SIEMENS

SIMATIC

Schema a contatti (KOP) per S7-300/400

Manuale di riferimento

Il presente manuale fa parte del pacchetto di documentazione con il numero di ordinazione:
6ES7810-4CA08-8EW1

Edizione 03/2006
A5E00706952-01

Prefazione,
Indice

Operazioni logiche combinatorie di bit **1**

Operazioni di confronto **2**

Operazioni di conversione **3**

Operazioni di conteggio **4**

Operazioni di blocco dati **5**

Operazioni di salto **6**

Operazioni matematiche con i numeri interi **7**

Operazioni matematiche in virgola mobile **8**

Operazioni di trasferimento **9**

Operazioni di controllo del programma **10**

Operazioni di scorrimento e rotazione **11**

Operazioni di bit di stato **12**

Operazioni di temporizzazione **13**

Operazioni logiche a parola **14**

Appendice

Sommario di tutte le operazioni KOP **A**

Esempi di programmazione **B**

Come operare con KOP **C**

Indice analitico

Istruzioni tecniche di sicurezza

Questo manuale contiene delle norme di sicurezza che devono essere rispettate per salvaguardare l'incolumità personale e per evitare danni materiali. Le indicazioni da rispettare per garantire la sicurezza personale sono evidenziate da un simbolo a forma di triangolo mentre quelle per evitare danni materiali non sono precedute dal triangolo. Gli avvisi di pericolo sono rappresentati come segue e segnalano in ordine descrescente i diversi livelli di rischio.

Pericolo

questo simbolo indica che la mancata osservanza delle opportune misure di sicurezza la morte o gravi lesioni fisiche.

Avvertenza

il simbolo indica che la mancata osservanza delle relative misure di sicurezza la morte o gravi lesioni fisiche.

Cautela

indica che la mancata osservanza delle relative misure di sicurezza può causare lesioni fisiche non gravi.

Cautela

indica che la mancata osservanza delle relative misure di sicurezza può causare danni materiali.

Attenzione

indica che, se non vengono rispettate le relative misure di sicurezza, possono subentrare condizioni o conseguenze indesiderate.

Nel caso in cui ci siano più livelli di rischio l'avviso di pericolo segnala sempre quello più elevato. Se in un avviso di pericolo si richiama l'attenzione con il triangolo sul rischio di lesioni alle persone, può anche essere contemporaneamente segnalato il rischio di possibili danni materiali.

Personale qualificato

L'apparecchio/sistema in questione deve essere installato e messo in servizio solo rispettando le indicazioni contenute in questa documentazione. La messa in servizio e l'esercizio di un apparecchio/sistema devono essere eseguiti solo da **personale qualificato**. Con riferimento alle indicazioni contenute in questa documentazione in merito alla sicurezza, come personale qualificato si intende quello autorizzato a mettere in servizio, eseguire la relativa messa a terra e contrassegnare le apparecchiature, i sistemi e i circuiti elettrici rispettando gli standard della tecnica di sicurezza.

Uso regolamentare delle apparecchiature/dei sistemi:

Si prega di tener presente quanto segue:

Avvertenza

L'apparecchiatura può essere destinata solo agli impieghi previsti nel catalogo e nella descrizione tecnica e può essere utilizzata solo insieme a apparecchiature e componenti di Siemens o di altri costruttori raccomandati o omologati dalla Siemens.

Per garantire un funzionamento ineccepibile e sicuro del prodotto è assolutamente necessario che le modalità di trasporto, di immagazzinamento, di installazione e di montaggio siano corrette, che l'apparecchiatura venga usata con cura e che si provveda ad una manutenzione appropriata.

Marchio di prodotto

I nomi di prodotto contrassegnati con ® sono marchi registrati della Siemens AG. Gli altri nomi di prodotto citati in questo manuale possono essere dei marchi il cui utilizzo da parte di terzi per i propri scopi può violare i diritti dei proprietari.

Esclusione di responsabilità

Abbiamo controllato che il contenuto di questa documentazione corrisponda all'hardware e al software descritti. Non potendo comunque escludere eventuali differenze, non possiamo garantire una concordanza perfetta. Il contenuto di questa documentazione viene tuttavia verificato periodicamente e le eventuali correzioni o modifiche vengono inserite nelle successive edizioni.

Siemens AG
Automation and Drives
Postfach 4848
90437 NÜRNBERG
GERMANIA

A5E00706952-01
03/2006

Copyright © Siemens AG 2006
Con riserva di eventuali modifiche tecniche

Prefazione

Scopo del manuale

Questo manuale ha lo scopo di supportare l'utente nella creazione di programmi nel linguaggio di programmazione KOP.

Esso descrive gli elementi del linguaggio di programmazione KOP, la sua sintassi e il modo di funzionamento.

Requisiti di base

I destinatari di questo manuale sono i programmatori di programmi S7, chi li mette in servizio e il personale di assistenza. Vengono presupposte delle nozioni generali nel campo della tecnica dell'automazione.

È inoltre necessario disporre delle conoscenze operative sui computer o strumenti di lavoro simili ai PC (p. es. dispositivi di programmazione) in ambiente Microsoft Windows 2000 Professional, Microsoft Windows XP Professional o Microsoft Windows Server 2003.

Validità del manuale

Il presente manuale ha validità per il pacchetto software STEP 7 V5.4.

Adempimento delle norme secondo l'IEC 1131-3

KOP corrisponde al linguaggio "Schema a contatti" stabilito nella norma DIN EN-61131-3 (int. IEC 1131-3), ma per quanto riguarda le operazioni vi sono delle differenze sostanziali. Informazioni precise sull'adempimento delle norme possono essere consultate nella tabella di adempimento delle norme nel file NORM_TAB.WRI di STEP 7.

Presupposti

Il presente manuale di KOP presuppone che l'utente sia in possesso delle nozioni teoriche inerenti i programmi S7 che sono riportate nella Guida online a STEP 7. Poiché i pacchetti dei linguaggi si basano sul software di base STEP 7 l'utente dovrebbe già sapere come utilizzare il software di base STEP 7 e la relativa documentazione.

Il presente manuale è parte integrante del pacchetto di documentazione "Nozioni di riferimento di STEP 7".

La tabella seguente riporta un riepilogo della documentazione relativa a STEP 7.

Documentazione	**Scopo**	**Numero di ordinazione**
Nozioni fondamentali di STEP 7 mediante • Primi passi ed esercitazioni con STEP 7 • Programmazione con STEP 7 • Configurazione dell'hardware e progettazione di collegamenti con STEP 7 • Manuale di conversione: STEP 7, da S5 a S7	Conoscenze di base per il personale tecnico: procedure per la realizzazione di compiti di controllo con STEP 7 e S7-300/400	6ES7810-4CA08-8EW0
Nozioni di riferimento di STEP 7 con • Manuali KOP/FUP/AWL per S7-300/400 • Funzioni standard e di sistema per S7-300/400 Volume 1 e Volume 2	Nozioni di riferimento sui linguaggi di programmazione KOP, FUP , AWL, nonché sulle funzioni standard e di sistema; perfezionamento delle conoscenze di base di STEP 7.	6ES7810-4CA08-8EW1

Guide online	**Scopo**	**Numero di ordinazione**
Guida a STEP 7	Conoscenze di base per la programmazione e la configurazione hardware con STEP 7	Parte del pacchetto software STEP 7
Guide di riferimento a AWL/KOP/FUP Guida di riferimento a SFB/SFC Guida di riferimento ai blocchi organizzativi	Guida di riferimento sensibile al contesto	Parte del pacchetto software STEP 7

Guida online

Come completamento del manuale è possibile avvalersi in fase operativa della dettagliata guida online integrata nel software.

Il sistema della guida è integrato nel software mediante differenti interfacce.

- La Guida al contesto offre informazioni sul contesto attuale, p. es. su una finestra di dialogo aperta o su una finestra attiva. È richiamabile con il pulsante "?" o con il tasto F1.
- Nel menu **?** sono disponibili diversi comandi: **Argomenti della Guida** apre l'indice della guida di STEP 7.
- Glossario relativo a tutte le applicazioni STEP 7 (Pulsante "Glosario").

Il presente manuale è un estratto della Guida a KOP. Manuale e guida online hanno quasi l'identica articolazione; è facile quindi passare dall'uno all'altra.

Ulteriore supporto

Per tutte le domande sull'uso dei prodotti descritti nel manuale, che non trovano risposta nella documentazione, rivolgersi al rappresentante Siemens locale.

Sito Internet delle rappresentanze Siemens:

http://www.siemens.com/automation/partner

Per la guida alla documentazione tecnica dei singoli prodotti e sistemi SIMATIC, consultare il sito:

http://www.siemens.com/simatic-tech-doku-portal

Il catalogo in linea e il sistema di ordinazione in linea si trova al sito:

http://mall.automation.siemens.com/

Centro di addestramento

Per facilitare l'approccio al sistema di automazione SIMATIC S7, la Siemens organizza corsi specifici. Rivolgersi a questo proposito al centro di addestramento locale più vicino o al centro di addestramento centrale di Norimberga.

Telefono: +49 (911) 895-3200.

Internet: http://www.sitrain.com

Technical Support

Per tutti i prodotti A&D è possibile rivolgersi al Technical Support

- mediante il modulo Web per la Support Request
  http://www.siemens.com/automation/support-request
- Telefono: + 49 180 5050 222
- Fax: + 49 180 5050 223

Per ulteriori informazioni sul Technical Support, consultare in Internet il sito
http://www.siemens.com/automation/service

Service & Support in Internet

Aggiuntivamente alla documentazione, mettiamo a disposizione della clientela diversi servizi in linea all'indirizzo sottoindicato.

http://www.siemens.com/automation/service&support

Su questo sito si possono trovare:

- la Newsletter con informazioni sempre aggiornate sui prodotti;
- i documenti appropriati relativi alla ricerca in Service & Support;
- il Forum, luogo di scambio di informazioni tra utenti e personale specializzato di tutto il mondo;
- il partner di riferimento locali di Automation & Drives;
- informazioni su assistenza tecnica sul posto, riparazioni, parti di ricambio e maggiori dettagli alla voce "Service".

Indice

1 Operazioni logiche combinatorie di bit 1-1

1.1 Sommario delle operazioni logiche combinatorie di bit 1-1
1.2 ---| |--- Contatto normalmente aperto 1-2
1.3 ---| / |--- Contatto normalmente chiuso 1-3
1.4 XOR Combinazione OR esclusivo 1-4
1.5 ---|NOT|--- Inverti il risultato della combinazione 1-5
1.6 ---( ) Bobina di uscita 1-6
1.7 ---(#)--- Connettore 1-8
1.8 ---(R) Resetta uscita 1-10
1.9 ---(S) Imposta uscita 1-12
1.10 RS Resetta imposta flip flop 1-14
1.11 SR Imposta resetta flip flop 1-16
1.12 ---(N)--- Rileva fronte RLC di discesa 1-18
1.13 ---(P)--- Rileva fronte RLC di salita 1-19
1.14 ---(SAVE) Salva RLC nel registro BIE 1-20
1.15 NEG Interroga rilevamento di fronte di discesa 1-21
1.16 POS Interroga rilevamento di fronte di salita 1-22
1.17 Leggi direttamente 1-23
1.18 Scrivi direttamente 1-24

2 Operazioni di confronto 2-1

2.1 Sommario delle operazioni di confronto 2-1
2.2 CMP ? I Confronta numeri interi a 16 bit 2-2
2.3 CMP ? D Confronta numeri interi a 32 bit 2-3
2.4 CMP ? R Confronta numeri in virgola mobile 2-4

3 Operazioni di conversione 3-1

3.1 Sommario delle operazioni di conversione 3-1
3.2 BCD_I Converti un numero BCD in un numero intero (a 16 bit) 3-2
3.3 I_BCD Converti un numero intero (a 16 bit) in un numero BCD 3-3
3.4 I_DI Converti un numero intero (a 16 bit) in un numero intero (a 32 bit) 3-4
3.5 BCD_DI Converti un numero BCD in un numero intero (a 32 bit) 3-5
3.6 DI_BCD Converti un numero intero (a 32 bit) in un numero BCD 3-6
3.7 DI_R Converti un numero intero (a 32 bit) in un numero in virgola mobile 3-7
3.8 INV_I Complemento a uno di numero intero (a 16 bit) 3-8
3.9 INV_DI Complemento a uno di numero intero (a 32 bit) 3-9
3.10 NEG_I Complemento a due di numero intero (a 16 bit) 3-10
3.11 NEG_DI Complemento a due di numero intero (a 32 bit) 3-11
3.12 NEG_R Complemento a due di numero in virgola mobile 3-12
3.13 ROUND Arrontonda a numero intero (a 32 bit) 3-13
3.14 TRUNC Arrotonda senza resto un numero intero (a 32 bit) 3-14
3.15 CEIL Genera da un numero in virgola mobile un numero intero superiore 3-15
3.16 FLOOR Genera da un numero in virgola mobile un numero intero inferiore 3-16

4 Operazioni di conteggio 4-1

4.1 Sommario delle operazioni di conteggio 4-1
4.2 ZÄHLER Parametrizza e conta in avanti/all'indietro 4-3
4.3 Z_VORW Parametrizza e conta in avanti 4-5
4.4 Z_RUECK Parametrizza e conta all'indietro 4-7
4.5 ---(SZ) Imposta valore di conteggio 4-9
4.6 ---(ZV) Conta in avanti 4-10
4.7 ---(ZR) Conta all'indietro 4-11

5 Operazioni di blocco dati 5-1

5.1 ---(OPN) Apri un blocco dati 5-1

6 Operazioni di salto 6-1

6.1 Sommario delle operazioni di salto 6-1
6.2 ---(JMP)--- Salto assoluto 6-2
6.3 ---(JMP)--- Se 1 salta in un FB (condizionato) 6-3
6.4 ---(JMPN) Se 0 salta in un FB (condizionato) 6-4
6.5 LABEL Etichetta di salto 6-5

7 Operazioni matematiche con i numeri interi 7-1

7.1 Sommario delle operazioni matematiche con i numeri interi 7-1
7.2 Valutazione dei bit nella parola di stato con operazioni con numeri in virgola fissa 7-2
7.3 ADD_I Somma numeri interi a 16 bit 7-3
7.4 SUB_I Sottrai numeri interi a 16 bit 7-4
7.5 MUL_I Moltiplica numeri interi a 16 bit 7-5
7.6 DIV_I Dividi numeri interi a 16 bit 7-6
7.7 ADD_DI Somma numeri interi a 32 bit 7-7
7.8 SUB_DI Sottrai numeri interi a 32 bit 7-8
7.9 MUL_DI Moltiplica numeri interi a 32 bit 7-9
7.10 DIV_DI Dividi numeri interi a 32 bit 7-10
7.11 MOD_DI Ricava resto della divisione (a 32 bit) 7-11

8 Operazioni matematiche in virgola mobile 8-1

8.1 Sommario delle operazioni matematiche con i numeri mobile 8-1
8.2 Valutazione dei bit nella parola di stato con operazioni in virgola mobile 8-2
8.3 Operazioni di base 8-3
8.3.1 ADD_R Somma numeri in virgola mobile 8-3
8.3.2 SUB_R Sottrai numeri in virgola mobile 8-5
8.3.3 MUL_R Moltiplica numeri in virgola mobile 8-6
8.3.4 DIV_R Dividi numeri in virgola mobile 8-7
8.3.5 ABS Formazione del valore assoluto di un numero in virgola mobile 8-8
8.4 Operazioni avanzati 8-9
8.4.1 SQR Formazione del quadrato di un numero in virgola mobile 8-9
8.4.2 SQRT Formazione della radice quadrata di un numero in virgola mobile 8-10
8.4.3 EXP Formazione del valore esponenziale di un numero in virgola mobile 8-11
8.4.4 LN Formazione del logaritmo naturale di un numero in virgola mobile 8-12
8.4.5 SIN Formazione del seno di un angolo come numero in virgola mobile 8-13
8.4.6 COS Formazione del coseno di un angolo come numero in virgola mobile 8-14
8.4.7 TAN Formazione della tangente di un angolo come numero in virgola mobile 8-15
8.4.8 ASIN Formazione dell'arcoseno di un numero in virgola mobile 8-16
8.4.9 ACOS Formazione dell'arcocoseno di un numero in virgola mobile 8-17
8.4.10 ATAN Formazione dell'arcotangente di un numero in virgola mobile 8-18

9 Operazioni di trasferimento 9-1

9.1 MOVE Assegna valore 9-1

10 Operazioni di controllo del programma 10-1

10.1 Sommario delle operazioni di comando del programma 10-1
10.2 ---(Call) Richiama FC/SFC dalla bobina (senza parametri) 10-2
10.3 CALL_FB Richiama FB dal box 10-4
10.4 CALL_FC Richiama FC dal box 10-6
10.5 CALL_SFB Richiama SFB dal box 10-8
10.6 CALL_SFC Richiama SFC dal box 10-10
10.7 Richiamo di una multi-istanza 10-12
10.8 Richiamo di blocchi da una biblioteca 10-13
10.9 Avvertenze importanti sulle funzionalità MCR 10-13
10.10 ---(MCR<) Attiva zona relè master control 10-14
10.11 ---(MCR>) Disattiva zona relè master control 10-16
10.12 ---(MCRA) Inizio zona relè master control 10-18
10.13 ---(MCRD) Fine zona relè master control 10-19
10.14 ---(RET) Salta indietro 10-20

11 Operazioni di scorrimento e rotazione 11-1

11.1 Operazioni di scorrimento 11-1
11.1.1 Sommario delle operazioni di scorrimento 11-1
11.1.2 SHR_I Fai scorrere numero intero a 16 bit verso destra 11-2
11.1.3 SHR_DI Fai scorrere numero intero a 32 bit verso destra 11-4
11.1.4 SHR_W Fai scorrere parola verso destra (a 16 bit) 11-5
11.1.5 SHL_W Fai scorrere parola verso sinistra (a 16 bit) 11-7
11.1.6 SHL_DW Fai scorrere doppia parola verso sinistra (a 32 bit) 11-9
11.1.7 SHR_DW Fai scorrere doppia parola verso destra (a 32 bit) 11-11
11.2 Operazioni di rotazione 11-13
11.2.1 Sommario delle operazioni di rotazione 11-13
11.2.2 ROL_DW Fai ruotare doppia parola verso sinistra (a 32 bit) 11-14
11.2.3 ROR_DW Fai ruotare doppia parola verso destra (a 32 bit) 11-16

12 Operazioni di bit di stato 12-1

12.1 Sommario delle operazioni di bit di stato 12-1
12.2 OV ---| |--- Bit di anomalia: overflow 12-2
12.3 OS ---| |--- Bit di anomalia: overflow con memoria 12-3
12.4 UO ---| |--- Bit di anomalia: operazione non ammessa 12-5
12.5 BIE ---| |--- Bit di anomalia: registro BIE 12-6
12.6 ==0 ---| |--- Bit di risultato uguale a 0 12-7
12.7 <>0 ---| |--- Bit di risulato diverso da 0 12-8
12.8 >=0 ---| |--- Bit di risultato maggiore o uguale a 0 12-9
12.9 >0 ---| |--- Bit di risultato maggiore di 0 12-10
12.10 <=0 ---| |--- Bit di risultato minore o uguale a 0 12-11
12.11 <0 ---| |--- Bit di risultato minore di 0 12-12

13 Operazioni di temporizzazione 13-1

13.1 Sommario delle operazioni di temporizzazione 13-1
13.2 Aree di memoria e componenti di un temporizzatore 13-2
13.3 S_IMPULS Parametrizza e avvia bobina temporizzatore come impulso (SI) 13-6
13.4 S_VIMP Parametrizza e avvia bobina temporizzatore come impulso prolungato (SV) 13-8
13.5 S_EVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione (SE) 13-10
13.6 S_SEVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inerzione con memoria (SS) 13-12
13.7 S_AVERZ Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione (SA) 13-14
13.8 ---(SI) Avvia bobina temporizzatore come impulso 13-16
13.9 ---(SV) Avvia bobina temporizzatore come impulso prolungato 13-18
13.10 ---(SE) Avvia bobina temporizzatore come ritardo all'inserzione 13-20
13.11 ---(SS) Avvia bobina temporizzatore come ritardo all'inserzione con memoria (SS) 13-22
13.12 ---(SA) Avvia bobina temporizzatore come ritardo alla disinserzione (SA) 13-24

14 Operazioni logiche a parola 14-1

14.1 Sommario delle operazioni logiche combinatorie a parola 14-1
14.2 WAND_W Combinazione AND parola 14-2
14.3 WOR_W Combinazione OR parola 14-3
14.4 WXOR_W Combinazione OR esclusivo parola 14-4
14.5 WAND_DW Combinazione AND doppia parola 14-5
14.6 WOR_DW Combinazione OR doppia parola 14-6
14.7 WXOR_DW Combinazione OR esclusivo doppia parola 14-7

A Sommario di tutte le operazioni KOP A-1

A.1 Operazioni KOP ordinate secondo il set mnemonico tedesco (SIMATIC) A-1
A.2 Operazioni KOP ordinate secondo il set mnemonico inglese (internazionale) A-5

B Esempi di programmazione B-1

B.1 Sommario B-1
B.2 Esempi: Operazioni logiche combinatorie a bit B-2
B.3 Esempio: Operazioni di temporizzazione B-6
B.4 Esempio: Operazioni di conteggio e confronto B-10
B.5 Esempio: Operazioni matematiche con i numeri interi B-13
B.6 Esempio: Operazioni logiche combinatorie a parola B-14

C Come operare con KOP C-1

C.1 Meccanismo EN/ENO C-1
C.1.1 Addendo con collegamento EN e collegamento ENO C-3
C.1.2 Addendo con collegamento EN e senza collegamento ENO C-4
C.1.3 Addendo senza collegamento EN e con collegamento ENO C-5
C.1.4 Addendo senza collegamento EN e senza collegamento ENO C-6
C.2 Assegnazione dei parametri C-7

Indice analitico ..... Indice analitico-1

1 Operazioni logiche combinatorie di bit

1.1 Sommario delle operazioni logiche combinatorie di bit

Descrizione

Le operazioni logiche combinatorie a bit operano con due cifre: 1 e 0. Queste due cifre costituiscono la base di un sistema numerico denominato sistema binario. Le due cifre 1 e 0 vengono denominate cifre binarie o bit. Nel mondo dei contatti e delle bobine, 1 sta a significare attivato o eccitato, e 0 sta per disattivato o diseccitato.

Le operazioni logiche combinatorie a bit interpretano gli stati di segnale di 1 e 0, e li combinano secondo la logica booleana per eseguire una varietà di funzioni. Queste combinazioni producono un risultato di 1 o 0 che è chiamato "risultato logico combinatorio" (RLC).

Qui di seguito sono elencate le operazioni logiche combinatorie a bit necessarie per eseguire le seguenti funzioni:

- ---| |--- Contatto normalmente aperto
- ---| / |--- Contatto normalmente chiuso
- ---(SAVE) Salva RLC nel registro BIE
- XOR Combina OR esclusivo
- ---( ) Bobina di uscita
- ---(#)--- Connettore
- ---|NOT|--- Inverti il risultato della combinazione

Le seguenti operazioni rispondono ad un RLC di 1:

- ---(S) Imposta bobina
- ---(R) Resetta bobina
- SR Imposta resetta flip flop
- RS Resetta imposta flip flop

Altre operazioni rispondono ad una transizione di fronte di salita o di discesa per eseguire le seguenti funzioni:

- ---(N)--- Rileva fronte RLC di discesa
- ---(P)--- Rileva fronte RLC di salita
- NEG Interroga rilevamento di fronte di discesa
- POS Interroga rilevamento di fronte di salita

- Leggi direttamente
- Scrivi direttamente

1.2 ---| |--- Contatto normalmente aperto

Simbolo

<Operando>

---| |---

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D, T, Z	Bit interrogato

Descrizione dell'operazione

---| |--- (Contatto normalmente aperto) è chiuso se il valore del bit interrogato, che viene memorizzato allo specificato **<operando>**, è "1". Se il contatto è chiuso, la corrente passa attraverso il contatto, e il bit di risultato logico combinatorio (RLC) è "1".

