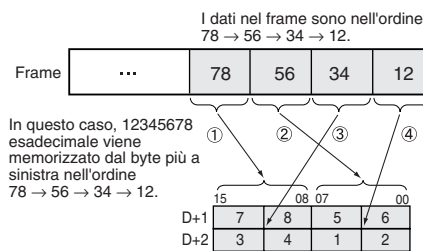
Esempio: memorizzazione del valore 1234 esadecimale in D+1



2. Dati in unità da 4 byte

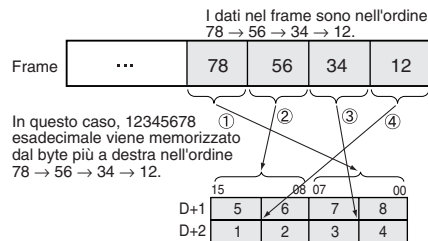
- Memorizzazione dei dati dal byte più a sinistra (bit da 12 a 15 di C = 0 esadecimale)

Esempio: memorizzazione del valore 12345678 esadecimale in D+1 e D+2



- Memorizzazione dei dati dal byte più a destra (bit da 12 a 15 di C = 8 esadecimale)

Esempio: memorizzazione del valore 12345678 esadecimale in D+1 e D+2



Esempio

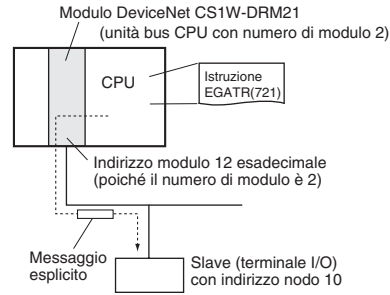
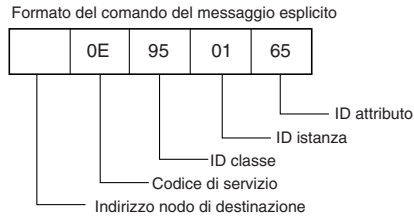
In questo esempio si utilizza EGATR(721) per leggere lo stato generale di uno slave DRT2 (terminale di I/O).



Quando CIO 000000 e A20206 (il flag di abilitazione porte di comunicazione per la porta 06) sono ON, l'istruzione EGATR(721) legge lo stato generale dello slave DRT2 (terminale di I/O). In questo caso viene letto il tempo di attivazione totale o il numero di operazioni di contatto per l'ingresso 3.

Codice di servizio = 0E esadecimale, ID classe = 95 esadecimale, ID istanza = 01 esadecimale e ID attributo = 65 esadecimale.

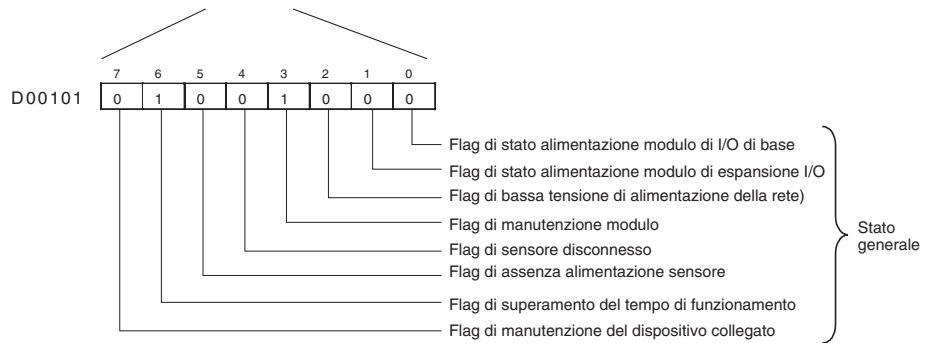
Lo stato generale viene restituito in 1 byte.



| | | | | | | |
|------|--------|---|---|---|---|--|
| S: | D00000 | 0 | 0 | 0 | A | Indirizzo del nodo slave = 10 = 0A esadecimale |
| S+1: | D00001 | 0 | 0 | 9 | 5 | ID classe = 95 esadecimale |
| S+2: | D00002 | 0 | 0 | 0 | 1 | ID istanza = 01 esadecimale |
| S+3: | D00003 | 0 | 0 | 6 | 5 | ID attributo = 65 esadecimale |

| | | | | | | |
|------|--------|---|---|---|---|--|
| C: | D00200 | 0 | 0 | 0 | 2 | Impostare 2 canali = 0002 esadecimale poiché vi sono 2 canali in D ... D+1. |
| C+1: | D00201 | 8 | 6 | 1 | 2 | Ordine dei byte = 8 esadecimale (dal byte più a destra), porta di comunicazione = 6 esadecimale (porta 6) e l'indirizzo modulo del modulo DeviceNet = 12 esadecimale |
| C+2: | D00202 | 0 | 0 | 0 | 0 | Tempo di monitoraggio risposta = 0000 esadecimale (2 s) |
| C+3: | D00203 | 0 | 0 | 0 | 0 | Tipo di formato esplicito = 0000 esadecimale (formato DeviceNet) |

| | | | | | | |
|------|--------|---|---|---|---|--|
| D: | D00100 | 0 | 0 | 0 | 1 | D contiene 0 esadecimale per 1 byte di dati restituiti nel byte più a destra di D+1. Lo stato generale dello slave viene restituito nei bit da 00 a 07. I dati vengono memorizzati nei bit 00 ... 07 poiché l'impostazione dell'ordine dei byte in C+1 (bit 12 - 15) è stata impostata su 8 esadecimale (dal byte più a destra). |
| D+1: | D00101 | 0 | 0 | 4 | 8 | |



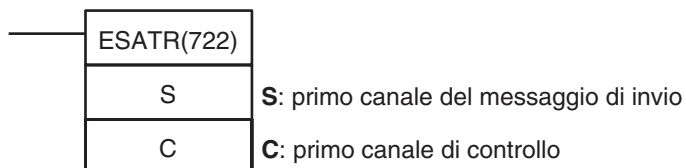
3-25-8 EXPLICIT SET ATTRIBUTE: ESATR(722)

Scopo

Invia un comando di scrittura informazioni in un messaggio esplicito (Set Attribute Single, codice di servizio: 10 esadecimale).

Questa istruzione è supportata solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | ESATR(722) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ESATR(722) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

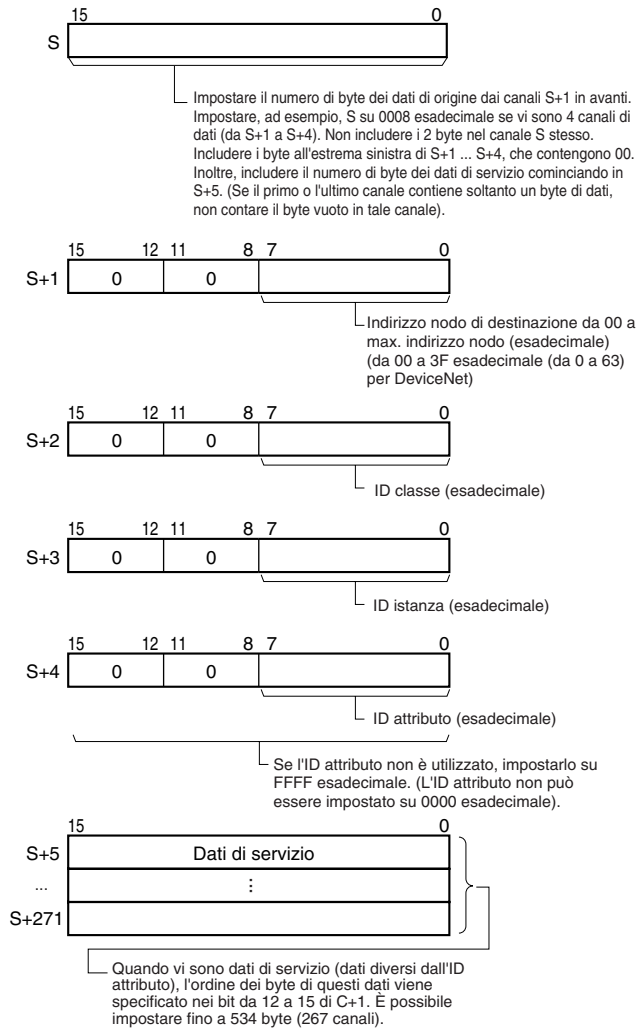
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

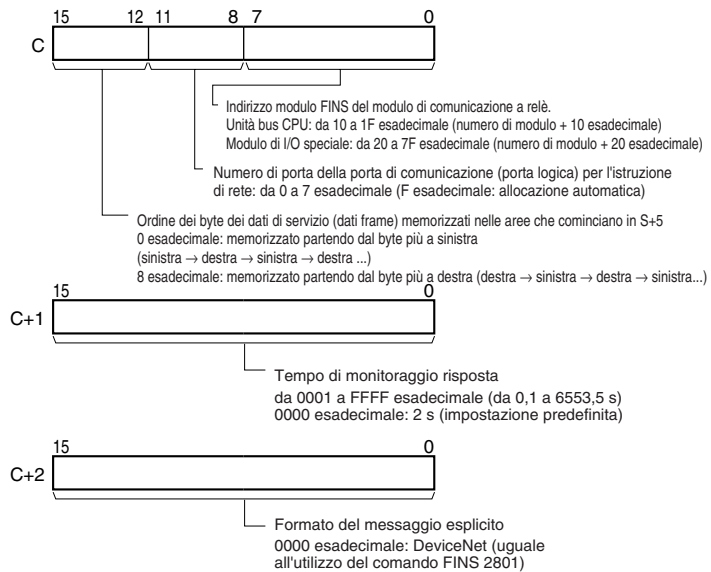
S: primo canale del messaggio di invio

Specifica il primo canale del messaggio di invio (da S a S+271 max.).



C: primo canale di controllo

Specifica il primo di tre canali di controllo (da C a C+2).



Caratteristiche operando

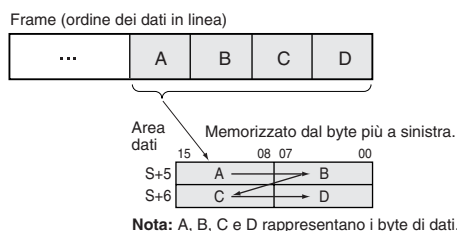
| Area | S | C |
|--|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6141 |
| Area di lavoro | Da W000 a W511 | Da W000 a W509 |
| Area bit di ritentività | Da H000 a H511 | Da H000 a H509 |
| Area bit ausiliaria | Da A000 a A959 | Da A000 ad A957 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4093 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4093 |
| Area DM | Da D00000 a D32767 | Da D00000 a D32765 |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32765 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32765 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

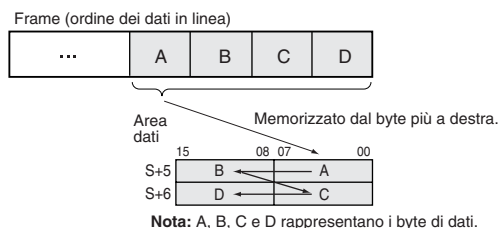
Invia un comando di messaggio esplicito con codice di servizio 10 esadecimale (memorizzato nell'intervallo di canali che comincia da S+2) all'indirizzo del nodo specificato in S+1, tramite il modulo di comunicazione con l'indirizzo di modulo FINS specificato nei bit da 00 a 07 di C. Quando si riceve la risposta al messaggio esplicito, essa viene memorizzata nell'intervallo di canali che comincia in D+2.

L'impostazione nei bit da 12 a 15 di C (0 o 8 esadecimale) determina l'ordine dei byte dei dati di servizio memorizzati in S+5.

- Memorizzazione dei dati dal byte più a sinistra
Impostare i bit da 12 a 15 di C su 0 esadecimale.



- Memorizzazione dei dati dal byte più a destra
Impostare i bit da 12 a 15 di C su 8 esadecimale.



Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C. OFF in tutti gli altri casi. |

Il flag di errore di comunicazione esplicita corrispondente sarà OFF se l'istruzione termina normalmente o ON se si verifica un errore.

Se subentra un errore (flag corrispondente in A213 ON), verrà utilizzato il rispettivo flag di errore porte di comunicazione per determinare se il messaggio esplicito non è stato inviato (flag corrispondente in A219 ON) o se il messaggio è stato inviato ma conteneva un errore (flag corrispondente in A219 OFF).

Il codice di completamento per le porte di comunicazione corrispondente (A203 - A210) sarà 0000 esadecimale se l'istruzione è terminata normalmente, un codice di errore del messaggio esplicito se si è verificato un errore nel messaggio esplicito o un codice di errore FINS se è subentrato un errore FINS.

Per ulteriori informazioni sul funzionamento generale delle istruzioni per messaggi espliciti, consultare 3-25-2 *Informazioni sulle istruzioni per messaggi espliciti*.

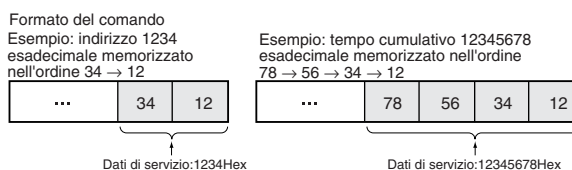
Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

| Nome | Indirizzo | Operazione |
|---|---------------------|--|
| Flag di abilitazione porte di comunicazione | Da A20200 a A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |
| Flag di errore di comunicazione esplicita | Da A21300 ad A21307 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di una comunicazione di messaggio esplicito. I flag saranno ON se il messaggio esplicito non è stato inviato o se è stato inviato ma è stata restituita una risposta di errore. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di messaggio esplicito successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |

| Nome | Indirizzo | Operazione |
|---|--------------------|---|
| Flag di errore porte di comunicazione | Da A21900 a A21907 | Questi flag sono ON ad indicare che il messaggio esplicito non è stato inviato dalle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di messaggio esplicito. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Codici di completamento per le porte di comunicazione | Da A203 a A210 | Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete. Il canale corrispondente conterrà 0000 mentre il flag di errore di comunicazione esplicita è OFF. Il canale corrispondente conterrà un codice di errore FINS quando il flag di errore di comunicazione esplicita e il flag di errore porte di comunicazione di tale porta sono entrambi ON. Il canale corrispondente conterrà il codice di errore appropriato relativo al messaggio esplicito quando il flag di errore di comunicazione esplicita di tale porta è ON e il flag di errore porte di comunicazione di tale porta è OFF. Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene avviata l'esecuzione del programma. |

Avvertenze

Assicurarsi che l'ordine dei byte nei dati di origine corrisponda all'ordine nel frame del messaggio esplicito (ordine dei dati nella linea). Ad esempio, quando i dati di servizio sono in unità di 2 o 4 byte, l'ordine dei dati nel frame sarà dal più a sinistra al più a destra in coppie a 2 cifre, come mostrato nello schema seguente.

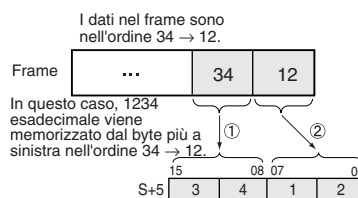


Nello schema seguente viene mostrato come i dati vengono memorizzati nelle aree dati quando i dati di servizio sono in unità da 2 o 4 byte.

1. Dati in unità da 2 byte

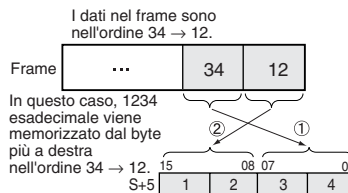
- Memorizzazione dei dati dal byte più a sinistra (bit da 12 a 15 di C = 0 esadecimale)

Esempio: memorizzazione del valore 1234 esadecimale in S+5



- Memorizzazione dei dati dal byte più a destra (bit da 12 a 15 di C = 8 esadecimale)

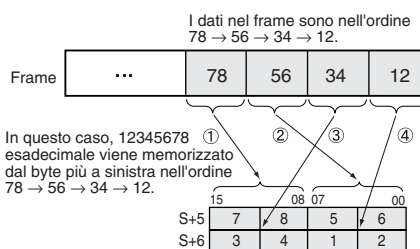
Esempio: memorizzazione del valore 1234 esadecimale in S+5



2. Dati in unità da 4 byte

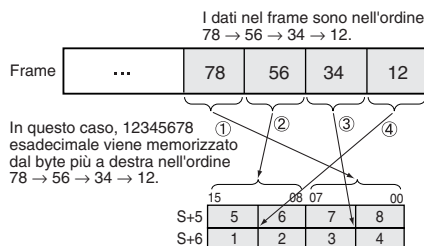
- Memorizzazione dei dati dal byte più a sinistra (bit da 12 a 15 di C = 0 esadecimale)

Esempio: memorizzazione del valore 12345678 esadecimale in S+5 e D+6



- Memorizzazione dei dati dal byte più a destra (bit da 12 a 15 di C = 8 esadecimale)

Esempio: memorizzazione del valore 12345678 esadecimale in S+5 e D+6



Esempio

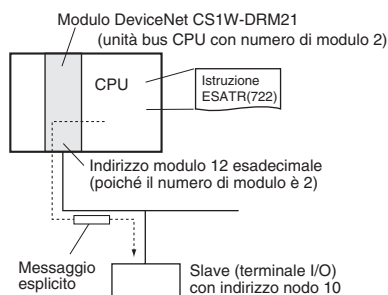
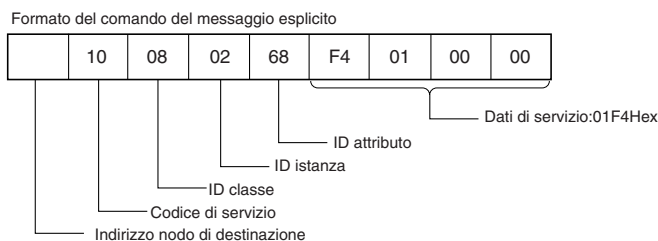
In questo esempio si utilizza ESATR(722) per sovrascrivere il valore impostato per il numero delle operazioni di contatto in uno slave DRT2 (terminale di I/O).



Quando CIO 000000 e A20206 (il flag di abilitazione porte di comunicazione per la porta 06) sono ON, l'istruzione EXPLT(720) scrive il valore impostato per il numero di operazioni di contatto in uno slave DRT2 (terminale di I/O).

Codice di servizio = 10 esadecimale, ID classe = 08 esadecimale, ID istanza = 02 esadecimale e ID attributo = 68 esadecimale.

In questo caso, il numero delle operazioni di contatto viene impostato su 500 (1F4 esadecimale), quindi i dati di servizio vengono impostati su 000001F4.



| | | | | | | |
|------|--------|---|---|---|---|---|
| S | D00000 | 0 | 0 | 0 | C | Numero di byte di dati: S+1 ... S+6 = 6 canali = 12 byte = 0C esadecimale |
| S:+1 | D00001 | 0 | 0 | 0 | A | Indirizzo del nodo slave = 10 = 0A esadecimale |
| S+2: | D00002 | 0 | 0 | 0 | 8 | ID classe = 08 esadecimale |
| S+3: | D00003 | 0 | 0 | 0 | 2 | ID istanza = 02 esadecimale |
| S+4: | D00004 | 0 | 0 | 6 | 8 | ID attributo = 68 esadecimale |
| S+5: | D00005 | 0 | 1 | F | 4 | Dati di servizio = F401 esadecimale |
| S+6: | D00006 | 0 | 0 | 0 | 0 | |

| | | | | | | |
|------|--------|---|---|---|---|--|
| C: | D00201 | 8 | 6 | 1 | 2 | Ordine dei byte = 8 esadecimale (dal byte più a destra), porta di comunicazione = 6 esadecimale (porta 6) e l'indirizzo modulo del modulo DeviceNet = 12 esadecimale |
| C+1: | D00202 | 0 | 0 | 0 | 0 | Tempo di monitoraggio risposta = 0000 esadecimale (2 s) |
| C+2: | D00203 | 0 | 0 | 0 | 0 | Tipo di formato esplicito = 0000 esadecimale (formato DeviceNet) |

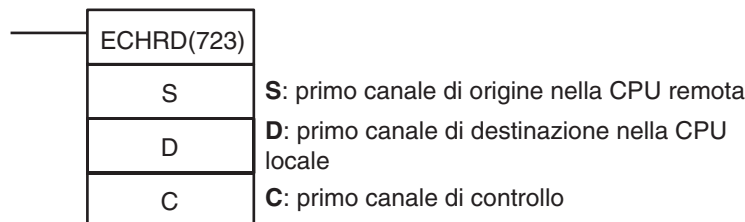
3-25-9 EXPLICIT WORD READ: ECHRD(723)

Scopo

Legge i dati sulla CPU locale da un'altra CPU collegata in rete (la CPU remota deve supportare i messaggi espliciti).

Questa istruzione è supportata solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ECHRD(723) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ECHRD(723) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: primo canale di origine nella CPU remota

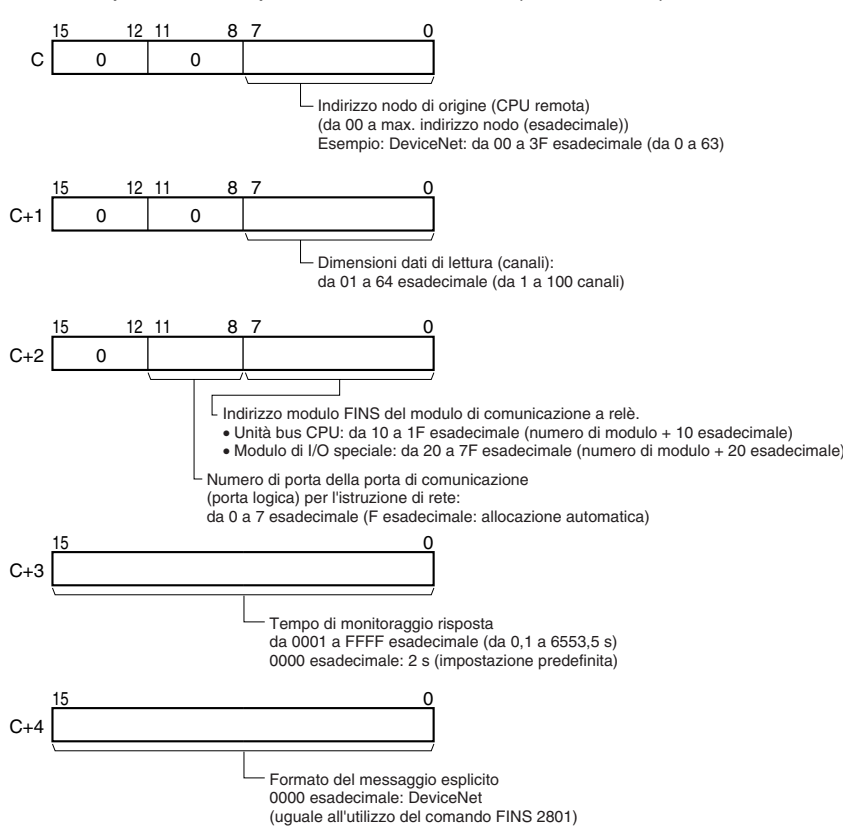
Specifica l'indirizzo iniziale del canale contenente i dati da leggere dalla CPU remota.

D: primo canale di destinazione nella CPU locale

Specifica l'indirizzo iniziale del canale in cui verranno memorizzati i dati di lettura nella CPU locale.

C: primo canale di controllo

Specifica il primo di cinque canali di controllo (da C a C+4).



Caratteristiche operando

| Area | S | D | C |
|-------------------------|--|----------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6139 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W507 |
| Area bit di ritentività | Da H000 a H511 | | Da H000 a H507 |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A959 | Da A000 a A955 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4091 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4091 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32763 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32763 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32763 (n = da 0 a C) |

| Area | S | D | C |
|--|---|---|---|
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Legge il numero specificato di canali dal primo canale di lettura (specificato in S) nella CPU remota con l'indirizzo di nodo specificato in C e memorizza i dati nei canali di memoria della CPU locale cominciando da D.

Nota L'istruzione ECHRD(723) invia un messaggio esplicito con il codice di servizio 1C esadecimale (Byte Data Read).

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C. OFF in tutti gli altri casi. |

Il flag di errore di comunicazione esplicita corrispondente sarà OFF se l'istruzione termina normalmente o ON se si verifica un errore.

Se subentra un errore (flag corrispondente in A213 ON), verrà utilizzato il rispettivo flag di errore porte di comunicazione per determinare se il messaggio esplicito non è stato inviato (flag corrispondente in A219 ON) o se il messaggio è stato inviato ma conteneva un errore (flag corrispondente in A219 OFF).

Il codice di completamento per le porte di comunicazione corrispondente (A203 - A210) sarà 0000 esadecimale se l'istruzione è terminata normalmente, un codice di errore del messaggio esplicito se si è verificato un errore nel messaggio esplicito o un codice di errore FINS se è subentrato un errore FINS.

Per ulteriori informazioni sul funzionamento generale delle istruzioni di rete, consultare 3-25-2 *Informazioni sulle istruzioni per messaggi espliciti*.

Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

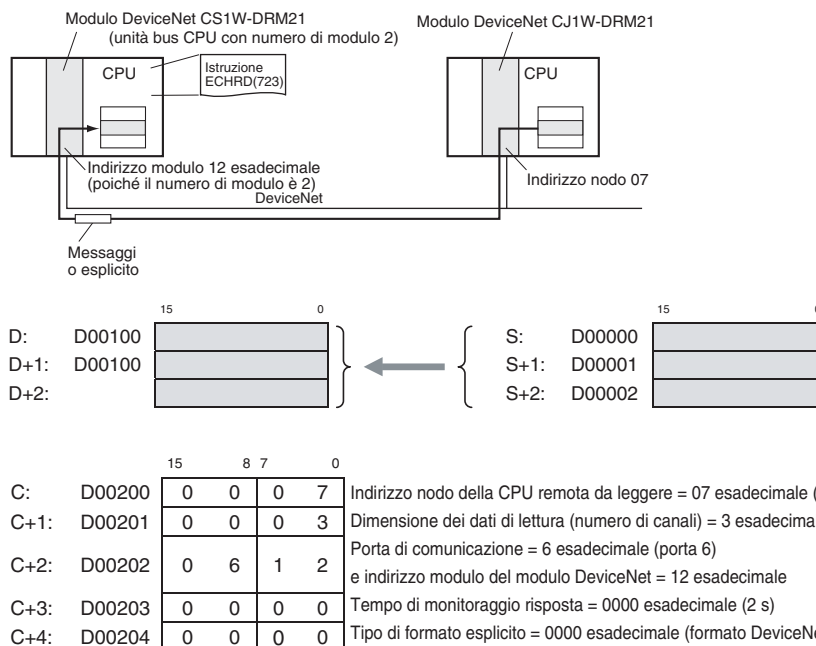
| Nome | Indirizzo | Operazione |
|---|---------------------|---|
| Flag di abilitazione porte di comunicazione | Da A20200 a A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |
| Flag di errore di comunicazione esplicita | Da A21300 ad A21307 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di una comunicazione di messaggio esplicito. I flag saranno ON se il messaggio esplicito non è stato inviato o se è stato inviato ma è stata restituita una risposta di errore. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di messaggio esplicito successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Flag di errore porte di comunicazione | Da A21900 a A21907 | Questi flag sono ON ad indicare che il messaggio esplicito non è stato inviato dalle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di messaggio esplicito. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |
| Codici di completamento per le porte di comunicazione | Da A203 a A210 | Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete. Il canale corrispondente conterrà 0000 mentre il flag di errore di comunicazione esplicita è OFF. Il canale corrispondente conterrà un codice di errore FINS quando il flag di errore di comunicazione esplicita e il flag di errore porte di comunicazione di tale porta sono entrambi ON. Il canale corrispondente conterrà il codice di errore appropriato relativo al messaggio esplicito quando il flag di errore di comunicazione esplicita di tale porta è ON e il flag di errore porte di comunicazione di tale porta è OFF. Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene avviata l'esecuzione del programma. |

Esempio

In questo esempio si utilizza ECHR(723) per leggere la memoria di I/O di una CPU della serie CJ sulla rete DeviceNet e memorizzare i dati nella memoria di I/O della CPU locale.



Quando CIO 000000 e A20206 (il flag di abilitazione porte di comunicazione per la porta 06) sono ON, l'istruzione ECHRD(723) legge da D00000 a D00002 dalla memoria di I/O della CPU della serie CJ con indirizzo di nodo 07 sulla rete DeviceNet e memorizza i dati in D00100 - D00102 della CPU locale.



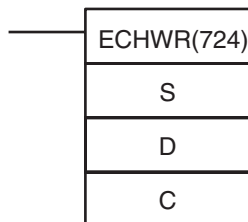
3-25-10 EXPLICIT WORD WRITE: ECHWR(724)

Scopo

Scrive i dati dalla CPU locale a un'altra CPU collegata in rete (la CPU remota deve supportare i messaggi espliciti).

Questa istruzione è supportata solo da CPU della serie CS/CJ versione 2.0 o successiva.

Simbolo programmazione ladder



S: primo canale di origine nella CPU locale
D: primo canale di destinazione nella CPU remota
C: primo canale di controllo

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | ECHWR(724) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @ECHWR(724) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

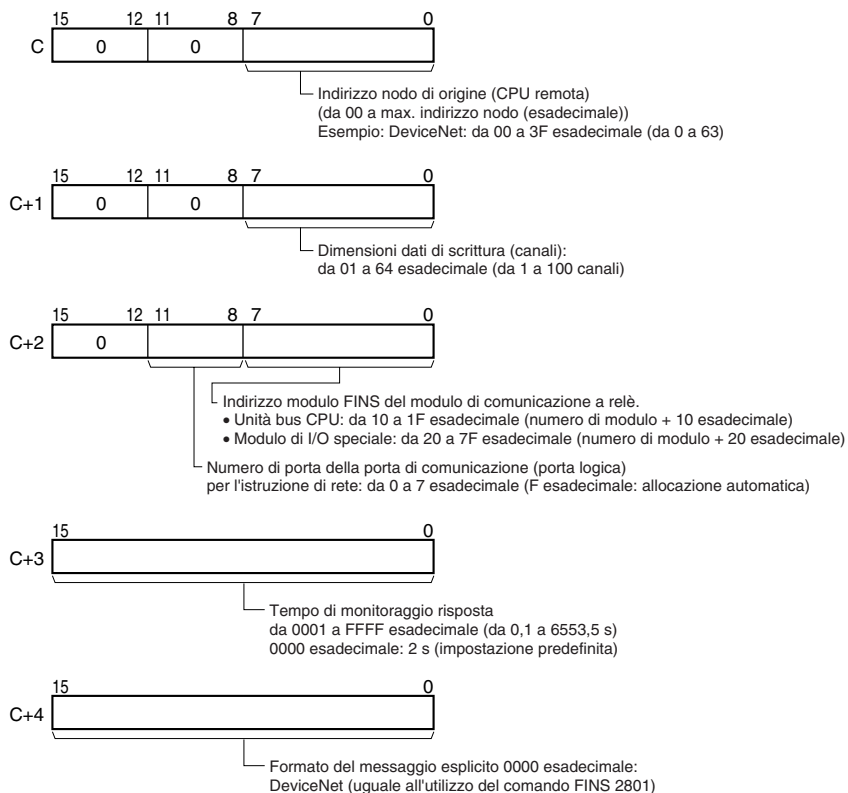
S: primo canale di origine nella CPU locale
 Specifica l'indirizzo del canale iniziale nella CPU locale contenente i dati di scrittura.

D: primo canale di destinazione nella CPU remota

Specifica l'indirizzo del canale iniziale della destinazione di scrittura nella CPU remota.

C: primo canale di controllo

Specifica il primo di cinque canali di controllo (da C a C+4).

**Caratteristiche operando**

| Area | S | D | C |
|--------------------------------------|--|----------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 | | Da CIO 0000 a CIO 6139 |
| Area di lavoro | Da W000 a W511 | | Da W000 a W507 |
| Area bit di ritenività | Da H000 a H511 | | Da H000 a H507 |
| Area bit ausiliaria | Da A000 a A959 | Da A448 a A959 | Da A000 a A955 |
| Area del temporizzatore | Da T0000 a T4095 | | Da T0000 a T4091 |
| Area del contatore | Da C0000 a C4095 | | Da C0000 a C4091 |
| Area DM | Da D00000 a D32767 | | Da D00000 a D32763 |
| Area EM senza banco | Da E00000 a E32767 | | Da E00000 a E32763 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | Da En_00000 a En_32763 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |

| Area | S | D | C |
|--|--|---|---|
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Scrive il numero specificato di canali cominciando da S dalla CPU locale nella destinazione di scrittura cominciando da D nella CPU remota con l'indirizzo del nodo specificato in C.

Nota L'istruzione ECHWR(724) invia un messaggio esplicito con il codice di servizio 1E esadecimale (Byte Data Write).

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se il flag di abilitazione porte di comunicazione è OFF per il numero della porta di comunicazione specificata in C. OFF in tutti gli altri casi. |

Il flag di errore di comunicazione esplicita corrispondente sarà OFF se l'istruzione termina normalmente o ON se si verifica un errore.

Se subentra un errore (flag corrispondente in A213 ON), verrà utilizzato il rispettivo flag di errore porte di comunicazione per determinare se il messaggio esplicito non è stato inviato (flag corrispondente in A219 ON) o se il messaggio è stato inviato ma conteneva un errore (flag corrispondente in A219 OFF).

Il codice di completamento per le porte di comunicazione corrispondente (A203 - A210) sarà 0000 esadecimale se l'istruzione è terminata normalmente, un codice di errore del messaggio esplicito se si è verificato un errore nel messaggio esplicito o un codice di errore FINS se è subentrato un errore FINS.

Per ulteriori informazioni sul funzionamento generale delle istruzioni per messaggi espliciti, consultare 3-25-2 *Informazioni sulle istruzioni per messaggi espliciti*.

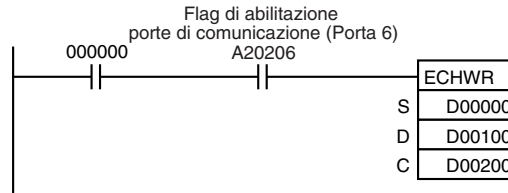
Nella tabella seguente sono riportati i bit e i flag relativi all'area ausiliaria.

| Nome | Indirizzo | Operazione |
|---|---------------------|--|
| Flag di abilitazione porte di comunicazione | Da A20200 a A20207 | Questi flag sono ON ad indicare che è possibile eseguire tali istruzioni di rete, inclusa PMCR(260), per le porte corrispondenti (da 00 a 07). Un flag viene disattivato quando un'istruzione di rete è in corso di esecuzione per la porta corrispondente e riattivato quando l'istruzione è stata completata. |
| Flag di errore di comunicazione esplicita | Da A21300 ad A21307 | Questi flag sono ON ad indicare che si è verificato un errore sulle porte corrispondenti (da 00 a 07) durante l'esecuzione di una comunicazione di messaggio esplicito. I flag saranno ON se il messaggio esplicito non è stato inviato o se è stato inviato ma è stata restituita una risposta di errore. Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di messaggio esplicito successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore. |

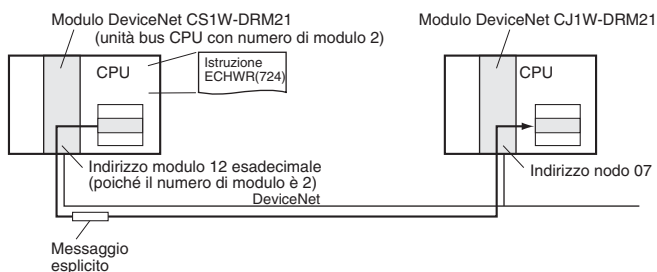
| Nome | Indirizzo | Operazione |
|---|--------------------|--|
| Flag di errore porte di comunicazione | Da A21900 a A21907 | <p>Questi flag sono ON ad indicare che il messaggio esplicito non è stato inviato dalle porte corrispondenti (da 00 a 07) durante l'esecuzione di un'istruzione di messaggio esplicito.</p> <p>Lo stato del flag viene mantenuto fino all'esecuzione dell'istruzione di rete successiva. Il flag verrà disattivato quando viene eseguita l'istruzione successiva, anche se in precedenza si è verificato un errore.</p> |
| Codici di completamento per le porte di comunicazione | Da A203 a A210 | <p>Questi canali contengono i codici di completamento per le porte corrispondenti (da 00 a 07) dopo l'esecuzione di un'istruzione di rete.</p> <p>Il canale corrispondente conterrà 0000 mentre il flag di errore di comunicazione esplicita è OFF.</p> <p>Il canale corrispondente conterrà un codice di errore FINS quando il flag di errore di comunicazione esplicita e il flag di errore porte di comunicazione di tale porta sono entrambi ON.</p> <p>Il canale corrispondente conterrà il codice di errore appropriato relativo al messaggio esplicito quando il flag di errore di comunicazione esplicita di tale porta è ON e il flag di errore porte di comunicazione di tale porta è OFF.</p> <p>Il canale corrispondente conterrà 0000 mentre è in corso l'esecuzione dell'istruzione di rete e il codice di completamento verrà scritto quando l'istruzione è stata completata. Questi canali vengono cancellati quando viene avviata l'esecuzione del programma.</p> |

Esempio

In questo esempio si utilizza ECHWR(724) per scrivere i dati dalla memoria di I/O della CPU locale nella memoria di I/O di una CPU della serie CJ sulla rete DeviceNet.



Quando CIO 000000 e A20206 (il flag di abilitazione porte di comunicazione per la porta 06) sono ON, l'istruzione ECHWR(724) legge da D00000 a D00002 dalla memoria di I/O della CPU locale e memorizza i dati in D00100 - D00102 nella CPU della serie CJ con indirizzo di nodo 07 sulla rete DeviceNet.



| | 15 | 8 | 7 | 0 | | |
|------|--------|---|---|---|---|--|
| C: | D00200 | 0 | 0 | 0 | 7 | Indirizzo nodo della CPU remota in cui scrivere = 07 esadecimale (nodo 07) Dimensione dei dati di scrittura (numero di canali) = 3 esadecimale Porta di comunicazione = 6 esadecimale (porta 6) e indirizzo modulo del modulo DeviceNet = 12 esadecimale Tempo di monitoraggio risposta = 0000 esadecimale (2 s) Tipo di formato esplicito = 0000 esadecimale (formato DeviceNet) |
| C+1: | D00201 | 0 | 0 | 0 | 3 | |
| C+2: | D00202 | 0 | 6 | 1 | 2 | |
| C+3: | D00203 | 0 | 0 | 0 | 0 | |
| C+4: | D00204 | 0 | 0 | 0 | 0 | |

3-26 Istruzioni per la memoria dei file

In questa sezione vengono descritte le istruzioni utilizzate con la memoria dei file (area EM o schede di memoria).

Nota È possibile anche manipolare la memoria dei file eseguendo l'istruzione CMND(490) per inviare un comando FINS alla CPU locale. Per ulteriori informazioni, fare riferimento al manuale *CS/CJ-series PLC Operation Manual*.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|-----------------|-----------|-----------------|--------|
| READ DATA FILE | FREAD | 700 | 1045 |
| WRITE DATA FILE | FWRIT | 701 | 1052 |

3-26-1 Avvertenze per l'utilizzo delle schede di memoria

Prima di utilizzare una scheda di memoria, prendere in considerazione i seguenti punti.

Formato

Le schede di memoria sono già formattate al momento dell'acquisto, non è quindi necessario formattarle. Per formattare una scheda usata, utilizzare sempre la CPU e CX-Programmer o una console di programmazione.

Se la scheda di memoria viene formattata direttamente in un notebook o in un altro computer, la CPU potrebbe non riconoscerla. In tal caso, non sarà possibile utilizzare la scheda anche dopo averla riformattata nella CPU.

Numero di file nella directory principale

Il numero di file che è possibile memorizzare nella directory principale di una scheda di memoria è limitato, così come esistono limitazioni per i dischi rigidi. Il limite dipende dal tipo e dal formato della scheda di memoria, ma sarà comunque compreso tra 128 e 512 file. Quando si utilizzano applicazioni che prevedono la scrittura di file di log o altri file a intervalli periodici, impostare il programma per la scrittura dei file in una sottodirectory anziché nella directory principale.

È possibile creare sottodirectory utilizzando un computer o l'istruzione CMND(490). Per un esempio specifico sull'utilizzo dell'istruzione CMND(490), fare riferimento a 3-25-5 *DELIVER COMMAND: CMND(490)*.

Numero di operazioni di scrittura

In generale, non esistono limitazioni relative al numero di operazioni di scrittura che è possibile eseguire nella memoria flash. Ai fini della garanzia, tuttavia, per le schede di memoria è stato stabilito un limite di 100.000 operazioni di scrittura. Ad esempio, se si esegue un'operazione di scrittura sulla scheda di memoria ogni 10 minuti, nell'arco di due anni verranno eseguite più di 100.000 operazioni di scrittura.

Dimensione minima dei file

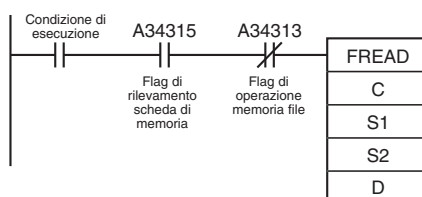
Se sulla scheda di memoria vengono salvati numerosi file di piccole dimensioni, ad esempio file contenenti soltanto pochi canali di dati dell'area DM, non sarà possibile sfruttare completamente la capacità della scheda di memoria. Ad esempio, se viene utilizzata una scheda di memoria con un'unità di allocazione di 4.096, per ciascun file verranno occupati almeno 4.096 byte di memoria, indipendentemente dalle dimensioni effettive del file. Se si salvano 10 canali di dati dell'area DM sulla scheda di memoria, verranno utilizzati 4.096 byte di memoria, anche se la dimensione effettiva del file è di 68 byte. L'impiego di file di dimensioni così ridotte riduce notevolmente il tasso di utilizzo della scheda di memoria. Tuttavia, se per migliorare l'utilizzo si riducono le dimensioni dell'unità di allocazione, verrà ridotta anche la velocità di accesso.

È possibile verificare le dimensioni dell'unità di allocazione della scheda di memoria utilizzando il comando CHKDSK dal prompt del DOS. La procedura specifica non è riportata in questa sede. Per ulteriori informazioni sulle dimensioni dell'unità di allocazione, consultare la documentazione del computer.

Avvertenze per l'accesso alla scheda di memoria

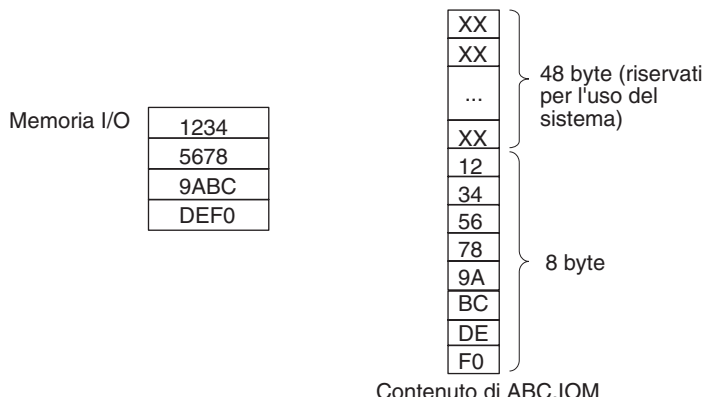
Quando il PLC accede alla scheda di memoria, l'indicatore BUSY della CPU si illumina. Osservare le seguenti precauzioni.

- 1,2,3...**
1. Non spegnere mai la CPU quando l'indicatore BUSY è acceso. Ciò può rendere inutilizzabile la scheda di memoria.
 2. Non rimuovere mai la scheda di memoria dalla CPU quando l'indicatore BUSY è illuminato. Prima di rimuovere la scheda di memoria, premere il pulsante di alimentazione della scheda per interrompere l'alimentazione e attendere che l'indicatore BUSY si spenga. In caso contrario, la scheda di memoria potrebbe venire danneggiata e non essere più utilizzabile.
 3. Inserire la scheda di memoria con l'etichetta rivolta a destra. Non tentare di inserirla con qualsiasi altro orientamento. Ciò potrebbe danneggiare la scheda di memoria o la CPU.
 4. Dopo l'inserimento, sono sufficienti pochi secondi per il riconoscimento della scheda di memoria da parte della CPU. Quando si effettua l'accesso a una scheda di memoria subito dopo l'accensione o l'inserimento della scheda, programmare una condizione NC per il flag di riconoscimento della scheda di memoria (A34315) come condizione di ingresso, come indicato di seguito.

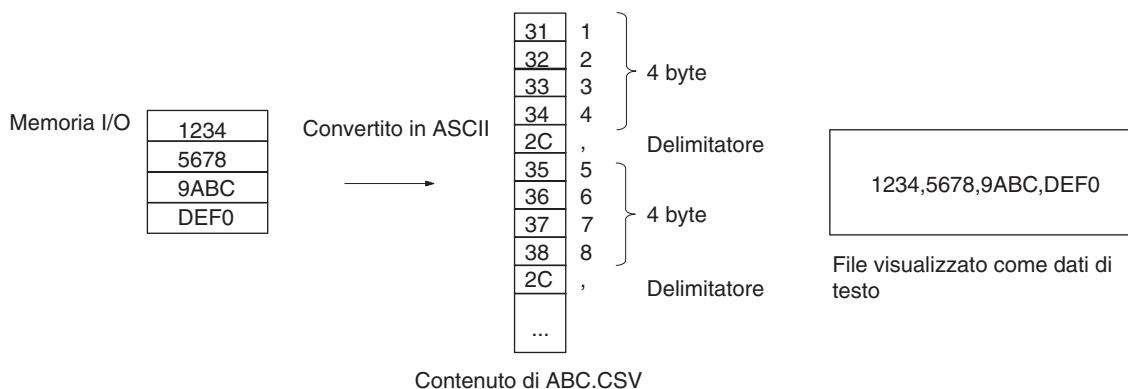


Nota La struttura dei file di dati è mostrata di seguito.

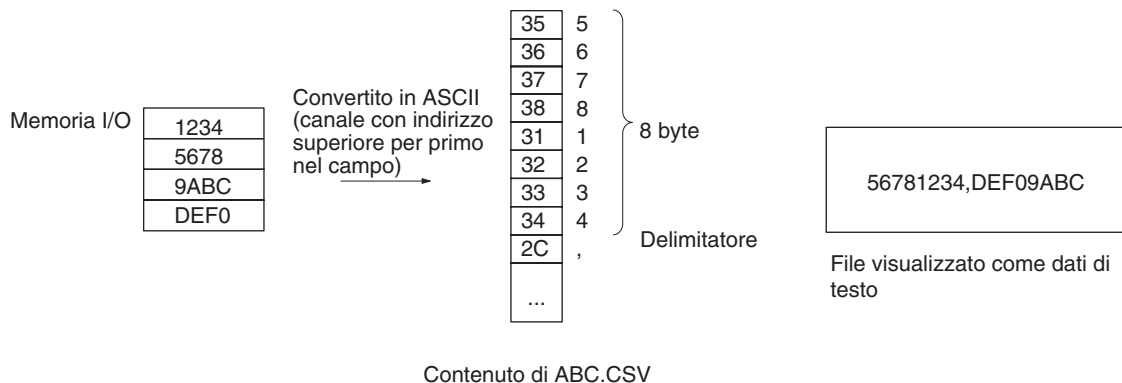
Per il formato binario (.IOM), i dati saranno come indicato di seguito quando 1234 esadecimale, 5678 esadecimale, 9ABC esadecimale e DEF0 esadecimale vengono memorizzati nel file ABC.IOM (anche se normalmente l'utente non deve occuparsi di questa struttura):



Per il formato CSV dei canali (.CSV), i dati saranno come indicato di seguito quando 1234 esadecimale, 5678 esadecimale, 9ABC esadecimale e DEF0 esadecimale vengono memorizzati nel file ABC.CSV (la struttura di base sarà la stessa per i dati di testo (.TXT)):

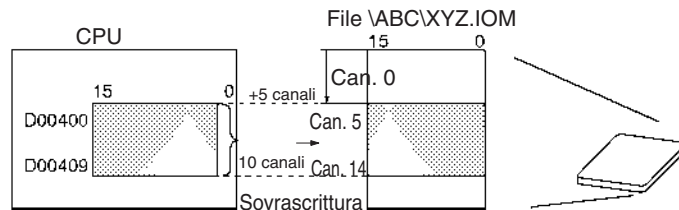
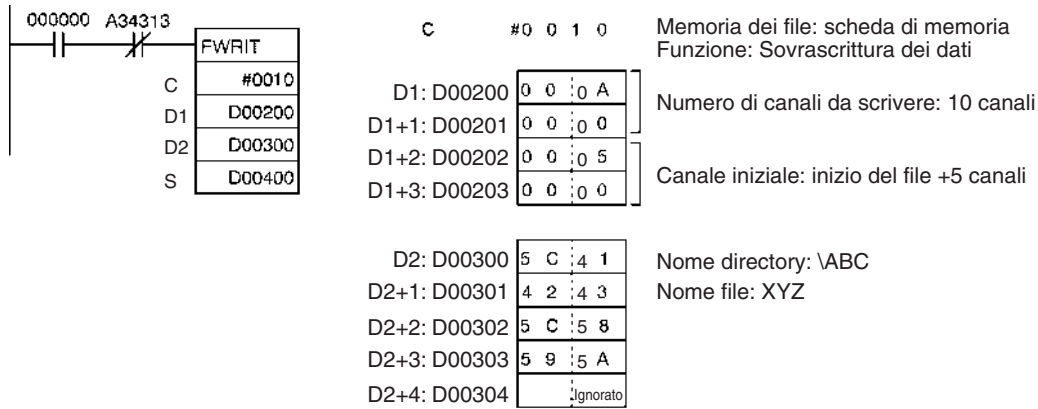


Per il formato CSV di canali lunghi (.CSV), i dati saranno come indicato di seguito quando 1234 esadecimale, 5678 esadecimale, 9ABC esadecimale e DEF0 esadecimale vengono memorizzati nel file ABC.CSV (la struttura di base sarà la stessa per i dati di testo (.TXT)):



Esempi

Nell'esempio seguente, quando CIO 000000 viene impostato su ON, FWRIT(701) legge 10 canali di dati da D00400 - D00409 e utilizza tali dati per sovrascrivere 10 canali nel file \ABC\XYZ.IOM cominciando dall'inizio del file + 5 canali.

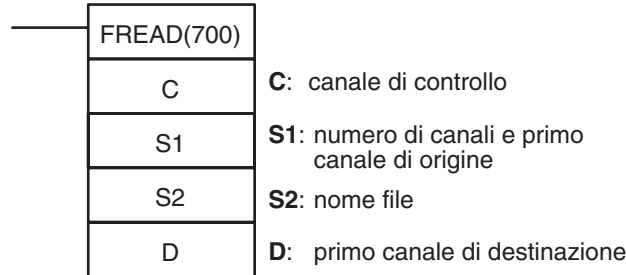


3-26-2 READ DATA FILE: FREAD(700)

Scopo

Legge la quantità di dati o i dati specificati dal file di dati indicato nella memoria dei file nell'area dati specificata della CPU.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FREAD(700) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FREAD(700) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

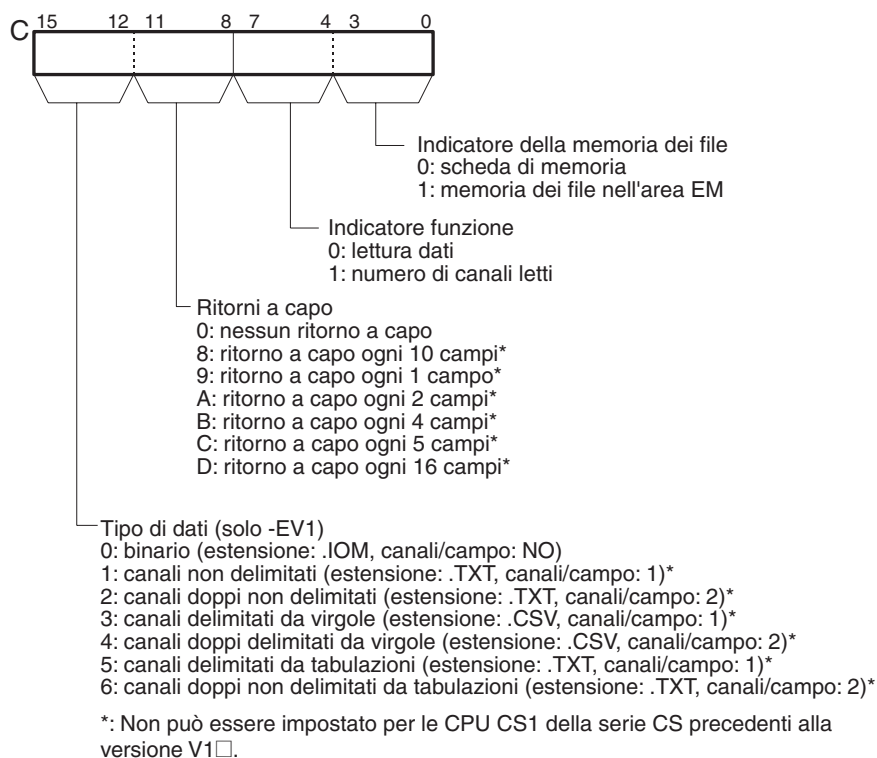
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo

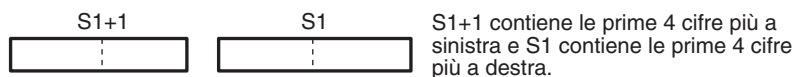
Come mostrato nello schema seguente, la prima cifra indica se il file di origine è nella scheda di memoria o nella memoria dei file nell'area EM, la seconda cifra del canale di controllo indica se devono essere letti i dati attuali o il numero dei canali di dati, la terza cifra indica la presenza di ritorni a capo e la quarta cifra indica il tipo di dati.



- Nota**
- Ogni campo conterrà 1 canale di memoria di I/O per il tipo di dati del canale e 2 canali di memoria di I/O per i tipi di dati a doppio canale.
 - Quando si leggono i dati con ritorni a capo, i bit da 00 a 11 di C devono essere impostati tra 8 e D esadecimale.
 - Con i canali doppi, il primo canale di dati viene memorizzato nell'indirizzo di memoria superiore, ossia verrebbe memorizzato 12345678 con 1234 in D00001 e 5678 in D00000.

S1 e S1+1: numero di elementi letti

Il valore esadecimale a 8 cifre in S1 e S1+1 specifica la quantità di canali o campi da leggere dalla memoria dei file. Se il numero di canali o campi specificati supera il numero di canali nel file di dati, i dati nel file verranno trasferiti normalmente e non si verificherà alcun errore.



| Tipo di dati | Bit da 12 a 15 di C | Contenuto di S1 e S1+1 |
|---------------|---|---|
| Binario | 0 esadecimale (binario) | Numero di canali da leggere dalla memoria dei file. Da 00000000 a 3FFFFFFF esadecimale |
| Canale | 1 esadecimale (non delimitato), 3 esadecimale (delimitato da virgole) o 5 esadecimale (delimitato da tabulazioni) | Numero di campi da leggere dalla memoria dei file, ossia il numero di canali da leggere dalla memoria dei file. Da 00000000 a 1FFFFFFF esadecimale |
| Doppio canale | 2 esadecimale (non delimitato), 4 esadecimale (delimitato da virgole) o 6 esadecimale (delimitato da tabulazioni) | Numero di campi da leggere dalla memoria dei file, ossia metà del numero di canali da leggere dalla memoria dei file. Da 00000000 a 0FFFFFFF esadecimale |

S1+2 e S1+3: primo canale di origine

Il valore esadecimale a 8 cifre in S1+2 e S1+3 specifica il canale iniziale di lettura dall'inizio del file.



| Tipo di dati | Bit da 12 a 15 di C | Contenuto di S1+2 e S1+3 |
|---------------|---|---|
| Binario | 0 esadecimale (binario) | Il canale da cui si comincia la lettura dall'inizio della memoria dei file. Da 00000000 a 3FFFFFFF esadecimale |
| Canale | 1 esadecimale (non delimitato), 3 esadecimale (delimitato da virgole) o 5 esadecimale (delimitato da tabulazioni) | Il campo da cui si comincia la lettura dall'inizio della memoria dei file, ossia il numero di canali dall'inizio. Da 00000000 a 1FFFFFFF esadecimale |
| Doppio canale | 2 esadecimale (non delimitato), 4 esadecimale (delimitato da virgole) o 6 esadecimale (delimitato da tabulazioni) | Il campo da cui si comincia la lettura dall'inizio della memoria dei file, ossia metà del numero di canali dall'inizio. Da 00000000 a 0FFFFFFF esadecimale |

- Nota**
1. S1+2 e S1+3 sono utilizzati soltanto per dati di tipo testo e CVS senza ritorni a capo (ovvero bit da 08 a 11 di C impostati su 0 esadecimale) o per dati binari. Impostare sempre S1+2 e S1+3 su 00000000 esadecimale quando si leggono dati con ritorni a capo (ovvero bit da 08 a 11 di C impostati su 8 - D esadecimale).
 2. S1 - S1+3 devono essere nella stessa area dati.
 3. S1 - S1+3 vengono utilizzati solo per la lettura dei dati.
 4. Se il canale iniziale specificato supera il numero di canali nel file di dati, verrà attivato il flag di errore lettura file (A34310) e i dati del file non verranno letti.

S2: nome file

S2 è l'indirizzo iniziale dei canali contenenti il percorso assoluto e il nome file in ASCII. Utilizzare ASCII a - z, A - Z e 0 - 9.

Il nome completo del percorso alla directory contenente il file di dati può essere lungo fino a 65 caratteri, inclusa la barra iniziale (ASCII 5C). Il nome del file può contenere fino a 8 caratteri, ma non sono ammessi caratteri nulli (ASCII 00) poiché un carattere nullo viene usato per contrassegnare la fine di ogni stringa di caratteri. Non includere l'estensione del nome file; l'estensione .IOM verrà aggiunta automaticamente.

| | | | |
|-------|-----|-----|--|
| S2 | F1 | F2 | Memorizzare la stringa di caratteri che comincia con il byte più a sinistra in S2. L'intero nome del percorso e il nome del file può contenere fino a 74 caratteri (byte), incluso il carattere iniziale della barra e il carattere nullo di fine. |
| S2+1 | F3 | F4 | |
| ⋮ | ⋮ | ⋮ | |
| ⋮ | ⋮ | ⋮ | |
| S2+38 | F73 | F74 | |

- Nota**
1. Accertarsi che la stringa di caratteri contenente il nome del percorso e il nome file non superi la fine dell'area dati.
 2. Se il file o la directory specificata non esiste, verrà attivato il flag di file mancante (A34311) e i dati del file non verranno letti.

Scrivere il nome del percorso e il nome file in ASCII cominciando dal byte più a sinistra di S2, come mostrato nell'esempio seguente per \ABC\XYZ.IOM. (L'estensione .IOM viene aggiunta automaticamente).

| | | | | | |
|------|-----|-----|------|----|----|
| S2 | "\" | "A" | S2 | 5C | 41 |
| S2+1 | "B" | "C" | S2+1 | 42 | 43 |
| S2+2 | "\" | "X" | S2+2 | 5C | 58 |
| S2+3 | "Y" | "Z" | S2+3 | 59 | 5A |
| S2+4 | NUL | | S2+4 | 00 | |

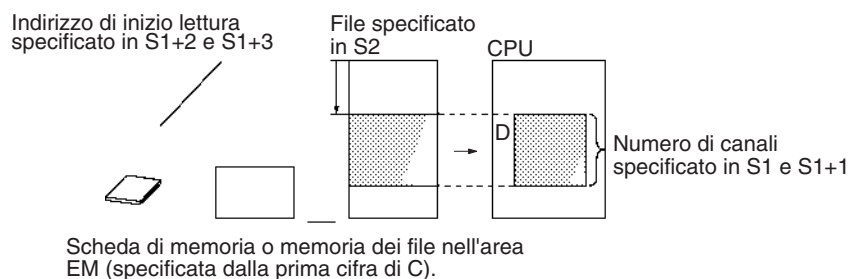
D: primo canale di destinazione

Quando i dati vengono letti, D specifica l'indirizzo iniziale in cui verranno memorizzati i dati letti dalla memoria dei file.

Quando è in corso la lettura del numero dei canali di dati, il numero dei canali viene scritto in D e D+1 in esadecimale a 8 cifre (da 00000000 a 7FFFFFFF). D contiene le prime 4 cifre più a destra e D+1 contiene le prime 4 cifre più a sinistra.

Descrizione**Letture dei dati (terza cifra di C = 0)**

FREAD(700) legge il numero di canali o campi specificato in S1 e S1+1 dal file specificato in S2 (con estensione del nome di file .IOM, .TXT o .CSV), cominciando dall'indirizzo specificato in S1+2 e S1+3. I dati vengono quindi scritti nella RAM, cominciando dal canale specificato in D.

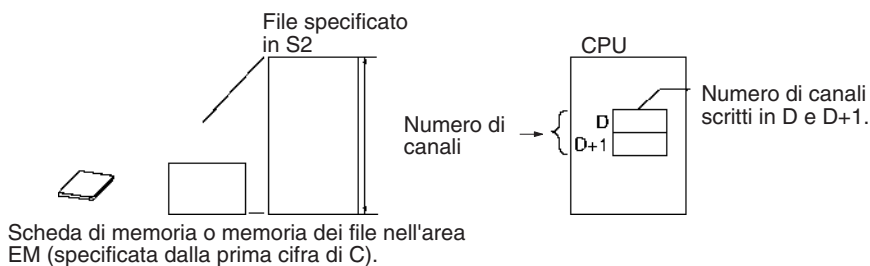


Nota I dati vengono memorizzati nell'ordine basato sugli indirizzi assoluti della memoria interna. Quindi i dati inviati sovrascriveranno i dati dell'area dati successiva se viene superata la capacità dell'area dati specificata in D. Per ulteriori informazioni, consultare *Avvertenze*.

Quando l'istruzione FREAD(700) viene eseguita, il numero di canali (o campi) specificato in S1 e S1+1 viene scritto in A346 e A347 (numero di dati da trasferire) e questo valore viene decrementato di 1 via via che ciascun canale o campo viene trasferito. È possibile controllare il contenuto di questi canali per verificare che sia stato trasferito il numero previsto di canali o campi.

Letture del numero dei canali di dati (terza cifra di C=1)

FREAD(700) trova il numero di canali nel file specificato in S2 (con estensione del nome file .IOM) e scrive il valore esadecimale a 8 cifre in D e D+1.



Caratteristiche operando

| Area | C | S1 | S2 | D |
|--|--|--|--|----------------|
| Area CIO | Da CIO 0000 a CIO6143 | Da CIO 0000 a CIO 6140 | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | Da W000 a W508 | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | Da H000 a 508 | Da H000 a W511 | |
| Area bit ausiliaria | Da A000 a A959 | Da A000 a A444 Da A448 ad A956 | Da A000 a A447 Da A448 a A959 | Da A448 a A959 |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4092 | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4092 | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | Da D00000 a D32764 | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32764 | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32764 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | - | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | - | *Da D00000 a *D32767 *Da E00000 a *E32767 *Da En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | - | | |
| Registri dati | - | | | |
| Registri indice | - | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | |

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se la memoria dei file specificata in C non esiste.</p> <p>ON se le impostazioni in C non sono incluse nell'intervallo specificato.</p> <p>ON se il nome file specificato in S2 non soddisfa le condizioni richieste.</p> <p>ON se il flag di operazione memoria file era ON.</p> <p>ON se non è stata specificata una costante per C (solo CPU CS1 della serie CS precedenti la versione V1□).</p> <p>ON se i dati specificati per S1 sono fuori intervallo (tutte le CPU, ad eccezione delle CPU CS1 della serie CS precedenti la versione V1□).</p> <p>ON se per D è stata specificata un'area non valida.</p> <p>Con le CPU CS1D: ON se non è stato possibile sincronizzare le CPU attiva e in attesa.</p> <p>OFF in tutti gli altri casi.</p> |

Nella tabella seguente sono riportati i flag relativi nell'area ausiliaria.

| Nome | Indirizzo | Operazione |
|--|---------------------|---|
| Tipo di scheda di memoria | Da A34300 ad A34302 | Contiene un numero binario che indica il tipo di scheda di memoria eventualmente installata. (0: Nessuna, 4: ROM flash) |
| Flag di errore formattazione scheda di memoria | A34307 | Viene attivato quando la scheda di memoria non è formattata o si verifica un errore di formattazione. |
| Flag di errore lettura file | A34310 | Viene attivato quando non è possibile leggere un file perché i dati sono danneggiati o il file contiene dati di tipo errato. |
| Flag di file mancante | A34311 | Viene attivato quando non è possibile leggere i dati perché il file specificato non esiste. |
| Flag di operazione memoria file | A34313 | Viene attivato in ciascuno dei seguenti casi: La CPU ha inviato un comando FINS utilizzando CMND(490). È in esecuzione un'istruzione FREAD(700) o FWRT(701). È in corso la sovrascrittura del programma tramite l'utilizzo di un bit di controllo. È in corso un'operazione di backup semplice. |
| Flag di accesso al file | A34314 | Viene attivato quando è in corso l'accesso ai dati del file. Utilizzare questo flag come condizione di esecuzione per evitare che venga eseguita un'istruzione per la memoria dei file mentre un'altra istruzione è in corso di esecuzione. |
| Flag di scheda di memoria rilevata | A34315 | Viene attivato quando viene rilevata una scheda di memoria. |
| Banco di inizio per la formattazione della memoria file nell'area EM | A344 | Contiene il numero del banco di inizio dell'area EM che è stato formattato per essere utilizzato come memoria dei file nell'area EM. Contiene FFFF laddove non è stata formattata nessuna area EM. Per convertire l'area EM da utilizzare come memoria dei file, è necessario impostare su 1 la memoria dei file nell'area EM nella configurazione del PLC e impostare il banco di inizio memoria file nell'area EM (da 0 a C). Tutti i banchi EM compresi tra il primo e l'ultimo banco vengono formattati come memoria per i file. |
| Flag di errore di formattazione della memoria dei file nell'area EM | A34306 | Viene attivato quando si verifica un errore di formattazione del banco iniziale nella memoria dei file nell'area EM. |
| Numero di dati da trasferire | Da A346 ad A347 | Il contenuto di questi canali indica lo stato del trasferimento del file di dati. Quando viene eseguita un'istruzione FREAD(700) o FWRT(701), il numero di canali o campi da trasferire viene scritto in questi canali. Il valore viene decrementato di 1 via via che ciascun canale o campo viene trasferito. A346 contiene i 16 bit più a destra e A347 contiene i 16 bit più a sinistra del valore binario a 32 bit. |

Avvertenze

Durante la normale elaborazione dell'istruzione, FREAD(700) viene utilizzata solo per avviare la lettura della memoria dei file. I tempi di esecuzione di un'istruzione indicati verso la fine di questo manuale sono i tempi necessari per avviare la lettura, non per completarla. La lettura effettiva (trasferimento) viene eseguita dall'elaborazione di accesso al file nella gestione delle periferiche. Pertanto, una volta eseguita l'istruzione FREAD(700), la lettura viene eseguita continuamente perfino se la condizione di esecuzione è OFF nei seguenti cicli. Quando il trasferimento è stato completato, verrà disattivato il flag di operazione memoria file (A34313). È possibile utilizzare questo flag per controllare in modo esclusivo le istruzioni per la memoria dei file.

Il tempo necessario a FREAD(700) per completare il trasferimento dei dati dipende dalla quantità di dati da trasferire, dal tempo di servizio assegnato all'elaborazione di accesso al file e da altre condizioni. Come indicazione di massima, il tempo per trasferire un tempo di ciclo di 10 ms per un file nella directory principale con le impostazioni predefinite per il tempo di servizio sarà di 0,92 s per 1.024 canali e di 4,64 s per 9.999 canali.

Il flag di operazione memoria file (A34313) verrà attivato quando viene eseguita FREAD(700). Se A34313 è già ON, si verificherà un errore e l'istruzione non verrà eseguita.

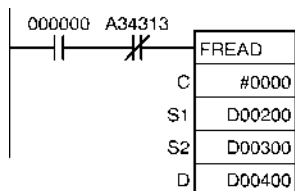
Il flag di errore di lettura file (A34310) verrà attivato e l'istruzione non verrà eseguita se il file specificato contiene il tipo di dati errato o se i dati del file sono danneggiati. Per i file di testo o CSV, il codice carattere deve essere in formato esadecimale e i delimitatori devono essere inseriti ogni 4 cifre per i dati a canale singolo e ogni 8 cifre per i dati a canale doppio. I dati verranno letti finché non verrà rilevato un carattere non valido.

Dopo l'inserimento, sono sufficienti pochi secondi per il riconoscimento della scheda di memoria da parte della CPU. Se si ha intenzione di accedere a una scheda di memoria subito dopo l'attivazione dell'alimentazione o dopo aver inserito una scheda di memoria, utilizzare il flag di scheda di memoria rilevata (A34315) in una condizione di ingresso NA per essere certi che la scheda di memoria sia stata rilevata, come mostrato di seguito.

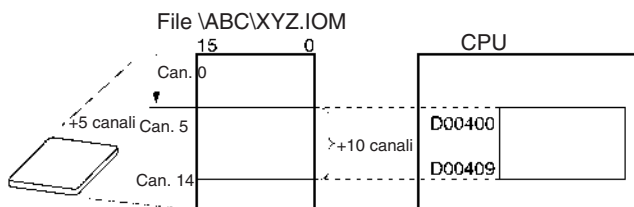


Esempi

Nell'esempio seguente, quando CIO 000000 viene impostato su ON, FREAD(700) legge 10 canali di dati dal file \ABC\XYZ.IOM cominciando dall'inizio del file + 5 canali e invia questi 10 canali a D00400 - D00409.



| | |
|--------------|--|
| C: # 0 0 0 0 | Memoria dei file: scheda di memoria |
| | Funzione: lettura dati |
| S1:D00200 | Numero di canali da leggere: |
| S1+1:D00201 | 10 canali |
| S1+2:D00202 | Canale iniziale: inizio del file +5 canali |
| S1+3:D00203 | |
| S2:D00300 | Nome directory: \ABC |
| S2+1:D00301 | Nome file: XYZ |
| S2+2:D00302 | |
| S2+3:D00303 | |
| S2+4:D00304 | |

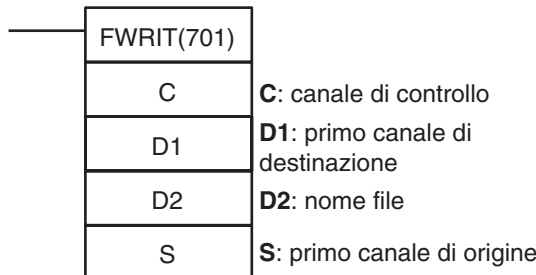


3-26-3 WRITE DATA FILE: FWRIT(701)

Scopo

Sovrascrive o aggiunge i dati specificati ai dati contenuti nel file di dati indicato nella memoria dei file dall'area dati della CPU. Se il file indicato non esiste, viene creato un nuovo file con tale nome. I dati possono essere scritti nel formato binario, testo o CSV.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FWRIT(701) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FWRIT(701) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

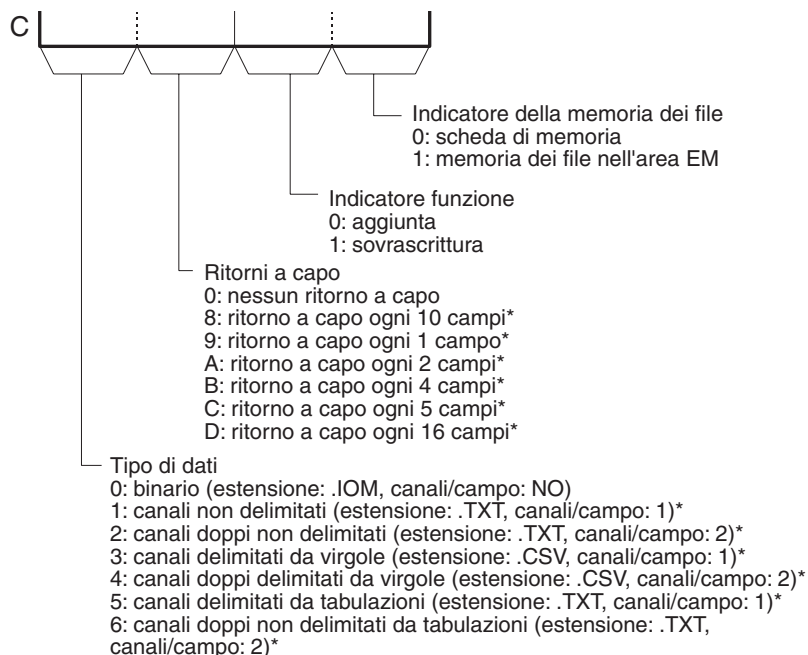
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo

Come mostrato nello schema seguente, la terza cifra del canale di controllo indica se aggiungere o sovrascrivere i dati nel file di dati e la quarta cifra indica se il file di destinazione si trova nella scheda di memoria o nella memoria dei file nell'area EM.



*: Non può essere impostato per le CPU CS1 della serie CS precedenti alla versione V1□.

- Nota**
- Ogni campo conterrà 1 canale di memoria di I/O per il tipo di dati del canale e 2 canali di memoria di I/O per i tipi di dati a doppio canale.
 - Con i canali doppi, il primo canale di dati viene letto dall'indirizzo di memoria superiore, ossia 12345678 verrebbe scritto con 1234 da D00001 e 5678 da D00000.
 - Se è stata specificata la delimitazione, il delimitatore specificato verrà aggiunto dopo ogni canale per il tipo di dati a canale singolo e dopo ogni due canali per il tipo di dati a canale doppio. (Verrà aggiunto il codice di virgola per la delimitazione tramite virgole e il codice di tabulazione per la delimitazione tramite tabulazioni).
 - Se sono stati specificati canali o canali doppi senza limitazione, i dati per tutti i campi verranno scritti in modo continuo senza alcun delimitatore.
 - Se sono stati specificati ritorni a capo, verrà aggiunto un ritorno a capo dopo ciascuna serie del numero specificato di canali. Se non è stato specificato nessun ritorno a capo, i dati verranno scritti in modo continuo senza alcun ritorno a capo.

D1 e D1+1: numero di elementi scritti

Il valore esadecimale a 8 cifre in D1 e D1+1 specifica la quantità di canali o campi da scrivere nella memoria dei file.



| Tipo di dati | Bit da 12 a 15 di C | Contenuto di D1 e D1+1 |
|---------------|---|---|
| Binario | 0 esadecimale (binario) | Numero di canali da scrivere dalla memoria dei file. Da 00000000 a 3FFFFFFF esadecimale |
| Canale | 1 esadecimale (non delimitato), 3 esadecimale (delimitato da virgole) o 5 esadecimale (delimitato da tabulazioni) | Numero di campi da scrivere dalla memoria dei file, ossia il numero di canali da scrivere dalla memoria dei file. Da 00000000 a 1FFFFFFF esadecimale |
| Doppio canale | 2 esadecimale (non delimitato), 4 esadecimale (delimitato da virgole) o 6 esadecimale (delimitato da tabulazioni) | Numero di campi da scrivere dalla memoria dei file, ossia metà del numero di canali da scrivere dalla memoria dei file. Da 00000000 a 0FFFFFFF esadecimale |

D1+2 e D1+3: primo canale di destinazione

Il valore esadecimale a 8 cifre in D1+2 e D1+3 specifica il canale iniziale di scrittura dall'inizio del file.



| Tipo di dati | Bit da 12 a 15 di C | Contenuto di D1+2 e D1+3 |
|---------------|---|---|
| Binario | 0 esadecimale (binario) | Il canale da cui si comincia la scrittura dall'inizio della memoria dei file. Da 00000000 a 3FFFFFFF esadecimale |
| Canale | 1 esadecimale (non delimitato), 3 esadecimale (delimitato da virgole) o 5 esadecimale (delimitato da tabulazioni) | Il campo da cui si comincia la scrittura dall'inizio della memoria dei file, ossia il numero di canali dall'inizio. Da 00000000 a 1FFFFFFF esadecimale |
| Doppio canale | 2 esadecimale (non delimitato), 4 esadecimale (delimitato da virgole) o 6 esadecimale (delimitato da tabulazioni) | Il campo da cui si comincia la scrittura dall'inizio della memoria dei file, ossia metà del numero di canali dall'inizio. Da 00000000 a 0FFFFFFF esadecimale |

- Nota**
- D1+2 e D1+3 sono utilizzati soltanto per sovrascrivere i dati e soltanto per 1) dati di tipo testo e CVS senza ritorni a capo (ovvero bit da 08 a 11 di C

impostati su 0 esadecimale) o 2) per dati binari. Impostare sempre D1+2 e D1+3 su 00000000 esadecimale quando si scrivono dati con ritorni a capo (ovvero bit da 08 a 11 di C impostati su 8 - D esadecimale).

2. D1 - D1+3 devono essere nella stessa area dati.
3. Se il canale iniziale specificato supera il numero di canali nel file di dati, verrà attivato il flag di errore scrittura file (A34308) e i dati del file non verranno scritti.

D2: nome file

D2 è l'indirizzo iniziale dei canali contenenti il percorso assoluto e il nome file in ASCII. Utilizzare ASCII a - z, A - Z e 0 - 9.

Il nome completo del percorso alla directory contenente il file di dati può essere lungo fino a 65 caratteri, inclusa la barra iniziale (ASCII 5C). Il nome del file può contenere fino a 8 caratteri, ma non sono ammessi caratteri nulli (ASCII 00) poiché un carattere nullo viene usato per contrassegnare la fine di ogni stringa di caratteri. Non includere l'estensione del nome file; l'estensione .IOM, .TXT o .CSV verrà aggiunta automaticamente.

| | | | |
|-------|-----|-----|--|
| D2 | F1 | F2 | Memorizzare la stringa di caratteri che comincia con il byte più a sinistra in D2. L'intero nome del percorso e il nome del file può contenere fino a 74 caratteri (byte), incluso il carattere iniziale della barra e il carattere nullo di fine. |
| D2+1 | F3 | F4 | |
| ⋮ | ⋮ | ⋮ | |
| D2+38 | F73 | F74 | |

- Nota**
1. Accertarsi che la stringa di caratteri contenente il nome del percorso e il nome file non superi la fine dell'area dati.
 2. Se la directory specificata non esiste, verrà attivato il flag di file mancante (A34311) e i dati del file non verranno scritti.

Scrivere il nome del percorso e il nome file in ASCII cominciando dal byte più a sinistra di D2, come mostrato nell'esempio seguente per \ABC\XYZ.IOM. (L'estensione viene aggiunta automaticamente).

| | | | | | |
|------|-----|------|--------|----|----|
| D2 | \ * | *A* | D2 | 5C | 41 |
| D2+1 | *B* | *C* | → D2+1 | 42 | 43 |
| D2+2 | \ * | *X* | D2+2 | 5C | 58 |
| D2+3 | *Y* | *Z * | D2+3 | 59 | 5A |
| D2+4 | NUL | | → D2+4 | 00 | |

S: primo canale di origine

S specifica l'indirizzo iniziale contenente i dati che verranno scritti nella memoria dei file. I dati vengono letti dagli indirizzi assoluti della memoria del PLC, quindi l'istruzione FWRIT(701) continuerà a leggere i dati di origine dall'area dati successiva se il numero di canali in corso di lettura supera la fine dell'area dati specificata in S.

Descrizione

Durante la normale elaborazione dell'istruzione, FWRIT(701) viene utilizzata solo per avviare la scrittura della memoria dei file. I tempi di esecuzione di un'istruzione indicati verso la fine di questo manuale sono i tempi necessari per avviare la scrittura, non per completarla. La scrittura effettiva (trasferimento) viene eseguita dall'elaborazione di accesso al file nella gestione delle periferiche. Pertanto, una volta eseguita l'istruzione FWRIT(701), la scrittura viene eseguita continuamente perfino se la condizione di esecuzione è OFF nei seguenti cicli. Quando il trasferimento è stato completato, verrà disattivato il flag di operazione memoria file (A34313). È possibile utilizzare questo flag per controllare in modo esclusivo le istruzioni per la memoria dei file.

Il tempo necessario a FWRIT(701) per completare il trasferimento dei dati dipende dalla quantità di dati da trasferire, dal tempo di servizio assegnato all'elaborazione di accesso al file e da altre condizioni. Come indicazione di massima, il tempo per trasferire un tempo di ciclo di 10 ms per un file nella

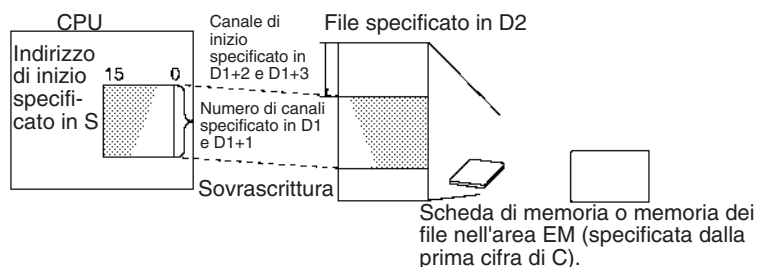
directory principale con le impostazioni predefinite per il tempo di servizio sarà di 1,97 s (nuovo file) o 1,33 s (file esistente) per 1.024 canali e di 6,64 s (nuovo file) o 6,12 s (file esistente) per 9.999 canali.

I dati di origine vengono letti dagli indirizzi assoluti della memoria interna nella RAM, quindi l'intero blocco di dati verrà letto anche se i dati si estendono in due o più aree dati. Ad esempio, se il primo indirizzo di destinazione è nell'area di lavoro ma la quantità di dati supera la capacità di quest'area, FWRIT(701) continuerà a leggere i dati all'inizio dell'area successiva (in questo caso, l'area del temporizzatore). Per una mappa della memoria in cui è indicata la posizione delle aree dati nella RAM, consultare l'Appendice D nel manuale dell'operatore *Controllori programmabili della serie CS/CJ (W339)*.

Quando l'istruzione FWRIT(701) viene eseguita, il numero di canali o campi specificato in D1 e D1+1 viene scritto in A346 e A347 (numero di dati da trasferire) e questo valore viene decrementato di 1 via via che ciascun canale o campo viene trasferito. È possibile controllare il contenuto di questi canali per verificare che sia stato trasferito il numero previsto di canali o campi.

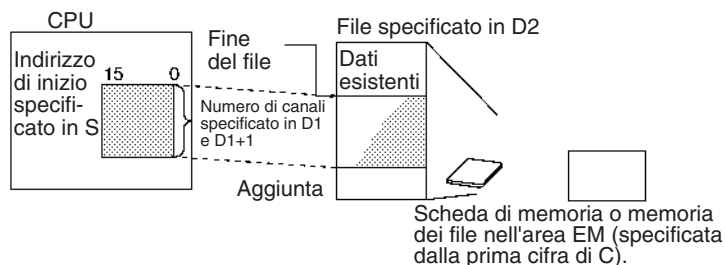
Sovrascrittura dei dati in un file esistente (terza cifra di C=1)

FWRIT(701) utilizza i dati dell'area dati cominciando dal canale specificato in S per sovrascrivere i dati della memoria dei file con il tipo di dati specificato. Sovrascrivere il numero di canali o campi specificato in D1 e D1+1 nel file specificato in D2 (con estensione del nome di file .IOM, .TXT o .CSV), cominciando dall'indirizzo specificato in D1+2 e D1+3.



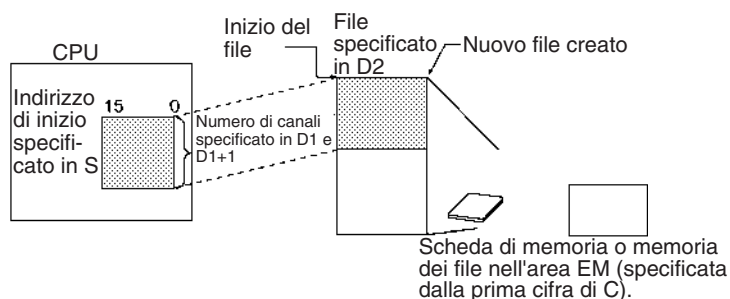
Aggiunta di dati a un file esistente (terza cifra di C=0)

FWRIT(701) aggiunge i dati dell'area dati cominciando dal canale specificato in S a un file di dati nella memoria dei file, nel tipo di dati specificato. Aggiunge il numero di canali o campi specificato in D1 e D1+1 nel file specificato in D2 (con estensione del nome di file .IOM, .TXT o .CSV), cominciando dall'indirizzo specificato in D1+2 e D1+3.



Creazione di un nuovo file con i dati di origine

Se il file specificato in D2 non esiste, l'istruzione FWRIT(701) crea un nuovo file con il nome e l'estensione indicati dall'utente (.IOM, .TXT o .CVS) e scrive i dati di origine specificati nel tipo di dati specificato cominciando dall'inizio del file. In questo caso, non importa se viene specificata un'operazione di aggiunta o di sovrascrittura dei dati.