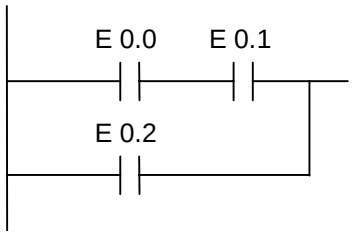
In caso contrario, se lo stato di segnale all'**<operando>** specificato è "0", il contatto è aperto. Se il contatto è aperto, la corrente non passa, e il bit di risultato logico combinatorio dell'operazione (RLC) è "0".

Nei collegamenti in serie, il contatto ---| |--- viene combinato con RLC bit per bit tramite AND. Nei collegamenti in parallelo, il contatto viene combinato con RLC tramite OR.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



La corrente circola se:

lo stato di segnale agli ingressi E 0.0 *e* E 0.1 è "1" *o* lo stato di segnale all'ingresso E 0.2 è "1".

(*e* = operazione AND; *o* = operazione OR)

1.3 ---| / |--- Contatto normalmente chiuso

Simbolo

<Operando>

---| / |---

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D, T, Z	Bit interrogato

Descrizione dell'operazione

---| / |--- (Contatto normalmente chiuso) è aperto se il valore del bit interrogato, che viene memorizzato all'**<operando>** specificato, è "0". Se il contatto è aperto, la corrente passa attraverso il contatto e il bit di risultato logico (RLC) è "1".

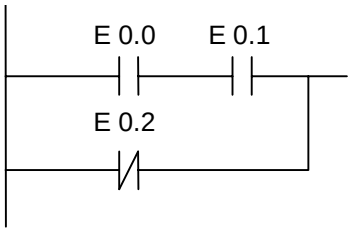
In caso contrario, se lo stato di segnale all'**<operando>** specificato è "1", il contatto è aperto. Se il contatto è aperto, la corrente non passa, e il bit di risultato logico combinatorio dell'operazione (RLC) è "0".

Nei collegamenti in serie, il contatto ---| / |--- viene combinato con RLC bit per bit tramite AND. Nei collegamenti inparallelo, il contatto viene combinato con RLC tramite OR.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



La corrente circola se:

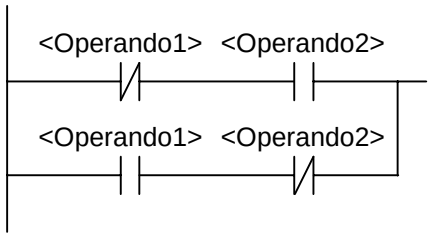
lo stato agli ingressi E 0.0 *e* E 0.1 è "1" *o* lo stato all'ingresso E 0.2 è "0".

(e = operazione AND; o = operazione OR)

1.4 XOR Combinazione OR esclusivo

Simbolo

Per la funzione **XOR** è necessario creare una rete di contatti normalmente aperti e normalmente chiusi (come descritto sotto).

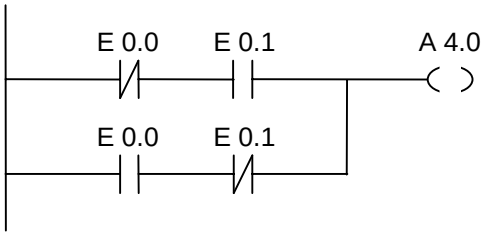


Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando1>	BOOL	E, A, M, L, D, T, Z	Bit interrogato
<Operando2>	BOOL	E, A, M, L, D, T, Z	Bit interrogato

Descrizione dell'operazione

XOR (Combinazione OR esclusivo) crea un RLC di "1", se lo stato di segnale di entrambi i bit indicati è differente.

Esempio



L'uscita A 4.0 è "1", se (E 0.0 = 0 *e* E 0.1 = 1) *o* (E 0.0 = 1 *e* E0.1 = 0).

(*o* = operazione OR, *e* = operazione AND)

1.5 ---|NOT|--- Inverti il risultato della combinazione

Simbolo

---|NOT|---

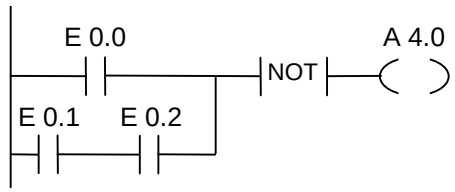
Descrizione dell'operazione

---|NOT|--- (Inverti il risultato della combinazione) inverte il bit RLC.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	-	1	x	-

Esempio



L'uscita A 4.0 è "0" se:

lo stato all'ingresso E 0.0 è "1" *o* lo stato di E 0.1. *e* E 0.2 è "1".

(e = operazione AND; o = operazione OR)

1.6 ---( ) Bobina di uscita

Simbolo

<Operando>

---( )

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D	Bit assegnato

Descrizione dell'operazione

---( ) (Bobina di uscita) opera come una bobina in uno schema di circuito. Se la corrente può circolare attraverso il circuito per raggiungere la bobina (RLC = 1), il bit all'**<operando>** viene impostato a "1". Se invece la corrente non può circolare attraverso il circuito per raggiungere la bobina (RLC = 0), il bit all'**<operando>** viene impostato a "0". Una bobina di uscita può essere posizionata solo all'estremità destra di un percorso di corrente di uno schema a contatti. Sono possibili al massimo 16 uscite multiple (vedere esempi). Si può creare un'uscita negata adoperando l'operazione ---|NOT|--- (Inverti il risultato della combinazione).

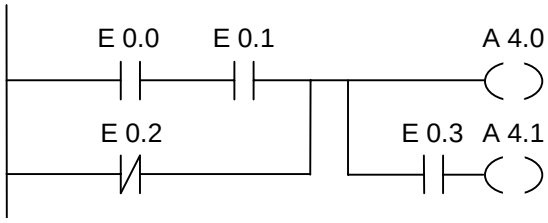
Dipendenza del MCR (Relé Master Control)

La dipendenza da MCR viene attivata se una bobina di uscita si trova in una zona MCR attiva. All'interno di una zona MCR attiva, l'MCR è attivato e la corrente può raggiungere una bobina di uscita. Il bit indirizzato viene impostato sullo stato di segnale del flusso di corrente. Se MCR è disattivato, uno "0" viene scritto agli operandi specificati, indipendentemente dallo stato di segnale del flusso di corrente.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	x	-	0

Esempio



L'uscita A 4.0 è "1" se:

(lo stato all'ingresso E 0.0 *e* E 0.1 è "1") *o* lo stato all'ingresso E 0.2 è "0".

L'uscita A 4.1 è "1" se:

(lo stato all'ingresso E 0.0 *e* E 0.1 è "1" *o* lo stato all'ingresso E 0.2 è "0") *e* lo stato all'ingresso E 0.3 è "1".

(e = operazione AND; o = operazione OR)

Se il percorso di corrente dell'esempio si trova in una zona MCR attiva:

Se MCR è attivato, A 4.0 e A 4.1 vengono settate secondo lo stato di segnale del flusso di corrente come descritto sopra.

Se MCR è disattivato, A 4.0 e A 4.1 vengono resettate a "0", indipendentemente dal flusso di segnale.

1.7 ---(#)--- Connettore

Simbolo

<Operando>

---(#)---

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, *L, D	Bit assegnato

* Un operando nello stack dei dati locali può essere utilizzato solo se esso è stato dichiarato nella tabella di dichiarazione di variabili nel settoreTEMP di un blocco di codice (FC, FB, OB).

Descrizione dell'operazione

---(#)--- (Connettore) è un elemento di assegnazione intermedia che memorizza il RLC (lo stato del flusso di segnale) in un **<operando>** specificato. Il connettore immagazzina il risultato degli elementi precedenti. Nei collegamenti in serie con altri elementi, l'operazione ---(#)--- viene inserita come un contatto. L'elemento ---(#)--- non può essere mai collegato alla sbarra di contatto o posizionato direttamente dopo una diramazione aperta, né può essere utilizzato come termine di una diramazione. Un elemento negato --(#)--- viene creato con l'elemento ---|NOT|--- (Inverti il risultato della combinazione).

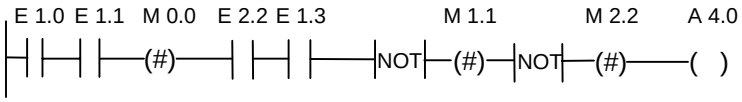
Dipendenza del MCR (Relé Master Control)

La dipendenza da MCR viene attivata se un connettore si trova in una zona MCR attiva. All'interno di una zona MCR attiva, MCR è attivato e la corrente può raggiungere un connettore. Il bit indirizzato viene quindi impostato sullo stato di segnale attuale del flusso di corrente. Se MCR è disattivato, uno "0" viene scritto all'operando specificato, indipendentemente dallo stato di segnale del flusso di corrente.

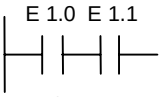
Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	x	-	0

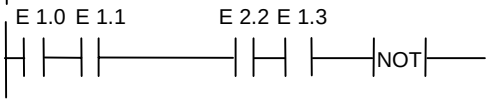
Esempio



M 0.0 ha RLC di



M 1.1 ha RLC di



M 2.2 ha RLC della combinazione logica di bit completa

1.8 ---(R) Resetta uscita

Simbolo

<Operando>

---(R)

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D, T, Z	Bit resettato

Descrizione dell'operazione

---(R) (Resetta uscita) viene eseguita solo se il RLC dell'operazione precedente è "1" (flusso di segnale alla bobina). Se la corrente può raggiungere la bobina (RLC è "1"), l'**<operando>** specificato viene impostato a "0". Se RLC è "0" (nessun flusso di segnale alla bobina), esso non ha effetto; lo stato di segnale dell'operando specificato rimane inalterato. L'**<operando>** può anche essere un temporizzatore (Nr. T) il cui valore di tempo viene impostato a "0", od un contatore (Nr. Z), il cui valore di conteggio viene impostato a "0".

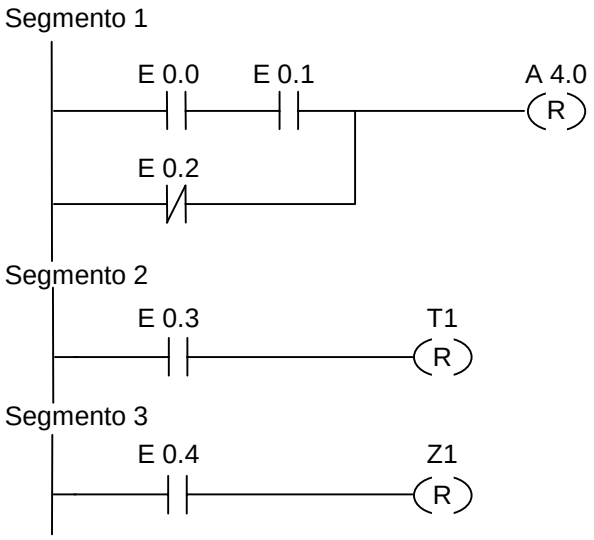
Dipendenza del MCR (Relé Master Control)

La dipendenza da MCR viene attivata se una bobina si trova in una zona MCR attiva. All'interno di una zona MCR attiva, MCR è attivato e la corrente può raggiungere una bobina. Il bit indirizzato viene quindi impostato a "0" . Se MCR è disattivato, lo stato di segnale attuale dell'operando specificato rimane inalterato, indipendentemente dallo stato di segnale del flusso di corrente.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	x	-	0

Esempio



L'uscita A 4.0 viene resettata se:

(lo stato all'ingresso E 0.0 *e* all'ingresso E 0.1 è "1") *o* lo stato all'ingresso E 0.2 è "1".
(e = operazione AND; o = operazione OR)

Il temporizzatore T1 viene resettato se:

lo stato di segnale all'ingresso E 0.3 è "1".

Il contatore Z1 viene resettato solo se:

lo stato di segnale all'ingresso E 0.3 è "1".

Se il percorso di corrente dell'esempio si trova in una zona MCR:

Se MCR è attivato, A 4.0, T1 e Z1 vengono resettati come descritto sopra.

Se MCR è disattivato, A 4.0, T1 e Z1 non vengono modificati, indipendentemente dallo stato di segnale di RLC (stato del flusso di segnale).

1.9 ---(S) Imposta uscita

Simbolo

<Operando>

---(S)

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D	Bit impostato

Descrizione dell'operazione

---(S) (Imposta uscita) viene eseguita solo se il RLC dell'operazione precedente è "1" (flusso di segnale alla bobina). Se RLC è "1", l'**<operando>** specificato dell'elemento viene impostato ad "1".

RLC = 0 non ha alcun effetto; lo stato di segnale dell'operando specificato rimane inalterato.

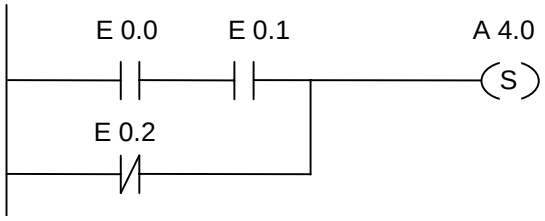
Dipendenza del MCR (Relé Master Control)

La dipendenza da MCR viene attivata se una bobina si trova in una zona MCR attiva. All'interno di una zona MCR attiva, MCR è attivato e la corrente può raggiungere una bobina. Il bit indirizzato viene quindi impostato a "1" . Se MCR è disattivato, lo stato di segnale attuale dell'operando specificato rimane inalterato, indipendentemente dallo stato di segnale del flusso di corrente.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	x	-	0

Esempio



L'uscita A 4.0 viene settata a "1" solo se:

(lo stato all'ingresso E 0.0 e all'ingresso E 0.1 è "1" tramite "AND") o lo stato all'ingresso E 0.2 è "1" tramite "OR".

Se RLC è "0", lo stato di segnale dell'uscita A 4.0 rimane uguale.

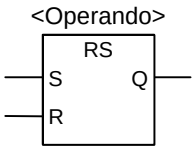
Se il percorso di corrente dell'esempio si trova in una zona MCR:

Se MCR è attivato, A 4.0 viene settata come descritto sopra.

Se MCR è disattivato, A 4.0 non viene modificata, indipendentemente dal flusso di segnale di RLC (stato del flusso di segnale).

1.10 RS Resetta imposta flip flop

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D	Bit impostato o resettato
S	BOOL	E, A, M, L, D	Abilitare l'operazione Imposta
R	BOOL	E, A, M, L, D	Abilitare l'operazione Resetta
Q	BOOL	E, A, M, L, D	Stato di segnale dell'<operando>

Descrizione dell'operazione

RS (Resetta imposta flip flop) viene impostata se lo stato di segnale all'ingresso S è "1" e lo stato all'ingresso R è "0". Invece, se lo stato di segnale all'ingresso R è "0" e all'ingresso S "1", flip flop viene impostato. Se ai due ingressi RLC è "1", flip flop RS esegue dapprima l'operazione Resetta, quindi l'operazione Imposta all'**<operando>** specificato, in modo tale che questo operando rimanga impostato per la parte rimanente del ciclo di programma.

Le operazioni S (Imposta) e R (Resetta) vengono eseguite solo se RLC = 1. Se RLC = 0, queste operazioni non vengono influenzate e l'operando specificato rimane invariato.

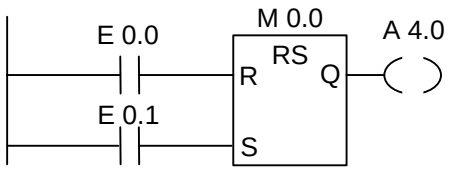
Dipendenza del MCR (Relé Master Control)

La dipendenza da MCR viene attivata se l'operazione Resetta imposta flip flop si trova in una zona MCR attiva. Se MCR è attivato, il bit indirizzato viene impostato a "1" o resettato a "0", come descritto sopra. Se MCR è disattivato, lo stato di segnale attuale dell'operando specificato rimane invariabile, indipendentemente dallo stato degli ingressi.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



Se lo stato all'ingresso E 0.0 è "1" e all'ingresso E 0.1 è "0", il merker M 0.0 viene resettato e l'uscita A 4.0 è "0". Invece, se lo stato di segnale all'ingresso E 0.0 = 0 e all'ingresso E 0.1 = 1, il merker M 0.0 viene impostato, e A 4.0 è "1". Se entrambi gli stati di segnale sono "0", non viene modificato nulla. Se entrambi gli stati di segnale sono "1", l'operazione Imposta prevale per via dell'ordine di esecuzione. M 0.0 viene impostato e A 4.0 è "1".

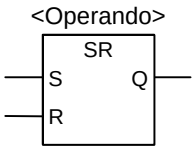
Se l'esempio soprariportato si trova in una zona MCR attiva

Se MCR è attivato, A 4.0 viene impostata o resettata come descritto sopra .

Se MCR è disattivato, A 4.0 non viene modificata, indipendentemente dallo stato di segnale degli ingressi.

1.11 SR Imposta resetta flip flop

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D	Bit impostato o resettato
S	BOOL	E, A, M, L, D	Abilita l'operazione Imposta
R	BOOL	E, A, M, L, D	Abilita l'operazione Resetta
Q	BOOL	E, A, M, L, D	Stato di segnale dell'<operando>

Descrizione dell'operazione

SR (Imposta resetta flip flop) viene impostata se lo stato di segnale all'ingresso S è "1", e lo stato all'ingresso R è "0". Invece, se lo stato di segnale all'ingresso S è "0", e all'ingresso R lo stato è "1", flip flop viene resettato. Se ai due ingressi RLC è "1", flip flop SR esegue dapprima l'operazione Imposta, quindi l'operazione Resetta, all'**<operando>** specificato, in modo tale che questo operando rimanga resettato per la parte rimanente del ciclo di programma.

Le operazioni S (Imposta) e R (Resetta) vengono eseguite solo se RLC = 1. Se RLC = 0, queste operazioni non vengono influenzate, e l'operando specificato rimane invariato.

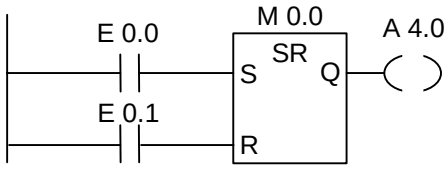
Dipendenza del MCR (Relé Master Control)

La dipendenza da MCR viene attivata se l'operazione Imposta resetta flip flop si trova in una zona MCR attiva. All'interno di una zona MCR attiva, MCR è attivato. Il bit indirizzato viene impostato a "1" o resettato a "0" come descritto sopra. Se MCR è disattivato, lo stato di segnale attuale dell'operando specificato rimane inalterato, indipendentemente dallo stato degli ingressi.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



Se lo stato all'ingresso E 0.0 è "1" e all'ingresso E 0.1 è "0", il merker M 0.0 viene impostato e l'uscita A 4.0 è "1". Invece, se lo stato di segnale all'ingresso E 0.0 = 0 e E 0.1 = 1, il merker M 0.0 viene resettato e A 4.0 è "0". Se entrambi gli stati di segnale sono "0", non viene modificato nulla. Se entrambi gli stati di segnale sono "1", l'operazione Resetta prevale per via dell'ordine di esecuzione. M 0.0 viene resettato e A 4.0 è "0".

Se l'esempio soprariportato si trova in una zona MCR attiva:

Se MCR è attivato, A 4.0 viene impostata come descritto sopra o resettata.

Se MCR è disattivato, A 4.0 non viene modificata, indipendentemente dallo stato di segnale degli ingressi.

1.12 ---(N)--- Rileva fronte RLC di discesa

Simbolo

<Operando>

---(N)---

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D	Il merker del fronte memorizza lo stato di segnale precedente di RLC

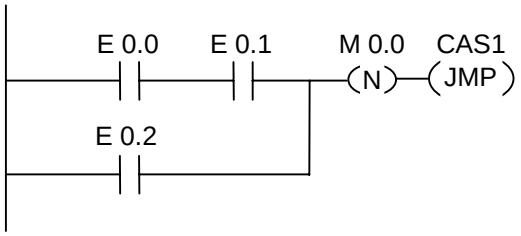
Descrizione dell'operazione

---(N)--- (Rileva fronte RLC di discesa) riconosce un cambiamento di segnale di RLC da "1" a "0", e riporta tale cambiamento a operazione terminata con RLC = 1. Lo stato attuale del RLC viene confrontato con lo stato di segnale dell'perando, il merker del fronte. Se stato di segnale dell'operando è "1" ed RLC prima dell'operazione è "0", RLC è "1" dopo l'operazione (impulso), ed è "0" negli altri casi. Il RLC prima dell'operazione viene memorizzato nell'operando.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	x	x	1

Esempio



Il merker del fronte M 0.0 memorizza il vecchio stato del RLC. Se lo stato di segnale di RLC cambia da "1" a "0" viene eseguito il salto all'etichetta CAS1.

1.13 ---(P)--- Rileva fronte RLC di salita

Simbolo

<Operando>

---(P)---

Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, L, D	Merker del fronte che memorizza lo stato di segnale di RLC precedente

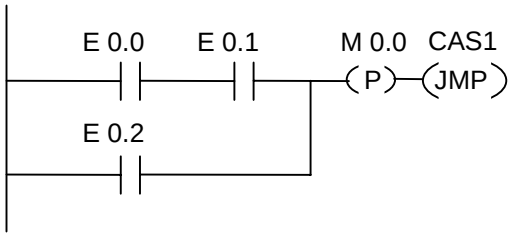
Descrizione dell'operazione

---(P)--- (Rileva fronte RLC di salita) rileva un cambiamento di segnale di RLC da "0" a "1", e riporta tale cambiamento a operazione terminata con RLC = 1. Lo stato di segnale attuale di RLC viene confrontato con lo stato di segnale dell'operando, il merker del fronte. Se lo stato di segnale dell'operando è "0", ed RLC prima dell'operazione è "1", RLC dopo l'operazione sarà "1" (impulso); in tutti gli altri casi, sarà "0". Il RLC prima dell'operazione viene memorizzato nell'operando.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	x	x	1

Esempio



Il merker del fronte M 0.0 memorizza il vecchio stato di RLC. Se lo stato di segnale di RLC passa da "0" a "1", viene eseguito il salto all'etichetta CAS1.

1.14 ---(SAVE) Salva RLC nel registro BIE

Simbolo

---(SAVE)

Descrizione dell'operazione

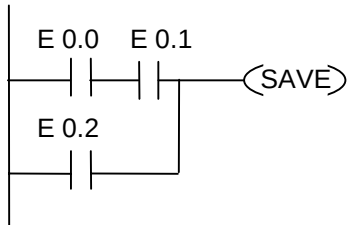
---(SAVE) (Salva RLC nel registro BIE) salva l´RLC nel bit BIE della parola di stato. Il bit di prima interrogazione/ER non viene resettato.
Per questo motivo, con una combinazione logica AND, anche lo stato del bit BIE viene combinato nel prossimo segmento.

Per quanto concerne l'operazione "SAVE" (KOP, FUP, AWL) si prega di leggere le regole seguenti, ignorando la descrizione riportata nei manuali e nelle Guide online.
Non si consiglia di utilizzare l'operazione SAVE e la seguente interrogazione del bit BIE nello stesso blocco o nel blocco subordinato visto che il bit BIE può subire delle modifiche a causa di diverse operazioni effettuate nel contempo. Si consiglia, invece, di utilizzare l'operazione SAVE prima di uscire dal blocco poichè in questo modo l'uscita ENO (=bit BIE) viene impostata sul valore del bit RLC. Dopo di che è possibile porre rimedio all'errore verificatosi nel blocco.

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	x	-	-	-	-	-	-	-	-

Esempio

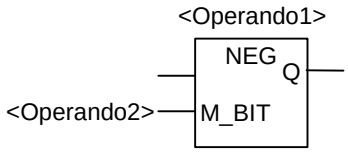


Lo stato del segmento (= RLC) viene memorizzato nel bit BIE.

BIE Bit di risultato binario (Parola di stato, bit 8)

1.15 NEG Interroga rilevamento di fronte di discesa

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando1>	BOOL	E, A, M, L, D	Segnale interrogato
<Operando2>	BOOL	E, A, M, L, D	Il merker di fronte M_BIT, memorizza il precedente stato di segnale dell'<operando1>
Q	BOOL	E, A, M, L, D	Riconoscimento del cambio di segnale

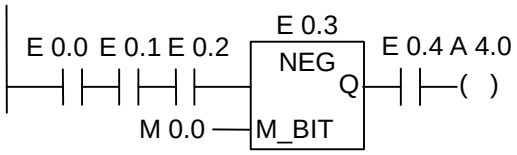
Descrizione dell'operazione

NEG (Interroga rilevamento di fronte di discesa) confronta lo stato di segnale dell'**<operando1>** con lo stato di segnale dell'interrogazione precedente memorizzato nell'**<operando2>**. Se lo stato attuale del RLC è 0, e lo stato precedente era "1" (rilevamento di un fronte di discesa), l'uscita Q è "1" dopo l'operazione, negli altri casi è "0".

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	1	x	1

Esempio

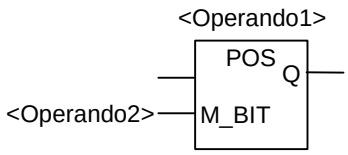


L'uscita 4.0 è "1" se:

(lo stato all'ingresso E 0.0 *e* E 0.1 *e* E 0.2 è "1") *e* E 0.3 ha un fronte negativo *e* lo stato all'ingresso E 0.4 è "1". (e = operazione AND)

1.16 POS Interroga rilevamento di fronte di salita

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando1>	BOOL	E, A, M, L, D	Segnale interrogato
<Operando2>	BOOL	E, A, M, L, D	Il merker del fronte M_BIT memorizza lo stato di segnale prcedente dell'<operando1>
Q	BOOL	E, A, M, L, D	Rilevamento del cambio di segnale

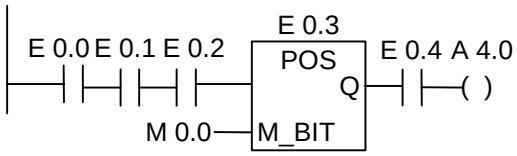
Descrizione dell'operazione

POS (Interroga rilevamento di fronte di salita) confronta lo stato di segnale dell'**<operando1>** con lo stato di segnale dell'interrogazione precedente memorizzato nell'**<operando2>**. Se lo stato attuale di RLC = 1, e lo stato precedente era "0" (rilevamento di un fronte di salita), l'uscita Q è "1" dopo l'operazione, negli altri casi è "0".

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	1	x	1

Esempio



L'uscita A 4.0 è "1" se:

(lo stato all'ingresso E 0.0 *e* E 0.1 *e* E 0.2 è "1") *e* E 0.3 ha un fronte positivo *e* lo stato all'ingresso E 0.4 è "1". (*e* = operazione AND).

1.17 Leggi direttamente

Descrizione dell'operazione

Per la funzione **Leggi direttamente** si deve creare un segmento come descritto sotto.

In applicazioni a tempo critico, è talvolta necessario leggere lo stato attuale di un ingresso digitale più spesso della norma (una volta per ciclo). L'operazione Leggi direttamente riceve lo stato dell'ingresso digitale dalle unità di ingresso nel momento in cui viene letto il relativo percorso di corrente. Altrimenti, è necessario attendere il successivo ciclo di OB1 perchè l'area di memoria degli ingressi venga aggiornata con l'area di memoria della periferia.

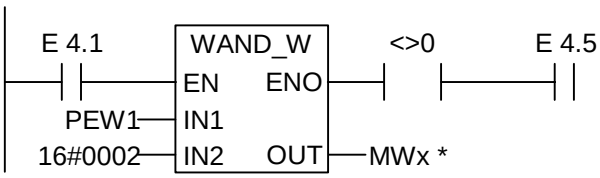
Se si desidera leggere l'ingresso (o più di un ingresso) direttamente dall'unità di ingresso è meglio utilizzare l'area di memoria Periferia degli ingressi (PE) invece dell'area di memoria degli ingressi (E). L'area di memoria della periferia può essere letta come byte, parola, o doppia parola. Per questo motivo, un singolo ingresso digitale non può essere letto tramite un contatto (bit).

Trasferimento condizionato di tensione in dipendenza dello stato di un ingresso diretto

1. La parola dell'area di memoria PE, che contiene i dati rilevanti, viene letta dalla CPU.
2. La parola dell'area di memoria PE viene quindi combinata tramite AND con una costante che permette un risultato diverso da zero, se il bit di ingresso è attivato ("1").
3. Viene verificata la condizione diversa da zero.

Esempio

Segmento KOP con l'operazione **Leggi direttamente** per l'ingresso E 1.1.



* MWx deve essere specificato per poter memorizzare il segmento. x sta per un numero permesso qualsiasi

Descrizione dell'operazione WAND_W:

PEW1 0000000000101010

W#16#0002 0000000000000010

Risultato 0000000000000010

In questo esempio, l'ingresso diretto E 1.1 è collegato in serie con gli ingressi E 4.1 e E 4.5.

La parola PEW1 contiene lo stato diretto di E 1.1. PEW1 viene combinata con W#16#0002 tramite AND. Il risultato è diverso da zero, se E 1.1 (secondo bit) è vero in PB1 ("1"). Il contatto A<>0 trasmette la tensione se l'accumulatore è diverso da zero.

1.18 Scrivi direttamente

Descrizione dell'operazione

Per la funzione **Scrivi direttamente** è necessario creare un segmento come descritto sotto.

In applicazioni a tempo critico è talvolta necessario trasferire ad un'unità di uscita lo stato attuale di una uscita digitale più spesso della norma (una volta alla fine di un ciclo OB). L'operazione Scrivi direttamente aggiorna lo stato di un'uscita digitale nell'unità di uscita nel momento in cui il relativo percorso di corrente viene scritto. Altrimenti, è necessario attendere la fine del ciclo principale perchè l'area di memoria della periferia viene aggiornata con lo stato dell'area di memoria delle uscite.

Se si desidera aggiornare direttamente l'uscita (o più di un'uscita), è meglio utilizzare l'area di memoria Periferia delle uscite (PA) invece dell'area delle uscite (A). L'area di memoria dell Periferia delle uscite può essere scritta come byte, parola, e doppia parola. Per questo motivo, una singola uscita digitale non può essere aggiornata tramite una bobina. Per poter scrivere lo stato di un'uscita digitale direttamente in un'unità di uscita, un byte, una parola o doppia parola dell'area di memoria delle uscite del settore delle uscite A , contenente il bit rilevante, viene copiato, in modo condizionato, nella relativa memoria PA (agli operandi dell'unità di uscita diretta).

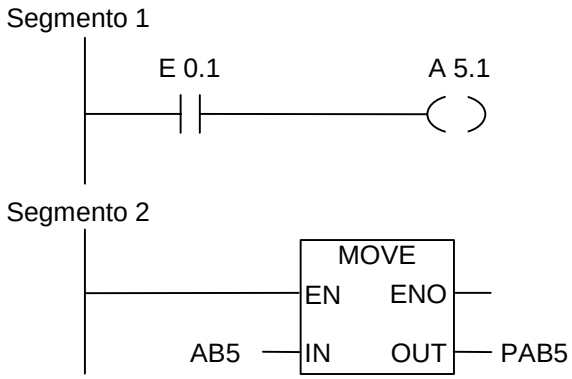
Avvertimenti

- Poiché il byte completo dell'area di memoria A viene scritto all'unità di uscita, tutti i bit di uscita del byte che viene aggiornato vengono altresì modificati, se l'operazione viene eseguita.
- Se un bit di uscita presenta stadi intermedi (1/0), che si manifestano durante il programma, e che non possono essere trasmessi alle unità di uscita, l'operazione Scrivi direttamente può provocare stati pericolosi (impulsi di trasferimento alle uscite).
- Come regola generale, bisogna tener presente nella configurazione che un'unità di uscita esterna può essere indirizzata come bobina una sola volta all'interno di un programma. L'osservanza di questa regola salvaguarda da eventuali problemi che possono sorgere con l'operazione Scrivi direttamente.

Esempio

Segmento KOP con l'operazione **Scrivi direttamente** e l'unità digitale di uscita 5, canale 1.

Gli stati dei bit del byte di uscita indirizzato (AB5) vengono o aggiornati o non influenzati. Nel segmento 1, A 5.1 viene assegnata allo stato di segnale di E 0.1. AB5 viene copiata nella corrispondente area di memoria diretta della periferia delle uscite (PAB5).



In questo esempio, A 5.1 è il bit di uscita richiesto.

Il byte PAB5 contiene lo stato del bit di uscita A 5.1.

Anche gli altri 7 bit in PAB5 vengono aggiornati copiando con l'operazione **MOVE**.

2 Operazioni di confronto

2.1 Sommario delle operazioni di confronto

Descrizione

Le operazioni di confronta confrontano gli ingressi IN1 e IN2 seconda dei tipi di confronto seguenti:

== IN1 uguale a IN2
<> IN1 diverso da IN2
> IN1 maggiore di IN2
< IN1 minore di IN2
>= IN1 maggiore di o uguale a IN2
<= IN1 minore di o uguale a IN2

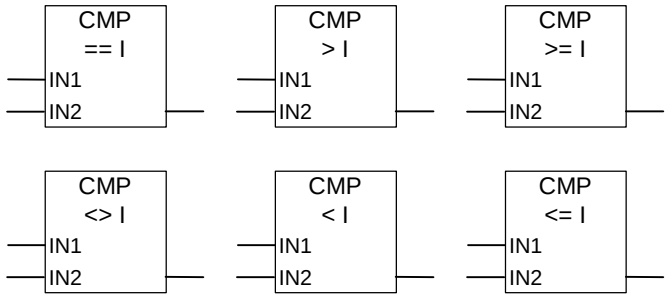
Se il confronto è vero, il RLC dell'operazione è "1". Il RLC viene combinato, rispettivamente, tramite AND con il RLC di un percorso di corrente, se l'elemento di confronto è collegato in serie, oppure tramite OR, se il box è collegato in parallelo.

Sono disponibili le seguenti operazioni di confronta:

- CMP ? I Confronta numeri interi a 16 bit
- CMP ? D Confronta numeri interi a 32 bit
- CMP ? R Confronta numeri in virgola mobile

2.2 CMP ? I Confronta numeri interi a 16 bit

Simboli



Parametro	Tipo di dati	Area di memoria	Descrizione
Ingresso di box	BOOL	E, A, M, L, D	Risultato della combinazione precedente
Uscita di box	BOOL	E, A, M, L, D	Risultato del confronto, viene elaborato solo se RLC all´ingresso di box = 1.
IN1	INT	E, A, M, L, D o costante	Primo valore di confronto
IN2	INT	E, A, M, L, D o costante	Secondo valore di confronto

Descrizione dell'operazione

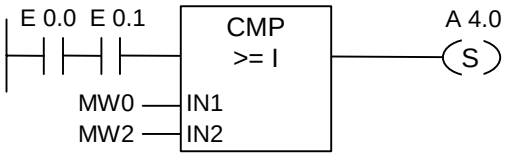
CMP ? I (Confronta numeri interi, a 16 bit) può essere usato come un normale contatto. Il box può essere posizionato come un qualsiasi contatto normale. IN1 e IN2 vengono confrontati secondo il tipo di confronto selezionato dall'utente.

Se il confronto è vero, il RLC dell'operazione è "1". Il RLC viene, rispettivamente, combinato tramite AND con il RLC dell'intero percorso di corrente se l'elemento di confronto è collegato in serie, oppure tramite OR se il box è collegato in parallelo.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	0	-	0	x	x	1

Esempio

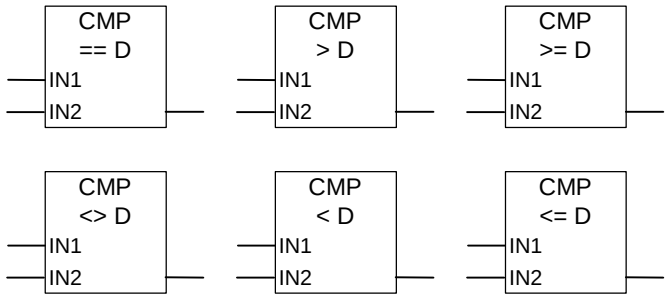


A 4.0 viene settata se E 0.0 *e* E 0.1 sono = 1, *e* MW0 >= MW2.

(*e* = operazione AND, *o* = operazione OR)

2.3 CMP ? D Confronta numeri interi a 32 bit

Simboli



Parametro	Tipo di dati	Area di memoria	Descrizione
Ingresso di box	BOOL	E, A, M, L, D	Risultato della combinazione precedente
Uscita di box	BOOL	E, A, M, L, D	Risultato del confronto, viene elaborato solo se RLC all´ingresso di box = 1.
IN1	DINT	E, A, M, L, D o costante	Primo valore di confronto
IN2	DINT	E, A, M, L, D o costante	Secondo valore di confronto

Descrizione dell'operazione

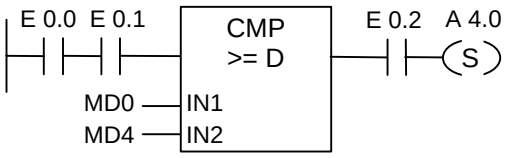
CMP ? D (Confronta numeri interi, a 32 bit) può essere usato come un normale contatto. Il box può essere posizionato come un qualsiasi contatto normale. IN1 e IN2 vengono confrontati secondo il tipo di confronto selezionato dall'utente.

Se il confronto è vero, il RLC dell'operazione è "1". Il RLC viene, rispettivamente, combinato tramite AND con il RLC dell'intero percorso di corrente se l'elemento di confronto è collegato in serie, oppure tramite OR se il box è collegato in parallelo.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	0	-	0	x	x	1

Esempio

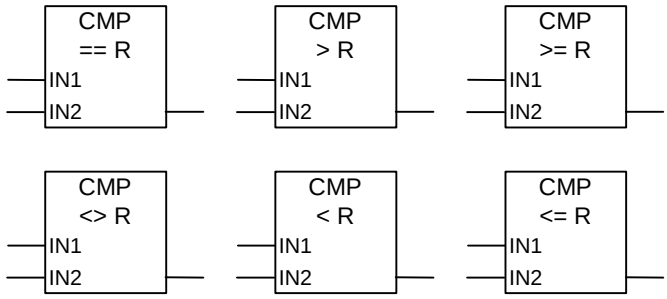


A 4.0 viene settata se E 0.0 *e* E 0.1 sono = 1 *e* se MD0 >= MD4 *e* E 0.2 = 1.

(e = operazione AND)

2.4 CMP ? R Confronta numeri in virgola mobile

Simboli



Parametro	Tipo di dati	Area di memoria	Descrizione
Ingresso di box	BOOL	E, A, M, L, D	Risultato della combinazione precedente
Uscita di box	BOOL	E, A, M, L, D	Risultato del confronto, viene elaborato solo se RLC = 1.
IN1	REAL	E, A, M, L, D o costante	Primo valore di confronto
IN2	REAL	E, A, M, L, D o costante	Secondo valore di confronto

Descrizione dell'operazione

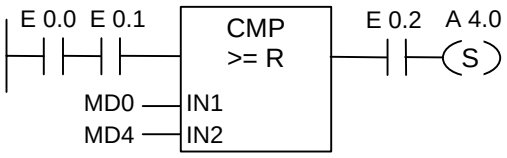
CMP ? R (Confronta numeri in virgola mobile) può essere usato come un normale contatto. Il box può essere posizionato come un qualsiasi contatto normale. IN1 e IN2 vengono confrontati secondo il tipo di confronto selezionato dall'utente.

Se il confronto è vero, il RLC dell'operazione è "1". Il RLC viene combinato, rispettivamente, tramite AND con il RLC dell'intero percorso di corrente, se l'elemento di confronto è collegato in serie, oppure tramite OR, se il box è collegato in parallelo.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

Esempio



A 4.0 viene settata se E 0.0 *e* E 0.1 sono = 1, *e* se MD0 >= MD4 *e* se E 0.2 = 1.

(e = operazione AND)

3 Operazioni di conversione

3.1 Sommario delle operazioni di conversione

Descrizione

Le operazioni di conversione leggono il contenuto del parametro IN e lo convergono, oppure ne cambiano il segne. Il risultato viene scritto nel parametro OUT.

Sono disponibili le seguenti operazioni di conversione:

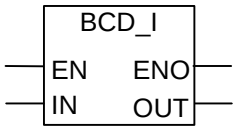
- BCD_I Converti un numero BCD in un numero intero (a 16 bit)
- I_BCD Converti un numero intero (a16 bit) in un numero BCD
- BCD_DI Converti un numero BCD in un numero intero (a 32 bit
- I_DI Converti un numero intero (a 16 bit) in un numero intero (a 32 bit)
- DI_BCD Converti un numero intero (a 32 bit) in un numero BCD
- DI_R Converti un numero intero (a 32 bit) in un numero in virgola mobile

- INV_I Complemento a uno di numero intero (a 16 bit)
- INV_DI Complemento a uno di numero intero (a 32 bit)
- NEG_I Complemento a due di numero intero (a 16 bit)
- NEG_DI Complemento a due di numero intero (a 32 bit)

- NEG_R Complemento a due di numero in virgola mobile
- ROUND Arrontonda a numero intero (a 32 bit)
- TRUNC Arrontonda senza resto parte numero intero (a 32 bit)
- CEIL Genera da un numero in virgola mobile un numero intero superiore
- FLOOR Genera da un numero in virgola mobile un numero intero inferiore

3.2 BCD_I Converti un numero BCD in un numero intero (a 16 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	WORD	E, A, M, L, D	Numero in formato BCD
OUT	INT	E, A, M, L, D	Valore di numero intero (a 16 bit) del numero BCD

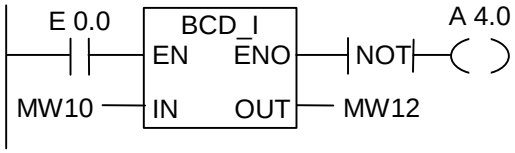
Descrizione dell'operazione

BCD_I (Converti un numero BCD in un numero intero (a 16 bit)) legge il contenuto specificato nel parametro IN come numero a tre cifre dal formato di cifra decimale in codice binario (+/- 999); e converte questo numero in un valore di numero intero (a 16 bit). Il risultato di numero intero viene emesso dal parametro OUT. ENO ha sempre lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	-	-	-	-	0	1	1	1

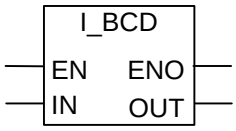
Esempio



Se l'ingresso E 0.0 = 1, il contenuto di MW10 viene letto come numero a tre cifre in formato BCD; e viene quindi convertito in un numero intero (a 16 bit). Il risultato viene memorizzato in MW12. L'uscita A 4.0 è "1" se la conversione non viene eseguita (ENO = EN = 0).

3.3 I_BCD Converti un numero intero (a 16 bit) in un numero BCD

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	INT	E, A, M, L, D	Numero intero (a 16 bit)
OUT	WORD	E, A, M, L, D	Valore BCD del numero intero (a 16 bit)

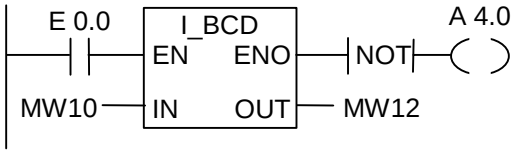
Descrizione dell'operazione

I_BCD (Converti un numero intero (a 16 bit) in un numero BCD) legge il contenuto specificato nel parametro IN come numero intero (a 16 bit), e converte questo valore in un numero a tre cifre in formato BCD (+/- 999). Il risultato in numero intero viene emesso dal parametro OUT. Se si verifica un'eccedenza (overflow), ENO = 0.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	-	-	x	x	0	x	x	1

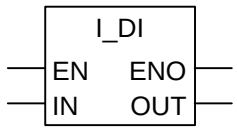
Esempio



Se l'ingresso E 0.0 = 1, il contenuto di MW10 viene letto come numero intero (a 16 bit), e viene quindi convertito in un numero a tre cifre in formato BCD. Il risultato viene memorizzato in MW12. L'uscita A 4.0 è "1" se si verifica un'eccedenza (overflow) o se la conversione non viene eseguita (E0.0 = 0).

3.4 I_DI Converti un numero intero (a 16 bit) in un numero intero (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	INT	E, A, M, L, D	Valore di numero intero (a 16 bit) da convertire
OUT	DINT	E, A, M, L, D	Risultato: numero intero (a 32 bit)

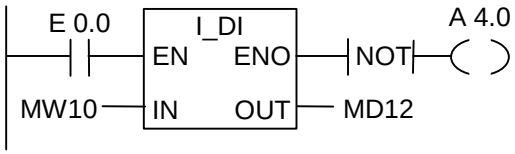
Descrizione dell'operazione

I_DI (Converti un numero intero (a 16 bit) in un numero intero (a 32 bit)) legge il contenuto specificato nel parametro IN come valore di numero intero (a 16 bit), e converte questo valore in un numero intero (a 32 bit). Il risultato viene fornito dal parametro OUT. ENO ha sempre lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	-	-	-	-	0	1	1	1

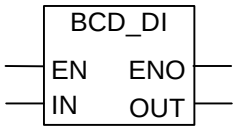
Esempio



Se E 0.0 = 1, il contenuto di MW10 viene letto come numero intero (a 16 bit), e quindi convertito in un numero intero (a 32 bit). Il risultato viene memorizzato in MD12. Se la conversione non viene eseguita, l'uscita A 4.0 è "1" (ENO = EN = 0).

3.5 BCD_DI Converti un numero BCD in un numero intero (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DWORD	E, A, M, L, D	Numero in formato BCD
OUT	DINT	E, A, M, L, D	Valore di numero intero (a 32 bit) del numero BCD

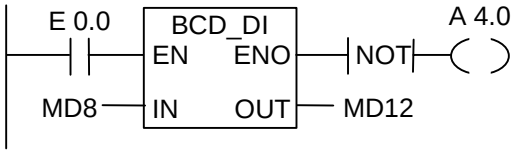
Descrizione dell'operazione

BCD_DI (Converti un numero BCD in un numero intero (a 32 bit)) legge il contenuto specificato nel parametro IN come numero a sette cifre dal formato di cifra decimale in codice binario (+/- 9999999), e converte questo numero in un valore di numero intero (a 32 bit). Il risultato in numero intero viene emesso dal parametro OUT. ENO ha sempre lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	-	-	-	-	0	1	1	1

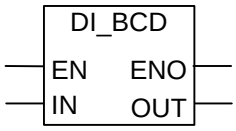
Esempio



Se l'ingresso E 0.0 = 1, il contenuto di MD8 viene letto come numero a sette cifre in formato BCD, e viene quindi convertito in un numero intero (a 32 bit). Il risultato viene memorizzato in MD12. L'uscita A 4.0 è "1" se la conversione non viene eseguita (ENO = EN = 0).