Caratteristiche operando

| Area | C | D1 | D2 | S |
|--|--|--|--|---|
| Area CIO | Da CIO 0000 a CIO 6143 | Da CIO 0000 a CIO 6140 | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | Da W000 a W508 | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | Da H000 a 508 | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | Da A000 a A444 Da A448 ad A956 | Da A000 a A447 Da A448 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | Da T0000 a T4092 | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4092 | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | Da D00000 a D32764 | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | Da E00000 a E32764 | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | Da En_00000 a En_32764 (n = da 0 a C) | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | – | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | – | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Solo valori specificati | – | | |
| Registri dati | – | | | |
| Registri indice | – | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da –2048 a +2047, da ,IR0 a da –2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | |

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se il tipo di memoria dei file specificato in C non esiste.</p> <p>ON se le impostazioni in C non sono incluse nell'intervallo specificato.</p> <p>ON se il nome file specificato in D2 non soddisfa le condizioni richieste.</p> <p>ON se il flag di operazione memoria file era ON.</p> <p>ON se non è stata specificata una costante per C (solo CPU CS1 della serie CS precedenti la versione V1).</p> <p>ON se i dati specificati per D1 sono fuori intervallo (tutte le CPU, ad eccezione delle CPU CS1 della serie CS precedenti la versione V1).</p> <p>ON se per S è stata specificata un'area non valida.</p> <p>Con le CPU CS1D: ON se non è stato possibile sincronizzare le CPU attiva e in attesa.</p> <p>OFF in tutti gli altri casi.</p> |

Nella tabella seguente sono riportati i flag relativi nell'area ausiliaria.

| Nome | Indirizzo | Operazione |
|--|---------------------|---|
| Tipo di scheda di memoria | Da A34300 ad A34302 | Contiene un numero binario che indica il tipo di scheda di memoria eventualmente installata. (0: Nessuna, 4: ROM flash) |
| Flag di errore formattazione scheda di memoria | A34307 | Viene attivato quando la scheda di memoria non è formattata o si verifica un errore di formattazione. |
| Flag di errore scrittura file | A34308 | Viene attivato quando si verifica un errore durante la scrittura nel file. |
| Flag di scrittura file non possibile | A34309 | Viene attivato quando non è possibile scrivere i dati perché il file è protetto da scrittura o la quantità di memoria disponibile è insufficiente. |
| Flag di file non trovato | A34311 | ON quando la directory specificata non esiste durante la scrittura di un file. |
| Flag di operazione memoria file | A34313 | Viene attivato in ciascuno dei seguenti casi: La CPU ha inviato un comando FINS utilizzando CMND(490). È in esecuzione un'istruzione FREAD(700) o FWRT(701). È in corso la sovrascrittura del programma tramite l'utilizzo di un bit di controllo. È in corso un'operazione di backup semplice. |
| Flag di accesso al file | A34314 | Viene attivato quando è in corso l'accesso ai dati del file. Utilizzare questo flag come condizione di esecuzione per evitare che venga eseguita un'istruzione per la memoria dei file mentre un'altra istruzione è in corso di esecuzione. |
| Flag di scheda di memoria rilevata | A34315 | Viene attivato quando viene rilevata una scheda di memoria. |
| Banco di inizio per la formattazione della memoria file nell'area EM | A344 | Contiene il numero del banco di inizio dell'area EM che è stato formattato per essere utilizzato come memoria dei file nell'area EM. Contiene FFFF laddove non è stata formattata nessuna area EM. Per convertire l'area EM da utilizzare come memoria dei file, è necessario impostare su 1 la memoria dei file nell'area EM nella configurazione del PLC e impostare il banco di inizio memoria file nell'area EM (da 0 a C). Tutti i banchi EM compresi tra il primo e l'ultimo banco vengono formattati come memoria per i file. |

| Nome | Indirizzo | Operazione |
|---|-----------------|---|
| Flag di errore di formattazione della memoria dei file nell'area EM | A34306 | Viene attivato quando si verifica un errore di formattazione del banco iniziale nella memoria dei file nell'area EM. |
| Numero di dati da trasferire | Da A346 ad A347 | Il contenuto di questi canali indica lo stato del trasferimento del file di dati. Quando viene eseguita un'istruzione FWRIT(701), il numero di canali o campi da trasferire viene scritto in questi canali. Il valore viene decrementato di 1 via via che ciascun canale viene trasferito. A346 contiene i 16 bit più a destra e A347 contiene i 16 bit più a sinistra del valore binario a 32 bit. |

Avvertenze

Il flag di operazione memoria file (A34313) verrà attivato quando viene eseguita FWRIT(701). Se A34313 è già ON, si verificherà un errore e l'istruzione non verrà eseguita.

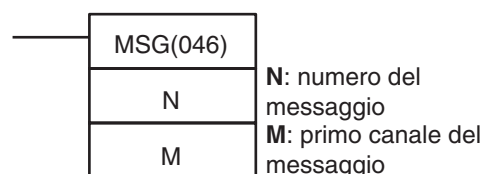
Verrà attivato il flag di scrittura file non possibile (A34309) e l'istruzione non verrà eseguita se non è stato possibile scrivere i dati poiché il file era protetto da scrittura oppure perché la memoria disponibile era insufficiente.

Il flag di errore di scrittura file (A34308) verrà attivato e l'istruzione non verrà eseguita se il file specificato non contiene il tipo di dati corretto o se i dati del file sono danneggiati.

Dopo l'inserimento, sono sufficienti pochi secondi per il riconoscimento della scheda di memoria da parte della CPU. Se si ha intenzione di accedere a una scheda di memoria subito dopo l'attivazione dell'alimentazione o dopo aver inserito una scheda di memoria, utilizzare il flag di scheda di memoria rilevata (A34315) in una condizione di ingresso NA per essere certi che la scheda di memoria sia stata rilevata, come mostrato di seguito.

**3-27 Istruzioni di visualizzazione: DISPLAY MESSAGE: MSG(046)****Scopo**

Legge i sedici canali di codice ASCII esteso specificati e visualizza il messaggio su un dispositivo periferico quale una console di programmazione.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | MSG(046) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MSG(046) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi**N: numero del messaggio**

Il numero del messaggio deve essere compreso tra 0000 e 0007 esadecimale (o da 0 a 7 decimale).

M: primo canale del messaggio

Quando si visualizza un messaggio, M specifica l'indirizzo del primo dei canali contenente il messaggio ASCII. Quando si cancella un messaggio, M può essere qualsiasi costante esadecimale (da 0000 a FFFF).

Caratteristiche operando

| Area | N | M |
|--|--|----------------------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 a A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da #0000 a #0007 (binario) o da &0 a &7 | Da #0000 a #FFFF (binario) |
| Registri dati | Da DR0 a DR15 | --- |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

Quando la condizione di esecuzione è ON, MSG(046) registra i 16 canali di dati ASCII (fino a 32 caratteri incluso il carattere nullo) da M - M+15 per il numero di messaggi specificato da N. Una volta che il messaggio è stato registrato, è possibile collegare una console di programmazione e il messaggio verrà visualizzato dopo ogni messaggio di errore generato.

Dopo che un messaggio è stato registrato, è possibile modificare la visualizzazione del messaggio scrivendo il messaggio nell'area di memorizzazione dei messaggi.

Per cancellare un messaggio che è stato registrato, eseguire MSG(046) con S impostato sul numero di messaggio del messaggio che si desidera eliminare e N impostato su una costante (da 0000 a FFFF).

Un messaggio registrato durante l'esecuzione del programma verrà mantenuto anche se tale esecuzione viene interrotta ma tutti i messaggi verranno cancellati quando il programma viene eseguito di nuovo.

Nota Per una tabella dei caratteri ASCII estesi, consultare l'Appendice A nel manuale *CS/CJ-series Programming Consoles Operation Manual (W341)*.

Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|--|
| Flag di errore | ER | ON se il contenuto di S non è 0000 – 0007 esadecimale. OFF in tutti gli altri casi. |

Avvertenze

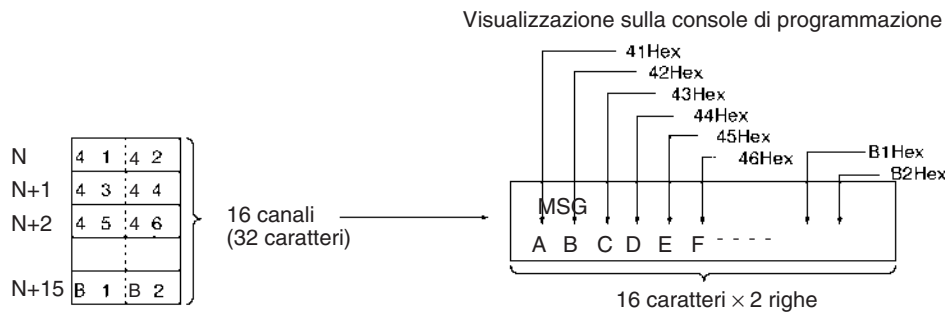
I messaggi registrati vengono aggiornati ogni volta che viene eseguita l'istruzione MSG(046).

Tutti i caratteri del messaggio che seguono il carattere nullo (00) vengono convertiti in spazi sul display della console di programmazione.

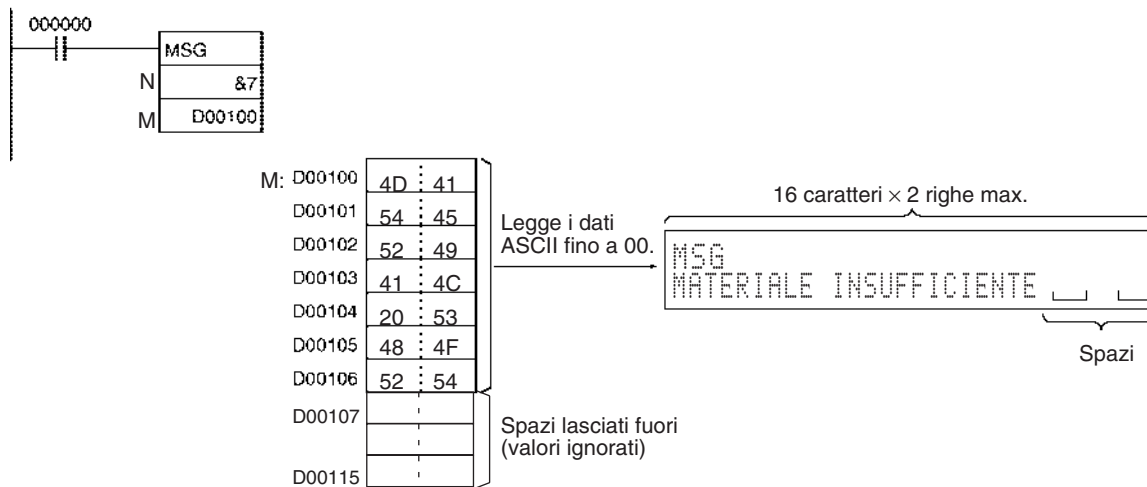
Il carattere memorizzato nel byte più a sinistra viene visualizzato prima del carattere nel byte più a destra.

Esempi

Nello schema seguente viene mostrato il modo in cui i 16 canali di dati esadecimali vengono convertiti in un messaggio visualizzato sulla console di programmazione.



Nell'esempio seguente, quando CIO 000000 è impostato su ON, i 16 canali di dati in D00100 - D00115 vengono letti come 32 caratteri di dati ASCII per il numero di messaggio 7 e visualizzati sulla periferica.



ASCII

| | | Quattro bit più a sinistra | | | | | | | | | | | | | | | |
|--------------------------|---|----------------------------|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Quattro bit più a destra | 0 | | | Sp | 0 | @ | P | ' | p | | | | | 一 | タ | ミ | |
| | 1 | | | ! | 1 | A | Q | a | q | | | | | 。 | ア | チ | ム |
| | 2 | | | " | 2 | B | R | b | r | | | | | 「 | イ | ツ | メ |
| | 3 | | | # | 3 | C | S | c | s | | | | | 」 | ウ | テ | モ |
| | 4 | | | \$ | 4 | D | T | d | t | | | | | 、 | エ | ト | ヤ |
| | 5 | | | % | 5 | E | U | e | u | | | | | ・ | オ | ナ | ユ |
| | 6 | | | & | 6 | F | V | f | v | | | | | ヲ | カ | ニ | ヨ |
| | 7 | | | ' | 7 | G | W | g | w | | | | | ア | キ | ヌ | ラ |
| | 8 | | | (| 8 | H | X | h | x | | | | | イ | ク | ネ | リ |
| | 9 | | |) | 9 | I | Y | i | y | | | | | ウ | ケ | ノ | ル |
| | A | | | * | : | J | Z | j | z | | | | | エ | コ | ハ | レ |
| | B | | | + | ; | K | [| k | { | | | | | オ | サ | ヒ | ロ |
| | C | | | , | < | L | ¥ | l | | | | | | ヤ | シ | フ | ワ |
| | D | | | - | = | M |] | m | } | | | | | ユ | ス | ヘ | ン |
| | E | | | . | > | N | ^ | n | ~ | | | | | ヨ | セ | ホ | ° |
| | F | | | / | ? | O | _ | o | | | | | | ッ | ソ | マ | ° |

3-28 Istruzioni per l'orologio

In questa sezione vengono descritte le istruzioni utilizzate per l'orologio di sistema.

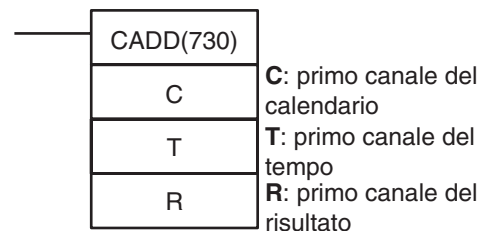
| Istruzione | Mnemonico | Codice funzione | Pagina |
|-------------------|-----------|-----------------|--------|
| CALENDAR ADD | CADD | 730 | 1061 |
| CALENDAR SUBTRACT | CSUB | 731 | 1065 |
| HOURS TO SECONDS | SEC | 065 | 1068 |
| SECONDS TO HOURS | HMS | 066 | 1070 |
| CLOCK ADJUSTMENT | DATE | 735 | 1073 |

3-28-1 CALENDAR ADD: CADD(730)

Scopo

Aggiunge tempo ai dati del calendario nei canali specificati.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | CADD(730) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CADD(730) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

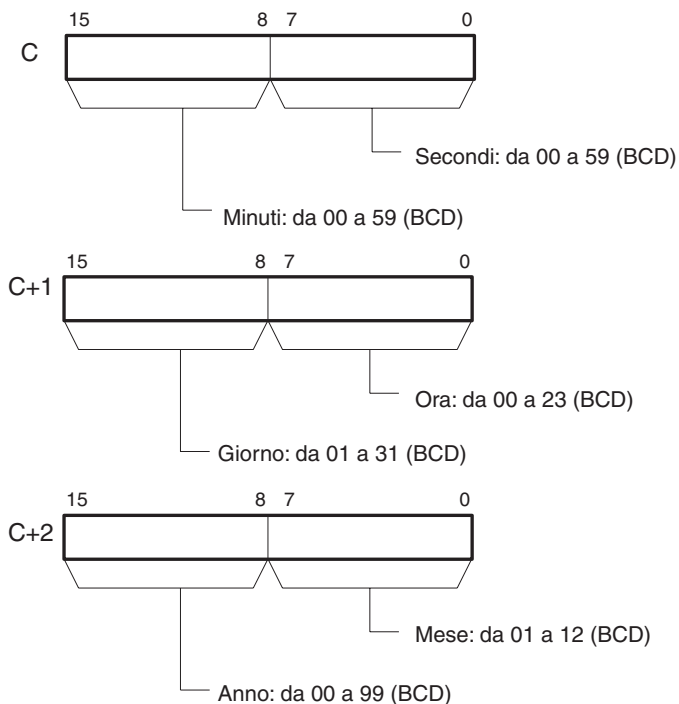
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

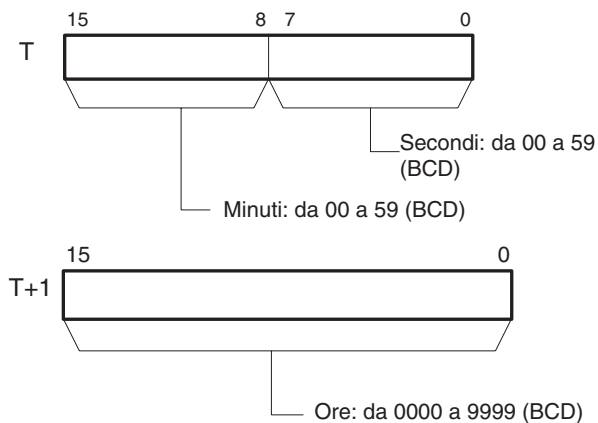
Da C a C+2: dati del calendario

Impostare i dati del calendario in C - C+2, come indicato nello schema seguente.

C - C+2 devono essere nella stessa area dati.

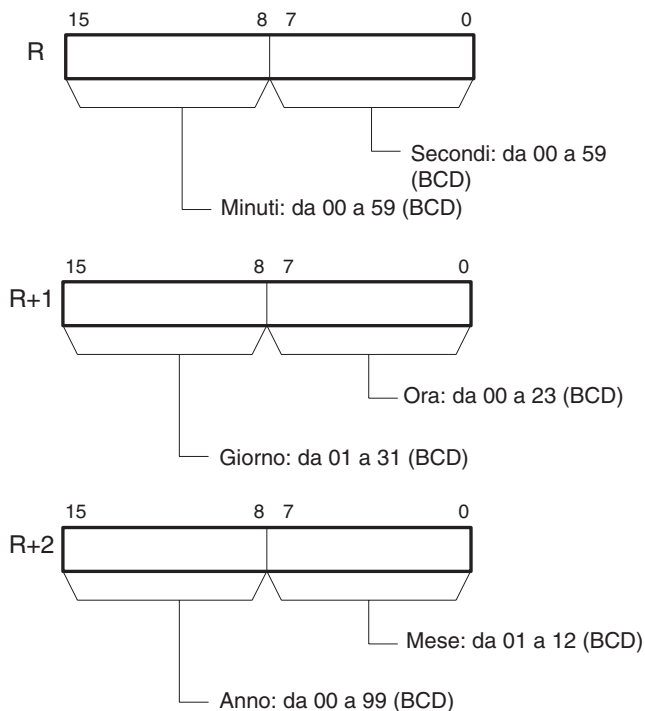
**T e T+1: dati temporali**

Impostare i dati relativi all'orario in T e T+1, come indicato nello schema seguente. T e T+1 devono essere nella stessa area dati.



Da R a R+2: dati del risultato

R - R+2 contengono il risultato dell'addizione. R - R+2 devono essere nella stessa area dati.

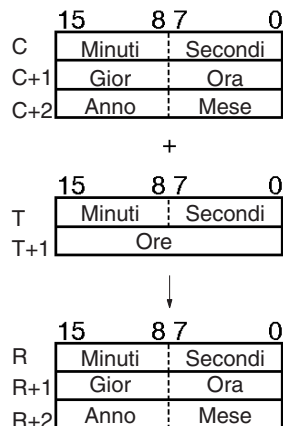
**Caratteristiche operando**

| Area | C | T | R |
|--------------------------------------|--|--|--|
| Area CIO | Da CIO 0000 a CIO 6141 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6141 |
| Area di lavoro | Da W000 a W509 | Da W000 a W510 | Da W000 a W509 |
| Area bit di ritenività | Da H000 a H509 | Da H000 a H510 | Da H000 a H509 |
| Area bit ausiliaria | Da A000 ad A957 | Da A000 ad A958 | Da A448 ad A957 |
| Area del temporizzatore | Da T0000 a T4093 | Da T0000 a T4094 | Da T0000 a T4093 |
| Area del contatore | Da C0000 a C4093 | Da C0000 a C4094 | Da C0000 a C4093 |
| Area DM | Da D00000 a D32765 | Da D00000 a D32766 | Da D00000 a D32765 |
| Area EM senza banco | Da E00000 a E32765 | Da E00000 a E32766 | Da E00000 a E32765 |
| Area EM con banco | Da En_00000 a En_32765 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | Da En_00000 a 3En_2765 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | --- |
| Registri dati | --- | | |

| Area | C | T | R |
|--|--|---|---|
| Registri indice | - | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR005+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

CADD(730) aggiunge i dati relativi al calendario (canali da C a C+2) ai dati temporali (canali T e T+1) e invia i dati risultanti relativi al calendario a R - R+2.

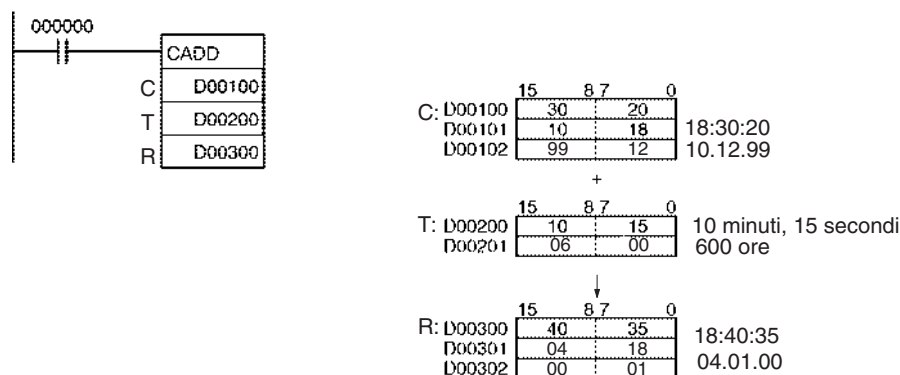


Flag

| Nome | Etichetta | Operazione |
|----------------|-----------|---|
| Flag di errore | ER | ON se i dati del calendario compresi tra C e C+2 non sono inclusi negli intervalli specificati. ON se i dati temporali in T e T+1 non sono inclusi negli intervalli specificati. OFF in tutti gli altri casi. |

Esempi

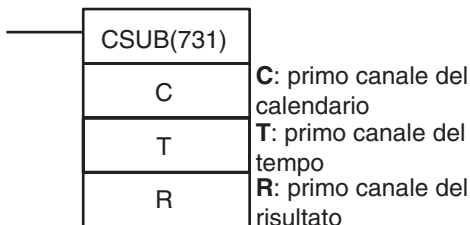
Nell'esempio seguente, quando CIO 000000 è impostato su ON, i dati relativi al calendario in D00100 - D00102 (anno, mese, giorno, ore, minuti, secondi) vengono aggiunti ai dati temporali in D00200 e D00201 (ore, minuti, secondi) e il risultato viene inviato a D00300 - D00302.



3-28-2 CALENDAR SUBTRACT: CSUB(731)

Scopo Sottrae tempo dai dati del calendario nei canali specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CSUB(731) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CSUB(731) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

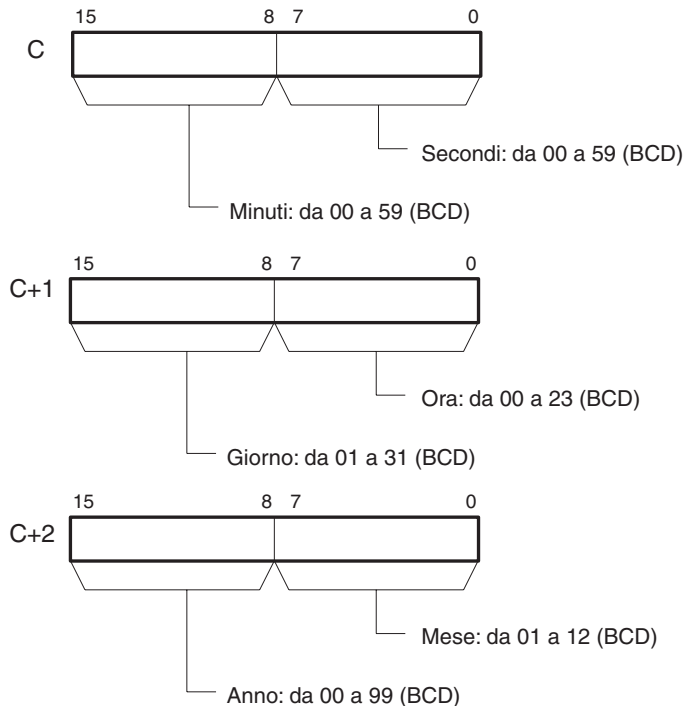
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Da C a C+2: dati del calendario

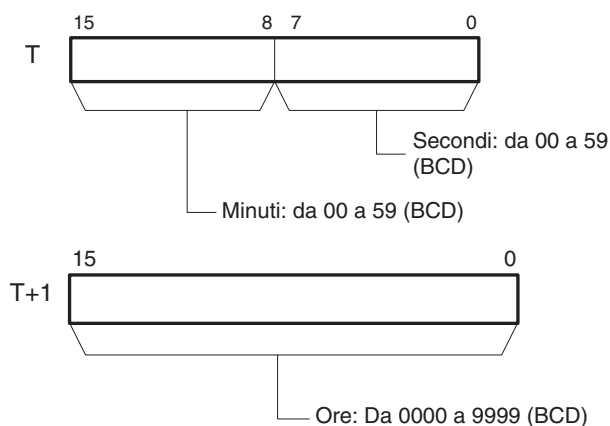
Impostare i dati del calendario in C - C+2, come indicato nello schema seguente.

C - C+2 devono essere nella stessa area dati.

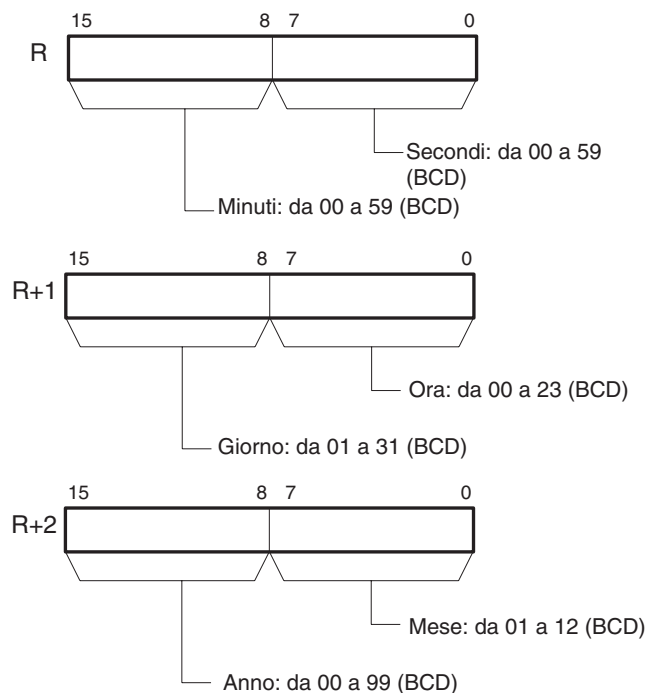


T e T+1: dati relativi all'orario

Impostare i dati relativi all'orario in T e T+1, come indicato dal seguente diagramma. T e T+1 devono essere nella stessa area dati.

**Da R a R+2: dati relativi al risultato**

I risultati dell'addizione sono nell'intervallo compreso tra R ed R+2. R ed R+2 devono essere nella stessa area dati.

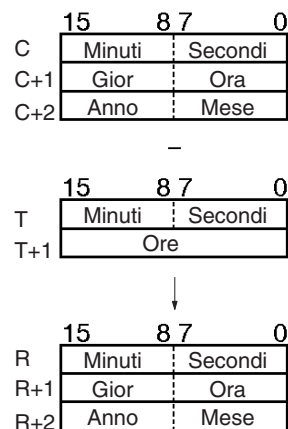
**Caratteristiche operando**

| Area | C | T | R |
|-------------------------|------------------------|------------------------|------------------------|
| Area CIO | Da CIO 0000 a CIO 6141 | Da CIO 0000 a CIO 6142 | Da CIO 0000 a CIO 6141 |
| Area di lavoro | Da W000 a W509 | Da W000 a W510 | Da W000 a W509 |
| Area bit di ritentività | Da H000 a H509 | Da H000 a H510 | Da H000 a H509 |
| Area bit ausiliaria | Da A000 ad A957 | Da A000 ad A958 | Da A448 ad A957 |
| Area del temporizzatore | Da T0000 a T4093 | Da T0000 a T4094 | Da T0000 a T4093 |
| Area del contatore | Da C0000 a C4093 | Da C0000 a C4094 | Da C0000 a C4093 |
| Area DM | Da D00000 a D32765 | Da D00000 a D32766 | Da D00000 a D32765 |

| Area | C | T | R |
|--|--|--|--|
| Area EM senza banco | Da E00000 a E32765 | Da E00000 a E32766 | Da E00000 a E32765 |
| Area EM con banco | En_00000 ... En_32765 (n = da 0 a C) | Da En_00000 a En_32766 (n = da 0 a C) | En_00000 ... 3En_2765 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Solo valori specificati | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR005+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

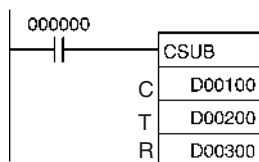
CSUB(731) sottrae i dati relativi all'orario (canali T e T+1) dai dati del calendario (canali da C a C+2) e invia il dati risultanti relativi al calendario nell'intervallo compreso tra R ed R+2.

**Flag**

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se i dati del calendario compresi tra C e C+2 non sono inclusi negli intervalli specificati. ON se i dati dell'orario compresi tra T e T+1 non sono inclusi negli intervalli specificati. OFF in tutti gli altri casi. |

Esempi

Nell'esempio che segue, quando CIO 000000 passa a ON, i dati relativi all'orario in D00200 e D00201 (ore, minuti, secondi) sono sottratti dai dati del calendario compresi tra D00100 e D00102 (anno, mese, giorno, ora, minuti, secondi) e il risultato viene inviato nell'intervallo da D00300 a D00302.



| | | | | | |
|-----------|----|---|----|---|----------------|
| | 15 | 8 | 7 | 0 | |
| C: D00100 | 30 | : | 20 | | 18:30:20 |
| D00101 | 10 | : | 18 | | 10 luglio 1998 |
| D00102 | 98 | : | 07 | | |

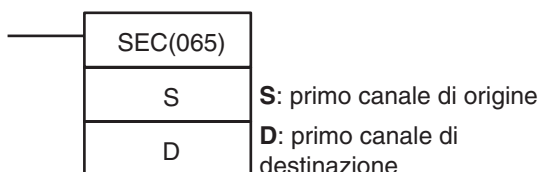
| | | | | | |
|-----------|----|---|----|---|-------------------------------|
| | 15 | 8 | 7 | 0 | |
| T: D00200 | 10 | : | 15 | | 50 ore, 10 minuti, 15 secondi |
| D00201 | 00 | : | 50 | | |

| | | | | | |
|-----------|----|---|----|---|---------------|
| | 15 | 8 | 7 | 0 | |
| R: D00300 | 20 | : | 05 | | 16:20:05 |
| D00301 | 08 | : | 16 | | 8 luglio 1998 |
| D00302 | 98 | : | 07 | | |

3-28-3 HOURS TO SECONDS: SEC(065)

Scopo Convertire i dati temporali in formato ore/minuti/secondi in un tempo equivalente espresso solo in secondi.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | SEC(065) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @SEC(065) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

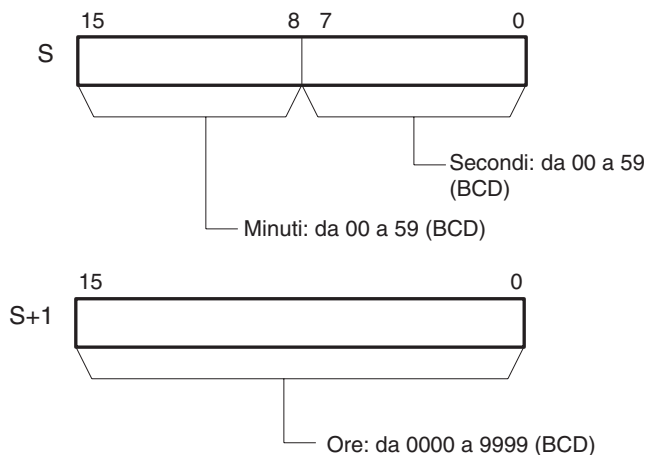
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

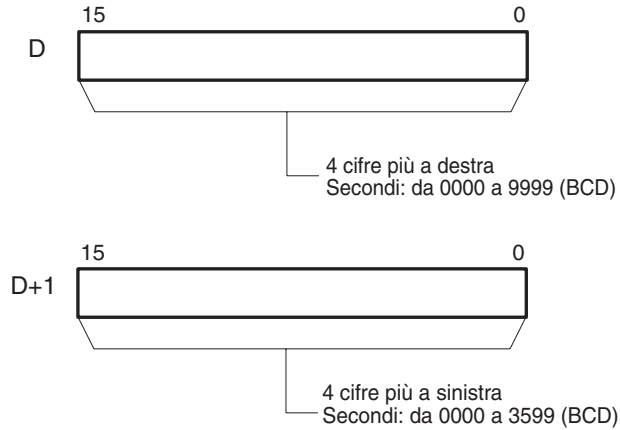
S ed S+1: dati di origine

Impostare i dati di origine espressi in ore/minuti/secondi in S ed S+1, come indicato dal seguente diagramma. S ed S+1 devono trovarsi nella stessa area dati.



D e D+1: dati relativi al risultato

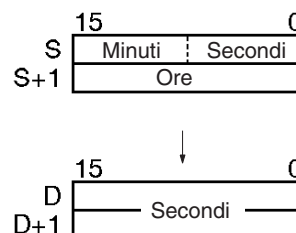
D e D+1 contengono i dati del risultato espressi esclusivamente in secondi. D e D+1 devono essere nella stessa area dati.

**Caratteristiche operando**

| Area | S | D |
|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Solo valori specificati | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

SEC(065) converte i dati BCD a 8 cifre con formato ore/minuti/secondi in S ed S+1 in dati BCD a 8 cifre espressi solo in secondi e scrive il risultato in D e in D+1.

**Flag**

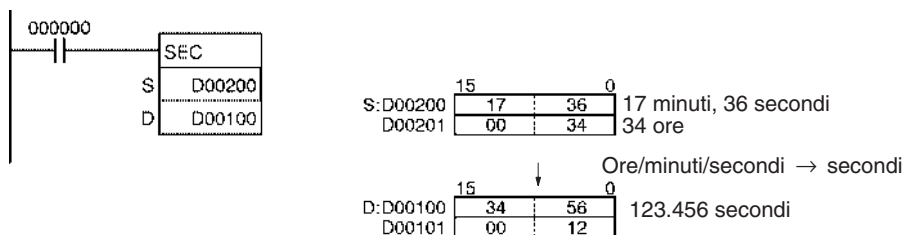
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i dati in minuti in S (bit da 08 a 15) non sono BCD e nell'intervallo compreso tra 00 e 59. ON se i dati in secondi in S (bit da 00 a 07) non sono BCD e nell'intervallo compreso tra 00 e 59. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il contenuto di D è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |

Avvertenze

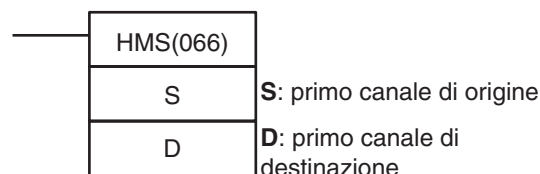
Il valore massimo per i dati di origine è pari a 9.999 ore, 59 minuti e 59 secondi (35.999.999 secondi).

Esempi

Nell'esempio che segue, quando CIO 000000 passa a ON, i dati con formato ore/minuti/secondi in D00200 e D00201 (34 ore, 17 minuti e 36 secondi) sono convertiti in dati espressi esclusivamente in secondi e il risultato viene inviato a D00100 e D00101.

**3-28-4 SECONDS TO HOURS: HMS(066)****Scopo**

Converte i dati temporali espressi in secondi in un tempo equivalente in formato ore/minuti/secondi.

Simbolo programmazione ladder**Variazioni**

| Variazioni | Eseguita a ogni ciclo con condizione ON | HMS(066) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @HMS(066) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

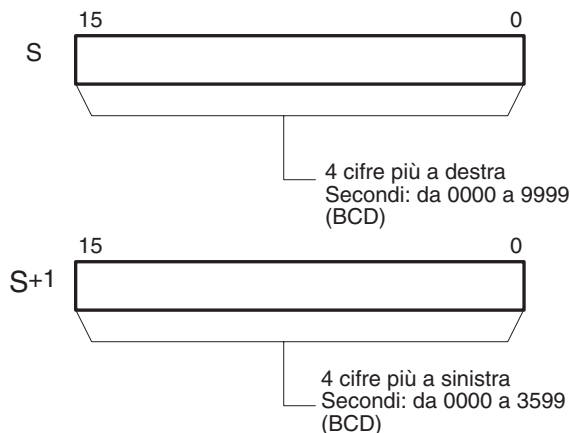
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S ed S+1: dati di origine

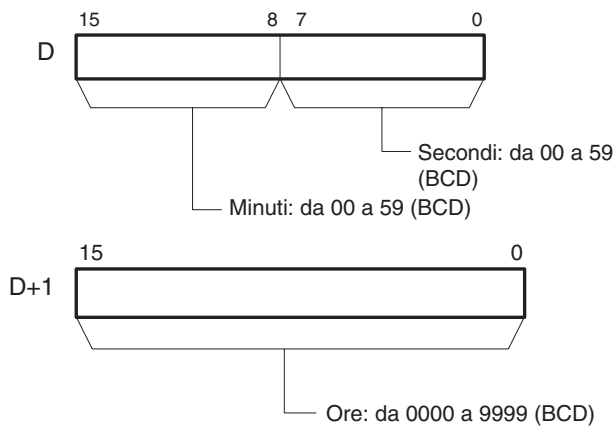
Impostare i dati di origine in secondi in S ed S+1, come indicato dal seguente diagramma.

S ed S+1 devono essere nella stessa area dati.



D e D+1: dati relativi al risultato

D e D+1 contengono i dati del risultato in formato ore/minuti/secondi. D e D+1 devono essere nella stessa area dati.

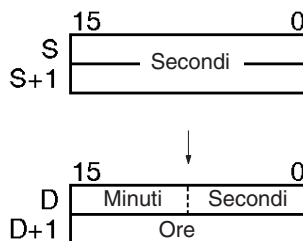


| Area | S | D |
|--------------------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6142 | |
| Area di lavoro | Da W000 a W510 | |
| Area bit di ritenività | Da H000 a H510 | |
| Area bit ausiliaria | Da A000 ad A958 | Da A448 ad A958 |
| Area del temporizzatore | Da T0000 a T4094 | |
| Area del contatore | Da C0000 a C4094 | |
| Area DM | Da D00000 a D32766 | |
| Area EM senza banco | Da E00000 a E32766 | |
| Area EM con banco | Da En_00000 a En_32766 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |

| Area | S | D |
|--|---|-----|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Da 00000000 a 35999999 (BCD) | --- |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

HMS(066) converte i dati BCD a 8 cifre espressi solo in secondi in S ed S+1 in dati BCD a 8 cifre con formato ore/minuti/secondi e invia il risultato in D e in D+1.



Flag

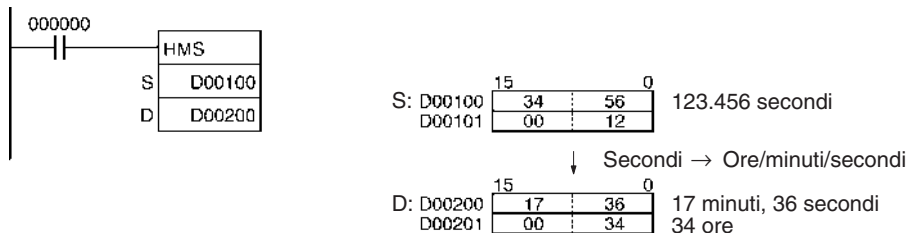
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i dati in secondi in S ed S+1 non sono BCD e nell'intervallo compreso tra 0 e 35.999.999. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il contenuto di D è 0000 dopo l'esecuzione. OFF in tutti gli altri casi. |

Avvertenze

Il valore massimo per i dati di origine è pari a 35.999.999 secondi (9.999 ore, 59 minuti e 59 secondi).

Esempi

Nell'esempio che segue, quando CIO 000000 passa a ON, i dati espressi in secondi in D00100 e D00101 (123.456 secondi) sono convertiti nel formato ore/minuti/secondi e il risultato viene inviato a D00200 e D00201.

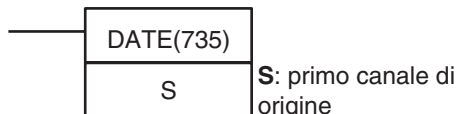


3-28-5 CLOCK ADJUSTMENT: DATE(735)

Scopo Modifica l'impostazione dell'orologio interno in base all'impostazione dei canali di origine specificati.

Nota È inoltre possibile modificare l'impostazione dell'orologio interno da un dispositivo periferico o tramite il comando CLOCK WRITE FINS (0702).

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | DATE(735) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @DATE(735) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

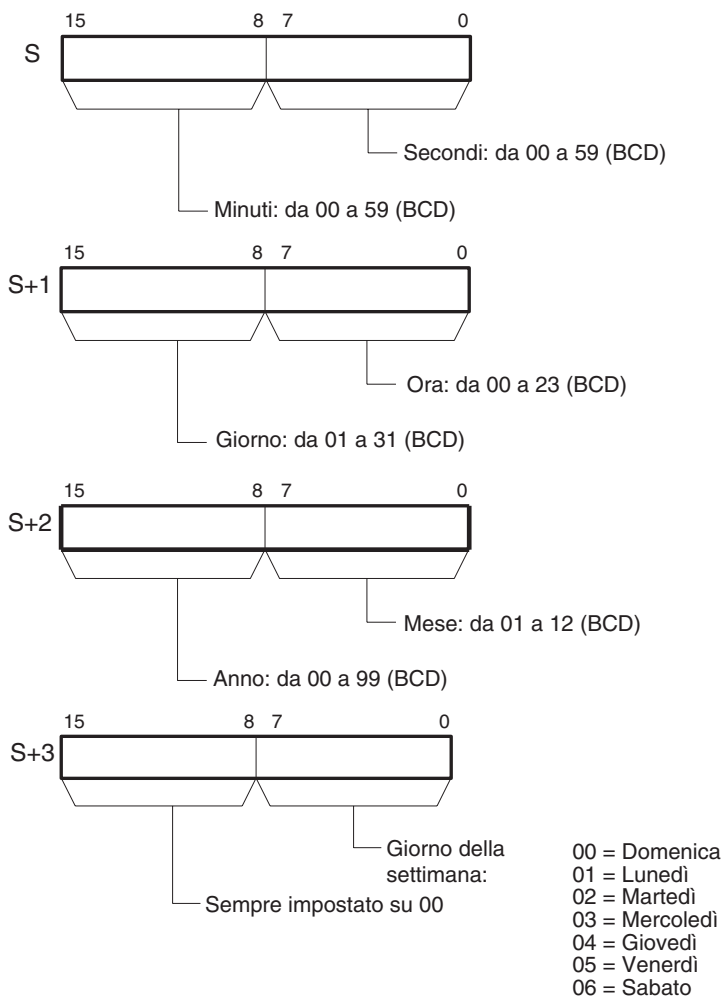
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Da S a S+3: nuova impostazione dell'orologio

Definire la nuova impostazione dell'orologio da S a S+3, come indicato dal seguente diagramma.

S - S+3 devono essere tutti nella stessa area dati.



Nella tabella seguente viene illustrata la struttura dell'area dell'orologio calendario.

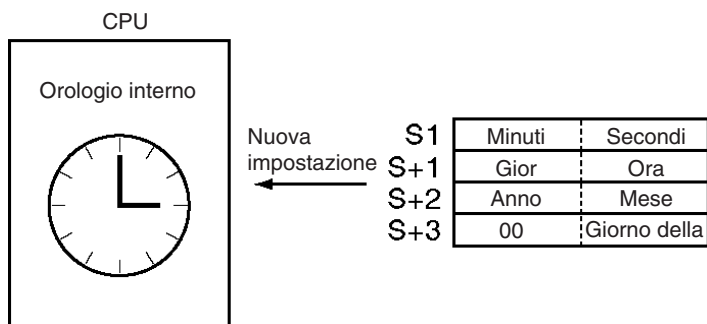
| Indirizzi | Contenuto |
|---------------------|---|
| Da A35100 ad A35107 | Secondo (da 00 a 59, BCD) |
| Da A35108 ad A35115 | Minuto (da 00 a 59, BCD) |
| Da A35200 ad A35207 | Ora (da 00 a 23, BCD) |
| Da A35208 ad A35215 | Giorno del mese (da 01 a 31, BCD) |
| Da A35300 ad A35307 | Mese (da 01 a 12, BCD) |
| Da A35308 ad A35315 | Anno (da 00 a 99, BCD) |
| Da A35400 ad A35407 | Giorno della settimana (da 00 a 06 = da domenica a sabato, esadecimale) |
| Da A35408 ad A35415 | Sempre impostato su 00 |

Caratteristiche operando

| Area | S |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6140 |
| Area di lavoro | Da W000 a W508 |
| Area bit di ritentività | Da H000 a H508 |
| Area bit ausiliaria | Da A000 a A956 |
| Area del temporizzatore | Da T0000 a T4092 |
| Area del contatore | Da C0000 a C4092 |
| Area DM | Da D00000 a D32764 |
| Area EM senza banco | Da E00000 a E32764 |
| Area EM con banco | Da En_00000 a En_32764 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

DATE(735) modifica l'impostazione dell'orologio interno in base ai dati corrispondenti specificati nei quattro canali di origine. La nuova impostazione dell'orologio viene immediatamente aggiornata nell'area dell'orologio calendario (da A351 ad A354).



Flag

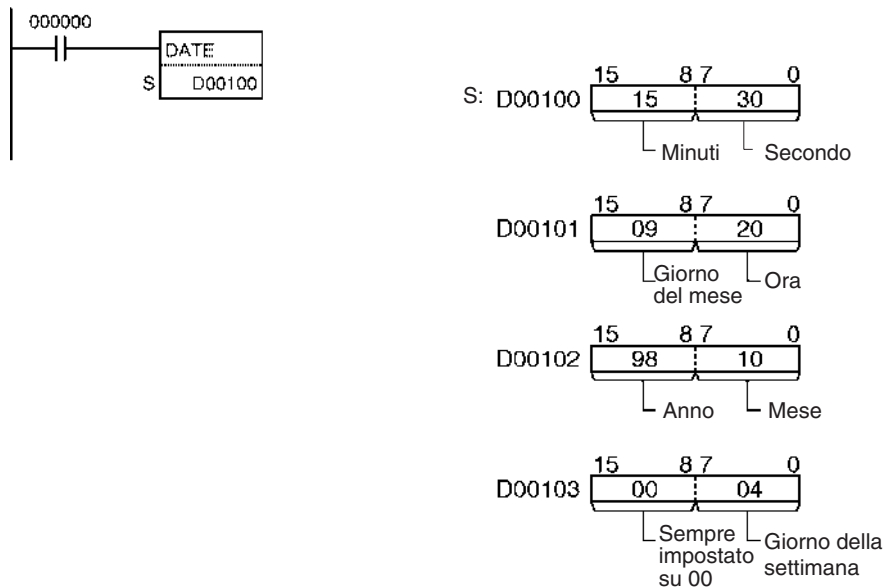
| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se la nuova impostazione dell'orologio da S a S+3 non è nell'intervallo specificato. OFF in tutti gli altri casi. |

Avvertenze

Anche se l'orologio interno viene impostato su una data inesistente (ad esempio, 31 novembre) non si verificherà alcun errore.

Esempi

Nell'esempio seguente, quando CIO 000000 passa a ON, l'orologio interno viene impostato sulle 20:15:30 di martedì 9 ottobre 1998.



3-29 Istruzioni di debug

3-29-1 Campionamento Trace Memory: TRSM(045)

Scopo

Quando viene eseguita l'istruzione TRSM(045), lo stato del bit o del canale preselezionato viene campionato e memorizzato nella memoria di registrazione. È possibile utilizzare TRSM(045) in qualsiasi punto del programma e per un numero di volte illimitato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo | TRSM(045) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Le operazioni di registrazione dei dati tramite un dispositivo periferico sono sintetizzate nell'elenco qui di seguito.

- 1,2,3...**
1. Utilizzando il dispositivo periferico, impostare i parametri indicati.
 - a) Impostare l'indirizzo del bit o canale da registrare.
 - b) Impostare la condizione di avvio. Il controllo della validità dei dati memorizzati nella memoria di registrazione può avvenire quando si verifica una delle seguenti condizioni:
 - i) Il bit di avvio registrazione passa da OFF a ON.
 - ii) Un determinato bit passa da OFF a ON.
 - iii) Il valore di un canale specificato corrisponde al valore impostato.
 - c) Per eseguire il campionamento all'esecuzione dell'istruzione TRSM(045), nel programma impostare l'intervallo di campionamento su "TRSM".
 - d) Impostare il ritardo.
 2. Quando si modifica l'impostazione del bit di avvio campionamento da OFF a ON tramite il dispositivo periferico, i dati specificati cominciano a essere campionati a ciascuna esecuzione di TRSM(045) e i dati campionati vengono memorizzati nella memoria di registrazione. Allo stesso tempo viene attivato il flag di esecuzione registrazione (A50813).
 3. Una volta che la condizione di avvio è soddisfatta, cioè se il bit di avvio registrazione e il bit specificato sono ON oppure se il valore del canale specificato corrisponde al valore impostato, i dati campionati saranno validi a cominciare dal campione successivo più o meno il numero di campioni definiti mediante l'impostazione del ritardo. Allo stesso tempo viene attivato il flag monitor trigger registrazione (A50811).
 4. La registrazione termina quando l'istruzione TRSM(045) è stata eseguita un numero di volte sufficiente a completare la memoria di registrazione. Al termine della registrazione viene attivato il flag registrazione completata (A50812) e disattivato quello di esecuzione registrazione (A50813).
 5. Leggere il contenuto della memoria di registrazione con il dispositivo periferico.

Nella tabella seguente sono riportati i bit e i flag relativi nell'area ausiliaria. Solo A50814 e A50815 possono essere controllati dall'utente e A00815 può essere impostato su ON esclusivamente da un dispositivo periferico e non dal programma.

| Nome | Indirizzo | Funzionamento |
|------------------------------------|-----------|---|
| Flag monitor trigger registrazione | A50811 | Questo flag viene attivato quando la condizione di avvio è stata indicata con il bit di avvio registrazione. Viene disattivato all'inizio del campionamento per la registrazione successiva dal bit di avvio campionamento. |
| Flag registrazione completata | A50812 | Questo flag viene attivato quando i campioni di registrazione hanno riempito la memoria di registrazione. Viene disattivato alla successiva impostazione su ON del bit di avvio campionamento. |
| Flag di esecuzione registrazione | A50813 | Viene attivato quando il bit di avvio campionamento passa da OFF a ON. Viene disattivato al termine della registrazione. |

| Nome | Indirizzo | Funzionamento |
|----------------------------|-----------|--|
| Bit di avvio registrazione | A50814 | Le condizioni di avvio della registrazione vengono stabilite quando questo bit passa da OFF a ON. I campioni sono registrati dopo il ritardo specificato (ritardo positivo); altrimenti resta valido il numero indicato di campioni esistenti (ritardo negativo). |
| Bit di avvio campionamento | A50815 | Quando si imposta questo bit su ON mediante un dispositivo periferico, i campioni di dati iniziano a venire memorizzati nella memoria di registrazione con uno dei tre metodi utilizzati per determinare il campionamento: 1) Campionamento periodico (intervalli da 10 a 2,550 ms) 2) Campionamento all'esecuzione dell'istruzione TRSM(045) 3) Campionamento al termine di ogni ciclo È necessario che questo bit venga attivato e disattivato mediante un dispositivo periferico. |

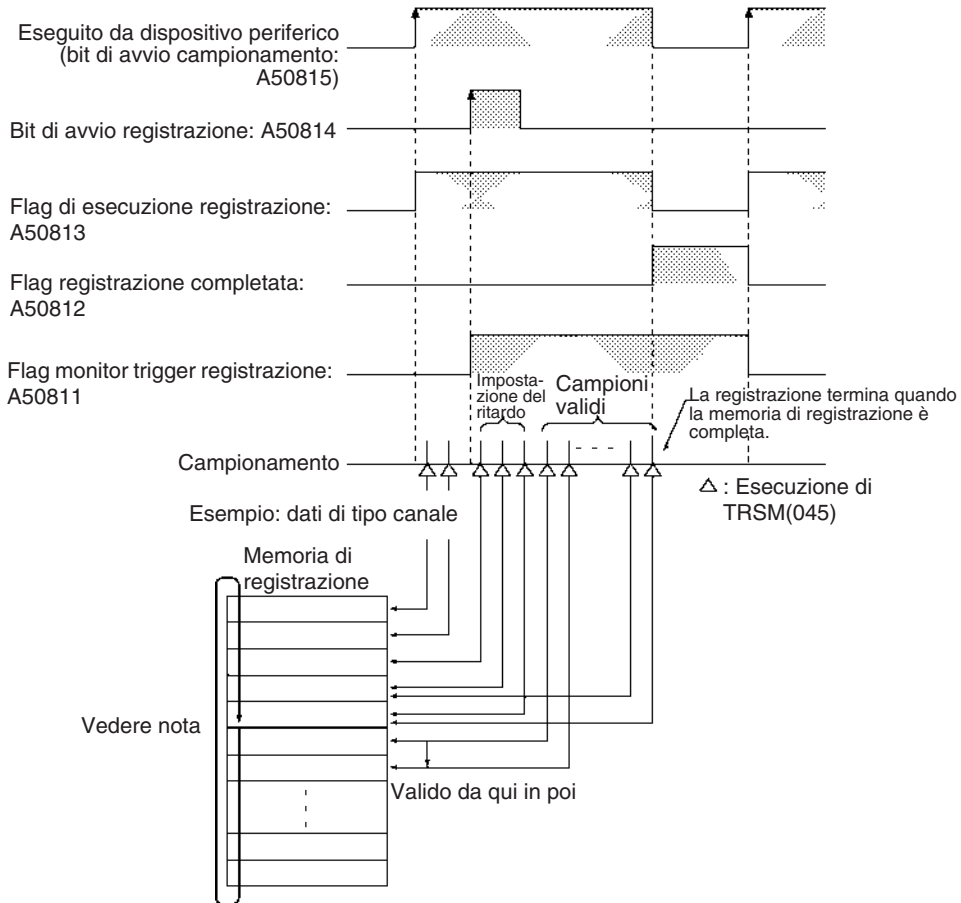
Avvertenze

TRSM(045) viene elaborata come NOP(000) quando la registrazione dei dati non viene eseguita o se l'intervallo di campionamento definito nei parametri con un dispositivo periferico non è impostato in modo da effettuare il campionamento durante l'esecuzione dell'istruzione TRSM(045).

Non attivare o disattivare il bit di avvio campionamento (A50815) dal programma, ma esclusivamente mediante un dispositivo periferico.

Esempio

Nell'esempio seguente è illustrato il funzionamento complessivo della registrazione dei dati.



Nota La memoria di registrazione è strutturata ad anello. I dati sono memorizzati nella parte finale dell'area della memoria di registrazione e quindi avvolti fino all'inizio dell'area, terminando esattamente prima del primo campione valido di dati.

3-30 Istruzioni di diagnostica per malfunzionamento

In questa sezione vengono illustrate le istruzioni utili a definire e gestire gli errori.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|-------------------------|-----------|-----------------|--------|
| FAILURE ALARM | FAL | 006 | 1079 |
| SEVERE FAILURE ALARM | FALS | 007 | 1087 |
| FAILURE POINT DETECTION | FPD | 269 | 1095 |

3-30-1 FAILURE ALARM: FAL(006)

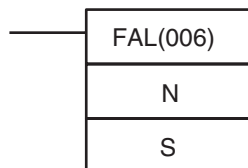
Scopo

Genera o cancella errori non fatali definiti dall'utente. Gli errori non fatali non causano l'arresto del PLC.

Con CPU CS1-H, CJ1-H e CJ1M, è possibile utilizzare FAL(006) anche per generare errori di sistema reversibili.

Simbolo programmazione ladder

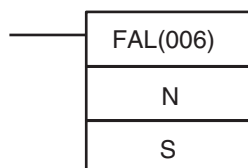
- Per generare o cancellare errori non fatali definiti dall'utente



N: numero FAL

S: primo canale o prima costante di messaggio (da 0000 a FFFF)

- Per generare errori di sistema reversibili (solo CS1-H, CJ1-H, CJ1M o CS1D)



N: numero FAL (valore in A529)

S: primo canale contenente il codice di errore e dettagli sull'errore stesso

Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FAL(006) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FAL(006) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Quando FAL(006) viene utilizzata per generare o cancellare errori definiti dall'utente funziona in modo leggermente diverso rispetto a quando la si utilizza per generare errori di sistema (solo CPU CS1-H, CJ1-H, CJ1M e CS1D).

Come generare o cancellare errori non fatali definiti dall'utente

Nella tabella che segue viene illustrato il funzionamento di ciascun operando.

Nota Il valore dell'operando N deve essere **diverso** dal contenuto di A529 (numero FAL/FALS generato dal sistema).

| N | S | Funzione |
|--|----------------------|---|
| 0 | Da #0001 a #01FF | Cancella l'errore non fatale e il relativo numero FAL. |
| | #FFFF | Cancella tutti gli errori non fatali. |
| | Altro* | Cancella gli errori non fatali più gravi. |
| da 1 a 511 (questi numeri FAL sono condivisi con numeri FALS) | Da #0000 a #FFFF | Genera un errore non fatale e il relativo numero FAL, ma nessun messaggio. |
| | Indirizzo del canale | Genera un errore non fatale e il relativo numero FAL. Il messaggio ASCII di 16 caratteri contenuto da S a S+7 verrà visualizzato sul dispositivo di programmazione. |

Nota *Altre impostazioni potrebbero prevedere le costanti da #0200 a #FFFE oppure un indirizzo del canale.

Come generare errori di sistema reversibili (solo CS1-H, CJ1-H, CJ1M o CS1D)

Nella tabella che segue viene illustrato il funzionamento di ciascun operando.

Nota Il valore dell'operando N deve essere **uguale** al contenuto di A529 (numero FAL/FALS generato dal sistema).

| Operando | Funzione |
|----------|--|
| N | Da 1 a 511 (questi numeri FAL sono condivisi con numeri FALS) |
| S | Codice dell'errore che verrà generato (vedere <i>Descrizione</i> più avanti). |
| S+1 | Codice dei dettagli dell'errore che verrà generato (vedere <i>Descrizione</i> più avanti). |

Caratteristiche operando

| Area | N | S |
|--------------------------------------|------------|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritentività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 a A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | da 0 a 511 | Da #0000 a #FFFF (binario) |
| Registri dati | --- | |

| Area | N | S |
|--|-----|--|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

Il funzionamento di FAL(006) dipende dal valore di N. Impostare N su 0000 per cancellare un errore e su un valore compreso tra 0001 e 01FF per generare un errore. Se il valore di N è uguale al contenuto di A529, verrà generato un errore di sistema (solo CPU CS1-H, CJ1-H, CJ1M e CS1D).

Come generare errori non fatali definiti dall'utente

Se FAL(006) viene eseguita con N impostato su un numero FAL (da &1 a &511) diverso dal contenuto di A529 (numero FAL/FALS generato dal sistema), verrà generato un errore non fatale con tale numero FAL e avrà luogo la seguente procedura:

1,2,3...

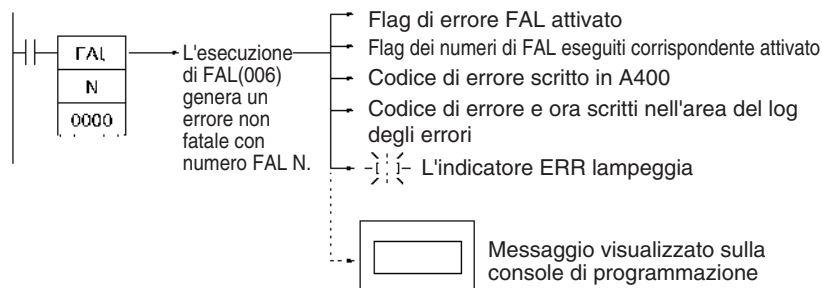
- Viene attivato il flag di errore FAL (A40215) (il PLC continua a funzionare).
- Viene attivato il flag dei numeri di FAL eseguiti per il numero FAL corrispondente. I flag da A36001 ad A39115 corrispondono ai numeri di FAL da 0001 a 01FF (da 1 a 511).
- Il codice di errore viene scritto in A400. I codici di errore da 4101 a 42FF corrispondono ai numeri di FAL da 0001 a 01FF (da 1 a 511).

Nota In caso di errore fatale o di errore non fatale grave durante l'esecuzione dell'istruzione FAL(006), il codice di errore dell'errore più grave verrà scritto in A400.

- Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).

Nota Con CPU CS1-H, CJ1-H e CJ1M, il registro degli errori non viene scritto nell'area del log degli errori nel caso in cui il PLC sia stato impostato in modo tale che gli errori generati da FAL(006) non vengano registrati, cioè se il bit 15 dell'indirizzo 129 della console di programmazione è impostato su 1.

- L'indicatore ERR della CPU lampeggia.
- Se è stato specificato un indirizzo del canale in S, il messaggio che inizia in S viene registrato e visualizzato sul dispositivo di programmazione.



Nella tabella seguente sono riportati i codici di errore e i flag di errore FAL per FAL(006).

| Numero FAL | Codici di errore FAL | Flag numeri di FAL eseguiti |
|------------------------|----------------------|-----------------------------|
| Da 1 a 511 decimale | Da 4101 a 42FF | Da A36001 ad A39115 |

Come visualizzare messaggi di errori non fatali definiti dall'utente

Se S è un indirizzo del canale e se un messaggio ASCII è stato ivi memorizzato, tale messaggio verrà visualizzato sul dispositivo periferico all'esecuzione di FAL(006). Se non si necessita di un messaggio, impostare S su una costante.

Il messaggio che inizia in S viene registrato all'esecuzione di FAL(006). Una volta registrato, il messaggio sarà visualizzato quando viene collegata una console di programmazione.

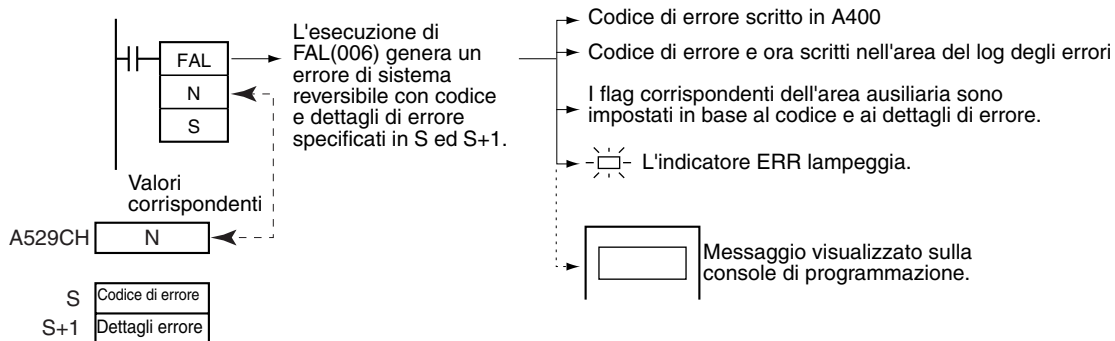
Nell'intervallo compreso tra S ed S+7 è possibile memorizzare un messaggio ASCII non superiore a 16 caratteri. In primo luogo verrà visualizzato il byte più a sinistra, cioè il più significativo, di ciascun canale.

Il codice di fine del messaggio è il carattere nullo (00 esadecimale). Se il carattere nullo viene omesso, tutti i 16 caratteri nei canali da S a S+7 saranno visualizzati.

Se il contenuto dei canali che contengono il messaggio è modificato in seguito all'esecuzione di FAL(006), il messaggio risulterà modificato di conseguenza.

Come generare errori di sistema reversibili (solo CS1-H, CJ1-H, CJ1M o CS1D)

Se FAL(006) viene eseguita con N impostato su un numero FAL (da &1 a &511) uguale al contenuto di A529 (numero FAL/FALS generato dal sistema), verrà generato un errore non fatale con codice e dettagli di errore specificati in S ed S+1. Contemporaneamente avrà luogo la seguente procedura:

**1,2,3...**

1. Il codice di errore specificato viene scritto in A400.
2. Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).
3. I flag corrispondenti dell'area ausiliaria sono impostati in base al codice e ai dettagli di errore.
4. L'indicatore ERR della CPU lampeggia e il PLC continua a funzionare.
5. Il messaggio di errore reversibile per l'errore di sistema specificato viene visualizzato sulla console di programmazione.

Nota

1. Quando si esegue il debug del programma, è possibile utilizzare FAL(006) per generare errori non fatali dal sistema. Ad esempio, è possibile generare un errore di sistema in modo intenzionale per verificare che i messaggi di errore siano visualizzati correttamente su un'interfaccia quale un terminale programmabile (PT, Programmable Terminal).
2. Il valore di A529 (numero FAL/FALS generato dal sistema) è un numero FAL fittizio (i numeri FAL, FALS e FPD sono condivisi) utilizzato quando si genera intenzionalmente un errore reversibile nel sistema. Si tratta di un numero FAL fittizio, per cui non modifica lo stato dei flag dei numeri di FAL eseguiti (da A36001 ad A39115) o il codice di errore. Quando è necessario generare due o più errori di sistema reversibili o meno, è possibile eseguire più volte le istruzioni FAL/FALS/FPD con gli stessi valori in A529 ed N, ma valori differenti in S ed S+1.

3. Se durante l'esecuzione dell'istruzione FAL(006) si verifica un errore più grave, compreso un errore irreversibile generato dal sistema o un errore FALS(007), il codice di errore dell'errore più grave verrà scritto in A400.
4. Per cancellare un errore di sistema generato da FAL(006), spegnere e riaccendere il PLC. È possibile tenere il PLC acceso, ma sarà comunque necessario eseguire la stessa procedura per cancellare l'errore specificato come se si fosse verificato a tutti gli effetti.

Nella tabella seguente viene indicato come specificare i codici e i dettagli di errore in S ed S+1.

| Nome errore | S | S+1 |
|---|---------------------|---|
| Errore task interrupt | 008B esadecimale | <ul style="list-style-type: none"> • Bit 15 OFF: errore di task ad interrupt Bit da 00 a 14: numero di task del task ad interrupt in cui si è verificato l'errore. • Bit 15 ON: esecuzione del task ad interrupt in conflitto con aggiornamento di modulo di I/O speciale Bit da 00 a 14: numero modulo del modulo di I/O speciale con conflitto di aggiornamento |
| Errore degli I/O di base | 009A esadecimale | Posizione rack del modulo in cui si è verificato l'errore <ul style="list-style-type: none"> • Bit da 08 a 15: numero rack (binario) del rack su cui è montato tale modulo • Bit da 00 a 07: numero slot (binario) dello slot su cui è montato tale modulo |
| Errore di configurazione del PLC | 009B esadecimale | Posizione dell'errore nelle impostazioni del PLC |
| Errore di verifica della tabella di I/O | 00E7 esadecimale | --- (non fisso) |
| Errore reversibile della scheda interna | 02F0 esadecimale | Informazioni sull'errore della scheda interna <ul style="list-style-type: none"> • Bit da 00 a 03: non valido • Bit da 04 a 15: errore definito dalla scheda interna |
| Errore nell'unità bus CPU CS1 | 0200 esadecimale | Numero di modulo dell'unità bus CPU CS1: da 0000 a 000F esadecimale |
| Errore del modulo di I/O speciale | 0300 esadecimale | Numero di modulo del modulo di I/O speciale: da 0000 a 005F o 00FF esadecimale (numero di modulo indeterminato) |
| Errore SYSMAC BUS | 00A0 esadecimale | Numero di modulo del modulo master SYSMAC BUS: 0000 o 0001 esadecimale |
| Errore batteria | 00F7 esadecimale | --- (non fisso) |
| Errore di configurazione dell'unità bus della CPU CS1 | 0400 esadecimale | Numero di modulo dell'unità bus CPU CS1: da 0000 a 000F esadecimale |
| Errore di configurazione del modulo di I/O speciale | 0500 esadecimale | Numero di modulo del modulo di I/O speciale: da 0000 a 005F esadecimale |

Come disabilitare le registrazioni di log degli errori per gli errori definiti dall'utente (solo CS1-H, CJ1-H, CJ1M o CS1D)

In genere, quando FAL(006) genera un errore definito dall'utente, il codice di errore e il momento in cui tale errore si verifica vengono scritti nell'area del log degli errori (da A100 ad A199). È possibile impostare la configurazione del PLC in modo tale che gli errori definiti dall'utente e generati con FAL(006) non vengano registrati nel log degli errori.

Sebbene l'errore non sia registrato nel log degli errori, vengono attivati il flag di errore FAL (40215) nonché il flag corrispondente nei flag dei numeri di FAL eseguiti (da A36001 ad A39115) e il codice di errore viene scritto in A400.

Se si desidera registrare soltanto gli errori generati dal sistema, disabilitare la registrazione di log degli errori per gli errori FAL (006) definiti dall'utente. Ad

esempio, questa funzione è utile durante il debug, se le istruzioni FAL(006) sono utilizzate in numerose applicazioni e il log degli errori tende a riempirsi di errori FAL(006) definiti dall'utente. Nella tabella seguente sono riportate le impostazioni di configurazione del PLC.

| Elemento | Impostazione | |
|---|--|-----|
| Indirizzo impostazione nella console di programmazione | Canale | 129 |
| | Bit | 15 |
| Nome | Registrazione del log di errore FAL | |
| Impostazioni | 0: registra errori FAL nel log degli errori. 1: non registrare errori FAL nel log degli errori. | |
| Impostazione predefinita | 0: registra errori FAL nel log degli errori. | |
| Frequenza di lettura delle impostazioni di configurazione del PLC | A ogni ciclo, se avviene un errore FAL | |

Anche se il bit 15 del canale 129 nella configurazione del PLC è impostato su 1 (non registrare errori FAL nel log degli errori), gli errori elencati qui di seguito vengono comunque registrati.

- Errori irreversibili generati da FALS(007)
- Errori di sistema reversibili
- Errori di sistema irreversibili
- Errori di sistema reversibili generati in modo intenzionale con FAL(006) o FPD(269)
- Errori di sistema irreversibili generati in modo intenzionale con FALS(007)

Come cancellare gli errori non fatali senza il dispositivo di programmazione

1. Cancellazione degli errori non fatali definiti dall'utente
Se FAL(006) viene eseguita con N impostato su 0, è possibile cancellare gli errori non fatali. Come illustrato nella tabella seguente, il valore di S determina l'elaborazione.

| S | Elaborazione |
|---|---|
| Da &1 a &511 (da 0001 a 01FF esadecimale) | Viene cancellato l'errore FAL del numero specificato. |
| FFFF esadecimale | Tutti gli errori non fatali, compresi quelli di sistema, vengono cancellati. |
| Da 0200 a FFFE esadecimale o specifica del canale | Errore non fatale più grave (anche se reversibile di sistema) mai verificatosi. Nel caso di più errori FAL, viene cancellato quello con il numero di FAL più piccolo. |

2. Cancellazione degli errori di sistema reversibili (solo CPU CS1-H, CJ1-H, CJ1M e CS1D)
Per cancellare errori di sistema reversibili generati con FAL(006), procedere in uno dei due modi seguenti.
 - Spegner e quindi riaccendere il PLC.
 - Se non si spegne il PLC, è necessario cancellare l'errore di sistema come se tale errore si fosse verificato a tutti gli effetti.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato compreso tra 0 e 511 decimale. ON se un errore di sistema reversibile è in fase di generazione (solo con CS1-H, CJ1-H, CJ1M e CS1D), ma il codice di errore o i dettagli di errore specificati sono scorretti. OFF in tutti gli altri casi. |

Nelle tabelle qui di seguito sono riportati i canali e i flag relativi nell'area ausiliaria.

- Solo canali dell'area ausiliaria / flag per errori definiti dall'utente

| Nome | Indirizzo | Funzionamento |
|-----------------------------|---------------------|--|
| Flag di errore FAL | A40215 | ON quando viene generato un errore con l'istruzione FAL(006). |
| Flag numeri di FAL eseguiti | Da A36001 ad A39115 | Quando si genera un errore con FAL(006), il flag corrispondente viene attivato. I flag da A36001 ad A39115 corrispondono ai numeri FAL da 0001 a 01FF. |

- Solo canali dell'area ausiliaria / flag per errori di sistema (solo CPU CS1-H, CJ1-H, CJ1M e CS1D)

| Nome | Indirizzo | Funzionamento |
|--------------------------------------|-----------|---|
| Numero FAL/FALS generato dal sistema | A529 | Quando un errore di sistema viene generato con FAL(006), viene utilizzato un numero FAL/FALS fittizio. Impostare lo stesso numero FAL/FALS fittizio in questo canale (da 0001 a 01FF esadecimale, da 1 a 511 decimale). |

- Solo canali dell'area ausiliaria / flag per errori definiti dall'utente ed errori di sistema

| Nome | Indirizzo | Funzionamento |
|---------------------------|-----------------|--|
| Area del log degli errori | Da A100 ad A199 | L'area del log degli errori contiene i codici di errore e l'indicazione di ora e giorno dell'evento relativi ai 20 errori più recenti, compresi quelli generati da FAL(006). |
| Codice di errore | A400 | Quando si verifica un errore, il relativo codice viene memorizzato in A400. I codici di errore per i numeri FAL da 0001 a 01FF vanno rispettivamente da 4101 a 42FF. Quando si verificano due o più errori contemporaneamente, il codice di errore dell'errore più grave viene memorizzato in A400. |

Avvertenze

N deve essere tra 0000 e 01FF. Se N non rientra nell'intervallo specificato, si verificherà un errore e verrà attivato il flag di errore.

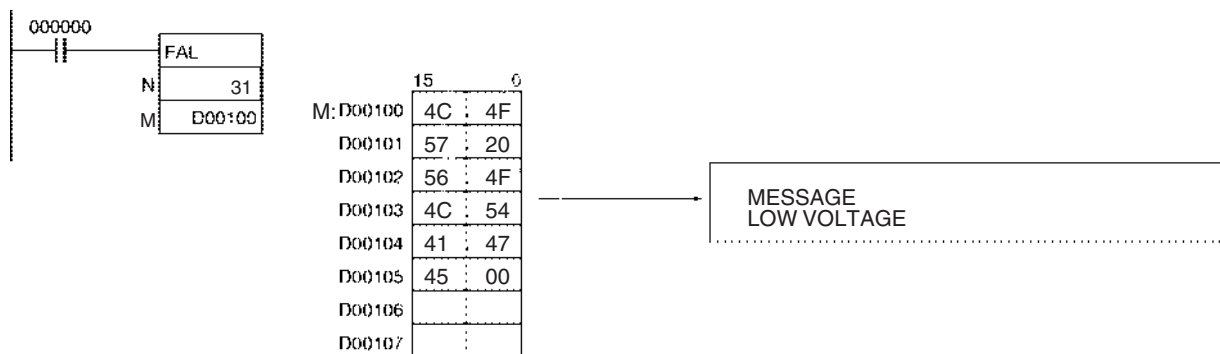
Esempi

Come generare un errore non fatale

Nell'esempio che segue, quando CIO 000000 è ON, FAL(006) genera un errore non fatale con numero FAL 31 e dà luogo alla seguente procedura.

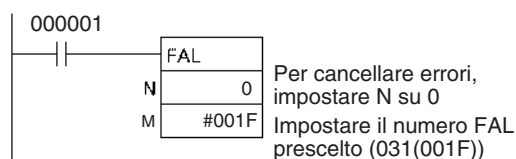
1,2,3...

1. Viene attivato il flag di errore FAL (A40215).
2. Viene attivato il flag dei numeri di FAL eseguiti corrispondente (A36114).
3. Il codice di errore corrispondente (411F) viene scritto in A400.
Nota Quando si verificano due o più errori contemporaneamente, il codice di errore dell'errore più grave (codice di errore più alto) viene memorizzato in A400.
4. Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).
5. L'indicatore ERR della CPU lampeggia.
6. Il messaggio ASCII da D00100 a D00107 viene visualizzato sul dispositivo periferico. Se non si necessita di un messaggio, specificare una costante per S.



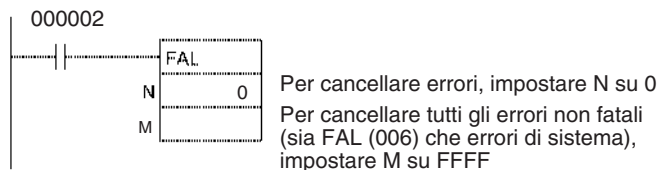
Come cancellare un determinato errore non fatale

Nell'esempio che segue, quando CIO 000001 è ON, FAL(006) cancella l'errore non fatale con numero FAL 31 e disattiva il flag dei numeri di FAL eseguiti corrispondente (A36114) nonché il flag di errore FAL (A40215).



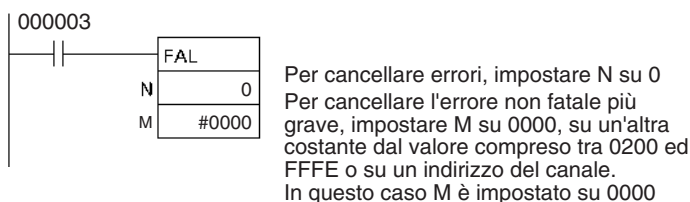
Come cancellare tutti gli errori non fatali

Nell'esempio che segue, quando CIO 000002 è ON, FAL(006) cancella tutti gli errori non fatali e disattiva i flag dei numeri di FAL eseguiti (da A36001 ad A39115) nonché il flag di errore FAL (A40215).



Come cancellare l'errore non fatale più grave

Nell'esempio che segue, quando CIO 000003 è ON, FAL(006) cancella l'errore non fatale più grave mai verificatosi e reimposta il codice di errore in A400. Se l'errore cancellato era stato originariamente generato con FAL(006), verranno disattivati il flag dei numeri di FAL eseguiti corrispondente nonché il flag di errore FAL (A40215).

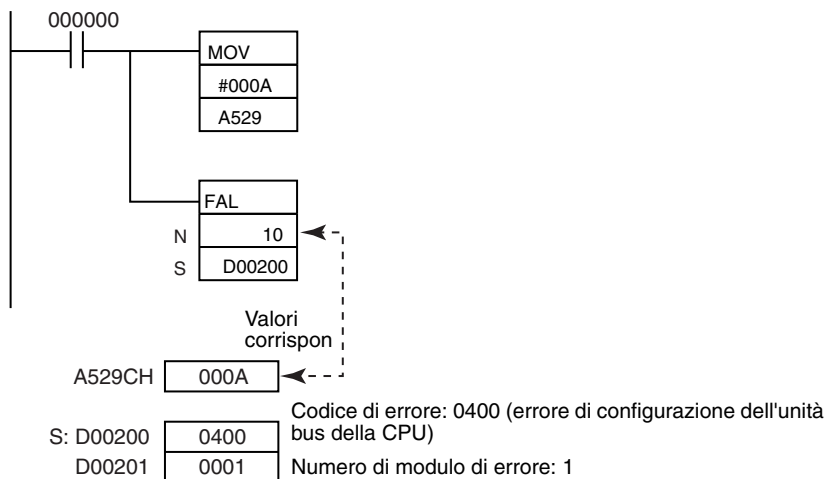


Come generare un errore di sistema reversibile (solo CS1-H, CJ1-H, CJ1M o CS1D)

Nell'esempio che segue, quando 000000 è ON, FAL(006) genera un errore di configurazione dell'unità bus della CPU per il numero di modulo 1. In questo caso, viene utilizzato il numero FAL fittizio 10 e il valore corrispondente (000A esadecimale) viene memorizzato in A529.

- 1,2,3...
1. Se si tratta dell'errore più grave, il codice di errore specificato (0400) viene scritto in A400.
 2. Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).

3. Il flag di errore di configurazione dell'unità bus della CPU (A40203) e il flag di errore di configurazione dell'unità bus della CPU per il numero di modulo 1 (A42701) verranno attivati.
4. L'indicatore ERR della CPU lampeggia.
5. Sulla console di programmazione viene visualizzato un messaggio (CPU BU ST ERR 01) che indica che si è verificato un errore nell'unità bus 1 della CPU.



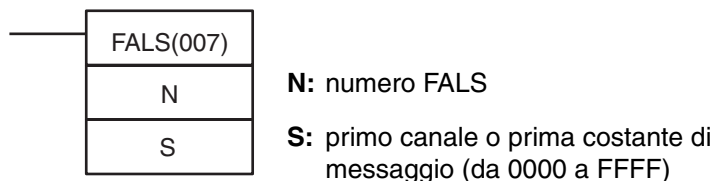
3-30-2 SEVERE FAILURE ALARM: FALS(007)

Scopo

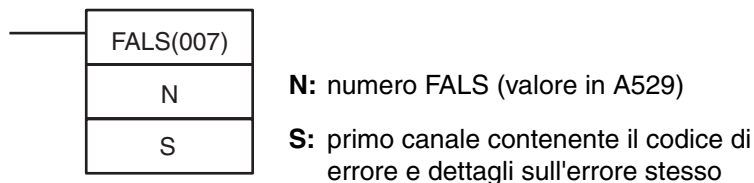
Genera errori fatali definiti dall'utente. Gli errori fatali causano l'arresto del PLC. Con CPU CS1-H, CJ1-H, CJ1M e CS1D, è possibile utilizzare FALS(007) anche per generare errori di sistema irreversibili.

Simbolo programmazione ladder

- Per generare errori fatali definiti dall'utente



- Per generare errori di sistema irreversibili (solo CS1-H, CJ1-H, CJ1M o CS1D)



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FALS(007) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

Come generare errori fatali definiti dall'utente

Nella tabella che segue viene illustrato il funzionamento di ciascun operando.

Nota Il valore dell'operando N deve essere **diverso** dal contenuto di A529 (numero FAL/FALS generato dal sistema).

| Operando | Funzione |
|----------|--|
| N | Da 1 a 511 (questi numeri FALS sono condivisi con numeri FAL) |
| S | Specifica il primo degli otto canali che contengono un messaggio ASCII da visualizzare sul dispositivo di programmazione. Specifica una costante (da 0000 a FFFF) nel caso in cui non sia necessario alcun messaggio. |

Come generare errori irreversibili dal sistema (solo CS1-H, CJ1-H, CJ1M o CS1D)

Nella tabella che segue viene illustrato il funzionamento di ciascun operando.

Nota Il valore dell'operando N deve essere **uguale** al contenuto di A529 (numero FAL/FALS generato dal sistema).

| Operando | Funzione |
|----------|--|
| N | Da 1 a 511 (questi numeri FALS sono condivisi con numeri FAL) |
| S | Codice dell'errore che verrà generato (vedere <i>Descrizione</i> più avanti). |
| S+1 | Codice dei dettagli dell'errore che verrà generato (vedere <i>Descrizione</i> più avanti). |

Caratteristiche operando

| Area | N | S |
|--|-------------------------|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritentività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Solo valori specificati | Da #0000 a #FFFF (binario) |
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

FALS(007) genera un errore fatale. Con CPU CS1-H, CJ1-H, CJ1M e CS1D è possibile utilizzare FALS(007) anche per generare errori di sistema irreversibili nonché errori fatali definiti dall'utente (se il valore di N è uguale al contenuto di A529, viene generato un errore di sistema).

Come generare errori fatali definiti dall'utente

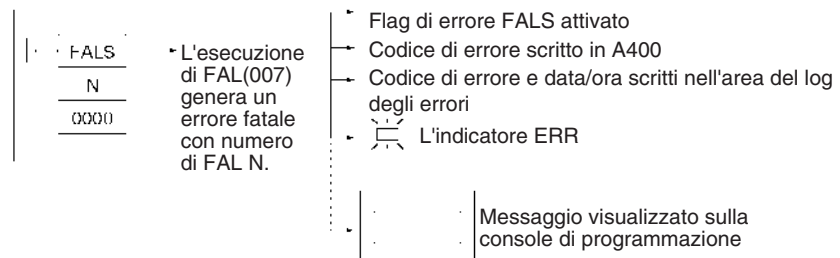
Se FALS(007) viene eseguita con N impostato su un numero FALS (da 1 a 511) diverso dal contenuto di A529 (numero FAL/FALS generato dal sistema), verrà generato un errore fatale con tale numero FALS e avrà luogo la seguente procedura:

1,2,3...

- Viene attivato il flag di errore FALS (A40106) e si interrompe il funzionamento del PLC.
- Il codice di errore viene scritto in A400. I codici di errore da C101 a C2FF corrispondono ai numeri FALS da 0001 a 01FF (da 1 a 511).

Nota Se si è verificato un errore più grave, cioè con un codice di errore più alto, rispetto all'istruzione FALS(007), A400 conterrà il codice di errore dell'errore più grave.

- Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).
- L'indicatore ERR della CPU si accende.
- Se è stato specificato un indirizzo del canale in S, il messaggio ASCII che inizia in S viene registrato e visualizzato sul dispositivo periferico.



Nella tabella seguente sono riportati i codici di errore per l'istruzione FALS(007).

| Numero FALS | Codici di errore FALS |
|-------------|-----------------------|
| Da 1 a 511 | Da C101 a C2FF |

Nota Il metodo di immissione del numero FALS N è differente per CX-Programmer e una console di programmazione. Indicare da #1 a #511 per CX-Programmer e da 001 a 511 per una console di programmazione.

Come visualizzare messaggi di errori fatali definiti dall'utente

Se S è un indirizzo del canale, il messaggio ASCII che inizia in S viene visualizzato sul dispositivo di programmazione all'esecuzione di FALS(007). Se non si necessita di un messaggio, impostare S su una costante.

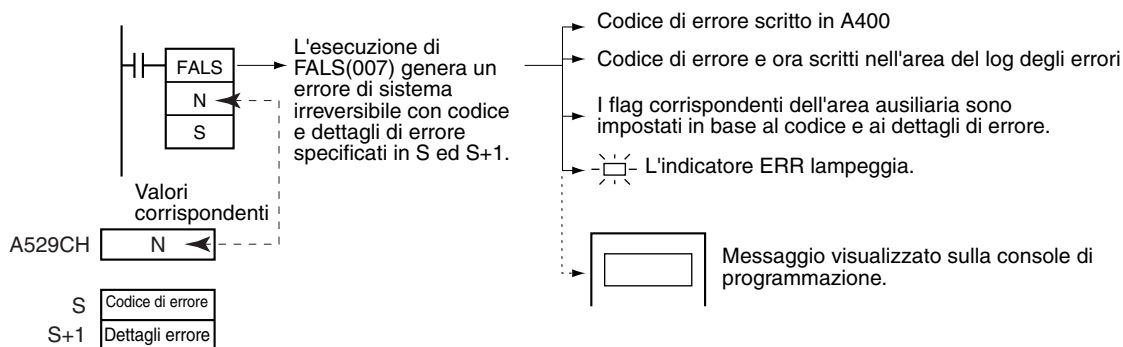
Il messaggio che inizia in S viene registrato all'esecuzione di FALS(007). Una volta registrato, il messaggio sarà visualizzato quando viene collegata una console di programmazione.

Nell'intervallo compreso tra S ed S+7 è possibile memorizzare un messaggio ASCII non superiore a 16 caratteri. In primo luogo verrà visualizzato il byte più a sinistra, cioè il più significativo, di ciascun canale.

Il codice di fine del messaggio è il carattere nullo (00 esadecimale). Se il carattere nullo viene omissso, tutti i 16 caratteri nei canali da S a S+7 saranno visualizzati.

Se il contenuto dei canali che contengono il messaggio è modificato in seguito all'esecuzione di FALS(007), il messaggio risulterà modificato di conseguenza.

Come generare errori di sistema reversibili (solo CS1-H, CJ1-H, CJ1M o CS1D)



Se FALS(007) viene eseguita con N impostato su un numero FAL (da 1 a 511) uguale al contenuto di A529 (numero FAL/FALS generato dal sistema), verrà generato un errore fatale con codice di errore e di dettagli di errore specificati in S ed S+1. Contemporaneamente avrà luogo la seguente procedura:

- 1,2,3...**
1. Il codice di errore specificato viene scritto in A400.
 2. Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).
 3. I flag corrispondenti dell'area ausiliaria sono impostati in base al codice e ai dettagli di errore.
 4. L'indicatore ERR della CPU si accende e il funzionamento del PLC viene interrotto.
 5. Il messaggio di errore irreversibile per l'errore di sistema specificato viene visualizzato sulla console di programmazione.

- Nota**
1. Il valore di A529 (numero FAL/FALS generato dal sistema) è un numero FAL fittizio (i numeri FAL, FALS e FPD sono condivisi) utilizzato quando si genera intenzionalmente un errore reversibile nel sistema. Si tratta di un numero FAL fittizio, per cui non corrisponde al codice di errore. Quando è necessario generare due o più errori di sistema, è possibile eseguire più volte le istruzioni FAL/FALS/FPD con gli stessi valori in A529 ed N, ma con valori differenti in S ed S+1.
 2. Se durante l'esecuzione dell'istruzione FALS(007) si verifica un errore più grave, compreso un errore irreversibile generato dal sistema o un altro errore FALS(007), il codice di errore dell'errore più grave verrà scritto in A400.
 3. Per cancellare un errore di sistema generato da FALS(007), spegnere e riaccendere il PLC. È possibile tenere il PLC acceso, ma sarà comunque necessario eseguire la stessa procedura per cancellare l'errore specificato come se si fosse verificato a tutti gli effetti. Per ulteriori informazioni, consultare il capitolo sulla risoluzione dei problemi del manuale *CS/CJ Series Operation Manual*.
 4. Nella tabella seguente è indicato come il bit di ritenività IOM influisca sullo stato della memoria di I/O e sullo stato delle uscite dei moduli di uscita in seguito a un errore di sistema irreversibile generato mediante FALS(007).

| Bit di ritenività dell'area IOM (A50012) | Stato della memoria I/O | Stato delle uscite dei moduli di uscita |
|--|-------------------------|---|
| ON | Mantenuto | OFF |
| OFF | Azzerato | OFF |

Nota Diversamente che con gli errori fatali generati dall'utente, gli errori di sistema generati da FALS(007) azzerano la memoria di I/O nel caso in cui in bit di ritentività IOM sia OFF. Il contenuto dei seguenti elementi viene cancellato: area CIO, area di lavoro, flag e valori attuali del temporizzatore, registri indice e registri dati.

Nella tabella che segue viene indicato come specificare i codici e i dettagli di errore in S ed S+1.

| Nome errore | S | S+1 |
|---|-------------------------|--|
| | Codice di errore | Dettagli errore |
| Errore di memoria | 80F1 esadeci male | <ul style="list-style-type: none"> • Bit da 00 a 09: posizione dell'errore di memoria Bit 00: programma utente Bit 04: configurazione del PLC Bit 05: tabella di I/O registrata Bit 07: tabella di routing Bit 08: configurazione dell'unità bus della CPU Bit 09: errore di trasferimento tramite la scheda di memoria • Bit da 10 a 15: non validi |
| Errore del bus degli I/O | 80C0 esadeci male | <ul style="list-style-type: none"> • Bit da 00 a 07: numero dello slot in cui si è verificato l'errore del bus di I/O Slot da 0 a 9: da 00 a 09 esadecimale Slot sconosciuto: 0F esadecimale • Bit da 08 a 15: numero del rack in cui si è verificato l'errore del bus di I/O Slot da 0 a 7: da 00 a 07 esadecimale Rack sconosciuto: 0F esadecimale |
| Errore di duplicazione numero unità | 80E9 esadeci male | Numero di modulo duplicato dell'unità bus CPU Da 0000 a 000F esadecimale |
| | | Numero di modulo duplicato del modulo di I/O speciale Da 8000 a 805F esadecimale |
| Errore di duplicazione numero rack | 80EA esadeci male | Numero duplicato di rack (assegnazioni di canali sovrapposti) Da 0000 a 0006 esadecimale |
| Errore irreversibile della scheda interna | 82F0 esadeci male | Causa dell'errore Bit da 00 a 03: errore definito dalla scheda interna Bit da 04 a 15: non validi |

| Nome errore | S | S+1 |
|--|---------------------|--|
| | Codice di errore | Dettagli errore |
| Errore di superamento del numero di punti di I/O massimo | 80E1 esadecimale | Bit da 13 a 15: causa dell'errore Bit da 00 a 12: dettagli <ul style="list-style-type: none"> • Il numero totale dei punti di I/O è troppo elevato. Bit da 13 a 15: 000 Bit da 00 a 12: numero di punti di I/O (binario) • Il numero degli ingressi ad interrupt è troppo elevato. Bit da 13 a 15: 001 Bit da 00 a 12: numero di ingressi ad interrupt (binario) Bit da 00 a 12: valore zero • Un numero di modulo del modulo slave è duplicato o un modulo slave C500 ha più di 320 punti di I/O. Bit da 13 a 15: 010 Bit da 00 a 12: numero di modulo del modulo slave (binario) • Il numero di modulo di un'interfaccia di I/O (esclusi i sistemi slave) è duplicato. Bit da 13 a 15: 011 Bit da 00 a 12: numero modulo (binario) • Un numero di modulo del modulo master è duplicato o non è compreso nell'intervallo di impostazione consentito. Bit da 13 a 15: 100 Bit da 00 a 12: numero di modulo del modulo master (binario) • Il numero dei rack di espansione è troppo elevato. Bit da 13 a 15: 101 Bit da 00 a 12: numero di rack di espansione (binario) • Un modulo C200H di I/O speciale o di I/O remoti non è stato riconosciuto. Bit da 13 a 15: 110 |
| Errore di impostazione della tabella di I/O | 80E0 esadecimale | --- (non fisso) |
| Errore di programma | 80F0 esadecimale | <ul style="list-style-type: none"> • Bit da 08 a 15: causa dell'errore Bit 15: errore di overflow UM Bit 14: errore di istruzione illegale Bit 13: errore di overflow comparativo Bit 12: errore di task Bit 11: errore di END mancante Bit 10: errore di accesso illegale Bit 09: errore BCD DM/EM indiretto Bit 08: errore di istruzione • Bit da 00 a 07: non validi |
| Errore di superamento del tempo di ciclo | 809F esadecimale | --- (non fisso) |

Cancellazione degli errori di sistema irreversibili FALS (007) (solo CPU CS1-H, CJ1-H, CJ1M e CS1D)

Per cancellare errori di sistema irreversibili generati con FAL(007), procedere in uno dei due modi seguenti.

1. Spegner e quindi riaccendere il PLC.
2. Se non si spegne il PLC, è necessario cancellare l'errore di sistema come se tale errore si fosse verificato a tutti gli effetti.

Cancellazione degli errori fatali FALS(007) definiti dall'utente

Per cancellare gli errori generati da FALS(007), eliminare in primo luogo la causa dell'errore e quindi cancellare l'errore da un dispositivo di programmazione oppure spegnere e riaccendere il PLC.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato compreso tra 0001 e 01FF (da 1 a 511 decimale). ON se un errore di sistema irreversibile è in fase di generazione (solo con CS1-H, CJ1-H, CJ1M e CS1D), ma il codice di errore o i dettagli di errore specificati sono scorretti. OFF in tutti gli altri casi. |

Nelle tabelle qui di seguito sono riportati i canali e i flag relativi nell'area ausiliaria.

- Solo canali dell'area ausiliaria / flag per errori definiti dall'utente

| Nome | Indirizzo | Funzionamento |
|---------------------|-----------|---|
| flag di errore FALS | A40106 | ON quando viene generato un errore con l'istruzione FAL(007). |

- Solo canali dell'area ausiliaria / flag per errori di sistema (solo CPU CS1-H, CJ1-H, CJ1M e CS1D)

| Nome | Indirizzo | Funzionamento |
|--------------------------------------|-----------|--|
| Numero FAL/FALS generato dal sistema | A529 | Quando un errore di sistema è generato con FALS(007), viene utilizzato un numero FAL/FALS fittizio. Impostare lo stesso numero FAL/FALS fittizio in questo canale (da 0001 a 01FF esadecimale, da 1 a 511 decimale). |

- Solo canali dell'area ausiliaria / flag per errori definiti dall'utente ed errori di sistema

| Nome | Indirizzo | Funzionamento |
|---------------------------|-----------------|--|
| Area del log degli errori | Da A100 ad A199 | L'area del log degli errori contiene i codici di errore e l'indicazione di ora e giorno dell'evento relativi ai 20 errori più recenti, compresi quelli generati da FALS(007). |
| Codice di errore | A400 | Quando si verifica un errore, il relativo codice viene memorizzato in A400. I codici di errore per i numeri FALS da 0001 a 01FF (da 1 a 511 decimale) vanno rispettivamente da C101 a C2FF. Quando si verificano due o più errori contemporaneamente, il codice di errore dell'errore più grave viene memorizzato in A400. |

Avvertenze

Il codice di fine del messaggio è il carattere nullo (00 esadecimale). Se il carattere nullo viene omissso, tutti i 16 caratteri nei canali da S a S+7 saranno visualizzati.

Il valore di N deve essere compreso tra 0001 e 01FF. Se N non rientra nell'intervallo specificato, si verificherà un errore e verrà attivato il flag di errore.

Esempi

Come generare un errore definito dall'utente

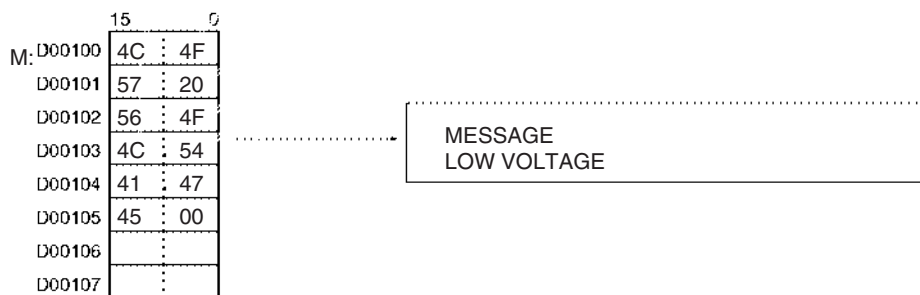
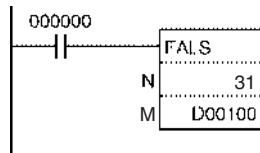
Nell'esempio che segue, quando CIO 000000 è ON, FALS(007) genera un errore fatale con numero FAL 31 e dà luogo alla seguente procedura.

1,2,3...

- Viene attivato il flag di errore FALS (A40106)
- Il codice di errore corrispondente (C11F) viene scritto in A400.

Nota A400 contiene il codice di errore dell'errore più grave tra quelli che si sono verificati, compresi gli errori di sistema reversibili o meno, nonché gli errori generati da FAL(006) e FAL(007).

3. Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).
4. L'indicatore ERR della CPU si accende.
5. Il messaggio ASCII da D00100 a D00107 viene visualizzato sul dispositivo periferico. Se non si necessita di un messaggio, specificare una costante per S.

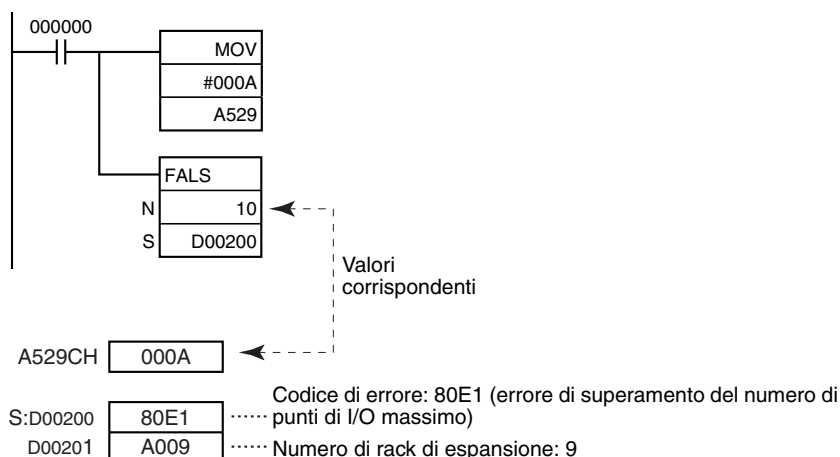


Come generare un errore di sistema reversibile (solo CPU CS1-H, CJ1-H, CJ1M e CS1D)

Nell'esempio che segue, quando CIO 000000 è ON, FALS(007) genera un errore di superamento del numero di punti di I/O massimo (numero eccessivo di rack di espansione collegati, in questo caso, 9). Di conseguenza viene utilizzato il numero FAL fittizio 10 e il valore corrispondente (000A esadecimale) è memorizzato in A529.

1,2,3...

1. Se si tratta dell'errore più grave, il codice di errore specificato (80E1) viene scritto in A400.
2. Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).
3. Il flag di superamento del numero di punti di I/O massimo (A40111) viene attivato.
4. L'indicatore ERR della CPU si accende e il funzionamento del PLC viene interrotto.
5. Viene visualizzato un messaggio (TOO MANY I/O PNT) sulla console di programmazione che segnala che si è verificato un errore di superamento del numero di punti di I/O massimo.

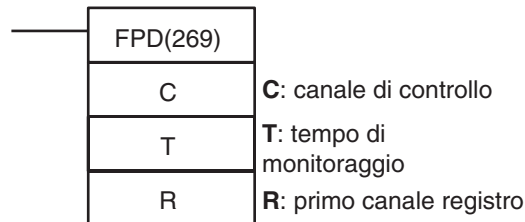


3-30-3 FAILURE POINT DETECTION: FPD(269)

Scopo

L'istruzione FPD(269) consente di rilevare un errore in un blocco di istruzioni monitorando il tempo che intercorre tra l'esecuzione dell'istruzione stessa e l'attivazione di un'uscita di diagnostica, individuando in tal modo l'ingresso che impedisce l'attivazione di un'uscita.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FPD(269) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

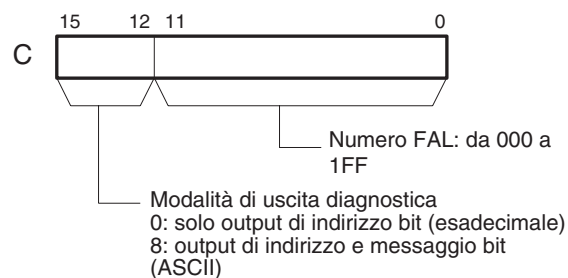
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| Non consentita | OK | OK | Non consentita |

Operandi

C: canale di controllo

C deve essere una costante compresa tra 0000 e 01FF o tra 8000 e 81FF. Nel diagramma che segue viene illustrata la funzione delle cifre nel canale di controllo.



T: tempo di monitoraggio

T deve essere compreso tra 0 e 9.999 decimale (tra 0000 e 270F esadecimale). Il valore 0 disattiva il monitoraggio del tempo, mentre i valori compresi tra 1 e 270F impostano il tempo di monitoraggio tra 0,1 e 999,9 secondi.

R: primo canale registro

Le funzioni dei canali registro sono descritte a pagina 1098.

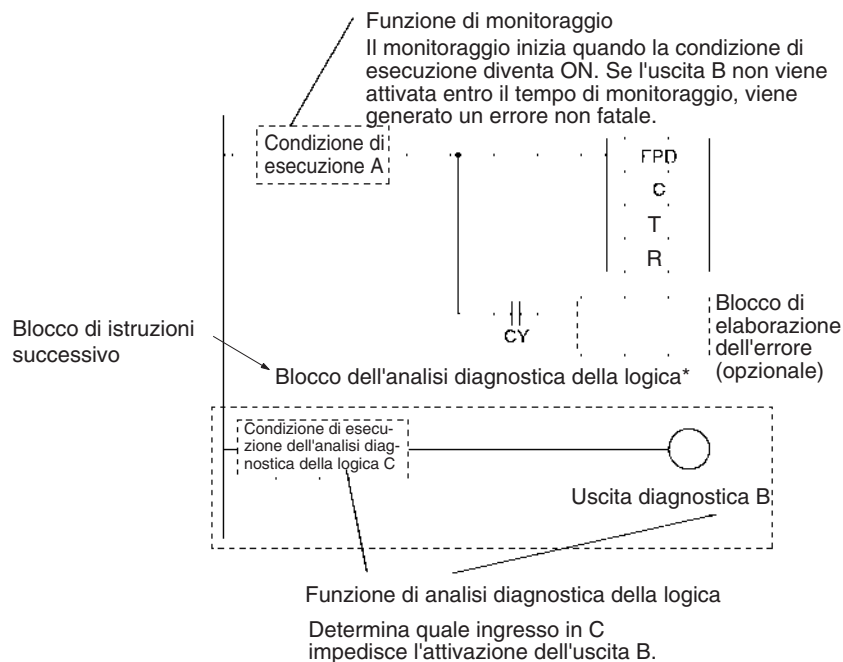
Caratteristiche operando

| Area | C | T | R |
|------------------------|-----|------------------------------------|-----------------|
| Area CIO | --- | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | --- | Da W000 a W511 | |
| Area bit di ritenività | --- | Da H000 a H511 | |
| Area bit ausiliaria | --- | Da A000 ad A447 Da A448 ad A959 | Da A448 ad A959 |

| Area | C | T | R |
|--|-------------------------|--|-----|
| Area del temporizzatore | --- | Da T0000 a T4095 | |
| Area del contatore | --- | Da C0000 a C4095 | |
| Area DM | --- | Da D00000 a D32767 | |
| Area EM senza banco | --- | Da E00000 a E32767 | |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | Solo valori specificati | Da #0000 a #270F (binario) | --- |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | |

Descrizione

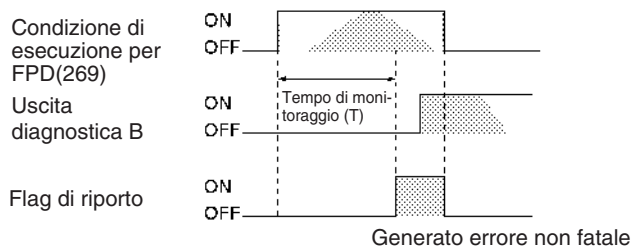
FPD(269) esegue il monitoraggio del tempo e un'analisi diagnostica della logica. Se l'uscita diagnostica non viene attivata entro il tempo di monitoraggio specificato, la funzione di monitoraggio del tempo genererà un errore non fatale con il numero FAL specificato. La funzione di analisi diagnostica della logica indica quale ingresso impedisce l'attivazione dell'uscita.



Nota *Il blocco dell'analisi diagnostica della logica inizia con la prima istruzione LD (non LD TR) oppure LD NOT dopo FPD(269) e termina con la prima istruzione OUT (non OUT TR) o altra istruzione sul lato destro.

Funzione di monitoraggio del tempo

L'istruzione FPD(269) inizia a monitorare il tempo al momento dell'esecuzione cioè quando la condizione di esecuzione A passa a ON; se l'uscita di diagnostica non viene attivata entro il tempo di monitoraggio specificato, l'istruzione genera un errore non fatale e attiva il flag di riporto.



Nota L'uscita diagnostica deve essere attivata entro il tempo di monitoraggio. Per impostare il tempo di monitoraggio in modo automatico, è possibile utilizzare la funzione di autoimpostazione.

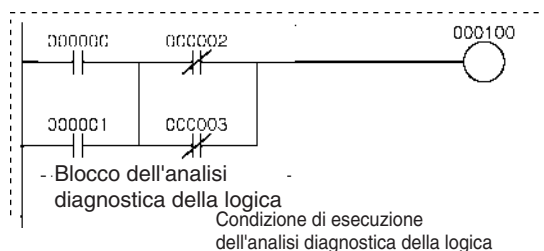
Quando viene attivato il flag di riporto, avviene la seguente elaborazione (purché il numero FAL non sia impostato su 000 in C).

- 1,2,3...**
1. Viene attivato il flag di errore FAL (A40215) (il PLC continua a funzionare).
 2. Viene attivato il flag dei numeri di FAL eseguiti per il numero FAL specificato. I flag da A36001 ad A39115 corrispondono ai numeri FAL da 001 a 1FF.
 3. Il codice di errore corrispondente viene scritto in A400. I codici di errore da 4101 a 42FF corrispondono ai numeri FAL da 001 a 1FF.
Se contemporaneamente si è verificato un errore più grave, cioè un errore con un codice di errore più alto, il codice di errore di tale errore viene memorizzato in A400.
 4. Il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199).
 5. L'indicatore ERR della CPU lampeggia.
 6. Se la modalità di uscita è stata impostata per l'output di indirizzo e messaggio bit (cifra più a sinistra di C impostata su 8), viene visualizzato il messaggio ASCII memorizzato da R+2 a R+10 come messaggio di errore non fatale.

Funzione di analisi diagnostica della logica

Per ogni ciclo in cui la condizione di esecuzione di FPD(269) è impostata su ON, l'istruzione individua il bit di ingresso che determina la disattivazione dell'uscita diagnostica e scrive l'indirizzo di tale bit nell'area di registro che inizia in R.

Nell'esempio che segue, se i bit di ingresso compresi tra CIO 000000 e CIO 000003 sono tutti impostati su ON, FPD(269) consente di evincere che la condizione di CIO 000002, normalmente chiusa, è responsabile della disattivazione prolungata del CIO 000100 di uscita. FPD(269) attiverrebbe il flag di indirizzo bit trovato (bit 15 di R) scrivendo l'indirizzo del bit nei canali registro da R+2 a R+4.



La funzione di analisi diagnostica della logica viene eseguita a ogni ciclo fino a quando la condizione di esecuzione di FPD(269) è impostata su ON. La funzione di analisi diagnostica della logica prescinde da quella di monitoraggio del tempo.

Quando due o più bit di ingresso impediscono l'attivazione dell'uscita diagnostica, l'indirizzo del primo bit di ingresso nella condizione di esecuzione (sulla riga più alta dell'istruzione e più vicina alla barra di distribuzione sinistra) viene inviato nell'intervallo compreso tra R+2 ed R+4.

La funzione di analisi diagnostica della logica verifica i bit di ingresso nelle istruzioni LD, LD NOT, AND, AND NOT, OR e OR NOT, comprese le variazioni di aggiornamento differenziato e immediato. Non verranno controllati i bit di ingresso di altre istruzioni e operandi indirizzati indirettamente tramite indirizzi indice.

Il blocco dell'analisi diagnostica della logica inizia con la prima istruzione LD (non LD TR) oppure LD NOT dopo FPD(269) e termina con la prima istruzione OUT (non OUT TR) o altra istruzione sul lato destro.

Le modalità di uscita diagnostica sono due e si impostano con la cifra più a sinistra di C.

1,2,3...**1. Modalità output di indirizzo bit (cifra più a sinistra di C = 0)**

Il bit 15 di R (il flag di indirizzo bit trovato) viene impostato su ON quando si trova un indirizzo di bit di ingresso e il bit 14 di R indica se tale ingresso è normalmente impostato su ON o su OFF.

L'indirizzo esadecimale a 8 cifre di memoria del PLC del bit di ingresso viene inviato a R+3 ed R+2.

2. Modalità output di indirizzo e messaggio bit (cifra più a sinistra di C = 8)

Il bit 15 di R (il flag di indirizzo bit trovato) viene impostato su ON quando si trova un indirizzo di bit di ingresso e il bit 14 di R indica se tale ingresso è normalmente impostato su ON o su OFF.

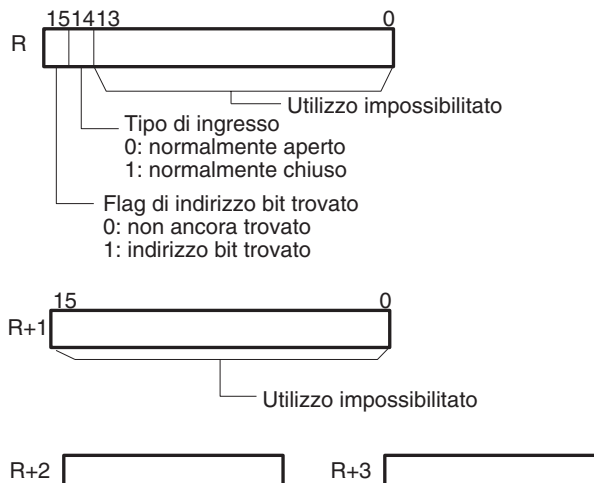
L'indirizzo del bit di ingresso viene inviato nell'intervallo compreso tra R+2 ed R+4 come 6 caratteri ASCII.

Funzioni del canale registro

I canali registro contengono i risultati della funzione di diagnostica ed eventualmente anche un messaggio ASCII di errore che viene visualizzato se la funzione di monitoraggio del tempo genera un errore. La funzione dei canali registro dipende dalla modalità di uscita diagnostica impostata con la cifra più a sinistra di C.

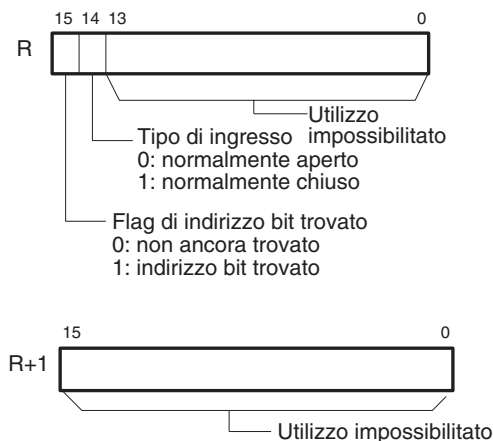
Output di indirizzo bit (C = 0□□□)

Quando la cifra più a sinistra di C è impostata su 0, l'indirizzo esadecimale a 8 cifre di memoria del PLC del bit di ingresso viene inviato in R+2 ed R+3. R contiene due flag che indicano se è stato trovato un bit di ingresso e se tale bit venga utilizzato in una condizione di ingresso generalmente aperta o chiusa.



Output di indirizzo e messaggio bit (C = 8□□□)

Quando la cifra più a sinistra di C è impostata su 8, l'indirizzo ASCII del bit di ingresso viene inviato nell'intervallo compreso tra R+2 ed R+4. R contiene due flag che indicano se è stato trovato un bit di ingresso e se tale bit venga utilizzato in una condizione di ingresso generalmente aperta o chiusa.



I canali registro da R+2 a R+4 indicano l'indirizzo dell'ingresso che ha impedito l'attivazione dell'uscita diagnostica. L'indirizzo del bit viene inviato a tali canali in ASCII. Nella tabella che segue sono indicate le rappresentazioni ASCII per ciascuna area.

| Area | Testo ASCII | Note |
|-------------------------|---------------------|--|
| Area ausiliaria | Da A00000 ad A95915 | --- |
| Area di ritenzione | Da H00000 a H51115 | --- |
| Area di lavoro | Da W00000 a W51115 | --- |
| Area CIO | Da 000000 a 665515 | --- |
| Flag dei task | Da TK0000 a TK0031 | --- |
| Area del temporizzatore | Da _T0000 a _T4095 | Le “_” rappresentano uno spazio ASCII (codice carattere 20). |
| Area del contatore | Da _C0000 a _C4095 | |

| | | | |
|-----|----|---|--------------------------------|
| | 15 | | |
| R+2 | W | 5 | |
| R+3 | 1 | 1 | Indirizzo bit scritto in ASCII |
| R+4 | 1 | 5 | |

Per W51115 i canali registro compresi tra R+2 ed R+5 avrebbero i seguenti valori:

| Canale | Bit da 8 a 15 | Bit da 0 a 7 |
|--------|------------------|--|
| R+2 | W | 5 |
| R+3 | 1 | 1 |
| R+4 | 1 | 5 |
| R+5 | 2D (esadecimale) | Tipo di ingresso (esadecimale) 30: normalmente aperto 31: normalmente chiuso |

L'utente può memorizzare un messaggio ASCII nei canali registro compresi tra R+6 ed R+10. Se la funzione di monitoraggio del tempo genera un errore non fatale, il messaggio verrà visualizzato sul dispositivo di programmazione. Segnalare la fine del messaggio con il carattere nullo (00 esadecimale).

| | | | | |
|------|----|---|---|---|
| | 15 | 8 | 7 | 0 |
| R+6 | | | | |
| R+7 | | | | |
| R+8 | | | | |
| R+9 | | | | |
| R+10 | | | | |

Disabilitazione della registrazione di log degli errori per errori FPD(269) non fatali (solo CS1-H, CJ1-H, CJ1M o CS1D)

Normalmente, quando la funzione di monitoraggio del tempo di FPD(269) genera un errore non fatale, il codice di errore e il momento in cui tale errore si è verificato vengono scritti nell'area del log degli errori (da A100 ad A199). Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D è possibile impostare la configurazione del PLC in modo tale che gli errori reversibili generati da FAL(006) non vengano registrati nel log degli errori.

Sebbene l'errore non sia registrato nel log degli errori, vengono attivati il flag di errore FAL (40215) nonché il flag corrispondente nei flag dei numeri di FAL eseguiti (da A36001 ad A39115) e il codice di errore viene scritto in A400.

Se si desidera registrare soltanto gli errori generati dal sistema, disabilitare la registrazione di log degli errori per gli errori di monitoraggio del tempo di FPD(269). Ad esempio, questa funzione è utile durante il debug, se le istruzioni FPD(269) e FAL(006) sono utilizzate in numerose applicazioni e il log degli errori tende a riempirsi di tali errori. Nella tabella seguente sono riportate le impostazioni di configurazione del PLC.

| Elemento | Impostazione | |
|---|--|-----|
| Indirizzo impostazione nella console di programmazione | Canale | 129 |
| | Bit | 15 |
| Nome | Registrazione del log di errore FAL | |
| Impostazioni | 0: registra errori FAL nel log degli errori. 1: non registrare errori FAL nel log degli errori. | |
| Impostazione predefinita | 0: registra errori FAL nel log degli errori. | |
| Frequenza di lettura delle impostazioni di configurazione del PLC | A ogni ciclo, se avviene un errore FAL | |

Anche se il bit 15 del canale 129 nella configurazione del PLC è impostato su 1 (non registrare errori FAL nel log degli errori), gli errori elencati qui di seguito vengono comunque registrati.

- Errori irreversibili generati da FALS(007)
- Errori di sistema reversibili

- Errori di sistema irreversibili
- Errori di sistema reversibili generati in modo intenzionale con FAL(006) o FPD(269)
- Errori di sistema irreversibili generati in modo intenzionale con FALS(007)

Impostazione del tempo di monitoraggio mediante la funzione di autoimpostazione

1,2,3...

Se per T è stato specificato un indirizzo di canale, è possibile impostare il tempo di monitoraggio in modo automatico con la funzione di autoimpostazione. Una volta impostato un indirizzo di canale per T, seguire la procedura indicata.

1. Attivare il bit di autoimpostazione FPD (A59800).
2. FPD(269) calcola il tempo dal momento in cui la condizione di esecuzione di FPD(269) viene impostata su ON fino all'attivazione dell'uscita diagnostica.
3. Se il tempo calcolato supera l'impostazione del tempo di monitoraggio, in T verrà memorizzata un'impostazione pari a una volta e mezza il tempo calcolato.

Flag

| Nome | Etichetta | Funzionamento |
|-----------------|-----------|--|
| Flag di errore | ER | ON se C non rientra nell'intervallo specificato compreso tra 0000 e 01FF o 8000 e 81FF. ON se T non rientra nell'intervallo specificato compreso tra 0000 e 270F. OFF in tutti gli altri casi. |
| Flag di riporto | CY | ON se l'uscita diagnostica è ancora disattivata allo scadere del tempo di monitoraggio. OFF in tutti gli altri casi. |

Nella tabella qui di seguito sono riportati i canali e i flag relativi nell'area ausiliaria.

| Nome | Indirizzo | Funzionamento |
|-----------------------------|---------------------|--|
| Flag di errore FAL | A40215 | ON se viene registrato un errore (FAL) non-fatale nel monitoraggio del tempo. |
| Flag numeri di FAL eseguiti | Da A36001 ad A39115 | Quando viene registrato un errore (FAL) non fatale nel monitoraggio del tempo, viene attivato il flag corrispondente. I flag da A36001 ad A39115 corrispondono ai numeri FAL da 0001 a 01FF. |
| Area del log degli errori | Da A100 ad A199 | L'area del log degli errori contiene i codici di errore e l'indicazione di ora e giorno dell'evento relativi ai 20 errori più recenti, compresi quelli generati da FPD(269). |
| Codice di errore | A400 | Quando si verifica un errore, il relativo codice viene memorizzato in A400. I codici di errore per i numeri FAL da 0001 a 01FF vanno rispettivamente da 4101 a 42FF. Quando si verificano due o più errori contemporaneamente, il codice di errore dell'errore più grave viene memorizzato in A400. |
| Bit di autoimpostazione FPD | A59800 | Attivare questo bit per impostare automaticamente (funzione di autoimpostazione) il tempo di monitoraggio quando viene eseguita l'istruzione FPD(269). |

Avvertenze

Durante l'utilizzo della funzione di monitoraggio del tempo è necessario che la condizione di esecuzione di FPD(269) resti impostata su ON per tutto il tempo di monitoraggio impostato in T.

La condizione di esecuzione di FPD(269) deve essere composta da una combinazione di ingressi normalmente aperti e normalmente chiusi.

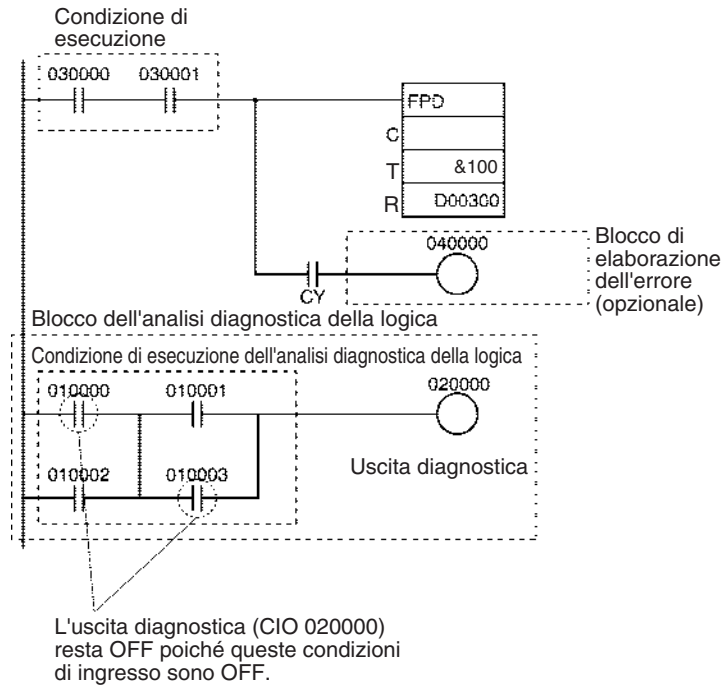
Il blocco di elaborazione dell'errore è opzionale. Se si include un blocco di elaborazione dell'errore, verificare che vengano utilizzate le uscite o altre istruzioni sul lato destro. In questa fase non è possibile utilizzare LD ed LD NOT.

FPD(269) può essere specificata più di una volta, ma è necessario che ogni istruzione presenti un'impostazione univoca di registro(R).

Il tempo del monitoraggio viene aggiornato soltanto all'esecuzione di FPD(269). Se il tempo del ciclo supera 100 ms, il tempo di monitoraggio non viene aggiornato normalmente e FPD(269) non funzionerà in modo corretto poiché il tempo di monitoraggio è aggiornato in unità di 100 ms.

Esempi

Nell'esempio di programma riportato qui di seguito sono descritte le funzioni di monitoraggio del tempo e di analisi diagnostica della logica. L'uscita diagnostica (CIO 020000) non viene attivata poiché CIO 010000 e CIO 010003 restano impostati su OFF nella condizione di esecuzione dell'analisi diagnostica della logica.



Funzione di monitoraggio del tempo

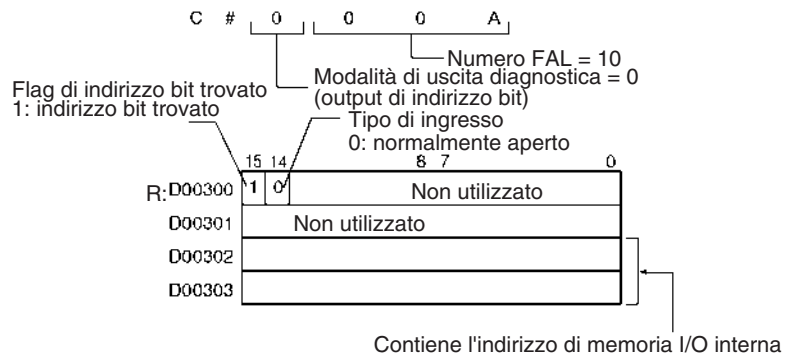
Se l'uscita diagnostica (CIO 020000) non passa a ON entro 10 secondi dall'attivazione di CIO 030000 e CIO 030001, viene generato un errore non fatale e avrà luogo la seguente procedura.

1,2,3...

1. Viene attivato il flag di riporto.
2. Quando le 3 cifre più a destra di C indicano un numero FAL di 00A esadecimale (10), viene attivato il flag dei numeri di FAL eseguiti corrispondente (A36010), il codice di errore relativo (410A) è scritto in A400 e il flag di errore FAL (A40215) viene a sua volta attivato.

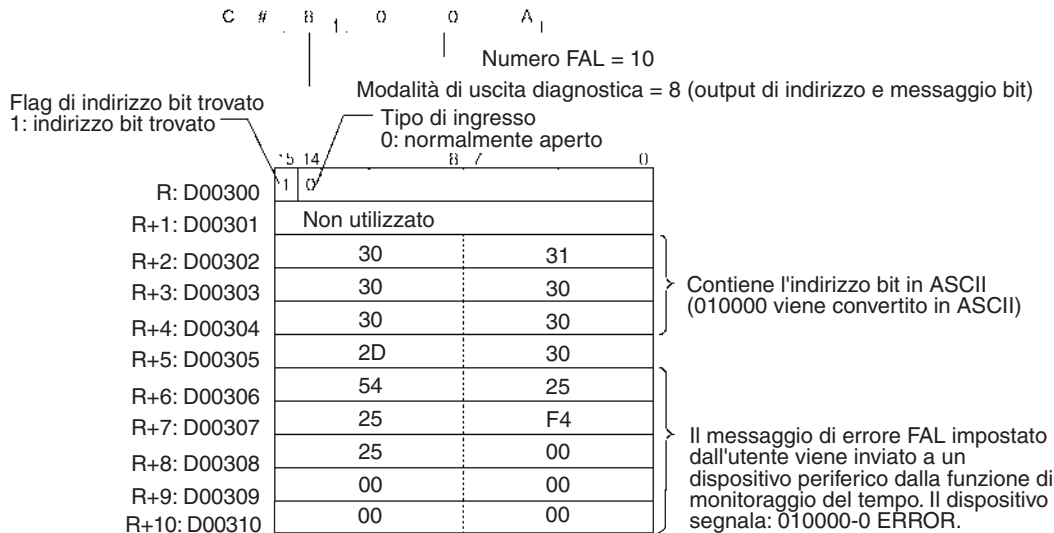
Funzione di analisi diagnostica della logica (C = 000A)

Poiché l'ultima cifra a sinistra di C è 0 (modalità output di indirizzo bit) l'indirizzo di memoria del PLC di CIO 010000 viene inviato a D00303 e D00302 (rispetto a CIO 010003, CIO 010000 è su una riga più in alto dell'istruzione).



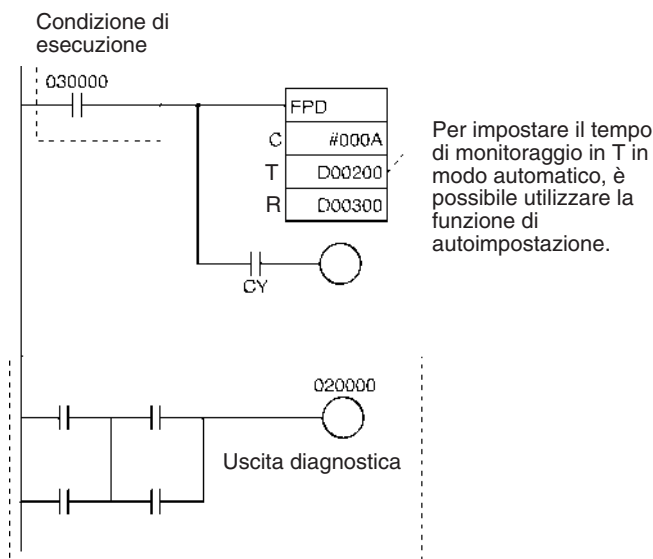
Funzione di analisi diagnostica della logica (C = 800A)

Poiché l'ultima cifra a sinistra di C è 8 (modalità output di indirizzo e messaggio bit) l'indirizzo di CIO 010000 (010000) viene inviato nell'intervallo compreso tra D00302 e D00304 in ASCII.

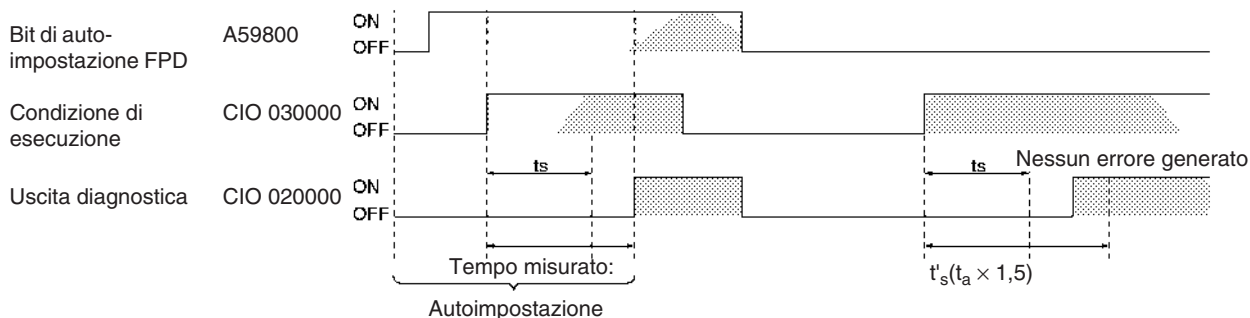


Impostazione del tempo di monitoraggio mediante la funzione di autoimpostazione

Quando un indirizzo di canale è stato specificato per T, è possibile definire il tempo di monitoraggio in modo automatico mediante la funzione di autoimpostazione.



Per avviare la funzione di autoimpostazione, attivare A59800 (il bit di autoimpostazione FPD). Mentre A59800 è impostato su ON, l'istruzione FPD(269) calcola il tempo impiegato dall'uscita di diagnostica (CIO 020000) per passare a ON in seguito all'attivazione della condizione di esecuzione (CIO 030000). Se in T il tempo calcolato supera il tempo di monitoraggio, il tempo calcolato viene moltiplicato per 1,5 e il valore così ottenuto viene memorizzato in T come il nuovo tempo di monitoraggio.



t_s : impostazione iniziale in T
 t_a : tempo misurato
 t'_s : nuova impostazione in T dopo l'autoimpostazione
 (quando $t_a > t_s$, $t'_s = t_a \times 1,5$)

3-31 Altre istruzioni

In questa sezione sono descritte le istruzioni per gestire il flag di riporto, per selezionare il banco EM e aumentare il tempo massimo di ciclo.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|------------------------------|-----------|-----------------|--------|
| SET CARRY | STC | 040 | 1104 |
| CLEAR CARRY | CLC | 041 | 1105 |
| SELECT EM BANK | EMBC | 281 | 1106 |
| EXTEND MAXIMUM CYCLE TIME | WDT | 094 | 1108 |
| SAVE CONDITION FLAGS | CCS | 282 | 1110 |
| LOAD CONDITION FLAGS | CCL | 283 | 1112 |
| CONVERT ADDRESS FROM CV | FRMCV | 284 | 1113 |
| CONVERT ADDRESS TO CV | TOCV | 285 | 1117 |
| DISABLE PERIPHERAL SERVICING | IOSP | 287 | 1121 |
| ENABLE PERIPHERAL SERVICING | IORS | 288 | 1123 |

3-31-1 SET CARRY: STC(040)

Imposta il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | STC(040) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @STC(040) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Descrizione

Quando la condizione di esecuzione è impostata su ON, STC(040) attiva il flag di riporto (CY). Ciononostante, questo flag verrà attivato e disattivato all'esecuzione di tutte le istruzioni successive che lo riguardano.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF o non modificato (vedere nota) |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag di riporto | CY | ON |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

ROL(027), ROLL(572), ROR(028) e RORL(573) utilizzano il flag di riporto nelle loro operazioni di scorrimento a rotazione. Per impostare e cancellare il flag di riporto durante una di tali istruzioni, utilizzare STC(040) CLC(041).

3-31-2 CLEAR CARRY: CLC(041)

Scopo

Disattiva il flag di riporto (CY).

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | CLC(041) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CLC(041) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Descrizione

Quando la condizione di esecuzione è impostata su ON, CLC(040) disattiva il flag di riporto (CY). Ciononostante, questo flag verrà attivato e disattivato all'esecuzione di tutte le istruzioni successive che lo riguardano.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|------------------------------------|
| Flag di errore | ER | OFF o non modificato (vedere nota) |
| Flag di uguaglianza | = | OFF o non modificato (vedere nota) |
| Flag di riporto | CY | OFF |
| Flag negativo | N | OFF o non modificato (vedere nota) |

Nota Nelle CPU CS1 e CJ1 questi flag sono impostati su OFF.
Nelle CPU CS1-H, CJ1-H, CJ1M e CS1D questi flag vengono lasciati inalterati.

Avvertenze

+C(402), +CL(403), +BC(406) e +BCL(407) utilizzano il flag di riporto nelle loro operazioni di addizione. Per impedire qualsiasi interferenza da parte di istruzioni precedenti, utilizzare CLC(041) appena prima delle istruzioni menzionate.

-C(412), -CL(413), -BC(416) e -BCL(417) utilizzano il flag di riporto nelle loro operazioni di sottrazione. Per impedire qualsiasi interferenza da parte di istruzioni precedenti, utilizzare CLC(041) appena prima delle istruzioni menzionate.

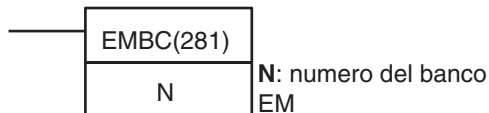
ROL(027), ROLL(572), ROR(028) e RORL(573) utilizzano il flag di riporto nelle loro operazioni di scorrimento a rotazione. Per impostare e cancellare il flag di riporto durante una di tali istruzioni, utilizzare STC(040) CLC(041).

Nota Le istruzioni +(400), +L(401), +B(404), +BL(405), -(410), -L(411), -B(414) e -BL(415) non includono il flag di riporto nelle loro operazioni di addizione e sottrazione. Preferire tali istruzioni per eseguire addizioni e sottrazioni.

3-31-3 SELECT EM BANK: EMBC(281)

Scopo Modifica il banco EM corrente.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | EMBC(281) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @EMBC(281) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero banco EM

Specifica il nuovo numero banco EM in formato esadecimale (da 0000 a 000C).

Caratteristiche operando

| Area | N |
|--------------------------------------|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritentività | Da H000 a H511 |
| Area bit ausiliaria | Da A000 a A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 |
| Costanti | Da #0000 a #000C (binario) |
| Registri dati | Da DR0 a DR15 |

| Area | N |
|--|--|
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

EMBC(281) modifica l'attuale banco EM (area di memoria dei dati estesa) in quello indicato dal numero banco EM (N). Allo stesso tempo, il nuovo numero banco EM viene inviato ad A301.

Nell'area EM sono disponibili fino a 13 banchi (da 0 a C) in ognuno dei quali vi sono 32.768 canali (da E00000 a E32767). Gli indirizzi EM possono essere identificati nei due modi descritti qui di seguito. Se si utilizza il primo metodo, è necessario specificare EMBC(281) per modificare l'attuale banco EM.

1,2,3...

1. Per specificare indirizzi nell'attuale banco EM, è possibile indicare indirizzi EM senza numero di banco, ad esempio, da E00000 a E32767.
2. Per specificare indirizzi in un determinato banco EM, è possibile indicare indirizzi EM con numero di banco, ad esempio, da En_00000 a En_32767 (n = da 0 a C).

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo specificato da 0000 a 000C. ON se N indica un numero di banco EM inesistente. Tale errore si verifica se il banco EM specificato è stato registrato come memoria per i file nelle impostazioni del PLC. OFF in tutti gli altri casi. |

Nella tabella seguente sono riportati i flag relativi nell'area ausiliaria.

| Nome | Indirizzo | Funzionamento |
|-------------------|-----------|---|
| Banco EM corrente | A301 | Contiene l'attuale numero banco EM in formato esadecimale (da 0000 a 000C). |

Avvertenze

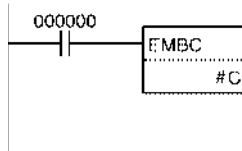
Se un'operazione viene invertita tra task, è mantenuto l'attuale numero banco EM modificato in un task ciclico. Ad esempio, se EMBC(281) viene utilizzato nel task 1 per modificare l'attuale numero banco EM dal banco B al banco C, quest'ultimo sarà il banco EM corrente per tutti i task ciclici anche quando l'operazione passa al task 2.

L'attuale numero banco EM modificato in un task ad interrupt è valido solo durante l'esecuzione dell'interrupt in cui ha subito la modifica. Il precedente numero banco EM verrà restituito al completamento dell'esecuzione del task ad interrupt.

Se il banco EM specificato è stato registrato come memoria per i file nelle impostazioni del PLC, si verifica un errore.

Esempi

Nell'esempio seguente, quando CIO 000000 passa a ON, l'attuale numero banco EM viene modificato in banco C e il nuovo numero banco (000C esadecimale) viene inviato in A301.



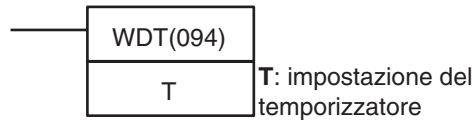
A301 000C

3-31-4 EXTEND MAXIMUM CYCLE TIME: WDT(094)

Scopo

Aumenta il tempo di ciclo massimo solo per il ciclo in cui viene eseguita l'istruzione. Se per una particolare elaborazione è temporaneamente necessario disporre di un tempo di ciclo più lungo, è possibile utilizzare WDT(094) per impedire che si verifichino errori causati da tempi di ciclo prolungati.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | WDT(094) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @WDT(094) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

T: impostazione del temporizzatore

Indica l'impostazione del temporizzatore watchdog tra 0000 e 0F9F esadecimale o tra &0000 e &3999 decimale.

Caratteristiche operando

| Area | T |
|--|--------------------------|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritentività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 0000 a 0F9F (binario) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Descrizione

WDT(094) prolunga il tempo di ciclo massimo nel ciclo in cui viene eseguita l'istruzione. L'impostazione del temporizzatore watchdog nella configurazione del PLC viene aumentata di un intervallo di $T \times 10$ ms (da 0 a 39.990 ms).

Nella tavola seguente sono riportate le impostazioni del temporizzatore watchdog nella configurazione del PLC. Il valore predefinito per il tempo di ciclo massimo è 1.000 ms, ma è possibile impostare qualsiasi valore compreso tra 1 e 40.000 ms in unità di 10 ms.

| Nome | Funzione | Impostazioni |
|-----------------------------|--|--|
| Tempo ciclo di monitoraggio | Se il tempo di ciclo supera l'impostazione massima, verrà registrato un errore di superamento del tempo di ciclo massimo (fatale). | 0: impostazione predefinita (1.000 ms) 1: impostazione tempo utente |
| | Imposta il tempo di ciclo massimo (l'impostazione è valida solo quando la prima impostazione è predefinita su 1). | Da 0001 a 0FA0 (da 1 a 40.000 ms in unità di 10 ms) |

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se l'impostazione del temporizzatore watchdog supera 40 secondi. OFF in tutti gli altri casi. |

Nella tabella qui di seguito sono riportati i flag e i canali relativi nell'area ausiliaria.

| Nome | Indirizzo | Funzionamento |
|--|-------------|---|
| Flag di superamento del tempo di ciclo massimo | A40108 | ON quando il tempo di ciclo corrente supera il tempo di ciclo massimo (tempo ciclo di monitoraggio) impostato nella configurazione del PLC. Si tratta di un errore fatale che determina l'interruzione dell'esecuzione del programma. |
| Tempo di ciclo massimo | A262 e A263 | Questi canali contengono il tempo di ciclo massimo in formato binario a 32 bit. Il valore viene aggiornato a ogni ciclo. |
| Tempo di ciclo attuale | A264 e A265 | Questi canali contengono il tempo di ciclo attuale in formato binario a 32 bit. Il valore viene aggiornato a ogni ciclo. |

Avvertenze

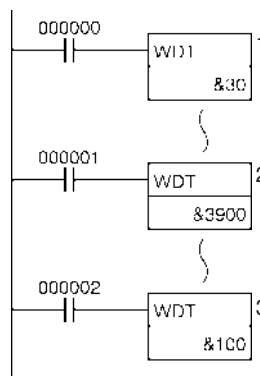
È possibile utilizzare WDT(094) più di una volta in un ciclo. In tal caso, gli aumenti del tempo di ciclo vengono sommati, ma il totale non deve superare 40.000 ms. Se il ciclo è stato già prolungato fino a 40.000 ms, non è più possibile eseguire WDT(094).

Esempi

Nell'esempio viene utilizzato il tempo di ciclo massimo predefinito di 1.000 ms.

1,2,3...

- Quando CIO 000000 passa a ON, la prima istruzione WDT(094) aumenta il tempo di ciclo massimo di 300 ms (30×10 ms). In questo caso il tempo di ciclo massimo sarà di 1.300 ms.
- Quando CIO 000001 passa a ON, la seconda istruzione WDT(094) tenta di aumentare il tempo di ciclo massimo di altri 39.000 ms. Poiché il nuovo tempo di ciclo massimo (40.300 ms) è maggiore del limite superiore di 40.000 ms, i 300 ms in eccesso vengono ignorati. Di conseguenza, la seconda istruzione WDT(094) prolunga il tempo di ciclo massimo di 38.700 ms effettivi.
- Quando CIO 000002 passa a ON, la terza istruzione WDT(094) tenta di aumentare il tempo di ciclo massimo di altri 1.000 ms. Poiché il tempo di ciclo massimo ha già raggiunto il limite superiore di 40.000 ms, la terza istruzione non verrà eseguita.



3-31-5 SAVE CONDITION FLAGS: CCS(282)

Salva lo stato attuale dei flag di condizione in un'area separata della CPU. Lo stato attuale dei flag viene mantenuto affinché possa essere letto (ripristinato) con CCL(283) in un altro punto del programma, in un task differente o persino in un ciclo successivo.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CCS(282) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CCS(282) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

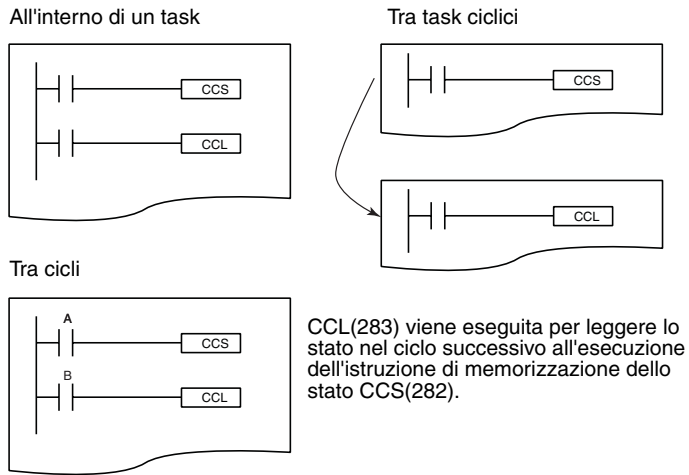
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| OK | OK | OK | OK |

Descrizione

Se la condizione di esecuzione è ON, CCS(282) memorizza lo stato attuale dei flag di condizione (esclusi i flag di sempre ON e sempre OFF) in un'area separata della CPU. Viene conservato lo stato dei seguenti flag di condizione: ER, CY, >, =, <, N, OF, UF, >=, <> e <=.

Successivamente, solo CCL(283), l'istruzione LOAD CONDITION FLAGS, consente di leggere (ripristinare) lo stato memorizzato dei flag di condizione. È possibile leggere lo stato in ognuno dei casi seguenti.

- All'interno di un task
- Tra diversi task ciclici
- Tra cicli

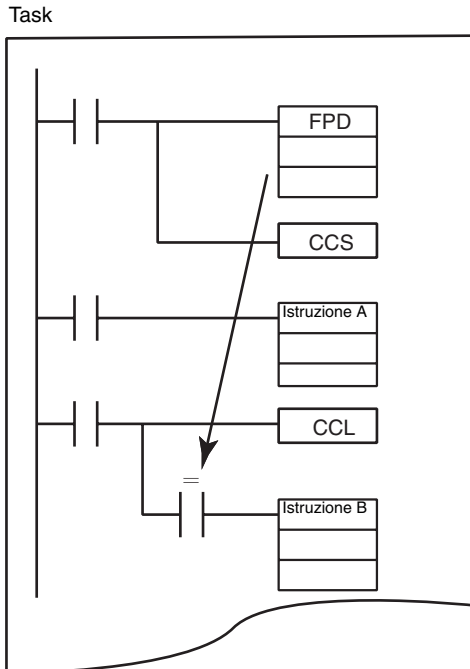


CCL(283) viene eseguita per leggere lo stato nel ciclo successivo all'esecuzione dell'istruzione di memorizzazione dello stato CCS(282).

- Nota**
1. Non è possibile salvare/caricare lo stato dei flag di condizione tra un task ciclico e uno ad interrupt.
 2. Quando viene eseguita, CCS(282) sovrascrive le informazioni esistenti sui flag di condizione.

Nel passaggio da un task all'altro vengono eliminati tutti i flag di condizione. Per salvare e caricare lo stato dei flag di condizione tra task o cicli, utilizzare le istruzioni CCS(282) e CCL(283).

Ad esempio, CCS(282) e CCL(283) consentono di utilizzare lo stato del flag CY (errore diagnostico del monitoraggio del tempo) dall'esecuzione di FPD(269) a una fase successiva del programma e non necessariamente solo in seguito all'esecuzione dell'istruzione.



I risultati del confronto vengono memorizzati nei flag di condizione. In questo caso i risultati dell'istruzione COMPARE possono essere utilizzati nell'istruzione B anche se sono influenzati dall'esecuzione dell'istruzione A.

Conserva lo stato dei flag di condizione in un'area separata della CPU.

Ripristina lo stato dei flag di condizione.

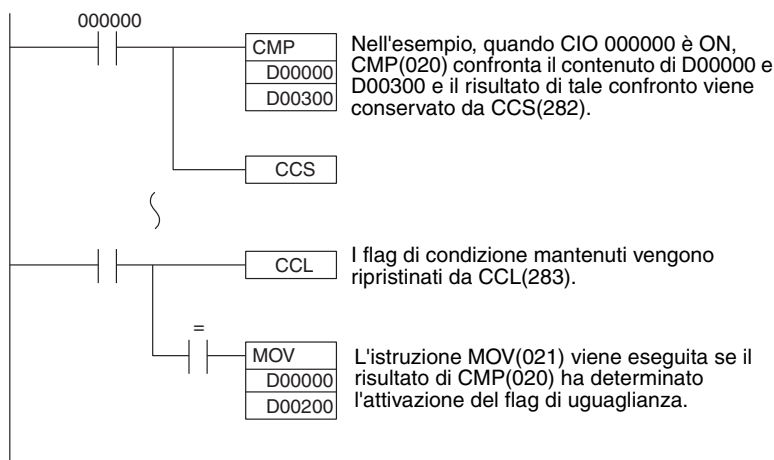
Il flag di uguaglianza riproduce il risultato dell'istruzione COMPARE e non quello dell'istruzione A.

Flag

Queste istruzioni non influiscono su alcun flag.

Esempi

Nell'esempio che segue, CCS(282) conserva i risultati di un confronto in modo tale da poterli utilizzare come condizione di esecuzione in una fase successiva del programma.

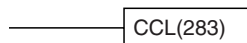


3-31-6 LOAD CONDITION FLAGS: CCL(283)

Ripristina lo stato dei flag di condizione salvati da CCS(282) in un'area separata del modulo CPU. Per cancellare i flag di condizione è inoltre possibile utilizzare CCL(283) indipendentemente.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CCL(283) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CCL(283) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

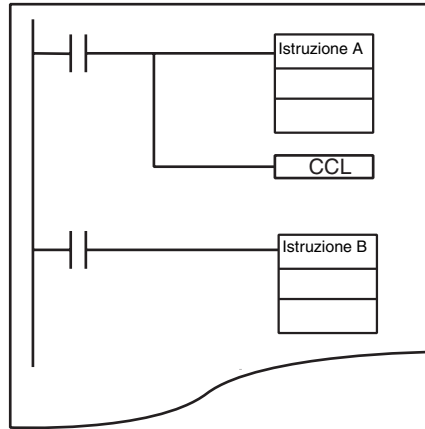
Descrizione

Se la condizione di esecuzione è ON, CCS(283) ripristina (legge) lo stato dei flag di condizione (esclusi i flag di sempre ON e sempre OFF). Viene ripristinato (letto) lo stato dei seguenti flag di condizione: ER, CY, >, =, <, N, OF, UF, >=, <> e <=.

I flag di condizione sono condivisi da tutte le istruzioni, per cui è possibile che il loro stato cambi più volte durante il ciclo del PLC all'esecuzione di ciascuna istruzione. In precedenza era necessario precisare le condizioni utilizzando i flag di condizione subito dopo l'istruzione di controllo, al fine di evitare che il loro stato venisse influenzato dalle istruzioni successive. Le istruzioni CCS(282) e CCL(283) consentono di separare l'istruzione di controllo dalle condizioni di esecuzione che dipendono dal risultato.

Ad esempio, CCS(282) è in grado di memorizzare lo stato del flag di uguaglianza al termine dell'esecuzione di un'istruzione di confronto e il risultato può essere ripristinato in un secondo tempo. Una volta eseguita l'istruzione, non è necessario utilizzare il risultato immediatamente.

Task



CCL(283) viene utilizzata separatamente dalle altre istruzioni per cancellare i flag di condizione al termine dell'esecuzione dell'istruzione A in modo tale che i risultati non influiscano sull'istruzione B e su quelle successive.

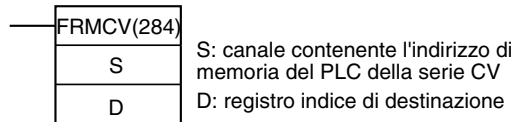
Per ulteriori esempi sull'utilizzo di CCS(282) e CCL(283), consultare 3-31-5 *SAVE CONDITION FLAGS: CCS(282)*.

Flag Queste istruzioni non influiscono su alcun flag.

3-31-7 CONVERT ADDRESS FROM CV: FRMVCV(284)

Scopo Converte l'indirizzo di memoria del PLC della serie CV nel corrispondente indirizzo di memoria del PLC della serie CS/CJ. FRMVCV(284) può risultare utile nella conversione di programmi della serie CV che utilizzano indirizzi di memoria del PLC per renderli compatibili ai PLC della serie CS/CJ. Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FRMVCV(284) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FRMVCV(284) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

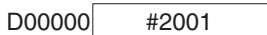
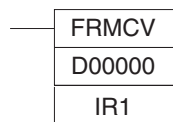
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Descrizione

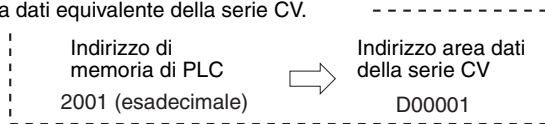
Quando la condizione di esecuzione è ON, FRMVCV(284) effettua le seguenti operazioni.

1. L'indirizzo di memoria del PLC della serie CV specificato in S viene convertito nell'indirizzo area dati equivalente della serie CV.
2. FRMVCV(284) determina l'indirizzo di memoria del PLC della serie CS/CJ che corrisponde allo stesso indirizzo area dati della serie CV.
3. L'indirizzo di memoria del PLC della serie CS/CJ viene inviato in D e un registro indice (da IR0 a IR15) deve essere specificato per D.

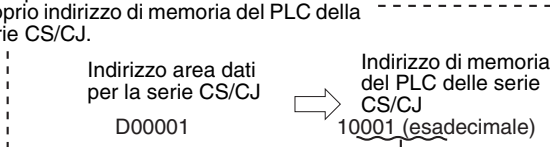
Nell'esempio seguente viene illustrato come utilizzare FRMVCV(284) per convertire l'indirizzo di memoria del PLC della serie CV per D00001.



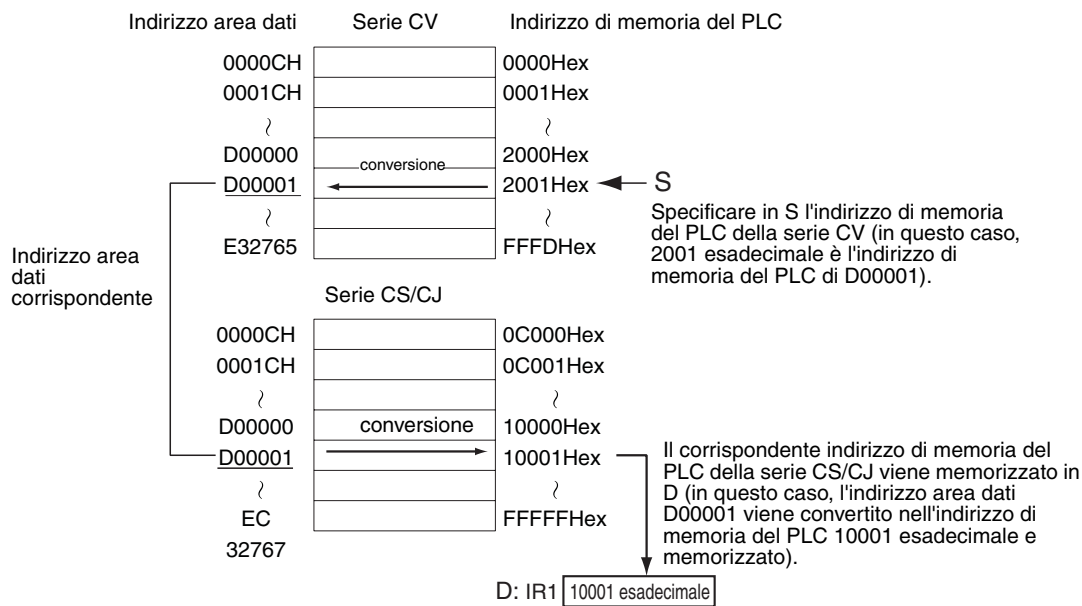
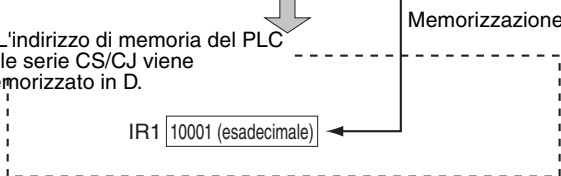
1. L'indirizzo di memoria del PLC della serie CV viene convertito nell'indirizzo area dati equivalente della serie CV.



2. L'indirizzo area dati della serie CV corrispondente viene convertito nel proprio indirizzo di memoria del PLC della serie CS/CJ.



3. L'indirizzo di memoria del PLC delle serie CS/CJ viene memorizzato in D.



Nota Se nella serie CS/CJ non c'è alcun equivalente all'indirizzo di memoria del PLC della serie CV specificato, si verifica un errore: viene attivato il flag di errore e l'indirizzo non sarà convertito.

Se come operando si specifica un registro indice con un prefisso "IR", l'istruzione verrà applicata al canale indicato dall'indirizzo di memoria del PLC nel registro indice, non al registro indice stesso. Una volta memorizzato l'indirizzo prescelto di memoria del PLC in un registro indice, è possibile utilizzare questo stesso registro in qualità di operando per un'istruzione.

È possibile specificare l'istruzione FRMCV(284) per convertire un programma della serie CV in uno dei due seguenti tipi di programmazione per utilizzarlo con PLC della serie CS/CJ. Per un approfondimento, leggere *Esempi* più avanti in questa sezione.

1. Con l'indirizzamento indiretto DM in modalità binaria (*DM)
(quando si specifica in modo indiretto un indirizzo area dati con un indirizzo di memoria del PLC in DM)
2. Con indirizzi di memoria del PLC della serie CV utilizzati direttamente come valori
(quando si memorizzano indirizzi di memoria del PLC nei registri indice con l'indirizzamento diretto mediante un'istruzione come MOV(021))

Caratteristiche operando

| Area | S | D |
|--|--|---------------|
| Area CIO | Da CIO 0000 a CIO 6143 | --- |
| Area di lavoro | Da W000 a W511 | --- |
| Area bit di ritenività | Da H000 a H511 | --- |
| Area bit ausiliaria | Da A448 ad A959 | --- |
| Area del temporizzatore | Da T0000 a T4095 | --- |
| Area del contatore | Da C0000 a C4095 | --- |
| Area DM | Da D00000 a D32767 | --- |
| Area EM senza banco | Da E00000 a E32767 | --- |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | --- |
| Costanti | Tutte le costanti tranne 09FF esadecimale, da 0A00 a 0AFF esadecimale o da 0D00 a 0E3F esadecimale | --- |
| Registri dati | Da DR0 a DR15 | --- |
| Registri indice | --- | Da IR0 a IR15 |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 | --- |

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se S specifica uno dei seguenti indirizzi di memoria del PLC inesistenti nelle serie CS/CJ: Area di memorizzazione temporanea (TR) (09FF esadecimale) Area del link CPU Bus (G) (da 0A00 a 0AFF esadecimale) Aree SFC (da 0D00 a 0E3F esadecimale) OFF in tutti gli altri casi. |

Esempi

Esempio 1: conversione di un programma della serie CV con indirizzamento indiretto DM in modalità binaria *DM

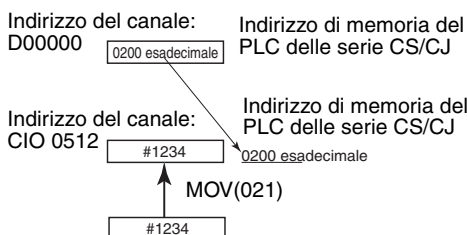
In questo esempio di FRMCV(284), un canale DM viene specificato in S, il corrispondente indirizzo di memoria del PLC è memorizzato in un registro indice e il registro indice stesso viene utilizzato per l'indirizzamento indiretto.

- Programma della serie CV (programmare utilizzando l'indirizzamento indiretto DM in modalità binaria)

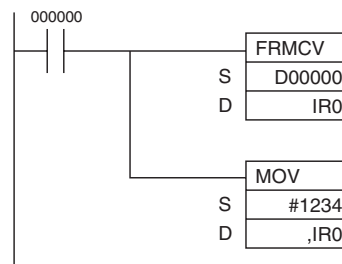


Configurazione del PLC
 Dati DM indiretti:
 quando gli indirizzi DM indiretti sono in formato binario, il contenuto del canale DM viene considerato come un indirizzo di memoria del PLC e specifica il corrispondente indirizzo in memoria I/O.

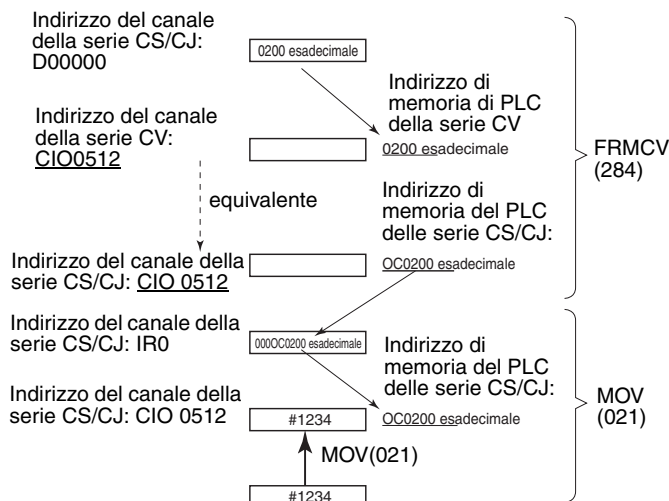
In questo caso, il valore in D00000 è 0200 esadecimale. Il corrispondente indirizzo area dati è CIO 0512, per cui #1234 viene trasferito a CIO 0512.



- Programma per la serie CS/CJ



In questo caso, il valore in D00000 è 0200 esadecimale. L'indirizzo area dati corrispondente per la serie CV è CIO 0512. L'indirizzo di memoria del PLC della serie CS/CJ per CIO 0512 è 0000C200 esadecimale; tale valore viene quindi memorizzato in IR0. L'operando di destinazione in MOV(021) indirizza indirettamente il contenuto di IR0, per cui #1234 viene trasferito in CIO 0512.



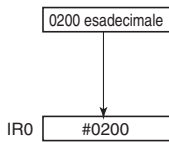
Esempio 2: conversione di un programma della serie CV con indirizzi di memoria del PLC memorizzati direttamente nei registri indice

In questo esempio di FRMCV(284), l'indirizzo di memoria del PLC della serie CV viene specificato direttamente in S.

- Programma della serie CV (programmare utilizzando gli indirizzi di memoria del PLC memorizzati direttamente in IR)

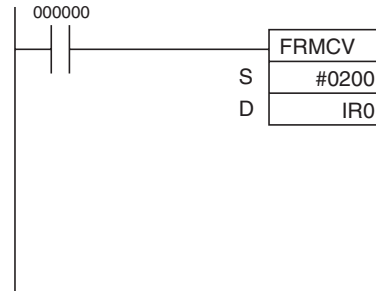


In questo caso, l'indirizzo di memoria del PLC 0200 esadecimale viene memorizzato nel registro indice IR0.

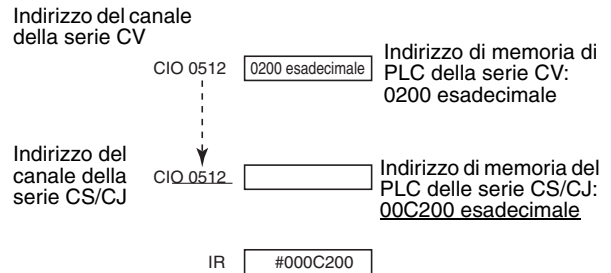


- Programma per la serie CS/CJ

Programma equivalente



In questo caso, l'indirizzo di memoria del PLC della serie CV 0200 esadecimale corrisponde a CIO 0512. L'indirizzo di memoria del PLC della serie CS/CJ per CIO 0512 è 0000C200 esadecimale; tale valore viene quindi memorizzato in IR0.



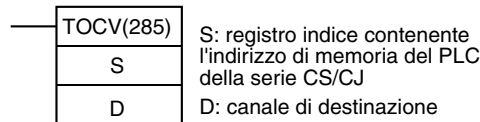
3-31-8 CONVERT ADDRESS TO CV: TOCV(285)

Scopo

Converte l'indirizzo di memoria del PLC della serie CS/CJ nell'indirizzo di memoria corrispondente del PLC della serie CV. TOCV(285) può risultare utile nella conversione di programmi della serie CS/CJ che utilizzano indirizzi di memoria del PLC per renderli compatibili ai PLC della serie CV.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H, CJ1M e CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | TOCV(285) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TOCV(285) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

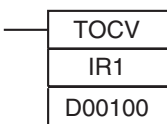
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Descrizione

Quando la condizione di esecuzione è ON, TOCV(285) effettua le seguenti operazioni.

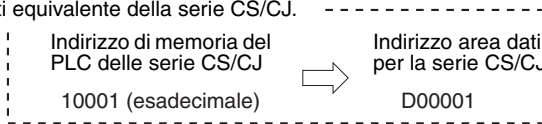
1. L'indirizzo di memoria del PLC della serie CS/CJ specificato in S viene convertito nell'indirizzo area dati equivalente della serie CS/CJ. Per S è necessario specificare un indirizzo indice (da IR0 a IR15).
2. TOCV(284) determina l'indirizzo di memoria del PLC della serie CV che corrisponde allo stesso indirizzo area dati della serie CS/CJ.
3. L'indirizzo di memoria del PLC della serie CV viene inviato in D.

Nell'esempio seguente viene illustrato come utilizzare TOCV(285) per convertire l'indirizzo di memoria del PLC della serie CS/CJ per D00001.

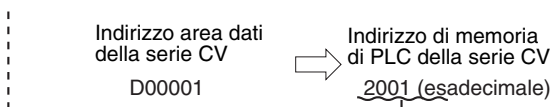


IR1 10001 (esadecimale)

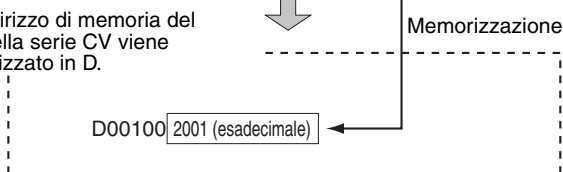
1. L'indirizzo di memoria del PLC della serie CS/CJ viene convertito nell'indirizzo area dati equivalente della serie CS/CJ.

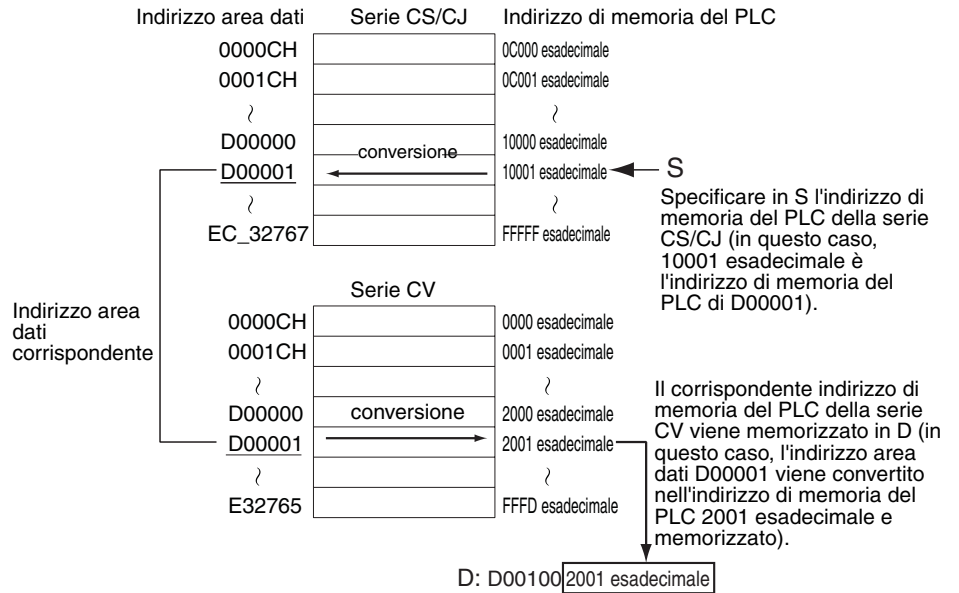


2. L'indirizzo area dati della serie CV corrispondente viene convertito nel proprio indirizzo di memoria del PLC della serie CV.



3. L'indirizzo di memoria del PLC della serie CV viene memorizzato in D.





- Nota**
1. Se nella serie CV non c'è alcun equivalente all'indirizzo di memoria del PLC della serie CS/CJ specificato, si verifica un errore: viene attivato il flag di errore e l'indirizzo non sarà convertito.
 2. I dati relativi all'indirizzo di memoria del PLC della serie CV memorizzati da TOCV(285) possono essere trasferiti mediante CX-Programmer in un PLC della serie CV.
 3. Nel programma della serie CV è possibile specificare lo stesso indirizzo area dati utilizzato nel programma della serie CS/CJ mediante l'indirizzamento indiretto del registro indice (prefisso ",IR") o l'indirizzamento indiretto DM in modalità binaria (*DM).

Caratteristiche operando

| Area | S | D |
|--------------------------------------|---------------|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A448 ad A959 |
| Area del temporizzatore | --- | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | Vedere nota 1 | --- |
| Registri dati | --- | Da DR0 a DR15 |

| Area | S | D |
|--|---------------|--|
| Registri indice | Da IR0 a IR15 | --- |
| Indirizzamento indiretto con i registri indice | --- | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 |

- Nota** 1. Se S indica uno dei seguenti indirizzi di memoria del PLC, inesistenti nella serie CV, si verificherà un errore e verrà attivato il flag di errore:

| Area o indirizzi | Indirizzi di memoria del PLC |
|------------------------|---|
| Area dei flag dei task | Da 0000 B800 a 0000 B801 esadecimale |
| Da A512 ad A959 | Da 0000 BA40 a 0000 BBFF esadecimale |
| Da CIO 2556 a CIO 6143 | Da 0000 C9FC a 0000 D7FF esadecimale |
| Da T1024 a T4095 | Da 0000 BE40 a 0000 BEFF esadecimale e da 0000 E400 a 0000 EFFF esadecimale |
| Da C1024 a C4095 | Da 0000 BF40 a 0000 BFFF esadecimale e da 0000 F400 a 0000 FFFF esadecimale |
| Area HR | Da 0000 D800 a 0000 D9FF esadecimale |
| Area WR | Da 0000 DE00 a 0000 DFFF esadecimale |
| Da D24576 a D32767 | Da 0001 6000 a 0001 7FFF esadecimale |
| Specifica del banco EM | Da 0001 8000 a 000F 7FFF esadecimale |
| Da E32766 a D32767 | Da 000F FFFE a 000F FFFF esadecimale |

2. Se per S viene specificata un'area diversa da quella del registro indice, si verificherà un errore e verrà attivato il flag di errore.

Flag

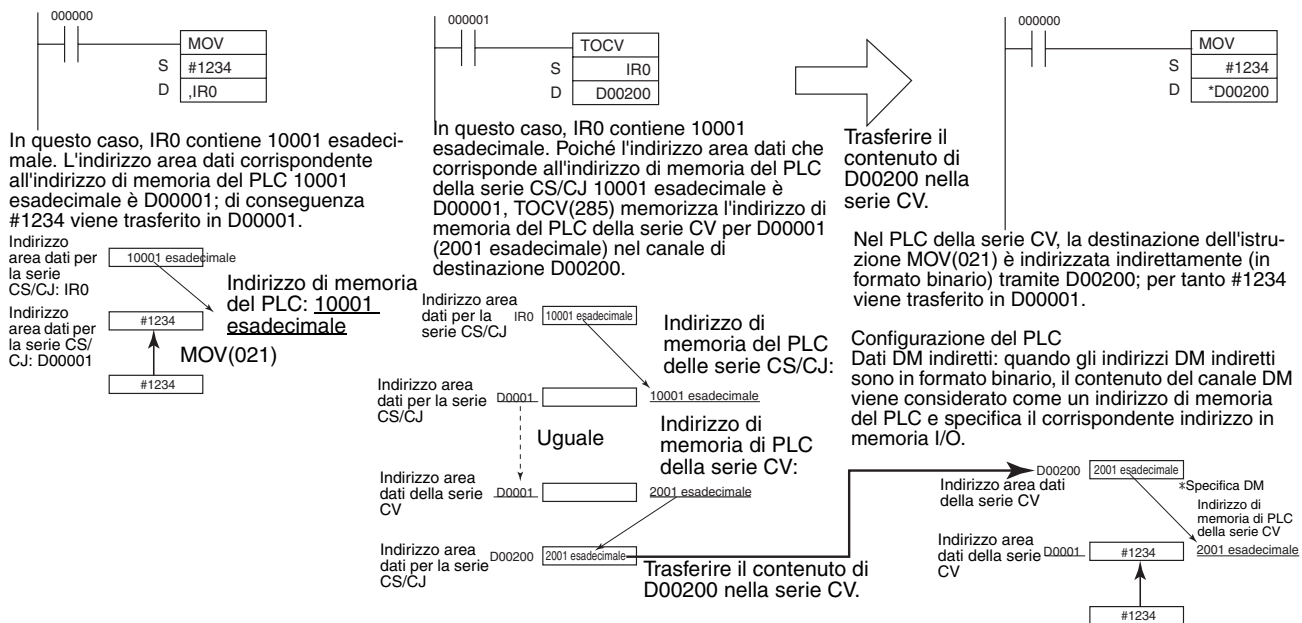
| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se S specifica un indirizzo di memoria del PLC inesistente nei PLC della serie CV. ON se S non è una costante o un registro indice. OFF in tutti gli altri casi. |

Esempio

Conversione di un programma della serie CS/CJ mediante l'indirizzamento indiretto del registro indice

- In questo esempio di TOCV(285), in S è specificato un indirizzo indice. L'indirizzo di memoria del PLC della serie CS/CJ in tale indice registro viene convertito nel suo equivalente della serie CV.
- L'indirizzo di memoria del PLC della serie CV viene trasferito nell'indirizzo area dati specificato.
- Utilizzare l'indirizzo di memoria del PLC della serie CV nel programma della serie CV.

- Programma per la serie CS/CJ (programmare utilizzando l'indirizzamento del registro indice)
- Programma per la serie CS/CJ
- Programma della serie CV



3-31-9 DISABLE PERIPHERAL SERVICING: IOSP(287) (solo CS1-H, CJ1-H e CJ1M)

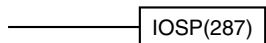
Scopo

Disabilita la gestione delle periferiche durante l'esecuzione del programma in modalità di elaborazione parallela o in modalità di gestione prioritaria delle periferiche.

Per ulteriori dettagli sulla modalità di elaborazione parallela o sulla modalità di gestione prioritaria delle periferiche, consultare *Sezione 6, Funzioni avanzate* nel *Manuale di programmazione del PLC della serie CS/CJ*.

Nota Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H e CJ1M. Non è possibile utilizzarla con moduli CPU CS1, CJ1 o CS1D.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | IOSP(287) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @IOSP(287) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

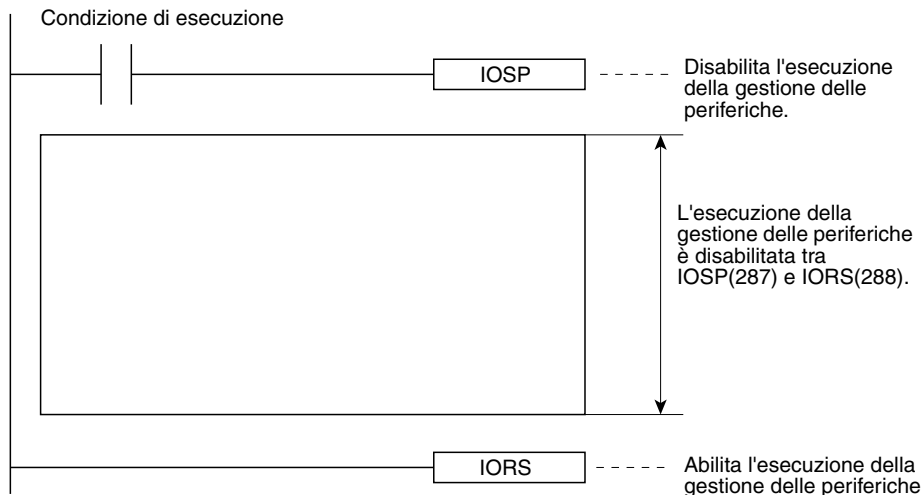
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Descrizione

Per disabilitare i tipi di gestione delle periferiche specificati qui di seguito, utilizzare IOSP(287) in un task ciclico in modalità di elaborazione parallela con accesso alla memoria sincrono o asincrono. La gestione delle periferiche verrà nuovamente abilitata all'esecuzione di IORS(288), l'istruzione ENABLE PERIPHERAL SERVICING.