3.6 DI_BCD Converti un numero intero (a 32 bit) in un numero BCD

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DINT	E, A, M, L, D	Numero intero (a 32 bit)
OUT	DWORD	E, A, M, L, D	Valore BCD del numero intero (a 32 bit)

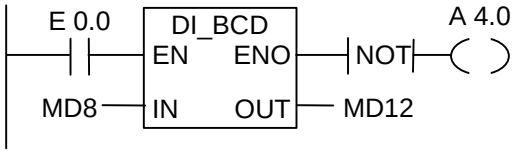
Descrizione dell'operazione

DI_BCD (Converti un numero intero (a 32 bit) in un numero BCD) legge il contenuto specificato nel parametro IN come numero intero (a 32 bit), e converte questo valore in un numero a sette cifre in formato BCD (+/- 9999999). Il risultato in numero intero viene emesso dal parametro OUT. Se si verifica un'eccedenza (overflow), ENO = 0.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	-	-	x	x	0	x	x	1

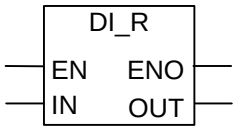
Esempio



Se l'ingresso E 0.0 = 1, il contenuto di MD8 viene letto come numero intero (a 32 bit), e viene quindi convertito in un numero a sette cifre in formato BCD. Il risultato viene memorizzato in MD12. L'uscita A 4.0 è "1" se si verifica un'eccedenza (overflow) o se la conversione non viene eseguita (E0.0 = 0).

3.7 DI_R Converti un numero intero (a 32 bit) in un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DINT	E, A, M, L, D	Numero intero (a 32 bit)
OUT	REAL	E, A, M, L, D	Numero in virgola mobile

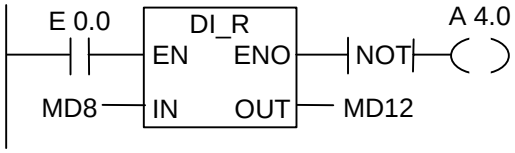
Descrizione dell'operazione

DI_R (Converti un numero intero (a 32 bit) in un numero in virgola mobile) legge il contenuto specificato nel parametro IN come valore di numero intero (a 32 bit), e converte questo valore in un numero in virgola mobile. Il risultato viene fornito dal parametro OUT. ENO ha sempre lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	-	-	-	-	0	1	1	1

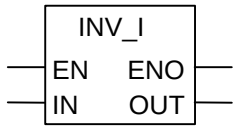
Esempio



Se E 0.0 = 1, il contenuto di MD8 viene letto come numero intero (a 32 bit), e quindi convertito in un numero in virgola mobile. Il risultato viene memorizzato in MD12. Se la conversione non viene eseguita, l'uscita A 4.0 è "1" (ENO = EN = 0).

3.8 INV_I Complemento a uno di numero intero (a 16 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	INT	E, A, M, L, D	Valore di ingresso (a 16 bit) di numero intero
OUT	INT	E, A, M, L, D	Complemento a uno del numero intero (a 16 bit) di IN

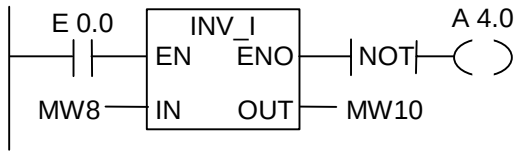
Descrizione dell'operazione

INV_I (Complemento a uno di numero intero (a 16 bit)) legge il contenuto specificato nel parametro IN, e combina il valore con la maschera esadecimale W#16#FFFF tramite OR esclusivo. Questa operazione converte ogni bit nel suo valore opposto. ENO ha sempre lo stesso valore di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	-	-	-	-	0	1	1	1

Esempio



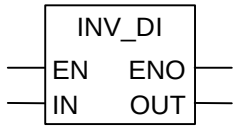
Se E 0.0 = 1, lo stato di ogni singolo bit di MW8 viene convertito.

Ad esempio:

MW8 = 01000001 10000001 diventa MW10 = 10111110 01111110. L'uscita è A 4.0 è "1" se la conversione non è stata eseguita (ENO = EN).

3.9 INV_DI Complemento a uno di numero intero (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DINT	E, A, M, L, D	Valore di ingresso (a 32 bit) di numero intero
OUT	DINT	E, A, M, L, D	Complemento a uno del numero intero (a 32 bit) di IN

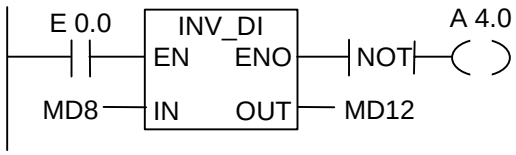
Descrizione dell'operazione

INV_DI (Complemento a uno di numero intero (a 32 bit)) legge il contenuto specificato nel parametro IN e combina il valore con la maschera esadecimale W#16#FFFF FFFF tramite OR esclusivo. Questa operazione converte ogni bit nel suo valore opposto. ENO ha sempre lo stesso valore di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	-	-	-	-	0	1	1	1

Esempio



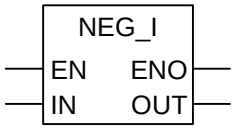
Se E 0.0 = 1, lo stato di ogni singolo bit di MD8 viene convertito.

Ad esempio:

MD8 = F0FF FFF0 diventa MD12 = 0F00 000F. L'uscita A 4.0 è "1" se la conversione non è stata eseguita (ENO = EN = 0).

3.10 NEG_I Complemento a due di numero intero (a 16 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	INT	E, A, M, L, D	Valore di ingresso (a 16 bit) di numero intero
OUT	INT	E, A, M, L, D	Complemento a due del numero intero (a 16 bit) di IN

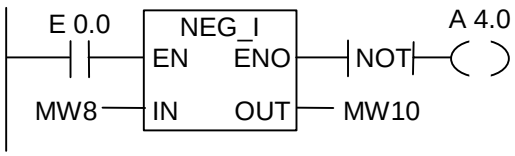
Descrizione dell'operazione

NEG_I (Complemento a due di numero intero (a 16 bit)) legge il contenuto specificato nel parametro IN, ed esegue l'operazione di complemento a due. L'operazione cambia il segno (per esempio, da un valore positivo ad un valore negativo). ENO ha sempre lo stesso valore di EN, con la seguente eccezione: se lo stato di segnale di EN = 1, e si verifica un'eccedenza (overflow), lo stato di segnale ENO = 0.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

Esempio



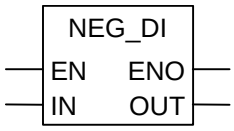
Se E 0.0 = 1, il valore di MW8 viene emesso dal parametro O in MW10 con il segno opposto.

MW8 = + 10 corrisponde a MW10 = - 10. L'uscita A 4.0 è "1" se la conversione non è stata eseguita (ENO = EN).

Se lo stato di segnale di EN = 1 e si verifica un'eccedenza (overflow), lo stato di segnale di ENO = 0.

3.11 NEG_DI Complemento a due di numero intero (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DINT	E, A, M, L, D	Valore di ingresso di numero intero (a 32 bit)
OUT	DINT	E, A, M, L, D	Complemento a due del numero intero (a 32 bit) di IN

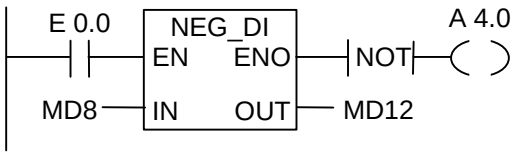
Descrizione dell'operazione

NEG_DI (Complemento a due di numero intero (a 32 bit)) legge il contenuto specificato nel parametro IN ed esegue l'operazione di complemento a due. L'operazione cambia il segno (per esempio, da un valore positivo ad un valore negativo). ENO ha sempre lo stesso valore di EN, con la seguente eccezione: se lo stato di segnale di EN = 1, e si verifica un'eccedenza (overflow), lo stato di segnale ENO = 0.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

Esempio



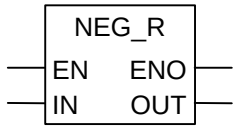
Se E 0.0 = 1, il valore di MD 8 viene fornito con il segno opposto dal parametro O di MD12.

MD8 = + 1000 corrisponde a MD12 = - 1000. L'uscita A 4.0 è "1" se la conversione non è stata eseguita (ENO = EN).

L'operazione cambia il segno (per esempio, da un valore positivo ad un valore negativo). ENO ha sempre lo stesso valore di EN, con la seguente eccezione: se lo stato di segnale di EN = 1, e si verifica un'eccedenza (overflow), lo stato di segnale ENO = 0.

3.12 NEG_R Complemento a due di numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Numero in virgola mobile di IN con segno opposto

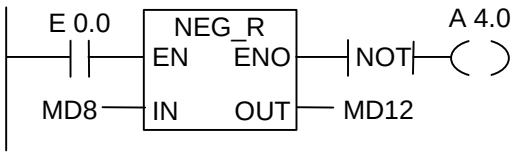
Descrizione dell'operazione

NEG_R (Complemento a due di numero in virgola mobile) legge il contenuto specificato nel parametro IN e cambia il segno. Questa operazione equivale ad una moltiplicazione per (-1). L'operazione cambia il segno (per esempio, da un valore positivo ad un valore negativo). ENO ha sempre lo stesso valore di EN, con la seguente eccezione: se lo stato di segnale di EN = 1, e si verifica un'eccedenza (overflow), lo stato di segnale ENO = 0.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	-	-	-	-	0	x	x	1

Esempio



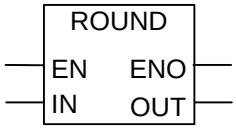
Se E 0.0 = 1, il valore di MD8 viene emesso dal parametro O di MD12 con il segno opposto.

MD8 = + 6,234 equivale a MD12 = - 6,234.

L'uscita A 4.0 è "1" se la conversione non è stata eseguita (ENO = EN = 0).

3.13 ROUND Arrontonda a numero intero (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D	Valore da arrotondare
OUT	DINT	E, A, M, L, D	IN arrotodato al numero intero successivo

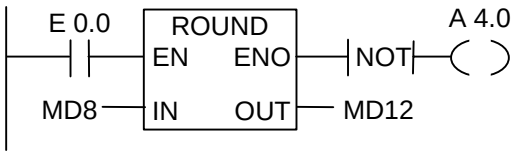
Descrizione dell'operazione

ROUND (Arrotonda a numero intero (a 32 bit)) legge il contenuto specificato nel parametro IN come numero in virgola mobile, e quindi converte questo numero in un numero intero (a 32 bit). Il risultato corrisponde al numero intero successivo ("Arrotonda al valore successivo"). Se il numero in virgola mobile è equidistante tra due numeri interi, viene fornito il numero pari. Il risultato viene memorizzato nel parametro OUT. Se si verifica un'eccedenza (overflow), ENO = 0.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	-	-	x	x	0	x	x	1

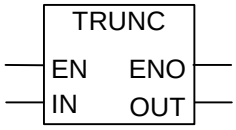
Esempio



Se E 0.0 = 1, il contenuto di MD8 viene letto come numero in virgola mobile e convertito nel numero intero successivo (a 32 bit). Il risultato di questa funzione "Arrotondare al valore successivo" viene memorizzato in MD12. L'uscita A 4.0 è "1", se si verifica un'eccedenza (overflow) o se l'istruzione non viene eseguita (E.0 = 0).

3.14 TRUNC Arrotonda senza resto un numero intero (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D	Numero in virgola mobile da convertire
OUT	DINT	E, A, M, L, D	Parte intera del valore IN

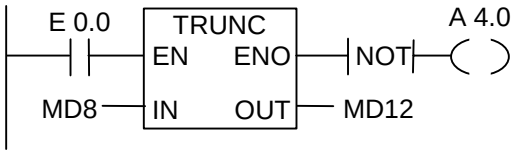
Descrizione dell'operazione

TRUNC (Arrotonda senza resto un numero intero (a 32 bit)) legge il contenuto specificato nel parametro IN come numero in virgola mobile, e converte questo numero in un numero intero (a 32 bit). Il risultato è il numero intero (a 32 bit) del numero in virgola mobile emesso dal parametro OUT. Se si verifica un'eccedenza (overflow), ENO = 0.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	-	-	x	x	0	x	x	1

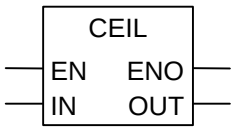
Esempio



Se E 0.0 = 1, il contenuto di MD8 viene letto come numero in virgola mobile, e quindi convertito in un numero intero (a 32 bit). Il risultato corrisponde alla parte intera di un numero in virgola mobile che viene memorizzata in MD12. L'uscita A 4.0 è "1" se si verifica un'eccedenza (overflow) o se l'istruzione non viene eseguita (E.0 = 0).

3.15 CEIL Genera da un numero in virgola mobile un numero intero superiore

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D	Valore di numero in virgola mobile da convertire
OUT	DINT	E, A, M, L, D	Il più piccolo numero intero (32 bit), maggiore del numero in virgola mobile

Descrizione dell'operazione

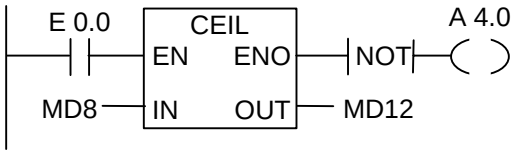
CEIL (Genera da un numero in virgola mobile un numero intero superiore) legge il contenuto specificato nel parametro IN come numero in virgola mobile, e lo converte in un numero intero (a 32 bit). Il risultato corrisponde al numero intero più piccolo che sia maggiore del numero in virgola mobile (Arrontondare secondo il modo "round to + infinity"). Se si verifica un'eccedenza (overflow), ENO = zero.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive *	X	-	-	X	X	0	X	X	1
Scrive **	0	-	-	-	-	0	0	0	1

* L'operazione viene eseguita (EN = 1)
** L'operazione non viene eseguita (EN = 0)

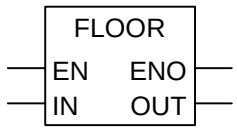
Esempio



Se E 0.0 = 1, il contenuto di MD8 viene letto come numero in virgola mobile e convertito con con la funzione "round to + infinity" in un numero intero (a 32 bit). Il risultato viene memorizzato in MD12. L'uscita A 4.0 è "1" se si verifica un'eccedenza (overflow) o se la operazione non viene eseguita (E 0.0 = 0).

3.16 FLOOR Genera da un numero in virgola mobile un numero intero inferiore

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D	Valore di numero in virgola mobile da convertire
OUT	DINT	E, A, M, L, D	Il più grande numero intero, minore del numero in virgola mobile

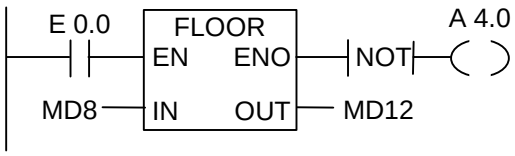
Descrizione dell'operazione

FLOOR (Genera da un numero in virgola mobile un numero intero inferiore) legge il contenuto specificato nel parametro IN come numero in virgola mobile, e lo converte in un numero intero (a 32 bit). Il risultato corrisponde al più grande numero intero che sia minore del numero in virgola mobile (Arrontondare secondo il modo "round to – infinity"). Se si verifica un'eccedenza (overflow), ENO = zero.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	-	-	x	x	0	x	x	1

Esempio



Se E 0.0 = 1, il contenuto di MD8 viene letto come numero in virgola mobile, e convertito in un numero intero (a 32 bit) con la funzione "round to – infinity". Il risultato viene memorizzato in MD12. L'uscita A 4.0 è "1" se si verifica un'eccedenza (overflow) o se l'operazione non viene eseguita (E0.0 = 0).

4 Operazioni di conteggio

4.1 Sommario delle operazioni di conteggio

Area di memoria

I contatori hanno un'area a loro riservata nella memoria della CPU. Questa area di memoria riserva una parola a 16 bit per ogni contatore. Il set delle operazioni KOP supporta 256 contatori. Le operazioni di conteggio sono le sole funzioni che hanno accesso all'area di memoria del contatore. Il campo di contatori dipende dalla CPU.

Valore di conteggio

I bit da 0 a 9 della parola del contatore contengono il valore di conteggio in codice binario. Quando si imposta un contatore, il valore di conteggio prestabilito viene trasferito dall'accumulatore nel contatore stesso. Il campo del contatore è compreso tra 0 e 999.

Si può modificare il valore di conteggio all'interno del campo utilizzando le operazioni seguenti:

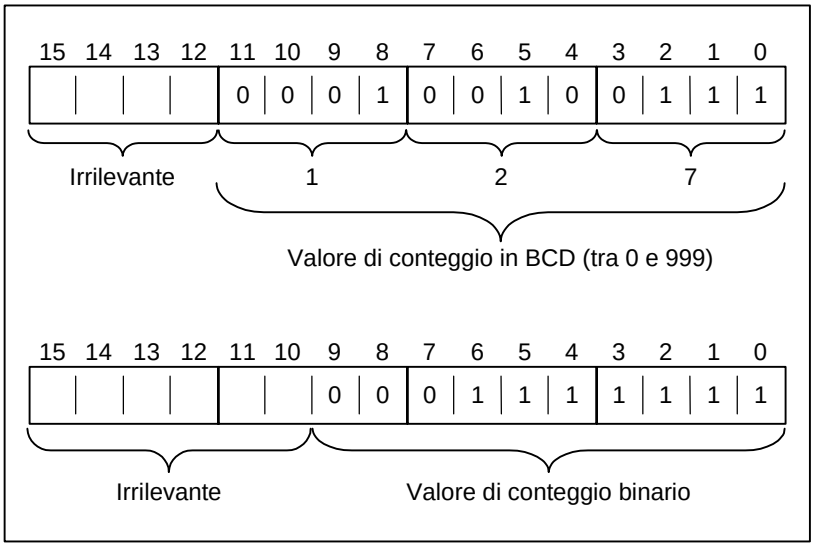
- ZÄHLER Parametrizza e conta in avanti/all'indietro
- Z_VORW Parametrizza e conta in avanti
- Z_RUECK Parametrizza e conta all'indietro
- ---(SZ) Imposta valore di conteggio
- ---(ZV) Conta in avanti
- ---(ZR) Conta all'indietro

Configurazione dei bit nel contatore

Per impostare un contatore con un determinato valore, occorre inserire un numero compreso tra 0 e 999 (p. es. 127), nel seguente formato: C#127. C# indica il formato decimale in codice binario.

I bit da 0 a 11 del contatore contengono il valore di conteggio nel formato decimale in codice binario (formato BCD: ogni set di quattro bit contiene il codice binario per un valore decimale).

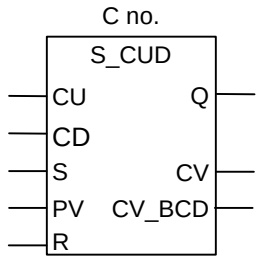
La figura mostra il contenuto del contatore dopo aver caricato il valore di conteggio 127, e il contenuto della cella del contatore dopo aver impostato il contatore.



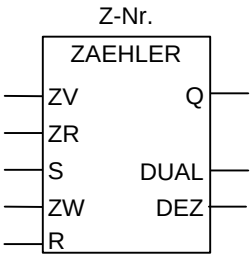
4.2 ZÄHLER Parametrizza e conta in avanti/all'indietro

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
C no.	Z-Nr.	COUNTER	Z	Numero di identificazione del contatore. Il campo dipende dalla CPU
CU	ZV	BOOL	E, A, M, L, D	Ingresso di conteggio in avanti
CD	ZR	BOOL	E, A, M, L, D	Ingresso di conteggio all'indietro
S	S	BOOL	E, A, M, L, D	Ingresso per preimpostare il contatore
PV	ZW	WORD	E, A, M, L, D oppure costante	Valore di conteggio immesso in forma C#<Valore> nell'intervallo tra 0 e 999
PV	ZW	WORD	E, A, M, L, D	Valore per preimpostare il contatore
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
CV	DUAL	WORD	E, A, M, L, D	Valore di conteggio attuale (esadecimale)
CV_BCD	DEZ	WORD	E, A, M, L, D	Valore di conteggio attuale (formato BCD)
Q	Q	BOOL	E, A, M, L, D	Stato del contatore

Descrizione dell'operazione

ZÄHLER (Parametrizzare e contare in avanti/all'indietro) viene preimpostato con un fronte di salita all'ingresso S con il valore dell'ingresso PV/ZW.

Il contatore viene incrementato di "1", se lo stato di segnale all'ingresso CU/ZV cambia da "0" a "1" e il valore del contatore è inferiore a "999".

Il contatore viene decrementato di "1", se all'ingresso CD/ZR vi è un fronte di salita e il valore del contatore è maggiore di "0". Se vi è un fronte di salita ad entrambi gli ingressi, vengono eseguite tutte e due le operazioni ed il valore di conteggio rimane invariato.

Se il contatore viene impostato e agli ingressi ZV/ZR l'RLC è =1, il contatore conteggerà in modo corrispondente nel ciclo successivo anche se non vi era stato un cambiamento di fronte.

Se all'ingresso R vi è 1 il contatore viene resettato. Resettare il contatore significa posizionare a zero il valore del contatore.

Lo stato di segnale all'uscita Q è "1", se il valore di conteggio è maggiore di 0, e 0 se il valore di conteggio è uguale a 0.

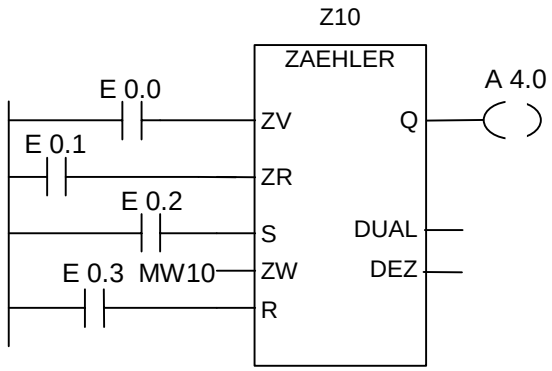
Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Nota

Evitare di utilizzare un contatore in diversi punti del programma, ma utilizzarlo in un punto solo (pericolo di errori di conteggio).

Esempio

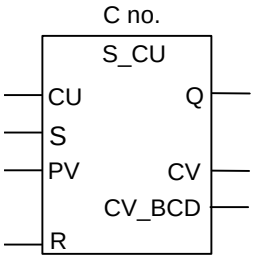


Con un cambiamento dello stato di segnale da "0" a "1" all'ingresso E 0.2, il contatore viene preimpostato con il valore di MW10. Se lo stato di segnale di E 0.0 cambia da "0" a "1", il valore del contatore Z10 viene incrementato di "1", eccetto quando il valore di Z10 è uguale a "999". Se E 0.1 cambia da "0" a "1", Z10 viene decrementato di "1", eccetto quando il valore di Z10 è uguale a zero. A 4.0 è "1", se Z10 è diverso da zero.