- Gestione degli eventi con moduli di I/O speciali
- Gestione degli eventi con le unità bus CPU
- Gestione delle porte periferiche
- Gestione delle porte RS-232C
- Gestione eventi con schede interne (solo serie CS)
- Gestione eventi, comprendente anche l'elaborazione dell'istruzione in background, che utilizza un numero della porta di comunicazione, cioè una porta logica interna.



Una volta disabilitata mediante IOSP(287), la gestione delle periferiche rimane tale fino all'esecuzione di IORS(288) e END(001) oppure fino all'arresto del PLC.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se IOSP(287) viene eseguita in un task ad interrupt. OFF in tutti gli altri casi. |

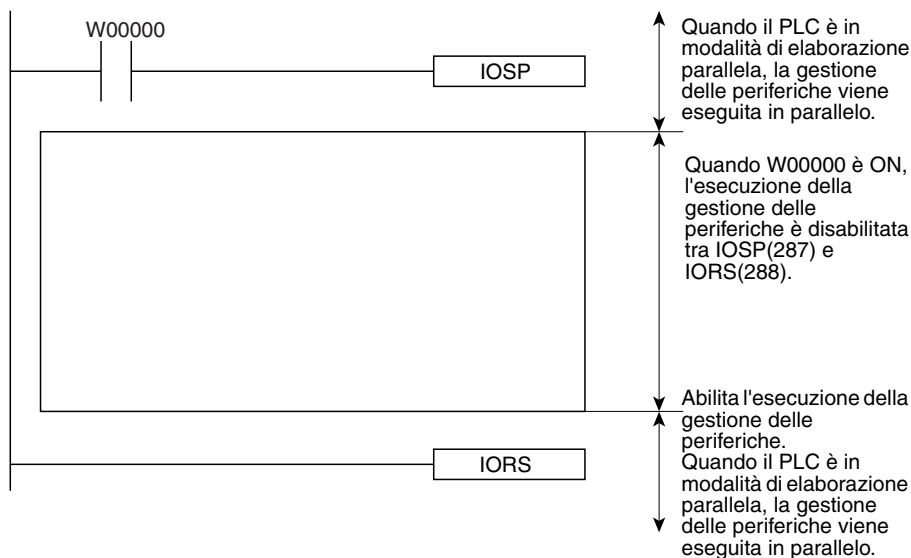
Avvertenze

Non è possibile eseguire IOSP(287) in un task ad interrupt. Se IOSP(287) viene eseguita in un task ad interrupt, si verificherà un errore e verrà attivato il flag di errore.

IOSP(287) non è in grado di disabilitare la gestione delle periferiche in più di un task. Per disabilitare la gestione delle periferiche in più di un task, programmare IOSP(287) in ciascuno dei task separatamente.

Esempio

Nell'esempio seguente viene illustrato l'utilizzo di IOSP(287) e IORS(288) per disabilitare la gestione delle periferiche in una sezione del programma.



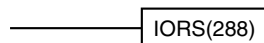
**3-31-10 ENABLE PERIPHERAL SERVICING: IORS(288)
(solo CS1-H, CJ1-H e CJ1M)**

Scopo

Durante l'esecuzione del programma in modalità di elaborazione parallela, abilita la gestione delle periferiche disabilitata tramite IOSP(287), l'istruzione DISABLE PERIPHERAL SERVICING.

Questa istruzione è supportata solo dalle CPU CS1-H, CJ1-H e CJ1M.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | IORS(288) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Descrizione

Per annullare la disabilitazione della gestione delle periferiche disposta da IOSP(287), l'istruzione DISABLE PERIPHERAL SERVICING, utilizzare IORS(288) in un task ciclico.

Non è necessario programmare IORS(288) con una condizione di esecuzione.

Non è possibile eseguire IORS(288) in un task ad interrupt. Se IORS(288) viene eseguita in un task ad interrupt, si verificherà un errore e verrà attivato il flag di errore.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se IORS(288) viene eseguita in un task ad interrupt. OFF in tutti gli altri casi. |

3-32 Istruzioni di programmazione a blocchi

In questa sezione vengono descritti i programmi a blocchi e le istruzioni di programmazione a blocchi.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|------------------------------|-----------------|-----------------|--------|
| BLOCK PROGRAM BEGIN | BPRG | 096 | 1128 |
| BLOCK PROGRAM END | BEND | 801 | 1128 |
| BLOCK PROGRAM PAUSE | BPPS | 811 | 1131 |
| BLOCK PROGRAM RESTART | BPRS | 812 | 1131 |
| CONDITIONAL BLOCK EXIT (NOT) | EXIT (NOT) | 806 | 1137 |
| IF (NOT) | IF (NOT) | 802 | 1133 |
| ELSE | ELSE | 803 | 1133 |
| IF END | IEND | 804 | 1133 |
| ONE CYCLE AND WAIT (NOT) | WAIT (NOT) | 805 | 1140 |
| TIMER WAIT | TIMW (BCD) | 813 | 1144 |
| | TIMWX (binario) | 816 | |
| COUNTER WAIT | CNTW (BCD) | 814 | 1147 |
| | CNTWX (binario) | 818 | |
| HIGH-SPEED TIMER WAIT | TMHW (BCD) | 817 | 1150 |
| | TMHWX (binario) | 815 | |
| LOOP | LOOP | 809 | 1153 |
| LOOP END (NOT) | LEND (NOT) | 810 | 1153 |

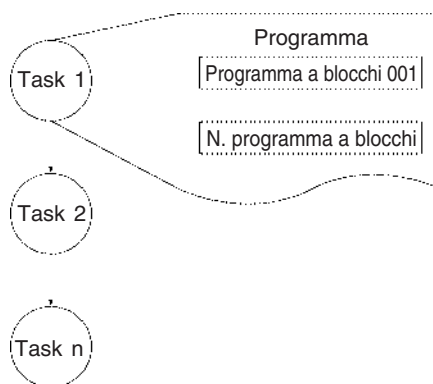
3-32-1 Introduzione

Programmi a blocchi

Fino a 128 programmi a blocchi nel programma utente completo (tutti i task) nella serie CS/CJ. L'esecuzione di ogni programma a blocchi viene controllata da un'unica condizione di esecuzione. Quando la condizione di esecuzione di BPRG(096) viene impostata su ON, tutte le istruzioni tra BPRG(096) e BEND(801) vengono eseguite incondizionatamente. Ad eccezione di BPRG(096), l'esecuzione di tutte le istruzioni di programmazione a blocchi non è influenzata dalla condizione di esecuzione. Ciò consente di raggruppare in un programma a blocchi la programmazione da eseguire con un'unica condizione di esecuzione.

Ciascun blocco viene avviato da una condizione di esecuzione nel diagramma ladder e tutte le istruzioni del blocco sono scritte in forma mnemonica. Il programma a blocchi è quindi una combinazione di istruzioni ladder e mnemoniche.

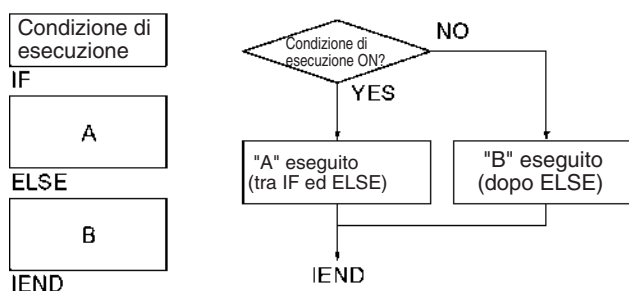
I programmi a blocchi consentono operazioni di programmazione che risulterebbero complesse con i diagrammi ladder, come le diramazioni condizionali e le progressioni step.



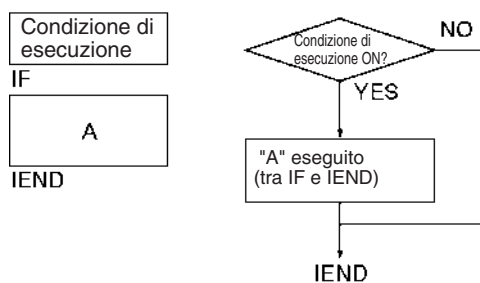
Utilizzo delle istruzioni di programmazione a blocchi

Fondamentalmente, IF(802), ELSE(803) e IEND(810) vengono utilizzate per condizioni di esecuzione e bit all'interno dei programmi a blocchi.

Se è necessario eseguire "A" o "B", IF A ELSE B IEND vengono utilizzate come illustrato qui di seguito.



Se è necessario eseguire "A" o niente, IF A IEND vengono utilizzate come illustrato qui di seguito.



Se l'esecuzione deve attendere che la condizione di esecuzione o il bit vengano impostati su ON (ad esempio nelle progressioni step), si utilizza WAIT(805).

Se l'esecuzione deve attendere per un periodo di tempo specificato, (ad esempio nelle progressioni step a tempo), utilizzare TIMW(813), TIMX(816), TMHW(815) o TMHWX(817).

Se l'esecuzione deve attendere che sia raggiunto un conteggio specificato (ad esempio nelle progressioni step con contatori), utilizzare CNTW(814)/CNTWX(818).

Se è necessario ripetere l'esecuzione fino a quando una condizione è soddisfatta nell'ambito di una parte di un programma a blocchi, utilizzare LOOP(809) e LEND(810).

Se l'esecuzione del programma a blocchi deve terminare a metà a causa di una condizione di esecuzione, utilizzare EXIT(806).

Se all'interno di un programma a blocchi è necessario mettere in pausa o riavviare un altro programma a blocchi in fase di esecuzione, utilizzare BPPS(811) e BPRS(812).

Istruzioni che assumono condizioni di esecuzione nei programmi a blocchi

Le istruzioni elencate qui di seguito possono assumere condizioni di esecuzione in un programma a blocchi.

| Tipo di istruzione | Nome dell'istruzione | Mnemonico |
|--|--------------------------|-----------------|
| Istruzioni di programmazione a blocchi | IF (NOT) | IF(802) (NOT) |
| | ONE CYCLE AND WAIT (NOT) | WAIT(805) (NOT) |
| | EXIT | EXIT(806) NOT |
| | LOOP END | LEND(810) NOT |
| Istruzioni del diagramma ladder | CONDITIONAL JUMP | CJP(510) |
| | CONDITIONAL JUMP NOT | CJPN(511) |

Istruzioni con limitazioni di applicazione nei programmi a blocchi

Le istruzioni elencate nella tabella seguente possono essere utilizzate esclusivamente per creare condizioni di esecuzione per IF(802), WAIT(805), EXIT(806), LEND(810), CJP(510) o CJPN(511) e non da sole. Se queste istruzioni vengono utilizzate da sole o in combinazione con qualsiasi altra istruzione, la loro esecuzione non è necessariamente prevedibile.

| Mnemonico | Nome |
|--------------------------------|--|
| LD/LD NOT | LOAD/LOAD NOT |
| AND/AND NOT | AND/AND NOT |
| OR/OR NOT | OR/OR NOT |
| UP/DOWN | CONDITION ON/CONDITION OFF |
| >, <=, >=, <=, <> (S) (L) | Istruzione di confronto con simboli (non istruzioni sul lato destro) |
| LD TST/TST NOT | Istruzioni test bit LOAD |
| AND TST/TST NOT | Istruzioni test bit AND |
| OR TST/TST NOT | Istruzioni test bit OR |
| >\$, <\$,=\$, >=\$, <=\$, <>\$ | Istruzioni di confronto delle stringhe di testo |

**Esempio corretto**

```
LD 000000
AND 000100
TST D00000 #0010
IF
```

Utilizzati come condizione di esecuzione per IF.

**Esempio scorretto**

```
LD 000000
AND 000100
TST D00000 #0010
MOV #0000 0010
```

Impossibile utilizzarli come condizione di esecuzione per MOV(021).

Istruzioni non disponibili nei programmi a blocchi

Le istruzioni che non possono essere utilizzate all'interno di programmi a blocchi sono elencate nella tabella seguente.

| Gruppo di istruzioni | Mnemonico | Nome | Alternativa |
|-------------------------------|-----------|--------------------|------------------------|
| Istruzioni di uscita sequenza | OUT | OUTPUT | Utilizzare SET e RSET. |
| | OUT NOT | OUTPUT NOT | |
| | DIFU(013) | DIFFERENTIATE UP | Nessuna |
| | DIFD(014) | DIFFERENTIATE DOWN | Nessuna |
| | KEEP(011) | KEEP | Nessuna |

| Gruppo di istruzioni | Mnemonico | Nome | Alternativa |
|--|------------------------|--|---|
| Istruzioni di controllo sequenza | FOR(512) e NEXT(513) | FOR-NEXT LOOPS | Utilizzare LOOP(809) e LEND(810) (NOT). |
| | BREAK(514) | BREAK LOOP | |
| | IL(002) e ILC(003) | INTERLOCK e INTERLOCK CLEAR | Dividere il programma a blocchi in blocchi di dimensioni inferiori. |
| | JMP(004)0 e JME(005) 0 | MULTIPLE JUMP e MULTIPLE JUMP END | Utilizzare JMP(004) e JME(005) (ma il salto viene fatto incondizionatamente). |
| | END(001) | END | Utilizzare BEND(801). |
| Istruzioni di temporizzatore e contatore | TIM | TIMER | Utilizzare TIMW(813), TIMWX(816), TMHW(815), TMHWX(817), CNTW(814) e CNTWX(818). Fino a quando funzionano il temporizzatore e il contatore, nel programma a blocchi non verranno eseguite altre istruzioni. |
| | TIMH(015) | HIGH-SPEED TIMER | |
| | TMHH(540) | ONE-MS TIMER | |
| | TTIM(087) | ACCUMULATIVE TIMER | |
| | TIML(542) | LONG TIMER | |
| | MTIM(543) | MULTI-OUTPUT TIMER | |
| | CNT | COUNTER | |
| | CNTR(012) | REVERSIBLE COUNTER | |
| Istruzioni di subroutine | SBN(092) e RET(093) | SUBROUTINE ENTRY e SUBROUTINE RETURN | Nessuna |
| Istruzioni di scorrimento | SFT(010) | SHIFT REGISTER | Utilizzare altre istruzioni di scorrimento. |
| Istruzioni di step | STEP(008) e SNXT(009) | STEP e STEP NEXT | Utilizzare WAIT(805) |
| Istruzioni di controllo dei dati | PID(190) | PID CONTROL | Nessuna |
| Istruzioni di diagnostica | FPD(269) | FAILURE POINT DETECTION | Nessuna |
| Istruzioni differenziate sul fronte di salita e di discesa | Mnemonici con @ | Istruzioni differenziate sul fronte di salita | Nessuna |
| | Mnemonici con % | Istruzioni differenziate sul fronte di discesa | Nessuna |

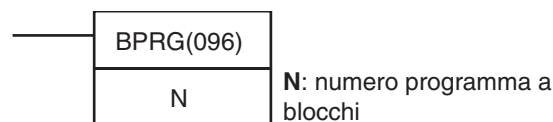
3-32-2 BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801)

Scopo

Definisce un'area di programmazione a blocchi. Per ogni istruzione BPRG(096) deve esistere un'istruzione BEND(801) corrispondente.

Simboli programmazione ladder

BLOCK PROGRAM BEGIN



BLOCK PROGRAM END

BEND(801)

Variazioni

BPRG(096)

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BPRG(096) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | Non supportata |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

BEND(801)

| | |
|------------|---|
| Variazioni | Eseguita sempre nel programma a blocchi |
|------------|---|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| (vedere nota) | OK | OK | OK |

Nota BPRG(096) è consentita una sola volta all'inizio di ogni programma a blocchi.

Operandi

N: numero del programma a blocchi

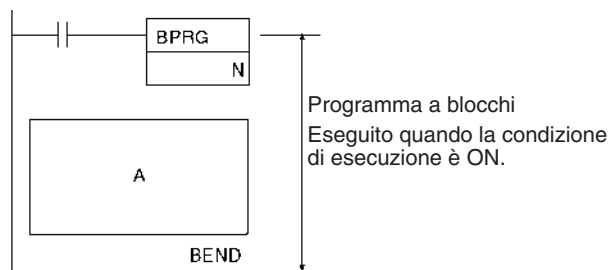
Il numero del programma a blocchi deve essere compreso tra 0 e 127 (decimale).

Caratteristiche operando (BPRG(096))

| Area | N |
|--|-----------------------|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 0 a 127 (decimale) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Descrizione

BPRG(096) esegue il programma a blocchi con il numero di blocco specificato in N, cioè quello immediatamente successivo e che termina con BEND(801). Tutte le istruzioni tra BPRG(096) e BEND(801) vengono eseguite con condizioni di esecuzione impostate su ON, vale a dire incondizionatamente.



Quando la condizione di esecuzione di BPRG(096) è OFF, il programma a blocchi non viene eseguito e per l'istruzione in tale programma a blocchi non sarà necessario alcun tempo di esecuzione.

Anche se la condizione di esecuzione di BPRG(096) è ON, è possibile interrompere l'esecuzione del programma a blocchi utilizzando BPPS(811) dall'interno di un altro programma a blocchi.

Flag

BPRG(096)

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se BPRG(096) è stata già eseguita. ON se N non è tra 0 e 127. ON se viene usato più di una volta lo stesso numero di programma a blocchi. OFF in tutti gli altri casi. |

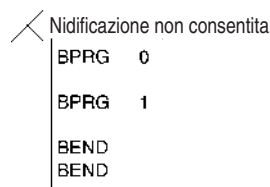
BEND(801)

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se un programma a blocchi non è in fase di esecuzione. OFF in tutti gli altri casi. |

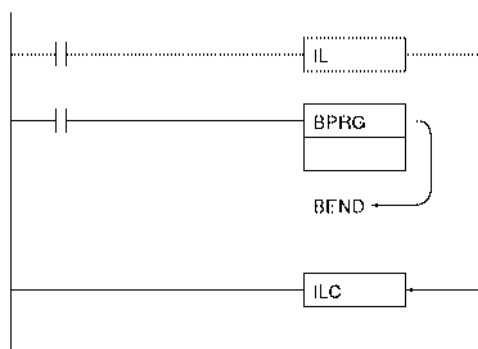
Avvertenze

All'interno del programma utente ciascun numero di programma a blocchi può essere utilizzato una sola volta.

I programmi a blocchi non possono venire nidificati.



Se il programma a blocchi è in una sezione del programma interbloccata e la condizione di esecuzione di IL(002) è OFF, il programma a blocchi non viene eseguito.

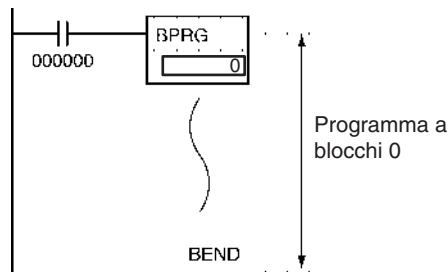


BPRG(096) e la corrispondente istruzione BEND(801) devono essere nello stesso task.

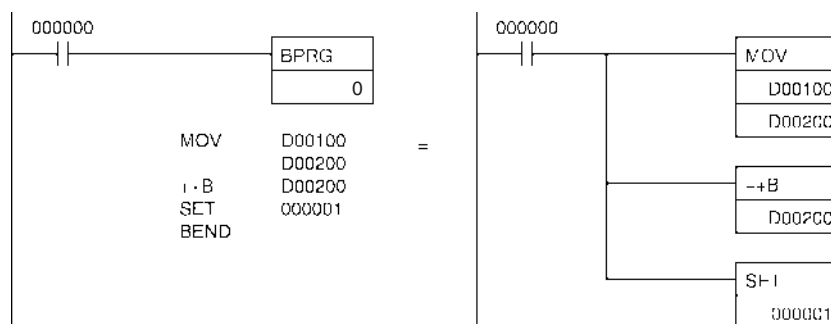
Nei seguenti casi si verifica un errore e si attiva il flag di errore: se BPRG(096) è nel mezzo di un programma a blocchi, se BEND(801) non è in un programma a blocchi, se N non è nell'intervallo compreso tra #0000 e #007F (binario), se non c'è alcun programma a blocchi o se lo stesso numero di programma a blocchi viene utilizzato più di una volta.

Esempi

Nell'esempio seguente, quando CIO 000000 è passa a ON, viene eseguito il programma a blocchi 0. Quando CIO 000000 è OFF, il programma a blocchi non verrà eseguito.



Entrambe le sezioni del programma indicate qui di seguito eseguono MOV(021), ++B(594) e SET con la stessa condizione di esecuzione, ossia quando CIO 000000 passa a ON.

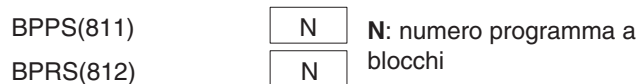


3-32-3 BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812)

Scopo

Mette in pausa e riavvia il programma a blocchi specificato da un altro programma a blocchi.

Simbolo programmazione ladder



Variazioni

| | |
|-------------------|--|
| Variazioni | Eseguita sempre nel programma a blocchi |
|-------------------|--|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota BPRG(096) e BPRS(812) devono essere utilizzate nelle aree di programmazione a blocchi anche per subroutine e task ad interrupt.

Operandi

N: numero del programma a blocchi

Il numero del programma a blocchi deve essere compreso tra 0 e 127 (decimale).

Caratteristiche operando

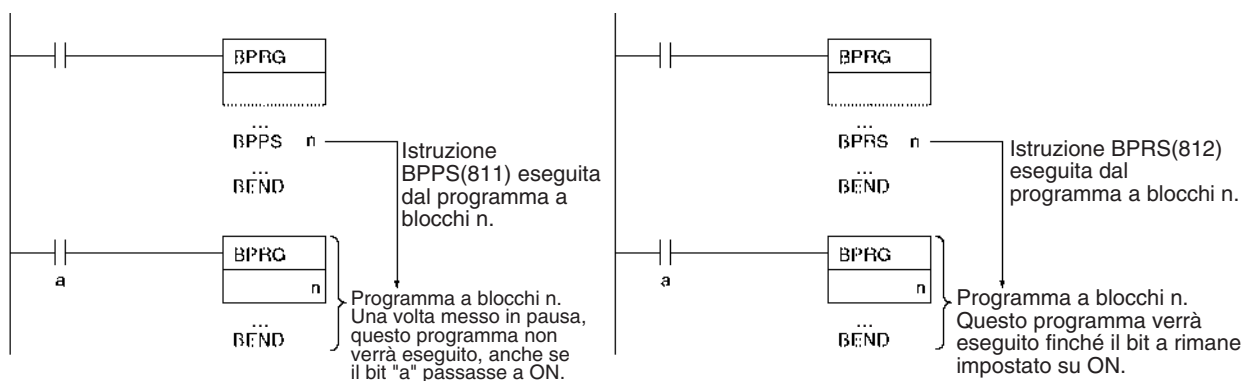
| Area | N |
|-------------------------|-----|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritentività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |

| Area | N |
|--|-----------------------|
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 0 a 127 (decimale) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Descrizione

BPPS(811) viene utilizzata all'interno di un programma a blocchi per mettere in pausa l'esecuzione di un altro programma a blocchi specificato dal numero del programma a blocchi N. Il programma a blocchi viene messo in pausa con BPPS(811) anche se l'istruzione BPRG(096) per il programma a blocchi ha la condizione di esecuzione impostata su ON. Il programma a blocchi non viene riavviato fino a che non viene eseguita BPRS(812).

BPRS(812) riavvia il programma a blocchi specificato dal numero del programma a blocchi N. Una volta riavviato, il programma a blocchi viene eseguito fino a quando la condizione di esecuzione dell'istruzione BPRG(096) per tale programma resta impostata su ON.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se BPPS(811) o BPRS(812) non sono in un programma a blocchi. ON se N non è tra 0 e 127. OFF in tutti gli altri casi. |

Avvertenze

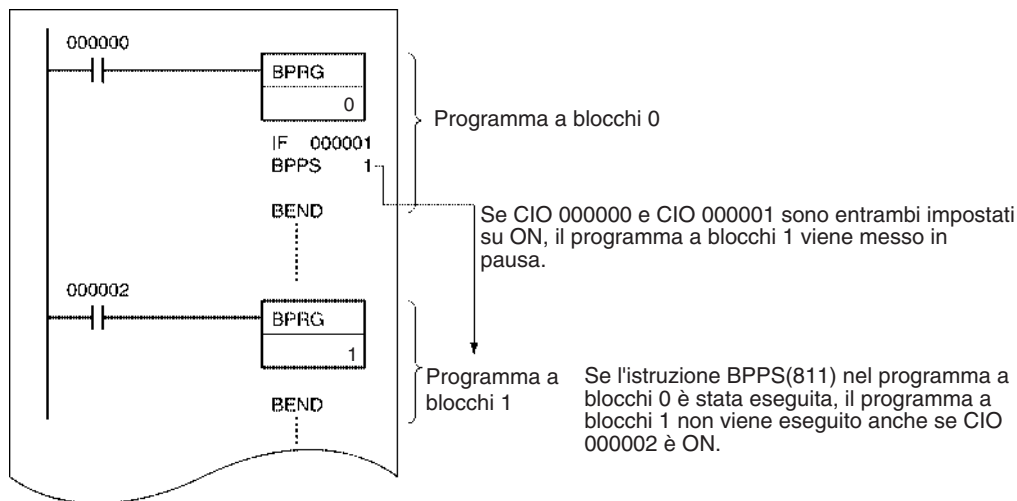
Se BPPS(811) o BPRS(812) non sono in un programma a blocchi o se N non è nell'intervallo compreso tra #0000 e #007F (binario), si verifica un errore e viene attivato il flag di errore.

BPPS(811) può essere utilizzata per mettere in pausa il programma a blocchi che la contiene. Quando il programma a blocchi viene riavviato utilizzando BPRS(812) da un altro programma a blocchi, il programma a blocchi messo in pausa ripartirà dall'istruzione successiva a BPPS(811).

Se un programma a blocchi messo in pausa contiene TIMW(813), TIMWX(816), TMHW(815) o TMHWX(817), il valore attuale del tempo continua a trascorrere anche durante la pausa del programma a blocchi.

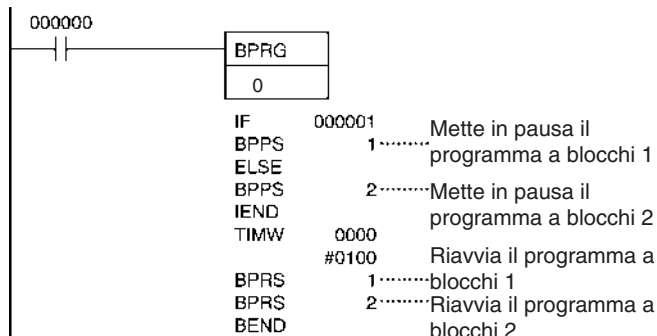
Esempi

L'esempio nel diagramma seguente illustra come mettere in pausa un programma a blocchi.



Nota Se il programma a blocchi che è in pausa compare dopo BPPS(811), non viene eseguito. Se il programma a blocchi compare prima di BPPS(811), viene messo in pausa all'avvio del ciclo successivo.

Se CIO 000000 è ON, il programma successivo mette in pausa l'esecuzione del programma a blocchi 1 o del programma a blocchi 2 in base allo stato di CIO 000001. Il programma a blocchi che è stato messo in pausa viene quindi riavviato dopo 10 secondi.



| Indirizzo | Istruzione | Operandi |
|-----------|------------|----------|
| 000000 | LD | 000000 |
| 000001 | BPRG(096) | 00 |
| 000002 | IF(802) | 000001 |
| 000003 | BPPS(811) | 01 |
| 000004 | ELSE(803) | |
| 000005 | BPPS(811) | 02 |
| 000006 | IEND(804) | |
| 000007 | TIMW(803) | 0000 |
| | | # 0100 |
| 000008 | BPRS(812) | 1 |
| 000009 | BPRS(812) | 2 |
| 000010 | BEND(801) | |

3-32-4 Diramazione: IF(802), ELSE(803) e IEND(804)

Scopo

Esegue la diramazione del programma a blocchi in base a una condizione di esecuzione o allo stato di un bit operando.

Simbolo programmazione

ladder

- IF(802) B B: bit operando
- IF(802)
- IF(802) NOT B
- ELSE(803)
- IEND(804)

Variazioni

| | |
|------------|---|
| Variazioni | Eseguita sempre nel programma a blocchi |
|------------|---|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota IF(802), ELSE(803) e IEND(804) devono essere utilizzate nelle aree di programmazione a blocchi anche per subroutine e task ad interrupt.

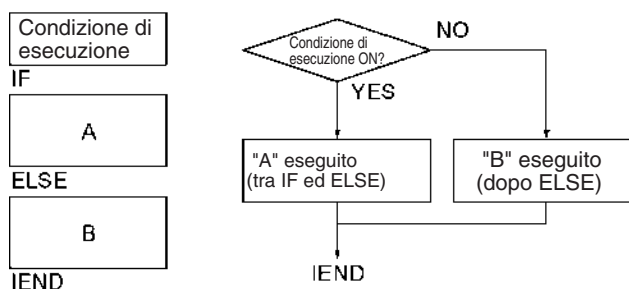
Caratteristiche operando

| Area | B |
|--|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit di ritentività | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A44715 Da A44800 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

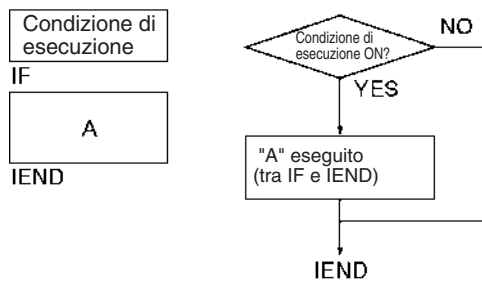
Descrizione

Funzionamento senza un operando per IF(802)

Se non viene specificato un bit operando, prima di IF(802) è necessario creare un'esecuzione che inizia con LD. Se la condizione di esecuzione è ON, vengono eseguite le istruzioni comprese tra IF(802) ed ELSE(803), mentre se la condizione di esecuzione è OFF, vengono eseguite le istruzioni tra ELSE(803) e IEND(804).

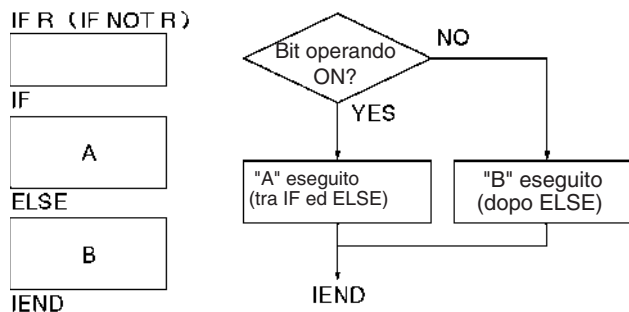


Se viene omessa l'istruzione ELSE(803) e la condizione di esecuzione è ON, sono eseguite le istruzioni comprese tra IF(802) e IEND(804), mentre se la condizione di esecuzione è OFF vengono eseguite soltanto le istruzioni successive a IEND(804).

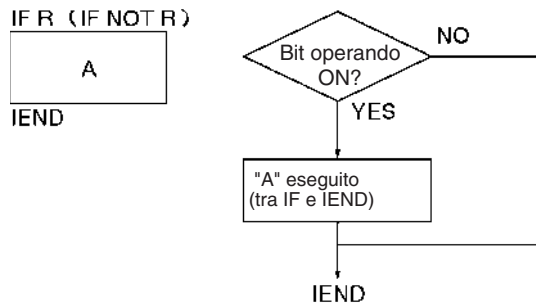


Funzionamento con un operando per IF(802) o IF NOT(802)

Per IF(802) o IF NOT(802) è possibile specificare un bit operando B. Se il bit operando è impostato su ON, vengono eseguite le istruzioni comprese tra IF(802) ed ELSE(803). Se il bit operando è impostato su OFF, vengono eseguite le istruzioni comprese tra ELSE(803) e IEND(804). Per IF NOT(802), vengono eseguite le istruzioni comprese tra IF(802) ed ELSE(803) se il bit operando è ON, mentre se è OFF, vengono eseguite le istruzioni tra ELSE(803) e IEND(804).



Se viene omessa l'istruzione ELSE(803) e il bit operando è ON, sono eseguite le istruzioni comprese tra IF(802) e IEND(804), mentre se tale bit è OFF, vengono eseguite soltanto le istruzioni successive a IEND(804). Se si utilizza IF NOT(802), si avrà la stessa situazione per lo stato opposto del bit operando.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se le istruzioni di diramazione non sono in un programma a blocchi. ON se le diramazioni nidificate sono più di 254. OFF in tutti gli altri casi. |

Avvertenze

Le istruzioni nei programmi a blocchi vengono in genere eseguite incondizionatamente. È tuttavia possibile utilizzare la diramazione per creare un'esecuzione condizionale basata sulle condizioni di esecuzione o sui bit operandi.

Per eseguire la diramazione tra A e B, utilizzare IF A ELSE B IEND.

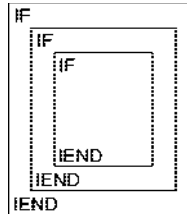
Per eseguire la diramazione solo in A, utilizzare IF A IEND.

Le diramazioni possono essere nidificate fino a un massimo di 253 livelli.

Se le istruzioni di diramazione non sono in un programma a blocchi o se vengono nidificate più di 254 diramazioni, si verifica un errore e viene attivato il flag di errore.

Nidificazione delle diramazioni

Nella diramazione di livello più elevato è possibile nidificare fino a 253 diramazioni.

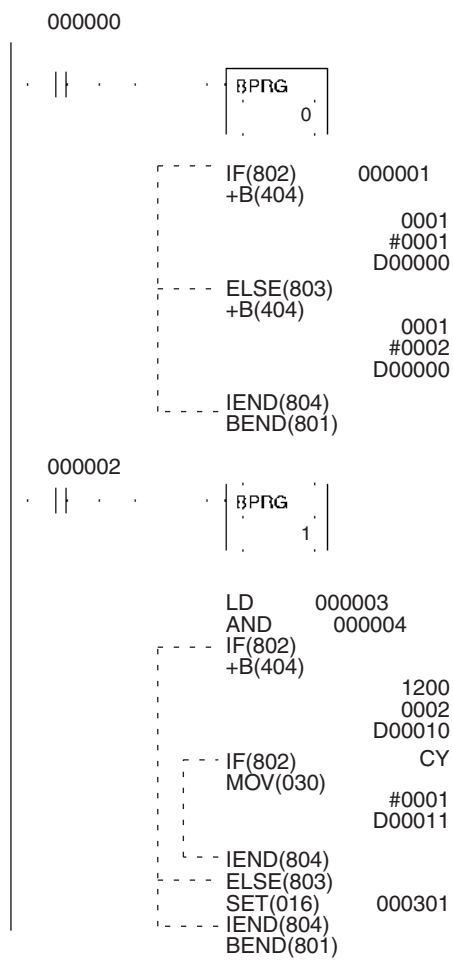


Esempi

Nell'esempio sono riportati due differenti programmi a blocchi controllati da CIO 000000 e CIO 000002.

In base allo stato di CIO 000001, il primo blocco esegue una di due addizioni. Il blocco viene eseguito quando lo stato di CIO 000000 è ON. Se CIO 000001 è ON, 0001 viene sommato al contenuto di CIO 0001. Se CIO 000001 è OFF, 0002 viene sommato al contenuto di CIO 0001. In entrambi i casi, il risultato viene inviato in D00000.

Il secondo blocco viene eseguito quando CIO 000002 è ON e indica la nidificazione di due livelli. Se CIO 000003 e CIO 000004 sono entrambi ON, il contenuto di CIO 1200 e quello di CIO 0002 vengono sommati e il risultato inviato in D00010; quindi 0001 viene spostato in D00011 in base allo stato di CY. Se CIO 000003 o CIO 000004 sono OFF, tutta l'operazione di addizione viene ignorata e CIO 000301 passa a ON.



| Indirizzo | Istruzione | Operandi |
|-----------|------------|----------|
| 000000 | LD | 000000 |
| 000001 | BPRG(096) | 00 |
| 000002 | IF(802) | 000001 |
| 000003 | +B(404) | |
| | | 0001 |
| | | #0001 |
| | | D00000 |
| 000004 | ELSE(803) | |
| 000005 | +B(404) | |
| | | 0001 |
| | | #0002 |
| | | D00000 |
| 000006 | IEND(804) | |
| 000007 | BEND(801) | |
| 000008 | LD | 000002 |
| 000009 | BPRG(096) | 1 |
| 000010 | LD | 000003 |
| 000011 | AND | 000004 |
| 000012 | IF(802) | |
| 000013 | +B(404) | |
| | | 1200 |
| | | 0002 |
| | | D00010 |
| | | CY |
| | | #0001 |
| | | D00011 |
| 000014 | IF(802) | A50004 |
| 000015 | MOV(030) | |
| | | #0001 |
| | | D00011 |
| 000016 | IEND(804) | |
| 000017 | ELSE(803) | |
| 000018 | SET(016) | 000301 |
| 000019 | IEND(804) | |
| 000020 | BEND(801) | |

3-32-5 CONDITIONAL BLOCK EXIT (NOT): EXIT (NOT)(806)

Scopo

Esce dal programma a blocchi, cioè nel programma a blocchi non esegue alcuna altra istruzione fino a BEND(801), in base allo stato del bit operando o della condizione di esecuzione. Se la condizione di esecuzione è ON, EXIT(806) senza un bit operando esce dal programma. Se il bit è ON, EXIT(806) con un bit operando esce dal programma. EXIT NOT(806) deve avere un bit operando e, se il bit è OFF, esce dal programma.

Simbolo programmazione ladder

EXIT(806)
 EXIT(806) B **B**: bit operando
 EXIT NOT(806) B

Variazioni

| | | |
|------------|---|---|
| Variazioni | Eseguita sempre nel programma a blocchi | EXIT(806) EXIT(806) B EXIT NOT(806) B |
|------------|---|---|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota EXIT(806) ed EXIT NOT(806) devono essere utilizzate nelle aree di programmazione a blocchi anche per subroutine e task ad interrupt.

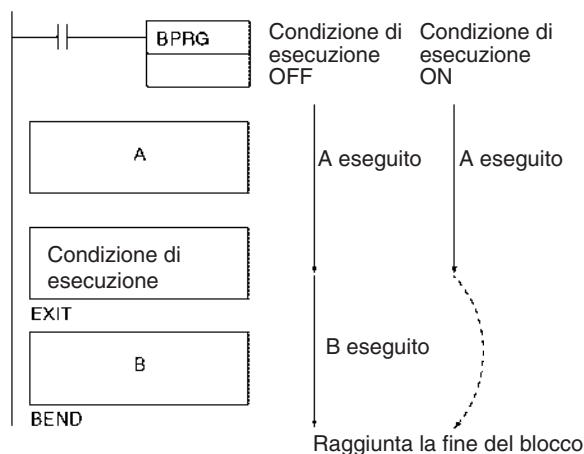
Caratteristiche operando

| Area | B |
|--|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit di ritentività | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A44715 Da A44800 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 |

Descrizione

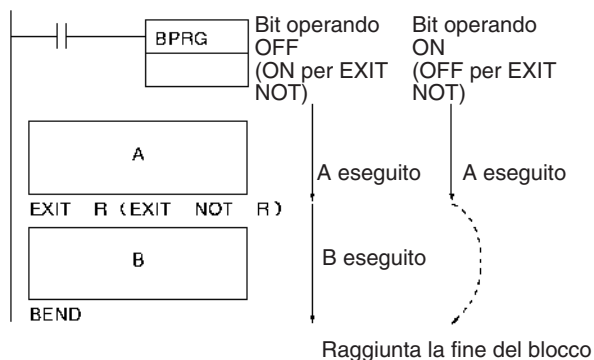
Funzionamento senza un operando

EXIT(806) può essere eseguita senza un operando. In questo caso, è necessario che per l'istruzione sia creata una condizione di esecuzione che inizia con LD. Se tale condizione è impostata su OFF, la parte rimanente del programma a blocchi verrà eseguita in modo normale. Se la condizione di esecuzione è ON, le istruzioni restanti nel programma a blocchi fino a BEND(801) non verranno eseguite.



Funzionamento con un operando

Se il bit operando B è impostato su OFF per EXIT(806), la parte restante del programma a blocchi verrà eseguita in modo normale. Se il bit operando è ON per EXIT (806), le istruzioni restanti nel programma a blocchi fino a BEND(801) non verranno eseguite. Per EXIT NOT(806), la parte restante del programma a blocchi viene eseguita se il bit operando è ON, mentre viene ignorata se tale bit è OFF.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se EXIT(806) o EXIT NOT(806) non sono in un programma a blocchi. OFF in tutti gli altri casi. |

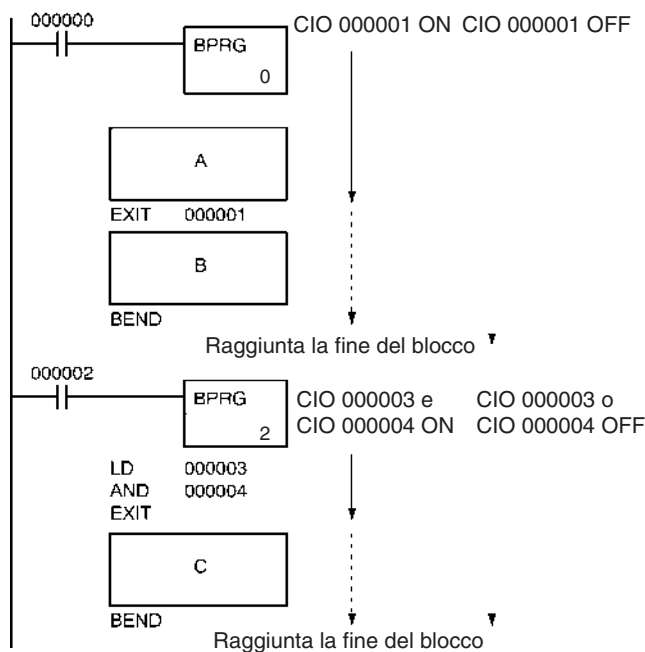
Avvertenze

Se EXIT(806) o EXIT NOT(806) non sono in un programma a blocchi, si verifica un errore e viene attivato il flag di errore.

Esempi

Quando CIO 000000 è OFF, il programma a blocchi viene eseguito. Se CIO 000001 è ON, A viene eseguito mentre B è ignorato e il programma di controllo passa a BEND(801). La sezione B del programma viene ignorata fino a quando CIO 000001 passa nuovamente a OFF.

Sebbene EXIT (NOT)(806) sia simile alla programmazione IF-IEND, il tempo di esecuzione è in genere più breve perché le istruzioni da EXIT (NOT)(806) alla fine del programma a blocchi non vengono eseguite.



3-32-6 ONE CYCLE AND WAIT (NOT): WAIT(805)/WAIT(805) NOT

Scopo Interrompe l'esecuzione della parte restante del programma a blocchi fino a quando la condizione di esecuzione passa a ON oppure un bit operando passa da ON a OFF o viceversa.

Simbolo programmazione ladder

WAIT(805)
 WAIT(805) B **B**: bit operando
 WAIT(805) NOT B

Variazioni

| | |
|-------------------|--|
| Variazioni | Eseguita sempre nel programma a blocchi |
|-------------------|--|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota WAIT(805) e WAIT(805) NOT devono essere utilizzate nelle aree di programmazione a blocchi anche per subroutine e task ad interrupt.

Caratteristiche operando

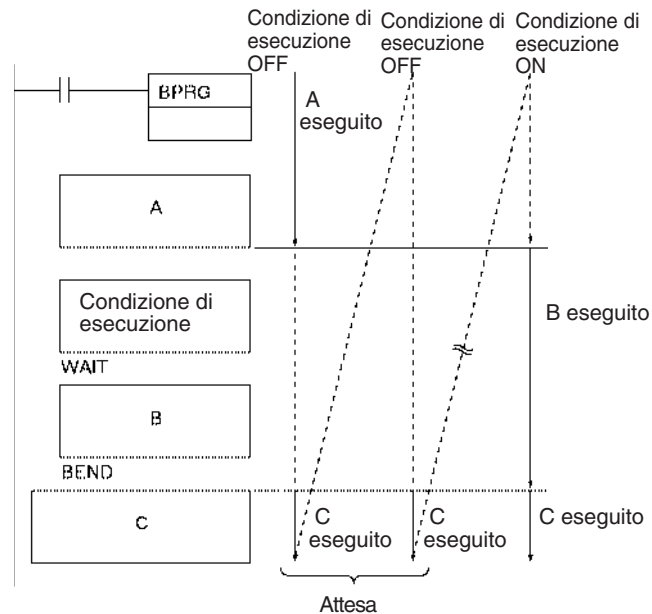
| Area | B |
|-------------------------|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit di ritenività | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A44715 Da A44800 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |

| Area | B |
|--|--|
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

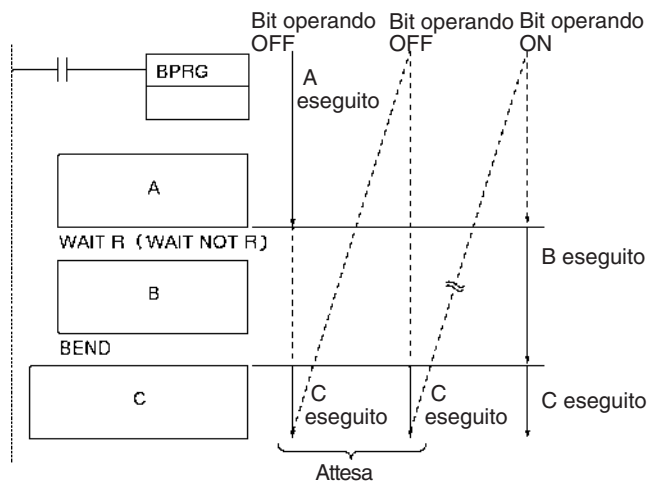
Funzionamento senza un operando

Se non è specificato un bit operando, prima di WAIT(805)/WAIT(805) NOT è necessario creare un'esecuzione che inizia con LD. Se la condizione di esecuzione per WAIT(805) è ON, il resto dell'istruzione nel programma a blocchi viene ignorato. Nel ciclo successivo non verrà eseguita alcuna istruzione del programma a blocchi, ad eccezione della condizione di esecuzione per WAIT(805). Quando la condizione di esecuzione passa a ON, verranno eseguite le istruzioni da WAIT(805) fino alla fine del programma.



Funzionamento con un operando

Per WAIT(805) o WAIT NOT(805) è possibile specificare un bit operando B. Se il bit operando è impostato su OFF (ON per WAIT(805) NOT), le istruzioni rimanenti del programma a blocchi verranno ignorate. Nel ciclo successivo non verrà eseguita alcuna istruzione del programma a blocchi, ad eccezione della condizione di esecuzione per WAIT(805) o WAIT(805) NOT. Quando la condizione di esecuzione passa a ON (OFF per WAIT(805) NOT), verranno eseguite le istruzioni da WAIT(805) o WAIT(805) NOT fino alla fine del programma.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se WAIT(805) o WAIT(805) NOT non sono in un programma a blocchi. OFF in tutti gli altri casi. |

Avvertenze

WAIT(805) e WAIT(805) NOT possono essere utilizzate per progressioni step all'interno dei programmi a blocchi.

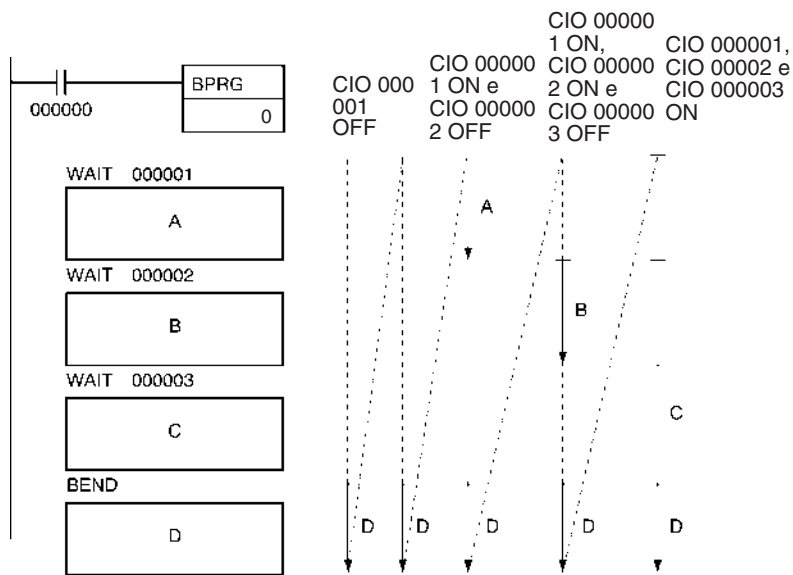
Se WAIT(805) o WAIT(805) NOT non sono in un programma a blocchi, si verifica un errore e viene attivato il flag di errore.

Nota Gli indirizzi di programma delle istruzioni WAIT con operandi specificati e gli indirizzi di programma della prima istruzione che crea le condizioni di esecuzione per le istruzioni WAIT senza operandi vengono registrati in memoria per consentire il proseguimento dell'esecuzione in base alla condizione di esecuzione o al bit operando. Se si esegue la modifica in linea da un dispositivo periferico, tuttavia, lo stato di WAIT è cancellato e il programma a blocchi viene nuovamente eseguito dall'inizio.

Esempi

Nell'esempio seguente, quando CIO 000000 è ON, viene eseguito il programma a blocchi 00. L'esecuzione avviene in questo modo:

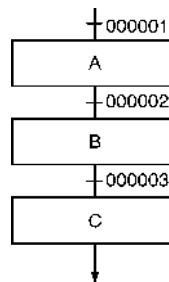
- 1,2,3... 1. Fino a quando CIO 000001 resta impostato su OFF, non verrà eseguito alcuno dei programmi a blocchi. Quando CIO 000001 passa a ON, "A" viene eseguito.
2. Se CIO 000002 è OFF dopo l'esecuzione di "A", la parte restante del programma a blocchi non viene eseguita fino a quando CIO 000002 passa a ON. Quando CIO 000002 passa a ON, "B" viene eseguito.
3. Se CIO 000003 è OFF dopo l'esecuzione di "B", la parte restante del programma a blocchi non viene eseguita fino a quando CIO 000003 passa a ON. Quando CIO 000003 passa a ON, "C" viene eseguito e l'elaborazione dell'esecuzione viene ripetuta.



Nella tabella seguente è riportata la relazione tra i bit operandi e l'esecuzione del programma a blocchi.

| Bit operandi | | | Esecuzione del programma | | |
|--------------|------------|------------|------------------------------------|--|--|
| CIO 000001 | CIO 000002 | CIO 000003 | Primo ciclo in cui CIO 000000 è ON | Ciclo successivo | Cicli seguenti |
| OFF | ON/OFF | ON/OFF | Nessuna esecuzione | Nessuna esecuzione; in attesa di CIO 000001. | Quando CIO 000001 passa a ON, "A" viene eseguito e lo stato di CIO 000002 viene controllato. |
| ON | OFF | ON/OFF | "A" eseguito | In attesa di CIO 000002 | Quando CIO 000002 passa a ON, "B" viene eseguito e lo stato di CIO 000003 viene controllato. |
| ON | ON | OFF | "A" e "B" vengono eseguiti | In attesa di CIO 000003 | Quando CIO 000003 passa a ON, "C" viene eseguito |
| ON | ON | ON | "A", "B" e "C" vengono eseguiti | "A", "B" e "C" vengono eseguiti | |

Come illustrato nell'esempio, WAIT(805) e WAIT(805) NOT possono essere utilizzate per eseguire step all'interno di un programma a blocchi.



3-32-7 TIMER WAIT: TIMW(813) and TIMWX(816)

Scopo Ritarda l'esecuzione delle rimanenti istruzioni del programma a blocchi del periodo di tempo specificato. Al completamento del conteggio, l'esecuzione riprende dall'istruzione successiva a TIMW(813)/TIMWX(816).

Simbolo programmazione ladder

Metodo di aggiornamento PV: BCD

TIMW(813) N N: numero di temporizzatore
 SV SV: valore impostato

Metodo di aggiornamento PV: binario

TIMWX(816) N N: numero di temporizzatore
 SV SV: valore impostato

Variazioni

| | |
|-------------------|--|
| Variazioni | Eseguita sempre nel programma a blocchi |
|-------------------|--|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Nota TIMW(813) e TIMWX(816) devono essere utilizzate nelle aree di programmazione a blocchi anche per le subroutine.

Operandi

N: numero del temporizzatore

BCD: da 0 a 4095 (decimale)

Binario: da 0 a 4095 (decimale)

S: valore impostato

BCD: da #0000 a #9999 (BCD)

Binario: da &0 a &65535 (decimale)

da #0000 a #FFFF (esadecimale)

Caratteristiche operando

| Area | N | SV |
|--------------------------------------|----------------|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritentività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A447 Da A448 a A959 |
| Area del temporizzatore | Da 0000 a 4095 | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |

| Area | N | SV |
|--|--|--|
| Costanti | --- | BCD: da #0000 a 9999 (BCD). Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

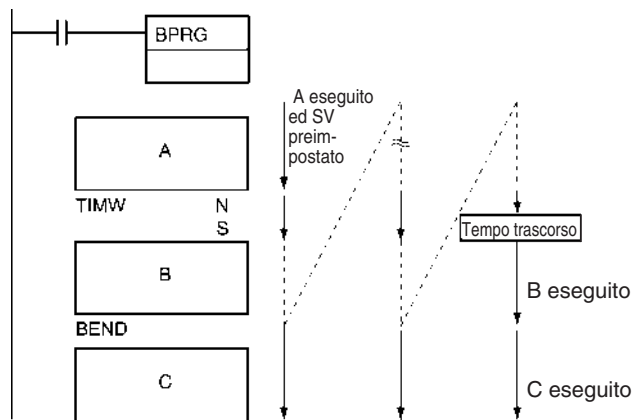
TIMW(813) e TIMWX(816) creano un temporizzatore con conteggio alla rovescia in ritardo all'eccitazione (temporizzatore a 100 ms impostato nel valore impostato SV) tra l'esecuzione dell'istruzione del programma a blocchi che le precede e le istruzioni successive. TIMW(813) può cronometrare da 0 a 999,9 s con una precisione del temporizzatore compresa tra 0 e 0,01 s. TIMWX(816) può cronometrare da 0 a 6.553,5 s con una precisione del temporizzatore compresa tra 0 e 0,01 s.

Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

La prima parte del programma a blocchi è eseguita quando il programma a blocchi viene specificato per la prima volta. Al raggiungimento di TIMW(813)/TIMWX(816), il flag di completamento è reimpostato su OFF, il temporizzatore viene preimpostato in base al valore impostato e l'esecuzione della parte restante del programma a blocchi deve attendere fino a che il valore impostato è trascorso.

Per aggiornare il temporizzatore in fase di decremento, è possibile eseguire solo TIMW(813) o TIMWX(816). Allo scadere del temporizzatore, è attivato il flag di completamento e la parte restante del programma a blocchi viene eseguita. Una volta eseguito il programma a blocchi completo, l'elaborazione viene ripetuta.

TIMW(813) e TIMWX(816) possono essere considerate come istruzioni WAIT con un temporizzatore per la condizione di esecuzione ed è quindi possibile utilizzarle per progressioni step a tempo.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se TIMW(813)/TIMWX(816) non sono in un programma a blocchi. ON se viene utilizzata un'identificazione IR indiretta per N in modalità BCD e l'indirizzo non è per il valore attuale di un temporizzatore. ON se in modalità BCD con SV non BCD. OFF in tutti gli altri casi. |

Avvertenze

Se il flag di completamento per il temporizzatore è sottoposto a impostazione forzata, la parte restante del programma a blocchi che segue il temporizzatore viene eseguita.

Se il flag di completamento per il temporizzatore è sottoposto a ripristino forzato, nel programma a blocchi viene eseguita soltanto TIMW(813) o TIMWX(816) fino a quando tale stato di ripristino non viene cancellato.

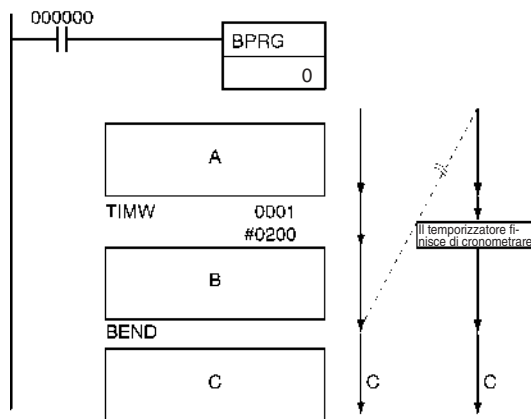
Anche quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 0000 a 2047 viene aggiornato. Quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 2048 a 4095 viene mantenuto.

I numeri di temporizzatore vengono utilizzati anche da altre istruzioni del temporizzatore. Se si utilizza lo stesso numero di temporizzatore per più di un'istruzione del temporizzatore, non è possibile prevedere il funzionamento. Utilizzare ciascun numero di temporizzatore una sola volta. L'unica occasione in cui è possibile un utilizzo attendibile dello stesso numero di temporizzatore si verifica quando è in funzione solo uno dei temporizzatori. Se si utilizza lo stesso numero di temporizzatore in più di un'istruzione del temporizzatore, si verifica un errore nel controllo del programma.

Se viene utilizzata un'identificazione IR indiretta per N in modalità BCD e l'indirizzo non è per il valore attuale di un temporizzatore o se il valore impostato non è BCD, si verificherà un errore e verrà attivato il flag di errore.

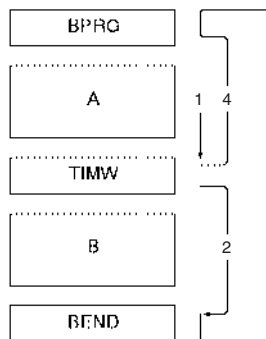
Esempi

Nell'esempio che segue, "B" viene eseguito 20 secondi dopo "A" ogni volta che CIO 000000 è ON.



| Indirizzo | Istruzione | Operando |
|-----------|------------|----------|
| 000200 | LD | 000000 |
| 000201 | BPRG | 0 |
| . | A | . |
| . | | . |
| 000210 | TIMW | 0001 |
| | | #0200 |
| . | B | . |
| . | | . |
| 000220 | BEND | --- |

L'esecuzione del programma procede da 2 a 3 a 4 e poi di nuovo a 2 nei 20 s precedenti all'esecuzione di "B", come illustrato nel diagramma qui di seguito.



3-32-8 COUNTER WAIT: CNTW(814) e CNTWX(818)

Scopo Ritarda l'esecuzione delle rimanenti istruzioni del programma a blocchi fino al completamento del conteggio specificato. L'esecuzione riprende dall'istruzione dopo CNTW(814)/CNTWX(818) al completamento del conteggio.

Simbolo programmazione ladder **Metodo di aggiornamento PV: BCD**

CNTW(814) N **N:** numero di contatore
 SV **SV:** valore impostato
 I **I:** ingresso conteggio

Metodo di aggiornamento PV: binario

CNTWX(818) N **N:** numero di contatore
 SV **SV:** valore impostato
 I **I:** ingresso conteggio

Variazioni

| | |
|-------------------|--|
| Variazioni | Eseguita sempre nel programma a blocchi |
|-------------------|--|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota CNTW(814) e CNTWX(818) devono essere utilizzate nelle aree di programmazione a blocchi anche per subroutine e task ad interrupt.

Operandi

N: Numero contatore

BCD: da 0 a 4095 (decimale)

Binario: da 0 a 4095 (decimale)

S: valore impostato

BCD: da #0000 a #9999 (BCD)

Binario: da &0 a &65535 (decimale)

da #0000 a #FFFF (esadecimale)

Caratteristiche operando

| Area | N | SV | I |
|--|--|---|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 | Da CIO 000000 a CIO 614315 |
| Area di lavoro | --- | Da W000 a W511 | Da W00000 a W51115 |
| Area bit di ritentività | --- | Da H000 a H511 | Da H00000 a H51115 |
| Area bit ausiliaria | --- | Da A000 ad A447 Da A448 ad A959 | Da A00000 ad A44715 Da A44800 ad A95915 |
| Area del temporizzatore | --- | Da T0000 a T4095 | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 | Da C0000 a C4095 | Da C0000 a C4095 |
| Flag dei task | --- | | Da TK0000 a TK0031 |
| Flag di condizione | --- | | ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | --- | | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area DM | --- | Da D00000 a D32767 | --- |
| Area EM senza banco | --- | Da E00000 a E32767 | --- |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | --- |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | --- |
| Costanti | --- | BCD: da #0000 a 9999 (BCD). Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) | --- |
| Registri dati | --- | Da DR0 a DR15 | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

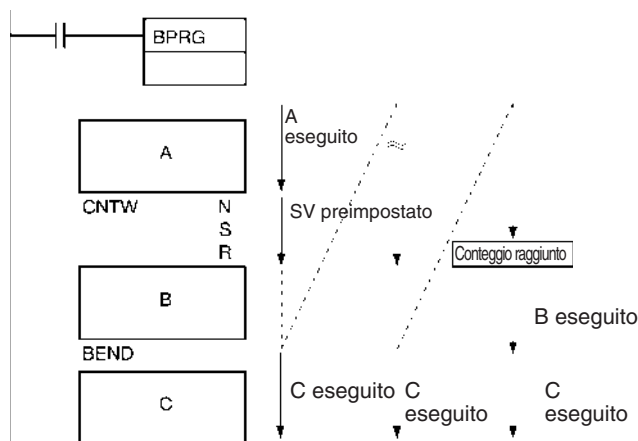
Descrizione

CNTW(814) e CNTWX(818) creano un contatore decrementale che ritarda l'esecuzione delle istruzioni che le seguono nel programma a blocchi fino al completamento del conteggio. Il valore impostato per CNTW(814) è specificato in BCD tra 0000 e 9999, mentre quello per CNTWX(818) è indicato in formato binario tra 0000 ed FFFF esadecimale.

La prima parte del programma a blocchi è eseguita quando il programma a blocchi viene specificato per la prima volta. Al raggiungimento di CNTW(814)/CNTWX(818), il flag di completamento è reimpostato su 0, il contatore viene preimpostato in base al valore impostato e l'esecuzione della parte restante del programma a blocchi deve attendere fino al completamento del conteggio. Il contatore conteggia gli impulsi (differenziazione sul fronte di salita) sull'ingresso contatore I.

Per aggiornare il contatore in fase di decremento, è possibile eseguire solo CNTW(814) e CNTWX(818). Allo scadere del contatore, è attivato il flag di completamento e la parte restante del programma a blocchi viene eseguita. Una volta eseguito il programma a blocchi completo, l'elaborazione viene ripetuta.

CNTW(814) e CNTWX(818) possono essere considerate come istruzioni WAIT con un contatore per la condizione di esecuzione ed è quindi possibile utilizzarle per progressioni step a tempo.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se CNTW(814)/CNTWX(818) non sono in un programma a blocchi. ON se viene utilizzata un'identificazione IR indiretta per N in modalità BCD e l'indirizzo non è per il valore attuale di un contatore. ON se, impostata la modalità BCD, il valore impostato non è BCD. OFF in tutti gli altri casi. |

Avvertenze

Se il flag di completamento per il contatore è sottoposto a impostazione forzata, la parte restante del programma a blocchi che segue CNTW(814)/CNTWX(818) viene eseguita.

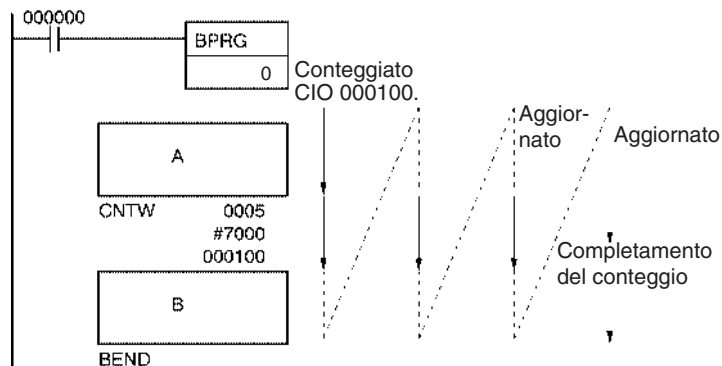
Se il flag di completamento per il contatore è sottoposto a ripristino forzato, nel programma a blocchi vengono eseguite soltanto le istruzioni CNTW(814)/CNTWX(818) fino a quando tale stato di ripristino non viene cancellato.

I numeri di contatore vengono utilizzati anche da altre istruzioni del contatore. Se si utilizza lo stesso numero di contatore per più di un'istruzione del contatore, non è possibile prevedere il funzionamento. Utilizzare ciascun numero di contatore una sola volta. L'unica occasione in cui è possibile un utilizzo attendibile dello stesso numero di contatore si verifica quando è in funzione solo uno dei contatori. Se si utilizza lo stesso numero di contatore in più di un'istruzione del contatore, si verifica un errore nel controllo del programma.

Se viene utilizzata un'identificazione IR indiretta per N in modalità BCD e l'indirizzo non è per il valore attuale di un contatore o se, impostata la modalità BCD, il valore impostato non è in tale formato, si verificherà un errore e verrà attivato il flag di errore.

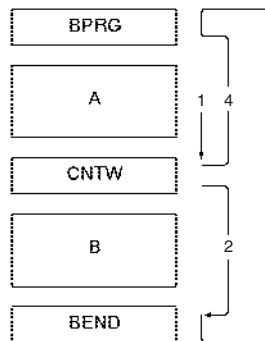
Esempi

Nell'esempio che segue, quando CIO 000000 è ON, "A" viene eseguito, mentre l'esecuzione del resto del programma a blocchi "B" dovrà attendere fino a 7.000 conteggi di CIO 000100.



| Indirizzo | Istruzione | Operando |
|-----------|------------|----------|
| 000200 | LD | 000000 |
| 000201 | BPRG | 0 |
| . | A | . |
| . | | . |
| 000210 | CNTW | 0005 |
| | | #7000 |
| | | 000100 |
| . | B | . |
| . | | . |
| 000220 | BEND | --- |

L'esecuzione del programma procede da 2 a 3 a 4 e poi di nuovo a 2 nei 7.000 conteggi precedenti all'esecuzione di "B", come illustrato nel diagramma qui di seguito.



3-32-9 HIGH-SPEED TIMER WAIT: TMHW(815) e TMHWX(817)

Scopo

Ritarda l'esecuzione delle rimanenti istruzioni del programma a blocchi del periodo di tempo specificato. L'esecuzione riprende dall'istruzione dopo TMHW(815)/TMHWX(817) allo scadere del temporizzatore.

Simbolo programmazione ladder

Metodo di aggiornamento PV: BCD

TMHW(815) N N: numero di temporizzatore
SV SV: valore impostato

Metodo di aggiornamento PV: binario

TMHWX(817) N N: numero di temporizzatore
SV SV: valore impostato

Variazioni

| | |
|------------|---|
| Variazioni | Eseguita sempre nel programma a blocchi |
|------------|---|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Nota TIMHW(815) e TMHWX(817) devono essere utilizzate nelle aree di programmazione a blocchi anche per le subroutine.

Operandi

N: numero del temporizzatore

BCD: da 0 a 4095 (decimale)

Binario: da 0 a 4095 (decimale)

S: valore impostato

BCD: da #0000 a #9999 (BCD)

Binario: da &0 a &65535 (decimale)

da #0000 a #FFFF (esadecimale)

Caratteristiche operando

| Area | N | SV |
|--|--|--|
| Area CIO | --- | Da CIO 0000 a CIO 6143 |
| Area di lavoro | --- | Da W000 a W511 |
| Area bit di ritenività | --- | Da H000 a H511 |
| Area bit ausiliaria | --- | Da A000 ad A447 Da A448 ad A959 |
| Area del temporizzatore | Da 0000 a 4095 | Da T0000 a T4095 |
| Area del contatore | --- | Da C0000 a C4095 |
| Area DM | --- | Da D00000 a D32767 |
| Area EM senza banco | --- | Da E00000 a E32767 |
| Area EM con banco | --- | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | --- | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | --- | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- | BCD: da #0000 a 9999 (BCD). Impossibile utilizzare "&" Binario: da &0 a &65535 (decimale) da #0000 a #FFFF (esadecimale) |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

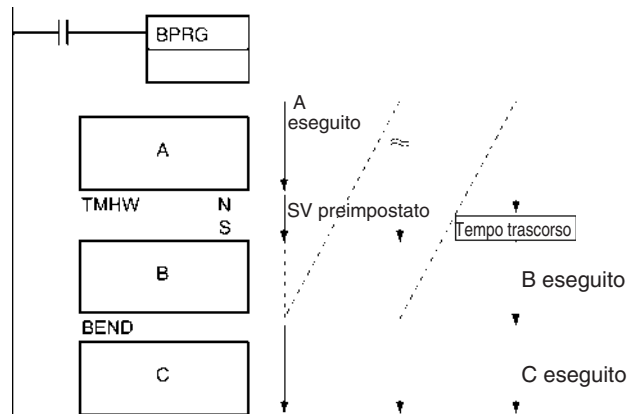
TMHW(815) e TMHWX(817) creano un temporizzatore con conteggio alla rovescia in ritardo all'eccitazione (temporizzatore a 10 ms impostato nel valore impostato SV) tra l'esecuzione dell'istruzione del programma a blocchi che le precede e le istruzioni successive. TMHW(815) può cronometrare da 0 a 99,99 s con una precisione del temporizzatore compresa tra 0 e 0,01 s. TMHWX(817) può cronometrare da 0 a 655,35 s con una precisione del temporizzatore compresa tra 0 e 0,01 s.

Nota La precisione del temporizzatore per CPU CS1D è di 10 ms + il tempo di ciclo.

La prima parte del programma a blocchi è eseguita quando il programma a blocchi viene specificato per la prima volta. Al raggiungimento di TMHW(815)/TMHWX(817), il flag di completamento è reimpostato su OFF, il temporizzatore viene preimpostato in base al valore impostato e l'esecuzione della parte restante del programma a blocchi deve attendere fino a che il valore impostato è trascorso.

Per aggiornare il temporizzatore in fase di decremento, è possibile eseguire solo TMHW(815) o TMHWX(817). Allo scadere del temporizzatore, è attivato il flag di completamento e la parte restante del programma a blocchi viene eseguita. Una volta eseguito il programma a blocchi completo, l'elaborazione viene ripetuta.

TMHW(815) e TMHWX(817) possono essere considerate come istruzioni WAIT con un temporizzatore per la condizione di esecuzione ed è quindi possibile utilizzarle per progressioni step a tempo.

**Flag**

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | <p>ON se TMHW(815)/TMHWX(817) non sono in un programma a blocchi.</p> <p>ON se viene utilizzata un'identificazione IR indiretta per N in modalità BCD e l'indirizzo non è per il valore attuale di un temporizzatore.</p> <p>ON se in modalità BCD con SV non BCD.</p> <p>OFF in tutti gli altri casi.</p> |

Avvertenze

Se il flag di completamento per il temporizzatore è sottoposto a impostazione forzata, la parte restante del programma a blocchi che segue TMHW(815)/TMHWX(817) viene eseguita.

Se il flag di completamento per il temporizzatore è sottoposto a ripristino forzato, nel programma a blocchi vengono eseguite soltanto le istruzioni TMHW(815)/TMHWX(817) fino a quando tale stato di ripristino non viene cancellato.

Anche quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 0000 a 2047

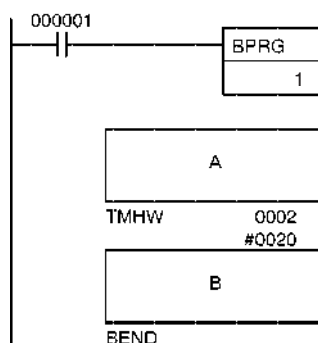
viene aggiornato. Quando il temporizzatore è in stato di attesa, il valore attuale dei temporizzatori programmati con numeri di temporizzatore da 2048 a 4095 viene mantenuto.

I numeri di temporizzatore vengono utilizzati anche da altre istruzioni del temporizzatore. Se si utilizza lo stesso numero di temporizzatore per più di una istruzione del temporizzatore, non è possibile prevedere il funzionamento. Utilizzare ciascun numero di temporizzatore una sola volta. L'unica occasione in cui è possibile un utilizzo attendibile dello stesso numero di temporizzatore si verifica quando è in funzione solo uno dei temporizzatori. Se si utilizza lo stesso numero di temporizzatore in più di una istruzione del temporizzatore, si verifica un errore nel controllo del programma.

Se viene utilizzata un'identificazione IR indiretta per N in modalità BCD e l'indirizzo non è per il valore attuale di un temporizzatore o se SV non è BCD, si verificherà un errore e verrà attivato il flag di errore.

Esempi

Nell'esempio che segue, "B" viene eseguito 20 secondi dopo "A" ogni volta che CIO 000000 è ON.



| Indirizzo | Istruzione | Operando |
|-----------|------------|----------|
| 000221 | LD | 000001 |
| 000222 | BPRG | 1 |
| . | A | . |
| . | . | . |
| 000250 | TMHW | 0002 |
| | | #0020 |
| . | B | . |
| . | . | . |
| 000281 | BEND | --- |

3-32-10 Loop Control: LOOP(809)/LEND(810)/LEND(810) NOT

Scopo

Crea un ciclo che viene eseguito più volte fino a quando una condizione di esecuzione passa a ON o a OFF oppure fino a quando una condizione di esecuzione passa a ON.

Simbolo programmazione ladder

- LOOP(809)
- LEND(810)
- LEND(810) B **B**: bit operando
- LEND(810) NOT B

Variazioni

| | |
|-------------------|--|
| Variazioni | Eseguita sempre nel programma a blocchi |
|-------------------|--|

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Nota LOOP(809), LEND(810) e LEND(810) NOT devono essere utilizzate nelle aree di programmazione a blocchi anche per subroutine e task ad interrupt.

Caratteristiche operando

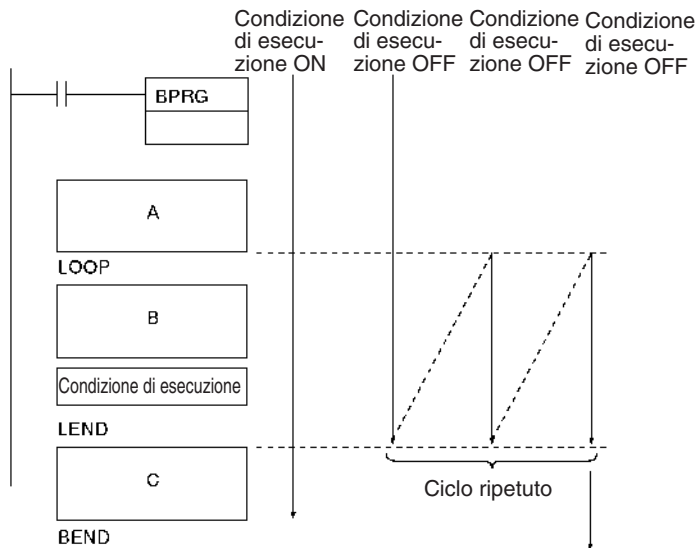
| Area | B |
|--|--|
| Area CIO | Da CIO 000000 a CIO 614315 |
| Area di lavoro | Da W00000 a W51115 |
| Area bit di ritentività | Da H00000 a H51115 |
| Area bit ausiliaria | Da A00000 ad A44715 Da A44800 ad A95915 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Flag dei task | Da TK0000 a TK0031 |
| Flag di condizione | ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER |
| Impulsi di clock | 0,02 s, 0,1 s, 0,2 s, 1 s, 1 min |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 |

Descrizione

LOOP(809) specifica l'inizio del programma ciclico. Le istruzioni LEND(810) e LEND(810) NOT specificano la fine del ciclo. Quando viene raggiunta l'istruzione LEND(810) o LEND(810) NOT, l'esecuzione del programma torna ciclicamente all'istruzione LOOP(809) immediatamente precedente finché il bit operando per LEND(810) o LEND(810) NOT non passa, rispettivamente, a ON o a OFF oppure finché lo stato della condizione di esecuzione per LEND(810) è ON.

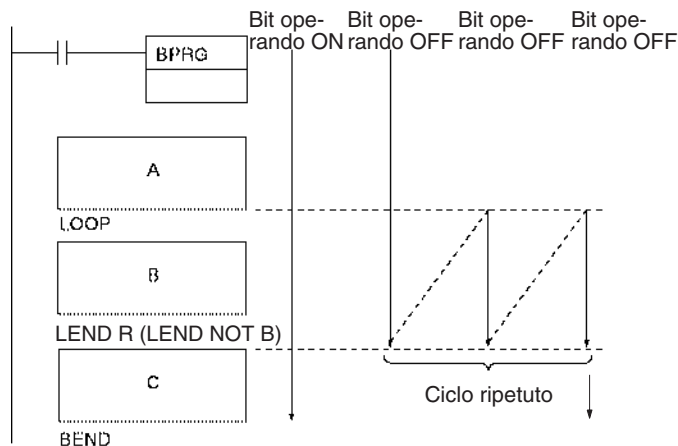
Utilizzo di una condizione di esecuzione per LEND(810)

LEND(810) può essere programmata con o senza un bit operando. Se non è specificato un bit operando, prima di LEND(810) è necessario creare un'esecuzione che inizia con LD. Se la condizione di esecuzione è OFF, l'esecuzione del ciclo viene ripetuta cominciando dalla prima istruzione successiva a LOOP(809). Se la condizione di esecuzione è impostata su ON, il ciclo viene interrotto e l'esecuzione procede all'istruzione successiva a LEND(810).



Utilizzo di un bit operando per LEND(810) o LEND(810) NOT

Sia LEND(810) che LEND(810) NOT possono essere programmate con un bit operando. Se il bit operando è OFF per LEND(810) (o ON per LEND(810) NOT), l'esecuzione del ciclo viene ripetuta a partire dall'istruzione successiva a LOOP(809). Se il bit operando è ON per LEND(810) (o OFF per LEND(810) NOT), il ciclo viene interrotto e l'esecuzione procede all'istruzione successiva a LEND(810) o LEND(810) NOT.



Nota Lo stato del bit operando risulta invertito per LEND(810) NOT.

- Nota**
1. L'esecuzione in un ciclo non aggiorna i dati di I/O. Se è necessario aggiornare i dati di I/O durante il ciclo, utilizzare IORF(184).
 2. Se i cicli vengono ripetuti per un periodo troppo lungo, è consentito superare il tempo di ciclo massimo. Scrivere il programma in modo tale da non superare il tempo di ciclo massimo.

Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se un'istruzione di controllo del ciclo non è in un programma a blocchi. OFF in tutti gli altri casi. |

Avvertenze

Non è possibile nidificare cicli all'interno di cicli.

Scorretto:

LOOP(809)
 LOOP(809)
 LEND(810)
 LEND(810)

Non invertire l'ordine di LOOP e LEND.

Scorretto:

LEND(810)
 :
 :
 LOOP(809)

In un ciclo è possibile utilizzare la diramazione blocco condizionale, ma l'intera operazione di diramazione deve essere all'interno del ciclo.

Corretto:

LOOP(809)
 IF(802)
 IF(802)
 IEND(804)
 IEND(804)
 LEND(810)

Scorretto:

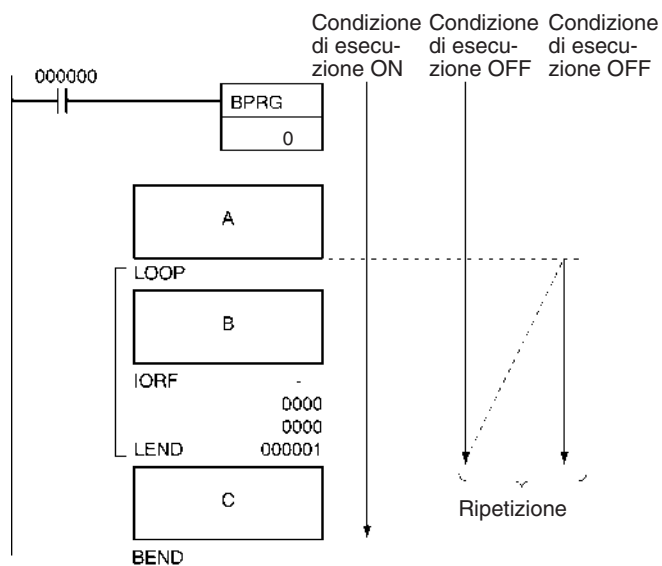
LOOP(809)
 IF(802)
 IF(802)
 IEND(804)
 LEND(810)
 IEND(804)

Se LOOP(809) non viene eseguita, avrà luogo l'elaborazione NOP.

Se un'istruzione di controllo del ciclo non è in un programma a blocchi, si verifica un errore e viene attivato il flag di errore.

Esempi

Nell'esempio seguente, quando CIO 000000 è ON, viene eseguito il programma a blocchi. Dopo l'esecuzione di "A", "B" e la successiva istruzione IORF(184) vengono eseguiti più volte fino a quando CIO 000001 è ON; quindi viene eseguito C e il programma a blocchi sarà terminato.



| Indirizzo | Istruzione | Operando |
|-----------|------------|----------|
| 000220 | LD | 000000 |
| 000201 | BPRG | 0 |
| . | A | . |
| . | | . |
| 000210 | LOOP | --- |
| . | B | . |
| . | | . |
| 000220 | IORF | . |
| | | . |
| | | 0000 |
| | | 0000 |
| 000221 | LEND | 000001 |
| . | C | . |
| . | | . |
| 000220 | BEND | --- |

3-33 Istruzioni di elaborazione delle stringhe di testo

In questa sezione vengono descritte le istruzioni utilizzate per gestire le stringhe di testo.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|-------------------------------------|---------------------------------|-----------------|--------|
| MOV STRING | MOV\$ | 664 | 1159 |
| CONCATENATE STRING | +\$ | 656 | 1161 |
| GET STRING LEFT | LEFT\$ | 652 | 1164 |
| GET STRING RIGHT | RGHT\$ | 653 | 1166 |
| GET STRING MIDDLE | MID\$ | 654 | 1168 |
| FIND IN STRING | FIND\$ | 660 | 1171 |
| STRING LENGTH | LEN\$ | 650 | 1173 |
| REPLACE IN STRING | RPLC\$ | 661 | 1175 |
| DELETE STRING | DEL\$ | 658 | 1178 |
| EXCHANGE STRING | XCHG\$ | 665 | 1180 |
| CLEAR STRING | CLR\$ | 666 | 1182 |
| INSERT INTO STRING | INS\$ | 657 | 1184 |
| Istruzioni di confronto di stringhe | =\$, <>\$, <\$, <=\$, >\$, >=\$ | Da 670 a 675 | 1187 |

3-33-1 Panoramica sull'elaborazione delle stringhe di testo

I dati dall'inizio fino a un codice NUL (00 esadecimale) vengono considerati come dati di una stringa di testo espressi in ASCII (esclusi i caratteri speciali di 1 byte). Vengono memorizzati dai byte più a sinistra a quelli più a destra e dai canali più a destra a quelli più a sinistra.

Nel caso in cui vi sia un numero dispari di caratteri, il valore 00 esadecimale (codice NUL) viene memorizzato nello spazio disponibile del byte più a sinistra dell'ultimo canale.

Esempio: stringa di testo ABCDE

| | | | | | |
|---|---|-----|--|--|--|
| A | → | B | | | |
| C | → | D | | | |
| E | → | NUL | | | |

=

| | |
|----|----|
| 41 | 42 |
| 43 | 44 |
| 45 | 00 |

Nel caso in cui vi sia un numero pari di caratteri, il valore 0000 esadecimale (due codici NUL) viene memorizzato nei byte più a sinistra e più a destra del canale successivo all'ultimo.

Esempio: stringa di testo ABCD

| | | | |
|-----|-----|--|--|
| A | B | | |
| C | D | | |
| NUL | NUL | | |

=

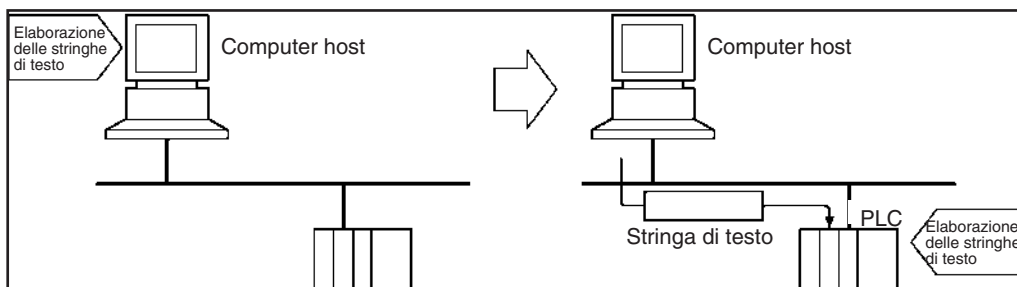
| | |
|----|----|
| 41 | 42 |
| 43 | 44 |
| 00 | 00 |

Come esemplificato nel diagramma seguente, è una stringa di testo può essere specificata semplicemente indicando il primo canale della stringa stessa. I dati della stringa di testo prima del carattere NUL successivo (00 esadecimale) vengono quindi considerati come un unico blocco di dati ASCII.

Esempio: MOV\$ D00000 D00100

| | | | | | | |
|--------|----|-----|---|---------|----|-----|
| D00000 | 41 | 42 | → | D000100 | 41 | 42 |
| D00001 | 43 | 44 | → | D000101 | 43 | 44 |
| D00002 | 45 | NUL | → | D000102 | 45 | NUL |

Per eseguire al PLC i vari tipi di elaborazione delle stringhe di testo (dati prodotto e così via), che in precedenza venivano effettuati al computer host, è possibile utilizzare le istruzioni di elaborazione delle stringhe di testo.