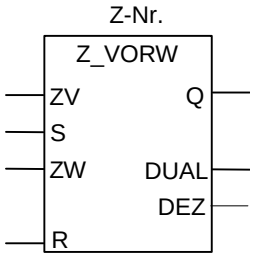
4.3 Z_VORW Parametrizza e conta in avanti

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
C no.	Nr. Z	COUNTER	Z	Numero di identificazione del contatore. Il campo dipende dalla CPU
CU	ZV	BOOL	E, A, M, L, D	Ingresso di conteggio in avanti
S	S	BOOL	E, A, M, L, D	Ingresso per preimpostare il contatore
PV	ZW	WORD	E, A, M, L, D oppure costante	Valore di conteggio immesso in forma C#<Valore> nell'intervallo tra 0 e 999
PV	ZW	WORD	E, A, M, L, D	Valore per preimpostare il contatore
PV	ZW	WORD	E, A, M, L, D	Valore per preimpostare il contatore
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
CV	DUAL	WORD	E, A, M, L, D	Valore di conteggio attuale (esadecimale)
CV_BCD	DEZ	WORD	E, A, M, L, D	Valore di conteggio attuale (formato BCD)
Q	Q	BOOL	E, A, M, L, D	Stato del contatore

Descrizione dell'operazione

Z_VORW (Parametrizza e conta in avanti) viene preimpostato con il valore dell'ingresso PV/ZW, se vi è un fronte di salita all'ingresso S .

Se all'ingresso R vi è un fronte di salita, il contatore viene resettato. Resettare il contatore significa posizionare a zero il valore del contatore.

Il contatore viene incrementato di "1", se lo stato di segnale all'ingresso CU/ZV cambia da "0" a "1", e il valore del contatore è inferiore a "999".

Se il contatore viene impostato e all'ingresso ZV l'RLC è =1, il contatore conteggerà in modo corrispondente nel ciclo successivo anche se non vi era stato un cambiamento di fronte.

Lo stato di segnale all'uscita Q è "1", se il valore di conteggio è maggiore di 0; e 0 se il valore di conteggio è uguale a 0.

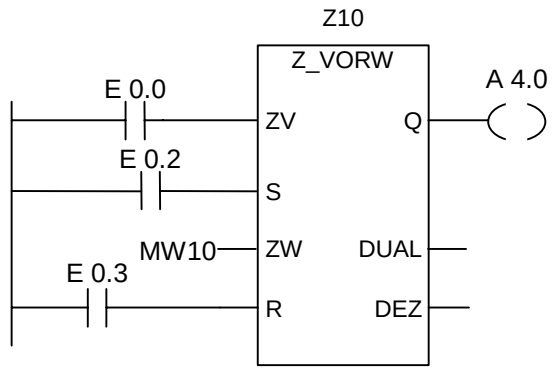
Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Nota

Evitare di utilizzare un contatore in diversi punti del programma, ma utilizzarlo in un punto solo (pericolo di errori di conteggio).

Esempio

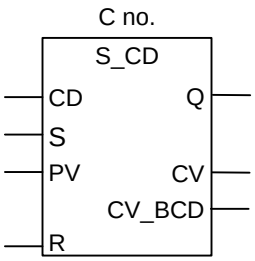


Con un cambiamento dello stato di segnale da "0" a "1" all'ingresso E 0.2, il contatore viene preimpostato con il valore di MW10. Se lo stato di segnale di E 0.0 cambia da "0" a "1", il valore del contatore Z10 viene incrementato di "1", eccetto quando il valore di Z10 è uguale a "999". A 4.0 è "1", se Z10 è diverso da zero.

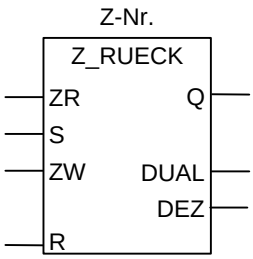
4.4 Z_RUECK Parametrizza e conta all'indietro

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
C no.	Z-Nr.	COUNTER	Z	Numero di identificazione del contatore. Il campo dipende dalla CPU
CD	ZR	BOOL	E, A, M, L, D	Ingresso di conteggio all'indietro
S	S	BOOL	E, A, M, L, D	Ingresso per preimpostare il contatore
PV	ZW	WORD	E, A, M, L, D oppure costante	Valore di conteggio immesso in forma C#<Valore> nell'intervallo tra 0 e 999
PV	ZW	WORD	E, A, M, L, D	Valore per preimpostare il contatore
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
CV	DUAL	WORD	E, A, M, L, D	Valore di conteggio attuale (esadecimale)
CV_BCD	DEZ	WORD	E, A, M, L, D	Valore di conteggio attuale (formato BCD)
Q	Q	BOOL	E, A, M, L, D	Stato del contatore

Descrizione dell'operazione

Z_RUECK (Parametrizzare e contare all'indietro) viene preimpostato con il valore dell'ingresso PV/ZW, se vi è un fronte di salita all'ingresso S.

Se all'ingresso R vi è 1, il contatore viene resettato. Resettare il contatore significa posizionare a zero il suo valore.

Se il contatore viene impostato e all'ingresso ZR l'RLC è =1, il contatore conteggerà in modo corrispondente nel ciclo successivo anche se non vi era stato un cambiamento di fronte.

Il contatore viene decrementato di "1", se lo stato di segnale all'ingresso ZR cambia da "0" a "1", ed il valore del contatore è maggiore di 0.

Lo stato di segnale all'uscita Q è "1", se il valore di conteggio è maggiore di 0; e 0 se il valore di conteggio è uguale a 0.

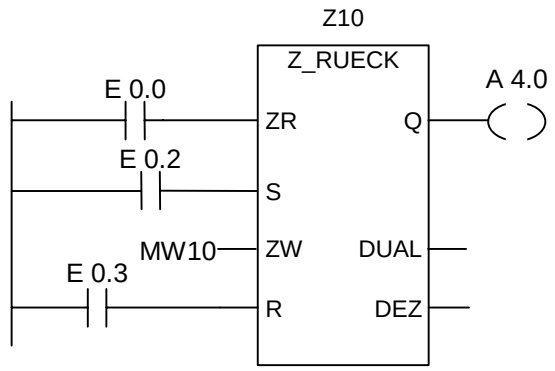
Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Nota

Evitare di utilizzare un contatore in diversi punti del programma, ma utilizzarlo in un punto solo (pericolo di errori di conteggio).

Esempio



Un cambiamento dello stato di segnale da "0" a "1" all'ingresso E 0.2 setta il contatore con il valore di MW10. Se lo stato di segnale di E 0.0 cambia da "0" a "1", il valore del contatore Z10 viene decrementato di "1", eccetto quando il valore di Z10 è uguale a "0". A 4.0 è "1", se Z10 è diverso da zero.

4.5 ---(SZ) Imposta valore di conteggio

Simbolo

Inglese	**Tedesco**
<C no.>	<Nr. Z>
---(SC)	**---(SZ)**
<Valore di default>	<Valore di default>

Parametro Inglese	**Parametro Tedesco**	**Tipo di dati**	**Area di memoria**	**Descrizione**
<C no.>	<Nr. Z>	COUNTER	Z	Numero del contatore che deve essere preimpostato
<Valore di default>	<Valore di default>	WORD	E, A, M, L, D	Valore di default BCD (0-999)

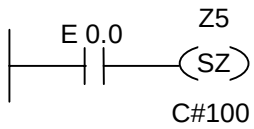
Descrizione dell'operazione

---(SZ) (Imposta valore di conteggio) viene eseguita solo se il RLC ha un fronte di salita. Il valore di default viene quindi trasferito al contatore specificato.

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	x	-	0

Esempio



Il contatore Z5 viene preimpostato con il valore di default "100" se all'ingresso E 0.0 c'è un fronte di salita (cambio da "0" a "1"). Se non è presente un fronte di salita, il valore del contatore Z5 rimane invariato.

4.6 ---(ZV) Conta in avanti

Simbolo

Inglese	Tedesco
<C no.>	<Nr. Z>
---(CU)	**---(ZV)**

Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
<C no.>	<Nr. Z>	COUNTER	Z	Numero di identificazione del contatore; il campo dipende dalla CPU

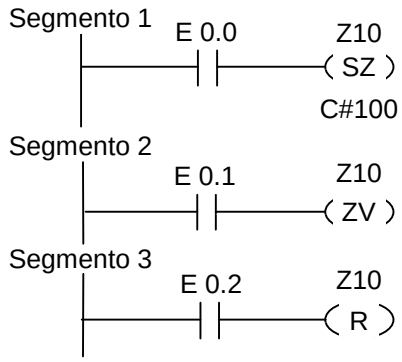
Descrizione dell'operazione

---(ZV) (Conta in avanti) incrementa di "1" il valore del contatore specificato se RLC ha un fronte di salita, e il valore del contatore è minore di "999". Se RLC non ha un fronte di salita oppure se il contatore ha già il valore "999", il valore del contatore rimane inalterato.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	-	-	0

Esempio



Se lo stato di segnale di E0.0 passa da "0" a "1" (fronte di salita in RLC), il valore di default "100" viene caricato nel contatore Z10.

Se lo stato di segnale di E 0.1 passa da "0" a "1" (fronte di salita in RLC), il valore del contatore Z10 viene incrementato di "1", a meno che il valore del contatore sia uguale a "999". Se non vi è un fronte di salita in RLC, il valore del contatore Z10 rimane invariato.

Se lo stato di segnale di E0.2 = 1, il contatore viene resettato a "0".

4.7 ---(ZR) Conta all'indietro

Simbolo

Inglese	**Tedesco**
<C no.>	<Nr. Z>
---(CD)	**---(ZR)**

Parametro Inglese	**Parametro Tedesco**	**Tipo di dati**	**Area di memoria**	**Descrizione**
<C no.>	<Nr. Z>	COUNTER	Z	Numero di identificazione del contatore; il campo dipende dalla CPU

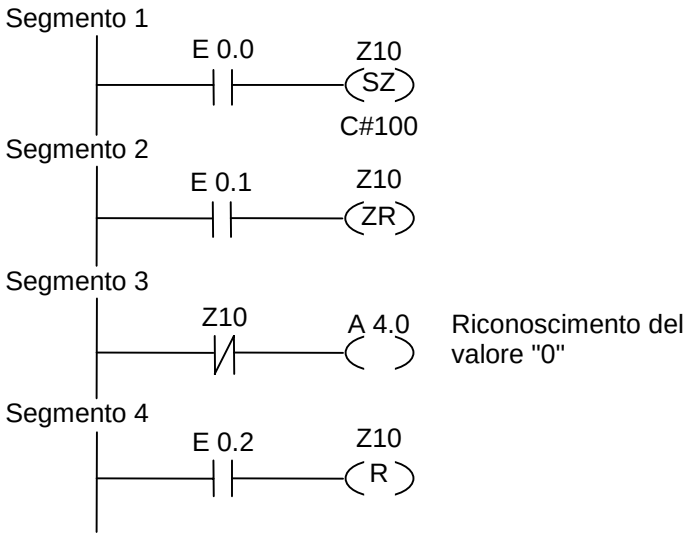
Descrizione dell'operazione

---(ZR) (Conta all'indietro) decrementa di "1" il valore del contatore specificato se il RLC ha un fronte di salita e il valore del contatore è maggiore di "0". Se RLC non ha un fronte di salita, o se il contatore ha già il valore "0", il valore del contatore rimane inalterato.

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	-	-	0

Esempio



Se lo stato di segnale di E 0.0 passa da "0" a "1" (fronte di salita in RLC), il valore di default "100" viene caricato nel contatore Z10.

Se lo stato di segnale di E 0.1 passa da "0" a "1" (fronte di salita in RLC), il valore del contatore Z10 viene decrementato di 1", a meno che il valore del contatore sia già "0". Se non vi è un fronte di salita in RLC, il valore del contatore Z10 rimane invariato.

Se il valore del contatore è zero, viene attivata A 4.0.

Se lo stato di segnale di E 0.2 = 1, il contatore viene resettato a "0".

5 Operazioni di blocco dati

5.1 ---(OPN) Apri un blocco dati

Simbolo internazionale

<N. DB> oppure <N. DI>
---(OPN)

Parametro	Tipo di dati	Area di memoria	Descrizione
<N. DB> <N. DI>	BLOCK_DB	DB, DI	Numero del DB/DI; il campo dipende dalla CPU

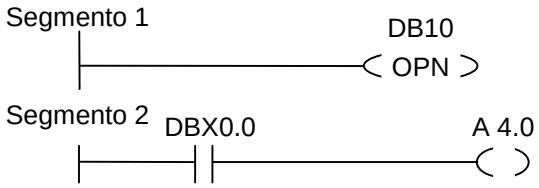
Descrizione dell'operazione

---(OPN) (Apri un blocco dati) apre un blocco dati (DB) oppure un blocco dati di istanza (DB di istanza) come primo o secondo blocco dati. L'operazione ---(OPN) è un richiamo incondizionato di un blocco dati. Il numero del blocco dati viene trasmesso nel registro DB o DI. I successivi comandi DB e DI accedono ai rispettivi blocchi, a seconda del contenuto dei registri.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	-	-	-	-

Esempio



Il blocco di dati 10 (DB10) viene aperto. L'indirizzo del contatto (DBX0.0) si riferisce al bit 0 del byte di dati 0 del record di dati attuale che si trova in DB10. Lo stato di segnale di questo bit viene assegnato all'uscita A 4.0.

6 Operazioni di salto

6.1 Sommario delle operazioni di salto

Descrizione

Questa operazione può venire utilizzata in tutti i blocchi logici: blocchi organizzativi (OB), blocchi funzionali (FB), e funzioni (FC).

Sono disponibili le seguenti operazioni di salto:

- ---(JMP)--- Salto assoluto
- ---(JMP)--- Se 1 salta in un FB (condizionato)
- ---(JMPN) Se 0 salta in un FB

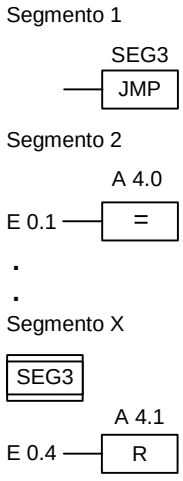
Etichetta come operando

L'operando di un'operazione di salto è un'etichetta. L'etichetta di salto indica la destinazione in cui il programma deve saltare.

L'etichetta viene inserita sopra la bobina dell'operazione di salto. Un'etichetta è costituita da un massimo di quattro caratteri. Il primo carattere deve essere una lettera dell'alfabeto; gli altri caratteri possono essere lettere o numeri (ad esempio, SEG3).

Etichetta come destinazione

L'etichetta di destinazione deve trovarsi all'inizio di una rete. L'etichetta di destinzione viene inserita all'inizio della rete selezionando LABEL dal box di selezione di KOP. Appare un box vuoto. In questo box, si digita il nome dell'etichetta.



6.2 ---(JMP)--- Salto assoluto

Simbolo

<Etichetta di salto>

---(JMP)

Descrizione dell'operazione

---(JMP) (Se 1 salta in un FB) funziona come un salto assoluto se nessun ulteriore elemento è interposto tra il percorso di corrente a sinistra e l'operazione (vedi esempio).

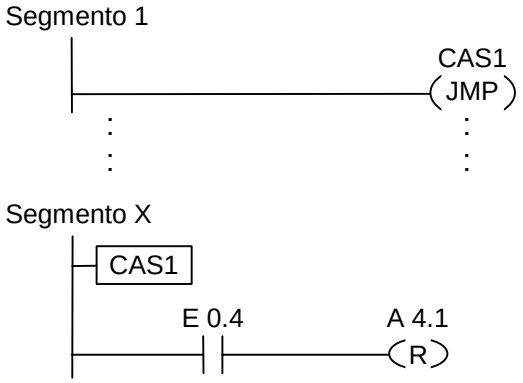
Per ogni - - - (JMP) deve essere presente anche una destinazione (etichetta di salto).

Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite.

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	-	-	-	-

Esempio



Il salto viene eseguito sempre, e vengono saltate le operazioni tra l'operazione di salto e l'etichetta.

6.3 ---(JMP)--- Se 1 salta in un FB (condizionato)

Simbolo

<Etichetta di salto>

---(JMP)

Descrizione dell'operazione

---(JMP) (Se 1 salta in un FB) funziona come un salto condizionato de il RLC della combinazione presedente è "1".

Per ogni ---(JMP) deve essere presente anche una destinazione (etichetta di salto).

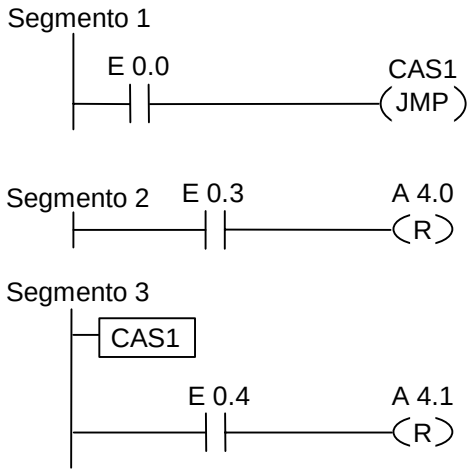
Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite!

Se non viene eseguito un salto condizionato, il RLC dopo l'operazione di salto passa a "1".

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	1	1	0

Esempio



Se E 0.0 = 0, viene eseguito il salto all'etichetta CAS1. A causa del salto, l'operazione per resettare l'uscita A 4.0 non viene eseguita, anche se E 0.3 = 1.

6.4 ---(JMPN) Se 0 salta in un FB (condizionato)

Simbolo

<Etichetta di salto>

---(JMPN)

Descrizione dell'operazione

---(JMPN) (Se 0 salta in un FB) funziona come salto condizionato, se il RLC della combinazione pecedente è "0".

Per ogni ---(JMPN) deve essere presente anche una destinazione (LABEL).

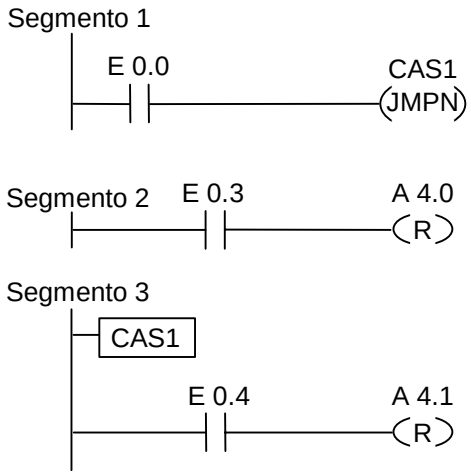
Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite!

Se non viene eseguito un salto condizionato, il RLC dopo l'operazione di salto passa a "1".

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	1	1	0

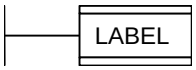
Esempio



Se E 0.0 = 0, viene eseguito il salto all'etichetta CAS1. A causa del salto, l'operazione per resettare l'uscita A 4.0 non viene eseguita, anche se E 0.3 = 1.

6.5 LABEL Etichetta di salto

Simbolo

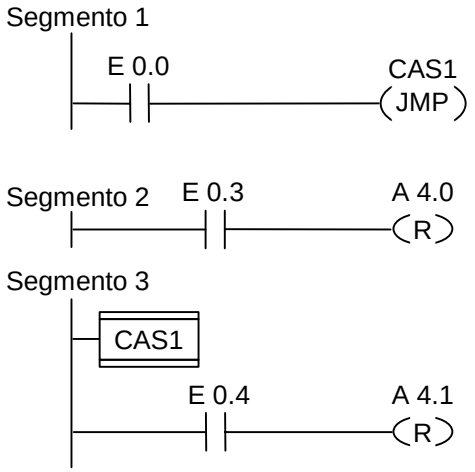


Descrizione

Etichetta di salto caratterizza la destinazione di un'operazione di salto. L'etichetta di salto consista in 4 caratteri - primo carattere: lettera, resto: lettera o alfanumerico.

Per ogni ---(JMP) o ---(JMPN) deve essere presente anche una etichetta di salto (LABEL).

Esempio



Se E 0.0 = 1, viene eseguito il salto all'etichetta CAS1. A causa del salto, l'operazione per resettare l'uscita A 4.0 non viene eseguita, anche se E 0.3 = 1.

7 Operazioni matematiche con i numeri interi

7.1 Sommario delle operazioni matematiche con i numeri interi

Descrizione

Con le operazioni matematiche con i numeri interi, le seguenti funzioni possono essere eseguite con **due** numeri interi (16 bit, 32 bit):

- ADD_I Somma numeri interi a 16 bit
- SUB_I Sottrai numeri interi a 16 bit
- MUL_I Moltiplica numeri interi a 16 bit
- DIV_I Dividi numeri interi a 16 bit

- ADD_DI Somma numeri interi a 32 bit
- SUB_DI Sottrai numeri interi a 32 bit
- MUL_DI Moltiplica numeri interi a 32 bit
- DIV_DI Dividi numeri interi a 32 bit
- MOD_DI Ricava resto della divisione (a 32 bit)

7.2 Valutazione dei bit nella parola di stato con operazioni con numeri in virgola fissa

Descrizione

Le operazioni di calcolo di base influenzano i bit seguenti nella parola di stato:

- A1 e A0
- OV
- OS

Le tabelle seguente riportano lo stato di segnale dei bit della parola di stato per i risultati delle operazioni con i numeri in virgola fissa (16 bit e 32 bit).

Campo di valori valido	A1	A0	OV	OS
0 (zero)	0	0	0	*
16 bit: –32 768 < =risultato < 0 (numero negativo) 32 bit: –2 147 483 648 <=risultato < 0 (numero negativo)	0	1	0	*
16 bit: 32 767 > =risultato > 0 (numero positivo) 32 bit: 2 147 483 647 > = risultato > 0 (numero positivo)	1	0	0	*

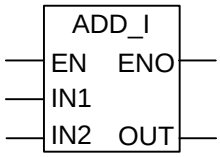
* Il bit OS non è influenzato dal risultato dell'operazione.