Ad esempio, è possibile trasferire dal computer host al PLC i dati relativi al piano di produzione, come, ad esempio, i nomi dei prodotti. Al PLC è inoltre possibile effettuare una serie di operazioni quali l'inserimento e la modifica di stringhe di testo, con conseguente riduzione del carico di elaborazione dei dati nel computer host.

Caratteri ASCII

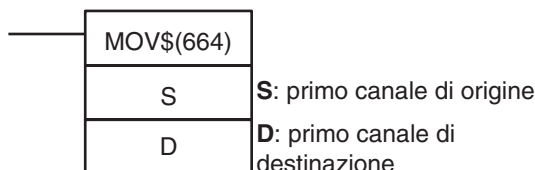
Nella tabella che segue sono indicati i caratteri ASCII gestibili dalle istruzioni di elaborazione delle stringhe di testo.

| | | Quattro bit più a sinistra | | | | | | | | | | | | | | | |
|--------------------------|---|----------------------------|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Quattro bit più a destra | 0 | | | Sp | 0 | @ | P | ' | p | | | | | 一 | タ | ミ | |
| | 1 | | | ! | 1 | A | Q | a | q | | | | | 。 | ア | チ | ム |
| | 2 | | | " | 2 | B | R | b | r | | | | | 「 | イ | ツ | メ |
| | 3 | | | # | 3 | C | S | c | s | | | | | 」 | ウ | テ | モ |
| | 4 | | | \$ | 4 | D | T | d | t | | | | | 、 | エ | ト | ヤ |
| | 5 | | | % | 5 | E | U | e | u | | | | | ・ | オ | ナ | ユ |
| | 6 | | | & | 6 | F | V | f | v | | | | | ヲ | カ | ニ | ヨ |
| | 7 | | | ' | 7 | G | W | g | w | | | | | ァ | キ | ヌ | ラ |
| | 8 | | | (| 8 | H | X | h | x | | | | | ィ | ク | ネ | リ |
| | 9 | | |) | 9 | I | Y | i | y | | | | | ゥ | ケ | ノ | ル |
| | A | | | * | : | J | Z | j | z | | | | | エ | コ | ハ | レ |
| | B | | | + | ; | K | [| k | { | | | | | オ | サ | ヒ | ロ |
| | C | | | , | < | L | ¥ | | | | | | | ャ | シ | フ | ワ |
| | D | | | - | = | M |] | m | } | | | | | ュ | ス | ヘ | ン |
| | E | | | . | > | N | ` | n | ~ | | | | | ョ | セ | ホ | ° |
| | F | | | / | ? | O | _ | o | | | | | | ッ | ソ | マ | |

3-33-2 MOV STRING: MOV\$(664)

Scopo Trasferisce una stringa di testo.

Simbolo programmazione ladder



Variazioni

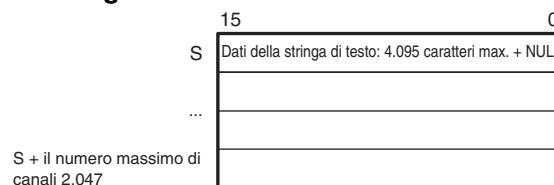
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MOV\$(664) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOV\$(664) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

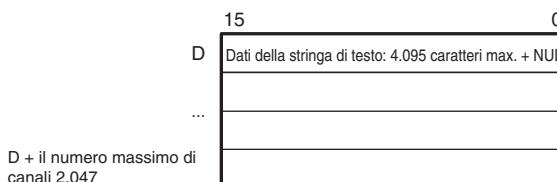
| | | | |
|-----------------------------|------------------------|------------|-------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

S: primo canale di origine



D: primo canale di destinazione



- Nota**
1. I dati compresi tra S ed S + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. Il dati compresi tra S ed S + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 possono essere sovrapposti.

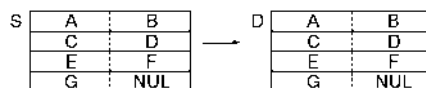
Caratteristiche operando

| Area | S | D |
|--------------------------------------|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |

| Area | S | D |
|--|--|---|
| Registri dati | --- | |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(-)IR0 a ,-(-)IR15 | |

Descrizione

MOV\$(664) prende i dati della stringa di testo specificata in S e, così come sono, li trasferisce in D come dati della stringa di testo, incluso il NUL finale. Il numero massimo di caratteri che può essere specificato da S è 4.095 (0FFF esadecimale).



Flag

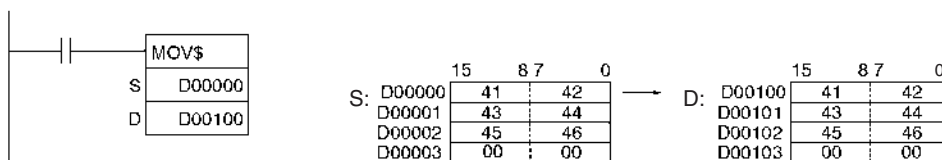
| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se i caratteri specificati da S sono più di 4.095. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore 0000 (esadecimale) viene trasferito in D. OFF in tutti gli altri casi. |

Avvertenze

Se i caratteri specificati da S sono più di 4.095, verrà generato un errore e verrà attivato il flag di errore.
Se il valore 0000 (esadecimale) è trasferito in D, verrà attivato il flag di uguaglianza.

Esempio

Nell'esempio, MOV\$(664) viene utilizzata per trasferire la stringa di testo ABCDEF.

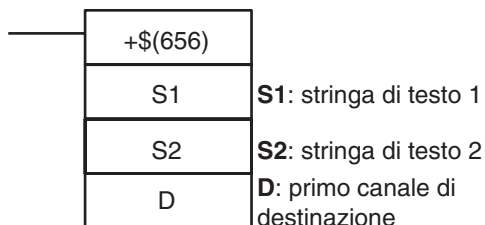


3-33-3 CONCATENATE STRING: +\$(656)

Scopo

Concatena due stringhe di testo.

Simbolo programmazione ladder



Variazioni

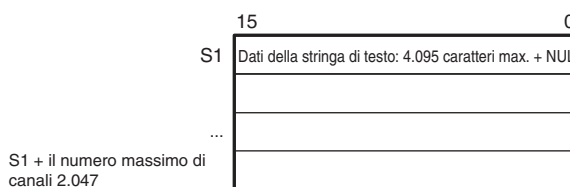
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | +\$ (656) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @+\$ (656) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

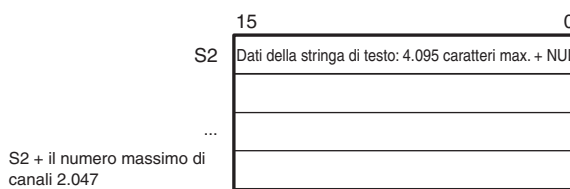
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

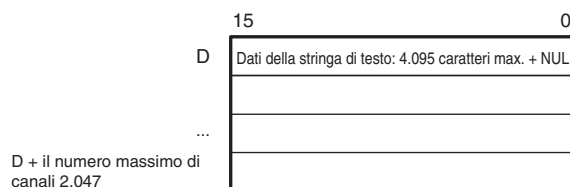
S1: stringa di testo 1



S2: stringa di testo 2



D: primo canale di destinazione



- Nota**
1. I dati compresi tra S1 ed S1 + il numero massimo di canali 2.047, tra S2 ed S2 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. I dati compresi tra S2 ed S2 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 non possono essere sovrapposti.

Caratteristiche operando

| Area | S1 | S2 | D |
|-------------------------|---------------------------------------|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a 32767 (n = da 0 a C) | | |

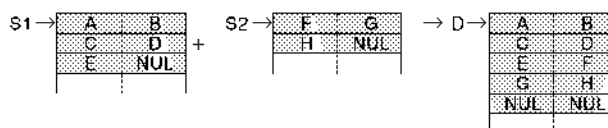
| Area | S1 | S2 | D |
|--|--|----|---|
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0V a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

+\$ (664) collega i dati della stringa di testo specificata da S1 alla stringa di testo indicata da S2 e invia il risultato in D come dati della stringa di testo (compreso il NUL finale).

Il numero massimo di caratteri che può essere specificato da S1 ed S2 è 4.095 (OFFF esadecimale). In assenza di un NUL tra 4.096 caratteri, verrà generato un errore e verrà attivato il flag di errore. Inoltre, il risultato della concatenazione non può essere superiore a 4.095 caratteri (OFFF esadecimale). Se il risultato della concatenazione è superiore, solo i primi 4.095 caratteri verranno inviati in D, con NUL aggiunto in qualità di carattere n. 4.096.

Se c'è un NUL sia per S1 che per S2, i due caratteri NUL (0000 esadecimale) vengono inviati in D.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i caratteri specificati da S1 ed S2 sono più di 4.095. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l' <i>esecuzione in background</i> è OFF quando viene specificata l' <i>elaborazione in background</i> . OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore 0000 (esadecimale) viene trasferito in D. OFF in tutti gli altri casi. |

Avvertenze

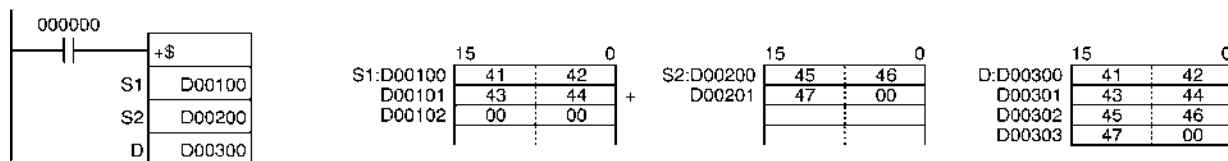
Se i caratteri specificati da S1 ed S2 sono più di 4.095, verrà generato un errore e verrà attivato il flag di errore.

Se il valore 0000 (esadecimale) è trasferito in D, verrà attivato il flag di uguaglianza.

Non sovrapporre il canale iniziale specificato da D con l'area dati del carattere per S2, altrimenti non è possibile eseguire l'istruzione normalmente.

Esempio

Qui +\$(656) viene utilizzata per collegare le stringhe di testo ABCD ed EFG e inviare il risultato in D.

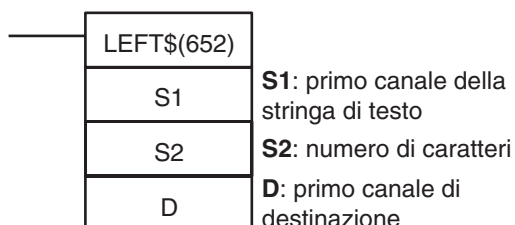


3-33-4 GET STRING LEFT: LEFT\$(652)

Scopo

Legge il numero di caratteri specificato a partire dalla sinistra (inizio) della stringa di testo.

Simbolo programmazione ladder



Variazioni

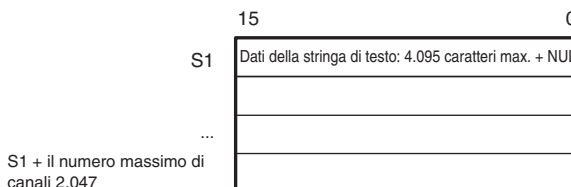
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | LEFT\$(652) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @LEFT\$(652) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

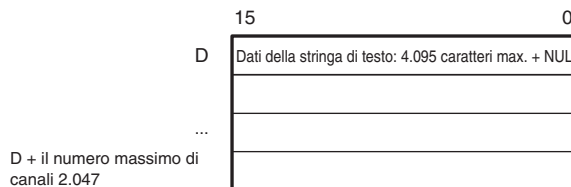
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S1: stringa di testo



S2: numero di caratteri (da 0000 a 0FFF esadecimale o da &0 a &4095)



- Nota**
1. I dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 possono essere sovrapposti.

Caratteristiche operando

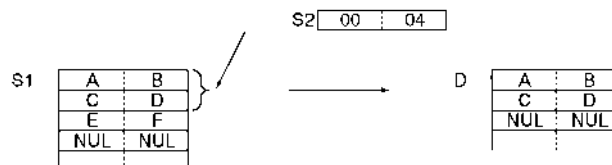
| Area | S1 | S2 | D |
|--|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a #0FFF (binario) o da &0 a &4095 | --- |
| Registri dati | --- | Da DR0 a DR15 | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

LEFT\$(652) legge il numero di caratteri specificati da S2, dalla sinistra (inizio) del primo canale della stringa di testo indicata da S1 fino al codice NUL (00 esadecimale) e invia i risultati in D, con NUL aggiunto alla fine.

Se il numero di caratteri letti supera quello dei caratteri specificati da S1, in D viene inviata l'intera stringa S1.

Se viene indicato il valore 0 (0000 esadecimale) come numero di caratteri da leggere, i due caratteri NUL (0000 esadecimale) verranno inviati in D.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se i caratteri specificati da S1 sono più di 4.095. ON se i caratteri specificati da S2 sono più di 4.095 (0FFF esadecimale). ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore 0000 (esadecimale) viene inviato in D. OFF in tutti gli altri casi. |

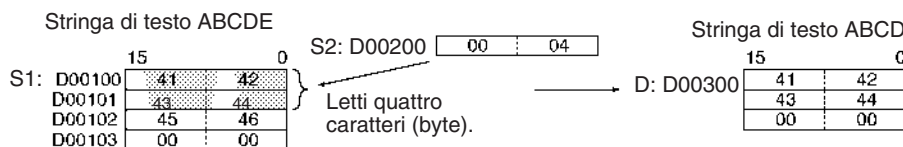
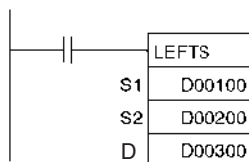
Avvertenze

Il numero massimo di caratteri da leggere che può essere specificato da S2 è 4.095 (0FFF esadecimale). In caso di superamento, viene generato un errore e verrà attivato il flag di errore.

Se il valore 0000 (esadecimale) è inviato in D, verrà attivato il flag di uguaglianza.

Esempio

L'istruzione LEFT\$(652) viene qui utilizzata per leggere quattro caratteri.

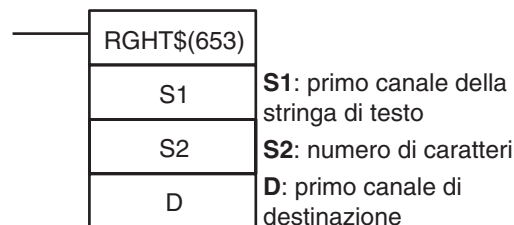


3-33-5 GET STRING RIGHT: RGHT\$(653)

Scopo

Legge il numero di caratteri specificato a partire dalla fine della stringa di testo.

Simbolo programmazione ladder



Variazioni

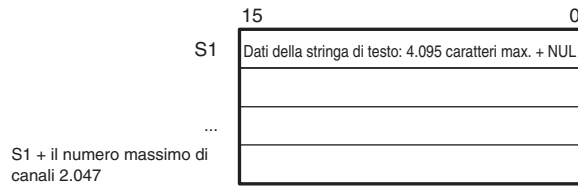
| Variazioni | Eseguita a ogni ciclo con condizione ON | RGHT\$(653) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RGHT\$(653) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

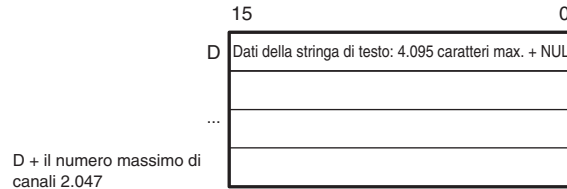
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S1: stringa di testo



S2: numero di caratteri (da 0000 a 0FFF esadecimale o da &0 a &4095)



- Nota**
1. I dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 possono essere sovrapposti.

Caratteristiche operando

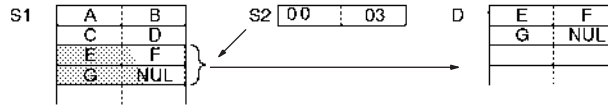
| Area | S1 | S2 | D |
|--|--|--|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a #0FFF (binario) o da &0 a &4095 | --- |
| Registri dati | --- | Da DR0 a DR15 | --- |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

RGHT\$(653) legge il numero di caratteri specificati da S2, dalla sinistra (inizio) del primo canale della stringa di testo indicata da S1 fino al codice NUL (00 esadecimale) e invia i risultati in D, con NUL aggiunto alla fine.

Se il numero di caratteri da leggere supera quello dei caratteri specificati da S1, in D viene inviata l'intera stringa S1.

Se viene indicato il valore 0 (0000 esadecimale) come numero di caratteri da leggere, i due caratteri NUL (0000 esadecimale) verranno inviati in D.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se i caratteri specificati da S1 sono più di 4.095. ON se i caratteri specificati da S2 sono più di 4.095 (0FFF esadecimale). ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore 0000 (esadecimale) viene inviato in D. OFF in tutti gli altri casi. |

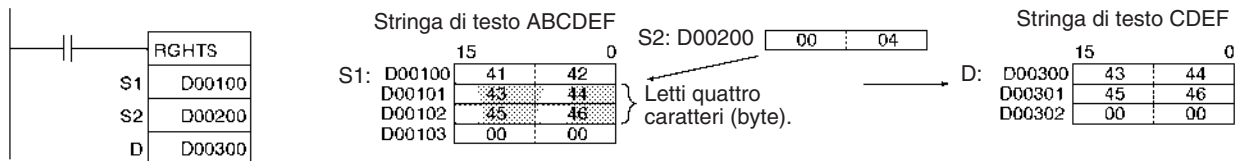
Avvertenze

Il numero massimo di caratteri da leggere che può essere specificato da S2 è 4.095 (0FFF esadecimale). Nel caso in cui è superato, viene generato un errore e verrà attivato il flag di errore.

Se il valore 0000 (esadecimale) è inviato in D, verrà attivato il flag di uguaglianza.

Esempio

L'istruzione RGHT\$(653) viene qui utilizzata per leggere quattro caratteri.

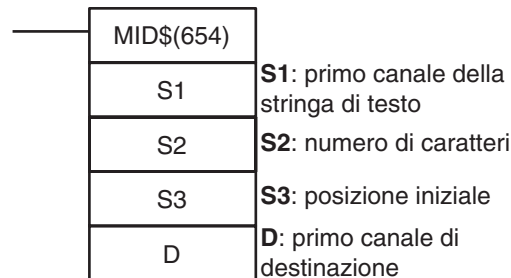


3-33-6 GET STRING MIDDLE: MID\$(654)

Scopo

Legge il numero di caratteri specificato a partire da qualsiasi posizione all'interno della stringa di testo.

Simbolo programmazione ladder



Variazioni

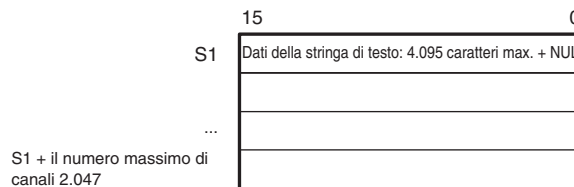
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | MID\$(654) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MID\$(654) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

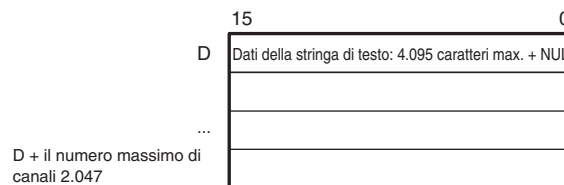
Operandi

S1: stringa di testo



S2: numero di caratteri (da 0000 a 0FFF esadecimale o da &0 a &4095)

S3: posizione iniziale (da 0001 a 0FFF esadecimale o da &1 a &4095)



- Nota**
1. I dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 possono essere sovrapposti.

Caratteristiche operando

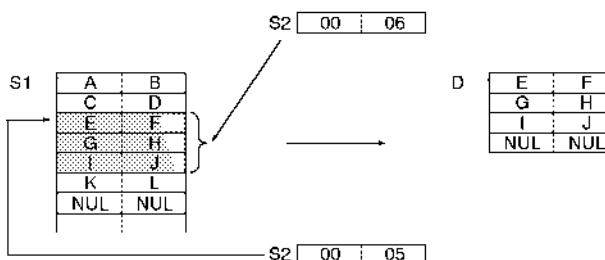
| Area | S1 | S2 | S3 | D |
|--------------------------------------|--|----|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | | |
| Area di lavoro | Da W000 a W511 | | | |
| Area bit di ritenività | Da H000 a H511 | | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | | |
| Area del contatore | Da C0000 a C4095 | | | |
| Area DM | Da D00000 a 32767 | | | |
| Area EM senza banco | Da E00000 a E32767 | | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | |

| Area | S1 | S2 | S3 | D |
|--|--|--|--|-----|
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | |
| Costanti | --- | Da #0000 a #0FFF (binario) o da &0 a &4095 | Da #0001 a #0FFF (binario) o da &1 a &4095 | --- |
| Registri dati | --- | Da DR0 a DR15 | | |
| Registri indice | --- | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | |

Descrizione

All'interno della stringa di testo individuata dal primo canale designato da S1 fino al codice NUL (00 esadecimale), MID\$(654) legge il numero di caratteri specificati da S2, dal primo canale indicato da S3 e invia il risultato in D come dati della stringa di testo con NUL aggiunto alla fine.

Se il numero dei caratteri da leggere supera la fine della stringa di testo specificata da S1, la stringa viene inviata fino alla fine.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i caratteri specificati da S1 sono più di 4.095. ON se i caratteri specificati da S2 sono più di 4.095 (0FFF esadecimale). ON se i dati in S3 sono nell'intervallo compreso tra 1 e 4.095 (da 0001 a 0FFF esadecimale). ON se S3 è maggiore di S1. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l' <i>esecuzione in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore 0000 (esadecimale) viene inviato in D. OFF in tutti gli altri casi. |

Avvertenze

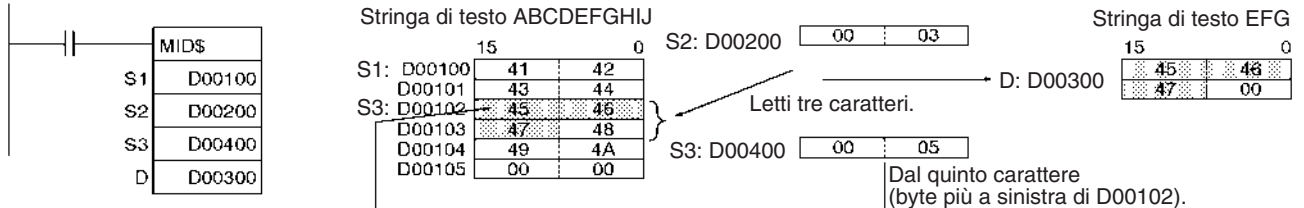
L'intervallo della posizione iniziale specificata da S3 è dal primo carattere al carattere n. 4.095 (da 0001 a 0FFF esadecimale). Se l'impostazione non rientra in tale intervallo, si verificherà un errore e verrà attivato il flag di errore. Il numero massimo di caratteri da leggere che può essere specificato da S2 è 4.095 (0FFF esadecimale). In caso di superamento, viene generato un errore e verrà attivato il flag di errore.

Se viene indicato il valore 0 (0000 esadecimale) come numero di caratteri da leggere, i due caratteri NUL (0000 esadecimale) verranno inviati in D.

Se il valore 0000 (esadecimale) è inviato in D, verrà attivato il flag di uguaglianza.

Esempio

L'istruzione MID\$(654) viene qui utilizzata per leggere tre caratteri.



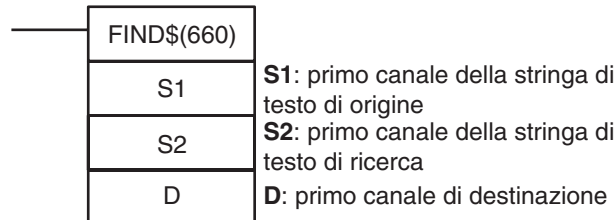
3-33-7 FIND IN STRING: FIND\$(660)

Scopo

Cerca la stringa di testo specificata all'interno di un'altra stringa.

Simbolo programmazione

ladder



Variazioni

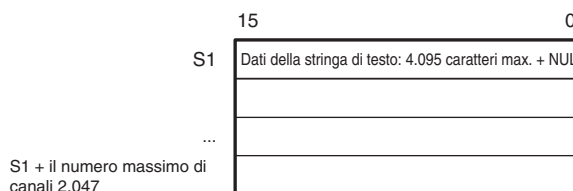
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | FIND\$(660) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @FIND\$(660) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

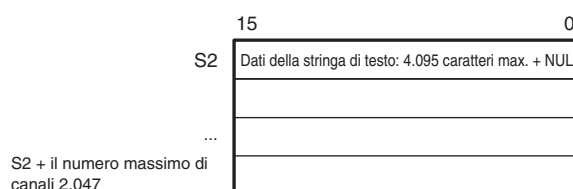
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S1: stringa di testo di origine



S2: stringa di testo di ricerca



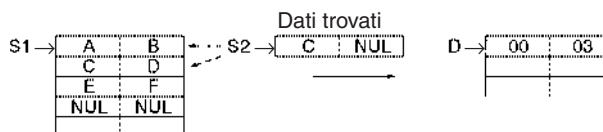
Nota I dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra S2 ed S2 + il numero massimo di canali 2.047 devono essere nella stessa area.

Caratteristiche operando

| Area | S1 | S2 | D |
|--|--|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | --- | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

FIND\$(660) trova la stringa di testo specificata da S2 nella stringa di testo indicata da S1 e invia il risultato, vale a dire un determinato numero di caratteri dall'inizio di S1, in D come dati binari. In assenza di una stringa di testo corrispondente, il valore 0000 esadecimale viene inviato in D.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se i caratteri specificati da S1 o S2 sono più di 4.095. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore 0000 (esadecimale) viene inviato in D. OFF in tutti gli altri casi. |

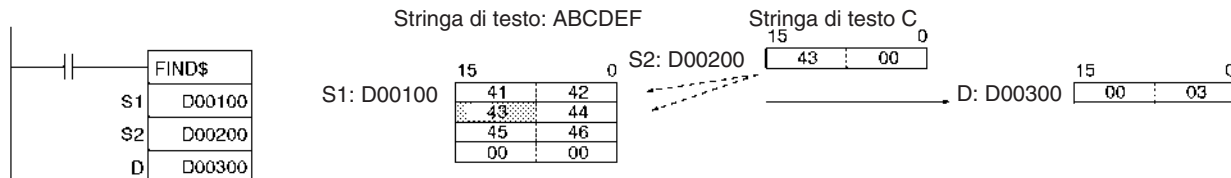
Avvertenze

Il numero massimo di caratteri da leggere che può essere specificato da S1 o S2 è 4.095 (0FFF esadecimale). In caso di superamento, viene generato un errore e verrà attivato il flag di errore.

Se il valore 0000 (esadecimale) è inviato in D, verrà attivato il flag di uguaglianza.

Esempio

L'istruzione FIND\$(660) viene qui utilizzata per cercare un carattere in una stringa di testo.

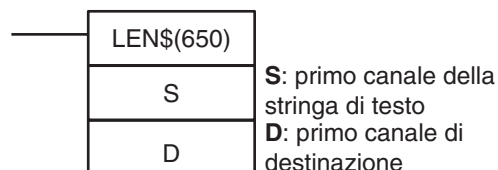


3-33-8 STRING LENGTH: LEN\$(650)

Scopo

Calcola la lunghezza di una stringa di testo.

Simbolo programmazione ladder



Variazioni

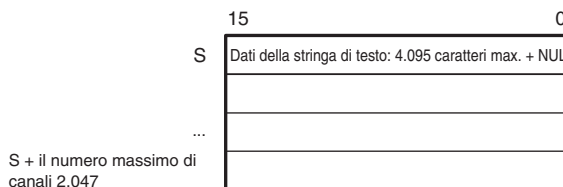
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | LEN\$(650) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @LEN\$(650) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Arete di programma applicabili

| Arete di programma a blocchi | Arete di programma step | Subroutine | Task ad interrupt |
|------------------------------|-------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: stringa di testo



Nota I dati compresi tra S ed S + il numero massimo di canali 2.047 devono essere nella stessa area.

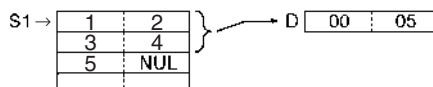
Caratteristiche operando

| Area | S | D |
|------------------------|------------------------------------|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritenività | Da H000 a H511 | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | Da A448 ad A959 |

| Area | S | D |
|--|--|---------------|
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | Da DR0 a DR15 |
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

LEFT\$(650) calcola il numero di caratteri dal primo canale della stringa di testo specificati da S fino al codice NUL incluso (00 esadecimale) e invia il risultato in D come dati binari. Se all'inizio della stringa di testo c'è un NUL, il risultato calcolato è pari a 0000 esadecimale.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se il risultato calcolato supera 4.095 caratteri. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l' <i>esecuzione in background</i> è OFF quando viene specificata l' <i>elaborazione in background</i> . OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato calcolato è 0. OFF in tutti gli altri casi. |

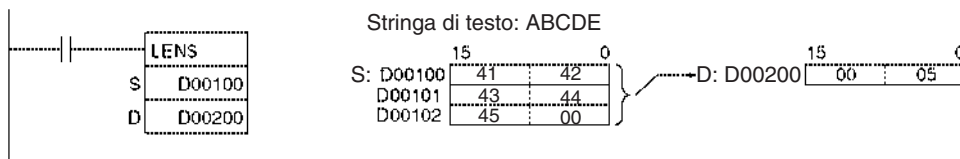
Avvertenze

Il numero massimo dei caratteri è 4.095 (0FFF esadecimale). Se il numero di caratteri è superiore, cioè in assenza di un NUL entro il carattere n. 4.096, verrà generato un errore e verrà attivato il flag di errore.

Se il valore 0000 (esadecimale) è inviato in D, verrà attivato il flag di uguaglianza.

Esempio

L'istruzione LENS\$(650) viene qui utilizzata per calcolare il numero dei caratteri e inviare il risultato.



3-33-9 REPLACE IN STRING: RPLC\$(661)

Scopo

Sostituisce una stringa di testo con quella specificata a partire da una determinata posizione.

Simbolo programmazione ladder

| | |
|-------------|--|
| RPLC\$(661) | S1: primo canale della stringa di testo |
| S1 | S2: primo canale della stringa di testo di sostituzione |
| S2 | S3: numero di caratteri |
| S3 | S4: posizione iniziale |
| S4 | D: primo canale di destinazione |
| D | |

Variazioni

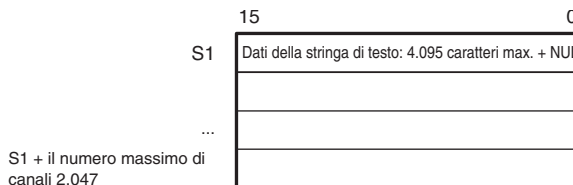
| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | RPLC\$(661) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @RPLC\$(661) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

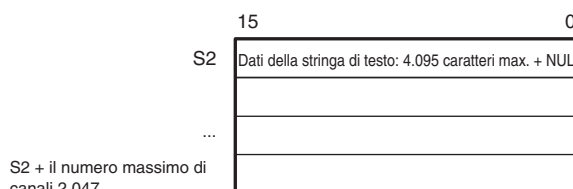
| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Operandi

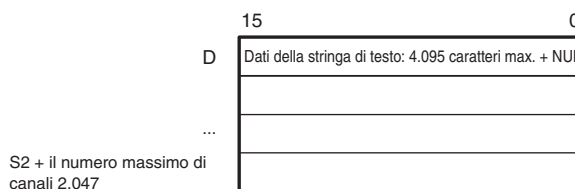
S1: stringa di testo



S2: stringa di testo di sostituzione



S3: numero di caratteri (da 0000 a 0FFF esadecimale o da &0 a &4095)
S4: posizione iniziale (da 0001 a 0FFF esadecimale o da &0 a &4095)



- Nota**
1. I dati compresi tra S1 ed S1 + il numero massimo di canali 2.047, tra S2 ed S2 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. I dati compresi tra D e D + il numero massimo di canali 2.047 e/o tra S1 ed S1 + il numero massimo di canali 2.047 nonché tra S2 ed S2 + il numero massimo di canali 2.047 possono essere sovrapposti.

Caratteristiche operando

| Area | S1 | S2 | S3 | S4 | D |
|--|--|----|--|--|-----|
| Area CIO | Da CIO 0000 a CIO 6143 | | | | |
| Area di lavoro | Da W000 a W511 | | | | |
| Area bit di ritentività | Da H000 a H511 | | | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | | | |
| Area del contatore | Da C0000 a C4095 | | | | |
| Area DM | Da D00000 a D32767 | | | | |
| Area EM senza banco | Da E00000 a E32767 | | | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | | |
| Costanti | --- | | Da #0000 a #0FFF (binario) o da &0 a &4095 | Da #0001 a #0FFF (binario) o da &1 a &4095 | --- |
| Registri dati | --- | | Da DR0 a DR15 | | --- |
| Registri indice | --- | | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | | |

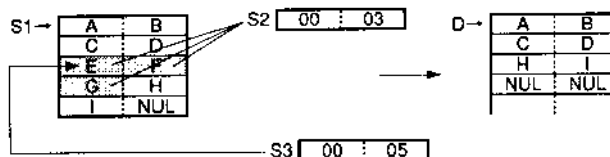
Descrizione

RPLC\$(661) sostituisce parte della stringa di testo designata da S1, dalla posizione iniziale designata da S4, con la stringa di testo specificata da S2 e invia il risultato in D come dati della stringa di testo (compreso il NUL finale). Il numero di caratteri da sostituire è specificato da S3.

Il numero massimo dei caratteri nel risultato è 4.095 (0FFF esadecimale). Se il numero è maggiore, verranno inviati solo 4.095 caratteri (con NUL aggiunto come carattere n. 4.096).

È possibile sostituire da 0 a 4.095 caratteri (da 0000 a 0FFF esadecimale). Se il numero è 0, la stringa di testo designata da S1 viene inviata in D senza subire alcuna modifica. Se la stringa di testo in S2 è NUL, l'operazione sarà equivalente all'eliminazione dell'intervallo di testo designato in S1.

Se la stringa di testo in S1 è completamente sostituita da NUL, due caratteri NUL (0000 esadecimale) vengono inviati in D.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | <p>ON se i caratteri specificati da S1 o S2 sono più di 4.095.</p> <p>ON se i caratteri designati da S3 sono più di 4.095 (0FFF esadecimale).</p> <p>ON se i dati in S4 sono nell'intervallo compreso tra 1 e 4.095 (da 0001 a 0FFF esadecimale).</p> <p>ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'<i>esecuzione in background</i> è OFF quando viene specificata l'elaborazione in background.</p> <p>OFF in tutti gli altri casi.</p> |
| Flag di uguaglianza | = | <p>ON se il valore 0000 (esadecimale) viene inviato in D.</p> <p>OFF in tutti gli altri casi.</p> |

Avvertenze

Il numero massimo dei caratteri per S1 o S2 è 4.095 (0FFF esadecimale). Se il numero di caratteri è maggiore, cioè in assenza di un NUL entro il carattere n. 4.096, verrà generato un errore e verrà attivato il flag di errore.

L'intervallo della posizione iniziale designata da S4 è compreso dal primo carattere al carattere n. 4.095 (da 0001 a 0FFF esadecimale). Se l'impostazione non rientra in tale intervallo, si verificherà un errore e verrà attivato il flag di errore.

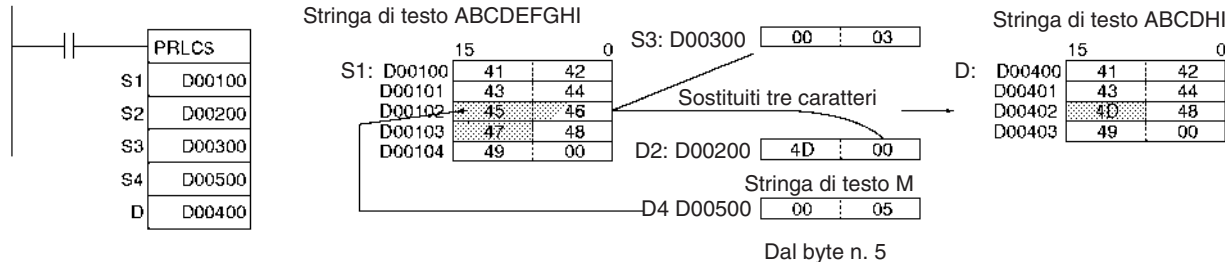
Se la posizione iniziale designata da S4 supera la stringa di testo designata da S1, verrà generato un errore e verrà attivato il flag di errore.

Se il valore 0000 (esadecimale) è inviato in D, verrà attivato il flag di uguaglianza.

Impostare il primo canale di destinazione D in modo tale che non venga sovrapposto con le aree definite con il primo canale della stringa di testo di sostituzione S2. Se tali aree vengono sovrapposte, il funzionamento di RPLC\$(654) non sarà corretto.

Esempio

L'istruzione RPLC\$(654) viene qui utilizzata per leggere tre caratteri.

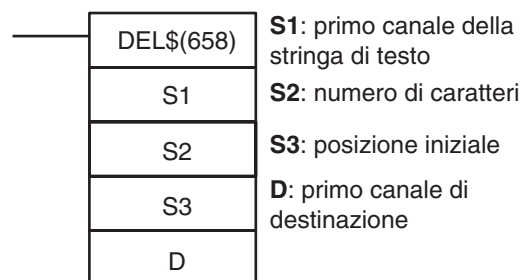


3-33-10 DELETE STRING: DEL\$(658)

Scopo

Elimina la stringa di testo specificata dall'interno di un'altra stringa.

Simbolo programmazione ladder



Variazioni

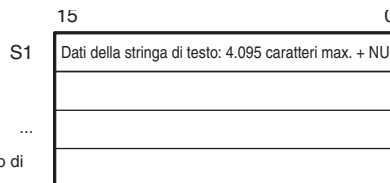
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | DEL\$(658) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @DEL\$(658) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

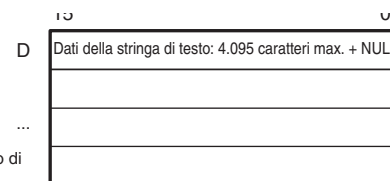
Operandi

S1: stringa di testo



S2: numero di caratteri (da 0000 a 0FFF esadecimale o da &0 a &4095)

S3: posizione iniziale (da 0001 a 0FFF esadecimale o da &1 a &4095)



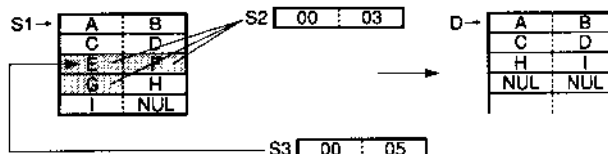
- Nota**
1. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047, tra S2 ed S2 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 possono essere sovrapposti.

Caratteristiche operando

| Area | S1 | S2 | S3 | D |
|--|--|---|---|-----|
| Area CIO | Da CIO 0000 a CIO 6143 | | | |
| Area di lavoro | Da W000 a W511 | | | |
| Area bit di ritenività | Da H000 a H511 | | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | | | |
| Area del contatore | Da C0000 a C4095 | | | |
| Area DM | Da D00000 a D32767 | | | |
| Area EM senza banco | Da E00000 a E32767 | | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | |
| Costanti | --- | Da #0000 a #0FFF (binario) o da &0 a &4095 | Da #0001 a #0FFF (binario) o da &1 a &4095 | --- |
| Registri dati | --- | Da DR0 a DR15 | | --- |
| Registri indice | --- | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | |

Descrizione

All'interno della stringa di testo designata da S1, DEL\$(658) elimina il numero di caratteri designati da S2, dal primo canale specificato da S3, e invia il risultato in D come dati della stringa di testo con NUL aggiunto alla fine.



Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i caratteri specificati da S1 sono più di 4.095. ON se i caratteri specificati da S2 sono più di 4.095 (0FFF esadecimale). ON se i dati in S3 sono nell'intervallo compreso tra 1 e 4.095 (da 0001 a 0FFF esadecimale). ON se S3 è maggiore di S1. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l' <i>esecuzione in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON quando il valore 0000 esadecimale viene inviato in D. OFF in tutti gli altri casi. |

Avvertenze

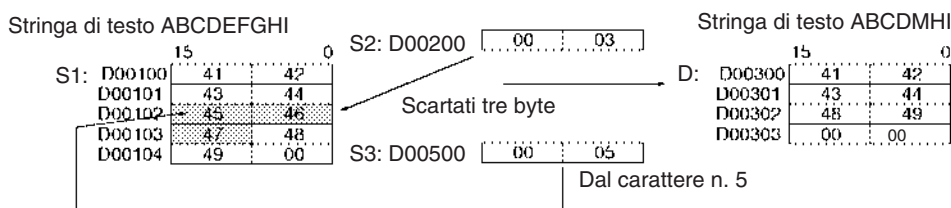
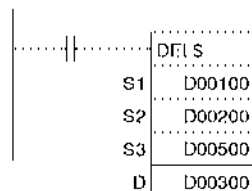
Il numero massimo dei caratteri per S1 è 4.095 (0FFF esadecimale). Se il numero di caratteri è maggiore, cioè in assenza di un NUL entro il carattere n. 4.096, verrà generato un errore e verrà attivato il flag di errore.

L'intervallo della posizione iniziale designata da S3 è dal primo carattere al carattere n. 4.095 (da 0001 a 0FFF esadecimale). Se l'impostazione non rientra in tale intervallo, si verificherà un errore e verrà attivato il flag di errore.

Se il numero dei canali specificati per S1 supera la lunghezza della stringa di testo, viene attivato il flag di errore.

Se il numero dei caratteri da eliminare supera la fine della stringa di testo di S1, verranno eliminati tutti i caratteri fino alla fine. Se tutti i caratteri dall'inizio alla fine di S1 sono specificati ai fini dell'eliminazione, il valore 000 esadecimale viene inviato in D.

Esempio

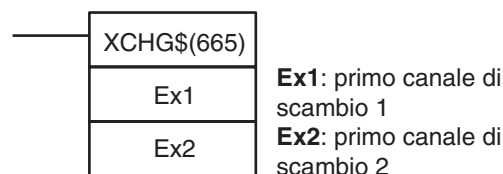


3-33-11 EXCHANGE STRING: XCHG\$(665)

Scopo

Sostituisce la stringa di testo specificata con un'altra stringa specificata.

Simbolo programmazione ladder



Variazioni

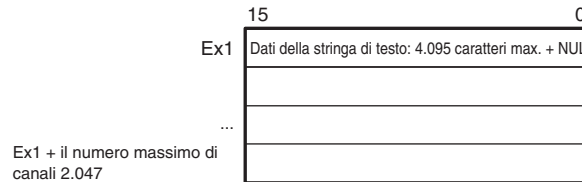
| Variazioni | Eseguita a ogni ciclo con condizione ON | XCHG\$(665) |
|-------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XCHG\$(665) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

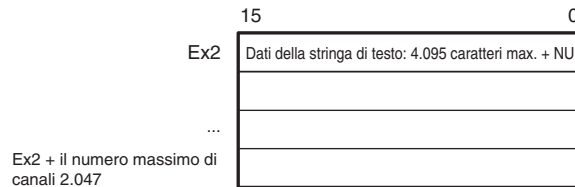
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Ex1: primo canale di scambio 1



Ex2: primo canale di scambio 2



- Nota**
1. I dati compresi tra Ex1 ed Ex1 + il numero massimo di canali 2.047 e tra Ex2 ed Ex2 + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. Il dati compresi tra Ex2 ed Ex2 + il numero massimo di canali 2.047 e tra Ex ed Ex + il numero massimo di canali 2.047 non possono essere sovrapposti.

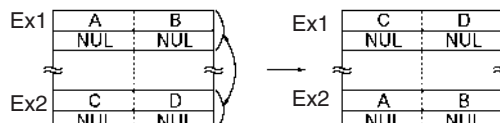
Caratteristiche operando

| Area | Ex1 | Ex2 |
|--------------------------------------|--|-----|
| Area CIO | Da CIO 0000 a CIO 6143 | |
| Area di lavoro | Da W000 a W511 | |
| Area bit di ritentività | Da H000 a H511 | |
| Area bit ausiliaria | Da A448 ad A959 | |
| Area del temporizzatore | Da T0000 a T4095 | |
| Area del contatore | Da C0000 a C4095 | |
| Area DM | Da D00000 a D32767 | |
| Area EM senza banco | Da E00000 a E32767 | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | |
| Costanti | --- | |
| Registri dati | --- | |

| Area | Ex1 | Ex2 |
|--|--|-----|
| Registri indice | --- | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | |

Descrizione

XCHG\$(665) scambia la stringa di testo designata da Ex1 con la stringa di testo specificata da Ex2. Se Ex1 o Ex2 sono NUL, due caratteri NUL (0000 esadecimale) verranno inviati all'altro dei due.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se i caratteri designati da Ex1 o Ex2 sono più di 4.095. ON se i dati di Ex1 ed Ex2 si sovrappongono. ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione in background è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |

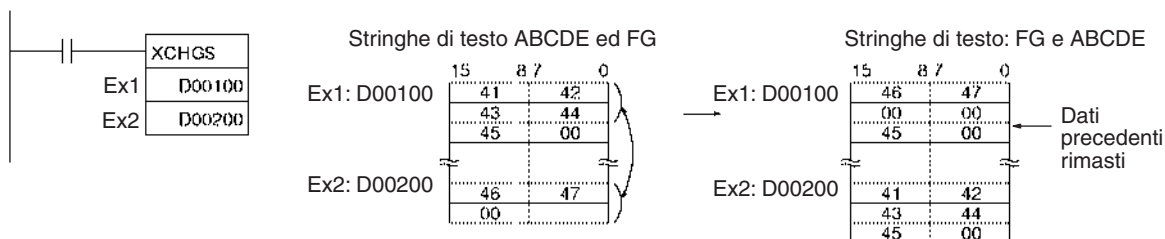
Avvertenze

Il numero massimo di caratteri che può essere designato da Ex1 o Ex2 è 4.095 (0FFF esadecimale). In caso di superamento, viene generato un errore e verrà attivato il flag di errore.

Se i dati della stringa di testo designati da Ex1 ed Ex2 si sovrappongono, verrà generato un errore e verrà attivato il flag di errore.

Esempio

L'istruzione XCHG\$(665) viene qui utilizzata per scambiare due stringhe di testo.

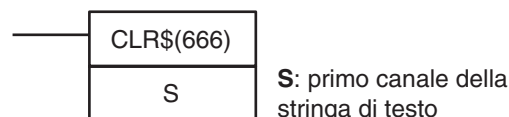


3-33-12 CLEAR STRING: CLR\$(666)

Scopo

Sostituisce il contenuto di una stringa di testo con NUL (00 esadecimale).

Simbolo programmazione ladder



Variazioni

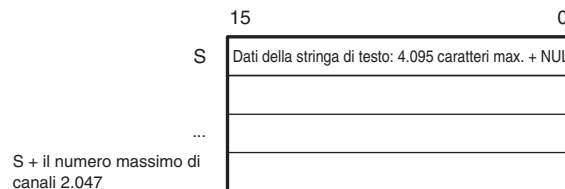
| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | CLR\$(666) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @CLR\$(666) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: primo canale della stringa di testo



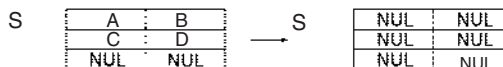
Nota I dati compresi tra S ed S + il numero massimo di canali 2.047 devono essere nella stessa area.

Caratteristiche operando

| Area | S |
|--|--|
| Area CIO | Da CIO 0000 a CIO 6143 |
| Area di lavoro | Da W000 a W511 |
| Area bit di ritenività | Da H000 a H511 |
| Area bit ausiliaria | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 |
| Area del contatore | Da C0000 a C4095 |
| Area DM | Da D00000 a D32767 |
| Area EM senza banco | Da E00000 a E32767 |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) |
| Costanti | --- |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 |

Descrizione

CLR\$(666) sostituisce con NUL (00 esadecimale) tutta la di testo stringa dal primo canale designato da S fino al codice NUL (00 esadecimale). Il numero massimo di caratteri che è possibile cancellare è 4.096. Se prima del carattere n. 4.096 non c'è alcun NUL, verranno cancellati soltanto 4.096 caratteri.

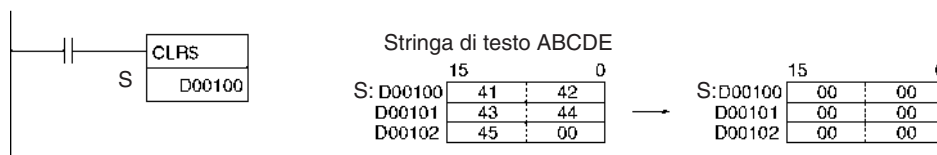


Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |

Esempio

L'istruzione CLR\$(666) viene qui utilizzata per cancellare la stringa di testo ABCDE.

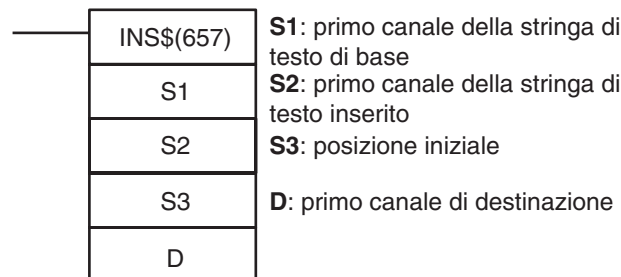


3-33-13 INSERT INTO STRING: INS\$(657)

Scopo

Elimina la stringa di testo specificata dall'interno di un'altra stringa.

Simbolo programmazione ladder



Variazioni

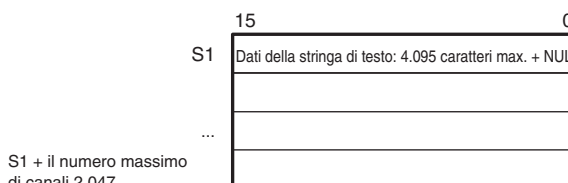
| Variazioni | Eseguita a ogni ciclo con condizione ON | INS\$(657) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @INS\$(657) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

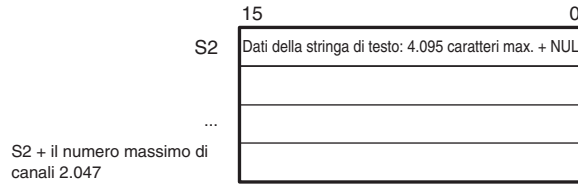
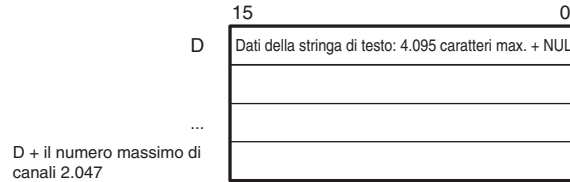
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S1: stringa di testo di base



S2: stringa di testo inserito**S3: posizione iniziale (da 0000 a 0FFF esadecimale o da &0 a &4095)**

- Nota**
1. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047, tra S2 ed S2 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 devono essere nella stessa area.
 2. Il dati compresi tra S2 ed S2 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 non possono essere sovrapposti. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra D e D + il numero massimo di canali 2.047 possono essere sovrapposti. Il dati compresi tra S1 ed S1 + il numero massimo di canali 2.047 e tra S2 ed S2 + il numero massimo di canali 2.047 possono anche essere sovrapposti.

Caratteristiche operando

| Area | S1 | S2 | S3 | D |
|--------------------------------------|--|----|---|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | | |
| Area di lavoro | Da W000 a W511 | | | |
| Area bit di ritentività | Da H000 a H511 | | | |
| Area bit ausiliaria | Da A000 ad A447 Da A448 ad A959 | | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | | |
| Area del contatore | Da C0000 a C4095 | | | |
| Area DM | Da D00000 a D32767 | | | |
| Area EM senza banco | Da E00000 a E32767 | | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | | |
| Costanti | --- | | Da #0000 a #0FFF (binario) o da &0 a &4095 | --- |
| Registri dati | --- | | Da DR0 a DR15 | --- |

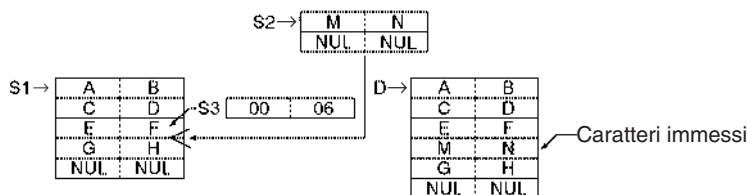
| Area | S1 | S2 | S3 | D |
|--|--|----|----|---|
| Registri indice | --- | | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | | |

Descrizione

All'interno della stringa di testo designata da S1, INS\$(657) inserisce la stringa di testo specificata da S2, dopo il canale iniziale designato da S3, e invia il risultato in D come dati della stringa di testo con NUL aggiunto alla fine.

Il numero massimo dei caratteri che è possibile inserire è 4.095 (0FFF esadecimale). Se il numero è superiore, verranno inviati in D solo 4.095 caratteri (con NUL aggiunto come carattere n. 4.096).

Se S1 o S2 è NUL, la stringa di testo specificata dall'altro verrà inviata in D senza subire alcuna modifica. Se S1 ed S2 sono entrambi NUL, due caratteri NUL (0000 esadecimale) vengono inviati in D.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se i caratteri specificati da S1 o S2 sono più di 4.095. ON se S3 è maggiore di 4.095 (0FFF esadecimale). ON se il flag di abilitazione porte di comunicazione specificato come <i>numero di porta Com</i> per l'esecuzione <i>in background</i> è OFF quando viene specificata l'elaborazione in background. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il valore 0000 (esadecimale) viene inviato in D. OFF in tutti gli altri casi. |

Avvertenze

Il numero massimo dei caratteri per S1 ed S2 è 4.095 (0FFF esadecimale). Se il numero di caratteri è maggiore, cioè in assenza di un NUL entro il carattere n. 4.096, verrà generato un errore e verrà attivato il flag di errore.

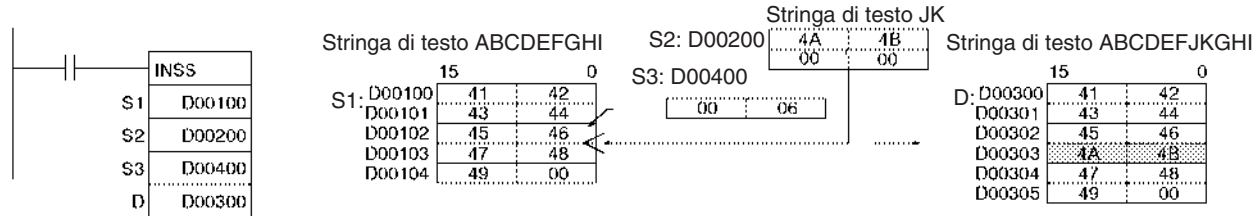
L'intervallo per la posizione iniziale designata da S3 è compreso tra 0 e 4.095. Se l'impostazione non rientra in tale intervallo, si verificherà un errore e verrà attivato il flag di errore.

Se il valore 0000 (esadecimale) è inviato in D, verrà attivato il flag di uguaglianza.

Non sovrapporre i canali di destinazione designati da D con i dati della stringa di testo specificati da S2, altrimenti non sarà possibile eseguire l'istruzione normalmente.

Esempio

L'istruzione INSS\$(657) viene qui utilizzata per inserire due caratteri.



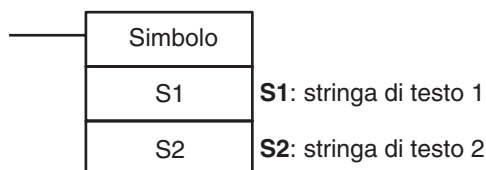
3-33-14 Istruzioni di confronto di stringhe (da 670 a 675)

Scopo

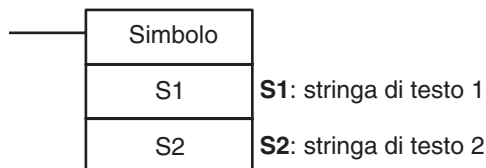
Le istruzioni di confronto di stringhe (= \$, <> \$, < \$, <= \$, > \$, >= \$) confrontano due stringhe di testo a partire dall'inizio, verificando i valori dei codici ASCII. Se il risultato del confronto è vero, viene creata una condizione di esecuzione ON per LOAD, AND od OR.

Simbolo programmazione ladder

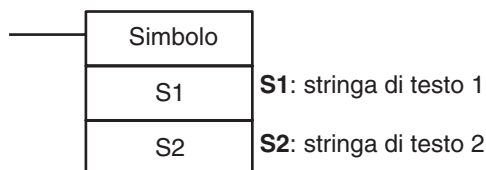
LD



AND (collegamento in serie)



OR (collegamento parallelo)



Variazioni

| | | |
|--------------------------------|---|-------------------------------------|
| Variazioni | Crea una condizione di esecuzione con stato ON a ogni ciclo se il confronto risulta vero | Istruzioni di confronto di stringhe |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| | | | |
|------------------------------------|-------------------------------|-------------------|--------------------------|
| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
| OK | OK | OK | OK |

Descrizione

Le istruzioni di confronto di stringhe consentono di confrontare le stringhe di testo designate da S1 ed S2. Se il risultato del confronto è vero, viene creata una condizione di esecuzione ON nel diagramma ladder. Il numero massimo dei caratteri per S1 o S2 è 4.095 (0FFF esadecimale).

Le istruzioni di confronto di stringhe vengono espresse utilizzando i 18 differenti codici mnemonici elencati qui di seguito (LD, AND e OR non compaiono nel diagramma ladder).

LD=\$, AND=\$, OR=\$
 LD<>\$, AND<>\$, OR<>\$
 LD<\$, AND<\$, OR<\$
 LD<=\$, AND<=\$, OR<=\$
 LD>\$, AND>\$, OR>\$
 LD>=\$, AND>=\$, OR>=\$

Nella tabella seguente sono indicati alcuni dettagli relativi a tali istruzioni.

| Codice mnemonico (con codice funzione) | Nome | Funzione |
|--|------------------------------------|---|
| LD=\$ (670) | LOAD STRING EQUALS | Vero quando la stringa di testo di S1 equivale alla stringa di testo di S2. |
| AND=\$ (670) | AND STRING EQUALS | |
| OR=\$ (670) | OR STRING EQUALS | |
| LD<>\$ (671) | LOAD STRING NOT EQUAL | Vero quando la stringa di testo di S1 non equivale alla stringa di testo di S2. |
| AND<>\$ (671) | AND STRING NOT EQUAL | |
| OR<>\$ (671) | OR STRING NOT EQUAL | |
| LD<\$ (672) | LOAD STRING LESS THAN | Vero quando la stringa di testo di S1 è minore della stringa di testo di S2. |
| AND<\$ (672) | AND STRING LESS THAN | |
| OR<\$ (672) | OR STRING LESS THAN | |
| LD<=\$ (673) | LOAD STRING LESS THAN OR EQUALS | Vero quando la stringa di testo di S1 è minore della stringa di testo S2 o uguale. |
| AND<=\$ (673) | AND STRING LESS THAN OR EQUALS | |
| OR<=\$ (673) | OR STRING LESS THAN OR EQUALS | |
| LD>\$ (674) | LOAD STRING GREATER THAN | Vero quando la stringa di testo di S1 è maggiore della stringa di testo di S2. |
| AND>\$ (674) | AND STRING GREATER THAN | |
| OR>\$ (674) | OR STRING GREATER THAN | |
| LD>=\$ (675) | LOAD STRING GREATER THAN OR EQUALS | Vero quando la stringa di testo di S1 è maggiore della stringa di testo di S2 o uguale. |
| AND>=\$ (675) | AND STRING GREATER THAN OR EQUALS | |
| OR>=\$ (675) | OR STRING GREATER THAN OR EQUALS | |

Metodi di confronto

Di seguito vengono descritti i metodi di confronto.

Il primo carattere (byte) di ogni stringa di testo viene messo a confronto con l'elemento corrispondente dell'altra stringa come codice ASCII. Se i due codici ASCII non equivalgono, tale rapporto maggiore/minore diventa il rapporto maggiore/minore per le due stringhe di testo. Se i due codici ASCII equivalgono, avrà luogo il confronto tra i caratteri successivi. Pertanto, se questi due codici ASCII non equivalgono, tale rapporto maggiore/minore diventa il rapporto maggiore/minore per le due stringhe di testo.

In questo modo, le due stringhe di testo vengono messe a confronto in sequenza, carattere per carattere. Se tutti i caratteri, NUL compreso, si equivalgono, le due stringhe di testo presenteranno un rapporto di uguaglianza.

Se le due stringhe di testo hanno lunghezze differenti, il NUL (00 esadecimale) verrà aggiunto alla stringa più breve per pareggiare tale differenza e il confronto avrà luogo su questa nuova base.

Esempi di confronto

AD (414400 esadecimale) e BC (424300 esadecimale):

AD < BC, poiché all'inizio delle stringhe di testo 41 (esadecimale) è minore di 42 (esadecimale).

ADC (41444300 esadecimale) e B (4200 esadecimale):

ADC < B, poiché all'inizio delle stringhe di testo 41 (esadecimale) è minore di 42 (esadecimale).

ABC (41424300 esadecimale) e ABD (41424400 esadecimale):

ABC < ABD, poiché all'inizio delle stringhe di testo i 41 e i 42 corrispondono, per cui il risultato viene determinato dal fatto che 43 è minore di 44.

ABC (41424300 esadecimale) e AB (414200 esadecimale):

ABC > AB, poiché all'inizio delle stringhe di testo i 41 e i 42 corrispondono, per cui il risultato viene determinato dal fatto che 43 è maggiore di 00.

AB (414200 esadecimale) e AB (414200 esadecimale):

AB = AB, poiché i 41, i 42 e gli 00 corrispondono tutti.

Procedere con la programmazione di un'istruzione dopo l'altra, considerando LD, AND e OR allo stesso modo. A differenza delle istruzioni AND, è possibile collegare le istruzioni LD e OR direttamente alla barra di distribuzione.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------------|-----------|--|
| Flag di errore | ER | ON se i caratteri specificati da S1 o S2 sono più di 4.095. OFF in tutti gli altri casi. |
| Flag di maggiore | > | ON se i risultati del confronto in S1 sono maggiori di quelli in S2. OFF in tutti gli altri casi. |
| Flag di maggiore o uguale | >= | ON se i risultati del confronto in S1 sono uguali a quelli di S2 o maggiori. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se i risultati del confronto in S1 equivalgono a quelli in S2. OFF in tutti gli altri casi. |
| Flag di non uguaglianza | <> | ON se i risultati del confronto in S1 non equivalgono a quelli in S2. OFF in tutti gli altri casi. |
| Flag di minore | < | ON se i risultati del confronto in S1 sono minori di quelli in S2. OFF in tutti gli altri casi. |
| Flag di minore o uguale | <= | ON se i risultati del confronto in S1 sono uguali a quelli di S2 o minori. OFF in tutti gli altri casi. |

Nota Le istruzioni di confronto di stringhe vengono utilizzate per modificare l'ordine delle stringhe di testo in ordine ASCII. Ad esempio, l'ordine ASCII, ascendente, è la sequenza alfabetica dalla A alla Z, per cui è possibile organizzare le stringhe di testo in ordine alfabetico.

Avvertenze

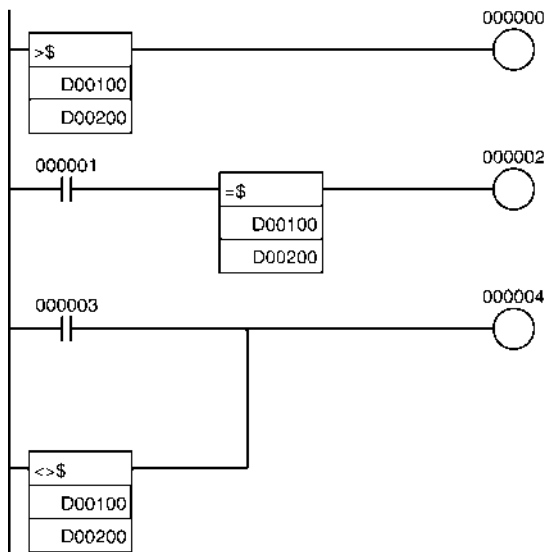
Dopo queste istruzioni, specificare un'istruzione sul lato destro. Le istruzioni di confronto di stringhe non possono comparire sul lato destro del diagramma ladder.

Non è possibile utilizzarle sull'ultimo rung di un blocco logico.

Il numero massimo dei caratteri che è possibile confrontare è 4.095 (0FFF esadecimale). Se tale numero di caratteri è maggiore, cioè in assenza di un NUL entro il carattere n. 4.096, si verificherà un errore e verrà attivato il flag di errore. In questo caso, una condizione di esecuzione OFF verrà inviata all'istruzione successiva.

Esempio

Le istruzioni di confronto di stringhe vengono qui utilizzate per confrontare dati.



| Indirizzo | Mnemonico | Operando |
|-----------|-----------|-------------------------|
| 000000 | LD > \$ | --- D00100 D00200 |
| 000001 | OUT | 000000 |
| 000002 | LD | 000001 |
| 000003 | AND=S | --- D00100 D00200 |
| 000004 | OUT | 000002 |
| 000005 | LD | 000003 |
| 000006 | OR <> \$ | --- D00100 D00200 |
| 000007 | OUT | 000004 |

Stringa di testo ABCD

| | | |
|--------|----|----|
| D00100 | 41 | 42 |
| D00101 | 44 | 43 |
| D00102 | 00 | 00 |

Stringa di testo ABC

| | | |
|--------|----|----|
| D00200 | 41 | 42 |
| D00201 | 43 | 00 |

Stringa di testo ABC

| | | |
|--------|----|----|
| D00100 | 41 | 42 |
| D00101 | 43 | 00 |

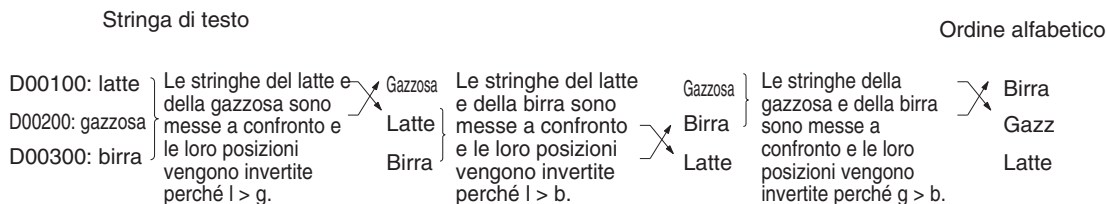
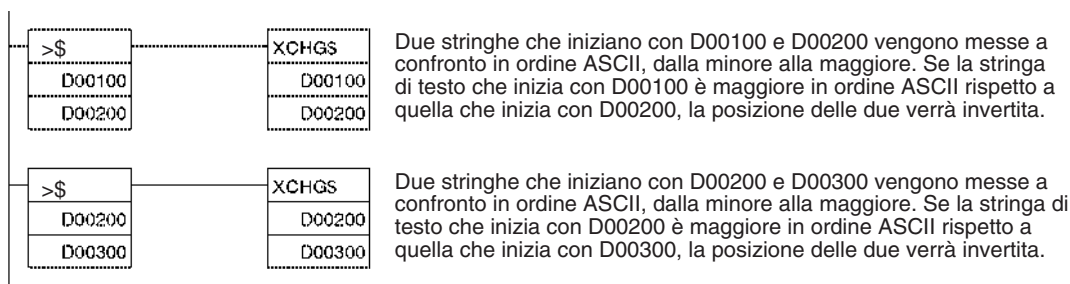
Stringa di testo ABC

| | | |
|--------|----|----|
| D00200 | 41 | 42 |
| D00201 | 43 | 00 |

| > \$ | = \$ | <>\$ |
|--------|--------|--------|
| D00100 | D00100 | D00100 |
| D00200 | D00200 | D00200 |
| ON | OFF | ON |
| OFF | ON | OFF |

Nell'esempio, tre stringhe di testo vengono riorganizzate in ordine alfabetico.
 Ordine iniziale:
 D00100: latte
 D00200: gazzosa
 D00300: birra

Una volta messe in ordine alfabetico, le stringhe saranno così: birra, gazzosa, latte.



In questo modo, tre stringhe di testo possono essere riorganizzate in ordine alfabetico.

3-34 Istruzioni di controllo dei task

In questa sezione vengono descritte le istruzioni utilizzate per controllare i task.

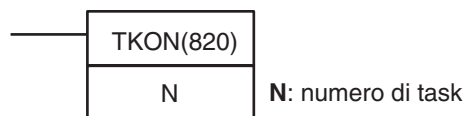
| Istruzione | Mnemonico | Codice funzione | Pagina |
|------------|-----------|-----------------|--------|
| TASK ON | TKON | 820 | 1192 |
| TASK OFF | TKOF | 821 | 1196 |

3-34-1 TASK ON: TKON(820)

Scopo

Rende il task specificato eseguibile. Inoltre fa sì che un task ad interrupt funzioni come un task ciclico supplementare. I task ciclici supplementari sono supportati solo dalle CPU CS1-H, CJ1-H e CJ1M.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TKON(820) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TKON(820) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Operandi

N: numero di task

L'intervallo consentito per N dipende dal tipo di task che viene specificato.

- Task ciclici:
N deve essere una costante compresa tra 0 e 31 decimale (i valori da 0 a 31 indicano i task ciclici da 0 a 31).
- Task ciclici supplementari (solo CPU CS1-H, CJ1-H, CJ1M e CS1D):
N deve essere una costante compresa tra 8000 e 8255 decimale (i valori da 8000 a 8255 indicano i task ciclici da 0 a 255).

Caratteristiche operando

| Area | N |
|--|--|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 00 a 31 o da 8000 a 8255 (decimale) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

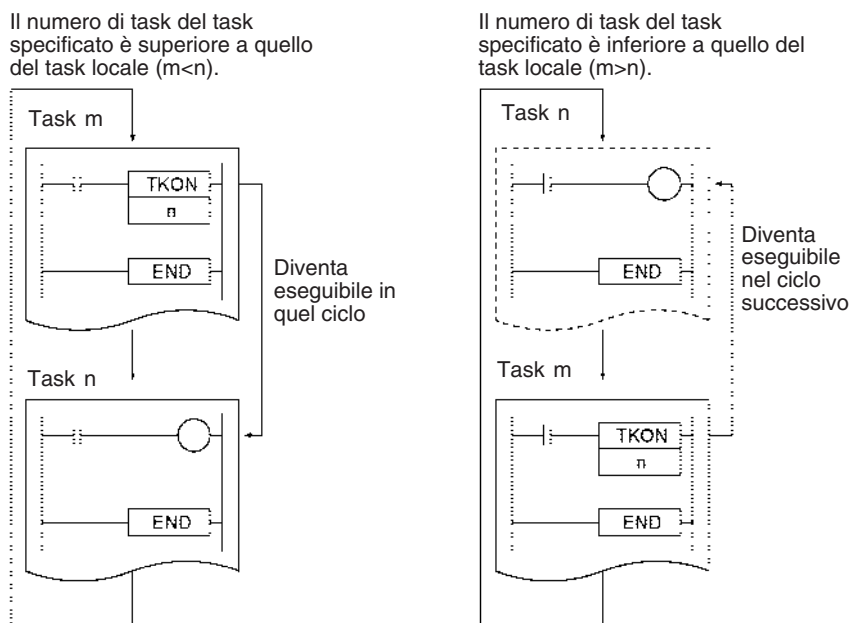
Descrizione

TKON(820) consente di attribuire al task ciclico specificato o al task ciclico supplementare uno stato eseguibile. Quando il valore di N è tra 0 e 31 (specificando un task ciclico), il flag dei task corrispondente (da TK00 a TK31) viene attivato.

L'istruzione può essere eseguita solo in un task ciclico normale o in un task ciclico supplementare. Se si tenta di eseguirla da un task ad interrupt, si verificherà un errore.

Il task ciclico o il task ciclico supplementare specificato in TKON(820) sarà eseguibile anche in cicli successivi purché non sia posto in stato di attesa da TKOF(821).

È possibile rendere eseguibile ogni task da qualsiasi task ciclico, ma se il numero di task è minore di quello del task locale, il task specificato non verrà eseguito fino al ciclo successivo. Il task verrà eseguito nello stesso ciclo se il suo numero di task è maggiore di quello del task locale.



Se il task specificato è già eseguibile o se è specificato il task locale, TKON(820) viene considerata come NOP(000).

È possibile mettere in stato di attesa un task eseguibile mediante TKOF(821), il CX-Programmer o un comando FINS.

I termini eseguibile ed esecuzione non sono intercambiabili. I task eseguibili vengono eseguiti nell'ordine dei loro numeri di task durante l'esecuzione del programma ciclico. Se un task eseguibile viene posto in stato di attesa prima che l'esecuzione del programma raggiunga il suo numero di task, non verrà eseguito.

- Nota**
- Per ogni task, la *scheda Proprietà generali* di CX-Programmer prevede un'impostazione (la casella *Avvio dell'operazione*) che specifica se il task ciclico è eseguibile all'avvio. Se si seleziona la casella *Avvio dell'operazione*, il task ciclico corrispondente viene messo in stato eseguibile automaticamente quando il PLC entra in funzione. Tutti gli altri task ciclici sono in stato non eseguibile.
Se l'operazione di cancellazione della memoria viene eseguita dalla console di programmazione, tuttavia, il task ciclico 0 è reso eseguibile in modo automatico.
 - Se un task è in stato non eseguibile, è possibile eseguire TKON(820) per rendere eseguibile lo stato di tale task. Allo stesso modo, lo stato eseguibile di un task ciclico può essere modificato in non eseguibile mediante l'istruzione TKOF(821).
 - I task ciclici o i task ciclici supplementari resi eseguibili vengono posti in stato eseguibile in quel ciclo in base all'ordine del numero di task. Di conseguenza, se un task viene posto in stato di attesa prima che l'elaborazione del ciclo lo abbia raggiunto, non sarà eseguito, poiché ogni task viene eseguito in base all'ordine del numero di task.

Flag

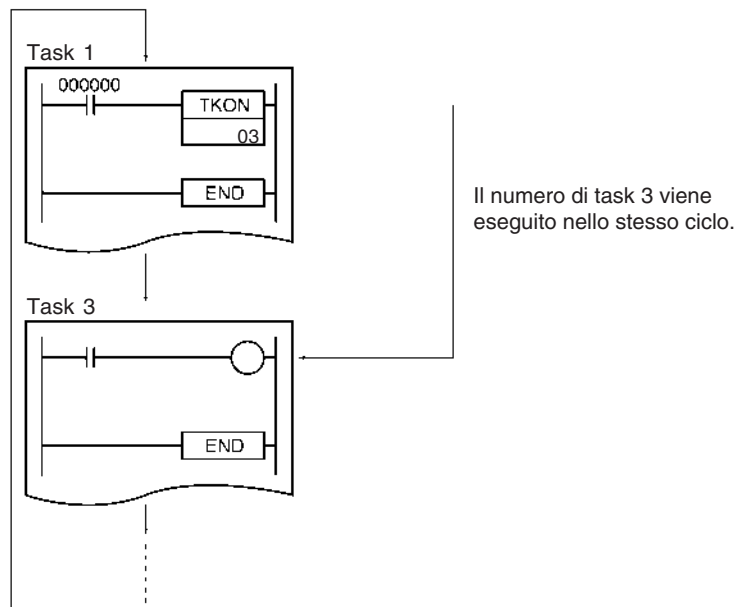
| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se N non è una costante compresa tra 00 e 31 o tra 8000 e 8255 (solo CPU CS1-H, CJ1-H e CJ1M). ON se il task specificato con N non esiste. ON se TKON(820) viene eseguita in un task ad interrupt. OFF in tutti gli altri casi. |

| Nome | Indirizzi | Funzionamento |
|---------------|----------------|---|
| Flag dei task | Da TK00 a TK31 | Questi flag vengono attivati quando il task ciclico corrispondente è eseguibile e disattivati quando il task ciclico corrispondente non è eseguibile o è in stato di attesa. L'intervallo da TK00 a TK31 corrisponde ai numeri di task ciclici da 00 a 31. |

Esempi

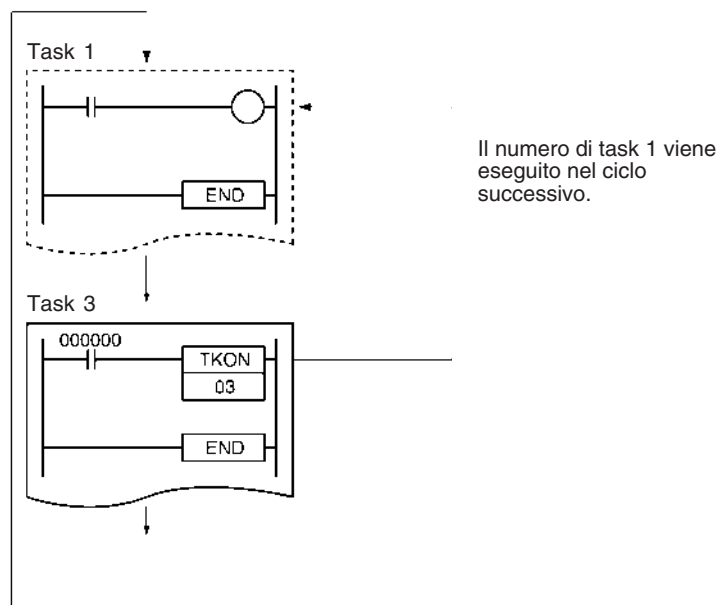
Specifica di un task successivo

Nell'esempio che segue, quando CIO 000000 è impostato su ON, il numero di task 3 è reso eseguibile nel numero di task 1. Il numero di task 3 verrà eseguito nello stesso ciclo quando l'esecuzione del programma raggiunge il numero di task 3.



Specifica di un task precedente

Nell'esempio che segue, quando CIO 000000 è impostato su ON, il numero di task 1 è reso eseguibile nel numero di task 3. Il numero di task 1 verrà eseguito nel ciclo successivo quando l'esecuzione del programma raggiunge il numero di task 1.

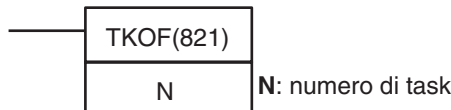


3-34-2 TASK OFF: TKOF(821)

Scopo

Mette il task ciclico specificato o il task ciclico supplementare in stato di attesa, ossia disabilita l'esecuzione del task. I task ciclici supplementari sono supportati solo dalle CPU CS1-H, CJ1-H e CJ1M.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | TKOF(821) |
|------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @TKOF(821) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| | Aggiornamento immediato | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | Non consentita |

Operandi

N: numero di task

L'intervallo consentito per N dipende dal tipo di task che viene specificato.

- Task ciclici:
N deve essere una costante compresa tra 0 e 31 decimale (i valori da 0 a 31 indicano i task ciclici da 0 a 31).
- Task ciclici supplementari (solo CPU CS1-H, CJ1-H, CJ1M e CS1D):
N deve essere una costante compresa tra 8000 e 8255 decimale (i valori da 8000 a 8255 indicano i task ciclici da 0 a 255).

Caratteristiche operando

| Area | N |
|--|--|
| Area CIO | --- |
| Area di lavoro | --- |
| Area bit di ritenività | --- |
| Area bit ausiliaria | --- |
| Area del temporizzatore | --- |
| Area del contatore | --- |
| Area DM | --- |
| Area EM senza banco | --- |
| Area EM con banco | --- |
| Indirizzi DM/EM indiretti in binario | --- |
| Indirizzi DM/EM indiretti in BCD | --- |
| Costanti | Da 00 a 31 o da 8000 a 8255 (decimale) |
| Registri dati | --- |
| Registri indice | --- |
| Indirizzamento indiretto con i registri indice | --- |

Descrizione

TKOF(821) assegna al task ciclico specificato o al task ciclico supplementare uno stato in attesa e disattiva il corrispondente flag dei task (da TK00 a TK31).

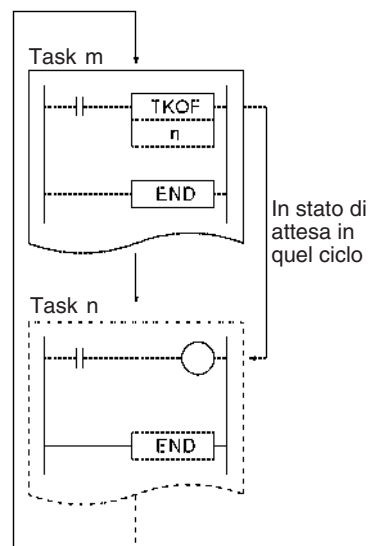
Anche il task specificato in TKOF(821) avrà lo stato di attesa nei cicli successivi fino a quando non riceve lo stato eseguibile da TKON(820), un dispositivo periferico con CX-Programmer o un comando FINS.

Da qualsiasi task ciclico è possibile assegnare uno stato di attesa a un task, ma se il numero di task è minore di quello del task locale, tale task non sarà posto in stato di attesa fino al ciclo successivo (l'esecuzione sarebbe già avvenuta). Il task verrà posto in stato di attesa nello stesso ciclo se il suo numero di task è maggiore di quello del task locale.

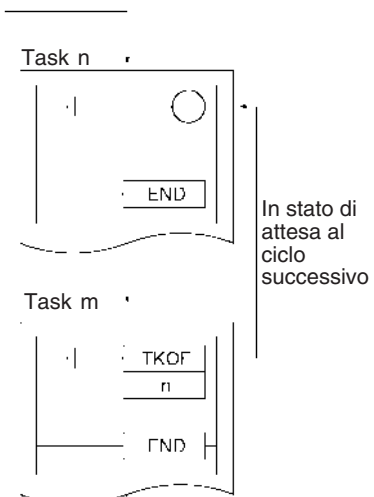
Se il task locale è specificato in TKOF(821), al task viene assegnato immediatamente lo stato di attesa e nel task stesso non verrà eseguita alcuna delle istruzioni successive.

- Nota**
1. Per ogni task, la *scheda Proprietà generali* di CX-Programmer prevede un'impostazione (la casella *Avvio dell'operazione*) che specifica se il task ciclico è eseguibile all'avvio. Se si seleziona la casella *Avvio dell'operazione*, il task ciclico corrispondente viene messo in stato eseguibile automaticamente quando il PLC entra in funzione. Tutti gli altri task ciclici sono in stato non eseguibile.
Se l'operazione di cancellazione della memoria viene eseguita dalla console di programmazione, tuttavia, il task ciclico 0 è reso eseguibile in modo automatico.
 2. Se un task è in stato non eseguibile, è possibile eseguire TKON(820) per rendere eseguibile lo stato di tale task. Allo stesso modo, lo stato eseguibile di un task ciclico può essere modificato in non eseguibile mediante l'istruzione TKOF(821).
 3. I task ciclici o i task ciclici supplementari con stato eseguibile possono essere posti in stato di attesa dall'istruzione TKOF(821).

Il numero di task del task specificato è superiore a quello del task locale ($m < n$).



Il numero di task del task specificato è inferiore a quello del task locale ($m > n$).



A un task normale impostato per l'esecuzione all'avvio verrà assegnato automaticamente lo stato eseguibile quando il PLC entra in funzione. Tutti gli altri task normali sono in stato non eseguibile.

È possibile mettere in stato di attesa un task eseguibile mediante TKOF(821), una periferica con CX-Programmer o un comando FINS.

I termini eseguibile ed esecuzione non sono intercambiabili. I task eseguibili vengono eseguiti nell'ordine dei loro numeri di task durante l'esecuzione del programma ciclico. Se un task eseguibile viene posto in stato di attesa prima

che l'esecuzione del programma raggiunga il suo numero di task, non verrà eseguito.

A differenza di TKON(820), è possibile specificare questa istruzione nei task ad interrupt e in quelli ciclici.

Flag

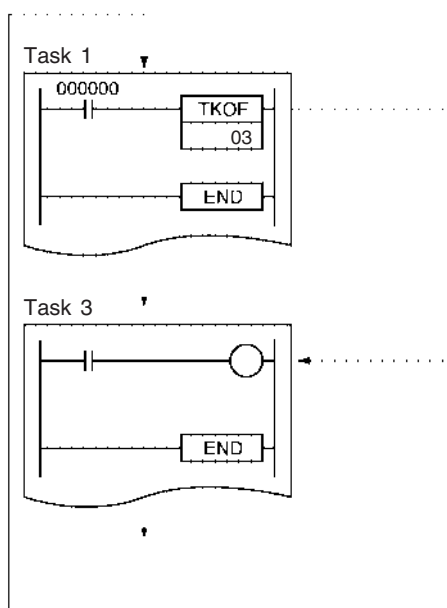
| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se N non è una costante compresa tra 00 e 31 o tra 8000 e 8255 (solo CPU CS1-H, CJ1-H e CJ1M). ON se il task specificato con N non esiste. ON se TKOF(821) viene eseguita in un task ad interrupt. OFF in tutti gli altri casi. |

| Nome | Indirizzi | Funzionamento |
|---------------|----------------|---|
| Flag dei task | Da TK00 a TK31 | Questi flag vengono attivati quando il task ciclico corrispondente è eseguibile e disattivati quando il task ciclico corrispondente non è eseguibile o è in stato di attesa. L'intervallo da TK00 a TK31 corrisponde ai numeri di task ciclici da 00 a 31. |

Esempi

Specifica di un task successivo

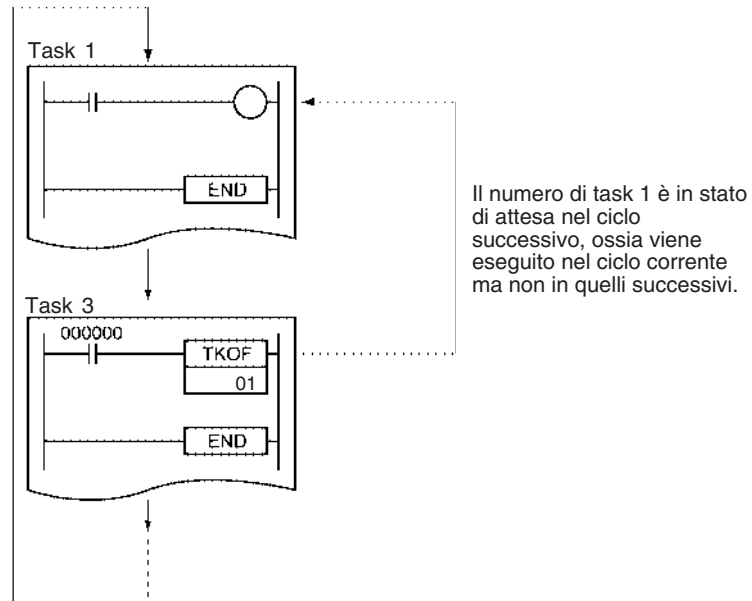
Nell'esempio che segue, quando CIO 000000 è impostato su ON, al numero di task 3 è assegnato lo stato di attesa nel numero di task 1. Il numero di task 3 non verrà eseguito in quel ciclo quando l'esecuzione del programma raggiunge il numero di task 3.