Campo di valori non valido	A1	A0	OV	OS
Superamento negativo del campo con addizione 16 bit: risultato = –65536 32 bit: risultato = –4 294 967 296	0	0	1	1
Superamento negativo del campo con moltiplicazione 16 bit: risultato < –32 768 (numero negativo) 32 bit: risultato < –2 147 483 648 (numero negativo)	0	1	1	1
Superamento positivo del campo con addizione, sottrazione 16 bit: risultato > 32 767 (numero positivo) 32 bit: risultato > 2 147 483 647 (numero positivo)	0	1	1	1
Superamento positivo del campo con moltiplicazione, divisione 16 bit: risultato > 32 767 (numero positivo) 32 bit: risultato > 2 147 483 647 (numero positivo)	1	0	1	1
Superamento negativo del campo con addizione, sottrazione 16 bit: risultato < –32 768 (numero negativo) 32 bit: risultato < –2 147 483 648 (numero negativo)	1	0	1	1
Divisione per zero	1	1	1	1

Operazione	A1	A0	OV	OS
+D: risultato = –4 294 967 296	0	0	1	1
/D o MOD: divisione per 0	1	1	1	1

7.3 ADD_I Somma numeri interi a 16 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	INT	E, A, M, L, D o costante	Primo valore della somma
IN2	INT	E, A, M, L, D o costante	Secondo valore della somma
OUT	INT	E, A, M, L, D	Risultato della somma

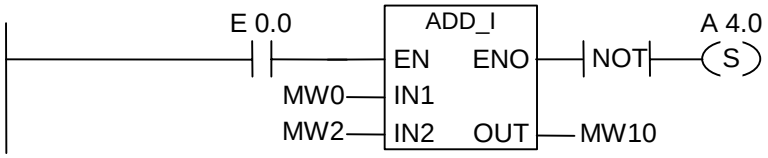
Descrizione dell'operazione

ADD_I (Somma numeri interi a 16 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN1 e IN2 vengono sommati. Il risultato può essere letto all'uscita O. Se il risultato non rientra nel campo permesso per i numeri interi (16 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

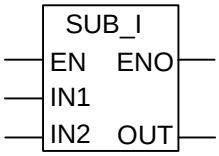
Esempio



Il box ADD_I viene attivato se E 0.0 = 1. Il risultato della somma MW0 + MW2 viene emesso da MW10. Se il risultato non rientra nel campo permesso per un numero intero (a 16 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.4 SUB_I Sottrai numeri interi a 16 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	INT	E, A, M, L, D o costante	Primo valore della sottrazione
IN2	INT	E, A, M, L, D o costante	Valore da sottrarre al primo valore
OUT	INT	E, A, M, L, D	Risultato della sottrazione

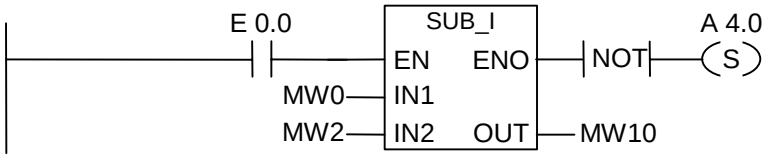
Descrizione dell'operazione

SUB_I (Sottrai numeri interi a 16 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN2 viene sottratto da IN1. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (a 16 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

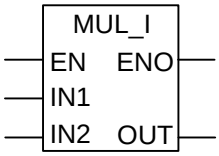
Esempio



Il box SUB_I viene attivato se E 0.0 = 1. Il risultato della sottrazione MW0 - MW2 viene emesso da MW10. Se il risultato non rientra nel campo permesso per un numero intero (a 16 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.5 MUL_I Moltiplica numeri interi a 16 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	INT	E, A, M, L, D o costante	Primo valore della moltiplicazione
IN2	INT	E, A, M, L, D o costante	Secondo valore della moltiplicazione
OUT	DINT	E, A, M, L, D	Risultato della moltiplicazione

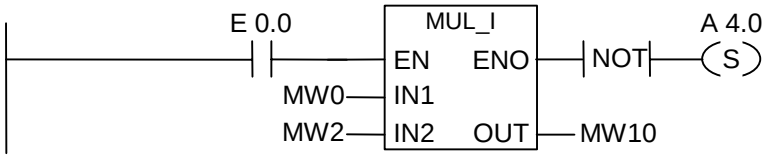
Descrizione dell'operazione

MUL_I (Moltiplica numeri interi a 16 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN1 e IN2 vengono moltiplicati. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (16 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

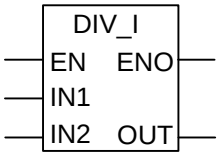
Esempio



Il box MUL_I viene attivato se E 0.0 = 1. Il risultato della moltiplicazione MW0 x MW2 viene emesso da MW10. Se il risultato non rientra nel campo permesso per un numero intero (a 16 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.6 DIV_I Dividi numeri interi a 16 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	INT	E, A, M, L, D o costante	Dividendo
IN2	INT	E, A, M, L, D o costante	Divisore
OUT	INT	E, A, M, L, D	Risultato della divisione

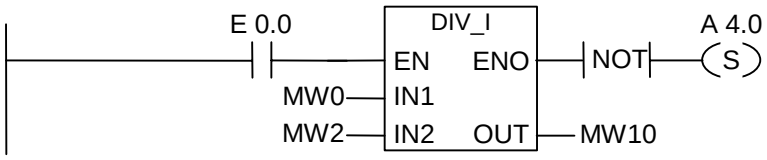
Descrizione dell'operazione

DIV_I (Dividi numeri interi a 16 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN1 viene diviso per IN2. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (a 16 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

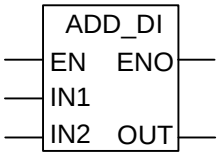
Esempio



Il box DIV_I viene attivato se E 0.0 = 1. Il quoziente della divisione MW0 diviso MW2 viene emesso da MW10. Se il risultato non rientra nel campo permesso per un numero intero (a 16 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.7 ADD_DI Somma numeri interi a 32 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DINT	E, A, M, L, D o costante	Primo valore della somma
IN2	DINT	E, A, M, L, D o costante	Secondo valore della somma
OUT	DINT	E, A, M, L, D	Risultato della somma

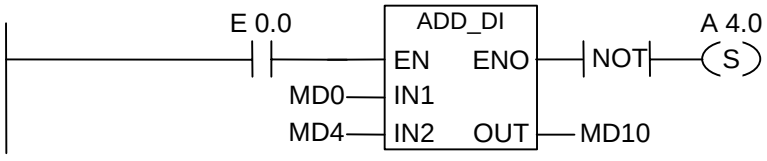
Descrizione dell'operazione

ADD_DI (Somma numeri interi a 32 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN1 e IN2 vengono sommati. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (a 32 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

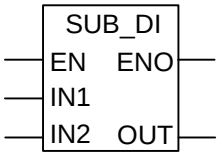
Esempio



Il box ADD_DI viene attivato se E 0.0 = 1. Il risultato della somma MD0 + MD4 viene emesso da MD10. Se il risultato non rientra nel campo permesso per un numero intero (a 32 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.8 SUB_DI Sottrai numeri interi a 32 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DINT	E, A, M, L, D o costante	Primo valore della sottrazione
IN2	DINT	E, A, M, L, D o costante	Valore da sottrarre al primo valore
OUT	DINT	E, A, M, L, D	Risultato della sottrazione

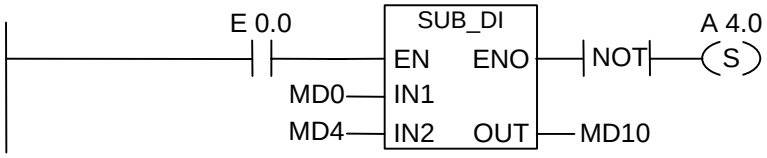
Descrizione dell'operazione

SUB_DI (Sottrai numeri interi a 32 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN2 viene sottratto da IN1. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (32 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

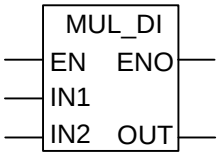
Esempio



Il box SUB_DI viene attivato se E 0.0 = 1. Il risultato della sottrazione MD0 - MD4 viene emesso da MD10. Se il risultato non rientra nel campo permesso per un numero intero (a 32 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.9 MUL_DI Moltiplica numeri interi a 32 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DINT	E, A, M, L, D o costante	Primo valore della moltiplicazione
IN2	DINT	E, A, M, L, D o costante	Secondo valore della moliplicazione
OUT	DINT	E, A, M, L, D	Risultato della moltiplicazione

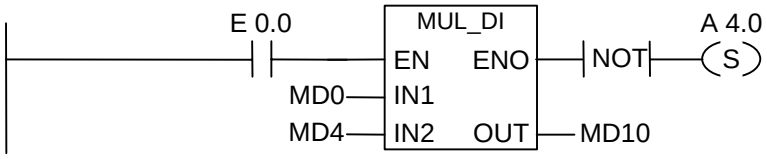
Descrizione dell'operazione

MUL_DI (Moltiplica numeri interi a 32 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN1 e IN2 vengono moltiplicati. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (a 32 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

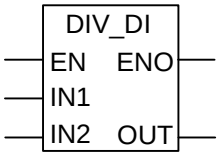
Esempio



Il box MUL_DI viene attivato se E 0.0 = 1. Il risultato della moltiplicazione MD0 x MD4 viene emesso da MD10. Se il risultato non rientra nel campo permesso per un numero intero (32 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.10 DIV_DI Dividi numeri interi a 32 bit

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DINT	E, A, M, L, D o costante	Dividendo
IN2	DINT	E, A, M, L, D o costante	Divisore
OUT	DINT	E, A, M, L, D	Risultato della divisione

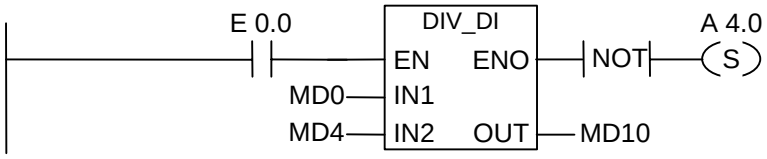
Descrizione dell'operazione

DIV_DI (Dividi numeri interi a 32 bit) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN1 viene diviso per IN2. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (a 32 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

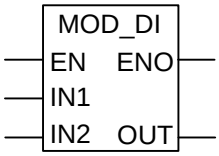
Esempio



Il box DIV_DI viene attivato se E 0.0 = 1. Il risultato della divisione MD0 diviso MD4 viene emesso da MD10. Se il risultato non rientra nel campo permesso per un numero intero (a 32 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

7.11 MOD_DI Ricava resto della divisione (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DINT	E, A, M, L, D o costante	Dividendo
IN2	DINT	E, A, M, L, D o costante	Divisore
OUT	DINT	E, A, M, L, D	Risultato della divisione

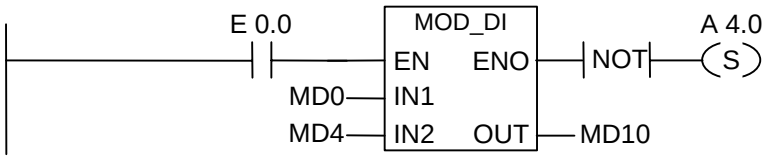
Descrizione dell'operazione

MOD_DI (Ricava resto della divisione (a 32 bit)) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". IN1 viene diviso per IN2. Il resto (frazione) della divisione può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per i numeri interi (32 bit), il bit OV ed il bit OS sono "1", ed ENO è "0"; non vengono quindi eseguite altre operazioni collegate tramite ENO (collegamento in cascata) dopo questa operazione matematica.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	x	0	x	x	1

Esempio



Il box MOD_DI viene attivato se E 0.0 = 1. Il resto (frazione) della divisione MD0 diviso MD4 viene emesso da MD10. Se il risultato non rientra nel campo permesso per un numero intero (a 32 bit), o se lo stato di segnale E 0.0 = 0, l'uscita A 4.0 viene impostata.

8 Operazioni matematiche in virgola mobile

8.1 Sommario delle operazioni matematiche con i numeri mobile

Descrizione

I numeri IEEE-FP in virgola mobile a 32 bit appartengono al tipo di dati denominato ”REAL”. Si possono adoperare le operazioni matematiche coi numeri in virgola mobile per effettuare le seguenti operazioni adoperando **due** numeri IEEE-FP in virgola mobile a 32 bit:

- ADD_R Somma
- SUB_R Sottrazione
- MUL_R Moltiplicazione
- DIV_R Divisione

Con l’aritmetica in virgola mobile, le seguenti funzioni possono essere eseguite con **un** numero in virgola mobile (32 bit, IEEE-FP):

- Formazione del valore assoluto (ABS)
- Formazione del quadrato (SQR) e della radice quadrata (SQRT)
- Formazione del logaritmo naturale (LN)
- Formazione del valore esponenziale (EXP) su base e (= 2,71828...)
- Formazione delle seguenti funzioni trigonometriche di un angolo rappresentato da un numero in virgola mobile (32 bit, IEEE-FP):
  - seno (SIN) ed arcoseno (ASIN)
  - coseno (COS) ed arcocoseno (ACOS)
  - tangente (TAN) ed arcotangente (ATAN)

8.2 Valutazione dei bit nella parola di stato con operazioni in virgola mobile

Descrizione

Le operazioni di calcolo di base influenzano i seguenti bit nella parola di stato:

- A1 e A0
- OV
- OS

Le tabelle seguente riportano lo stato di segnale dei bit della parola di stato per i risultati delle operazioni con i numeri in virgola mobile (a 32 bit).

Campo di validità	**A1**	**A0**	**OV**	**OS**
+0, -0 (zero)	0	0	0	*
-3.402823E+38 < risultato < -1.175494E-38 (numero negativo)	0	1	0	*
+1.175494E-38 < risultato < +3.402823E+38 (numero positivo)	1	0	0	*

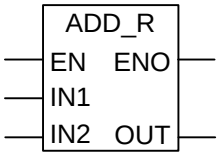
* Il bit OS non è influenzato dal risultato dell'operazione.

Campo di non validità	**A1**	**A0**	**OV**	**OS**
superamento negativo di capacità -1.175494E-38 < risultato < -1.401298E-45 (numero negativo)	0	0	1	1
superamento negativo di capacità +1.401298E-45 < risultato < +1.175494E-38 (numero positivo)	0	0	1	1
overflow Risultato < -3.402823E+38 (numero negativo)	0	1	1	1
overflow Risultato > 3.402823E+38 (numero positivo)	1	0	1	1
Numero in virgola mobile non valido od operazione non permessa (valore di ingresso al di fuori del campo di validità dei valori)	1	1	1	1

8.3 Operazioni di base

8.3.1 ADD_R Somma numeri in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	REAL	E, A, M, L, D o costante	Primo valore della somma
IN2	REAL	E, A, M, L, D o costante	Secondo valore della somma
OUT	REAL	E, A, M, L, D	Risultato della somma

Descrizione dell'operazione

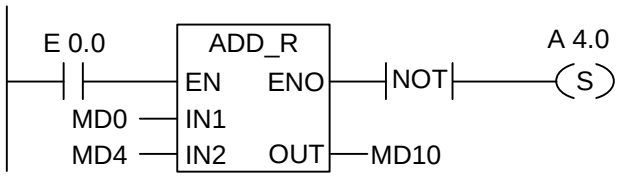
ADD_R (Somma numeri in virgola mobile) addiziona IN1 e IN2 quando vi è uno stato di "1" all'ingresso di abilitazione (EN). Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero in virgola mobile (overflow o superamento negativo di capacità), il bit OV ed il bit OS sono "1" ed ENO è "0"; dopo questa operazione matematica, non vengono quindi eseguite altre funzioni collegate tramite ENO (collegamento in cascata).

Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

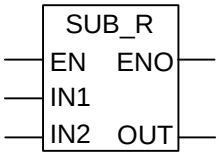
Esempio



Il box ADD_R viene attivato se E 0.0 = 1. Il risultato dell'addizione MD0 + MD4 viene emesso da MD10. Se il risultato non rientra nel campo permesso per un numero in virgola mobile, o se l'istruzione non viene elaborata (E 0.0 = 0), l'uscita A 4.0 viene settata.

8.3.2 SUB_R Sottrai numeri in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	REAL	E, A, M, L, D o costante	Primo valore della sottrazione
IN2	REAL	E, A, M, L, D o costante	Secondo valore della sottrazione
OUT	REAL	E, A, M, L, D	Risultato della sottrazione

Descrizione dell'operazione

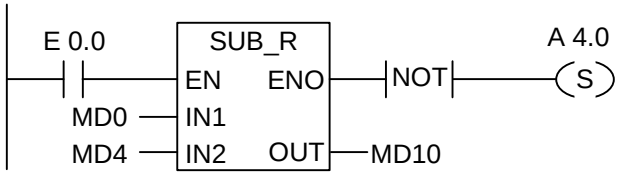
SUB_R (Sottrai numeri in virgola mobile) viene attivato dallo stato di "1" all'ingresso di abilitazione (EN). IN2 viene sottratto da IN1. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero in virgola mobile (eccedenza o superamento negativo di capacità), il bit OV ed il bit OS sono "1", e ENO è "0"; dopo questa operazione matematica, non vengono quindi eseguite altre funzioni collegate tramite ENO (collegamento in cascata).

Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

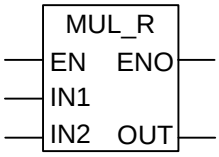
Esempio



Il box SUB_R viene attivato se E 0.0 = 1. Il risultato della sottrazione MD0 - MD4 viene emesso in MD10. Se il risultato non rientra nel campo permesso per un numero in virgola mobile, o se l'istruzione non viene elaborata, l'uscita A 4.0 viene settata.

8.3.3 MUL_R Moltiplica numeri in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	REAL	E, A, M, L, D o costante	Primo valore della moltiplicazione
IN2	REAL	E, A, M, L, D o costante	Secondo valore della moltiplicazione
OUT	REAL	E, A, M, L, D	Risultato della moltiplicazione

Descrizione dell'operazione

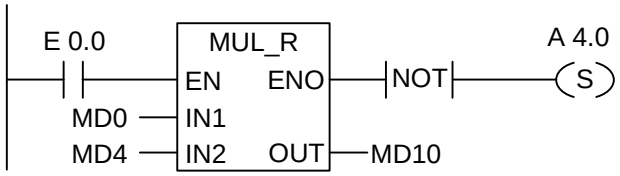
MUL_R (Moltiplica numeri in virgola mobile) viene attivato dallo stato di "1" all'ingresso di abilitazione (EN). IN1 e IN2 vengono moltiplicati. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero in virgola mobile (eccedenza o superamento negativo di capacità), il bit OV ed il bit OS sono "1", e ENO è "0"; dopo questa operazione matematica, non vengono quindi eseguite altre funzioni collegate tramite ENO (collegamento in cascata).

Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

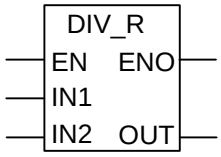
Esempio



Il box MUL_R viene attivato se E 0.0 = 1. Il risultato della moltiplicazione MD0 x MD4 viene emesso in MD10. Se il risultato non rientra nel campo permesso per un numero in virgola mobile o se l'istruzione non viene elaborata, l'uscita A 4.0 viene settata.

8.3.4 DIV_R Dividi numeri in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	REAL	E, A, M, L, D o costante	Dividendo
IN2	REAL	E, A, M, L, D o costante	Divisore
OUT	REAL	E, A, M, L, D	Risultato della divisione

Descrizione dell'operazione

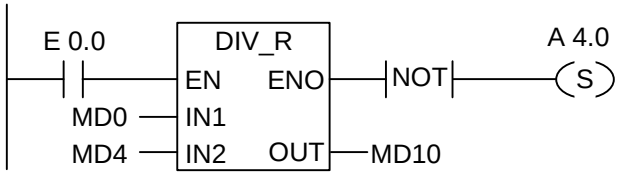
DIV_R (Dividi numeri in virgola mobile) viene attivato dallo stato di "1" all'ingresso di abilitazione (EN). IN1 viene diviso per IN2. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero in virgola mobile (eccedenza e superamento negativo di capacità), il bit OV ed il bit OS sono "1", e ENO è "0"; dopo questa operazione matematica, non vengono quindi eseguite altre funzioni collegate tramite ENO (collegamento in cascata).

Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

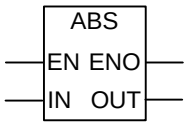
Esempio



Il box DIV_R viene attivato se E 0.0 = 1. Il risultato della divisione MD0 diviso MD4 viene emesso in MD10. Se il risultato non rientra nel campo permesso per un numero in virgla mobile o se l'istruzione non viene elaborata, l'uscita A 4.0 viene settata.

8.3.5 ABS Formazione del valore assoluto di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: valore assoluto del numero in virgola mobile

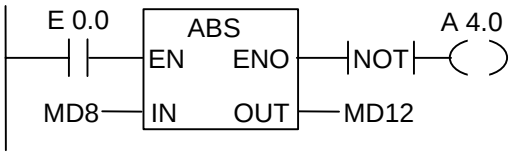
Descrizione dell'operazione

ABS (Forma valore assoluto di un numero in virgola mobile) consente di formare il valore assoluto di un numero in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	1	-	-	-	-	0	1	1	1

Esempio



Se E 0.0 = 1, il valore assoluto di MD8 viene emesso in MD12.

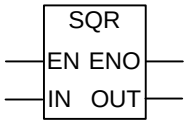
MD8 = + 6,234 risulta in MD12 =6,234.

L'uscita A 4.0 è "1", se la conversione non viene eseguita (ENO = EN = 0).

8.4 Operazioni avanzati

8.4.1 SQR Formazione del quadrato di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: quadrato del numero in virgola mobile

Descrizione dell'operazione

SQR (Formazione del quadrato di un numero in virgola mobile) calcola il quadrato di un numero in virgola mobile.

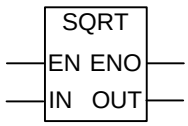
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.2 SQRT Formazione della radice quadrata di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: radice quadrata del numero in virgola mobile

Descrizione dell'operazione

SQRT (Formazione della radice quadrata di un numero in virgola mobile) calcola la radice quadrata di un numero in virgola mobile. Questa operazione produce un risultato positivo se l'operando è maggiore di "0". Unica eccezione: la radice quadrata di -0 è -0.

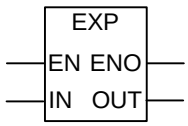
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.3 EXP Formazione del valore esponenziale di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: valore esponenziale del numero in virgola mobile

Descrizione dell'operazione

EXP (Formazione del valore esponenziale di un numero in virgola mobile) calcola il valore esponenziale di un numero in virgola mobile su base e (=2,71828...).

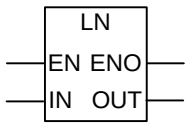
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.4 LN Formazione del logaritmo naturale di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: logaritmo naturale di un numero in virgola mobile

Descrizione dell'operazione

LN (Formazione del logaritmo naturale di un numero in virgola mobile) calcolano il logaritmo naturale di un numero in virgola mobile.

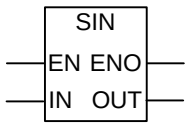
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.5 SIN Formazione del seno di un angolo come numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: seno di un numero in virgola mobile

Descrizione dell'operazione

SIN (Formazione del seno di un angolo come numero in virgola mobile) calcola il seno di un angolo come numero in virgola mobile. Il numero in virgola mobile rappresenta un angolo misurato in radianti.

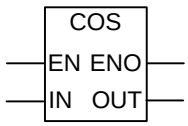
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.6 COS Formazione del coseno di un angolo come numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: coseno di un numero in virgola mobile

Descrizione dell'operazione

COS (Formazione del coseno di un angolo come numero in virgola mobile) calcola il coseno di un angolo come numero in virgola mobile. Il numero in virgola mobile rappresenta un angolo misurato in radianti.

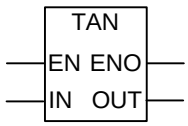
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.7 TAN Formazione della tangente di un angolo come numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: tangente di un numero in virgola mobile

Descrizione dell'operazione

TAN (Formazione della tangente di un angolo come numero in virgola mobile) calcola la tangente di un angolo di un numero in virgola mobile. Il numero in virgola mobile rappresenta un angolo misurato in radianti.

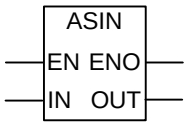
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.8 ASIN Formazione dell'arcoseno di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: arcoseno del numero in virgola mobile

Descrizione dell'operazione

ASIN (Formazione dell'arcoseno di un numero in virgola mobile) calcola l'arcoseno di un numero in virgola, il cui campo di definizione è -1 <= valore di ingresso <= 1. Il risultato rappresenta un angolo misurato in radianti nel seguente campo di valori

$$-\pi/2 \leq \text{valore di uscita} \leq +\pi/2$$

laddove π = 3,1415....

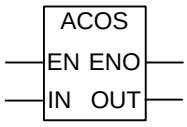
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.9 ACOS Formazione dell'arcocoseno di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: arcocoseno del numero in virgola mobile

Descrizione dell'operazione

ACOS (Formazione dell'arcocoseno di un numero in virgola mobile) calcola l'arcocoseno di un numero in virgola mobile, il cui campo di definizione è -1 <= valore di ingresso <= 1. Il risultato rappresenta un angolo misurato in radianti nel seguente campo di valori

$$0 \leq \text{valore di uscita} \leq +\pi$$

laddove π = 3,1415....

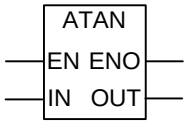
Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

8.4.10 ATAN Formazione dell'arcotangente di un numero in virgola mobile

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	REAL	E, A, M, L, D o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, L, D	Valore di uscita: arcotangente del numero in virgola mobile

Descrizione dell'operazione

ATAN (Formazione dell'arcotangente di un numero in virgola mobile) calcola l'arcotangente di un numero in virgola mobile. Il risultato rappresenta un angolo misurato in radianti nel seguente campo di valori

$$-\pi/2 \leq \text{valore di uscita} \leq +\pi/2$$

laddove π = 3,1415....

Vedere anche Valutazione dei bit nella parola di stato con operazioni in virgola mobile.

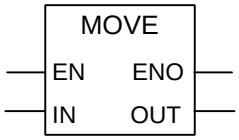
Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive:	x	x	x	x	x	0	x	x	1

9 Operazioni di trasferimento

9.1 MOVE Assegna valore

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	Tutti i tipi di dati semplici con una lunghezza di 8, 16 o 32 bit	E, A, M, L, D oppure costante	Valore sorgente
OUT	Tutti i tipi dati semplici con una lunghezza di 8, 16 o 32 bit	E, A, M, L, D	Operando di destinazione

Descrizione dell'operazione

MOVE (Assegna valore) viene attivato dall'ingresso di abilitazione EN. Il valore indicato dall'ingresso IN viene copiato all'operando indicato dall'uscita OUT. ENO ha lo stesso stato di segnale di EN. L'operazione MOVE può copiare solo oggetti di dati della lunghezza BYTE, WORD o DWORD. I tipi di dati definiti dall'utente come campi o strutture vanno copiati con l'operazione SFC BLKMOVE.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	-	-	-	-	0	1	1	1

Dipendenza da MCR (Relè Master Control)

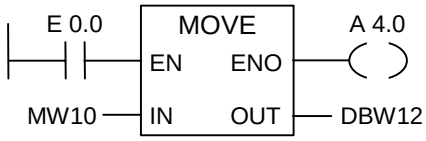
La dipendenza MCR viene attivata solo se il box MOVE viene posizionato all'interno di una zona MCR attiva. All'interno di una zona MCR precedentemente attivata, se MCR è attivato ed è presente un flusso di corrente all'ingresso di abilitazione, i dati indirizzati vengono copiati come descritto sopra. Se MCR è disattivato e viene eseguita un'operazione MOVE, il valore "0" viene scritto all'operando indicato da O indipendentemente dallo stato attuale di IN.

Nota

Con il trasferimento di un valore in un tipo di dati di lunghezza diversa, i byte con valore maggiore vengono tagliati secondo necessità o integrati con zeri. Esempi:

Doppia parola	**1111 1111**	**0000 1111**	**1111 0000**	**0101 0101**
Trasferimento	**Risultato**			
in una doppia parola:	1111 1111	0000 1111	1111 0000	0101 0101
in un byte:				0101 0101
in una parola:			1111 0000	0101 0101
Byte				**1111 0000**
Trasferimento	**Risultato**			
in un byte:				1111 0000
in una parola:			0000 0000	1111 0000
in una doppia parola:	0000 0000	0000 0000	0000 0000	1111 0000

Esempio



L'operazione viene eseguita se E 0.0 = 1; il contenuto di MW10 viene quindi copiato alla parola di dati 12 dell'attuale blocco di dati aperto.

A 4.0 è "1" se l'operazione viene eseguita.

Se i percorsi di corrente dell'esempio si trovano all'interno di una zona MCR attivata, si ha:

Se MCR è attivato, i dati vengono copiati come descritto sopra da MW10 a DBW12.

Se MCR è disattivato, il valore "0" viene scritto in DBW12.

10 Operazioni di controllo del programma

10.1 Sommario delle operazioni di comando del programma

Descrizione

Sono disponibili le seguenti operazioni di comando del programma:

- ---(Call) Richiama FC/SFC dalla bobina (senza parametri)
- CALL_FB Richiama FB dal box
- CALL_FC Richiama FC dal box
- CALL_SFB Richiama SFB dal box
- CALL_SFC Richiama SFC dal box
- Richiamo di multiistanze
- Richiamo di blocchi da una biblioteca

- Avvertenze importanti sulle funzionalità MCR
- ---(MCR<) Attiva zona relè master control
- ---(MCR>) Disattiva zona relè master control
- ---(MCRA) Inizio zona relè master control
- ---(MCRD) Fine zona relè master control

- RET Salta indietro

10.2 ---(Call) Richiama FC/SFC dalla bobina (senza parametri)

Simbolo

<FC/SFC-Nr.>

---(CALL)

Parametro	Tipo di dati	Area di memoria	Descrizione
<FC/SFC-Nr.>	BLOCK_FC	-	Numero di FC/SFC, il campo dipende dalla CPU

Descrizione dell'operazione

---(Call) (Richiama FC/SFC dalla bobina (senza parametri)) richiama una funzione (FC) o una funzione di sistema (SFC) senza parametri. Un richiamo viene eseguito solo se RLC alla bobina CALL è "1". Se l'operazione ---(Call) viene eseguita, essa realizza le seguenti funzioni

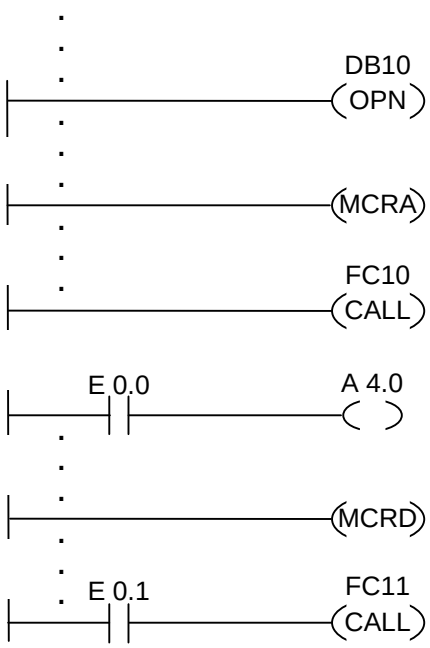
- salva l'indirizzo di cui c'è bisogno per ritornare al blocco richiamante,
- sostituisce l'attuale campo di dati locali con il campo di dati locali precedente,
- colloca il bit MA (bit attivo MCR) nello stack di blocco e
- crea un nuovo campo di dati locali per la funzione richiamata.

Al termine, l'elaborazione del programma continua nella funzione o funzione di sistema richiamata.

Parola di stato

		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Richiamo condizionato:	Scrive:	-	-	-	-	0	0	1	1	0
Richiamo assoluto:	Scrive:	-	-	-	-	0	0	1	-	0

Esempio



L'esempio sopraindicato dei percorsi di corrente dello schema a contatti illustra le sezioni di programma di un blocco funzionale definito dall'utente. In questo blocco, viene aperto il DB10 ed attivata la funzione MCR. Se il richiamo assoluto di FC10 viene eseguito, ha luogo ciò che segue:

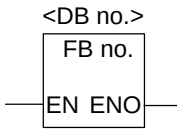
vengono memorizzati l'indirizzo di ritorno del blocco funzionale richiamante e i selettori di DB10 e del blocco dati di istanza del blocco funzionale richiamante . Il bit MA, impostato ad "1" dall'operazione MCRA, viene collocato nello stack di blocco, e quindi impostato a "0" per il blocco richiamato (FC10). L'elaborazione del programma continua in FC10. Se FC10 richiede la funzione MCR, il MCR FC10 deve essere riattivato. Se l'elaborazione di FC10 è terminata, l' elaborazione del programma ritorna al FB richiamante. Il bit MA viene ripristinato. DB10 ed il blocco dati di istanza del blocco funzionale definito dall'utente ritornano ad essere DB attuale. Il programma continua con il percorso di corrente seguente, in questo caso assegnando all'uscita A 4.0 lo stato di segnale di E 0.0. Il richiamo di FC11 è un richiamo condizionato. Questo richiamo viene eseguito solo se E 0.1 = 1. Se il richiamo viene eseguito, il controllo del programma viene trasmesso a FC11 come descritto per FC10, e ritorna indietro dopo l'elaborazione di FC11.

Nota

Dopo il ritorno nel blocco richiamante non è sempre sicuro che il DB aperto in precedenza sia nuovamente aperto. Osservare la nota nel file Leggimi.

10.3 CALL_FB Richiama FB dal box

Simbolo



Il simbolo dipende dal blocco funzionale (a seconda se vi sono parametri, e quanti ve ne sono). EN, ENO e il nome, ovvero il numero del FB devono essere presenti.

Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
FB no. DB no.	BLOCK_FB BLOCK_DB	- -	Numero del FB/DB; il campo dipende dalla CPU

Descrizione dell'operazione

CALL_FB (Richiama FB dal box) viene eseguita se EN = 1. Se l'operazione CALL_FB viene eseguita, essa realizza le seguenti funzioni:

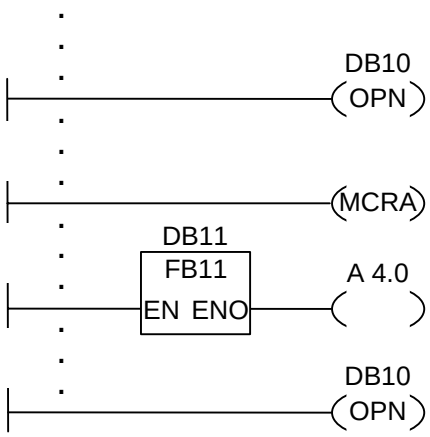
- salva l'indirizzo di cui c'è bisogno per ritornare al blocco richiamante
- salva i selettori di entrambi gli attuali blocchi di dati (DB e DB di istanza)
- sostituisce l'attuale campo di dati locali con il campo di dati locali precedente
- colloca il bit MA (bit attivo MCR) nello stack di blocco
- crea un nuovo campo di dati locali per la funzione richiamata.

Al termine, l'elaborazione del programma continua nel blocco funzionale richiamato. Il bit BIE viene interrogato per verificare lo stato di ENO. A quest'ultimo, l'utente deve assegnare lo stato desiderato (valutazione errore) nel blocco richiamato con ---(SAVE).

Parola di stato

		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Richiamo condizionato:	Scrive:	x	-	-	-	0	0	x	x	x
Richiamo assoluto:	Scrive:	-	-	-	-	0	0	x	x	x

Esempio



L'esempio sopraindicato dei percorsi di corrente dello schema a contatti illustra le sezioni di programma di un blocco funzionale definito dall'utente. In questo blocco funzionale, viene aperto il DB10 ed attivata la funzione MCR. Se il richiamo assoluto di FB11 viene eseguito, ha luogo ciò che segue:

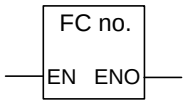
L'indirizzo di ritorno del blocco funzionale richiamante e i selettori di DB10 e del blocco dati di istanza del blocco funzionale richiamante vengono memorizzati. Il bit MA, impostato ad "1" dall'operazione MCRA, viene collocato nello stack di blocco e quindi impostato a "0" per il blocco funzionale richiamato FB11. L'elaborazione del programma continua in FB11. Se FB11 richiede la funzione MCR, il MCR deve essere riattivato nel blocco funzionale. Lo stato del RLC deve essere memorizzato nel bit BIE con l'operazione ---(SAVE) per poter eseguire la valutazione dell'errore nel FB richiamante. Se l'elaborazione di FB11 è terminata, l'elaborazione del programma ritorna al blocco funzionale richiamante. Il bit MA viene ripristinato. Il blocco dati di istanza del blocco funzionale definito dall'utente ritorna ad essere il DB attuale. Se FB11 viene eseguito correttamente, ENO = 1 e quindi A 4.0 = 1.

Nota

Nei richiami FB/SFB, il numero del blocco dati aperto precedentemente non viene salvato. Il DB richiesto deve essere aperto di nuovo.

10.4 CALL_FC Richiama FC dal box

Simbolo



Il simbolo dipende dalla funzione (a seconda se vi sono parametri, e quanti ve ne sono). EN, ENO e il nome, ovvero il numero della FC devono essere presenti.

Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
FC no.	BLOCK_FC	-	Numero della FC; il campo dipende dalla CPU

Descrizione dell'operazione

CALL_FC (Richiama FC dal box) richiama una funzione (FC). Il richiamo viene eseguito solo se EN =1. Se l'operazione CALL_FC viene eseguita, essa realizza le seguenti funzioni

- salva l'indirizzo di cui c'è bisogno per ritornare al blocco richiamante
- sostituisce l'attuale campo di dati locali con il campo di dati locali precedente
- colloca il bit MA (bit attivo MCR) nello stack di blocco
- crea un nuovo campo di dati locali per la funzione richiamata.

Al termine, l'elaborazione del programma continua nella funzione richiamata.

Il bit BIE viene interrogato per verificare lo stato di ENO. A quest'ultimo, l'utente deve assegnare con ---(SAVE) lo stato desiderato (valutazione errore) nel blocco richiamato.

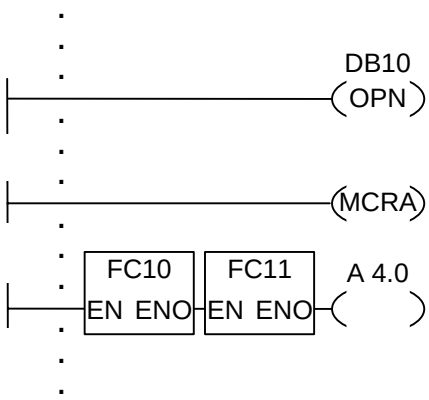
Se si richiama un'FC e la tabella di dichiarazione delle variabili del blocco richiamato dispone di dichiarazioni del tipo IN, OUT e IN_OUT, le variabili verranno visualizzate nel programma del blocco richiamante come lista di parametri formali.

Con il richiamo di una FC **è obbligatorio** assegnare parametri attuali ai parametri formali nella posizione di richiamo. Eventuali valori iniziali nella dichiarazione FC sono irrilevanti.

Parola di stato

		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Richiamo condizionato:	Scrive:	-	-	-	-	0	0	x	x	x
Richiamo assoluto:	Scrive:	x	-	-	-	0	0	x	x	x

Esempio



L'esempio sopraindicato dei percorsi di corrente dello schema a contatti illustra le sezioni di programma di un blocco funzionale definito dall'utente. In questo blocco funzionale, viene aperto il DB10 ed attivata la funzione MCR. Se il richiamo assoluto di FC10 viene eseguito, ha luogo ciò che segue:

vengono memorizzati l'indirizzo di ritorno del blocco funzionale richiamante e i selettori di DB10 e del blocco dati di istanza del blocco funzionale richiamante . Il bit MA, impostato ad "1" dall'operazione MCRA, viene collocato nello stack di blocco e quindi impostato a "0" per il blocco richiamato (FC10). L'elaborazione del programma continua in FC10. Se FC10 richiede la funzione MCR, il MCR di FC10 deve essere riattivato. Lo stato di RLC deve essere memorizzato nel bit BIE con l'operazione ---(SAVE) per poter eseguire la valutazione dell'errore nel FB richiamante. Se l'elaborazione di FC10 è terminata, l'elaborazione del programma ritorna al FB richiamante. Il bit MA viene ripristinato. Dopo l'elaborazione della FC10, a seconda dello stato di ENO, il programma continua nel FB richiamante:

ENO = 1 FC11 viene elaborata

ENO = 0 L'elaborazione continua nel segmento successivo

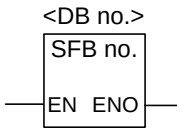
Se viene eseguito correttamente anche FC11, ENO = 1 e quindi A **4.0 = 1**.

Nota

Dopo il ritorno nel blocco richiamante non è sempre sicuro che il DB aperto in precedenza sia nuovamente aperto. Osservare la nota nel file Leggimi.

10.5 CALL_SFB Richiama SFB dal box

Simbolo



Il simbolo dipende dal blocco funzionale di sistema (a seconda se vi sono parametri, e quanti ve ne sono). EN, ENO e il nome ovvero il numero del SFB devono essere presenti.

Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
SFB no. DB no.	BLOCK_SFB BLOCK_DB	- -	Numero del SFB/DB; il campo dipende dalla CPU

Descrizione dell'operazione

CALL_SFB (Richiama SFB dal box) viene eseguita se EN = 1. Se l'operazione CALL_SFB viene eseguita, essa realizza le seguenti funzioni:

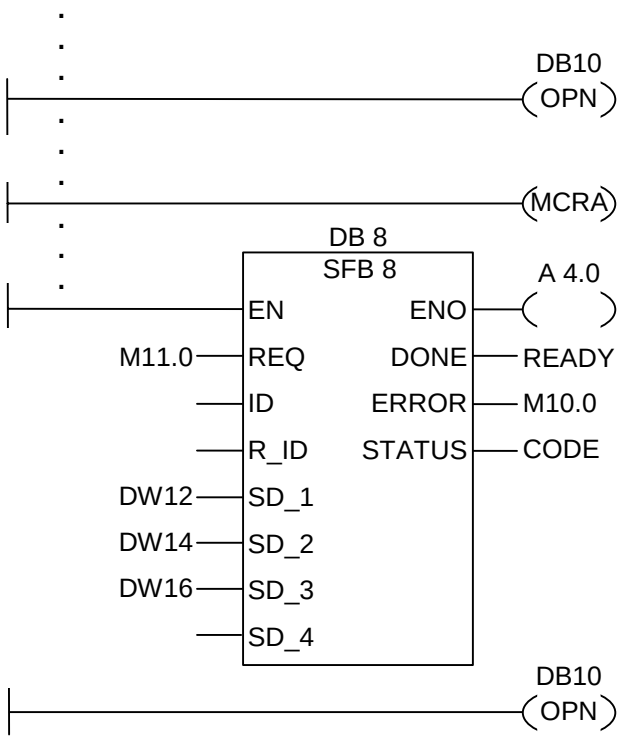
- salva l'indirizzo di cui c'è bisogno per ritornare al blocco richiamante
- salva i selettori di entrambi gli attuali blocchi di dati (DB e DB di istanza)
- sostituisce l'attuale campo di dati locali con il campo di dati locali precedente
- colloca il bit MA (bit attivo MCR) nello stack di blocco
- crea un nuovo campo di dati locali per la funzione richiamata.

Al termine, l'elaborazione del programma continua nel blocco funzionale di sistema richiamato. ENO è "1" se il blocco funzionale di sistema è stato richiamato (EN = 1) e non si sono verificati errori.

Parola di stato

		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Richiamo condizionato:	Scrive:	x	-	-	-	0	0	x	x	x
Richiamo assoluto:	Scrive:	-	-	-	-	0	0	x	x	x

Esempio



Nei percorsi di corrente sopra riportati, relativi ad uno schema a contatti, si tratta di sezioni di programma di un blocco funzionale scritto dall'utente. DB10 viene aperto in questo blocco, e MCR viene attivato. Se viene eseguito il richiamo assoluto di SFB8, si ha quanto segue.

L'indirizzo di ritorno del blocco funzionale richiamante e i selettori di DB10 e del blocco dati di istanza del blocco funzionale richiamante vengono memorizzati. Il bit MA, impostato ad "1" dall'operazione MCRA, viene collocato nello stack di blocco, e quindi impostato a "0" per il blocco funzionale di sistema richiamato (SFB8). L'elaborazione del programma continua in SFB8. Se l'elaborazione di SFB8 è terminata, l'elaborazione del programma ritorna al blocco funzionale richiamante. Il bit MA viene ripristinato. Il blocco dati di istanza del blocco funzionale definito dall'utente ritorna ad essere il DB attuale. Se SFB8 viene eseguito correttamerte, ENO = 1 e quindi A4.0 = 1.

Nota

Nei richiami FB/SFB, il numero del blocco dati aperto precedentemente non viene salvato. Il DB richiesto deve essere nuovamente aperto.

10.6 CALL_SFC Richiama SFC dal box

Simbolo

Il simbolo dipende dalla funzione di sistema (a seconda se vi sono parametri, e quanti ve ne sono). EN, ENO e il nome, ovvero il numero della SFC devono essere presenti.

Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
SFC no.	BLOCK_SFC	-	Numero della SFC; il campo dipende dalla CPU

Descrizione dell'operazione

CALL_SFC (Richiama SFC dal box) richiama una funzione di sistema (SFC). Il richiamo viene eseguito se EN = 1 Se l'operazione CALL_SFC viene eseguita, essa realizza le seguenti funzioni

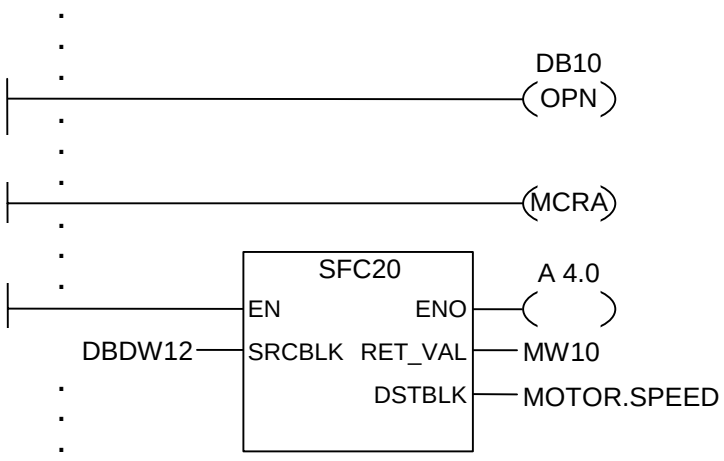
- salva l'indirizzo di cui c'è bisogno per ritornare al blocco richiamante,
- sostituisce l'attuale campo di dati locali con il campo di dati locali precedente,
- colloca il bit MA (bit attivo MCR) nello stack di blocco e
- crea un nuovo campo di dati locali per la funzione richiamata.

Al termine, l'elaborazione del programma continua nella funzione di sistema richiamata. ENO è "1" se la funzione è stata richiamata (EN = 1), e non si è verificato alcun errore.

Parola di stato

		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Richiamo condizionato:	Scrive:	-	-	-	-	0	0	x	x	x
Richiamo assoluto:	Scrive:	x	-	-	-	0	0	x	x	x

Esempio



L'esempio sopraindicato dei percorsi di corrente dello schema a contatti illustra le sezioni di programma di un blocco funzionale definito dall'utente. In questo blocco funzionale, viene aperto il DB10 ed attivata la funzione MCR. Se il richiamo assoluto di SFC20 viene eseguito, ha luogo ciò che segue:

vengono memorizzati l'indirizzo di ritorno del blocco funzionale richiamante e i selettori di DB10 e del blocco dati di istanza del blocco funzionale richiamante . Il bit MA, impostato ad "1" dall'operazione MCRA , viene collocato nello stack di blocco, e quindi impostato a "0" per il blocco richiamato SFC20. L'elaborazione del programma continua in SFC20. Se l'elaborazione di SFC20 è terminata, l'elaborazione del programma ritorna al FB richiamante. Il bit MA viene ripristinato.

Dopo l'esecuzione di SFC20, ed in dipendenza da ENO, il programma continua nel FB richiamante.