Il numero di task 3 è in stato di attesa nello stesso ciclo, ossia non viene eseguito nel ciclo corrente o in quelli successivi.

Specifica di un task precedente

Nell'esempio che segue, quando CIO 000000 è impostato su ON, al numero di task 1 è assegnato lo stato di attesa nel numero di task 3. Il numero di task 1 non verrà eseguito nel ciclo successivo quando l'esecuzione del programma raggiunge il numero di task 1.



3-35 Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva)

In questa sezione vengono descritte le istruzioni utilizzate nella modifica dei modelli del PLC.

| Istruzione | Mnemonico | Codice funzione | Pagina |
|------------------------|-----------|-----------------|--------|
| BLOCK TRANSFER | XFERC | 565 | 1201 |
| SINGLE WORD DISTRIBUTE | DISTC | 566 | 1203 |
| DATA COLLECT | COLLC | 567 | 1206 |
| MOVE BIT | MOVBC | 568 | 1211 |
| BIT COUNTER | BCNTC | 621 | 1212 |

Le istruzioni di conversione del modello offrono la stessa funzionalità delle altre istruzioni, ma, come le istruzioni per la serie C, utilizzano dati BCD per gli operandi (laddove le serie CJ/CS utilizzano dati binari). Come riportato nella tabella qui sopra, sono disponibili cinque istruzioni di conversione del modello; tutte presentano una C alla fine del codice mnemonico della funzione equivalente per i dati binari dell'operando.

Le istruzioni di conversione del modello consentono di convertire i programmi della serie C in programmi della serie CS/CJ senza modificare o dati dell'operando per queste istruzioni.

Nella conversione di programmi della serie C in programmi della serie CS/CJ su CX-Programmer versione 5.0 o successiva (vedere nota), queste istruzioni vengono utilizzate in modo automatico (ad esempio XFER diventerà XFERC), senza che vi sia la necessità di procedere alla correzione manuale dei dati degli operandi.

Nella conversione di programmi della serie C in programmi della serie CS/CJ su CX-Programmer versione 4.0 o precedente (vedere nota), ogni operando per il quale è specificata una costante viene convertito da BCD al formato binario; tuttavia sarà necessario correggere manualmente i dati degli operandi per i quali è specificato un indirizzo di canale.

Nota Si procede alla conversione specificando la serie CS/CJ come "tipo di dispositivo" nella finestra Modifica del PLC.

Differenze dalle istruzioni per la serie C La "serie C" include C200H, C1000H, C2000H, C200HS, C2000HX/HG/HE (-Z), CQM1, CQM1H, CPM1/CPM1A, CPM2C ed SRM1.

| Nome | Istruzione di conversione del modello (CPU versione 3.0 o successiva) | Istruzione corrispondente per la serie C | Differenze dalle istruzioni per la serie C | | Conversione del tipo di dispositivo in CS/CJ con CX-Programmer Ver. 4.0 o precedente | Conversione del tipo di dispositivo in CS/CJ con CX-Programmer Ver. 5.0 o successiva |
|------------------------|---|--|---|--|--|--|
| | Mnemonico (codice funzione) | Mnemonico (codice funzione) | C200H, C1000H o C2000H | C200HS, C2000HX/HG/HE(-Z), CQM1, CQM1H, CPM1/CPM1A, CPM2C o SRM1 | | |
| BLOCK TRANSFER | XFERC(565) | XFER(70) | Uguale | Uguale | Convertita in XFER. Se un indirizzo di canale viene specificato per il primo operando (numero di canali da trasferire), nel programma dovrà essere corretto manualmente in dati binari. | XFER è convertita in XFERC. Gli operandi non richiedono correzioni. |
| SINGLE WORD DISTRIBUTE | DISTC(566) | DIST(80) | Oltre all'operazione di distribuzione dei dati, offre l'operazione PUSH ONTO STACK, precedentemente non supportata. | Uguale (operazione di distribuzione e PUSH ONTO STACK) | Convertita in DIST. Se un indirizzo di canale viene specificato per il terzo operando (dati offset), nel programma dovrà essere corretto manualmente in dati binari. | DIST è convertita in DISTC. Gli operandi non richiedono correzioni. |
| DATA COLLECT | COLLC(567) | COLL(81) | Oltre all'operazione di raccolta dei dati, offre l'operazione STACK DATA READ, precedentemente non supportata. | Uguale (operazione di raccolta dati e STACK DATA READ) | Convertita in COLL. Se un indirizzo di canale viene specificato per il secondo operando (dati offset), nel programma dovrà essere corretto manualmente in dati binari. | COLL è convertita in COLLC. Gli operandi non richiedono correzioni. |
| MOVE BIT | MOVBC(568) | MOVB(82) | Uguale | Uguale | Convertita in MOVBC. Se un indirizzo di canale viene specificato per il secondo operando (dati di controllo), nel programma dovrà essere corretto manualmente in dati binari. | MOVB è convertita in MOVBC. Gli operandi non richiedono correzioni. |
| BIT COUNTER | BCNTC(621) | BCNT(67) | Uguale | Uguale | Convertita in BCNT. Se un indirizzo di canale viene specificato per il primo operando (numero di canali da conteggiare), nel programma dovrà essere corretto manualmente in dati binari. | BCNT è convertita in BCNTC. Gli operandi non richiedono correzioni. |

Nota Le differenze nel funzionamento dei flag di condizione sono illustrate qui di seguito. Per ulteriori informazioni, fare riferimento alle descrizioni dei flag di condizione di ciascuna istruzione.

- Se il contenuto di un'area DM utilizzata per l'indirizzamento indiretto non è BCD (*BCD) o se non viene rispettato l'intervallo di indirizzamento dell'area DM, il funzionamento dei flag di condizione varia da un'istruzione all'altra.
- Per DISTC(566), il funzionamento dei flag di condizione è diverso da quello dei modelli C200H, C1000H e C2000H per quanto riguarda l'operazione PUSH ONTO STACK.
- Per COLL(567), il funzionamento dei flag di condizione è diverso da quello dei modelli C200H, C1000H e C2000H per quanto riguarda l'operazione STACK DATA READ.

Differenze dalle istruzioni delle serie CS/CJ precedenti

| Nome | Istruzione di conversione del modello (CPU versione 3.0 o successiva) | Istruzione corrispondente per la serie C | Differenze dalle istruzioni delle serie CS/CJ precedenti |
|------------------------|---|--|---|
| | Mnemonico (codice funzione) | Mnemonico (codice funzione) | |
| BLOCK TRANSFER | XFERC(565) | XFER(70) | Il tipo di dati per il primo operando (numero di canali da trasferire) è BCD (da 0000 a 9999) invece che binario (da 0000 a FFFF esadecimale). |
| SINGLE WORD DISTRIBUTE | DISTC(566) | DIST(80) | Oltre all'operazione di distribuzione dei dati, è supportata un'operazione PUSH ONTO STACK. Il tipo di dati per il terzo operando (dati offset) è BCD (distribuzione dei dati: da 0000 a 7999, PUSH ONTO STACK: da 0000 a 9999) invece che binario (da 0000 a FFFF esadecimale). |

| Nome | Istruzione di conversione del modello (CPU versione 3.0 o successiva) | Istruzione corrispondente per la serie C | Differenze dalle istruzioni delle serie CS/CJ precedenti |
|--------------|---|--|--|
| | Mnemonico (codice funzione) | Mnemonico (codice funzione) | |
| DATA COLLECT | COLLC(567) | COLL(81) | Oltre all'operazione di distribuzione dei dati, è supportata un'operazione STACK DATA READ. Il tipo di dati per il secondo operando (dati offset) è BCD (distribuzione dei dati: da 0000 a 7999, STACK DATA READ per FIFO: da 9000 a 9999, STACK DATA READ per LIFO: da 8000 a 8999) invece che binario (da 0000 a FFFF esadecimale). |
| MOVE BIT | MOVBC(568) | MOVB(82) | Il tipo di dati per le specifiche del bit di origine e di destinazione nel secondo operando (dati di controllo) è BCD (da 00 a 15) invece che binario (da 00 a 0F esadecimale). |
| BIT COUNTER | BCNTC(621) | BCNT(67) | Il tipo di dati per il primo operando (numero di canali da conteggiare) è BCD (da 0000 a 9999) invece che binario (da 0000 a FFFF esadecimale). Il tipo di dati memorizzati per il terzo operando (risultati del conteggio) è BCD (da 0000 a 9999) invece che binario (da 0000 a FFFF esadecimale). |

Nota Le differenze nel funzionamento dei flag di condizione sono illustrate qui di seguito. Per ulteriori informazioni, fare riferimento alle descrizioni dei flag di condizione di ciascuna istruzione.

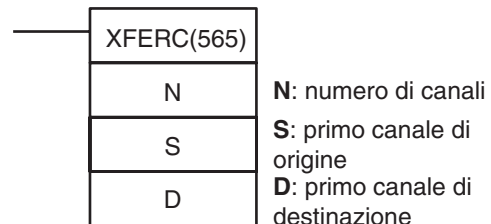
- Se i dati per gli operandi sopra menzionati non sono BCD, verrà attivato il flag di errore.
- Per DISTC(566), il funzionamento dei flag di condizione era stato implementato per l'operazione PUSH ONTO STACK.
- Per COLLC(567), il funzionamento dei flag di condizione era stato implementato per l'operazione STACK DATA READ.

3-35-1 BLOCK TRANSFER: XFERC(565)

Scopo

Trasferisce il numero di canali consecutivi specificato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | XFERC(565) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @XFERC(565) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

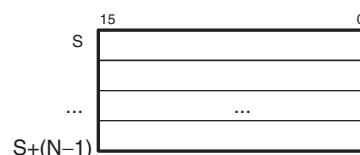
Operandi

N: numero di canali

Specifica il numero di canali da trasferire. L'intervallo possibile per il valore di N è compreso tra 0000 e 9999 BCD.

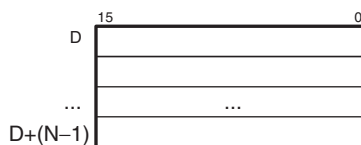
S: primo canale di origine

Specifica il primo canale di origine.



D: primo canale di destinazione

Specifica il primo canale di destinazione.

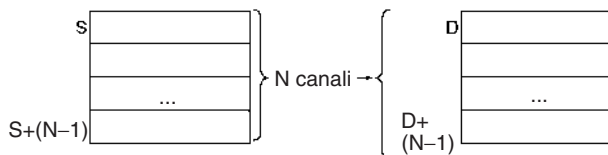


Caratteristiche operando

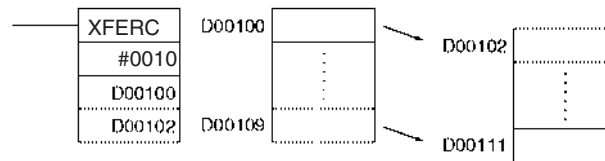
| Area | N | S | D |
|--|--|-----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #9999 (BCD) | --- | --- |
| Registri dati | Da DR0 a DR15 | --- | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

XFERC(565) copia N canali che iniziano con S (da S a S+(N-1)) negli N canali che iniziano con D (da D a D+(N-1)).



È possibile sovrapporre i canali di origine e di destinazione, per cui XFERC(565) consente di eseguire operazioni di scorrimento del canale.



Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se i dati in N (numero dei canali) non sono in formato BCD. |

Nota Nei PLC della serie C, l'istruzione BLOCK TRANSFER (XFER) provoca l'attivazione del flag di errore se il contenuto di un canale DM (*DM) indirizzato indirettamente non è BCD o se viene superato il contorno dell'area DM. In questi casi XFERC(565) non provoca l'attivazione del flag di errore.

Avvertenze

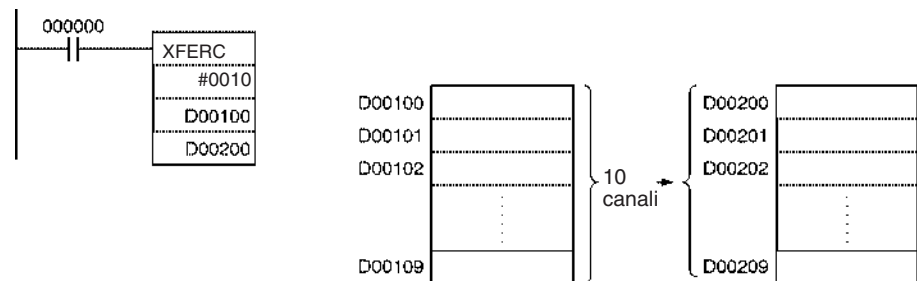
Assicurarsi che i canali di origine (da S a S+N-1) e di destinazione (da D a D+N-1) non superino la fine dell'area dati.

Quando viene trasferito un numero elevato di canali, il completamento di XFERC(565) può non essere rapido. In questo caso, se avviene una caduta di tensione durante l'esecuzione dell'istruzione, è possibile che il trasferimento di XFERC(565) non venga completato.

Il contenuto di N deve essere in formato BCD. In caso contrario si verificherà un errore e verrà attivato il flag di errore.

Esempio

Quando CIO 000000 è ON, i 10 canali da D00100 a D00109 vengono copiati nell'intervallo compreso tra D00200 e D00209.

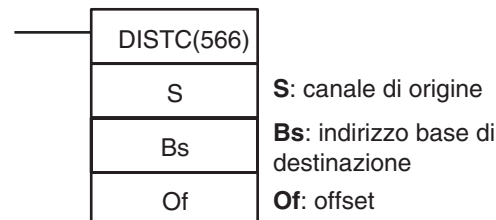


3-35-2 SINGLE WORD DISTRIBUTE: DISTC(566)

Scopo

Trasferisce il canale di origine in un canale di destinazione calcolato aggiungendo il valore di offset all'indirizzo di base.

Simbolo programmazione ladder



Variazioni

| | | |
|-------------------------|--|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | DISTC(566) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @DISTC(566) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

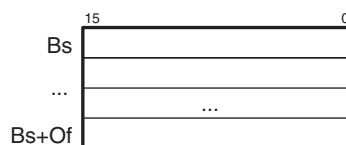
Operandi

Bs: indirizzo base di destinazione

Specifica l'indirizzo base di destinazione. L'offset viene aggiunto a questo indirizzo per calcolare il canale di destinazione.

Of: offset

- Operazione di distribuzione dei dati (da 0000 a 7999 BCD)
Questo valore viene aggiunto all'indirizzo base per calcolare il canale di destinazione. L'offset può avere qualsiasi valore da 0000 a 7999 in formato BCD, ma Bs e Bs+Of devono essere nella stessa area dati.



- Operazione PUSH ONTO STACK (da 9000 a 9999 BCD)
Quando la cifra più a sinistra di Of è 9, le tre cifre più a destra di Of specificano il numero di canali nello stack. L'offset può avere qualsiasi valore da 9000 a 9999 in formato BCD.

Caratteristiche operando

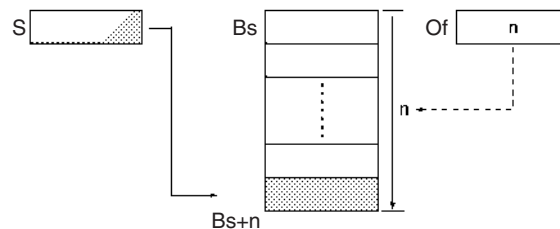
| Area | S | Bs | Of |
|--------------------------------------|--|-----------------|---|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | Da A448 ad A959 | Da A000 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @E00000 a @E32767 Da @En_00000 a @En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0000 a #FFFF (binario) | --- | Da #0000 a #7999 per la distribuzione Da #9000 a #9999 per l'operazione di stack |

| Area | S | Bs | Of |
|--|--|-----|---------------|
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

Operazione di distribuzione dei dati

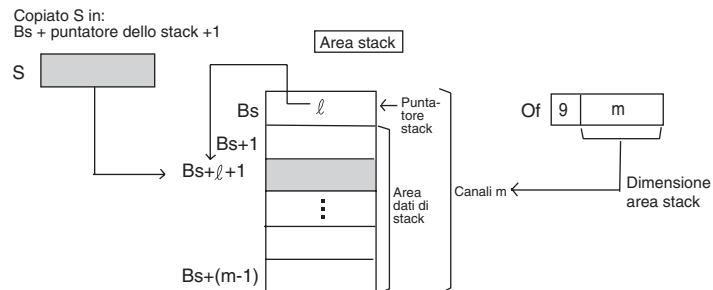
DISTC(566) copia S nel canale di destinazione calcolato aggiungendo Of a Bs. Modificando il valore di Of, è possibile utilizzare la stessa istruzione DISTC(566) per distribuire il canale di origine in vari canali dell'area dati.



Operazione PUSH ONTO STACK

Quando la cifra più a sinistra (bit da 12 a 15) di Of è 9 BCD, DISTC(566) elabora uno stack da Bs a Bs+Of-9000. L'indirizzo base di destinazione (Bs) contiene il puntatore dello stack e gli altri canali nello stack contengono i dati di stack.

DISTC(566) copia S nel canale di destinazione calcolato aggiungendo il puntatore dello stack (contenuto di Bs) + 1 all'indirizzo Bs. Modificando il valore di Of, è possibile utilizzare la stessa istruzione DISTC(566) per distribuire il canale di origine in vari canali dell'area dati.



Ogni volta che il contenuto di S è copiato in un canale dell'area dati di stack, il puntatore dello stack in Bs viene incrementato automaticamente di +1.

Nota Utilizzare COLLC(567) per leggere i dati di stack dall'area stack.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se è specificata l'operazione PUSH ONTO STACK, ma i dati del puntatore dello stack in Bs non sono BCD. ON se è specificata l'operazione PUSH ONTO STACK e il puntatore dello stack indica un canale che supera l'area dati di stack. |
| Flag di uguaglianza | = | ON se i dati di origine sono 0000. OFF in tutti gli altri casi. |

Nota Nei PLC della serie C, l'istruzione SINGLE WORD DISTRIBUTE (DIST) provoca l'attivazione del flag di errore se il contenuto di un canale DM (*DM)

indirizzato indirettamente non è BCD o se viene superato il contorno dell'area DM. In questi casi DISTC(566) non provoca l'attivazione del flag di errore.

Avvertenze

Una volta eseguita DISTC(566) con l'operazione PUSH ONTO STACK per assegnare un'area stack, specificare sempre la stessa lunghezza dell'area stack nelle successive istruzioni DISTC(566). Se nelle successive istruzioni DISTC(566) viene specificata una dimensione diversa dell'area stack, il funzionamento non sarà attendibile.

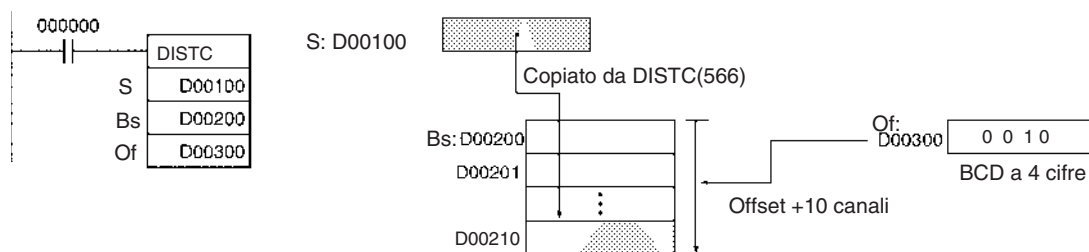
Verificare che, in seguito all'aggiunta in Bs, le dimensioni dell'offset o dello stack specificate da Of non superino la fine dell'area dati.

Esempi

Operazione di distribuzione dei dati

Il byte più a sinistra di D00300 è 0, quindi DISTC(566) esegue l'operazione di distribuzione dei dati.

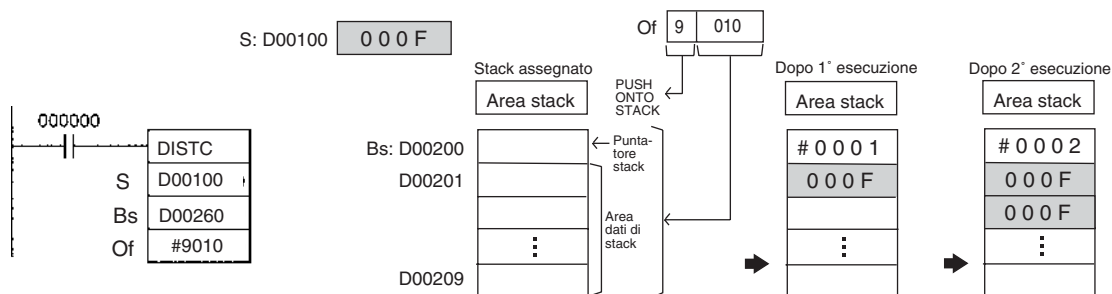
Nell'esempio che segue, quando CIO 000000 è ON, il contenuto di D00100 viene copiato in D00210 (D00200 + 10) purché il contenuto di D00300 sia 0010 BCD. Modificando l'offset in D00300, è possibile copiare il contenuto di D00100 in altri canali.



Operazione PUSH ONTO STACK

Il byte più a sinistra di Of è 9, quindi DISTC(566) esegue l'operazione PUSH ONTO STACK.

Nell'esempio che segue, quando CIO 000000 è ON, DISTC(566) assegna un'area stack a 10 canali (le 3 cifre più a destra di Of sono #010) tra D00200 e D00209. Allo stesso tempo, il contenuto di D00100 viene copiato nel canale calcolato aggiungendo D00200 + puntatore dello stack +1. Infine, il puntatore dello stack viene incrementato di +1.

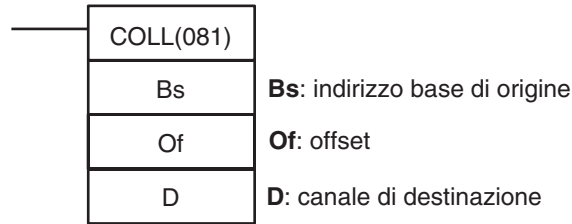


3-35-3 DATA COLLECT: COLLC(567)

Scopo

Trasferisce il canale di origine, calcolato aggiungendo un valore di offset all'indirizzo di base, nel canale di destinazione.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | COLLC(567) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @COLLC(567) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

Bs: indirizzo base di origine

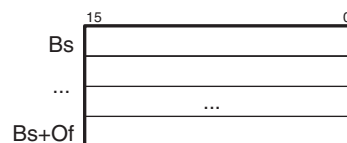
Specifica l'indirizzo base di origine. L'offset viene aggiunto a questo indirizzo per calcolare il canale di origine.

Of: offset

Il valore di Of determina il funzionamento di COLLC(567).

- Operazione di raccolta dei dati (Of = da 0000 a 7999 BCD)

Il valore Of viene aggiunto all'indirizzo base per calcolare il canale di origine. L'offset può avere qualsiasi valore da 0000 a 7999 BCD, ma Bs e Bs+Of devono essere nella stessa area dati.



- Operazione STACK DATA READ LIFO (Of = da 8000 a 8999 BCD)
Se la cifra più a sinistra di Of è 8, COLLC(567) funziona come un'istruzione di stack LIFO. Lo stack inizia in Bs con la lunghezza specificata nelle 3 cifre più a destra di Of.
- Operazione STACK DATA READ FIFO (Of = da 9000 a 9999 BCD)
Se la cifra più a sinistra di Of è 9, COLLC(567) funziona come un'istruzione di stack FIFO. Lo stack inizia in Bs con la lunghezza specificata nelle 3 cifre più a destra di Of.

Caratteristiche operando

| Area | Bs | Of | D |
|-------------------------|------------------------|----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritentività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |

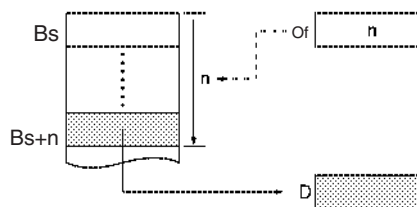
| Area | Bs | Of | D |
|--|--|---|-----|
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | Da #0000 a #7999 per la raccolta dei dati Da #8000 a #8999 per STACK DATA READ LIFO Da #9000 a #9999 per STACK DATA READ FIFO | --- |
| Registri dati | --- | Da DR0 a DR15 | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

In base al valore di Of, COLLC(567) funziona come istruzione di raccolta dei dati, istruzione di stack FIFO o istruzione di stack LIFO.

Operazione di raccolta dei dati (Of = da 0000 a 7999 BCD)

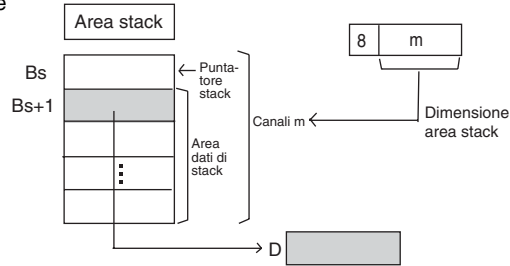
COLLC(567) copia il canale di origine (calcolato aggiungendo Of a Bs) nel canale di destinazione. Modificando il valore di Of, è possibile utilizzare la stessa istruzione COLLC(567) per raccogliere nell'area dati i dati provenienti da diversi canali di origine.

**Operazione STACK DATA READ LIFO (Of = da 8000 a 8999 BCD)**

Se la cifra più a sinistra di Of è 8, COLLC(567) funziona come un'istruzione di stack LIFO (LIFO significa Last-In-First-Out). In questo caso, le 3 cifre più a destra di Of specificano le dimensioni dello stack.

COLLC(567) copia in D i dati registrati nello stack nel periodo più recente. Il canale di origine è Bs + il puntatore dello stack (contenuto di Bs). Una volta che i dati sono stati copiati, il puntatore dello stack viene decrementato di 1.

I dati vengono copiati da Bs + puntatore dello stack.



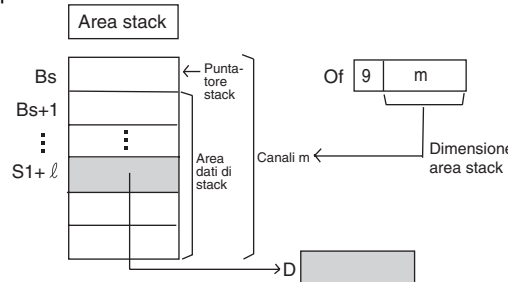
Nota Utilizzare DISTC(566) per scrivere i dati di stack nell'area stack.

Operazione STACK DATA READ FIFO (Of = da 9000 a 9999 BCD)

Se la cifra più a sinistra di Of è 9, COLLC(567) funziona come un'istruzione di stack FIFO (FIFO significa First-In-First-Out). In questo caso, le 3 cifre più a destra di Of specificano le dimensioni dello stack.

COLLC(567) copia in D i dati provenienti dal canale meno recente registrato nello stack. Il canale di origine è Bs + 1. Una volta che i dati sono stati copiati, il puntatore dello stack viene decrementato di 1.

I dati vengono copiati da Bs + 1.



Nota Utilizzare DISTC(566) per scrivere i dati di stack nell'area stack.

Flag

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|---|
| Flag di errore | ER | ON se i dati offset in Of non sono BCD. ON se è specificata un'operazione di stack LIFO o FIFO, ma i dati del puntatore dello stack in Bs non sono BCD. ON se è specificata un'operazione di stack LIFO o FIFO e il puntatore dello stack indica un canale che supera l'area dati di stack. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se i dati di origine sono 0000. OFF in tutti gli altri casi. |

Nota Nei PLC della serie C, l'istruzione DATA COLLECT (COLL) provoca l'attivazione del flag di errore se il contenuto di un canale DM (*DM) indirizzato indirettamente non è BCD o se viene superato il contorno dell'area DM. In questi casi COLLC(567) non provoca l'attivazione del flag di errore.

Avvertenze

Una volta eseguita DISTC(566) con l'operazione PUSH ONTO STACK per assegnare un'area stack, specificare sempre la stessa lunghezza dell'area stack nelle istruzioni COLLC(567). Se nelle istruzioni COLLC(567) viene specificata una dimensione diversa dell'area stack, il funzionamento non sarà attendibile.

Verificare che, in seguito all'aggiunta in Bs, le dimensioni dell'offset o dello stack specificate da Of non superino la fine dell'area dati.

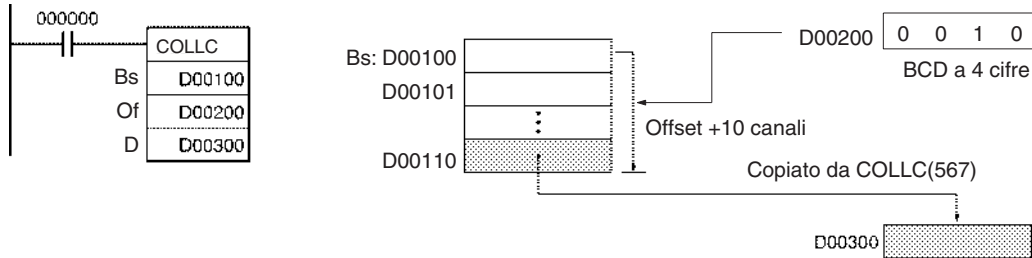
I dati offset in Of devono essere in formato BCD.

Esempi

Operazione di raccolta dei dati

Il byte più a sinistra di D00200 è 0, quindi COLLC(567) esegue l'operazione di raccolta dei dati.

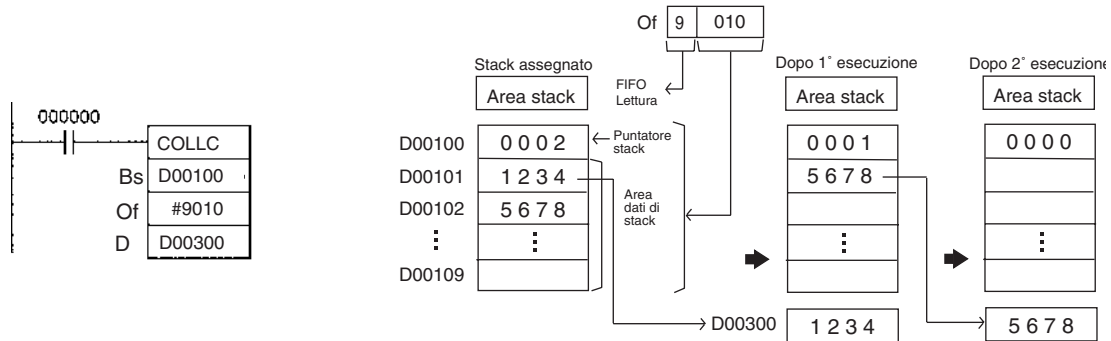
Nell'esempio che segue, quando CIO 000000 è ON, il contenuto di D00110 (D00100 + 10) viene copiato in D00300 purché il contenuto di D00200 sia 10 (0010 BCD). Modificando l'offset in D00200, è possibile copiare il contenuto degli altri canali in D00300.



Operazione di stack FIFO

Il byte più a sinistra di Of è 9, quindi COLLC(567) esegue l'operazione di stack FIFO.

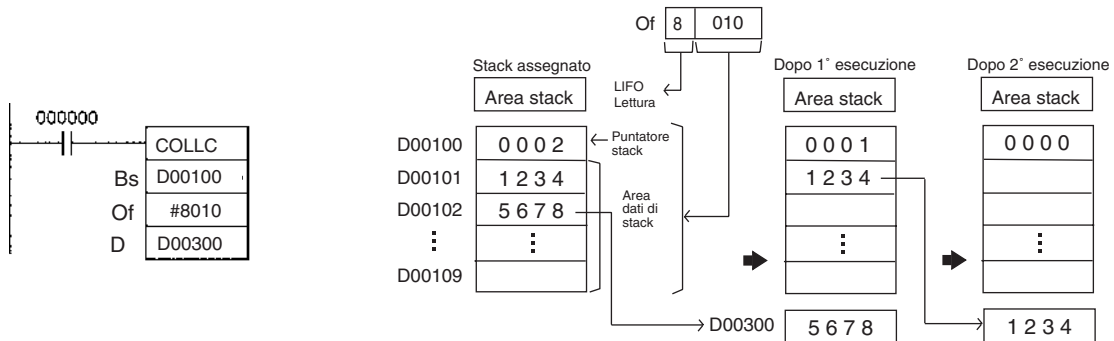
Nell'esempio che segue, quando CIO 000000 è ON, COLLC(567) assegna un'area stack a 10 canali (le 3 cifre più a destra di Of sono #010) tra D00100 e D00109. Allo stesso tempo, il contenuto di D00101 (Bs + 1) viene copiato in D00300. Infine, il puntatore dello stack viene decrementato di 1.



Operazione di stack LIFO

Il byte più a sinistra di Of è 8, quindi COLLC(567) esegue l'operazione di stack LIFO.

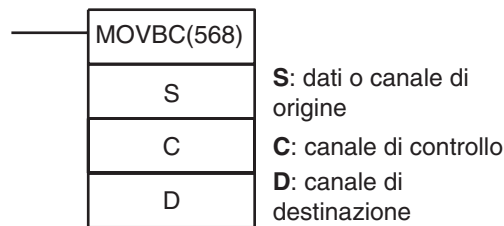
Nell'esempio che segue, quando CIO 000000 è ON, COLLC(567) assegna un'area stack a 10 canali (le 3 cifre più a destra di Of sono #010) tra D00100 e D00109. Allo stesso tempo, il contenuto del canale di origine (D00100 + puntatore dello stack) viene copiato in D00300. Infine, il puntatore dello stack viene decrementato di 1.



3-35-4 MOVE BIT: MOVBC(568)

Scopo Trasferisce il bit specificato.

Simbolo programmazione ladder



Variazioni

| Variazioni | Eseguita a ogni ciclo con condizione ON | MOVBC(568) |
|--------------------------------|--|----------------|
| | Eseguita una sola volta per differenziazione sul fronte di salita | @MOVBC(568) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

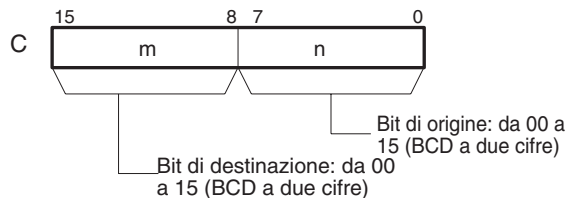
Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

C: canale di controllo

Le prime due cifre a destra di C indicano che il bit di S è il bit di origine, mentre le prime due cifre a sinistra di C indicano che il bit di D è il bit di destinazione.



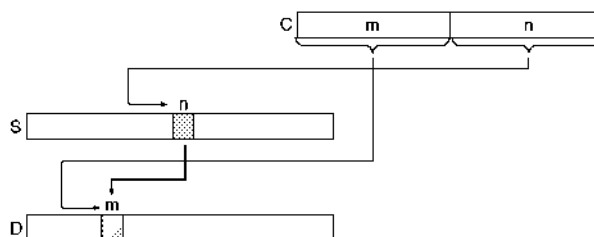
Caratteristiche operando

| Area | S | C | D |
|--------------------------------------|--|---|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |

| Area | S | C | D |
|--|--|-------------------------|-----|
| Costanti | Da #0000 a #FFFF (binario) | Solo valori specificati | --- |
| Registri dati | Da DR0 a DR15 | | |
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

MOVBC(568) copia il bit specificato (n) da S al bit specificato (m) in D. Gli altri bit nel canale di destinazione vengono lasciati inalterati.



Nota Per copiare un bit in un canale, è possibile specificare lo stesso canale per S e D.

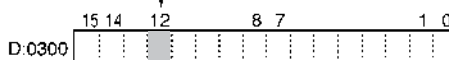
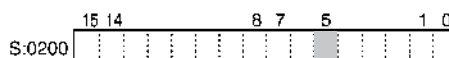
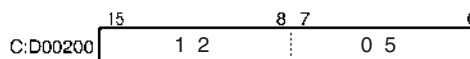
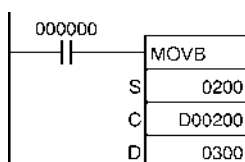
Flag

| Nome | Etichetta | Funzionamento |
|----------------|-----------|--|
| Flag di errore | ER | ON se le cifre più a destra e più a sinistra di C non sono BCD o non rientrano nell'intervallo specificato da 00 a 15. OFF in tutti gli altri casi. |

Nota Nei PLC della serie C, l'istruzione MOVE BIT (MOVB) provoca l'attivazione del flag di errore se il contenuto di un canale DM (*DM) indirizzato indirettamente non è BCD o se viene superato il contorno dell'area DM. In questi casi MOVBC(568) non provoca l'attivazione del flag di errore.

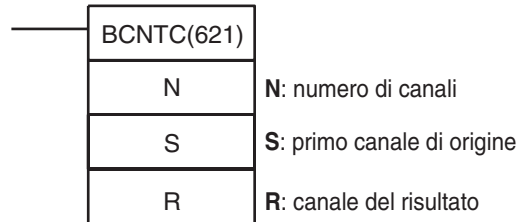
Esempi

Nell'esempio seguente, quando CIO 000000 è ON, il 5° bit del canale di origine (CIO 0200) viene copiato nel 12° bit del canale di destinazione (CIO 0300) in base al valore 1205 del canale di controllo.

**3-35-5 BIT COUNTER: BCNTC(621)****Scopo**

Conta il numero totale di bit impostati su ON presenti nei canali specificati.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | BCNTC(621) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @BCNTC(621) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

N: numero di canali

Il numero di canali deve essere compreso tra 0001 e 9999 (BCD).

S: primo canale di origine

S ed S+(N-1) devono essere nella stessa area dati.

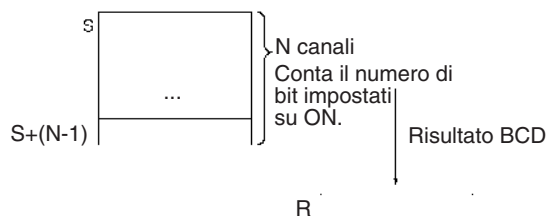
Caratteristiche operando

| Area | N | S | R |
|--------------------------------------|--|-----|-----------------|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | Da A448 ad A959 |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | Da #0001 a #9999 (BCD) | --- | |
| Registri dati | Da DR0 a DR15 | --- | Da DR0 a DR15 |

| Area | N | S | R |
|--|---|---|---|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

BCNTC(621) conteggia il numero totale dei bit impostati su ON in tutti i canali compresi tra S ed S+(N-1) e invia il risultato in formato BCD a R.

**Flag**

| Nome | Etichetta | Funzionamento |
|---------------------|-----------|--|
| Flag di errore | ER | ON se N non rientra nell'intervallo compreso tra 0001 e 9999 BCD. ON se il risultato supera 9999 BCD. OFF in tutti gli altri casi. |
| Flag di uguaglianza | = | ON se il risultato è 0000. OFF in tutti gli altri casi. |

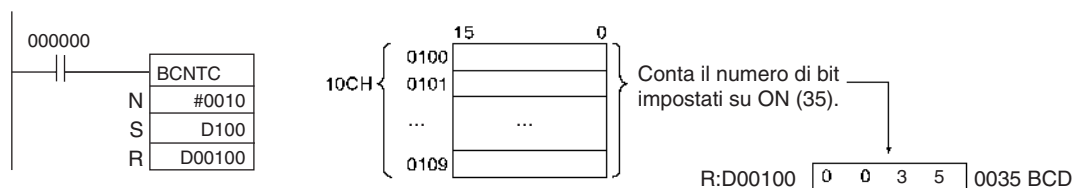
Nota Nei PLC della serie C, l'istruzione BIT COUNTER (BITC) provoca l'attivazione del flag di errore se il contenuto di un canale DM (*DM) indirizzato indirettamente non è BCD o se viene superato il contorno dell'area DM. In questi casi BCNTC(621) non provoca l'attivazione del flag di errore.

Avvertenze

Se N non è un valore BCD compreso tra 0001 e 9999 o se il risultato supera 9.999, si verificherà un errore.

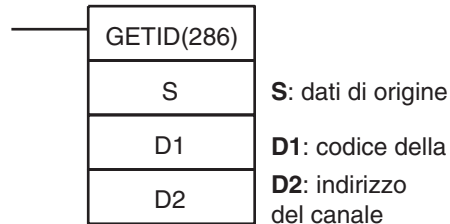
Esempio

Nell'esempio seguente, quando CIO 000000 è impostato su ON, BCNTC(621) conta il numero totale dei bit impostati su ON nei 10 canali compresi tra CIO 0100 e CIO 0109 e scrive il risultato in D00100.

**3-35-6 GET VARIABLE ID: GETID(286)****Scopo**

Invia il codice del tipo di variabile del comando FINS (area dati) e l'indirizzo di canale per la variabile o l'indirizzo specificato. Questa istruzione viene in genere utilizzata per ottenere l'indirizzo assegnato a una variabile in un blocco funzione.

Simbolo programmazione ladder



Variazioni

| | | |
|--------------------------------|---|----------------|
| Variazioni | Eseguita a ogni ciclo con condizione ON | GETID(286) |
| | Eseguita una sola volta per differenziazione sul fronte di salita | @GETID(286) |
| | Eseguita una sola volta per differenziazione sul fronte di discesa | Non supportata |
| Aggiornamento immediato | | Non supportata |

Aree di programma applicabili

| Aree di programma a blocchi | Aree di programma step | Subroutine | Task ad interrupt |
|-----------------------------|------------------------|------------|-------------------|
| OK | OK | OK | OK |

Operandi

S: Dati di origine

Specifica la variabile o l'indirizzo per i quali avviene il recupero del tipo di variabile e dell'indirizzo di canale.

D1: codice della variabile

Contiene il codice del tipo di variabile FINS (codice dell'area dati) dei dati di origine.

D2: indirizzo del canale

Contiene l'indirizzo del canale dei dati di origine esadecimale a 4 cifre.

Caratteristiche operando

| Area | S | D1 | D2 |
|--------------------------------------|--|----|----|
| Area CIO | Da CIO 0000 a CIO 6143 | | |
| Area di lavoro | Da W000 a W511 | | |
| Area bit di ritenività | Da H000 a H511 | | |
| Area bit ausiliaria | Da A000 ad A959 | | |
| Area del temporizzatore | Da T0000 a T4095 | | |
| Area del contatore | Da C0000 a C4095 | | |
| Area DM | Da D00000 a D32767 | | |
| Area EM senza banco | Da E00000 a E32767 | | |
| Area EM con banco | Da En_00000 a En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in binario | Da @D00000 a @D32767 Da @ E00000 a @ E32767 Da @ En_00000 a @ En_32767 (n = da 0 a C) | | |
| Indirizzi DM/EM indiretti in BCD | Da *D00000 a *D32767 Da *E00000 a *E32767 Da *En_00000 a *En_32767 (n = da 0 a C) | | |
| Costanti | --- | | |
| Registri dati | Da DR0 a DR15 | | |

| Area | S | D1 | D2 |
|--|--|----|----|
| Registri indice | --- | | |
| Indirizzamento indiretto con i registri indice | Da ,IR0 a ,IR15 Da -2048 a +2047, da ,IR0 a da -2048 a +2047, a ,IR15 Da DR0 a DR15, da IR0 a IR15 Da ,IR0+(++) a ,IR15+(++) Da ,-(--)IR0 a ,-(--)IR15 | | |

Descrizione

GETID(286) recupera l'indirizzo dell'area dati per la variabile di origine o l'indirizzo specificato, invia il codice dell'area dati in D1 come esadecimale a 4 cifre e invia il numero dell'indirizzo del canale in D2 come esadecimale a 4 cifre.

Nella tabella che segue sono riportati i codici del tipo di variabile (area dati) e gli intervalli di indirizzo corrispondenti per le aree dati del PLC.

| Area dati | | Dimen- sione dati | Codice dell'area dati (inviato in D1) | Indirizzo (inviato in D2) |
|------------------------------|-----------------------|-------------------------|---|---|
| Area CIO | CIO | Canale | 00B0 esadecimale | Da 0000 a 17FF esadecimale (da 0000 a 6143) |
| Area di lavoro | W | | 00B1 esadecimale | Da 0000 a 01FF esadecimale (da 000 a 511) |
| Area bit di ritentività | H | | 00B2 esadecimale | Da 0000 a 01FF esadecimale (da 000 a 511) |
| Area bit ausiliaria | | | 00B3 esadecimale | Da 0000 a 03BF esadecimale (da 000 a 959) |
| Area DM | | | 0082 esadecimale | Da 0000 a 7FFF esadecimale (da 00000 a 32767) |
| Area EM (banco specifico) | En_ (n = da 0 a C) | | Da 00A0 a 00AC esadecimale | Da 0000 a 7FFF esadecimale (da 00000 a 32767) |
| Area EM (banco corrente) | E | | 0098 esadecimale | Da 0000 a 7FFF esadecimale (da 00000 a 32767) |

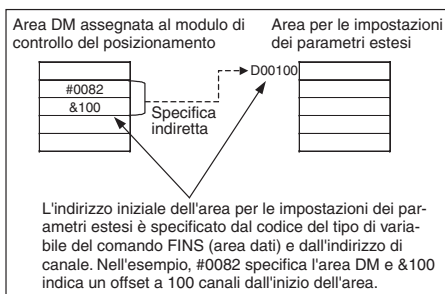
Se non viene utilizzata la specifica AT, CX-Programmer Ver. 5.0 e successiva assegna indirizzi alle variabili dei blocchi funzione in modo automatico. Ad esempio, se è necessario specificare indirettamente le impostazioni estese dei parametri di un modulo speciale, come un modulo di controllo del posizionamento, e si utilizza una variabile all'inizio dell'area delle impostazioni dei parametri estesi, dovrà essere definito l'indirizzo di tale variabile. In questo caso, è possibile utilizzare GETID(286) per recuperare l'indirizzo dell'area dati della variabile.

Flag

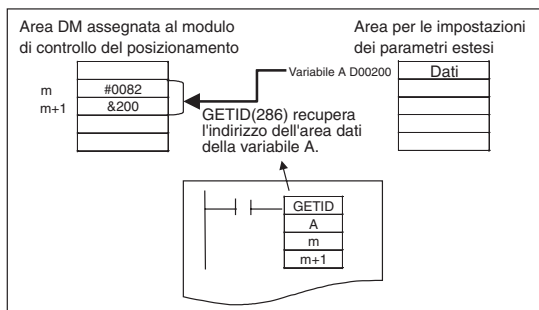
| Nome | Etichetta | Funzionamento |
|----------------|-----------|---|
| Flag di errore | ER | ON se S non è nell'intervallo consentito. |

Esempio

Funzionamento normale



Utilizzo dei blocchi funzione



CAPITOLO 4

Tempi di esecuzione delle istruzioni e numero di step

In questa sezione vengono forniti i tempi di esecuzione delle istruzioni e il numero di step necessari per ciascuna istruzione della serie CS/CJ.

| | | |
|--------|---|------|
| 4-1 | Tempi di esecuzione delle istruzioni e numero di step per la serie CS | 1221 |
| 4-1-1 | Istruzioni di ingresso sequenza | 1222 |
| 4-1-2 | Istruzioni di uscita sequenza | 1223 |
| 4-1-3 | Istruzioni di controllo sequenza | 1224 |
| 4-1-4 | Istruzioni di temporizzatore e contatore | 1225 |
| 4-1-5 | Istruzioni di confronto | 1226 |
| 4-1-6 | Istruzioni di spostamento dei dati | 1228 |
| 4-1-7 | Istruzioni di scorrimento dei dati | 1229 |
| 4-1-8 | Istruzioni di incremento e decremento | 1230 |
| 4-1-9 | Istruzioni matematiche con simboli | 1231 |
| 4-1-10 | Istruzioni di conversione | 1233 |
| 4-1-11 | Istruzioni logiche | 1235 |
| 4-1-12 | Istruzioni matematiche speciali | 1235 |
| 4-1-13 | Istruzioni matematiche a virgola mobile | 1236 |
| 4-1-14 | Istruzioni a virgola mobile in doppia precisione | 1237 |
| 4-1-15 | Istruzioni di elaborazione dei dati delle tabelle | 1238 |
| 4-1-16 | Istruzioni di controllo dei dati | 1240 |
| 4-1-17 | Istruzioni di subroutine | 1241 |
| 4-1-18 | Istruzioni di controllo degli interrupt | 1241 |
| 4-1-19 | Istruzioni di step | 1241 |
| 4-1-20 | Istruzioni per moduli di I/O di base | 1242 |
| 4-1-21 | Istruzioni per la comunicazione seriale | 1243 |
| 4-1-22 | Istruzioni di rete | 1243 |
| 4-1-23 | Istruzioni per la memoria dei file | 1244 |
| 4-1-24 | Istruzioni di visualizzazione | 1244 |
| 4-1-25 | Istruzioni per l'orologio | 1245 |
| 4-1-26 | Istruzioni di debug | 1245 |
| 4-1-27 | Istruzioni di diagnostica per malfunzionamento | 1245 |
| 4-1-28 | Altre istruzioni | 1246 |
| 4-1-29 | Istruzioni di programmazione a blocchi | 1246 |
| 4-1-30 | Istruzioni di elaborazione delle stringhe di testo | 1248 |
| 4-1-31 | Istruzioni di controllo dei task | 1249 |
| 4-1-32 | Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva) | 1249 |
| 4-1-33 | Istruzioni speciali per blocchi funzione (solo CPU versione 3.0 o successiva) | 1250 |
| 4-2 | Tempi di esecuzione delle istruzioni e numero di step per la serie CJ | 1250 |
| 4-2-1 | Istruzioni di ingresso sequenza | 1251 |
| 4-2-2 | Istruzioni di uscita sequenza | 1252 |
| 4-2-3 | Istruzioni di controllo sequenza | 1253 |
| 4-2-4 | Istruzioni di temporizzatore e contatore | 1254 |

| | | |
|--------|---|------|
| 4-2-5 | Istruzioni di confronto | 1255 |
| 4-2-6 | Istruzioni di spostamento dei dati | 1257 |
| 4-2-7 | Istruzioni di scorrimento dei dati | 1258 |
| 4-2-8 | Istruzioni di incremento e decremento | 1260 |
| 4-2-9 | Istruzioni matematiche con simboli | 1260 |
| 4-2-10 | Istruzioni di conversione | 1262 |
| 4-2-11 | Istruzioni logiche | 1264 |
| 4-2-12 | Istruzioni matematiche speciali | 1265 |
| 4-2-13 | Istruzioni matematiche a virgola mobile | 1265 |
| 4-2-14 | Istruzioni a virgola mobile in doppia precisione | 1267 |
| 4-2-15 | Istruzioni di elaborazione dei dati delle tabelle | 1268 |
| 4-2-16 | Istruzioni di controllo dei dati | 1269 |
| 4-2-17 | Istruzioni di subroutine | 1270 |
| 4-2-18 | Istruzioni di controllo degli interrupt | 1271 |
| 4-2-19 | Istruzioni per contatore veloce e uscita a impulsi | 1271 |
| 4-2-20 | Istruzioni di step | 1273 |
| 4-2-21 | Istruzioni per moduli di I/O di base | 1273 |
| 4-2-22 | Istruzioni per la comunicazione seriale | 1274 |
| 4-2-23 | Istruzioni di rete | 1275 |
| 4-2-24 | Istruzioni per la memoria dei file | 1276 |
| 4-2-25 | Istruzioni di visualizzazione | 1276 |
| 4-2-26 | Istruzioni per l'orologio | 1276 |
| 4-2-27 | Istruzioni di debug | 1277 |
| 4-2-28 | Istruzioni di diagnostica per malfunzionamento | 1277 |
| 4-2-29 | Altre istruzioni | 1278 |
| 4-2-30 | Istruzioni di programmazione a blocchi | 1278 |
| 4-2-31 | Istruzioni di elaborazione delle stringhe di testo | 1280 |
| 4-2-32 | Istruzioni di controllo dei task | 1281 |
| 4-2-33 | Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva) | 1281 |
| 4-2-34 | Istruzioni speciali per blocchi funzione (solo CPU versione 3.0 o successiva) | 1282 |
| 4-2-35 | Numero di step di programma del blocco funzione (CPU versione 3.0 o successiva) | 1282 |
| 4-2-36 | Linee guida per la conversione delle capacità del programma rispetto ai PLC OMRON precedenti | 1283 |
| 4-2-37 | Tempo di esecuzione delle istanze di blocchi funzione (CPU versione 3.0 o successiva) | 1284 |

4-1 Tempi di esecuzione delle istruzioni e numero di step per la serie CS

Nella seguente tabella sono riportati i tempi di esecuzione per tutte le istruzioni disponibili per i PLC della serie CS.

Il tempo di esecuzione totale delle istruzioni nell'ambito di un intero programma utente equivale al tempo di esecuzione del programma durante il calcolo del tempo di ciclo (vedere nota).

Nota Ai programmi utente vengono allocati dei task che possono essere eseguiti nell'ambito di task ciclici e di task ad interrupt che soddisfano le condizioni di interrupt.

I tempi di esecuzione della maggioranza delle istruzioni variano in base alla CPU utilizzata (CS1□-CPU6□H, CS1□-CPU6□, CS1□-CPU4□H, CS1□-CPU4□) e alle condizioni di esecuzione dell'istruzione. Nella seguente tabella sono riportati i dati relativi ai tempi di elaborazione di ciascuna istruzione: la riga superiore indica il tempo minimo insieme alle condizioni di esecuzione, mentre la riga inferiore indica il tempo massimo insieme alle condizioni di esecuzione.

Il tempo di esecuzione può inoltre variare quando la condizione di esecuzione è OFF.

Nella tabella seguente, nella colonna *Lunghezza (step)*, è inoltre riportata la lunghezza di ciascuna istruzione. Il numero di step necessari nell'area del programma utente per ciascuna delle istruzioni della serie CS varia da 1 a 7 in base all'istruzione e agli operandi specificati. Il numero di step in un programma non corrisponde al numero di istruzioni.

Nota 1. La capacità dei programmi per i PLC della serie CS viene misurata in step, mentre la capacità dei programmi per i PLC OMRON precedenti, ad esempio i PLC delle serie C e CV, veniva misurata in canali. In altre parole, 1 step equivale a 1 canale. Tuttavia, poiché per alcune istruzioni della serie CS la quantità di memoria richiesta per l'esecuzione è diversa rispetto ai modelli precedenti, la conversione per un PLC della serie CS della capacità di un programma utente sviluppato per un altro PLC in base al criterio secondo cui 1 canale equivale a 1 step non produrrà un valore accurato. Per indicazioni sulla conversione delle capacità di programmi sviluppati per PLC OMRON precedenti, fare riferimento alle informazioni riportate alla fine di 4-1 *Tempi di esecuzione delle istruzioni e numero di step per la serie CS*.

Per la maggior parte delle istruzioni è supportata la forma differenziale (contrassegnata con ↑, ↓, @ e %). L'utilizzo della differenziazione aumenta i tempi di esecuzione dei seguenti valori.

| Simbolo | CPU CS1-H | | CPU CS1 | |
|---------|-----------|--------|---------|-------|
| | CPU6□H | CPU4□H | CPU6□ | CPU4□ |
| ↑ o ↓ | +0,24 | +0,32 | +0,41 | +0,45 |
| @ o % | +0,24 | +0,32 | +0,29 | +0,33 |

2. Utilizzare i tempi seguenti come linee guida quando non vengono eseguite le istruzioni.

| CPU CS1-H | | CPU CS1 | |
|-----------|-----------|--------------------|--------------------|
| CPU6□H | CPU4□H | CPU6□ | CPU4□ |
| Circa 0,1 | Circa 0,2 | Circa da 0,1 a 0,3 | Circa da 0,2 a 0,4 |

4-1-1 Istruzioni di ingresso sequenza

| Istruzione | Mnemonico | Codice | Lunghezza (step) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|-------------------|-----------|--------|------------------|-----------------------------------|--------|--------|--------|-------------------------|
| | | | | CPU6□H | CPU4□H | CPU6□ | CPU4□ | |
| LOAD | LD | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| | !LD | --- | 2 | +21,14 | +21,16 | +21,16 | +21,16 | Aumento per la serie CS |
| | | | | +45,1 | +45,1 | +45,1 | +45,1 | Aumento per C200H |
| LOAD NOT | LD NOT | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| | !LD NOT | --- | 2 | +21,14 | +21,16 | +21,16 | +21,16 | Aumento per la serie CS |
| | | | | +45,1 | +45,1 | +45,1 | +45,1 | Aumento per C200H |
| AND | AND | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| | !AND | --- | 2 | +21,14 | +21,16 | +21,16 | +21,16 | Aumento per la serie CS |
| | | | | +45,1 | +45,1 | +45,1 | +45,1 | Aumento per C200H |
| AND NOT | AND NOT | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| | !AND NOT | --- | 2 | +21,14 | +21,16 | +21,16 | +21,16 | Aumento per la serie CS |
| | | | | +45,1 | +45,1 | +45,1 | +45,1 | Aumento per C200H |
| OR | OR | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| | !OR | --- | 2 | +21,14 | +21,16 | +21,16 | +21,16 | Aumento per la serie CS |
| | | | | +45,1 | +45,1 | +45,1 | +45,1 | Aumento per C200H |
| OR NOT | OR NOT | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| | !OR NOT | --- | 2 | +21,14 | +21,16 | +21,16 | +21,16 | Aumento per la serie CS |
| | | | | +45,1 | +45,1 | +45,1 | +45,1 | Aumento per C200H |
| AND LOAD | AND LD | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| OR LOAD | OR LD | --- | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| NOT | NOT | 520 | 1 | 0,02 | 0,04 | 0,04 | 0,08 | --- |
| CONDITION ON | UP | 521 | 3 | 0,3 | 0,42 | 0,46 | 0,54 | --- |
| CONDITION OFF | DOWN | 522 | 4 | 0,3 | 0,42 | 0,46 | 0,54 | --- |
| LOAD BIT TEST | LD TST | 350 | 4 | 0,14 | 0,24 | 0,25 | 0,37 | --- |
| LOAD BIT TEST NOT | LD TSTN | 351 | 4 | 0,14 | 0,24 | 0,25 | 0,37 | --- |
| AND BIT TEST NOT | AND TSTN | 351 | 4 | 0,14 | 0,24 | 0,25 | 0,37 | --- |
| OR BIT TEST | OR TST | 350 | 4 | 0,14 | 0,24 | 0,25 | 0,37 | --- |
| OR BIT TEST NOT | OR TSTN | 351 | 4 | 0,14 | 0,24 | 0,25 | 0,37 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-2 Istruzioni di uscita sequenza

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|--------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|-------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| OUTPUT | OUT | --- | 1 | 0,02 | 0,04 | 0,17 | 0,21 | --- |
| | !OUT | --- | 2 | +21,37 | +21,37 | +21,37 | +21,37 | Aumento per la serie CS |
| | | | | +49,3 | +49,3 | +49,3 | +49,3 | Aumento per C200H |
| OUTPUT NOT | OUT NOT | --- | 1 | 0,02 | 0,04 | 0,17 | 0,21 | --- |
| | !OUT NOT | --- | 2 | +21,37 | +21,37 | +21,37 | +21,37 | Aumento per la serie CS |
| | | | | +49,3 | +49,3 | +49,3 | +49,3 | Aumento per C200H |
| KEEP | KEEP | 011 | 1 | 0,06 | 0,08 | 0,25 | 0,29 | --- |
| DIFFERENTIATE UP | DIFU | 013 | 2 | 0,24 | 0,40 | 0,46 | 0,54 | --- |
| DIFFERENTIATE DOWN | DIFD | 014 | 2 | 0,24 | 0,40 | 0,46 | 0,54 | --- |
| SET | SET | --- | 1 | 0,02 | 0,06 | 0,17 | 0,21 | --- |
| | !SET | --- | 2 | +21,37 | +21,37 | +21,37 | +21,37 | Aumento per la serie CS |
| | | | | +49,3 | +49,3 | +49,3 | +49,3 | Aumento per C200H |
| RESET | RSET | --- | 1 | 0,02 | 0,06 | 0,17 | 0,21 | Canale specificato |
| | !RSET | --- | 2 | +21,37 | +21,37 | +21,37 | +21,37 | Aumento per la serie CS |
| | | | | +49,3 | +49,3 | +49,3 | +49,3 | Aumento per C200H |
| MULTIPLE BIT SET | SETA | 530 | 4 | 5,8 | 6,1 | 7,8 | 7,8 | Con impostazione di 1 bit |
| | | | | 25,7 | 27,2 | 38,8 | 38,8 | Con impostazione di 1.000 bit |
| MULTIPLE BIT RESET | RSTA | 531 | 4 | 5,7 | 6,1 | 7,8 | 7,8 | Con reset di 1 bit |
| | | | | 25,8 | 27,1 | 38,8 | 38,8 | Con reset di 1.000 bit |
| SINGLE BIT SET | SETB | 532 | 2 | 0,24 | 0,34 | --- | --- | --- |
| | !SETB | | 3 | +21,44 | +21,54 | --- | --- | --- |
| SINGLE BIT RESET | RSTB | 534 | 2 | 0,24 | 0,34 | --- | --- | --- |
| | !RSTB | | 3 | +21,44 | +21,54 | --- | --- | --- |
| SINGLE BIT OUTPUT | OUTB | 534 | 2 | 0,22 | 0,32 | --- | --- | --- |
| | !OUTB | | 3 | +21,42 | +21,52 | --- | --- | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-3 Istruzioni di controllo sequenza

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------------|---------|--------|--------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| END | END | 001 | 1 | 5,5 | 6,0 | 4,0 | 4,0 | --- |
| NO OPERATION | NOP | 000 | 1 | 0,02 | 0,04 | 0,08 | 0,12 | --- |
| INTERLOCK | IL | 002 | 1 | 0,06 | 0,06 | 0,12 | 0,12 | --- |
| INTERLOCK CLEAR | ILC | 003 | 1 | 0,06 | 0,06 | 0,12 | 0,12 | --- |
| MULTI-INTERLOCK DIFFERENTIATION HOLD (vedere la nota 2) | MILH | 517 | 3 | 6,1 | 6,5 | --- | --- | Durante l'interblocco |
| | | | | 7,5 | 7,9 | --- | --- | Non durante l'interblocco e senza l'interblocco impostato |
| | | | | 8,9 | 9,7 | --- | --- | Non durante l'interblocco e con l'interblocco impostato |
| MULTI-INTERLOCK DIFFERENTIATION RELEASE (vedere nota 2) | MILR | 518 | 3 | 6,1 | 6,5 | --- | --- | Durante l'interblocco |
| | | | | 7,5 | 7,9 | --- | --- | Non durante l'interblocco e senza l'interblocco impostato |
| | | | | 8,9 | 9,7 | --- | --- | Non durante l'interblocco e con l'interblocco impostato |
| MULTI-INTERLOCK CLEAR (vedere la nota 2) | MILC | 519 | 2 | 5,0 | 5,6 | --- | --- | Interblocco non cancellato |
| | | | | 5,7 | 6,2 | --- | --- | Interblocco cancellato |
| JUMP | JMP | 004 | 2 | 0,38 | 0,48 | 8,1 | 8,1 | --- |
| JUMP END | JME | 005 | 2 | --- | --- | --- | --- | --- |
| CONDITIONAL JUMP | CJP | 510 | 2 | 0,38 | 0,48 | 7,4 | 7,4 | Quando viene soddisfatta la condizione JMP |
| CONDITIONAL JUMP NOT | CJPN | 511 | 2 | 0,38 | 0,48 | 8,5 | 8,5 | Quando viene soddisfatta la condizione JMP |
| MULTIPLE JUMP | JMP0 | 515 | 1 | 0,06 | 0,06 | 0,12 | 0,12 | --- |
| MULTIPLE JUMP END | JME0 | 516 | 1 | 0,06 | 0,06 | 0,12 | 0,12 | --- |
| FOR LOOP | FOR | 512 | 2 | 0,52 | 0,54 | 0,12 | 0,21 | Designazione di una costante |
| BREAK LOOP | BREAK | 514 | 1 | 0,06 | 0,06 | 0,12 | 0,12 | --- |
| NEXT LOOP | NEXT | 513 | 1 | 0,18 | 0,16 | 0,17 | 0,17 | Se il ciclo continua |
| | | | | 0,22 | 0,40 | 0,12 | 0,12 | Se il ciclo termina |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-1-4 Istruzioni di temporizzatore e contatore

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|---------------------|-----------|--------|--------------------------------|-----------------------------|---------|---------|---------|----------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| TIMER | TIM | --- | 3 | 0,56 | 0,88 | 0,37 | 0,42 | --- |
| | TIMX | 550 | 3 | 0,56 | 0,88 | --- | --- | --- |
| COUNTER | CNT | --- | 3 | 0,56 | 0,88 | 0,37 | 0,42 | --- |
| | CNTX | 546 | 3 | 0,56 | 0,88 | --- | --- | --- |
| HIGH-SPEED TIMER | TIMH | 015 | 3 | 0,88 | 1,14 | 0,37 | 0,42 | --- |
| | TIMHX | 551 | 3 | 0,88 | 1,14 | --- | --- | --- |
| ONE-MS TIMER | TMHH | 540 | 3 | 0,86 | 1,12 | 0,37 | 0,42 | --- |
| | TMHHX | 552 | 3 | 0,86 | 1,12 | --- | --- | --- |
| ACCUMULATIVE TIMER | TTIM | 087 | 3 | 16,1 | 17,0 | 21,4 | 21,4 | --- |
| | | | | 10,9 | 11,4 | 14,8 | 14,8 | Durante il reset |
| | | | | 8,5 | 8,7 | 10,7 | 10,7 | Durante l'interblocco |
| | TTIMX | 555 | 3 | 16,1 | 17,0 | --- | --- | --- |
| | | | | 10,9 | 11,4 | --- | --- | Durante il reset |
| | | | | 8,5 | 8,7 | --- | --- | Durante l'interblocco |
| LONG TIMER | TIML | 542 | 4 | 7,6 | 10,0 | 12,8 | 12,8 | --- |
| | | | | 6,2 | 6,5 | 7,8 | 7,8 | Durante l'interblocco |
| | TIMLX | 553 | 4 | 7,6 | 10,0 | --- | --- | --- |
| | | | | 6,2 | 6,5 | --- | --- | Durante l'interblocco |
| MULTI-OUTPUT TIMER | MTIM | 543 | 4 | 20,9 | 23,3 | 26,0 | 26,0 | --- |
| | | | | 5,6 | 5,8 | 7,8 | 7,8 | Durante il reset |
| | MTIMX | 554 | 4 | 20,9 | 23,3 | --- | --- | --- |
| | | | | 5,6 | 5,8 | --- | --- | Durante il reset |
| REVERSIBLE COUNTER | CNTR | 012 | 3 | 16,9 | 19,0 | 20,9 | 20,9 | --- |
| | CNTRX | 548 | 3 | 16,9 | 19,0 | --- | --- | --- |
| RESET TIMER/COUNTER | CNR | 545 | 3 | 9,9 | 10,6 | 13,9 | 13,9 | Durante il reset di 1 canale |
| | | | | 4,16 ms | 4,16 ms | 5,42 ms | 5,42 ms | Durante il reset di 1.000 canali |
| | CNRX | 547 | 3 | 9,9 | 10,6 | --- | --- | Durante il reset di 1 canale |
| | | | | 4,16 ms | 4,16 ms | --- | --- | Durante il reset di 1.000 canali |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-5 Istruzioni di confronto

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---|--------------------|--------|--------------------------------|-----------------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| Istruzioni di confronto di ingresso (senza segno) | LD, AND, OR += | 300 | 4 | 0,10 | 0,16 | 0,21 | 0,37 | --- |
| | LD, AND, OR + <> | 305 | | | | | | |
| | LD, AND, OR + < | 310 | | | | | | |
| | LD, AND, OR +<= | 315 | | | | | | |
| | LD, AND, OR +> | 320 | | | | | | |
| | LD, AND, OR +>= | 325 | | | | | | |
| Istruzioni di confronto di ingresso (doppio, senza segno) | LD, AND, OR +=+L | 301 | 4 | 0,10 | 0,16 | 0,29 | 0,54 | --- |
| | LD, AND, OR +<>+L | 306 | | | | | | |
| | LD, AND, OR +<+L | 311 | | | | | | |
| | LD, AND, OR +<=+L | 316 | | | | | | |
| | LD, AND, OR +>+L | 321 | | | | | | |
| | LD, AND, OR +>=+L | 326 | | | | | | |
| Istruzioni di confronto di ingresso (con segno) | LD, AND, OR +=+S | 302 | 4 | 0,10 | 0,16 | 6,50 | 6,50 | --- |
| | LD, AND, OR +<>+S | 307 | | | | | | |
| | LD, AND, OR +<+S | 312 | | | | | | |
| | LD, AND, OR +<= | 317 | | | | | | |
| | LD, AND, OR +>+S | 322 | | | | | | |
| | LD, AND, OR +>=+S | 327 | | | | | | |
| Istruzioni di confronto di ingresso (doppio, con segno) | LD, AND, OR +=+SL | 303 | 4 | 0,10 | 0,16 | 6,50 | 6,50 | --- |
| | LD, AND, OR +<>+SL | 308 | | | | | | |
| | LD, AND, OR +<+SL | 313 | | | | | | |
| | LD, AND, OR +<=+SL | 318 | | | | | | |
| | LD, AND, OR +>+SL | 323 | | | | | | |
| | LD, AND, OR +>=+SL | 328 | | | | | | |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|--|-------------------|--------|--------------------------------|-----------------------------|---------|--------|--------|--|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| Istruzioni di confronto di date e ore (vedere la nota 2) | LD, AND, OR +DT | 341 | 4 | 25,1 | 36,4 | --- | --- | I tempi di esecuzione ON e OFF sono uguali a quelli indicati a sinistra. |
| | LD, AND, OR +<>DT | 342 | 4 | 25,2 | 36,4 | --- | --- | |
| | LD, AND, OR +<DT | 343 | 4 | 25,2 | 36,4 | --- | --- | |
| | LD, AND, OR +<=DT | 344 | 4 | 25,2 | 36,4 | --- | --- | |
| | LD, AND, OR +>DT | 345 | 4 | 25,1 | 36,4 | --- | --- | |
| | LD, AND, OR +>=DT | 346 | 4 | 25,2 | 36,4 | --- | --- | |
| COMPARE | CMP | 020 | 3 | 0,04 | 0,04 | 0,17 | 0,29 | --- |
| | !CMP | 020 | 7 | +42,1 | +42,1 | +42,4 | +42,4 | Aumento per la serie CS |
| | | | | +90,4 | +90,4 | +90,5 | +90,5 | Aumento per C200H |
| DOUBLE COMPARE | CMPL | 060 | 3 | 0,08 | 0,08 | 0,25 | 0,46 | --- |
| SIGNED BINARY COMPARE | CPS | 114 | 3 | 0,08 | 0,08 | 6,50 | 6,50 | --- |
| | !CPS | 114 | 7 | +35,9 | +35,9 | +42,4 | +42,4 | Aumento per la serie CS |
| | | | | +84,1 | +84,1 | +90,5 | +90,5 | Aumento per C200H |
| DOUBLE SIGNED BINARY COMPARE | CPSL | 115 | 3 | 0,08 | 0,08 | 6,50 | 6,50 | --- |
| TABLE COMPARE | TCMP | 085 | 4 | 14,0 | 15,2 | 21,9 | 21,9 | --- |
| MULTIPLE COMPARE | MCMP | 019 | 4 | 20,5 | 22,8 | 31,2 | 31,2 | --- |
| UNSIGNED BLOCK COMPARE | BCMP | 068 | 4 | 21,5 | 23,7 | 32,6 | 32,6 | --- |
| AREA RANGE COMPARE | ZCP | 088 | 3 | 5,3 | 5,4 | --- | --- | --- |
| DOUBLE AREA RANGE COMPARE | ZCPL | 116 | 3 | 5,5 | 6,7 | --- | --- | --- |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-1-6 Istruzioni di spostamento dei dati

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|----------------------------------|-----------|--------|--------------------------------|-----------------------------------|---------|--------|--------|-------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| MOVE | MOV | 021 | 3 | 0,18 | 0,20 | 0,25 | 0,29 | --- |
| | !MOV | 021 | 7 | +21,38 | +21,40 | +42,36 | +42,36 | Aumento per la serie CS |
| | | | | +90,52 | +90,52 | +90,52 | +90,52 | Aumento per C200H |
| DOUBLE MOVE | MOVL | 498 | 3 | 0,32 | 0,34 | 0,42 | 0,50 | --- |
| MOVE NOT | MVN | 022 | 3 | 0,18 | 0,20 | 0,25 | 0,29 | --- |
| DOUBLE MOVE NOT | MVNL | 499 | 3 | 0,32 | 0,34 | 0,42 | 0,50 | --- |
| MOVE BIT | MOVB | 082 | 4 | 0,24 | 0,34 | 7,5 | 7,5 | --- |
| MOVE DIGIT | MOVD | 083 | 4 | 0,24 | 0,34 | 7,3 | 7,3 | --- |
| MULTIPLE BIT TRANSFER | XFRB | 062 | 4 | 10,1 | 10,8 | 13,6 | 13,6 | Trasferimento di 1 bit |
| | | | | 186,4 | 189,8 | 269,2 | 269,2 | Trasferimento di 255 bit |
| BLOCK TRANSFER | XFER | 070 | 4 | 0,36 | 0,44 | 11,2 | 11,2 | Trasferimento di 1 canale |
| | | | | 300,1 | 380,1 | 633,5 | 633,5 | Trasferimento di 1.000 canali |
| BLOCK SET | BSET | 071 | 4 | 0,26 | 0,28 | 8,5 | 8,5 | Impostazione di 1 canale |
| | | | | 200,1 | 220,1 | 278,3 | 278,3 | Impostazione di 1.000 canali |
| DATA EXCHANGE | XCHG | 073 | 3 | 0,40 | 0,56 | 0,5 | 0,7 | --- |
| DOUBLE DATA EXCHANGE | XCGL | 562 | 3 | 0,76 | 1,04 | 0,9 | 1,3 | --- |
| SINGLE WORD DISTRIBUTE | DIST | 080 | 4 | 5,1 | 5,4 | 7,0 | 7,0 | --- |
| DATA COLLECT | COLL | 081 | 4 | 5,1 | 5,3 | 7,1 | 7,1 | --- |
| MOVE TO REGISTER | MOVR | 560 | 3 | 0,08 | 0,08 | 0,42 | 0,50 | --- |
| MOVE TIMER/COUNTERPV TO REGISTER | MOVRW | 561 | 3 | 0,42 | 0,50 | 0,42 | 0,50 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-7 Istruzioni di scorrimento dei dati

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-----------------------------------|-----------|--------|--------------------------------|-----------------------------|---------|---------|---------|-----------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SHIFT REGISTER | SFT | 010 | 3 | 7,4 | 10,4 | 10,4 | 10,4 | Scorrimento di 1 canale |
| | | | | 433,2 | 488,0 | 763,1 | 763,1 | Scorrimento di 1.000 canali |
| REVERSIBLE SHIFT REGISTER | SFTR | 084 | 4 | 6,9 | 7,2 | 9,6 | 9,6 | Scorrimento di 1 canale |
| | | | | 615,3 | 680,2 | 859,6 | 859,6 | Scorrimento di 1.000 canali |
| ASYNCHRONOUS SHIFT REGISTER | ASFT | 017 | 4 | 6,2 | 6,4 | 7,7 | 7,7 | Scorrimento di 1 canale |
| | | | | 1,22 ms | 1,22 ms | 2,01 ms | 2,01 ms | Scorrimento di 1.000 canali |
| WORD SHIFT | WSFT | 016 | 4 | 4,5 | 4,7 | 7,8 | 7,8 | Scorrimento di 1 canale |
| | | | | 171,5 | 171,7 | 781,7 | 781,7 | Scorrimento di 1.000 canali |
| ARITHMETIC SHIFT LEFT | ASL | 025 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE SHIFT LEFT | ASLL | 570 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| ARITHMETIC SHIFT RIGHT | ASR | 026 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE SHIFT RIGHT | ASRL | 571 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| ROTATE LEFT | ROL | 027 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE ROTATE LEFT | ROLL | 572 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| ROTATE LEFT WITHOUT CARRY | RLNC | 574 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE ROTATE LEFT WITHOUT CARRY | RLNL | 576 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| ROTATE RIGHT | ROR | 028 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE ROTATE RIGHT | RORL | 573 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| ROTATE RIGHT WITHOUT CARRY | RRNC | 575 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE ROTATE RIGHT WITHOUT CARRY | RRNL | 577 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| ONE DIGIT SHIFT LEFT | SLD | 074 | 3 | 5,9 | 6,1 | 8,2 | 8,2 | Scorrimento di 1 canale |
| | | | | 561,1 | 626,3 | 760,7 | 760,7 | Scorrimento di 1.000 canali |
| ONE DIGIT SHIFT RIGHT | SRD | 075 | 3 | 6,9 | 7,1 | 8,7 | 8,7 | Scorrimento di 1 canale |
| | | | | 760,5 | 895,5 | 1,07 ms | 1,07 ms | Scorrimento di 1.000 canali |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---------------------------------|-----------|--------|---|-----------------------------------|---------|--------|--------|-----------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SHIFT N-BIT DATA LEFT | NSFL | 578 | 4 | 7,5 | 8,3 | 10,5 | 10,5 | Scorrimento di 1 bit |
| | | | | 40,3 | 45,4 | 55,5 | 55,5 | Scorrimento di 1.000 bit |
| SHIFT N-BIT DATA RIGHT | NSFR | 579 | 4 | 7,5 | 8,3 | 10,5 | 10,5 | Scorrimento di 1 bit |
| | | | | 50,5 | 55,3 | 69,3 | 69,3 | Scorrimento di 1.000 bit |
| SHIFT N-BITS LEFT | NASL | 580 | 3 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE SHIFT N-BITS LEFT | NSLL | 582 | 3 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| SHIFT N-BITS RIGHT | NASR | 581 | 3 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE SHIFT N-BITS RIGHT | NSRL | 583 | 3 | 0,40 | 0,56 | 0,50 | 0,67 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-8 Istruzioni di incremento e decremento

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|-------------------------------|-----------|--------|---|-----------------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| INCREMENT BINARY | ++ | 590 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE INCREMENT BINARY | ++L | 591 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| DECREMENT BINARY | -- | 592 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE DECREMENT BINARY | --L | 593 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |
| INCREMENT BCD | ++B | 594 | 2 | 6,4 | 4,5 | 7,4 | 7,4 | --- |
| DOUBLE INCREMENT BCD | ++BL | 595 | 2 | 5,6 | 4,9 | 6,1 | 6,1 | --- |
| DECREMENT BCD | --B | 596 | 2 | 6,3 | 4,6 | 7,2 | 7,2 | --- |
| DOUBLE DECREMENT BCD | --BL | 597 | 2 | 5,3 | 4,7 | 7,1 | 7,1 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-9 Istruzioni matematiche con simboli

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SIGNED BINARY ADD WITHOUT CARRY | + | 400 | 4 | 0,18 | 0,20 | 0,25 | 0,37 | --- |
| DOUBLE SIGNED BINARY ADD WITHOUT CARRY | +L | 401 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |
| SIGNED BINARY ADD WITH CARRY | +C | 402 | 4 | 0,18 | 0,20 | 0,25 | 0,37 | --- |
| DOUBLE SIGNED BINARY ADD WITH CARRY | +CL | 403 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |
| BCD ADD WITHOUT CARRY | +B | 404 | 4 | 8,2 | 8,4 | 14,0 | 14,0 | --- |
| DOUBLE BCD ADD WITHOUT CARRY | +BL | 405 | 4 | 13,3 | 14,5 | 19,0 | 19,0 | --- |
| BCD ADD WITH CARRY | +BC | 406 | 4 | 8,9 | 9,1 | 14,5 | 14,5 | --- |
| DOUBLE BCD ADD WITH CARRY | +BCL | 407 | 4 | 13,8 | 15,0 | 19,6 | 19,6 | --- |
| SIGNED BINARY SUBTRACT WITHOUT CARRY | - | 410 | 4 | 0,18 | 0,20 | 0,25 | 0,37 | --- |
| DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY | -L | 411 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |
| SIGNED BINARY SUBTRACT WITH CARRY | -C | 412 | 4 | 0,18 | 0,20 | 0,25 | 0,37 | --- |
| DOUBLE SIGNED BINARY SUBTRACT WITH CARRY | -CL | 413 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|---|-----------|--------|---|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| BCD SUBTRACT WITHOUT CARRY | -B | 414 | 4 | 8,0 | 8,2 | 13,1 | 13,1 | --- |
| DOUBLE BCD SUBTRACT WITHOUT CARRY | -BL | 415 | 4 | 12,8 | 14,0 | 18,2 | 18,2 | --- |
| BCD SUBTRACT WITH CARRY | -BC | 416 | 4 | 8,5 | 8,6 | 13,8 | 13,8 | --- |
| DOUBLE BCD SUBTRACT WITH CARRY | -BCL | 417 | 4 | 13,4 | 14,7 | 18,8 | 18,8 | --- |
| SIGNED BINARY MULTIPLY | * | 420 | 4 | 0,38 | 0,40 | 0,50 | 0,58 | --- |
| DOUBLE SIGNED BINARY MULTIPLY | *L | 421 | 4 | 7,23 | 8,45 | 11,19 | 11,19 | --- |
| UNSIGNED BINARY MULTIPLY | *U | 422 | 4 | 0,38 | 0,40 | 0,50 | 0,58 | --- |
| DOUBLE UNSIGNED BINARY MULTIPLY | *UL | 423 | 4 | 7,1 | 8,3 | 10,63 | 10,63 | --- |
| BCD MULTIPLY | *B | 424 | 4 | 9,0 | 9,2 | 12,8 | 12,8 | --- |
| DOUBLE BCD MULTIPLY | *BL | 425 | 4 | 23,0 | 24,2 | 35,2 | 35,2 | --- |
| SIGNED BINARY DIVIDE | / | 430 | 4 | 0,40 | 0,42 | 0,75 | 0,83 | --- |
| DOUBLE SIGNED BINARY DIVIDE | /L | 431 | 4 | 7,2 | 8,4 | 9,8 | 9,8 | --- |
| UNSIGNED BINARY DIVIDE | /U | 432 | 4 | 0,40 | 0,42 | 0,75 | 0,83 | --- |
| DOUBLE UNSIGNED BINARY DIVIDE | /UL | 433 | 4 | 6,9 | 8,1 | 9,1 | 9,1 | --- |
| BCD DIVIDE | /B | 434 | 4 | 8,6 | 8,8 | 15,9 | 15,9 | --- |
| DOUBLE BCD DIVIDE | /BL | 435 | 4 | 17,7 | 18,9 | 26,2 | 26,2 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-10 Istruzioni di conversione

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|--------------------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| BCD-TO-BINARY | BIN | 023 | 3 | 0,22 | 0,24 | 0,25 | 0,29 | --- |
| DOUBLE BCD-TO-DOUBLE BINARY | BINL | 058 | 3 | 6,5 | 6,8 | 9,1 | 9,1 | --- |
| BINARY-TO-BCD | BCD | 024 | 3 | 0,24 | 0,26 | 8,3 | 8,3 | --- |
| DOUBLE BINARY-TO-DOUBLE BCD | BCDL | 059 | 3 | 6,7 | 7,0 | 9,2 | 9,2 | --- |
| 2'S COMPLEMENT | NEG | 160 | 3 | 0,18 | 0,20 | 0,25 | 0,29 | --- |
| DOUBLE 2'S COMPLEMENT | NEGL | 161 | 3 | 0,32 | 0,34 | 0,42 | 0,5 | --- |
| 16-BIT TO 32-BIT SIGNED BINARY | SIGN | 600 | 3 | 0,32 | 0,34 | 0,42 | 0,50 | --- |
| DATA DECODER | MLPX | 076 | 4 | 0,32 | 0,42 | 8,8 | 8,8 | Decodifica di 1 cifra (da 4 a 16) |
| | | | | 0,98 | 1,20 | 12,8 | 12,8 | Decodifica di 4 cifre (da 4 a 16) |
| | | | | 3,30 | 4,00 | 20,3 | 20,3 | Decodifica di 1 cifra (da 8 a 256) |
| | | | | 6,50 | 7,90 | 33,4 | 33,4 | Decodifica di 2 cifre (da 8 a 256) |
| DATA ENCODER | DMPX | 077 | 4 | 7,5 | 7,9 | 10,4 | 10,4 | Codifica di 1 cifra (da 16 a 4) |
| | | | | 49,6 | 50,2 | 59,1 | 59,1 | Codifica di 4 cifre (da 16 a 4) |
| | | | | 18,2 | 18,6 | 23,6 | 23,6 | Codifica di 1 cifra (da 256 a 8) |
| | | | | 55,1 | 57,4 | 92,5 | 92,5 | Codifica di 2 cifre (da 256 a 8) |
| ASCII CONVERT | ASC | 086 | 4 | 6,8 | 7,1 | 9,7 | 9,7 | Conversione di 1 cifra in ASCII |
| | | | | 11,2 | 11,7 | 15,1 | 15,1 | Conversione di 4 cifre in ASCII |
| ASCII TO HEX | HEX | 162 | 4 | 7,1 | 7,4 | 10,1 | 10,1 | Conversione di 1 cifra |
| COLUMN TO LINE | LINE | 063 | 4 | 19,0 | 23,1 | 29,1 | 29,1 | --- |
| LINE TO COLUMN | COLM | 064 | 4 | 23,2 | 27,5 | 37,3 | 37,3 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SIGNED BCD-TO-BINARY | BINS | 470 | 4 | 8,0 | 8,3 | 12,1 | 12,1 | Impostazione del formato dati n. 0 |
| | | | | 8,0 | 8,3 | 12,1 | 12,1 | Impostazione del formato dati n. 1 |
| | | | | 8,3 | 8,6 | 12,7 | 12,7 | Impostazione del formato dati n. 2 |
| | | | | 8,5 | 8,8 | 13,0 | 13,0 | Impostazione del formato dati n. 3 |
| DOUBLE SIGNED BCD-TO-BINARY | BISL | 472 | 4 | 9,2 | 9,6 | 13,6 | 13,6 | Impostazione del formato dati n. 0 |
| | | | | 9,2 | 9,6 | 13,7 | 13,7 | Impostazione del formato dati n. 1 |
| | | | | 9,5 | 9,9 | 14,2 | 14,2 | Impostazione del formato dati n. 2 |
| | | | | 9,6 | 10,0 | 14,4 | 14,4 | Impostazione del formato dati n. 3 |
| SIGNED BINARY-TO-BCD | BCDS | 471 | 4 | 6,6 | 6,9 | 10,6 | 10,6 | Impostazione del formato dati n. 0 |
| | | | | 6,7 | 7,0 | 10,8 | 10,8 | Impostazione del formato dati n. 1 |
| | | | | 6,8 | 7,1 | 10,9 | 10,9 | Impostazione del formato dati n. 2 |
| | | | | 7,2 | 7,5 | 11,5 | 11,5 | Impostazione del formato dati n. 3 |
| DOUBLE SIGNED BINARY-TO-BCD | BDSL | 473 | 4 | 8,1 | 8,4 | 11,6 | 11,6 | Impostazione del formato dati n. 0 |
| | | | | 8,2 | 8,6 | 11,8 | 11,8 | Impostazione del formato dati n. 1 |
| | | | | 8,3 | 8,7 | 12,0 | 12,0 | Impostazione del formato dati n. 2 |
| | | | | 8,8 | 9,2 | 12,5 | 12,5 | Impostazione del formato dati n. 3 |
| GRAY CODE CONVERSION (vedere la nota 2) | GRY | 474 | 4 | 46,9 | 72,1 | --- | --- | Binario a 8 bit |
| | | | | 49,6 | 75,2 | --- | --- | BCD a 8 bit |
| | | | | 57,7 | 87,7 | --- | --- | Angolo a 8 bit |
| | | | | 61,8 | 96,7 | --- | --- | Binario a 15 bit |
| | | | | 64,5 | 99,6 | --- | --- | BCD a 15 bit |
| | | | | 72,8 | 112,4 | --- | --- | Angolo a 15 bit |
| | | | | 52,3 | 87,2 | --- | --- | 360° binario |
| | | | | 55,1 | 90,4 | --- | --- | 360° BCD |
| 64,8 | 98,5 | --- | --- | 360° angolo | | | | |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-1-11 Istruzioni logiche

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|----------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| LOGICAL AND | ANDW | 034 | 4 | 0,18 | 0,20 | 0,25 | 0,37 | --- |
| DOUBLE LOGICAL AND | ANDL | 610 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |
| LOGICAL OR | ORW | 035 | 4 | 0,22 | 0,32 | 0,25 | 0,37 | --- |
| DOUBLE LOGICAL OR | ORWL | 611 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |
| EXCLUSIVE OR | XORW | 036 | 4 | 0,22 | 0,32 | 0,25 | 0,37 | --- |
| DOUBLE EXCLUSIVE OR | XORL | 612 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |
| EXCLUSIVE NOR | XNRW | 037 | 4 | 0,22 | 0,32 | 0,25 | 0,37 | --- |
| DOUBLE EXCLUSIVE NOR | XNRL | 613 | 4 | 0,32 | 0,34 | 0,42 | 0,54 | --- |
| COMPLEMENT | COM | 029 | 2 | 0,22 | 0,32 | 0,29 | 0,37 | --- |
| DOUBLE COMPLEMENT | COML | 614 | 2 | 0,40 | 0,56 | 0,50 | 0,67 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-12 Istruzioni matematiche speciali

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-----------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|--|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| BINARY ROOT | ROTB | 620 | 3 | 49,6 | 50,0 | 530,7 | 530,7 | --- |
| BCD SQUARE ROOT | ROOT | 072 | 3 | 13,7 | 13,9 | 514,5 | 514,5 | --- |
| ARITHMETIC PROCESS | APR | 069 | 4 | 6,7 | 6,9 | 32,3 | 32,3 | Designazione di SIN e COS |
| | | | | 17,2 | 18,4 | 78,3 | 78,3 | Designazione dell'approssimazione lineare a segmenti |
| FLOATING POINT DIVIDE | FDIV | 079 | 4 | 116,6 | 176,6 | 176,6 | 176,6 | --- |
| BIT COUNTER | BCNT | 067 | 4 | 0,3 | 0,38 | 22,1 | 22,1 | Conteggio di 1 canale |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-13 Istruzioni matematiche a virgola mobile

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---------------------------------------|------------------|--------|--------------------------------|-----------------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| FLOATING TO 16-BIT | FIX | 450 | 3 | 10,6 | 10,8 | 14,5 | 14,5 | --- |
| FLOATING TO 32-BIT | FIXL | 451 | 3 | 10,8 | 11,0 | 14,6 | 14,6 | --- |
| 16-BIT TO FLOATING | FLT | 452 | 3 | 8,3 | 8,5 | 11,1 | 11,1 | --- |
| 32-BIT TO FLOATING | FLTL | 453 | 3 | 8,3 | 8,5 | 10,8 | 10,8 | --- |
| FLOATING-POINT ADD | +F | 454 | 4 | 8,0 | 9,2 | 10,2 | 10,2 | --- |
| FLOATING-POINT SUBTRACT | -F | 455 | 4 | 8,0 | 9,2 | 10,3 | 10,3 | --- |
| FLOATING POINT DIVIDE | /F | 457 | 4 | 8,7 | 9,9 | 12,0 | 12,0 | --- |
| FLOATING-POINT MULTIPLY | *F | 456 | 4 | 8,0 | 9,2 | 10,5 | 10,5 | --- |
| DEGREES TO RADIANS | RAD | 458 | 3 | 10,1 | 10,2 | 14,9 | 14,9 | --- |
| RADIANS TO DEGREES | DEG | 459 | 3 | 9,9 | 10,1 | 14,8 | 14,8 | --- |
| SINE | SIN | 460 | 3 | 42,0 | 42,2 | 61,1 | 61,1 | --- |
| COSINE | COS | 461 | 3 | 31,5 | 31,8 | 44,1 | 44,1 | --- |
| TANGENT | TAN | 462 | 3 | 16,3 | 16,6 | 22,6 | 22,6 | --- |
| ARC SINE | ASIN | 463 | 3 | 17,6 | 17,9 | 24,1 | 24,1 | --- |
| ARC COSINE | ACOS | 464 | 3 | 20,4 | 20,7 | 28,0 | 28,0 | --- |
| ARC TANGENT | ATAN | 465 | 3 | 16,1 | 16,4 | 16,4 | 16,4 | --- |
| SQUARE ROOT | SQRT | 466 | 3 | 19,0 | 19,3 | 28,1 | 28,1 | --- |
| EXPONENT | EXP | 467 | 3 | 65,9 | 66,2 | 96,7 | 96,7 | --- |
| LOGARITHM | LOG | 468 | 3 | 12,8 | 13,1 | 17,4 | 17,4 | --- |
| EXPONENTIAL POWER | PWR | 840 | 4 | 125,4 | 126,0 | 181,7 | 181,7 | --- |
| Confronto di simboli a virgola mobile | LD, AND, OR +=F | 329 | 3 | 6,6 | 8,3 | --- | --- | --- |
| | LD, AND, OR +<>F | 330 | | | | | | |
| | LD, AND, OR +<F | 331 | | | | | | |
| | LD, AND, OR +<=F | 332 | | | | | | |
| | LD, AND, OR +>F | 333 | | | | | | |
| | LD, AND, OR +>=F | 334 | | | | | | |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-------------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| FLOATING-POINT TO ASCII | FSTR | 448 | 4 | 48,5 | 48,9 | --- | --- | --- |
| ASCII TO FLOATING-POINT | FVAL | 449 | 3 | 21,1 | 21,3 | --- | --- | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-14 Istruzioni a virgola mobile in doppia precisione

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|----------------------------------|------------------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| DOUBLE SYMBOL COMPARISON | LD, AND, OR +=D | 335 | 3 | 8,5 | 10,3 | --- | --- | --- |
| | LD, AND, OR +<>D | 336 | | | | | | |
| | LD, AND, OR +<D | 337 | | | | | | |
| | LD, AND, OR +<=D | 338 | | | | | | |
| | LD, AND, OR +>D | 339 | | | | | | |
| | LD, AND, OR +>=D | 340 | | | | | | |
| DOUBLE FLOATING TO 16-BIT BINARY | FIXD | 841 | 3 | 11,7 | 12,1 | --- | --- | --- |
| DOUBLE FLOATING TO 32-BIT BINARY | FIXLD | 842 | 3 | 11,6 | 12,1 | --- | --- | --- |
| 16-BIT BINARY TO DOUBLE FLOATING | DBL | 843 | 3 | 9,9 | 10,0 | --- | --- | --- |
| 32-BIT BINARY TO DOUBLE FLOATING | DBLL | 844 | 3 | 9,8 | 10,0 | --- | --- | --- |
| DOUBLE FLOATING-POINT ADD | +D | 845 | 4 | 11,2 | 11,9 | --- | --- | --- |
| DOUBLE FLOATING-POINT SUBTRACT | -D | 846 | 4 | 11,2 | 11,9 | --- | --- | --- |
| DOUBLE FLOATING-POINT MULTIPLY | *D | 847 | 4 | 12,0 | 12,7 | --- | --- | --- |
| DOUBLE FLOATING-POINT DIVIDE | /D | 848 | 4 | 23,5 | 24,2 | --- | --- | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|---------------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| DOUBLE DEGREES TO RADIANS | RADD | 849 | 3 | 27,4 | 27,8 | --- | --- | --- |
| DOUBLE RADIANS TO DEGREES | DEGD | 850 | 3 | 11,2 | 11,9 | --- | --- | --- |
| DOUBLE SINE | SIND | 851 | 3 | 45,4 | 45,8 | --- | --- | --- |
| DOUBLE COSINE | COSD | 852 | 3 | 43,0 | 43,4 | --- | --- | --- |
| DOUBLE TANGENT | TAND | 853 | 3 | 20,1 | 20,5 | --- | --- | --- |
| DOUBLE ARC SINE | ASIND | 854 | 3 | 21,5 | 21,9 | --- | --- | --- |
| DOUBLE ARC COSINE | ACOSD | 855 | 3 | 24,7 | 25,1 | --- | --- | --- |
| DOUBLE ARC TANGENT | ATAND | 856 | 3 | 19,3 | 19,7 | --- | --- | --- |
| DOUBLE SQUARE ROOT | SQRTD | 857 | 3 | 47,4 | 47,9 | --- | --- | --- |
| DOUBLE EXPONENT | EXPD | 858 | 3 | 121,0 | 121,4 | --- | --- | --- |
| DOUBLE LOGARITHM | LOGD | 859 | 3 | 16,0 | 16,4 | --- | --- | --- |
| DOUBLE EXPONENTIAL POWER | PWRD | 860 | 4 | 223,9 | 224,2 | --- | --- | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-15 Istruzioni di elaborazione dei dati delle tabelle

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|------------------------|-----------|--------|--------------------------------|-----------------------------|---------|---------|---------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SET STACK | SSET | 630 | 3 | 8,0 | 8,3 | 8,5 | 8,5 | Designazione di 5 canali nell'area di stack |
| | | | | 231,6 | 251,8 | 276,8 | 276,8 | Designazione di 1.000 canali nell'area di stack |
| PUSH ONTO STACK | PUSH | 632 | 3 | 6,5 | 8,6 | 9,1 | 9,1 | --- |
| FIRST IN FIRST OUT | FIFO | 633 | 3 | 6,9 | 8,9 | 10,6 | 10,6 | Designazione di 5 canali nell'area di stack |
| | | | | 352,6 | 434,3 | 1,13 ms | 1,13 ms | Designazione di 1.000 canali nell'area di stack |
| LAST IN FIRST OUT | LIFO | 634 | 3 | 7,0 | 9,0 | 9,9 | 9,9 | --- |
| DIMENSION RECORD TABLE | DIM | 631 | 5 | 15,2 | 21,6 | 142,1 | 142,1 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|----------------------|-----------|--------|--------------------------------|-----------------------------------|---------|---------|---------|---------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SET RECORD LOCATION | SETR | 635 | 4 | 5,4 | 5,9 | 7,0 | 7,0 | --- |
| GET RECORD NUMBER | GETR | 636 | 4 | 7,8 | 8,4 | 11,0 | 11,0 | --- |
| DATA SEARCH | SRCH | 181 | 4 | 15,5 | 19,5 | 19,5 | 19,5 | Ricerca di 1 canale |
| | | | | 2,42 ms | 3,34 ms | 3,34 ms | 3,34 ms | Ricerca di 1.000 canali |
| SWAP BYTES | SWAP | 637 | 3 | 12,2 | 13,6 | 13,6 | 13,6 | Scambio di 1 canale |
| | | | | 1,94 ms | 2,82 ms | 2,82 ms | 2,82 ms | Scambio di 1.000 canali |
| FIND MAXIMUM | MAX | 182 | 4 | 19,2 | 24,9 | 24,9 | 24,9 | Ricerca di 1 canale |
| | | | | 2,39 ms | 3,36 ms | 3,36 ms | 3,36 ms | Ricerca di 1.000 canali |
| FIND MINIMUM | MIN | 183 | 4 | 19,2 | 25,3 | 25,3 | 25,3 | Ricerca di 1 canale |
| | | | | 2,39 ms | 3,33 ms | 3,33 ms | 3,33 ms | Ricerca di 1.000 canali |
| SUM | SUM | 184 | 4 | 28,2 | 38,5 | 38,5 | 38,3 | Aggiunta di 1 canale |
| | | | | 1,42 ms | 1,95 ms | 1,95 ms | 1,95 ms | Aggiunta di 1.000 canali |
| FRAME CHECKSUM | FCS | 180 | 4 | 20,0 | 28,3 | 28,3 | 28,3 | Per una tabella di 1 canali |
| | | | | 1,65 ms | 2,48 ms | 2,48 ms | 2,48 ms | Per una tabella di 1.000 canali |
| STACK SIZE READ | SNUM | 638 | 3 | 6,0 | 6,3 | --- | --- | --- |
| STACK DATA READ | SREAD | 639 | 4 | 8,0 | 8,4 | --- | --- | --- |
| STACK DATA OVERWRITE | SWRIT | 640 | 4 | 7,2 | 7,6 | --- | --- | --- |
| STACK DATA INSERT | SINS | 641 | 4 | 7,8 | 9,9 | --- | --- | --- |
| | | | | 354,0 | 434,8 | --- | --- | Per una tabella di 1.000 canali |
| STACK DATA DELETE | SDEL | 642 | 4 | 8,6 | 10,6 | --- | --- | --- |
| | | | | 354,0 | 436,0 | --- | --- | Per una tabella di 1.000 canali |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-16 Istruzioni di controllo dei dati

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------------|---------|--------|--------|--|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| PID CONTROL | PID | 190 | 4 | 436,2 | 678,2 | 678,2 | 678,2 | Esecuzione iniziale |
| | | | | 332,3 | 474,9 | 474,9 | 474,9 | Campionamento |
| | | | | 97,3 | 141,3 | 141,3 | 141,3 | Nessun campionamento |
| LIMIT CONTROL | LMT | 680 | 4 | 16,1 | 22,1 | 22,1 | 22,1 | --- |
| DEAD BAND CONTROL | BAND | 681 | 4 | 17,0 | 22,5 | 22,5 | 22,5 | --- |
| DEAD ZONE CONTROL | ZONE | 682 | 4 | 15,4 | 20,5 | 20,5 | 20,5 | --- |
| TIME-PROPORTIONAL OUTPUT (vedere la nota 2) | TPO | 685 | 4 | 10,4 | 14,8 | --- | --- | Tempo di esecuzione OFF |
| | | | | 54,5 | 82,0 | --- | --- | Tempo di esecuzione ON con designazione del duty-cycle o limite di uscita visualizzato |
| | | | | 61,0 | 91,9 | --- | --- | Tempo di esecuzione ON con designazione variabile manipolata o limite di uscita attivato |
| SCALING | SCL | 194 | 4 | 37,1 | 53,0 | 56,8 | 56,8 | --- |
| SCALING 2 | SCL2 | 486 | 4 | 28,5 | 40,2 | 50,7 | 50,7 | --- |
| SCALING 3 | SCL3 | 487 | 4 | 33,4 | 47,0 | 57,7 | 57,7 | --- |
| AVERAGE | AVG | 195 | 4 | 36,3 | 52,6 | 53,1 | 53,1 | Media di un'operazione |
| | | | | 291,0 | 419,9 | 419,9 | 419,9 | Media di 64 operazioni |
| PID CONTROL WITH AUTOTUNING | PIDAT | 191 | 4 | 446,3 | 712,5 | --- | --- | Esecuzione iniziale |
| | | | | 339,4 | 533,9 | --- | --- | Campionamento |
| | | | | 100,7 | 147,1 | --- | --- | Nessun campionamento |
| | | | | 189,2 | 281,6 | --- | --- | Esecuzione iniziale dell'autotuning |
| | | | | 535,2 | 709,8 | --- | --- | Autotuning durante il campionamento |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-1-17 Istruzioni di subroutine

| Istruzione | Mnemonic o | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|--------------------------|---------------|--------|---|-----------------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SUBROUTINE CALL | SBS | 091 | 2 | 1,26 | 1,96 | 17,0 | 17,0 | --- |
| SUBROUTINE ENTRY | SBN | 092 | 2 | --- | --- | --- | --- | --- |
| SUBROUTINE RETURN | RET | 093 | 1 | 0,86 | 1,60 | 20,60 | 20,60 | --- |
| MACRO | MCRO | 099 | 4 | 23,3 | 23,3 | 23,3 | 23,3 | --- |
| GLOBAL SUBROUTINE CALL | GSBN | 751 | 2 | --- | --- | --- | --- | --- |
| GLOBAL SUBROUTINE ENTRY | GRET | 752 | 1 | 1,26 | 1,96 | --- | --- | --- |
| GLOBAL SUBROUTINE RETURN | GSBS | 750 | 2 | 0,86 | 1,60 | --- | --- | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-18 Istruzioni di controllo degli interrupt

| Istruzione | Mnemonic | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---------------------|----------|--------|---|-----------------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SET INTERRUPT MASK | MSKS | 690 | 3 | 25,6 | 38,4 | 39,5 | 39,5 | --- |
| READ INTERRUPT MASK | MSKR | 692 | 3 | 11,9 | 11,9 | 11,9 | 11,9 | --- |
| CLEAR INTERRUPT | CLI | 691 | 3 | 27,4 | 41,3 | 41,3 | 41,3 | --- |
| DISABLE INTERRUPTS | DI | 693 | 1 | 15,0 | 16,8 | 16,8 | 16,8 | --- |
| ENABLE INTERRUPTS | EI | 694 | 1 | 19,5 | 21,8 | 21,8 | 21,8 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-19 Istruzioni di step

| Istruzione | Mnemonic | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|-------------|----------|--------|---|-----------------------------------|---------|--------|--------|--|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| STEP DEFINE | STEP | 008 | 2 | 17,4 | 20,7 | 27,1 | 27,1 | Bit di controllo degli step impostato su ON |
| | | | | 11,8 | 13,7 | 24,4 | 24,4 | Bit di controllo degli step impostato su OFF |
| STEP START | SNXT | 009 | 2 | 6,6 | 7,3 | 10,0 | 10,0 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-20 Istruzioni per moduli di I/O di base

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|--|-----------|--------|---|-----------------------------|---------|--------|--------|--|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| I/O REFRESH | IORF | 097 | 3 | 58,5 | 63,2 | 81,7 | 81,7 | Aggiornamento di 1 canale (IN) per i moduli di I/O di base C200H |
| | | | | 62,6 | 67,0 | 86,7 | 86,7 | Aggiornamento di 1 canale (OUT) per i moduli di I/O di base C200H |
| | | | | 15,5 | 16,4 | 23,5 | 23,5 | Aggiornamento di 1 canale (IN) per i moduli di I/O di base CS |
| | | | | 17,20 | 18,40 | 25,6 | 25,6 | Aggiornamento di 1 canale (OUT) per i moduli di I/O di base della serie CS |
| | | | | 303,3 | 343,9 | 357,1 | 357,1 | Aggiornamento di 10 canali (IN) per i moduli di I/O di base C200H |
| | | | | 348,2 | 376,6 | 407,5 | 407,5 | Aggiornamento di 10 canali (OUT) per i moduli di I/O di base C200H |
| | | | | 319,9 | 320,7 | 377,5 | 377,6 | Aggiornamento di 60 canali (IN) per i moduli di I/O di base CS |
| | | | | 358,00 | 354,40 | 460,1 | 460,1 | Aggiornamento di 60 canali (OUT) per i moduli di I/O di base CS |
| 7-SEGMENT DECODER | SDEC | 078 | 4 | 6,5 | 6,9 | 14,1 | 14,1 | --- |
| DIGITAL SWITCH INPUT (vedere la nota 2) | DSW | 210 | 6 | 50,7 | 73,5 | --- | --- | A 4 cifre, valore di ingresso dei dati: 0 |
| | | | | 51,5 | 73,4 | --- | --- | A 4 cifre, valore di ingresso dei dati: F |
| | | | | 51,3 | 73,5 | --- | --- | A 8 cifre, valore di ingresso dei dati: 0 |
| | | | | 50,7 | 73,4 | --- | --- | A 8 cifre, valore di ingresso dei dati: F |
| TEN KEY INPUT (vedere la nota 2) | TKY | 211 | 4 | 9,7 | 13,2 | --- | --- | Valore di ingresso dei dati: 0 |
| | | | | 10,7 | 14,8 | --- | --- | Valore di ingresso dei dati: F |
| HEXADECIMAL KEY INPUT (vedere la nota 2) | HKY | 212 | 5 | 50,3 | 70,9 | --- | --- | Valore di ingresso dei dati: 0 |
| | | | | 50,1 | 71,2 | --- | --- | Valore di ingresso dei dati: F |
| MATRIX INPUT (vedere la nota 2) | MTR | 213 | 5 | 47,8 | 68,1 | --- | --- | Valore di ingresso dei dati: 0 |
| | | | | 48,0 | 68,0 | --- | --- | Valore di ingresso dei dati: F |
| 7-SEGMENT DISPLAY OUTPUT (vedere la nota 2) | 7SEG | 214 | 5 | 58,1 | 83,3 | --- | --- | 4 cifre |
| | | | | 63,3 | 90,3 | --- | --- | 8 cifre |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|-----------------------|-----------|--------|--------------------------------|--|---------|--------|--------|-------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| INTELLIGENT I/O READ | IORD | 222 | 4 | I tempi di lettura e scrittura dipendono dal modulo di I/O speciale per cui viene eseguita l'istruzione. | | | | --- |
| INTELLIGENT I/O WRITE | IOWR | 223 | 4 | | | | | --- |
| CPU BUS I/O REFRESH | DLNK | 226 | 4 | 287,8 | 315,5 | --- | --- | Allocazione di 1 canale |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-1-21 Istruzioni per la comunicazione seriale

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------|---------|---------|---------|--|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| PROTOCOL MACRO | PMCR | 260 | 5 | 100,1 | 142,1 | 276,8 | 276,8 | Invio di 0 canali, ricezione di 0 canali |
| | | | | 134,2 | 189,6 | 305,9 | 305,9 | Invio di 249 canali, ricezione di 249 canali |
| TRANSMIT | TXD | 236 | 4 | 68,5 | 98,8 | 98,8 | 98,8 | Invio di 1 byte |
| | | | | 734,3 | 1,10 ms | 1,10 ms | 1,10 ms | Invio di 256 byte |
| RECEIVE | RXD | 235 | 4 | 89,6 | 131,1 | 131,1 | 131,1 | Memorizzazione di 1 byte |
| | | | | 724,2 | 1,11 ms | 1,11 ms | 1,11 ms | Memorizzazione di 256 byte |
| TRANSMIT VIA SERIAL COMMUNICATIONS UNIT | TXDU | 256 | 4 | 131,5 | 202,4 | --- | --- | Invio di 1 byte |
| RECEIVE VIA SERIAL COMMUNICATIONS UNIT | RXDU | 255 | 4 | 131 | 200,8 | --- | --- | Memorizzazione di 1 byte |
| CHANGE SERIAL PORT SETUP | STUP | 237 | 3 | 341,2 | 400,0 | 440,4 | 440,4 | --- |

- Nota** Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-22 Istruzioni di rete

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|-----------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| NETWORK SEND | SEND | 090 | 4 | 84,4 | 123,9 | 123,9 | 123,9 | --- |
| NETWORK RECEIVE | RECV | 098 | 4 | 85,4 | 124,7 | 124,7 | 124,7 | --- |
| DELIVER COMMAND | CMND | 490 | 4 | 106,8 | 136,8 | 136,8 | 136,8 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| EXPLICIT MESSAGE SEND (vedere la nota 2) | EXPLT | 720 | 4 | 127,6 | 190,0 | --- | --- | --- |
| EXPLICIT GET ATTRIBUTE (vedere la nota 2) | EGATR | 721 | 4 | 123,9 | 185,0 | --- | --- | --- |
| EXPLICIT SET ATTRIBUTE (vedere la nota 2) | ESATR | 722 | 3 | 110,0 | 164,4 | --- | --- | --- |
| EXPLICIT WORD READ (vedere la nota 2) | ECHRD | 723 | 4 | 106,8 | 158,9 | --- | --- | --- |
| EXPLICIT WORD WRITE (vedere la nota 2) | ECHWR | 724 | 4 | 106,0 | 158,3 | --- | --- | --- |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-1-23 Istruzioni per la memoria dei file

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-----------------|-----------|--------|--------------------------------|-----------------------------|---------|---------|---------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| READ DATA FILE | FREAD | 700 | 5 | 391,4 | 632,4 | 684,1 | 684,1 | Directory di 2 caratteri + nome di file in formato binario |
| | | | | 836,1 | 1,33 ms | 1,35 ms | 1,35 ms | Directory di 73 caratteri + nome di file in formato binario |
| WRITE DATA FILE | FWRIT | 701 | 5 | 387,8 | 627,0 | 684,7 | 684,7 | Directory di 2 caratteri + nome di file in formato binario |
| | | | | 833,3 | 1,32 ms | 1,36 ms | 1,36 ms | Directory di 73 caratteri + nome di file in formato binario |

- Nota** Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-24 Istruzioni di visualizzazione

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-----------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| DISPLAY MESSAGE | MSG | 046 | 3 | 10,1 | 14,2 | 14,3 | 14,3 | Visualizzazione del messaggio |
| | | | | 8,4 | 11,3 | 11,3 | 11,3 | Eliminazione del messaggio visualizzato |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-25 Istruzioni per l'orologio

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| CALENDAR ADD | CADD | 730 | 4 | 38,3 | 201,9 | 209,5 | 209,5 | --- |
| CALENDAR SUBTRACT | CSUB | 731 | 4 | 38,6 | 170,4 | 184,1 | 184,1 | --- |
| HOURS TO SECONDS | SEC | 065 | 3 | 21,4 | 29,3 | 35,8 | 35,8 | --- |
| SECONDS TO HOURS | HMS | 066 | 3 | 22,2 | 30,9 | 42,1 | 42,1 | --- |
| CLOCK ADJUSTMENT | DATE | 735 | 2 | 60,5 | 87,4 | 95,9 | 95,9 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-26 Istruzioni di debug

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|----------------------------|-----------|--------|--------------------------------|-----------------------------|---------|---------|---------|------------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| Campionamento Trace Memory | TRSM | 045 | 1 | 80,4 | 120,0 | 120,0 | 120,0 | Campionamento di 1 bit e 0 canali |
| | | | | 848,1 | 1,06 ms | 1,06 ms | 1,06 ms | Campionamento di 31 bit e 6 canali |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-27 Istruzioni di diagnostica per malfunzionamento

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-------------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| FAILURE ALARM | FAL | 006 | 3 | 15,4 | 16,7 | 16,7 | 16,7 | Registrazione degli errori |
| | | | | 179,8 | 244,8 | 244,8 | 244,8 | Eliminazione degli errori in ordine di priorità |
| | | | | 432,4 | 657,1 | 657,1 | 657,1 | Eliminazione di tutti gli errori |
| | | | | 161,5 | 219,4 | 219,4 | 219,4 | Eliminazione di singoli errori |
| SEVERE FAILURE ALARM | FALS | 007 | 3 | --- | --- | --- | --- | --- |
| FAILURE POINT DETECTION | FPD | 269 | 4 | 140,9 | 202,3 | 202,3 | 202,3 | All'esecuzione |
| | | | | 163,4 | 217,6 | 217,6 | 217,6 | La prima volta |
| | | | | 185,2 | 268,9 | 268,9 | 268,9 | All'esecuzione |
| | | | | 207,5 | 283,6 | 283,6 | 283,6 | La prima volta |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-28 Altre istruzioni

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|------------------------------|-----------|--------|--------------------------------|-----------------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| SET CARRY | STC | 040 | 1 | 0,06 | 0,06 | 0,12 | 0,12 | --- |
| CLEAR CARRY | CLC | 041 | 1 | 0,06 | 0,06 | 0,12 | 0,12 | --- |
| SELECT EM BANK | EMBC | 281 | 2 | 14,0 | 15,1 | 15,1 | 15,1 | --- |
| EXTEND MAXIMUM CYCLE TIME | WDT | 094 | 2 | 15,0 | 19,7 | 19,7 | 19,7 | --- |
| SAVE CONDITION FLAGS | CCS | 282 | 1 | 8,6 | 12,5 | --- | --- | --- |
| LOAD CONDITION FLAGS | CCL | 283 | 1 | 9,8 | 13,9 | --- | --- | --- |
| CONVERT ADDRESS FROM CV | FRMCV | 284 | 3 | 13,6 | 19,9 | --- | --- | --- |
| CONVERT ADDRESS TO CV | TOCV | 285 | 3 | 11,9 | 17,2 | --- | --- | --- |
| DISABLE PERIPHERAL SERVICING | IOSP | 287 | --- | 13,9 | 19,8 | --- | --- | --- |
| ENABLE PERIPHERAL SERVICING | IORS | 288 | --- | 63,6 | 92,3 | --- | --- | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-29 Istruzioni di programmazione a blocchi

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|------------------------|---------------------------------|--------|--------------------------------|-----------------------------------|---------|--------|--------|---------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| BLOCK PROGRAM BEGIN | BPRG | 096 | 2 | 12,1 | 13,0 | 13,0 | 13,0 | --- |
| BLOCK PROGRAM END | BEND | 801 | 1 | 9,6 | 12,3 | 13,1 | 13,1 | --- |
| BLOCK PROGRAM PAUSE | BPPS | 811 | 2 | 10,6 | 12,3 | 14,9 | 14,9 | --- |
| BLOCK PROGRAM RESTART | BPRS | 812 | 2 | 5,1 | 5,6 | 8,3 | 8,3 | --- |
| CONDITIONAL BLOCK EXIT | EXIT (condizione di esecuzione) | 806 | 1 | 10,0 | 11,3 | 12,9 | 12,9 | Condizione EXIT soddisfatta |
| | | | | 4,0 | 4,9 | 7,3 | 7,3 | Condizione EXIT non soddisfatta |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | Condizioni |
|------------------------------|---|--------|--------------------------------|-----------------------------|---------|--------|--------|---------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| CONDITIONAL BLOCK EXIT | EXIT (indirizzo bit) | 806 | 2 | 6,8 | 13,5 | 16,3 | 16,3 | Condizione EXIT soddisfatta |
| | | | | 4,7 | 7,2 | 10,7 | 10,7 | Condizione EXIT non soddisfatta |
| CONDITIONAL BLOCK EXIT (NOT) | EXIT NOT (indirizzo bit) | 806 | 2 | 12,4 | 14,0 | 16,8 | 16,8 | Condizione EXIT soddisfatta |
| | | | | 7,1 | 7,6 | 11,2 | 11,2 | Condizione EXIT non soddisfatta |
| Diramazione | IF (condizione di esecuzione) | 802 | 1 | 4,6 | 4,8 | 7,2 | 7,2 | Condizione IF vera |
| | | | | 6,7 | 7,3 | 10,9 | 10,9 | Condizione IF falsa |
| Diramazione | IF (numero condizione intermedia) | 802 | 2 | 6,8 | 7,2 | 10,4 | 10,4 | Condizione IF vera |
| | | | | 9,0 | 9,6 | 14,2 | 14,2 | Condizione IF falsa |
| Diramazione (NOT) | IF NOT (numero condizione intermedia) | 802 | 2 | 7,1 | 7,6 | 10,9 | 10,9 | Condizione IF vera |
| | | | | 9,2 | 10,1 | 14,7 | 14,7 | Condizione IF falsa |
| Diramazione | ELSE | 803 | 1 | 6,2 | 6,7 | 9,9 | 9,9 | Condizione IF vera |
| | | | | 6,8 | 7,7 | 11,2 | 11,2 | Condizione IF falsa |
| Diramazione | IEND | 804 | 1 | 6,9 | 7,7 | 11,0 | 11,0 | Condizione IF vera |
| | | | | 4,4 | 4,6 | 7,0 | 7,0 | Condizione IF falsa |
| ONE CYCLE AND WAIT | WAIT (condizione di esecuzione) | 805 | 1 | 12,6 | 13,7 | 16,7 | 16,7 | Condizione WAIT soddisfatta |
| | | | | 3,9 | 4,1 | 6,3 | 6,3 | Condizione WAIT non soddisfatta |
| ONE CYCLE AND WAIT | WAIT (numero condizione intermedia) | 805 | 2 | 12,0 | 13,4 | 16,5 | 16,5 | Condizione WAIT soddisfatta |
| | | | | 6,1 | 6,5 | 9,6 | 9,6 | Condizione WAIT non soddisfatta |
| ONE CYCLE AND WAIT (NOT) | WAIT NOT (numero condizione intermedia) | 805 | 2 | 12,2 | 13,8 | 17,0 | 17,0 | Condizione WAIT soddisfatta |
| | | | | 6,4 | 6,9 | 10,1 | 10,1 | Condizione WAIT non soddisfatta |
| COUNTER WAIT | CNTW | 814 | 4 | 17,9 | 22,6 | 27,4 | 27,4 | Impostazione predefinita |
| | | | | 19,1 | 23,9 | 28,7 | 28,7 | Esecuzione normale |
| | CNTWX | 818 | 4 | 17,9 | 22,6 | --- | --- | Impostazione predefinita |
| | | | | 19,1 | 23,9 | --- | --- | Esecuzione normale |
| HIGH-SPEED TIMER WAIT | TMHW | 815 | 3 | 25,8 | 27,9 | 34,1 | 34,1 | Impostazione predefinita |
| | | | | 20,6 | 22,7 | 28,9 | 28,9 | Esecuzione normale |
| | TMHWX | 817 | 3 | 25,8 | 27,9 | --- | --- | Impostazione predefinita |
| | | | | 20,6 | 22,7 | --- | --- | Esecuzione normale |
| | | | | 9,3 | 10,8 | --- | --- | Condizione LEND non soddisfatta |
| Controllo del ciclo | LOOP | 809 | 1 | 7,9 | 9,1 | 12,3 | 12,3 | --- |
| Controllo del ciclo | LEND (condizione di esecuzione) | 810 | 1 | 7,7 | 8,4 | 10,9 | 10,9 | Condizione LEND soddisfatta |
| | | | | 6,8 | 8,0 | 9,8 | 9,8 | Condizione LEND non soddisfatta |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---------------------|--|--------|---|-----------------------------------|---------|--------|--------|---------------------------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| Controllo del ciclo | LEND (numero condizione intermedia) | 810 | 2 | 9,9 | 10,7 | 14,4 | 14,4 | Condizione LEND soddisfatta |
| | | | | 8,9 | 10,3 | 13,0 | 13,0 | Condizione LEND non soddisfatta |
| Controllo del ciclo | LEND NOT (numero condizione intermedia) | 810 | 2 | 10,2 | 11,2 | 14,8 | 14,8 | Condizione LEND soddisfatta |
| | | | | 9,3 | 10,8 | 13,5 | 13,5 | Condizione LEND non soddisfatta |
| TIMER WAIT | TIMW | 813 | 3 | 22,3 | 25,2 | 33,1 | 33,1 | Impostazione predefinita |
| | | | | 24,9 | 27,8 | 35,7 | 35,7 | Esecuzione normale |
| | TIMWX | 816 | 3 | 22,3 | 25,2 | --- | --- | Impostazione predefinita |
| | | | | 24,9 | 27,8 | --- | --- | Esecuzione normale |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-30 Istruzioni di elaborazione delle stringhe di testo

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | Condizioni |
|---------------------|-----------|--------|---|-----------------------------------|---------|--------|--------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| MOV STRING | MOV\$ | 664 | 3 | 45,6 | 66,0 | 84,3 | 84,3 | Trasferimento di 1 carattere |
| CONCATE-NATE STRING | +\$ | 656 | 4 | 86,5 | 126,0 | 167,8 | 167,8 | 1 carattere + 1 carattere |
| GET STRING LEFT | LEFT\$ | 652 | 4 | 53,0 | 77,4 | 94,3 | 94,3 | Recupero di 1 carattere da 2 caratteri |
| GET STRING RIGHT | RGHT\$ | 653 | 4 | 52,2 | 76,3 | 94,2 | 94,2 | Recupero di 1 carattere da 2 caratteri |
| GET STRING MIDDLE | MID\$ | 654 | 5 | 56,5 | 84,6 | 230,2 | 230,2 | Recupero di 1 carattere da 3 caratteri |
| FIND IN STRING | FIND\$ | 660 | 4 | 51,4 | 77,5 | 94,1 | 94,1 | Ricerca di 1 carattere in 2 caratteri |
| STRING LENGTH | LEN\$ | 650 | 3 | 19,8 | 28,9 | 33,4 | 33,4 | Rilevamento di 1 carattere |
| REPLACE IN STRING | RPLC\$ | 661 | 6 | 175,1 | 258,7 | 479,5 | 479,5 | Sostituzione del primo di 2 caratteri con 1 carattere |
| DELETE STRING | DEL\$ | 658 | 5 | 63,4 | 94,2 | 244,6 | 244,6 | Eliminazione del carattere iniziale di 2 caratteri |
| EXCHANGE STRING | XCHG\$ | 665 | 3 | 60,6 | 87,2 | 99,0 | 99,0 | Scambio di 1 carattere con 1 carattere |
| CLEAR STRING | CLR\$ | 666 | 2 | 23,8 | 36,0 | 37,8 | 37,8 | Cancellazione di 1 carattere |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-------------------------------------|-------------------|--------|--------------------------------|-----------------------------|---------|--------|--------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| INSERT INTO STRING | INS\$ | 657 | 5 | 136,5 | 200,6 | 428,9 | 428,9 | Inserimento di 1 carattere dopo il primo di 2 caratteri |
| Istruzioni di confronto di stringhe | LD, AND, OR += \$ | 670 | 4 | 48,5 | 69,8 | 86,2 | 86,2 | Confronto di 1 carattere con 1 carattere |
| | LD, AND, OR +<>\$ | 671 | | | | | | |
| | LD, AND, OR +<\$ | 672 | | | | | | |
| | LD, AND, OR +>\$ | 674 | | | | | | |
| | LD, AND, OR += \$ | 675 | | | | | | |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-1-31 Istruzioni di controllo dei task

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| TASK ON | TKON | 820 | 2 | 19,5 | 26,3 | 26,3 | 26,3 | --- |
| TASK OFF | TKOF | 821 | 2 | 13,3 | 19,0 | 26,3 | 26,3 | --- |

4-1-32 Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva).

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|------------------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|---|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| BLOCK TRANSFER | XFERC | 565 | 4 | 6,4 | 6,5 | --- | --- | Trasferimento di 1 canale |
| | | | | 481,6 | 791,6 | --- | --- | Trasferimento di 1.000 canali |
| SINGLE WORD DISTRIBUTE | DISTC | 566 | 4 | 3,4 | 3,5 | --- | --- | Distribuzione dei dati |
| | | | | 5,9 | 7,3 | --- | --- | Operazione di stack |
| DATA COLLECT | COLLC | 567 | 4 | 3,5 | 3,85 | --- | --- | Distribuzione dei dati |
| | | | | 8 | 9,1 | --- | --- | Operazione di stack |
| | | | | 8,3 | 9,6 | --- | --- | Operazione di stack Lettura FIFO di 1 canale |
| | | | | 2,052,3 | 2,097,5 | --- | --- | Operazione di stack Lettura FIFO di 1.000 canali |
| MOVE BIT | MOVBC | 568 | 4 | 4,5 | 4,88 | --- | --- | --- |
| BIT COUNTER | BCNTC | 621 | 4 | 4,9 | 5 | --- | --- | Conteggio di 1 canale |
| | | | | 1,252.4 | 1284.4 | --- | --- | Conteggio di 1.000 canali |

4-1-33 Istruzioni speciali per blocchi funzione (solo CPU versione 3.0 o successiva)

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | Condizioni |
|-----------------|-----------|--------|--------------------------------|-----------------------------|---------|--------|--------|------------|
| | | | | CPU-6□H | CPU-4□H | CPU-6□ | CPU-4□ | |
| GET VARIABLE ID | GETID | 286 | 4 | 14 | 22.2 | --- | --- | --- |

Linee guida per la conversione delle capacità del programma rispetto ai PLC OMRON precedenti

Nella seguente tabella sono fornite le linee guida per la conversione della capacità del programma (unità di misura: canali) dei PLC OMRON precedenti (PLC SYSMAC C200HX/HG/HE, CVM1 o della serie CV) nella capacità del programma (unità di misura: step) dei PLC della serie CS.

Aggiungere il seguente valore (n) alla capacità di programma (unità di misura: canali) dei PLC precedenti per ogni istruzione per ottenere la capacità di programma (unità di misura: step) dei PLC della serie CS.

| Step per la serie CS = "a" (canali) di PLC precedenti + n | | | |
|---|---|---|--|
| Istruzioni | Variazioni | Valore di n nella conversione da C200HX/HG/HE alla serie CS | Valore di n nella conversione da PLC serie CV o CVM1 alla serie CS |
| Istruzioni di base | Nessuno | OUT, SET, RSET o KEEP(011): -1 Altre istruzioni: 0 | 0 |
| | Differenziazione sul fronte di salita | Nessuno | +1 |
| | Aggiornamento immediato | Nessuno | 0 |
| | Differenziazione sul fronte di salita e aggiornamento immediato | Nessuno | +2 |
| Istruzioni speciali | Nessuno | 0 | -1 |
| | Differenziazione sul fronte di salita | +1 | 0 |
| | Aggiornamento immediato | Nessuno | +3 |
| | Differenziazione sul fronte di salita e aggiornamento immediato | Nessuno | +4 |

Se ad esempio si utilizza OUT con un indirizzo compreso tra CIO 000000 e CIO 25515, la capacità di programma di un PLC C200HX/HG/HE è di 2 canali per istruzione, mentre quella dei PLC della serie CS è pari a 1 (2 - 1) step per istruzione.

Se ad esempio se si utilizza !MOV (istruzione MOVE con aggiornamento immediato), la capacità di programma di un PLC della serie CV è di 4 canali per istruzione, mentre quella dei PLC della serie CS è pari a 7 (4 + 3) step.

4-2 Tempi di esecuzione delle istruzioni e numero di step per la serie CJ

La seguente tabella riporta i tempi di esecuzione per tutte le istruzioni disponibili per i PLC CJ.

Il tempo di esecuzione totale delle istruzioni nell'ambito di un intero programma utente equivale al tempo di esecuzione del programma durante il calcolo del tempo di ciclo (vedere nota).

Nota Ai programmi utente vengono allocati dei task che possono essere eseguiti nell'ambito di task ciclici e di task ad interrupt che soddisfano le condizioni di interrupt.

I tempi di esecuzione per la maggioranza delle istruzioni variano in base alla CPU (CJ1H-CPU6□H, CJ1H-CPU4□H, CJ1M-CPU□□ e CJ1G-CPU4□) e alle condizioni di esecuzione dell'istruzione. Nella seguente tabella sono riportati i dati relativi ai tempi di elaborazione di ciascuna istruzione: la riga superiore indica il tempo minimo insieme alle condizioni di esecuzione, mentre la riga inferiore indica il tempo massimo insieme alle condizioni di esecuzione.

Il tempo di esecuzione può inoltre variare quando la condizione di esecuzione è OFF.

Nella tabella seguente, nella colonna *Lunghezza (step)*, è inoltre riportata la lunghezza di ciascuna istruzione. Il numero di step necessari nell'area del programma utente per ciascuna delle istruzioni della serie CJ varia da 1 a 7 a seconda dell'istruzione e degli operandi specificati. Il numero di step in un programma non corrisponde al numero di istruzioni.

- Nota**
1. La capacità dei programmi per i PLC della serie CJ viene misurata in step, mentre la capacità dei programmi per i PLC OMRON precedenti, ad esempio i PLC delle serie C e CV, veniva misurata in canali. In altre parole, 1 step equivale a 1 canale. Tuttavia, poiché per alcune istruzioni della serie CJ la quantità di memoria richiesta per l'esecuzione è diversa rispetto ai modelli precedenti, la conversione per un PLC della serie CJ della capacità di un programma utente sviluppato per un altro PLC in base al criterio secondo cui 1 canale equivale a 1 step non produrrà un valore accurato. Per indicazioni sulla conversione delle capacità di programmi sviluppati per PLC OMRON precedenti, fare riferimento alle informazioni riportate alla fine di 4-1 *Tempi di esecuzione delle istruzioni e numero di step per la serie CS*.
 2. Per la maggior parte delle istruzioni è supportata la forma differenziale (contrassegnata con ↑, ↓, @ e %). L'utilizzo della differenziazione aumenta i tempi di esecuzione dei seguenti valori.

| Simbolo | CJ1-H | | CJ1M | CJ1 |
|---------|----------|----------|---------|----------|
| | CPU6□H | CPU4□H | CPU□□ | CPU4□ |
| ↑ o ↓ | +0.24 μs | +0.32 μs | +0.5 μs | +0.45 μs |
| @ o % | +0.24 μs | +0.32 μs | +0.5 μs | +0.33 μs |

3. Utilizzare i tempi seguenti come linee guida quando non vengono eseguite le istruzioni.

| CJ1-H | | CJ1M | CJ1 |
|--------------|--------------|-----------------------|-----------------------|
| CPU6□H | CPU4□H | CPU□□ | CPU4□ |
| Circa 0,1 μs | Circa 0,2 μs | Circa da 0,2 a 0,5 μs | Circa da 0,2 a 0,4 μs |

4-2-1 Istruzioni di ingresso sequenza

| Istruzione | Mnemonico | Co-dice | Lunghezza (step) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|------------|-----------|---------|------------------|-----------------------------|--------|--------|--------------------------|---------------|-------------------------------------|
| | | | | CPU6□H | CPU4□H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| LOAD | LD | --- | 1 | 0,02 | 0,04 | 0,08 | 0,10 | 0,10 | --- |
| | !LD | --- | 2 | +21,14 | +21,16 | +21,16 | +24,10 | +28,07 | Aumento per aggiornamento immediato |
| LOAD NOT | LD NOT | --- | 1 | 0,02 | 0,04 | 0,08 | 0,10 | 0,10 | --- |
| | !LD NOT | --- | 2 | +21,14 | +21,16 | +21,16 | +24,10 | +28,07 | Aumento per aggiornamento immediato |
| AND | AND | --- | 1 | 0,02 | 0,04 | 0,08 | 0,10 | 0,10 | --- |
| | !AND | --- | 2 | +21,14 | +21,16 | +21,16 | +24,10 | +28,07 | Aumento per aggiornamento immediato |

| Istruzione | Mnemonico | Codice | Lunghezza (step) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-------------------|-----------|--------|------------------|-----------------------------|------------|--------|---------------------------------------|----------------------|-------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| AND NOT | AND NOT | --- | 1 | 0,02 | 0,04 | 0,08 | 0,10 | 0,10 | --- |
| | !AND NOT | --- | 2 | +21,14 | +21,16 | +21,16 | +24,10 | +28,07 | Aumento per aggiornamento immediato |
| OR | OR | --- | 1 | 0,02 | 0,04 | 0,08 | 0,10 | 0,10 | --- |
| | !OR | --- | 2 | +21,14 | +21,16 | +21,16 | +24,10 | +28,07 | Aumento per aggiornamento immediato |
| OR NOT | OR NOT | --- | 1 | 0,02 | 0,04 | 0,08 | 0,10 | 0,10 | --- |
| | !OR NOT | --- | 2 | +21,14 | +21,16 | +21,16 | +24,10 | +28,07 | Aumento per aggiornamento immediato |
| AND LOAD | AND LD | --- | 1 | 0,02 | 0,04 | 0,08 | 0,05 | 0,05 | --- |
| OR LOAD | OR LD | --- | 1 | 0,02 | 0,04 | 0,08 | 0,05 | 0,05 | --- |
| NOT | NOT | 520 | 1 | 0,02 | 0,04 | 0,08 | 0,05 | 0,05 | --- |
| CONDITION ON | UP | 521 | 3 | 0,3 | 0,42 | 0,54 | 0,50 | 0,50 | --- |
| CONDITION OFF | DOWN | 522 | 4 | 0,3 | 0,42 | 0,54 | 0,50 | 0,50 | --- |
| LOAD BIT TEST | LD TST | 350 | 4 | 0,14 | 0,24 | 0,37 | 0,35 | 0,35 | --- |
| LOAD BIT TEST NOT | LD TSTN | 351 | 4 | 0,14 | 0,24 | 0,37 | 0,35 | 0,35 | --- |
| AND BIT TEST NOT | AND TSTN | 351 | 4 | 0,14 | 0,24 | 0,37 | 0,35 | 0,35 | --- |
| OR BIT TEST | OR TST | 350 | 4 | 0,14 | 0,24 | 0,37 | 0,35 | 0,35 | --- |
| OR BIT TEST NOT | OR TSTN | 351 | 4 | 0,14 | 0,24 | 0,37 | 0,35 | 0,35 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-2 Istruzioni di uscita sequenza

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--------------------|-----------|--------|--------------------------------|-----------------------------|------------|--------|---------------------------------------|----------------------|-------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| OUTPUT | OUT | --- | 1 | 0,02 | 0,04 | 0,21 | 0,35 | 0,35 | --- |
| | !OUT | --- | 2 | +21,37 | +21,37 | +21,37 | +23,07 | +28,60 | Aumento per aggiornamento immediato |
| OUTPUT NOT | OUT NOT | --- | 1 | 0,02 | 0,04 | 0,21 | 0,35 | 0,35 | --- |
| | !OUT NOT | --- | 2 | +21,37 | +21,37 | +21,37 | +23,07 | +28,60 | Aumento per aggiornamento immediato |
| KEEP | KEEP | 11 | 1 | 0,06 | 0,08 | 0,29 | 0,40 | 0,40 | --- |
| DIFFERENTIATE UP | DIFU | 13 | 2 | 0,24 | 0,40 | 0,54 | 0,50 | 0,50 | --- |
| DIFFERENTIATE DOWN | DIFD | 14 | 2 | 0,24 | 0,40 | 0,54 | 0,50 | 0,50 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--------------------|-----------|--------|--------------------------------|-----------------------------|------------|--------|---------------------------------|----------------------|-------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| SET | SET | --- | 1 | 0,02 | 0,06 | 0,21 | 0,30 | 0,30 | --- |
| | !SET | --- | 2 | +21,37 | +21,37 | +21,37 | +23,17 | +28,60 | Aumento per aggiornamento immediato |
| RESET | RSET | --- | 1 | 0,02 | 0,06 | 0,21 | 0,30 | 0,30 | Canale specificato |
| | !RSET | --- | 2 | +21,37 | +21,37 | +21,37 | +23,17 | +28,60 | Aumento per aggiornamento immediato |
| MULTIPLE BIT SET | SETA | 530 | 4 | 5,8 | 6,1 | 7,8 | 11,8 | 11,8 | Con impostazione di 1 bit |
| | | | | 25,7 | 27,2 | 38,8 | 64,1 | 64,1 | Con impostazione di 1.000 bit |
| MULTIPLE BIT RESET | RSTA | 531 | 4 | 5,7 | 6,1 | 7,8 | 11,8 | 11,8 | Con reset di 1 bit |
| | | | | 25,8 | 27,1 | 38,8 | 64,0 | 64,0 | Con reset di 1.000 bit |
| SINGLE BIT SET | SETB | 532 | 2 | 0,24 | 0,34 | --- | 0,5 | 0,5 | --- |
| | !SETB | | 3 | +21,44 | +21,54 | --- | +23,31 | +23,31 | --- |
| SINGLE BIT RESET | RSTB | 533 | 2 | 0,24 | 0,34 | --- | 0,5 | 0,5 | --- |
| | !RSTB | | 3 | +21,44 | +21,54 | --- | +23,31 | +23,31 | --- |
| SINGLE BIT OUTPUT | OUTB | 534 | 2 | 0,22 | 0,32 | --- | 0,45 | 0,45 | --- |
| | !OUTB | | 3 | +21,42 | +21,52 | --- | +23,22 | +23,22 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-3 Istruzioni di controllo sequenza

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--|-----------|--------|--------------------------------|-----------------------------|------------|-------|---------------------------------|----------------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| END | END | 1 | 1 | 5,5 | 6,0 | 4,0 | 7,9 | 7,9 | --- |
| NO OPERATION | NOP | 0 | 1 | 0,02 | 0,04 | 0,12 | 0,05 | 0,05 | --- |
| INTERLOCK | IL | 2 | 1 | 0,06 | 0,06 | 0,12 | 0,15 | 0,15 | --- |
| INTERLOCK CLEAR | ILC | 3 | 1 | 0,06 | 0,06 | 0,12 | 0,15 | 0,15 | --- |
| MULTI-INTERLOCK DIFFERENTIATION HOLD (vedere la nota 2), | MILH | 517 | 3 | 6,1 | 6,5 | --- | 10,3 | 11,7 | Durante l'interblocco |
| | | | | 7,5 | 7,9 | --- | 13,3 | 14,6 | Non durante l'interblocco e senza l'interblocco impostato |
| | | | | 8,9 | 9,7 | --- | 16,6 | 18,3 | Non durante l'interblocco e con l'interblocco impostato |
| MULTI-INTERLOCK DIFFERENTIATION RELEASE (vedere nota 2) | MILR | 518 | 3 | 6,1 | 6,5 | --- | 10,3 | 11,7 | Durante l'interblocco |
| | | | | 7,5 | 7,9 | --- | 13,3 | 14,6 | Non durante l'interblocco e senza l'interblocco impostato |
| | | | | 8,9 | 9,7 | --- | 16,6 | 18,3 | Non durante l'interblocco e con l'interblocco impostato |
| MULTI-INTERLOCK CLEAR (vedere la nota 2). | MILC | 519 | 2 | 5,0 | 5,6 | --- | 8,3 | 12,5 | Interblocco non cancellato |
| | | | | 5,7 | 6,2 | --- | 9,6 | 14,2 | Interblocco cancellato |
| JUMP | JMP | 4 | 2 | 0,38 | 0,48 | 8,1 | 0,95 | 0,95 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|----------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|--|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| JUMP END | JME | 5 | 2 | --- | --- | --- | --- | --- | --- |
| CONDITIONAL JUMP | CJP | 510 | 2 | 0,38 | 0,48 | 7,4 | 0,95 | 0,95 | Quando viene soddisfatta la condizione JMP |
| CONDITIONAL JUMP NOT | CJPN | 511 | 2 | 0,38 | 0,48 | 8,5 | 0,95 | 0,95 | Quando viene soddisfatta la condizione JMP |
| MULTIPLE JUMP | JMP0 | 515 | 1 | 0,06 | 0,06 | 0,12 | 0,15 | 0,15 | --- |
| MULTIPLE JUMP END | JME0 | 516 | 1 | 0,06 | 0,06 | 0,12 | 0,15 | 0,15 | --- |
| FOR LOOP | FOR | 512 | 2 | 0,21 | 0,21 | 0,21 | 1,00 | 1,00 | Designazione di una costante |
| BREAK LOOP | BREAK | 514 | 1 | 0,12 | 0,12 | 0,12 | 0,15 | 0,15 | --- |
| NEXT LOOP | NEXT | 513 | 1 | 0,17 | 0,17 | 0,17 | 0,45 | 0,45 | Se il ciclo continua |
| | | | | 0,12 | 0,12 | 0,12 | 0,55 | 0,55 | Se il ciclo termina |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-2-4 Istruzioni di temporizzatore e contatore

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni | |
|--------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|-----------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | | |
| TIMER | TIM | --- | 3 | 0,56 | 0,88 | 0,42 | 1,30 | 1,30 | --- | |
| | TIMX | 550 | | | | --- | | --- | | |
| COUNTER | CNT | --- | 3 | 0,56 | 0,88 | 0,42 | 1,30 | 1,30 | --- | |
| | CNTX | 546 | | | | --- | | --- | | |
| HIGH-SPEED TIMER | TIMH | 15 | 3 | 0,88 | 1,14 | 0,42 | 1,80 | 1,80 | --- | |
| | TIMHX | 551 | | | | --- | | --- | | |
| ONE-MS TIMER | TMHH | 540 | 3 | 0,86 | 1,12 | 0,42 | 1,75 | 1,75 | --- | |
| | TMHHX | 552 | | | | --- | | --- | | |
| ACCUMULATIVE TIMER | TTIM | 87 | 3 | 16,1 | 17,0 | 21,4 | 27,4 | 30,9 | --- | |
| | | | | 10,9 | 11,4 | 14,8 | | 19,0 | 21,2 | Durante il reset |
| | | | | 8,5 | 8,7 | 10,7 | | 15,0 | 16,6 | Durante l'interblocco |
| | TTIMX | 555 | 3 | 16,1 | 17,0 | --- | 27,4 | --- | --- | |
| | | | | 10,9 | 11,4 | --- | | 19,0 | --- | Durante il reset |
| | | | | 8,5 | 8,7 | --- | | 15,0 | --- | Durante l'interblocco |
| LONG TIMER | TIML | 542 | 4 | 7,6 | 10,0 | 12,8 | 16,3 | 17,2 | --- | |
| | | | | 6,2 | 6,5 | 7,8 | | 13,8 | 15,3 | Durante l'interblocco |
| | TIMLX | 553 | 4 | 7,6 | 10,0 | --- | 16,3 | --- | --- | |
| | | | | 6,2 | 6,5 | --- | | 13,8 | --- | Durante l'interblocco |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---------------------|-----------|--------|--------------------------------|-----------------------------|------------|---------|--------------------------|---------------|----------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| MULTI-OUTPUT TIMER | MTIM | 543 | 4 | 20,9 | 23,3 | 26,0 | 38,55 | 43,3 | --- |
| | | | | 5,6 | 5,8 | 7,8 | 12,9 | 13,73 | Durante il reset |
| | MTIMX | 554 | | 20,9 | 23,3 | --- | 38,55 | --- | --- |
| | | | | 5,6 | 5,8 | --- | 12,9 | --- | Durante il reset |
| REVERSIBLE COUNTER | CNTR | 12 | 3 | 16,9 | 19,0 | 20,9 | 31,8 | 27,2 | --- |
| | CNTRX | 548 | | --- | --- | | | | |
| RESET TIMER/COUNTER | CNR | 545 | 3 | 9,9 | 10,6 | 13,9 | 14,7 | 17,93 | Durante il reset di 1 canale |
| | | | | 4,16 ms | 4,16 ms | 5,42 ms | 6,21 ms | 6,30 ms | Durante il reset di 1.000 canali |
| | CNRX | 547 | | 9,9 | 10,6 | --- | 14,7 | 17,93 | Durante il reset di 1 canale |
| | | | | 4,16 ms | 4,16 ms | --- | 6,21 ms | 6,30 ms | Durante il reset di 1.000 canali |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-5 Istruzioni di confronto

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---|-------------------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| Istruzioni di confronto di ingresso (senza segno) | LD, AND, OR += | 300 | 4 | 0,10 | 0,16 | 0,37 | 0,35 | 0,35 | --- |
| | LD, AND, OR + <> | 305 | | | | | | | |
| | LD, AND, OR + < | 310 | | | | | | | |
| | LD, AND, OR +<= | 315 | | | | | | | |
| | LD, AND, OR +> | 320 | | | | | | | |
| | LD, AND, OR +>= | 325 | | | | | | | |
| Istruzioni di confronto di ingresso (doppio, senza segno) | LD, AND, OR +=+L | 301 | 4 | 0,10 | 0,16 | 0,54 | 0,35 | 0,35 | --- |
| | LD, AND, OR +<>+L | 306 | | | | | | | --- |
| | LD, AND, OR +<+L | 311 | | | | | | | --- |
| | LD, AND, OR +<=+L | 316 | | | | | | | --- |
| | LD, AND, OR +>+L | 321 | | | | | | | --- |
| | LD, AND, OR +>=+L | 326 | | | | | | | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---|--------------------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|-------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| Istruzioni di confronto di ingresso (con segno) | LD, AND, OR +=+S | 302 | 4 | 0,10 | 0,16 | 6,50 | 0,35 | 0,35 | --- |
| | LD, AND, OR +<>+S | 307 | | | | | | | |
| | LD, AND, OR +<+S | 312 | | | | | | | |
| | LD, AND, OR +<= | 317 | | | | | | | |
| | LD, AND, OR +>+S | 322 | | | | | | | |
| | LD, AND, OR +>=+S | 327 | | | | | | | |
| Istruzioni di confronto di ingresso (doppio, con segno) | LD, AND, OR +=+SL | 303 | 4 | 0,10 | 0,16 | 6,50 | 0,35 | 0,35 | --- |
| | LD, AND, OR +<>+SL | 308 | | | | | | | |
| | LD, AND, OR +<+SL | 313 | | | | | | | |
| | LD, AND, OR +<=+SL | 318 | | | | | | | |
| | LD, AND, OR +>+SL | 323 | | | | | | | |
| | LD, AND, OR +>=+SL | 328 | | | | | | | |
| Istruzioni di confronto di date e ore (vedere la nota 2). | LD, AND, OR +DT | 341 | 4 | 25,1 | 36,4 | --- | 18,8 | 39,6 | --- |
| | LD, AND, OR +<>DT | 342 | 4 | 25,2 | 36,4 | --- | 45,6 | 40,6 | --- |
| | LD, AND, OR +<DT | 343 | 4 | 25,2 | 36,4 | --- | 45,6 | 40,7 | --- |
| | LD, AND, OR +<=DT | 344 | 4 | 25,2 | 36,4 | --- | 18,8 | 39,6 | --- |
| | LD, AND, OR +>DT | 345 | 4 | 25,1 | 36,4 | --- | 45,6 | 41,1 | --- |
| | LD, AND, OR +>=DT | 346 | 4 | 25,2 | 36,4 | --- | 18,8 | 39,6 | --- |
| COMPARE | CMP | 20 | 3 | 0,04 | 0,04 | 0,29 | 0,10 | 0,10 | --- |
| | !CMP | 20 | 7 | 42,1 | 42,1 | 42,4 | +45,2 | 45,2 | Aumento per aggiornamento immediato |
| DOUBLE COMPARE | CMPL | 60 | 3 | 0,08 | 0,08 | 0,46 | 0,50 | 0,50 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|------------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|-------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| SIGNED BINARY COMPARE | CPS | 114 | 3 | 0,08 | 0,08 | 6,50 | 0,30 | 0,30 | --- |
| | ICPS | 114 | 7 | 35,9 | 35,9 | 42,4 | +45,2 | 45,2 | Aumento per aggiornamento immediato |
| DOUBLE SIGNED BINARY COMPARE | CPSL | 115 | 3 | 0,08 | 0,08 | 6,50 | 0,50 | 0,50 | --- |
| TABLE COMPARE | TCMP | 85 | 4 | 14,0 | 15,2 | 21,9 | 29,77 | 32,13 | --- |
| MULTIPLE COMPARE | MCMP | 19 | 4 | 20,5 | 22,8 | 31,2 | 45,80 | 48,67 | --- |
| UNSIGNED BLOCK COMPARE | BCMP | 68 | 4 | 21,5 | 23,7 | 32,6 | 47,93 | 51,67 | --- |
| EXPANDED BLOCK COMPARE | BCMP2 | 502 | 4 | --- | --- | --- | 13,20 | 19,33 | Numero di canali di dati: 1 |
| | | | | --- | --- | --- | 650,0 | 754,67 | Numero di canali di dati: 255 |
| AREA RANGE COMPARE | ZCP | 88 | 3 | 5,3 | 5,4 | --- | 11,53 | 12,43 | --- |
| DOUBLE AREA RANGE COMPARE | ZCPL | 116 | 3 | 5,5 | 6,7 | --- | 11,28 | 11,90 | --- |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-2-6 Istruzioni di spostamento dei dati

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-----------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|-------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| MOVE | MOV | 21 | 3 | 0,18 | 0,20 | 0,29 | 0,30 | 0,30 | --- |
| | IMOV | 21 | 7 | 21,38 | 21,40 | 42,36 | +35,1 | 43,0 | Aumento per aggiornamento immediato |
| DOUBLE MOVE | MOVL | 498 | 3 | 0,32 | 0,34 | 0,50 | 0,60 | 0,60 | --- |
| MOVE NOT | MVN | 22 | 3 | 0,18 | 0,20 | 0,29 | 0,35 | 0,35 | --- |
| DOUBLE MOVE NOT | MVNL | 499 | 3 | 0,32 | 0,34 | 0,50 | 0,60 | 0,60 | --- |
| MOVE BIT | MOVb | 82 | 4 | 0,24 | 0,34 | 7,5 | 0,50 | 0,50 | --- |
| MOVE DIGIT | MOVd | 83 | 4 | 0,24 | 0,34 | 7,3 | 0,50 | 0,50 | --- |
| MULTIPLE BIT TRANSFER | XFRb | 62 | 4 | 10,1 | 10,8 | 13,6 | 20,9 | 22,1 | Trasferimento di 1 bit |
| | | | | 186,4 | 189,8 | 269,2 | 253,3 | 329,7 | Trasferimento di 255 bit |
| BLOCK TRANSFER | XFER | 70 | 4 | 0,36 | 0,44 | 11,2 | 0,8 | 0,8 | Trasferimento di 1 canale |
| | | | | 300,1 | 380,1 | 633,5 | 650,2 | 650,2 | Trasferimento di 1.000 canali |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-----------------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| BLOCK SET | BSET | 71 | 4 | 0,26 | 0,28 | 8,5 | 0,55 | 0,55 | Impostazione di 1 canale |
| | | | | 200,1 | 220,1 | 278,3 | 400,2 | 400,2 | Impostazione di 1.000 canali |
| DATA EXCHANGE | XCHG | 73 | 3 | 0,40 | 0,56 | 0,7 | 0,80 | 0,80 | --- |
| DOUBLE DATA EXCHANGE | XCGL | 562 | 3 | 0,76 | 1,04 | 1,3 | 1,5 | 1,5 | --- |
| SINGLE WORD DISTRIBUTE | DIST | 80 | 4 | 5,1 | 5,4 | 7,0 | 6,6 | 12,47 | --- |
| DATA COLLECT | COLL | 81 | 4 | 5,1 | 5,3 | 7,1 | 6,5 | 12,77 | --- |
| MOVE TO REGISTER | MOVR | 560 | 3 | 0,08 | 0,08 | 0,50 | 0,60 | 0,60 | --- |
| MOVE TIMER/COUNTER PV TO REGISTER | MOVRW | 561 | 3 | 0,42 | 0,50 | 0,50 | 0,60 | 0,60 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-7 Istruzioni di scorrimento dei dati

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-----------------------------|-----------|--------|--------------------------------|-----------------------------|------------|---------|--------------------------|---------------|-----------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| SHIFT REGISTER | SFT | 10 | 3 | 7,4 | 10,4 | 10,4 | 11,9 | 15,3 | Scorrimento di 1 canale |
| | | | | 433,2 | 488,0 | 763,1 | 1,39 ms | 1,43 ms | Scorrimento di 1.000 canali |
| REVERSIBLE SHIFT REGISTER | SFTR | 84 | 4 | 6,9 | 7,2 | 9,6 | 11,4 | 15,5 | Scorrimento di 1 canale |
| | | | | 615,3 | 680,2 | 859,6 | 1,43 ms | 1,55 ms | Scorrimento di 1.000 canali |
| ASYNCHRONOUS SHIFT REGISTER | ASFT | 17 | 4 | 6,2 | 6,4 | 7,7 | 13,4 | 14,2 | Scorrimento di 1 canale |
| | | | | 1,22 ms | 1,22 ms | 2,01 ms | 2,75 ms | 2,99 ms | Scorrimento di 1.000 canali |
| WORD SHIFT | WSFT | 16 | 4 | 4,5 | 4,7 | 7,8 | 9,6 | 12,3 | Scorrimento di 1 canale |
| | | | | 171,5 | 171,7 | 781,7 | 928,0 | 933,3 | Scorrimento di 1.000 canali |
| ARITHMETIC SHIFT LEFT | ASL | 25 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE SHIFT LEFT | ASLL | 570 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| ARITHMETIC SHIFT RIGHT | ASR | 26 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE SHIFT RIGHT | ASRL | 571 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|-----------------------------------|-----------|--------|--------------------------------|-----------------------------|------------|---------|--------------------------|---------------|-----------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| ROTATE LEFT | ROL | 27 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE ROTATE LEFT | ROLL | 572 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| ROTATE LEFT WITHOUT CARRY | RLNC | 574 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE ROTATE LEFT WITHOUT CARRY | RLNL | 576 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| ROTATE RIGHT | ROR | 28 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE ROTATE RIGHT | RORL | 573 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| ROTATE RIGHT WITHOUT CARRY | RRNC | 575 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE ROTATE RIGHT WITHOUT CARRY | RRNL | 577 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| ONE DIGIT SHIFT LEFT | SLD | 74 | 3 | 5,9 | 6,1 | 8,2 | 7,6 | 12,95 | Scorrimento di 1 canale |
| | | | | 561,1 | 626,3 | 760,7 | 1,15 ms | 1,27 ms | Scorrimento di 1.000 canali |
| ONE DIGIT SHIFT RIGHT | SRD | 75 | 3 | 6,9 | 7,1 | 8,7 | 8,6 | 15,00 | Scorrimento di 1 canale |
| | | | | 760,5 | 895,5 | 1,07 ms | 1,72 ms | 1,82 ms | Scorrimento di 1.000 canali |
| SHIFT N-BIT DATA LEFT | NSFL | 578 | 4 | 7,5 | 8,3 | 10,5 | 14,8 | 16,0 | Scorrimento di 1 bit |
| | | | | 40,3 | 45,4 | 55,5 | 86,7 | 91,3 | Scorrimento di 1.000 bit |
| SHIFT N-BIT DATA RIGHT | NSFR | 579 | 4 | 7,5 | 8,3 | 10,5 | 14,7 | 15,9 | Scorrimento di 1 bit |
| | | | | 50,5 | 55,3 | 69,3 | 114,1 | 119,6 | Scorrimento di 1.000 bit |
| SHIFT N-BITS LEFT | NASL | 580 | 3 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE SHIFT N-BITS LEFT | NSLL | 582 | 3 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| SHIFT N-BITS RIGHT | NASR | 581 | 3 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE SHIFT N-BITS RIGHT | NSRL | 583 | 3 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-8 Istruzioni di incremento e decremento

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|-------------------------|-----------|--------|--------------------------------|-----------------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| INCREMENT BINARY | ++ | 590 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE INCREMENT BINARY | ++L | 591 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| DECREMENT BINARY | -- | 592 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE DECREMENT BINARY | --L | 593 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |
| INCREMENT BCD | ++B | 594 | 2 | 6,4 | 4,5 | 7,4 | 12,3 | 14,7 | --- |
| DOUBLE INCREMENT BCD | ++BL | 595 | 2 | 5,6 | 4,9 | 6,1 | 9,24 | 10,8 | --- |
| DECREMENT BCD | --B | 596 | 2 | 6,3 | 4,6 | 7,2 | 11,9 | 14,9 | --- |
| DOUBLE DECREMENT BCD | --BL | 597 | 2 | 5,3 | 4,7 | 7,1 | 9,0 | 10,7 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-9 Istruzioni matematiche con simboli

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|--|-----------|--------|--------------------------------|-----------------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| SIGNED BINARY ADD WITHOUT CARRY | + | 400 | 4 | 0,18 | 0,20 | 0,37 | 0,30 | 0,30 | --- |
| DOUBLE SIGNED BINARY ADD WITHOUT CARRY | +L | 401 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |
| SIGNED BINARY ADD WITH CARRY | +C | 402 | 4 | 0,18 | 0,20 | 0,37 | 0,40 | 0,40 | --- |
| DOUBLE SIGNED BINARY ADD WITH CARRY | +CL | 403 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |
| BCD ADD WITHOUT CARRY | +B | 404 | 4 | 8,2 | 8,4 | 14,0 | 18,9 | 21,5 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| DOUBLE BCD ADD WITHOUT CARRY | +BL | 405 | 4 | 13,3 | 14,5 | 19,0 | 24,4 | 27,7 | --- |
| BCD ADD WITH CARRY | +BC | 406 | 4 | 8,9 | 9,1 | 14,5 | 19,7 | 22,6 | --- |
| DOUBLE BCD ADD WITH CARRY | +BCL | 407 | 4 | 13,8 | 15,0 | 19,6 | 25,2 | 28,8 | --- |
| SIGNED BINARY SUBTRACT WITHOUT CARRY | — | 410 | 4 | 0,18 | 0,20 | 0,37 | 0,3 | 0,3 | --- |
| DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY | —L | 411 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |
| SIGNED BINARY SUBTRACT WITH CARRY | —C | 412 | 4 | 0,18 | 0,20 | 0,37 | 0,3 | 0,3 | --- |
| DOUBLE SIGNED BINARY SUBTRACT WITH CARRY | —CL | 413 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |
| BCD SUBTRACT WITHOUT CARRY | —B | 414 | 4 | 8,0 | 8,2 | 13,1 | 18,1 | 20,5 | --- |
| DOUBLE BCD SUBTRACT WITHOUT CARRY | —BL | 415 | 4 | 12,8 | 14,0 | 18,2 | 23,2 | 26,7 | --- |
| BCD SUBTRACT WITH CARRY | —BC | 416 | 4 | 8,5 | 8,6 | 13,8 | 19,1 | 21,6 | --- |
| DOUBLE BCD SUBTRACT WITH CARRY | —BCL | 417 | 4 | 13,4 | 14,7 | 18,8 | 24,3 | 27,7 | --- |
| SIGNED BINARY MULTIPLY | * | 420 | 4 | 0,38 | 0,40 | 0,58 | 0,65 | 0,65 | --- |
| DOUBLE SIGNED BINARY MULTIPLY | *L | 421 | 4 | 7,23 | 8,45 | 11,19 | 13,17 | 15,0 | --- |
| UNSIGNED BINARY MULTIPLY | *U | 422 | 4 | 0,38 | 0,40 | 0,58 | 0,75 | 0,75 | --- |
| DOUBLE UNSIGNED BINARY MULTIPLY | *UL | 423 | 4 | 7,1 | 8,3 | 10,63 | 13,30 | 15,2 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|-------------------------------|-----------|--------|--------------------------------|-----------------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| BCD MULTIPLY | *B | 424 | 4 | 9,0 | 9,2 | 12,8 | 17,5 | 19,7 | --- |
| DOUBLE BCD MULTIPLY | *BL | 425 | 4 | 23,0 | 24,2 | 35,2 | 36,3 | 45,7 | --- |
| SIGNED BINARY DIVIDE | / | 430 | 4 | 0,40 | 0,42 | 0,83 | 0,70 | 0,70 | --- |
| DOUBLE SIGNED BINARY DIVIDE | /L | 431 | 4 | 7,2 | 8,4 | 9,8 | 13,7 | 15,5 | --- |
| UNSIGNED BINARY DIVIDE | /U | 432 | 4 | 0,40 | 0,42 | 0,83 | 0,8 | 0,8 | --- |
| DOUBLE UNSIGNED BINARY DIVIDE | /UL | 433 | 4 | 6,9 | 8,1 | 9,1 | 12,8 | 14,7 | --- |
| BCD DIVIDE | /B | 434 | 4 | 8,6 | 8,8 | 15,9 | 19,3 | 22,8 | --- |
| DOUBLE BCD DIVIDE | /BL | 435 | 4 | 17,7 | 18,9 | 26,2 | 27,1 | 34,7 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-10 Istruzioni di conversione

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|--------------------------------|-----------|--------|--------------------------------|-----------------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| BCD-TO-BINARY | BIN | 023 | 3 | 0,22 | 0,24 | 0,29 | 0,40 | 0,40 | --- |
| DOUBLE BCD-TO-DOUBLE BINARY | BINL | 058 | 3 | 6,5 | 6,8 | 9,1 | 12,3 | 13,7 | --- |
| BINARY-TO-BCD | BCD | 024 | 3 | 0,24 | 0,26 | 8,3 | 7,62 | 9,78 | --- |
| DOUBLE BINARY-TO-DOUBLE BCD | BCDL | 059 | 3 | 6,7 | 7,0 | 9,2 | 10,6 | 12,8 | --- |
| 2'S COMPLEMENT | NEG | 160 | 3 | 0,18 | 0,20 | 0,29 | 0,35 | 0,35 | --- |
| DOUBLE 2'S COMPLEMENT | NEGL | 161 | 3 | 0,32 | 0,34 | 0,5 | 0,60 | 0,60 | --- |
| 16-BIT TO 32-BIT SIGNED BINARY | SIGN | 600 | 3 | 0,32 | 0,34 | 0,50 | 0,60 | 0,60 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|-----------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| DATA DECODER | MLPX | 076 | 4 | 0,32 | 0,42 | 8,8 | 0,85 | 0,85 | Decodifica di 1 cifra (da 4 a 16) |
| | | | | 0,98 | 1,20 | 12,8 | 1,60 | 1,60 | Decodifica di 4 cifre (da 4 a 16) |
| | | | | 3,30 | 4,00 | 20,3 | 4,70 | 4,70 | Decodifica di 1 cifra (da 8 a 256) |
| | | | | 6,50 | 7,90 | 33,4 | 8,70 | 8,70 | Decodifica di 2 cifre (da 8 a 256) |
| DATA ENCODER | DMPX | 077 | 4 | 7,5 | 7,9 | 10,4 | 9,4 | 13,9 | Codifica di 1 cifra (da 16 a 4) |
| | | | | 49,6 | 50,2 | 59,1 | 57,3 | 71,73 | Codifica di 4 cifre (da 16 a 4) |
| | | | | 18,2 | 18,6 | 23,6 | 56,8 | 82,7 | Codifica di 1 cifra (da 256 a 8) |
| | | | | 55,1 | 57,4 | 92,5 | 100,0 | 150,7 | Codifica di 2 cifre (da 256 a 8) |
| ASCII CONVERT | ASC | 086 | 4 | 6,8 | 7,1 | 9,7 | 8,3 | 14,6 | Conversione di 1 cifra in ASCII |
| | | | | 11,2 | 11,7 | 15,1 | 19,1 | 21,8 | Conversione di 4 cifre in ASCII |
| ASCII TO HEX | HEX | 162 | 4 | 7,1 | 7,4 | 10,1 | 12,1 | 15,6 | Conversione di 1 cifra |
| COLUMN TO LINE | LINE | 063 | 4 | 19,0 | 23,1 | 29,1 | 37,0 | 40,3 | --- |
| LINE TO COLUMN | COLM | 064 | 4 | 23,2 | 27,5 | 37,3 | 45,7 | 48,2 | --- |
| SIGNED BCD-TO-BINARY | BINS | 470 | 4 | 8,0 | 8,3 | 12,1 | 16,2 | 17,0 | Impostazione del formato dati n. 0 |
| | | | | 8,0 | 8,3 | 12,1 | 16,2 | 17,1 | Impostazione del formato dati n. 1 |
| | | | | 8,3 | 8,6 | 12,7 | 16,5 | 17,7 | Impostazione del formato dati n. 2 |
| | | | | 8,5 | 8,8 | 13,0 | 16,5 | 17,6 | Impostazione del formato dati n. 3 |
| DOUBLE SIGNED BCD-TO-BINARY | BISL | 472 | 4 | 9,2 | 9,6 | 13,6 | 18,4 | 19,6 | Impostazione del formato dati n. 0 |
| | | | | 9,2 | 9,6 | 13,7 | 18,5 | 19,8 | Impostazione del formato dati n. 1 |
| | | | | 9,5 | 9,9 | 14,2 | 18,6 | 20,1 | Impostazione del formato dati n. 2 |
| | | | | 9,6 | 10,0 | 14,4 | 18,7 | 20,1 | Impostazione del formato dati n. 3 |
| SIGNED BINARY-TO-BCD | BCDS | 471 | 4 | 6,6 | 6,9 | 10,6 | 13,5 | 16,4 | Impostazione del formato dati n. 0 |
| | | | | 6,7 | 7,0 | 10,8 | 13,8 | 16,7 | Impostazione del formato dati n. 1 |
| | | | | 6,8 | 7,1 | 10,9 | 13,9 | 16,8 | Impostazione del formato dati n. 2 |
| | | | | 7,2 | 7,5 | 11,5 | 14,0 | 17,1 | Impostazione del formato dati n. 3 |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--|-----------|--------|--------------------------------|-----------------------------|-------------|-------|--------------------------|---------------|------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| DOUBLE SIGNED BINARY-TO-BCD | BDSL | 473 | 4 | 8,1 | 8,4 | 11,6 | 11,4 | 12,5 | Impostazione del formato dati n. 0 |
| | | | | 8,2 | 8,6 | 11,8 | 11,7 | 12,73 | Impostazione del formato dati n. 1 |
| | | | | 8,3 | 8,7 | 12,0 | 11,8 | 12,8 | Impostazione del formato dati n. 2 |
| | | | | 8,8 | 9,2 | 12,5 | 11,9 | 13,0 | Impostazione del formato dati n. 3 |
| GRAY CODE CONVERSION (vedere la nota 2), | GRY | 474 | 4 | 46,9 | 72,1 | --- | 80,0 | 71,2 | Binario a 8 bit |
| | | | | 49,6 | 75,2 | --- | 83,0 | 75,6 | BCD a 8 bit |
| | | | | 57,7 | 87,7 | --- | 95,9 | 86,4 | Angolo a 8 bit |
| | | | | 61,8 | 96,7 | --- | 104,5 | 91,6 | Binario a 15 bit |
| | | | | 64,5 | 99,6 | --- | 107,5 | 96,1 | BCD a 15 bit |
| | | | | 72,8 | 112,4 | --- | 120,4 | 107,3 | Angolo a 15 bit |
| | | | | 52,3 | 87,2 | --- | 88,7 | 82,4 | 360° binario |
| | | | | 55,1 | 90,4 | --- | 91,7 | 86,8 | 360° BCD |
| 64,8 | 98,5 | --- | 107,3 | 98,1 | 360° angolo | | | | |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-2-11 Istruzioni logiche

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|----------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| LOGICAL AND | ANDW | 034 | 4 | 0,18 | 0,20 | 0,37 | 0,30 | 0,30 | --- |
| DOUBLE LOGICAL AND | ANDL | 610 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |
| LOGICAL OR | ORW | 035 | 4 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE LOGICAL OR | ORWL | 611 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |
| EXCLUSIVE OR | XORW | 036 | 4 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE EXCLUSIVE OR | XORL | 612 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |
| EXCLUSIVE NOR | XNRW | 037 | 4 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE EXCLUSIVE NOR | XNRL | 613 | 4 | 0,32 | 0,34 | 0,54 | 0,60 | 0,60 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| COMPLEMENT | COM | 029 | 2 | 0,22 | 0,32 | 0,37 | 0,45 | 0,45 | --- |
| DOUBLE COMPLEMENT | COML | 614 | 2 | 0,40 | 0,56 | 0,67 | 0,80 | 0,80 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-12 Istruzioni matematiche speciali

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-----------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|--|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| BINARY ROOT | ROTB | 620 | 3 | 49,6 | 50,0 | 530,7 | 56,5 | 82,7 | --- |
| BCD SQUARE ROOT | ROOT | 072 | 3 | 13,7 | 13,9 | 514,5 | 59,3 | 88,4 | --- |
| ARITHMETIC PROCESS | APR | 069 | 4 | 6,7 | 6,9 | 32,3 | 14,0 | 15,0 | Designazione di SIN e COS |
| | | | | 17,2 | 18,4 | 78,3 | 32,2 | 37,9 | Designazione dell'approssimazione lineare a segmenti |
| FLOATING POINT DIVIDE | FDIV | 079 | 4 | 116,6 | 176,6 | 176,6 | 246,0 | 154,7 | --- |
| BIT COUNTER | BCNT | 067 | 4 | 0,3 | 0,38 | 22,1 | 0,65 | 0,65 | Conteggio di 1 canale |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-13 Istruzioni matematiche a virgola mobile

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| FLOATING TO 16-BIT | FIX | 450 | 3 | 10,6 | 10,8 | 14,5 | 16,2 | 19,5 | --- |
| FLOATING TO 32-BIT | FIXL | 451 | 3 | 10,8 | 11,0 | 14,6 | 16,6 | 21,7 | --- |
| 16-BIT TO FLOATING | FLT | 452 | 3 | 8,3 | 8,5 | 11,1 | 12,2 | 14,6 | --- |
| 32-BIT TO FLOATING | FLTL | 453 | 3 | 8,3 | 8,5 | 10,8 | 14,0 | 15,8 | --- |
| FLOATING-POINT ADD | +F | 454 | 4 | 8,0 | 9,2 | 10,2 | 13,3 | 15,7 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---------------------------------------|------------------|--------|--------------------------------|-----------------------------|------------|-------|---------------------------------------|----------------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| FLOATING-POINT SUBTRACT | -F | 455 | 4 | 8,0 | 9,2 | 10,3 | 13,3 | 15,8 | --- |
| FLOATING POINT DIVIDE | /F | 457 | 4 | 8,7 | 9,9 | 12,0 | 14,0 | 17,6 | --- |
| FLOATING-POINT MULTIPLY | *F | 456 | 4 | 8,0 | 9,2 | 10,5 | 13,2 | 15,8 | --- |
| DEGREES TO RADIANS | RAD | 458 | 3 | 10,1 | 10,2 | 14,9 | 15,9 | 20,6 | --- |
| RADIANS TO DEGREES | DEG | 459 | 3 | 9,9 | 10,1 | 14,8 | 15,7 | 20,4 | --- |
| SINE | SIN | 460 | 3 | 42,0 | 42,2 | 61,1 | 47,9 | 70,9 | --- |
| COSINE | COS | 461 | 3 | 31,5 | 31,8 | 44,1 | 41,8 | 51,0 | --- |
| TANGENT | TAN | 462 | 3 | 16,3 | 16,6 | 22,6 | 20,8 | 27,6 | --- |
| ARC SINE | ASIN | 463 | 3 | 17,6 | 17,9 | 24,1 | 80,3 | 122,9 | --- |
| ARC COSINE | ACOS | 464 | 3 | 20,4 | 20,7 | 28,0 | 25,3 | 33,5 | --- |
| ARC TANGENT | ATAN | 465 | 3 | 16,1 | 16,4 | 16,4 | 45,9 | 68,9 | --- |
| SQUARE ROOT | SQRT | 466 | 3 | 19,0 | 19,3 | 28,1 | 26,2 | 33,2 | --- |
| EXPONENT | EXP | 467 | 3 | 65,9 | 66,2 | 96,7 | 68,8 | 108,2 | --- |
| LOGARITHM | LOG | 468 | 3 | 12,8 | 13,1 | 17,4 | 69,4 | 103,7 | --- |
| EXPONENTIAL POWER | PWR | 840 | 4 | 125,4 | 126,0 | 181,7 | 134,0 | 201,0 | --- |
| Confronto di simboli a virgola mobile | LD, AND, OR +=F | 329 | 3 | 6,6 | 8,3 | --- | 12,6 | 15,37 | --- |
| | LD, AND, OR +<>F | 330 | | | | | | | |
| | LD, AND, OR +<F | 331 | | | | | | | |
| | LD, AND, OR +<=F | 332 | | | | | | | |
| | LD, AND, OR +>F | 333 | | | | | | | |
| | LD, AND, OR +>=F | 334 | | | | | | | |
| FLOATING-POINT TO ASCII | FSTR | 448 | 4 | 48,5 | 48,9 | --- | 58,4 | 85,7 | --- |
| ASCII TO FLOATING-POINT | FVAL | 449 | 3 | 21,1 | 21,3 | --- | 31,1 | 43,773 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-14 Istruzioni a virgola mobile in doppia precisione

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|----------------------------------|------------------|--------|--------------------------------|-----------------------------|------------|-------|---------------------------------------|----------------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| DOUBLE SYMBOL COMPARISON | LD, AND, OR +=D | 335 | 3 | 8,5 | 10,3 | --- | 16,2 | 19,9 | --- |
| | LD, AND, OR +<>D | 336 | | | | | | | |
| | LD, AND, OR +<D | 337 | | | | | | | |
| | LD, AND, OR +=<D | 338 | | | | | | | |
| | LD, AND, OR +>D | 339 | | | | | | | |
| | LD, AND, OR +=>D | 340 | | | | | | | |
| DOUBLE FLOATING TO 16-BIT BINARY | FIXD | 841 | 3 | 11,7 | 12,1 | --- | 16,1 | 21,6 | --- |
| DOUBLE FLOATING TO 32-BIT BINARY | FIXLD | 842 | 3 | 11,6 | 12,1 | --- | 16,4 | 21,7 | --- |
| 16-BIT BINARY TO DOUBLE FLOATING | DBL | 843 | 3 | 9,9 | 10,0 | --- | 14,3 | 16,5 | --- |
| 32-BIT BINARY TO DOUBLE FLOATING | DBLL | 844 | 3 | 9,8 | 10,0 | --- | 16,0 | 17,7 | --- |
| DOUBLE FLOATING-POINT ADD | +D | 845 | 4 | 11,2 | 11,9 | --- | 18,3 | 23,6 | --- |
| DOUBLE FLOATING-POINT SUBTRACT | -D | 846 | 4 | 11,2 | 11,9 | --- | 18,3 | 23,6 | --- |
| DOUBLE FLOATING-POINT MULTIPLY | *D | 847 | 4 | 12,0 | 12,7 | --- | 19,0 | 25,0 | --- |
| DOUBLE FLOATING-POINT DIVIDE | /D | 848 | 4 | 23,5 | 24,2 | --- | 30,5 | 44,3 | --- |
| DOUBLE DEGREES TO RADIAN | RADD | 849 | 3 | 27,4 | 27,8 | --- | 32,7 | 49,1 | --- |
| DOUBLE RADIAN TO DEGREE | DEGD | 850 | 3 | 11,2 | 11,9 | --- | 33,5 | 48,4 | --- |
| DOUBLE SINE | SIND | 851 | 3 | 45,4 | 45,8 | --- | 67,9 | 76,7 | --- |
| DOUBLE COSINE | COSD | 852 | 3 | 43,0 | 43,4 | --- | 70,9 | 72,3 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| DOUBLE TANGENT | TAND | 853 | 3 | 20,1 | 20,5 | --- | 97,9 | 157,0 | --- |
| DOUBLE ARC SINE | ASIND | 854 | 3 | 21,5 | 21,9 | --- | 32,3 | 37,3 | --- |
| DOUBLE ARC COSINE | ACOSD | 855 | 3 | 24,7 | 25,1 | --- | 29,9 | 42,5 | --- |
| DOUBLE ARC TANGENT | ATAND | 856 | 3 | 19,3 | 19,7 | --- | 24,0 | 34,4 | --- |
| DOUBLE SQUARE ROOT | SQRD | 857 | 3 | 47,4 | 47,9 | --- | 52,9 | 81,9 | --- |
| DOUBLE EXPONENT | EXPD | 858 | 3 | 121,0 | 121,4 | --- | 126,3 | 201,3 | --- |
| DOUBLE LOGARITHM | LOGD | 859 | 3 | 16,0 | 16,4 | --- | 21,6 | 29,3 | --- |
| DOUBLE EXPONENTIAL POWER | PWRD | 860 | 4 | 223,9 | 224,2 | --- | 232,3 | 373,4 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-15 Istruzioni di elaborazione dei dati delle tabelle

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|------------------------|-----------|--------|--------------------------------|-----------------------------|------------|---------|--------------------------|---------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| SET STACK | SSET | 630 | 3 | 8,0 | 8,3 | 8,5 | 14,2 | 20,3 | Designazione di 5 canali nell'area di stack |
| | | | | 231,6 | 251,8 | 276,8 | 426,5 | 435,3 | Designazione di 1.000 canali nell'area di stack |
| PUSH ONTO STACK | PUSH | 632 | 3 | 6,5 | 8,6 | 9,1 | 15,7 | 16,4 | --- |
| FIRST IN FIRST OUT | FIFO | 633 | 3 | 6,9 | 8,9 | 10,6 | 15,8 | 16,8 | Designazione di 5 canali nell'area di stack |
| | | | | 352,6 | 434,3 | 1,13 ms | 728,0 | 732,0 | Designazione di 1.000 canali nell'area di stack |
| LAST IN FIRST OUT | LIFO | 634 | 3 | 7,0 | 9,0 | 9,9 | 16,6 | 17,2 | --- |
| DIMENSION RECORD TABLE | DIM | 631 | 5 | 15,2 | 21,6 | 142,1 | 27,8 | 27,1 | --- |
| SET RECORD LOCATION | SETR | 635 | 4 | 5,4 | 5,9 | 7,0 | 12,8 | 13,2 | --- |
| GET RECORD NUMBER | GETR | 636 | 4 | 7,8 | 8,4 | 11,0 | 16,1 | 18,3 | --- |
| DATA SEARCH | SRCH | 181 | 4 | 15,5 | 19,5 | 19,5 | 29,1 | 26,4 | Ricerca di 1 canale |
| | | | | 2,42 ms | 3,34 ms | 3,34 ms | 4,41 ms | 3,60 ms | Ricerca di 1.000 canali |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|----------------------|-----------|--------|--------------------------------|-----------------------------|------------|---------|--------------------------|---------------|---------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| SWAP BYTES | SWAP | 637 | 3 | 12,2 | 13,6 | 13,6 | 21,0 | 18,4 | Scambio di 1 canale |
| | | | | 1,94 ms | 2,82 ms | 2,82 ms | 3,65 ms | 3,15 ms | Scambio di 1.000 canali |
| FIND MAXIMUM | MAX | 182 | 4 | 19,2 | 24,9 | 24,9 | 35,3 | 32,0 | Ricerca di 1 canale |
| | | | | 2,39 ms | 3,36 ms | 3,36 ms | 4,39 ms | 3,57 ms | Ricerca di 1.000 canali |
| FIND MINIMUM | MIN | 183 | 4 | 19,2 | 25,3 | 25,3 | 35,4 | 31,9 | Ricerca di 1 canale |
| | | | | 2,39 ms | 3,33 ms | 3,33 ms | 4,39 ms | 3,58 ms | Ricerca di 1.000 canali |
| SUM | SUM | 184 | 4 | 28,2 | 38,5 | 38,3 | 49,5 | 44,1 | Aggiunta di 1 canale |
| | | | | 1,42 ms | 1,95 ms | 1,95 ms | 2,33 ms | 2,11 ms | Aggiunta di 1.000 canali |
| FRAME CHECKSUM | FCS | 180 | 4 | 20,0 | 28,3 | 28,3 | 34,8 | 31,5 | Per una tabella di 1 canali |
| | | | | 1,65 ms | 2,48 ms | 2,48 ms | 3,11 ms | 2,77 ms | Per una tabella di 1.000 canali |
| STACK SIZE READ | SNUM | 638 | 3 | 6,0 | 6,3 | --- | 12,1 | 13,7 | --- |
| STACK DATA READ | SREAD | 639 | 4 | 8,0 | 8,4 | --- | 18,1 | 20,6 | --- |
| STACK DATA OVERWRITE | SWRIT | 640 | 4 | 7,2 | 7,6 | --- | 16,9 | 18,8 | --- |
| STACK DATA INSERT | SINS | 641 | 4 | 7,8 | 9,9 | --- | 18,2 | 20,5 | --- |
| | | | | 354,0 | 434,8 | --- | 730,7 | 732,0 | Per una tabella di 1.000 canali |
| STACK DATA DELETE | SDEL | 642 | 4 | 8,6 | 10,6 | --- | 19,3 | 22,0 | --- |
| | | | | 354,0 | 436,0 | --- | 732,0 | 744,0 | Per una tabella di 1.000 canali |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-16 Istruzioni di controllo dei dati

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|----------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| PID CONTROL | PID | 190 | 4 | 436,2 | 678,2 | 678,2 | 612,0 | 552,6 | Esecuzione iniziale |
| | | | | 332,3 | 474,9 | 474,9 | 609,3 | 548,0 | Campionamento |
| | | | | 97,3 | 141,3 | 141,3 | 175,3 | 162,0 | Nessun campionamento |
| LIMIT CONTROL | LMT | 680 | 4 | 16,1 | 22,1 | 22,1 | 27,1 | 26,1 | --- |
| DEAD BAND CONTROL | BAND | 681 | 4 | 17,0 | 22,5 | 22,5 | 27,4 | 26,6 | --- |
| DEAD ZONE CONTROL | ZONE | 682 | 4 | 15,4 | 20,5 | 20,5 | 28,0 | 26,4 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|--|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|--|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| TIME-PROPORTIONAL OUTPUT (vedere la nota 2). | TPO | 685 | 4 | 10,6 | 14,8 | --- | 20,2 | 19,8 | Tempo di esecuzione OFF |
| | | | | 54,5 | 82,0 | --- | 92,7 | 85,1 | Tempo di esecuzione ON con designazione del duty-cycle o limite di uscita visualizzato |
| | | | | 61,0 | 91,9 | --- | 102,5 | 95,3 | Tempo di esecuzione ON con designazione variabile manipolata o limite di uscita attivato |
| SCALING | SCL | 194 | 4 | 37,1 | 53,0 | 56,8 | 25,0 | 32,8 | --- |
| SCALING 2 | SCL2 | 486 | 4 | 28,5 | 40,2 | 50,7 | 22,3 | 29,1 | --- |
| SCALING 3 | SCL3 | 487 | 4 | 33,4 | 47,0 | 57,7 | 25,6 | 30,0 | --- |
| AVERAGE | AVG | 195 | 4 | 36,3 | 52,6 | 53,1 | 62,9 | 59,1 | Media di un'operazione |
| | | | | 291,0 | 419,9 | 419,9 | 545,3 | 492,7 | Media di 64 operazioni |
| PID CONTROL WITH AUTOTUNING | PIDAT | 191 | 4 | 446,3 | 712,5 | --- | 765,3 | 700,0 | Esecuzione iniziale |
| | | | | 339,4 | 533,9 | --- | 620,7 | 558,0 | Campionamento |
| | | | | 100,7 | 147,1 | --- | 180,0 | 166,1 | Nessun campionamento |
| | | | | 189,2 | 281,6 | --- | 233,7 | 225,1 | Esecuzione iniziale dell'autotuning |
| | | | | 535,2 | 709,8 | --- | 575,3 | 558,2 | Autotuning durante il campionamento |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-2-17 Istruzioni di subroutine

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|--------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| SUBROUTINE CALL | SBS | 91 | 2 | 1,26 | 1,96 | 17,0 | 2,04 | 2,04 | --- |
| SUBROUTINE ENTRY | SBN | 92 | 2 | --- | --- | --- | --- | --- | --- |
| SUBROUTINE RETURN | RET | 93 | 1 | 0,86 | 1,60 | 20,60 | 1,80 | 1,80 | --- |
| MACRO | MCRO | 99 | 4 | 23,3 | 23,3 | 23,3 | 47,9 | 50,3 | --- |
| GLOBAL SUBROUTINE CALL | GSBN | 751 | 2 | --- | --- | --- | --- | --- | --- |
| GLOBAL SUBROUTINE ENTRY | GRET | 752 | 1 | 1,26 | 1,96 | --- | 2,04 | 2,04 | --- |
| GLOBAL SUBROUTINE RETURN | GSBS | 750 | 2 | 0,86 | 1,60 | --- | 1,80 | 1,80 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-18 Istruzioni di controllo degli interrupt

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| SET INTERRUPT MASK | MSKS | 690 | 3 | 25,6 | 38,4 | 39,5 | 44,7 | 42,9 | --- |
| READ INTERRUPT MASK | MSKR | 692 | 3 | 11,9 | 11,9 | 11,9 | 16,9 | 15,9 | --- |
| CLEAR INTERRUPT | CLI | 691 | 3 | 27,4 | 41,3 | 41,3 | 42,7 | 44,5 | --- |
| DISABLE INTERRUPTS | DI | 693 | 1 | 15,0 | 16,8 | 16,8 | 30,3 | 28,5 | --- |
| ENABLE INTERRUPTS | EI | 694 | 1 | 19,5 | 21,8 | 21,8 | 37,7 | 34,4 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-19 Istruzioni per contatore veloce e uscita a impulsi

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| MODE CONTROL | INI | 880 | 4 | --- | --- | --- | 77,00 | 80,4 | Avvio del confronto del contatore veloce |
| | | | | --- | --- | --- | 43,00 | 43,0 | Interruzione del confronto del contatore veloce |
| | | | | --- | --- | --- | 43,40 | 48,8 | Modifica del valore attuale dell'uscita a impulsi |
| | | | | --- | --- | --- | 51,80 | 50,8 | Modifica del valore attuale del contatore veloce |
| | | | | --- | --- | --- | 31,83 | 28,5 | Modifica del valore attuale del contatore in modalità interrupt di ingresso |
| | | | | --- | --- | --- | 45,33 | 49,8 | Interruzione dell'uscita a impulsi |
| | | | | --- | --- | --- | 36,73 | 30,5 | Interruzione dell'uscita PWM(891) |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|----------------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|---------------------------------------|----------------------|--|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| HIGH-SPEED COUNTER PV READ | PRV | 881 | 4 | --- | --- | --- | 42,40 | 43,9 | Lettura del valore attuale dell'uscita a impulsi |
| | | | | --- | --- | --- | 53,40 | 65,9 | Lettura del valore attuale del contatore veloce |
| | | | | --- | --- | --- | 33,60 | 30,5 | Lettura del valore attuale del contatore in modalità interrupt di ingresso |
| | | | | --- | --- | --- | 38,80 | 40,0 | Lettura dello stato dell'uscita a impulsi |
| | | | | --- | --- | --- | 39,30 | 66,9 | Lettura dello stato del contatore veloce |
| | | | | --- | --- | --- | 38,30 | 34,5 | Lettura dello stato di PWM(891) |
| | | | | --- | --- | --- | 117,73 | 145,7 | Lettura dei risultati del confronto a intervalli del contatore veloce |
| | | | | --- | --- | --- | 48,20 | 48,5 | Lettura della frequenza del contatore veloce 0 |
| COMPARISON TABLE LOAD | CTBL | 882 | 4 | --- | --- | --- | 238,0 | 235,0 | Registrazione della tabella dei valori di riferimento e avvio del confronto per 1 valore di riferimento |
| | | | | --- | --- | --- | 14,42 ms | 9,97 ms | Registrazione della tabella dei valori di riferimento e avvio del confronto per 48 valori di riferimento |
| | | | | --- | --- | --- | 289,0 | 276,0 | Registrazione della tabella di intervalli e avvio del confronto |
| | | | | --- | --- | --- | 198,0 | 183,0 | Solo registrazione della tabella dei valori di riferimento per 1 valore di riferimento |
| | | | | --- | --- | --- | 14,40 ms | 9,61 ms | Solo registrazione della tabella dei valori di riferimento per 48 valori di riferimento |
| | | | | --- | --- | --- | 259,0 | 239,0 | Solo registrazione della tabella di intervalli |
| COUNTER FREQUENCY CONVERT | PRV2 | 883 | 4 | --- | --- | --- | 23,03 | 22,39 | --- |
| SPEED OUTPUT | SPED | 885 | 4 | --- | --- | --- | 56,00 | 89,3 | Modalità continua |
| | | | | --- | --- | --- | 62,47 | 94,9 | Modalità indipendente |
| SET PULSES | PULS | 886 | 4 | --- | --- | --- | 26,20 | 32,9 | --- |
| PULSE OUTPUT | PLS2 | 887 | 5 | --- | --- | --- | 100,80 | 107,5 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---------------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|-----------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| ACCELERATION CONTROL | ACC | 888 | 4 | --- | --- | --- | 90,80 | 114,8 | Modalità continua |
| | | | | --- | --- | --- | 80,00 | 122,1 | Modalità indipendente |
| ORIGIN SEARCH | ORG | 889 | 3 | --- | --- | --- | 106,13 | 116,0 | Ricerca dell'origine |
| | | | | --- | --- | --- | 52,00 | 102,1 | Ritorno all'origine |
| PULSE WITH VARIABLE DUTY FACTOR | PWM | 891 | 4 | --- | --- | --- | 25,80 | 33,0 | --- |

Nota Supportata solo dalle CPU versione 2.0 o successiva.

4-2-20 Istruzioni di step

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|--|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| STEP DEFINE | STEP | 008 | 2 | 17,4 | 20,7 | 27,1 | 35,9 | 37,1 | Bit di controllo degli step impostato su ON |
| | | | | 11,8 | 13,7 | 24,4 | 13,8 | 18,3 | Bit di controllo degli step impostato su OFF |
| STEP START | SNXT | 009 | 2 | 6,6 | 7,3 | 10,0 | 12,1 | 14,0 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-21 Istruzioni per moduli di I/O di base

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|--|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| I/O REFRESH | IORF | 097 | 3 | 15,5 | 16,4 | 23,5 | 26,7 | 30,4 | Aggiornamento di 1 canale (IN) per i moduli di I/O di base |
| | | | | 17,20 | 18,40 | 25,6 | 29,7 | 35,0 | Aggiornamento di 1 canale (OUT) per i moduli di I/O di base |
| | | | | 319,9 | 320,7 | 377,6 | 291,0 | 100,0 | Aggiornamento di 60 canali (IN) per i moduli di I/O di base |
| | | | | 358,00 | 354,40 | 460,1 | 325,0 | 134,7 | Aggiornamento di 60 canale (OUT) per i moduli di I/O di base |
| 7-SEGMENT DECODER | SDEC | 078 | 4 | 6,5 | 6,9 | 14,1 | 8,1 | 15,7 | --- |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---|-----------|--------|--------------------------------|--|------------|-------|---------------------------------------|----------------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| DIGITAL SWITCH INPUT (vedere la nota 2). | DSW | 210 | 6 | 50,7 | 73,5 | --- | 77,7 | 77,6 | A 4 cifre, valore di ingresso dei dati: 0 |
| | | | | 51,5 | 73,4 | --- | 77,9 | 77,6 | A 4 cifre, valore di ingresso dei dati: F |
| | | | | 51,3 | 73,5 | --- | 83,2 | 80,0 | A 8 cifre, valore di ingresso dei dati: 0 |
| | | | | 50,7 | 73,4 | --- | 77,9 | 77,7 | A 8 cifre, valore di ingresso dei dati: F |
| TEN KEY INPUT (vedere la nota 2). | TKY | 211 | 4 | 9,7 | 13,2 | --- | 18,7 | 18,6 | Valore di ingresso dei dati: 0 |
| | | | | 10,7 | 14,8 | --- | 20,2 | 19,1 | Valore di ingresso dei dati: F |
| HEXADECIMAL KEY INPUT (vedere la nota 2). | HKY | 212 | 5 | 50,3 | 70,9 | --- | 77,3 | 78,1 | Valore di ingresso dei dati: 0 |
| | | | | 50,1 | 71,2 | --- | 76,8 | 77,3 | Valore di ingresso dei dati: F |
| MATRIX INPUT (vedere la nota 2). | MTR | 213 | 5 | 47,8 | 68,1 | --- | 76,4 | 77,7 | Valore di ingresso dei dati: 0 |
| | | | | 48,0 | 68,0 | --- | 77,7 | 76,9 | Valore di ingresso dei dati: F |
| 7-SEGMENT DISPLAY OUTPUT (vedere la nota 2). | 7SEG | 214 | 5 | 58,1 | 83,3 | --- | 89,6 | 89,9 | 4 cifre |
| | | | | 63,3 | 90,3 | --- | 98,3 | 99,2 | 8 cifre |
| INTELLIGENT I/O READ | IORD | 222 | 4 | I tempi di lettura e scrittura dipendono dal modulo di I/O speciale per cui viene eseguita l'istruzione. | | | 225,3 | 217,7 | Prima esecuzione |
| | | | | | | | 232,0 | 241,7 | Quando occupato |
| | | | | | | | 223,0 | 215,3 | Alla fine |
| INTELLIGENT I/O WRITE | IOWR | 223 | 4 | | | | 245,3 | 219,7 | Prima esecuzione |
| | | | | | | | 231,0 | 225,7 | Quando occupato |
| | | | | | | | 244,0 | 218,7 | Alla fine |
| CPU BUS I/O REFRESH | DLNK | 226 | 4 | 287,8 | 315,5 | --- | 321,3 | 458,7 | Allocazione di 1 canale |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-2-22 Istruzioni per la comunicazione seriale

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|----------------|-----------|--------|--------------------------------|-----------------------------|------------|---------|---------------------------------------|----------------------|--|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| PROTOCOL MACRO | PMCR | 260 | 5 | 100,1 | 142,1 | 276,8 | 158,4 | 206,0 | Invio di 0 canali, ricezione di 0 canali |
| | | | | 134,2 | 189,6 | 305,9 | 210,0 | 256,7 | Invio di 249 canali, ricezione di 249 canali |
| TRANSMIT | TXD | 236 | 4 | 68,5 | 98,8 | 98,8 | 109,3 | 102,9 | Invio di 1 byte |
| | | | | 734,3 | 1,10 ms | 1,10 ms | 1,23 ms | 1,16 ms | Invio di 256 byte |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|---|-----------|--------|--------------------------------|-----------------------------|------------|---------|--------------------------|---------------|----------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| RECEIVE | RXD | 235 | 4 | 89,6 | 131,1 | 131,1 | 144,0 | 132,1 | Memorizzazione di 1 byte |
| | | | | 724,2 | 1,11 ms | 1,11 ms | 1,31 ms | 1,22 ms | Memorizzazione di 256 byte |
| TRANSMIT VIA SERIAL COMMUNICATIONS UNIT | TXDU | 256 | 4 | 131,5 | 202,4 | --- | 213,4 | 208,6 | Invio di 1 byte |
| RECEIVE VIA SERIAL COMMUNICATIONS UNIT | RXDU | 255 | 4 | 131 | 200,8 | --- | 211,8 | 206,8 | Memorizzazione di 1 byte |
| CHANGE SERIAL PORT SETUP | STUP | 237 | 3 | 341,2 | 400,0 | 440,4 | 504,7 | 524,7 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-23 Istruzioni di rete

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|--|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| NETWORK SEND | SEND | 090 | 4 | 84,4 | 123,9 | 123,9 | 141,6 | 195,0 | --- |
| NETWORK RECEIVE | RECV | 098 | 4 | 85,4 | 124,7 | 124,7 | 142,3 | 196,7 | --- |
| DELIVER COMMAND | CMND | 490 | 4 | 106,8 | 136,8 | 136,8 | 167,7 | 226,7 | --- |
| EXPLICIT MESSAGE SEND (vedere la nota 2). | EXPLT | 720 | 4 | 127,6 | 190,0 | --- | 217,0 | 238,0 | --- |
| EXPLICIT GET ATTRIBUTE (vedere la nota 2). | EGATR | 721 | 4 | 123,9 | 185,0 | --- | 210,0 | 232,7 | --- |
| EXPLICIT SET ATTRIBUTE (vedere la nota 2). | ESATR | 722 | 3 | 110,0 | 164,4 | --- | 188,3 | 210,3 | --- |
| EXPLICIT WORD READ (vedere la nota 2). | ECHRD | 723 | 4 | 106,8 | 158,9 | --- | 176,3 | 220,3 | --- |
| EXPLICIT WORD WRITE (vedere la nota 2). | ECHWR | 724 | 4 | 106,0 | 158,3 | --- | 175,7 | 205,3 | --- |

- Nota**
1. Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.
 2. Supportata solo dalle CPU versione 2.0 o successiva.

4-2-24 Istruzioni per la memoria dei file

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|-----------------|-----------|--------|--------------------------------|-----------------------------------|------------|---------|--------------------------|---------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| READ DATA FILE | FREAD | 700 | 5 | 391,4 | 632,4 | 684,1 | 657,3 | 641,3 | Directory di 2 caratteri + nome di file in formato binario |
| | | | | 836,1 | 1,33 ms | 1,35 ms | 1,45 ms | 1,16 ms | Directory di 73 caratteri + nome di file in formato binario |
| WRITE DATA FILE | FWRIT | 701 | 5 | 387,8 | 627,0 | 684,7 | 650,7 | 637,3 | Directory di 2 caratteri + nome di file in formato binario |
| | | | | 833,3 | 1,32 ms | 1,36 ms | 1,44 ms | 1,16 ms | Directory di 73 caratteri + nome di file in formato binario |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-25 Istruzioni di visualizzazione

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|-----------------|-----------|--------|--------------------------------|-----------------------------------|------------|-------|--------------------------|---------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| DISPLAY MESSAGE | MSG | 046 | 3 | 10,1 | 14,2 | 14,3 | 16,8 | 17,3 | Visualizzazione del messaggio |
| | | | | 8,4 | 11,3 | 11,3 | 14,7 | 14,7 | Eliminazione del messaggio visualizzato |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-26 Istruzioni per l'orologio

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|-------------------|-----------|--------|--------------------------------|-----------------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| CALENDAR ADD | CADD | 730 | 4 | 38,3 | 201,9 | 209,5 | 217,0 | 194,0 | --- |
| CALENDAR SUBTRACT | CSUB | 731 | 4 | 38,6 | 170,4 | 184,1 | 184,7 | 167,0 | --- |
| HOURS TO SECONDS | SEC | 065 | 3 | 21,4 | 29,3 | 35,8 | 36,1 | 35,4 | --- |
| SECONDS TO HOURS | HMS | 066 | 3 | 22,2 | 30,9 | 42,1 | 45,1 | 45,7 | --- |
| CLOCK ADJUSTMENT | DATE | 735 | 2 | 216,0 | 251,5 | 120,0 | 118,7 | 128,3 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-27 Istruzioni di debug

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-----------------------------|-----------|--------|--------------------------------|-----------------------------|------------|---------|---------------------------------------|----------------------|------------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| TRACE MEMORY SAMPLING | TRSM | 045 | 1 | 80,4 | 120,0 | 120,0 | 207,0 | 218,3 | Campionamento di 1 bit e 0 canali |
| | | | | 848,1 | 1,06 ms | 1,06 ms | 1,16 ms | 1,10 ms | Campionamento di 31 bit e 6 canali |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-28 Istruzioni di diagnostica per malfunzionamento

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-------------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|---------------------------------------|----------------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| FAILURE ALARM | FAL | 006 | 3 | 15,4 | 16,7 | 16,7 | 26,1 | 24,47 | Registrazione degli errori |
| | | | | 179,8 | 244,8 | 244,8 | 294,0 | 264,0 | Eliminazione degli errori in ordine di priorità |
| | | | | 432,4 | 657,1 | 657,1 | 853,3 | 807,3 | Eliminazione di tutti gli errori |
| | | | | 161,5 | 219,4 | 219,4 | 265,7 | 233,0 | Eliminazione di singoli errori |
| SEVERE FAILURE ALARM | FALS | 007 | 3 | --- | --- | --- | --- | --- | --- |
| FAILURE POINT DETECTION | FPD | 269 | 4 | 140,9 | 202,3 | 202,3 | 220,7 | 250,0 | All'esecuzione |
| | | | | 163,4 | 217,6 | 217,6 | 250,3 | 264,3 | La prima volta |
| | | | | 185,2 | 268,9 | 268,9 | 220,7 | 321,7 | All'esecuzione |
| | | | | 207,5 | 283,6 | 283,6 | 320,7 | 336,0 | La prima volta |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-29 Altre istruzioni

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|------------------------------|-----------|--------|--------------------------------|-----------------------------------|------------|-------|---------------------------------------|----------------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| SET CARRY | STC | 040 | 1 | 0,06 | 0,06 | 0,12 | 0,15 | 0,15 | --- |
| CLEAR CARRY | CLC | 041 | 1 | 0,06 | 0,06 | 0,12 | 0,15 | 0,15 | --- |
| SELECT EM BANK | EMBC | 281 | 2 | 14,0 | 15,1 | 15,1 | --- | --- | --- |
| EXTEND MAXIMUM CYCLE TIME | WDT | 094 | 2 | 15,0 | 19,7 | 19,7 | 23,6 | 22,0 | --- |
| SAVE CONDITION FLAGS | CCS | 282 | 1 | 8,6 | 12,5 | --- | 14,2 | 12,9 | --- |
| LOAD CONDITION FLAGS | CCL | 283 | 1 | 9,8 | 13,9 | --- | 16,3 | 15,7 | --- |
| CONVERT ADDRESS FROM CV | FRMCV | 284 | 3 | 13,6 | 19,9 | --- | 23,1 | 31,8 | --- |
| CONVERT ADDRESS TO CV | TOCV | 285 | 3 | 11,9 | 17,2 | --- | 22,5 | 31,4 | --- |
| DISABLE PERIPHERAL SERVICING | IOSP | 287 | --- | 13,9 | 19,8 | --- | 21,5 | 21,5 | --- |
| ENABLE PERIPHERAL SERVICING | IORS | 288 | --- | 63,6 | 92,3 | --- | 22,2 | 22,2 | --- |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-30 Istruzioni di programmazione a blocchi

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μ s) | | | | | Condizioni |
|------------------------|---------------------------------|--------|--------------------------------|-----------------------------------|------------|-------|---------------------------------------|----------------------|---------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/ 21 | CJ1M CPU11/ 21 | |
| BLOCK PROGRAM BEGIN | BPRG | 096 | 2 | 12,1 | 13,0 | 13,0 | 27,5 | 30,4 | --- |
| BLOCK PROGRAM END | BEND | 801 | 1 | 9,6 | 12,3 | 13,1 | 23,2 | 27,1 | --- |
| BLOCK PROGRAM PAUSE | BPPS | 811 | 2 | 10,6 | 12,3 | 14,9 | 16,0 | 21,7 | --- |
| BLOCK PROGRAM RESTART | BPRS | 812 | 2 | 5,1 | 5,6 | 8,3 | 9,0 | 10,2 | --- |
| CONDITIONAL BLOCK EXIT | EXIT (condizione di esecuzione) | 806 | 1 | 10,0 | 11,3 | 12,9 | 23,8 | 26,0 | Condizione EXIT soddisfatta |
| | | | | 4,0 | 4,9 | 7,3 | 7,2 | 8,4 | Condizione EXIT non soddisfatta |
| CONDITIONAL BLOCK EXIT | EXIT (indirizzo bit) | 806 | 2 | 6,8 | 13,5 | 16,3 | 28,4 | 30,6 | Condizione EXIT soddisfatta |
| | | | | 4,7 | 7,2 | 10,7 | 11,4 | 13,1 | Condizione EXIT non soddisfatta |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|------------------------------|---|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|---------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| CONDITIONAL BLOCK EXIT (NOT) | EXIT NOT (indirizzo bit) | 806 | 2 | 12,4 | 14,0 | 16,8 | 28,4 | 31,2 | Condizione EXIT soddisfatta |
| | | | | 7,1 | 7,6 | 11,2 | 11,8 | 13,5 | Condizione EXIT non soddisfatta |
| Diramazione | IF (condizione di esecuzione) | 802 | 1 | 4,6 | 4,8 | 7,2 | 6,8 | 8,5 | Condizione IF vera |
| | | | | 6,7 | 7,3 | 10,9 | 12,2 | 13,9 | Condizione IF falsa |
| Diramazione | IF (numero condizione intermedia) | 802 | 2 | 6,8 | 7,2 | 10,4 | 11,0 | 12,7 | Condizione IF vera |
| | | | | 9,0 | 9,6 | 14,2 | 16,5 | 18,5 | Condizione IF falsa |
| Diramazione (NOT) | IF NOT (numero condizione intermedia) | 802 | 2 | 7,1 | 7,6 | 10,9 | 11,5 | 13,1 | Condizione IF vera |
| | | | | 9,2 | 10,1 | 14,7 | 16,8 | 18,9 | Condizione IF falsa |
| Diramazione | ELSE | 803 | 1 | 6,2 | 6,7 | 9,9 | 11,4 | 12,6 | Condizione IF vera |
| | | | | 6,8 | 7,7 | 11,2 | 13,4 | 15,0 | Condizione IF falsa |
| Diramazione | IEND | 804 | 1 | 6,9 | 7,7 | 11,0 | 13,5 | 15,4 | Condizione IF vera |
| | | | | 4,4 | 4,6 | 7,0 | 6,93 | 8,1 | Condizione IF falsa |
| ONE CYCLE AND WAIT | WAIT (condizione di esecuzione) | 805 | 1 | 12,6 | 13,7 | 16,7 | 28,6 | 34,0 | Condizione WAIT soddisfatta |
| | | | | 3,9 | 4,1 | 6,3 | 5,6 | 6,9 | Condizione WAIT non soddisfatta |
| ONE CYCLE AND WAIT | WAIT (numero condizione intermedia) | 805 | 2 | 12,0 | 13,4 | 16,5 | 27,2 | 30,0 | Condizione WAIT soddisfatta |
| | | | | 6,1 | 6,5 | 9,6 | 10,0 | 11,4 | Condizione WAIT non soddisfatta |
| ONE CYCLE AND WAIT (NOT) | WAIT NOT (numero condizione intermedia) | 805 | 2 | 12,2 | 13,8 | 17,0 | 27,8 | 30,6 | Condizione WAIT soddisfatta |
| | | | | 6,4 | 6,9 | 10,1 | 10,5 | 11,8 | Condizione WAIT non soddisfatta |
| COUNTER WAIT | CNTW | 814 | 4 | 17,9 | 22,6 | 27,4 | 41,0 | 43,5 | Prima esecuzione |
| | | | | 19,1 | 23,9 | 28,7 | 42,9 | 45,7 | Esecuzione normale |
| | CNTWX | 818 | 4 | 17,9 | 22,6 | --- | 41,0 | 43,5 | Prima esecuzione |
| | | | | 19,1 | 23,9 | --- | 42,9 | 45,7 | Esecuzione normale |
| HIGH-SPEED TIMER WAIT | TMHW | 815 | 3 | 25,8 | 27,9 | 34,1 | 47,9 | 53,7 | Prima esecuzione |
| | | | | 20,6 | 22,7 | 28,9 | 40,9 | 46,2 | Esecuzione normale |
| | TMHWX | 817 | 3 | 25,8 | 27,9 | --- | 47,9 | 53,7 | Prima esecuzione |
| | | | | 20,6 | 22,7 | --- | 40,9 | 46,2 | Esecuzione normale |
| Controllo del ciclo | LOOP | 809 | 1 | 7,9 | 9,1 | 12,3 | 15,6 | 17,6 | --- |
| Controllo del ciclo | LEND (condizione di esecuzione) | 810 | 1 | 7,7 | 8,4 | 10,9 | 13,5 | 15,5 | Condizione LEND soddisfatta |
| | | | | 6,8 | 8,0 | 9,8 | 17,5 | 19,8 | Condizione LEND non soddisfatta |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|---------------------|---|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|---------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| Controllo del ciclo | LEND (numero condizione intermedia) | 810 | 2 | 9,9 | 10,7 | 14,4 | 17,5 | 19,9 | Condizione LEND soddisfatta |
| | | | | 8,9 | 10,3 | 13,0 | 21,6 | 24,5 | Condizione LEND non soddisfatta |
| Controllo del ciclo | LEND NOT (numero condizione intermedia) | 810 | 2 | 10,2 | 11,2 | 14,8 | 21,9 | 24,9 | Condizione LEND soddisfatta |
| | | | | 9,3 | 10,8 | 13,5 | 17,8 | 20,4 | Condizione LEND non soddisfatta |
| TIMER WAIT | TIMW | 813 | 3 | 22,3 | 25,2 | 33,1 | 47,4 | 52,0 | Impostazione predefinita |
| | | | | 24,9 | 27,8 | 35,7 | 46,2 | 53,4 | Esecuzione normale |
| | TIMWX | 816 | 3 | 22,3 | 25,2 | 33,1 | 47,4 | 52,0 | Impostazione predefinita |
| | | | | 24,9 | 27,8 | 35,7 | 46,2 | 53,4 | Esecuzione normale |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-31 Istruzioni di elaborazione delle stringhe di testo

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|--------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| MOV STRING | MOV\$ | 664 | 3 | 45,6 | 66,0 | 84,3 | 79,3 | 72,7 | Trasferimento di 1 carattere |
| CONCATENATE STRING | +\$ | 656 | 4 | 86,5 | 126,0 | 167,8 | 152,0 | 137,0 | 1 carattere + 1 carattere |
| GET STRING LEFT | LEFT\$ | 652 | 4 | 53,0 | 77,4 | 94,3 | 93,6 | 84,8 | Recupero di 1 carattere da 2 caratteri |
| GET STRING RIGHT | RGHT\$ | 653 | 4 | 52,2 | 76,3 | 94,2 | 92,1 | 83,3 | Recupero di 1 carattere da 2 caratteri |
| GET STRING MIDDLE | MID\$ | 654 | 5 | 56,5 | 84,6 | 230,2 | 93,7 | 84,0 | Recupero di 1 carattere da 3 caratteri |
| FIND IN STRING | FIND\$ | 660 | 4 | 51,4 | 77,5 | 94,1 | 89,1 | 96,7 | Ricerca di 1 carattere in 2 caratteri |
| STRING LENGTH | LEN\$ | 650 | 3 | 19,8 | 28,9 | 33,4 | 33,8 | 30,1 | Rilevamento di 1 carattere |
| REPLACE IN STRING | RPLC\$ | 661 | 6 | 175,1 | 258,7 | 479,5 | 300,7 | 267,7 | Sostituzione del primo di 2 caratteri con 1 carattere |
| DELETE STRING | DEL\$ | 658 | 5 | 63,4 | 94,2 | 244,6 | 11,3 | 99,3 | Eliminazione del carattere iniziale di 2 caratteri |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|-------------------------------------|-------------------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| EXCHANGE STRING | XCHG\$ | 665 | 3 | 60,6 | 87,2 | 99,0 | 105,2 | 95,3 | Scambio di 1 carattere con 1 carattere |
| CLEAR STRING | CLR\$ | 666 | 2 | 23,8 | 36,0 | 37,8 | 42,0 | 36,8 | Cancellazione di 1 carattere |
| INSERT INTO STRING | INS\$ | 657 | 5 | 136,5 | 200,6 | 428,9 | 204,0 | 208,0 | Inserimento di 1 carattere dopo il primo di 2 caratteri |
| Istruzioni di confronto di stringhe | LD, AND, OR += \$ | 670 | 4 | 48,5 | 69,8 | 86,2 | 79,9 | 68,5 | Confronto di 1 carattere con 1 carattere |
| | LD, AND, OR +<>\$ | 671 | | | | | | | |
| | LD, AND, OR +<\$ | 672 | | | | | | | |
| | LD, AND, OR +>\$ | 674 | | | | | | | |
| | LD, AND, OR +>=\$ | 675 | | | | | | | |

Nota Quando si utilizza un operando a doppia lunghezza, aggiungere 1 al valore riportato nella colonna della lunghezza nella tabella seguente.

4-2-32 Istruzioni di controllo dei task

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| TASK ON | TKON | 820 | 2 | 19,5 | 26,3 | 26,3 | 33,1 | 32,5 | --- |
| TASK OFF | TKOF | 821 | 2 | 13,3 | 19,0 | 26,3 | 19,7 | 20,2 | --- |

4-2-33 Istruzioni di conversione del modello (solo CPU versione 3.0 o successiva).

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (μs) | | | | | Condizioni |
|------------------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|-------------------------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| BLOCK TRANSFER | XFERC | 565 | 4 | 6,4 | 6,5 | --- | 33,1 | 31,1 | Trasferimento di 1 canale |
| | | | | 481,6 | 791,6 | --- | 3.056,1 | 2.821,1 | Trasferimento di 1.000 canali |
| SINGLE WORD DISTRIBUTE | DISTC | 566 | 4 | 3,4 | 3,5 | --- | 19 | 18,1 | Distribuzione dei dati |
| | | | | 5,9 | 7,3 | --- | 39,5 | 38,5 | Operazione di stack |

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|--------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|---|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| DATA COLLECT | COLLC | 567 | 4 | 3,5 | 3,85 | --- | 24,9 | 29,7 | Distribuzione dei dati |
| | | | | 8 | 9,1 | --- | 22,1 | 25,3 | Operazione di stack |
| | | | | 8,3 | 9,6 | --- | 25,5 | 31 | Operazione di stack Lettura FIFO di 1 canale |
| | | | | 2.052,3 | 2.097,5 | --- | 8.310,1 | 7.821,1 | Operazione di stack Lettura FIFO di 1.000 canali |
| MOVE BIT | MOVBC | 568 | 4 | 4,5 | 4,88 | --- | 28,1 | 22,1 | --- |
| BIT COUNTER | BCNTC | 621 | 4 | 4,9 | 5 | --- | 30,6 | 28,8 | Conteggio di 1 canale |
| | | | | 1.252,4 | 1.284,4 | --- | 5.814,1 | 5.223,8 | Conteggio di 1.000 canali |

4-2-34 Istruzioni speciali per blocchi funzione (solo CPU versione 3.0 o successiva)

| Istruzione | Mnemonico | Codice | Lunghezza (step) (vedere nota) | Tempo di esecuzione ON (µs) | | | | | Condizioni |
|-----------------|-----------|--------|--------------------------------|-----------------------------|------------|-------|--------------------------|---------------|------------|
| | | | | CPU6□ H | CPU4□ H | CPU4□ | CJ1M esclusa la CPU11/21 | CJ1M CPU11/21 | |
| GET VARIABLE ID | GETID | 286 | 4 | 14 | 22,2 | --- | 23,4 | 21,3 | |

4-2-35 Numero di step di programma del blocco funzione (CPU versione 3.0 o successiva)

Utilizzare la seguente equazione per calcolare il numero di step di programma quando si creano le definizioni di blocchi funzione e le istanze vengono copiate nel programma utente mediante CPU della serie CS/CJ versione 3.0 o successiva.

| |
|---|
| <p>Numero di step = Numero di istanze × (dimensione della porzione di chiamata m + dimensione della porzione di trasferimento dei parametri degli I/O n × numero di parametri) + numero di step di istruzioni nella definizione del blocco funzione p (vedere nota).</p> |
|---|

Nota Il numero di step di istruzioni nella definizione del blocco funzione (p) rimane inalterata per le istanze successive quando la stessa definizione di blocco funzione viene copiata in più punti, ossia per più istanze. Di conseguenza, nella precedente equazione il numero di istanze non viene moltiplicato per il numero di step di istruzioni nella definizione del blocco funzione (p).

| Contenuto | | | CPU della serie CS/CJ versione 3.0 o successiva |
|-----------|---|--|---|
| m | Porzione di chiamata | | 57 step |
| n | Porzione di trasferimento dei parametri degli I/O Il tipo dei dati è indicato tra parentesi. | Variabile di I/O a 1 bit (BOOL) | 6 step |
| | | Variabile di I/O a 1 canale (INT, UINT, WORD) | 6 step |
| | | Variabile di I/O a 2 canali (DINT, UDINT, DWORD, REAL) | 6 step |
| | | Variabile di I/O a 4 canali (LINT, ULINT, LWORD, LREAL) | 12 step |
| p | Numero di step di istruzioni nella definizione del blocco funzione | Numero totale di step di istruzioni (uguale al programma utente standard) + 27 step. | |

Esempio:

Variabili di ingresso con tipo di dati a 1 canale (INT): 5

Variabili di uscita con tipo di dati a 1 canale (INT): 5

Sezione di definizione del blocco funzione: 100 step

Numero di step per 1 istanza = 57 + (5 + 5) × 6 step + 100 step + 27 step = 244 step

4-2-36 Linee guida per la conversione delle capacità di programma rispetto ai PLC OMRON precedenti

Nella seguente tabella sono fornite le linee guida per la conversione della capacità del programma (unità di misura: canali) dei PLC OMRON precedenti (PLC SYSMAC C200HX/HG/HE, CVM1 o della serie CV) nella capacità del programma (unità di misura: step) dei PLC della serie CJ.

Aggiungere il seguente valore (n) alla capacità di programma (unità di misura: canali) dei PLC precedenti per ogni istruzione per ottenere la capacità di programma (unità di misura: step) dei PLC della serie CJ.

| Step per la serie CJ = "a" (canali) di PLC precedenti + n | | | |
|---|---|---|--|
| Istruzioni | Variazioni | Valore di n nella conversione da C200HX/HG/HE alla serie CJ | Valore di n nella conversione da PLC serie CV o CVM1 alla serie CJ |
| Istruzioni di base | Nessuna | OUT, SET, RSET o KEEP(011): -1 Altre istruzioni: 0 | 0 |
| | Differenziazione sul fronte di salita | Nessuna | +1 |
| | Aggiornamento immediato | Nessuno | 0 |
| | Differenziazione sul fronte di salita e aggiornamento immediato | Nessuno | +2 |
| Istruzioni speciali | Nessuna | 0 | -1 |
| | Differenziazione sul fronte di salita | +1 | 0 |
| | Aggiornamento immediato | Nessuno | +3 |
| | Differenziazione sul fronte di salita e aggiornamento immediato | Nessuno | +4 |

Se ad esempio si utilizza OUT con un indirizzo compreso tra CIO 000000 e CIO 25515, la capacità di programma dei PLC precedenti è di 2 canali per istruzione, mentre quella dei PLC della serie CJ è pari a 1 (2 - 1) step per istruzione.

Se ad esempio se si utilizza !MOV (istruzione MOVE con aggiornamento immediato), la capacità di programma di un PLC della serie CV è di 4 canali per istruzione, mentre quella dei PLC della serie CJ è pari a 7 (4 + 3) step.

4-2-37 Tempo di esecuzione delle istanze di blocchi funzione (CPU versione 3.0 o successiva)

Utilizzare la seguente equazione per calcolare l'effetto dell'esecuzione delle istanze sul tempo di ciclo quando si creano definizioni di blocchi funzione e le istanze vengono copiate nel programma utente mediante CPU della serie CS/CJ versione 3.0 o successiva.

Effetto dell'esecuzione delle istanze sul tempo di ciclo
 = Tempo di avvio (A)
 + tempo di elaborazione del trasferimento dei parametri degli I/O (B)
 + tempo di esecuzione delle istruzioni nella definizione del blocco funzione (C)

Nella seguente tabella è riportata la durata relativa ad A, B e C.

| Operazione | | | Modello di CPU | | |
|------------|---|---|----------------------------|----------------------------|------------|
| | | | CS1H-CPU6□H CJ1H-CPU6□H | CS1G-CPU4□H CJ1G-CPU4□H | CJ1M-CPU□□ |
| A | Tempo di avvio | Tempo di avvio escluso il trasferimento dei parametri degli I/O | 6.8 μs | 8.8 μs | 15.0 μs |
| B | Tempo di elaborazione del trasferimento dei parametri degli I/O Il tipo dei dati è indicato tra parentesi. | Variabile di I/O a 1 bit (BOOL) | 0.4 μs | 0.7 μs | 1.0 μs |
| | | Variabile di I/O a 1 canale (INT, UINT, WORD) | 0.3 μs | 0.6 μs | 0.8 μs |
| | | Variabile di I/O a 2 canali (DINT, UDINT, DWORD, REAL) | 0.5 μs | 0.8 μs | 1.1 μs |
| | | Variabile di I/O a 4 canali (LINT, ULINT, LWORD, LREAL) | 1.0 μs | 1.6 μs | 2.2 μs |
| C | Tempo di esecuzione delle istruzioni di definizione dei blocchi funzione | Tempo totale di elaborazione delle istruzioni (uguale al programma utente standard) | | | |

Esempio: CS1H-CPU63H

Variabili di ingresso con tipo di dati a 1 canale (INT): 3

Variabili di uscita con tipo di dati a 1 canale (INT): 2

Tempo totale di elaborazione delle istruzioni nella sezione di definizione del blocco funzione: 10 μs

Tempo di esecuzione per 1 istanza = 6,8 μs + (3 + 2) × 0,3 μs + 10 μs = 18,3 μs

Nota Il tempo di esecuzione aumenta in base al numero di istanze se la stessa definizione di blocco funzione viene copiata in più punti.

Appendice A

Tabella dei codici ASCII

ASCII

| | | Quattro bit più a sinistra | | | | | | | | | | | | | | | |
|--------------------------|---|----------------------------|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Quattro bit più a destra | 0 | | | Sp | 0 | @ | P | ' | p | | | | | 一 | タ | ミ | |
| | 1 | | | ! | 1 | A | Q | a | q | | | | | 。 | ア | チ | ム |
| | 2 | | | " | 2 | B | R | b | r | | | | | 「 | イ | ツ | メ |
| | 3 | | | # | 3 | C | S | c | s | | | | | 」 | ウ | テ | モ |
| | 4 | | | \$ | 4 | D | T | d | t | | | | | 、 | エ | ト | ヤ |
| | 5 | | | % | 5 | E | U | e | u | | | | | ・ | オ | ナ | ユ |
| | 6 | | | & | 6 | F | V | f | v | | | | | ヲ | カ | ニ | ヨ |
| | 7 | | | ' | 7 | G | W | g | w | | | | | ア | キ | ヌ | ラ |
| | 8 | | | (| 8 | H | X | h | x | | | | | イ | ク | ネ | リ |
| | 9 | | |) | 9 | I | Y | i | y | | | | | ウ | ケ | ノ | ル |
| | A | | | * | : | J | Z | j | z | | | | | エ | コ | ハ | レ |
| | B | | | + | ; | K | [| k | { | | | | | オ | サ | ヒ | ロ |
| | C | | | , | < | L | ¥ | l | | | | | | ヤ | シ | フ | ワ |
| | D | | | - | = | M |] | m | } | | | | | ユ | ス | ヘ | ン |
| | E | | | . | > | N | ^ | n | ~ | | | | | ヨ | セ | ホ | ° |
| | F | | | / | ? | O | _ | o | | | | | | ッ | ソ | マ | |

Index

A

- aggiornamento
 - con IORF(097), 885
 - istruzioni di aggiornamento differenziato, 169
 - istruzioni di aggiornamento immediato, 169
- ambiente operativo
 - precauzioni, xxvi
- applicazioni
 - precauzioni, xxvi
- Area DM
 - utilizzo di bit dell'area DM in condizioni di esecuzione, 174
- Area EM
 - utilizzo di bit dell'area EM in condizioni di esecuzione, 174
- ASCII
 - conversione da ASCII in esadecimale, 490
 - conversione da dati a virgola mobile, 604
 - conversione da esadecimale in ASCII, 486
 - conversione in dati a virgola mobile, 609
 - elaborazione delle stringhe di testo, 1158
 - tabella dei caratteri, 10

B

- bit
 - impostazione e ripristino, 192
- bit a autoritenuta
 - utilizzo di KEEP(011), 181
- bit di controllo
 - Bit di avvio campionamento, 1077
 - Bit di avvio registrazione, 1077

C

- capacità del programma, 2
- checksum
 - calcolo, 700
- ciclo
 - BREAK(514), 232
 - FOR(512) e NEXT(513), 229
- codici funzione
 - istruzioni elencate in base ai codici funzione, 125
- comandi FINS, 1003
 - invio di comandi alla CPU locale, 1010
- comunicazione
 - descrizione della comunicazione seriale, 926
 - istruzioni per la comunicazione seriale, 87
 - ricezione dalla porta RS-232C, 944
 - tempi di esecuzione delle istruzioni, 1243, 1274
 - tempi di esecuzione delle istruzioni di rete, 1243, 1275
 - trasmissione dalla porta RS-232C, 937
- comunicazione seriale
 - descrizione, 926

- comunicazioni
 - istruzioni per la comunicazione seriale, 926–972
- condizione di esecuzione
 - invio in corso, 195
- confronto, 837
- confronto a intervalli, 310, 313, 840
- contatori, 233–274
 - applicazioni di esempio, 269
 - contatore reversibile, 263
 - reimpostazione con CNR(545), 267
 - tempi di esecuzione, 1225, 1254
- conteggio veloce
 - lettura del valore attuale, 827, 833
- Controllo PID, 720, 731, 1113, 1117, 1121, 1123
- conversione
 - Vedere anche* dati, conversione
- conversione degli indirizzi di memoria, 1113, 1117
- CPU CJ1-H, 3
- CPU CJ1M, 3
- CPU CS1, 2
- CPU CS1-H, 2

D

- dati
 - conversione
 - radianti e gradi, 578–579, 634, 636
 - ricerca, 685
 - dati a virgola mobile, 559, 614
 - confronto, 600
 - conversione, 613
 - conversione in ASCII, 604, 609
 - divisione, 552
 - esponenti, 595, 651
 - formato, 614
 - funzioni matematiche, 613
 - funzioni trigonometriche, 613
 - istruzioni a virgola mobile in doppia precisione, 67
 - istruzioni matematiche a virgola mobile, 63, 558–600, 613–657
 - logaritmi, 597, 653
 - radice quadrata, 649
 - radici quadrate, 593
- dati BCD, 11
- dati binari con segno, 11
 - rimozione del segno, 476
- dati binari senza segno, 11
- debug
 - istruzioni di debug, 93, 1075–1078
 - istruzioni di diagnostica per malfunzionamento, 94, 1079–1104
- decimale a virgola mobile, 12
- diagrammi ladder
 - controllo dello stato del bit

- utilizzo di DIFU(013) e DIFD(014), 184–186
- utilizzo di KEEP(011), 180–184
- utilizzo di SET e RSET, 187–189
- utilizzo di SETA(530) e RSTA(531), 189–192, 195
- direttive dell'Unione Europea, xxx
- display a 7 segmenti
 - conversione dei dati, 888
- duty-cycle
 - impulsi con duty-cycle variabile, 865

E

- elaborazione dell'interrupt di spegnimento
 - disattivazione, 814
- elaborazione dello stack
 - tempi di esecuzione, 1238, 1268
- errori
 - codici
 - programmazione, 1079, 1087
 - errori di accesso, 13
 - errori di elaborazione delle istruzioni, 13
 - errori di istruzione illegale, 13
 - errori di programma, 13
 - errori programmati dall'utente, 1079, 1087
 - fatali
 - cancellazione, 1087
 - generare, 1087
 - flag di errore di comunicazione, 957, 965, 982
 - messaggi
 - programmazione, 1058
 - messaggi di programmazione, 1058
 - non fatali
 - cancellazione, 1079
 - generare, 1079
 - overflow UM, 13
- errori di funzionamento fatali
 - generare e cancellare, 1087
- errori di funzionamento non fatali
 - generare e cancellare, 1079
- errori di sistema
 - impedire la memorizzazione nel log degli errori, 1081
- esponenti, 595, 651

F

- file di dati
 - lettura, 1045
 - scrittura, 1052
- flag
 - CY
 - cancellazione, 1105
 - flag AER, 13
 - flag di errore di istruzione illegale, 13
 - flag di errore di overflow UM, 13
 - Flag di esecuzione registrazione, 1077

- flag ER, 13
- Flag monitor trigger registrazione, 1077
- Flag registrazione completata, 1077
- Flag di condizione
 - caricamento dello stato, 1112
 - salvataggio stato, 1110
- formati dei dati, 11
- formato dei dati
 - dati a virgola mobile, 614
- frame checksum
 - calcolo, 700
- funzioni trigonometriche
 - arcocoseno, 589, 645
 - arcoseno, 587, 643
 - arcotangente, 591, 647
 - conversione da gradi a radianti, 578, 634
 - conversione da radianti a gradi, 579, 636
 - coseno, 583, 639
 - seno, 581, 637
 - tangente, 585, 641

G

- gestione delle periferiche
 - abilitazione, 1123
 - disattivazione, 1121
- gradi
 - conversione da gradi a radianti, 578, 634

I

- impostazione dei bit, 192
- indirizzamento
 - numeri temporizzatore, 272
 - numero contatore, 272
 - operandi, 6
 - Vedere anche* registri indice
- indirizzo di memoria del PLC
 - vedere anche* indirizzo di memoria I/O interna
- indirizzo di memoria I/O
 - vedere anche* indirizzo di memoria I/O interna
- indirizzo di memoria I/O interna
 - impostazione dell'indirizzo del valore attuale di un temporizzatore/contatore in un registro indice, 342
 - impostazione dell'indirizzo di canale/bit in un registro indice, 340
- installazione
 - precauzioni, xxvi
- interblocchi, 201–219
- interrupt
 - abilitazione completa, 816
 - cancellazione, 810
 - disattivazione completa, 814
 - lettura dello stato della maschera, 805

- mascheramento, 798
- programmato
 - lettura intervallo, 805
 - riepilogo del controllo degli interrupt, 818
- interrupt di spegnimento, 815–816
- istruzioni, 141–274
 - classificato in base alla funzione, 16
 - controllo delle condizioni di esecuzione
 - UP(521) e DOWN(522), 173
 - controllo di contatori veloci e uscite a treno di impulsi, 823
 - elencate in base al codice funzione, 125
 - in ordine alfabetico, 108
 - istruzioni di confronto, 37, 275–310
 - istruzioni di confronto di ingresso, 275–281, 600, 657
 - istruzioni di confronto di stringhe, 1187–1192
 - istruzioni di contatore, 33, 233–274
 - istruzioni di controllo degli interrupt, 80, 798–822
 - istruzioni di controllo dei dati, 75, 720–772
 - istruzioni di controllo dei task, 105–107, 1192–1199
 - istruzioni di controllo sequenza, 29, 197–233
 - istruzioni di conversione, 54, 465–510
 - istruzioni di debug, 93, 1075–1078
 - istruzioni di decremento, 48, 393–408
 - istruzioni di diagnostica per malfunzionamento, 94, 1079–1104
 - istruzioni di elaborazione dei dati delle tabelle, 67, 71, 660–704, 1237, 1267
 - istruzioni di elaborazione delle stringhe di testo, 102, 1158–1192
 - istruzioni di incremento, 48, 393–408
 - istruzioni di ingresso sequenza, 24, 153–177
 - istruzioni di programmazione a blocchi, 96, 1124–1157
 - istruzioni di rete, 88, 973–1013
 - istruzioni di scorrimento dei dati, 44, 344–392
 - istruzioni di spostamento dei dati, 41, 315
 - istruzioni di step, 84, 867–884
 - istruzioni di subroutine, 79, 773–797
 - istruzioni di temporizzatore, 33, 233–274
 - istruzioni di uscita a treno di impulsi, 823
 - istruzioni di uscita sequenza, 26, 177–191
 - istruzioni di visualizzazione, 92, 1058–1285
 - istruzioni differenziate, 3
 - istruzioni logiche, 60, 517–534
 - istruzioni matematiche a virgola mobile, 63, 558–600, 613–657
 - istruzioni matematiche con simboli, 49, 409–465
 - istruzioni matematiche speciali, 62, 534–1214
 - istruzioni per contatore veloce, 823
 - istruzioni per l'orologio, 92, 1061–1121
 - istruzioni per la comunicazione seriale, 87, 926–972
 - istruzioni per la memoria dei file, 91, 1042–1045
 - istruzioni per moduli di I/O di base, 84, 885–921
 - numero di step, 1219
 - step per istruzione, 1221, 1250
 - tempi di esecuzione, 1221, 1250
 - tempi di esecuzione delle istruzioni, 1219
 - variazioni di istruzione, 4
- Istruzioni a virgola mobile in doppia precisione, 613
- istruzioni di checksum, 660
- istruzioni di confronto
 - tempi di esecuzione, 1227, 1255, 1257
- istruzioni di confronto a virgola mobile in doppia precisione, 657
- Istruzioni di confronto di ingresso a virgola mobile in singola precisione, 600
- istruzioni di controllo degli interrupt
 - tempi di esecuzione, 1241, 1271
- istruzioni di controllo dei dati
 - tempi di esecuzione, 1240, 1269
- istruzioni di controllo dei task
 - tempi di esecuzione, 1249, 1281
- istruzioni di controllo sequenza
 - tempi di esecuzione, 1224, 1253
- istruzioni di conversione
 - tempi di esecuzione, 1234
- istruzioni di debug
 - tempi di esecuzione, 1245, 1277
- istruzioni di decremento
 - tempi di esecuzione, 1230, 1260
- istruzioni di diagnostica per malfunzionamento
 - tempi di esecuzione, 1245, 1277
- istruzioni di incremento
 - tempi di esecuzione, 1230, 1260
- istruzioni di ingresso
 - tempi di esecuzione, 1222, 1251
- istruzioni di rete
 - tempi di esecuzione, 1243, 1275
- istruzioni di ricerca, 660
- istruzioni di scorrimento dei dati
 - tempi di esecuzione, 1229, 1258
- istruzioni di spostamento dei dati
 - tempi di esecuzione, 1228, 1258
- istruzioni di stack, 660
 - tempi di esecuzione, 1238, 1268
- istruzioni di step
 - tempi di esecuzione, 1241, 1271, 1273
- istruzioni di subroutine
 - tempi di esecuzione, 1241, 1270
- istruzioni di uscita
 - tempi di esecuzione, 1223, 1252
- istruzioni di visualizzazione
 - tempi di esecuzione, 1245, 1276
- istruzioni logiche
 - tempi di esecuzione, 1235, 1264
- istruzioni matematiche a virgola mobile
 - tempi di esecuzione, 1236, 1265
- istruzioni matematiche con simboli
 - tempi di esecuzione, 1231, 1260

istruzioni matematiche speciali
 tempi di esecuzione, 1235, 1265

istruzioni per contatore veloce e uscita a treno di impulsi,
823

istruzioni per l'orologio
 tempi di esecuzione, 1245, 1276

istruzioni per la comunicazione seriale
 tempi di esecuzione, 1243, 1274

istruzioni per la memoria dei file
 tempi di esecuzione, 1244, 1276

L

log degli errori
 impedire la memorizzazione di errori definiti dall'utente,
 1083

logaritmo, 597, 653

M

matematica
 addizione con virgola mobile, 570
 addizione virgola mobile, 626
 aggiunta di un intervallo di canali, 697
 calcolo della media, 769
 divisione con virgola mobile, 552, 576
 esponenti, 595, 651
 estrapolazione lineare, 542
 funzioni trigonometriche, 540
 istruzioni matematiche a virgola mobile, 63, 558–600,
 613–657
 istruzioni matematiche con simboli, 49, 409–465
 istruzioni matematiche speciali, 62, 534–1214
 logaritmo, 597, 653
 moltiplicazione con virgola mobile, 574
 moltiplicazione virgola mobile, 630
 radice quadrata, 534, 536, 593, 649
 ricerca del valore massimo in un intervallo, 689
 ricerca del valore minimo in un intervallo, 693
 sottrazione con virgola mobile, 572
 sottrazione virgola mobile, 628
 Vedere anche funzioni trigonometriche

memoria dei file
 istruzioni per la memoria dei file, 91, 1042–1045
 tempi di esecuzione delle istruzioni, 1244, 1276

messaggi
 programmazione, 1058

Moduli CPU CJ1, 3

Moduli di I/O ad alta densità gruppo 2
 aggiornamento con IORF(097), 886

moduli di I/O di base
 istruzioni per moduli di I/O di base, 84, 885–921

moduli di I/O speciali
 lettura della memoria del modulo, 913
 scrittura nella memoria del modulo, 917

O

operandi, 5
 inserimento di dati, 5

ora
 notazione di conversione oraria, 1068, 1070

orologio
 aggiunta di tempo all'orologio, 1061
 istruzioni per l'orologio, 92, 1061–1121
 sottrazione dal tempo dell'orologio, 1065

P

PLC serie CV
 conversione degli indirizzi di memoria, 1113, 1117

porta RS-232C
 ricezione dalla porta RS-232C, 944
 trasmissione dalla porta RS-232C, 937

precauzioni
 ambiente operativo, xxvi
 applicazioni, xxvi
 generali, xxiv
 sicurezza, xxiv

precauzioni di sicurezza
 Vedere anche precauzioni

programmazione
 capacità del programma, 2
 conversione di programmi, 1250, 1283
 creazione di programmi con struttura a step, 867
 errori di programma, 13
 messaggi di programmazione, 1058
 pausa/riavvio dei programmi a blocchi, 1131
 preparazione dei dati nelle aree dati, 331
 tempi di esecuzione delle istruzioni, 1221, 1250
 utilizzo di bit TR, 170

programmi a blocchi
 descrizione, 1124–1128
 diramazione, 1133, 1140, 1144, 1147, 1150, 1153
 istruzioni di programmazione a blocchi, 96, 1124–1157
 pausa/riavvio, 1131
 tempi di esecuzione delle istruzioni, 1246, 1278

programmi step
 creazione, 867

protocol macro, 928

R

radianti
 conversione da radianti a gradi, 579, 636

radice quadrata
 dati a virgola mobile, 593, 649
 dati BCD, 536
 dati binari con segno
 Vedere anche matematica

registrazione

- flag e bit di controllo, 1077
 - registrazione dei dati
 - Vedere anche* registrazione
 - registri indice
 - impostazione dell'indirizzo del valore attuale di un temporizzatore/contatore in un registro indice, 342
 - impostazione dell'indirizzo di canale/bit in un registro indice, 340
 - indirizzamento, 8
 - relé di blocco
 - utilizzo di KEEP(011), 180
 - reti
 - istruzioni di rete, 88, 973–1013
 - ripristino dei bit, 192
- S**
- salto, 219, 227
 - CJP(510) e CJPN(511), 223
 - schede di memoria
 - avvertenze, 1042
 - serie CJ
 - definizione, xix
 - Serie CS
 - definizione, xix
 - serie di istruzioni
 - (410), 424
 - (592), 397
 - *(420), 443
 - *B(424), 450
 - *BL(425), 452
 - *D(847), 630
 - *F(456), 574, 630
 - *L(421), 445
 - *U(422), 447
 - *UL(423), 449
 - +\$ (656), 1161
 - +(400), 410
 - ++(590), 393
 - ++B(594), 401
 - ++BL(595), 403
 - ++L(591), 395
 - +B(404), 418
 - +BC(406), 421
 - +BCL(407), 423
 - +BL(405), 419
 - +C(402), 414
 - +CL(403), 416
 - +D(845), 626
 - +F(454), 570, 626
 - +L(401), 412
 - /(430), 454
 - /B(434), 462
 - /BL(435), 464
 - /D(848), 632
 - /F(457), 576
 - /L(431), 456
 - /U(432), 458
 - /UL(433), 460
 - 7SEG(214), 908
 - ACC(888), 855
 - ACOS(464), 589, 645
 - ACOSD(855), 645
 - AND, 157
 - AND LD, 164
 - AND NOT, 159
 - ANDL(610), 519
 - ANDW(034), 517
 - APR(069), 540
 - ASC(086), 486
 - ASIN(463), 587, 643
 - ASIND(854), 643
 - ATAN(465), 591, 647
 - ATAND(856), 647
 - AVG(195), 769
 - B(414), 435
 - B(596), 405
 - BAND(681), 743
 - BC(416), 440
 - BCD(024), 469
 - BCDL(059), 470
 - BCDS(471), 505
 - BCL(417), 441
 - BCMP(068), 304
 - BCNT(067), 556
 - BDSL(473), 507
 - BIN(023), 466
 - BINL(058), 467
 - BINS(470), 499
 - BISL(472), 502
 - BL(415), 436
 - BL(597), 407
 - BPPS(811), 1131
 - BPRS(812), 1131
 - BREAK(514), 232
 - BSET(071), 331
 - C(412), 430
 - CADD(730), 1061
 - CCL(283), 1112
 - CCS(282), 1110
 - CJP(510), 223
 - CJPN(511), 223
 - CL(413), 432
 - CLC(041), 1105
 - CLI(691), 810
 - CLR\$(666), 1182
 - CMND(490), 973
 - CMP(020), 287
 - CMPL(060), 290
 - CNR(545), 267
 - CNT, 260

- CNTR(012), 263
CNTRX(548), 263
CNTW(814), 1147
CNTWX(818), 1147
CNTX(546), 260
COLL(081), 338, 1206
COLM(064), 496
COM(029), 531
COML(614), 533
COS(461), 583, 639
COSD(852), 639
CPS(114), 293
CPSL(115), 296
CSUB(731), 1065
CTBL(882), 837
-D(846), 628
DBL(843), 623
DBLL(844), 624
DEG(459), 579, 636
DEGD(850), 636
DEL\$(658), 1178
DI(693), 814
DIFD(014), 184–186
 utilizzo negli interblocchi, 203
 utilizzo nei salti, 222, 226, 228
DIFU(013), 184–186
 utilizzo negli interblocchi, 203
 utilizzo nei salti, 222, 226, 228
DIM(631), 678
DIST(080), 336
DLNK(226), 921
DMPX(077), 482
DOWN(522), 173
DSW(210), 890
EI(694), 816
ELSE(803), 1133
END (001), 197
EXIT(806), 1137
EXP(467), 595, 651
EXPD(858), 651
-F(455), 572, 628
FAL(006), 1079
FALS(007), 1087
FCS(180), 700
FDIV(079), 552
FIFO(633), 672
FIND\$(660), 1171
FIX(450), 563, 620
FIXD(841), 620
FIXL(451), 565, 604, 621
FIXLD(842), 621
FLT(452), 566, 623
FLTL(453), 568, 624
FOR(512), 229
FREAD(700), 1045
FRMCV(284), 1113
FSTR(448), 604
FVAL(449), 609
FWRIT(701), 1052
GETR(636), 683
GRET(752), 797
GSBN(751), 794
GSBS(750), 786
HEX(162), 490
HKY(212), 899
HMS(066), 1070
IEND(804), 1133
IF (802), 1133
IF(802), 1140
IL(002), 201–219
ILC(003), 201–219
INI(880), 823
INS\$(657), 1184
IORD(222), 913
IORF(097), 885
IORS(288), 1123
IOSP(287), 1121
IOWR(223), 917
istruzioni di confronto a virgola mobile in doppia precisione (da 335 a 340), 657
Istruzioni di confronto di ingresso a virgola mobile in singola precisione (da 329 a 334), 600
JME(005), 219
JME0(516), 227
JMP(004), 219
JMP0(515), 227
KEEP(011), 180
-L(411), 426
--L(593), 399
LD, 153
LD NOT, 155
LEFT\$(652), 1164
LENS\$(650), 1173
LEND(810), 1153
LIFO(634), 675
LINE(063), 494
LMT(680), 741
LOG(468), 597, 653
LOGD(859), 653
LOOP(809), 1153
MAX(182), 689
MCMP(019), 299, 313
MCRO(099), 779
MID\$(654), 1168
MIN(183), 693
MLPX(076), 477
MOV\$(664), 1159
MOV(021), 315
MOVB(082), 321
MOVD(083), 323
MOVL(498), 318
MOVR(560), 340

MOVW(561), 342
MSG(046), 1058
MSKR(692), 805
MSKS(690), 798
MTIM(543), 254
MTIMX(554), 254
MVN(022), 317
MVNL(499), 320
NEG(160), 472
NEGL(161), 474
NEXT(513), 229
NOP(000), 198
NOT(520), 172
OR, 161
OR LD, 166
OR NOT, 163
ORG(889), 862
ORW(035), 520
ORWL(611), 522
OUT, 177
OUT NOT, 178
OUTB(534), 195
PID(190), 720, 731, 1113, 1117, 1121, 1123
PIDAT(191), 731
PLS2(887), 849
PMCR(260), 928
PRV(881), 827, 833
PULS (886), 846
PUSH(632), 669
PWM(891), 865
PWRD(860), 655
RAD(458), 578, 634
RADD(849), 634
RECV(098), 973
RET(093), 786, 797
RGHT\$(653), 1166
ROOT(072), 536
ROTB(620), 534
RPLC\$(661), 1175
RSET, 187
RSTA(531), 189–192, 195
RSTB(533), 192
RXD(235), 944
SBN(092), 783, 794
SBS(091), 773, 786, 921
SCL(194), 757
SCL2(486), 762
SCL3(487), 766
SDEC(078), 888
SDEL(642), 716
SEC(065), 1068
SEND(090), 973, 991
SET, 187
SETA(530), 189–192, 195
SETB(532), 192
SETR(635), 681
SIGN(600), 476
SIN(460), 581, 637
SIND(851), 637
SINS(641), 713
SNUM(638), 704
SNXT(009), 868
SPED(885), 841
SQRT(466), 593, 649
SQRTD(857), 649
SRCH(181), 685
SREAD(639), 707
SSET(630), 666
Stato del bit di verifica, 174
STEP(008), 868
STUP(237), 968
SUM(184), 697
SWAP(637), 687, 704, 707, 710, 713, 716
SWRIT(640), 710
TAN(462), 585
TAND(853), 641
TCMP(085), 301
TIM, 235
TIMH(015), 240
TIMHWX(817), 1150
TIMHX(551), 240
TIML(542), 251
TIMLX(553), 251
TIMW(813), 1144
TIMWX(816), 1144
TIMX(550), 235
TKOF(821), 1196
TKON(820), 1192
TKY(211), 896
TMHH(540), 244
TMHHX(552), 244
TMHW(815), 1150
TOCV(285), 1117
TRSM(045), 1075
TST(350), 174
TSTN(351), 174
TTIM(087), 247
TTIMX(555), 247
TXD(236), 937
UP(521), 173
WDT(094), 1108
XCGL(562), 334
XCHG\$(665), 1180
XCHG(073), 333
XFER(070), 328
XFRB(062), 326
XNRL(613), 529
XNRW(037), 528
XORL(612), 526
XORW(036), 524
ZCP(088), 310
ZCPL(116), 313

ZONE(682), 746
simulazione di errori di sistema, 1079–1080, 1087
sistema SYSMAC LINK
 comunicazione, 973–979
sistema SYSMAC NET Link
 comunicazione, 973–979
stack
 istruzioni di stack, 660
stringhe di testo
 istruzioni di elaborazione delle stringhe di testo, 102,
 1158–1192
 tempi di esecuzione delle istruzioni, 1249, 1281
subroutine
 tempi di esecuzione, 1241, 1270

T

tabelle di confronto, 837
task
 istruzioni di controllo dei task, 105–107, 1192–1199
 programmi a blocchi nei task, 1125
 tempi di esecuzione delle istruzioni, 1249, 1281
task ciclici supplementari, 1192, 1196
tempi di esecuzione, 1219, 1221–1284
tempi di esecuzione delle istruzioni, 1221–1284
tempo di ciclo
 aumenta il tempo di ciclo massimo., 1108
 tempi di esecuzione delle istruzioni, 1219
tempo di ciclo massimo
 aumento, 1108
temporizzatore watchdog
 aumento, 1108
temporizzatori, 233–274
 applicazioni di esempio, 269
 reimpostazione con CNR(545), 267
 tempi di esecuzione, 1225, 1254
 temporizzatore in ritardo del programma a blocchi, 1150

U

unità Bus CPU
 aggiornamento, 921
uscite a impulsi
 controllo, 855
uscite a treno di impulsi, 823
 controllo, 823

V

velocità di uscita, 841

Storico delle revisioni

Il suffisso al numero di catalogo stampato sulla copertina del manuale indica il codice di revisione del documento.

N. cat. W340-IT2-11



Codice di revisione

Nella seguente tabella sono indicate le modifiche apportate al manuale nel corso di ciascuna revisione. I numeri di pagina si riferiscono alla versione precedente.

| Codice di revisione | Data | Contenuto modificato |
|---------------------|---------------|--|
| 01 | Febbraio 1999 | Stesura originale |
| 02 | Ottobre 1999 | Revisioni e integrazioni per la versione 1 del modulo CPU. Per un elenco, vedere a pagina 118. |
| 03 | Maggio 2000 | Apportate le seguenti revisioni e modifiche: Pagina xiii: aggiunta una precauzione. Pagina 8: eliminata una nota. Pagine 162, 166, 177, 180, 183, 189, 196, 198, 262, 531, 560 e 705: eliminati i registri indice dalle specifiche degli operandi. Pagina 170: eliminata la frase che inizia con "An error will occur if a JMP0(515)". Pagine 178, 181 e 184: aggiunta una precauzione sui numeri dei temporizzatori e una precauzione sull'utilizzo dei salti nel programma. Pagina 181: aggiunta una precauzione sull'aggiornamento del flag di completamento. Pagine 179, 182, 184: modificate le precauzioni sull'aggiornamento. Pagina 554: eliminate dalla precauzione le informazioni tra parentesi. Pagine 576, 577, 579, 581 e 583: modificata la descrizione per includere CS1W-INT01. Pagina 578: aggiunta una nota sull'utilizzo di CLI con MSKS. Pagine 578 e 583: modificata la precauzione sulla priorità di interrupt. Pagine 639, 647, 651 e 655: modificata la designazione della porta seriale. Pagina 642: aggiunto un riferimento al manuale. Pagina 675: aggiunte informazioni sulla struttura del file. Pagina 709: aggiunta una precauzione sul tempo di ciclo troppo lungo. |
| 04 | Novembre 2000 | Apportate le seguenti revisioni e modifiche: Pagine 169 e 170: eliminata la precauzione sul tempo di ciclo. Pagine 176, 180, 183, 186, 196, 199, 743, 746 e 749: corrette le indicazioni sul numero temporizzatore, numero contatore e valore impostato. Pagine 189 e 192: corrette le indicazioni sugli intervalli PV e SV. Pagine 209 e 210: modificato il programma ladder ed eliminata la nota di attenzione. Pagina 342: eliminata dalle precauzioni la descrizione dell'istruzione CLEAR CARRY. Pagina 395: riscritta la condizione ON del flag di errore. Pagina 531: aggiunta al diagramma l'indicazione di temporizzazione per l'aggiornamento della costante PID. Pagine 533 e 534: aggiunta alla fine la descrizione di PID con un esempio. Pagina 536: aggiunto alla tabella il bit 01 di C+5. Pagine 567, 572, 730, 732, 788 e 791: eliminata la nota sotto la tabella dei flag. Pagina 580: modificata la Nota 1 all'inizio della pagina. Pagina 613: modificati gli indirizzi CIO. Pagina 704: modificati i numeri FAL nella tabella degli operandi. |
| 05 | Maggio 2001 | Modificato il nome del manuale: "Serie CS1" modificato in "Serie CS" o "Serie CS/CJ," aggiunti PC della serie CJ e aggiunta l'indicazione "solo serie CS" alle funzioni specificate. Altre modifiche e integrazioni a riguardo sono state apportate nelle seguenti pagine: xv, 2, 661, 667, 678, Pagina 116: eliminata la sezione 3-2. Pagine 589, 590, 594 e 595: aggiunte informazioni per S e D. Pagina 598: modificate le intestazioni. |
| 06 | Ottobre 2001 | Nel manuale sono state aggiunte informazioni relative a nuovi prodotti, tra cui le CPU ad alta velocità (CPU CS1-H e CJ1-H), e le nuove istruzioni supportate. Poiché le modifiche apportate in tal senso sono particolarmente estese, non vengono elencate in questa sede. |
| 06A | Febbraio 2002 | Pagina 666: invertite le specifiche dei bit nella colonna <i>Dati di controllo</i> per i bit da 04 a 07 di C+6 e per i bit da 00 a 03 di C+6. |

| Codice di revisione | Data | Contenuto modificato |
|---------------------|----------------|--|
| 07 | Luglio 2002 | <p>Manuale modificato per aggiungere le CPU CJ1M e le nuove istruzioni supportate, tra cui il supporto dell'aggiornamento binario del valore attuale (PV) di temporizzatori/contatori. Poiché le modifiche apportate in tal senso sono particolarmente estese, non vengono elencate in questa sede.</p> <p>Aggiunte nuove istruzioni per temporizzatori e contatori: TIMX, TIMHX, TMHHX, TTIMX, TIMLX, MTIMX, CNTX, CNTRX e CNRX.</p> <p>Aggiunto BCMP2.</p> <p>Modificato ovunque "PC" in "PLC" quando si intende "controllore programmabile".</p> <p>Pagina x: Aggiunto un manuale e aggiornate le versioni del prodotto.</p> <p>Pagine 379 e 389: modificato l'esempio di programmazione.</p> <p>Pagina 489: modificato il simbolo "minore di" nel simbolo "minore o uguale".</p> <p>Pagina 490: modificato il grafico.</p> <p>Pagina 628: nell'esempio, modificato l'operando e aggiunta una nota.</p> <p>Pagine 648 e 651: modificata la prima voce del flag di errore.</p> <p>Pagina 666: corretti i numeri di bit nella tabella.</p> <p>Pagina 701: modificato il grafico per R+1.</p> <p>Pagine da 728 a 748: istruzioni rielaborate.</p> <p>Pagine 787, 814 e da 816 a 832: aggiunte informazioni sull'allocazione automatica delle porte.</p> <p>Pagine 820 e 825: aggiunte le precauzioni.</p> <p>Pagina 833: aggiunte precauzioni per l'utilizzo delle schede di memoria.</p> <p>Pagina 873: modificata la parte inferiore della pagina.</p> |
| 08 | Settembre 2002 | <p>Manuale modificato per aggiungere le CPU CS1D.</p> <p>Sono state apportate inoltre le seguenti modifiche.</p> <p>Pagina xiii: aggiunta nuova nota di attenzione.</p> <p>Pagine da xiv a xviii: sostituite le precauzioni per le applicazioni con la medesima sezione del <i>Manuale di programmazione</i>.</p> <p>Pagina 4: modificata la descrizione dell'operazione di aggiornamento immediato.</p> <p>Pagina 9: aggiunti tipi di dati.</p> <p>Pagine 222 e 225: aggiunta al grafico la frase "Non utilizzare".</p> <p>Pagina 683: corretto il grafico sulla risposta a rampa.</p> |
| 09 | Giugno 2003 | <p>Pagine 10 e 11: aggiunta una nota con esempi sulle istruzioni eseguibili quando le condizioni di ingresso sono OFF.</p> <p>Pagina 24: aggiornata la tabella e aggiunta una nota sulle istruzioni non supportate dalle CPU CS1D e dalle CPU CS1 con suffisso -V1.</p> <p>Pagine da 26 a 28: aggiornata la tabella e aggiunta una nota sulle istruzioni non supportate dalle CPU CS1D.</p> <p>Pagine 36 e 37: aggiornata la tabella e aggiunta una nota sulle istruzioni non supportate dalle CPU CS1D.</p> <p>Pagine 144, 148 e 152: aggiornate le tabelle e aggiunte alcune note sui nuovi modelli di CPU.</p> <p>Pagina 233: aggiunta una nota con informazioni sull'aggiunta di contatori tramite modifica in linea.</p> <p>Pagina 293: eliminate le informazioni sulla condizione del primo canale di destinazione.</p> <p>Pagina 679: aggiunte informazioni al grafico.</p> <p>Pagine 681 e 691: aggiunti alcuni termini alla tabella per meglio spiegare le impostazioni dei parametri.</p> <p>Pagina 692: corretti i numeri di bit (scambiati) per l'intervallo di uscita e per il modulo integrale e derivativo.</p> <p>Pagina 710: modificate le informazioni sull'emissione di valori negativi nei risultati di scala.</p> <p>Pagina 781: aggiunte alla tabella condizioni del flag di errore.</p> <p>Pagina 791: aggiunte informazioni nella nota sull'esecuzione di PLS2(887).</p> <p>Pagina 794: apportate correzioni alla tabella.</p> <p>Pagina 797: aggiunte informazioni nella nota sull'esecuzione di PLS2(887).</p> <p>Pagina 824: corretta la programmazione ladder per il processo B.</p> <p>Pagina 831: "Modulo di I/O" corretto in "Modulo I/O speciale".</p> <p>Pagine 844 e 845: modificate le informazioni sui primi canali/indirizzi di invio e lettura.</p> <p>Pagina 894: modificato il manuale di riferimento.</p> <p>Pagina 899: spostate le informazioni sulla struttura del file dalla pagina 912 del manuale precedente a questa pagina.</p> <p>Pagina : spostate le informazioni sulla struttura del file dalle pagine 912-913 del manuale precedente a questa pagina.</p> <p>Pagina 1110: aggiunta la tabella dei codici ASCII dalla pagina 916.</p> |

| Codice di revisione | Data | Contenuto modificato |
|---------------------|---------------|---|
| 10 | Dicembre 2003 | <p>Aggiunte le informazioni sulle funzioni supportate dalle nuove versioni delle CPU (troppo numerose, da elencare separatamente).</p> <p>Pagine da xi a xx: aggiornate le informazioni sulle precauzioni.</p> |
| 11 | Luglio 2004 | <p>Manuale modificato per aggiungere il modulo CPU versione 3.0 e le nuove istruzioni supportate. Poiché le modifiche apportate in tal senso sono particolarmente estese, non vengono elencate in questa sede.</p> <p>Nuove istruzioni: TXDU, RXDU, XFERC, DISTC, COLLC, MOVBC, BCNTC e GETID</p> <p>Istruzioni revisionate: TXD, RXD, PRV, PRV2, istruzioni di rete</p> <p>Modulo CPU aggiunto: CJ1H-CPU67H</p> <p>Sono state apportate inoltre le seguenti correzioni e modifiche.</p> <p>Pagina 99: corretti i codici di funzione per CNTWX e TWHWX.</p> <p>Pagine 183 e 229: aggiunte le precauzioni.</p> <p>Pagina 271: corretti i codici mnemonici nella tabella.</p> <p>Pagina 428: modificata l'intestazione.</p> <p>Pagina 676: sostituita la precauzione.</p> <p>Pagina 677: corretti i numeri di record.</p> <p>Pagina 857: sostituita la tabella degli identificatori porta.</p> |