ENO = 1 A 4.0 = 1

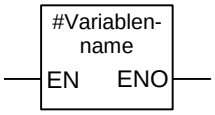
ENO = 0 A 4.0 = 0

Nota

Dopo il ritorno nel blocco richiamante non è sempre sicuro che il DB aperto in precedenza sia nuovamente aperto. Osservare la nota nel file Leggimi.

10.7 Richiamo di una multi-istanza

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
# Nome della variabile	FB, SFB	-	Nome della multi-istanza

Descrizione

Una multi-istanza viene creata con la dichiarazione di una variabile statica del tipo di dati di un blocco funzionale. Nel catalogo degli elementi del programma vengono elencate solo le multiistanze già dichiarate.
Il simbolo di una multi-istanza cambia a seconda della presenza e del numero di parametri presenti. EN, EN0 e il nome della variabile sono sempre presenti.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	0	0	x	x	x

10.8 Richiamo di blocchi da una biblioteca

Le biblioteche usate nel SIMATIC Manager vengono offerte all'utente per la selezione.

Da queste biblioteche è possibile selezionare blocchi

- integrati nel sistema operativo della CPU dell'utente (Biblioteca "Standard library" per i progetti STEP 7 appartenenti alla versione 3 e "stdlibs (V2)" per progetti STEP 7 della versione 2)
- depositati dall'utente stesso in biblioteche in quanto destinati ad un uso multiplo.

10.9 Avvertenze importanti sulle funzionalità MCR

Attenzione ai blocchi nei quali il relè master control è stato attivato con MCRA

- Se il relè master control è disattivato, nelle sezioni di programma tra ---(MCR<) e ---(MCR>), attraverso tutte le assegnazioni viene scritto il valore 0! Ciò riguarda naturalmente **tutti** i box che contengono un'assegnazione, compresa l'assegnazioni di parametri ai blocchi!
- Il relè master control è disattivato esattamente quando davanti a un comando **MCR<** l'RLC era = 0.

Pericolo STOP del sistema di automazione o comportamento di esecuzione indefinito!

Per il calcolo degli indirizzi il compilatore accede anche in scrittura ai dati locali dietro le variabili temporanee definite in VAR_TEMP. Per questo motivo le sequenze di comandi seguenti portano il PLC su STOP o provocano un comportamento di esecuzione indefinito.

Accessi a parametri formali

- Accessi a componenti di parametri FC complessi del tipo STRUCT, UDT, ARRAY, STRING
- Accessi a componenti di parametri FB complessi del tipo STRUCT, UDT, ARRAY, STRING dell'area IN_OUT in un blocco con multiistanza (versione blocchi 2)
- Accessi a parametri di un FB con multiistanza (versione blocchi 2) quando il loro indirizzo è maggiore di 8180.0
- L'accesso nell'FB con multiistanza (versione blocchi 2) a un parametro del tipo BLOCK_DB apre il DB 0. I successivi accessi ai dati portano la CPU su STOP. Con TIMER, COUNTER, BLOCK_FC, BLOCK_FB si utilizzano sempre anche T 0, Z 0, FC 0 e FB 0.

Assegnazione di parametri

- Richiami con i quali vengono trasmessi parametri

KOP/FUP

- Diramazioni a T e connettori in KOP o FUP iniziano con RLC = 0.

Rimedio

Annullare la dipendenza MCR dei comandi indicati:

1. disattivare il relè master control con fine zona relè master control (MCRA) prima dell'istruzione o del segmento in questione
2. attivare il relè master control con inizio zona relè master control (MCRD) dopo l'istruzione o il segmento in questione.

10.10 ---(MCR<) Attiva zona relè master control

Simbolo

---(MCR<)

Descrizione dell'operazione

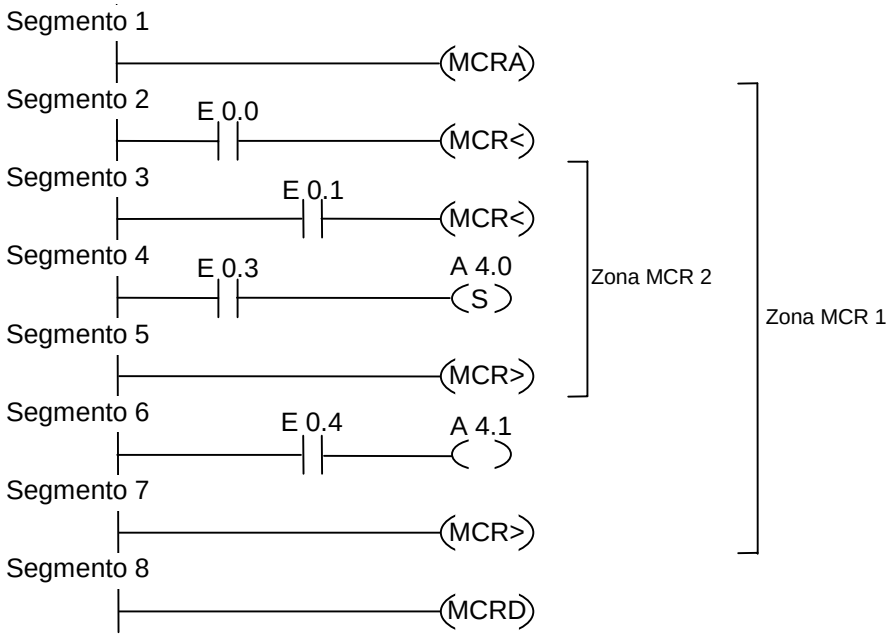
---(MCR<) (Attiva zona relè master control) memorizza il RLC nello stack di MCR. Lo stack di parentesi di MCR lavora come un buffer LIFO (ovvero last in, first out), il quale può registrare al massimo 8 registrazioni (8 livelli). Se lo stack è già pieno, l'operazione ---(MCR<) provoca una anomalia dello stack di MCR (MCRF). Gli elementi seguenti dipendono da MCR, e sono influenzati dallo stato di segnale di RLC memorizzato nello stack di MCR finché rimane aperta una zona MCR.

- --(#) Connettore
- --( ) Bobina di uscita
- --(S) Imposta uscita
- --(R) Resetta uscita
- RS Imposta resetta flip flop
- SR Resetta imposta flip flop
- MOVE Assegna valore

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	1	-	0

Esempio



Il percorso di corrente MCRA attiva MCR. Si possono creare fino a otto zone MCR. Nell'esempio riportato vi sono due zone MCR. Le operazioni vengono eseguite nel modo seguente:

E 0.0 = 1 (MCR è ON nella zona 1): lo stato di segnale di E 0.4 viene assegnato a A 4.1

E 0.0 = 0 (MCR è OFF nella zona 1): A 4.1 è "0", a prescindere dallo stato di segnale di E 0.4

E 0.0 *e* E 0.1 = 1 (MCR è ON nella zona 2): A 4.0 viene impostata a "1" se E 0.3 = 1 (e = operazione AND)

E 0.0 *e* E 0.1 = 0 (MCR è OFF nella zona 2): A 4.0 non viene modificata, indipendentemente dallo stato di E 0.3 (e = operazione AND)

10.11 ---(MCR>) Disattiva zona relè master control

Simbolo

---(MCR>)

Descrizione dell'operazione

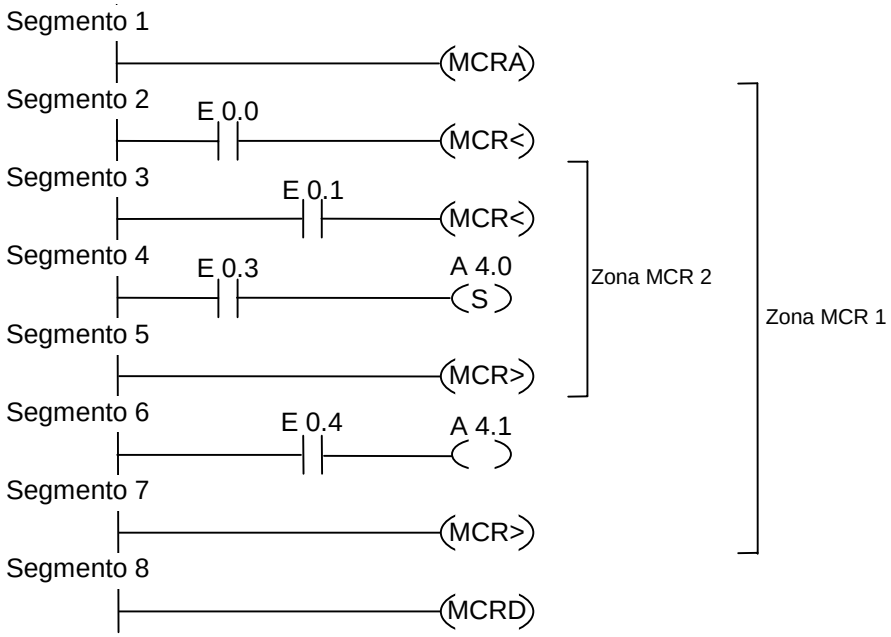
---(MCR>) (Disattiva zona relè master control) cancella una registrazione RLC nello stack di MCR. Lo stack di parentesi di MCR lavora come un buffer LIFO (ovvero last in, first out), il quale può registrare al massimo 8 registrazioni (8 livelli). Se lo stack è già vuoto, l'operazione ---(MCR<) provoca una anomalia dello stack di MCR (MCRF). Gli elementi seguenti dipendono da MCR, e sono influenzati dallo stato di segnale di RLC memorizzato nello stack di MCR finché rimane aperta una zona MCR:

- --(#) Connettore
- --( ) Bobina di uscita
- --(S) Imposta uscita
- --(R) Resetta uscita
- RS Imposta resetta flip flop
- SR Resetta imposta flip flop
- MOVE Assegna valore

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	1	-	0

Esempio



MCR viene attivato dall'operazione ---(MCRA) . Si possono creare fino a otto zone MCR. Nell'esempio riportato, ci sono due zone MCR. Il primo percorso di corrente ---(MCR>) (MCR OFF) appartiene al secondo percorso di corrente ---(MCR<) (MCR ON). Tutti i percorsi di corrente tra questi due appartengono alla seconda zona MCR. Le operazioni vengono eseguite nel modo seguente:

E 0.0 = 1: lo stato di segnale di E 0.4 viene assegnato a A 4.1

E 0.0 = 0: A 4.1 è "0", a prescindere dallo stato di segnale di E 0.4

E 0.0 *e* E 0.1 = 1: A 4.0 viene impostata a "1" se E 0.3 = 1 (*e* = operazione AND)

E 0.0 *e* E 0.1 = 0: A 4.0 non viene modificata, indipendentemente dallo stato di E 0.3 (*e* = operazione AND)

10.12 ---(MCRA) Inizio zona relè master control

Simbolo

---(MCRA)

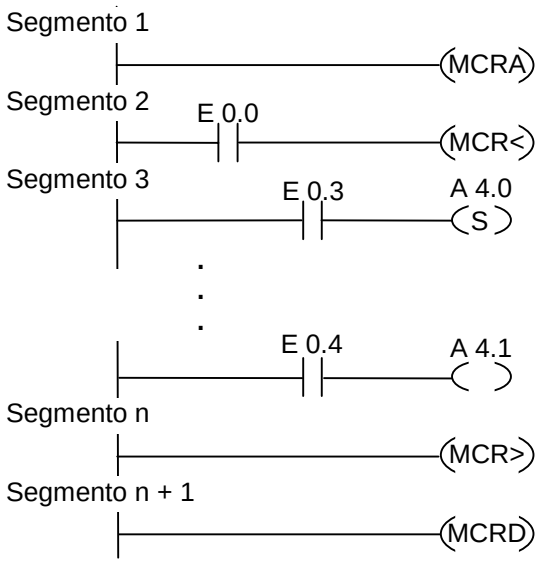
Descrizione dell'operazione

---(MCRA) (Inizio zona relè master control) attiva il relè master control. Dopo questa operazione, è possibile programmare operazioni con le seguenti zone MCR: ---(MCR<) e ---(MCR>).

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	-	-	-	-

Esempio



Il percorso di corrente MCRA attiva MCR. I percorsi di corrente tra le operazioni MCR< e MCR> (uscite A 4.0, A 4.1) vengono eseguite nel modo seguente:

E 0.0 = 1 (MCR è ON): A 0.4 viene impostata a "1" se lo stato di segnale di E 0.3 è "1", oppure non viene modificata, se lo stato di segnale di E 0.3 è "0". Lo stato di segnale di E 0.4 viene assegnato all'uscita A 4.1.

E 0.0 = 0 (MCR è OFF): A 4.0 non viene modificata, a prescindere dallo stato di segnale di E 0.3. A 4.1 è "0", indipendentemente dallo stato di segnale di E 0.4.

Nel prossimo percorso di corrente, l'operazione ---(MCRD) disattiva MCR. Ciò significa che con le due operazioni ---(MCR<) e ---(MCR>) non è più possibile programmare zone MCR.

10.13 ---(MCRD) Fine zona relè master control

Simbolo

---(MCRD)

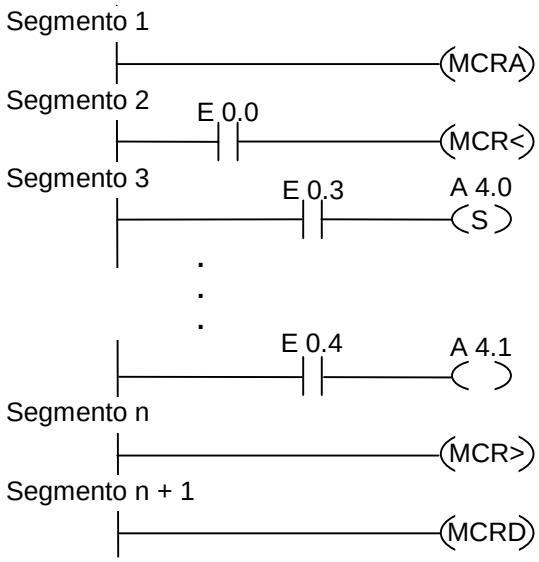
Descrizione dell'operazione

---(MCRD) (Fine zona relè master control) disattiva MCR. Dopo questa operazione, l'utente non può più programmare le zone MCR.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	-	-	-	-

Esempio



Il percorso di corrente ---(MCRA) attiva MCR. I percorsi di corrente tra le operazioni MCR< e MCR> (uscite A 4.0, A 4.1) vengono eseguite nel modo seguente:

E 0.0 = 1 (MCR è ON): A 0.4 viene impostata a "1" se lo stato di segnale di E 0.3 è "1", oppure non viene modificata, se lo stato di segnale di E 0.3 è "0". Lo stato di segnale di E 0.4 viene assegnato all'uscita A 4.1.

E 0.0 = 0 (MCR è OFF): A 4.0 non viene modificata, a prescindere dallo stato di segnale di E 0.3. A 4.1 è "0", indipendentemente dallo stato di segnale di E 0.4.

Nel prossimo percorso di corrente, l'operazione ---(MCRD) disattiva MCR. Ciò significa che con le due operazioni ---(MCR<) e ---(MCR>) non è più possibile programmare zone MCR.

10.14 ---(RET) Salta indietro

Simbolo

---(RET)

Descrizione dell'operazione

RET (Salta indietro) viene adoperata per terminare i blocchi in modo condizionato. Per questa operazione, è necessaria una combinazione logica precedente.

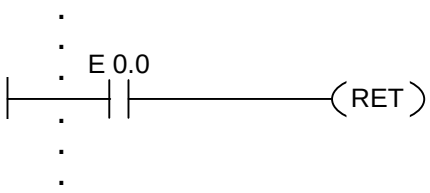
Parola di stato

Salto condizionato (Salto indietro se RLC = 1):

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	*	-	-	-	0	0	1	1	0

* L'operazione **RET** si forma internamente sulla sequenza "SAVE; BEB;". Ciò significa che anche il bit BIE viene influenzato.

Esempio



Il blocco viene terminato se E 0.0 = 1.

11 Operazioni di scorrimento e rotazione

11.1 Operazioni di scorrimento

11.1.1 Sommario delle operazioni di scorrimento

Descrizione

L’utente ha la possibilità di utilizzare le operazioni di scorrimento per trasferire il contenuto dell’ingresso IN bit per bit, a sinistra o a destra (vedere Registri CPU). Uno scorrimento a sinistra per un numero n di bit moltiplica il contenuto dell’ingresso IN per 2^n; facendo invece scorrere a destra sempre per un numero n di bit, si divide il contenuto dell’ingresso IN per 2^n. Per esempio, se l’utente fa scorrere a sinistra l’equivalente binario del valore decimale 3 nella misura di tre bit, ottiene alla fine l’equivalente binario del valore decimale 24. Se fa scorrere, invece, a destra l’equivalente binario del valore decimale 16 nella misura di 2 bit, otterrà l’equivalente binario del valore decimale 4.

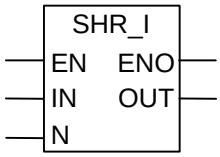
Il numero che l’utente fornisce per il parametro di ingresso N sta ad indicare il numero di bit nella misura in cui si deve effettuare lo scorrimento. I posti bit che vengono lasciati vuoti dall’operazione di scorrimento vengono colmati con degli zero, oppure con lo stato di segnale del bit del segno (0 sta per positivo, 1 sta per negativo). Lo stato del segnale del bit che è stato fatto scorrere per ultimo viene caricato nel bit A1 della parola di stato. I bit A0 e OV della parola di stato vengono resettati a 0. L’utente può usare le operazioni di salto per valutare il bit A1.

Sono disponibili le seguenti operazioni di scorrimento:

- SHR_I Fai scorrere numero intero a 16 bit verso destra
- SHR_DI Fai scorrere numero intero a 32 bit verso destra
- SHL_W Fai scorrere parola verso sinistra (a 16 bit)
- SHR_W Fai scorrere parola verso destra (a 16 bit)
- SHL_DW Fai scorrere doppia parola verso sinistra (a 32 bit)
- SHR_DW Fai scorrere doppia parola verso destra (a 32 bit)

11.1.2 SHR_I Fai scorrere numero intero a 16 bit verso destra

Simbolo

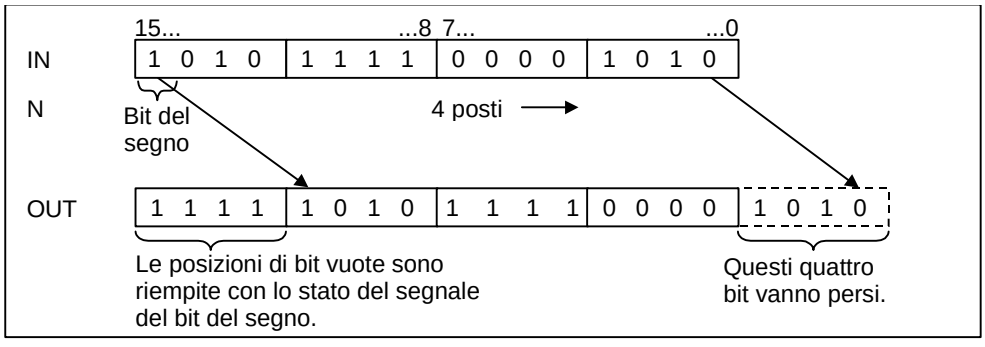


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	INT	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire lo scorrimento
OUT	INT	E, A, M, L, D	Risultato dell'operazione di scorrimento

Descrizione dell'operazione

SHR_I (Fai scorrere numero intero a 16 bit verso destra) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione SHR_I fa scorrere a destra, bit per bit, i bit da 0 a 15 dell'ingresso IN. Non sono interessati i bit da 16 a 31. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. Se N è maggiore di 16, il comando opera come se N = 16. Le posizioni di bit che vengono fatte scorrere da sinistra per occupare i posti liberi ricevono lo stato di segnale del bit 15 (bit di segno del numero intero). Ciò significa che le posizioni di bit vengono occupate con il valore "0" se si tratta di un numero intero positivo, e con il valore "1" se si tratta di un numero intero negativo. Il risultato dell'operazione di trasferimento può essere rilevato all'uscita OUT. L'operazione SHR_I imposta a "0" i bit A0 e OV se N è diverso da 0.

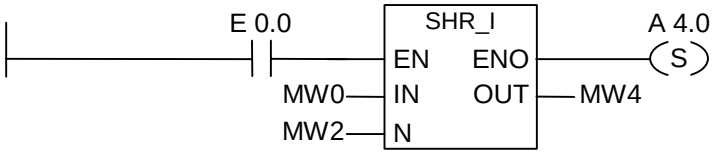
ENO ha lo stesso stato di segnale di EN.



Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

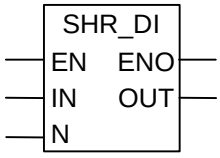
Esempio



Il box SHR_I viene attivato se E 0.0 = 1. MW0 viene caricato e fatto scorrere a destra nella misura del numero di bit specificato in MW2. Il risultato viene scritto in MW4. A 4.0 viene impostata.

11.1.3 SHR_DI Fai scorrere numero intero a 32 bit verso destra

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DINT	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire lo scorrimento
OUT	DINT	E, A, M, L, D	Risultato dell'operazione di scorrimento

Descrizione dell'operazione

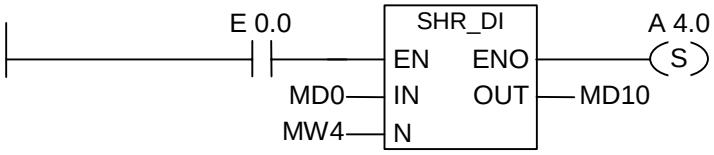
SHR_DI (Fai scorrere numero intero a 32 bit verso destra) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione SHR_DI fa scorrere a destra, bit per bit, i bit da 0 a 31 dall'ingresso IN. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. Se N è maggiore di 32, il comando opera come se N = 32. Le posizioni di bit che vengono fatte scorrere da sinistra per occupare i posti liberi ricevono lo stato di segnale del bit 31 (bit di segno del numero intero). Ciò significa che queste posizioni di bit vengono occupate con il valore "0" se si tratta di un numero intero positivo, e con il valore "1" se si tratta di un numero intero negativo. Il risultato dell'operazione di trasferimento può essere rilevato all'uscita OUT. L'operazione SHR_DI imposta a "0" i bit A0 e OV se N è diverso da 0.

ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

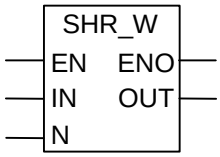
Esempio



Il box SHR_DI viene attivato se E 0.0 = 1. MD0 viene caricato e fatto scorrere a destra nella misura del numero di bit specificato in MW4. La doppia parola di risultato viene scritta a MD10. A 4.0 viene impostata.

11.1.4 SHR_W Fai scorrere parola verso destra (a 16 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	WORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire lo scorrimento
OUT	WORD	E, A, M, L, D	Parola di risultato dell'operazione di scorrimento

Descrizione dell'operazione

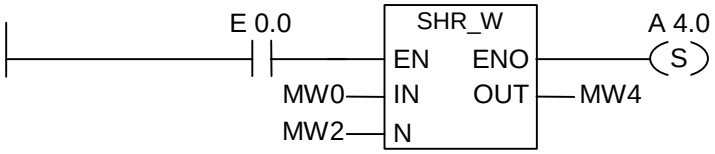
SHR_W (Fai scorrere parola verso destra (a 16 bit)) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione SHR_W fa scorrere a destra, bit per bit, i bit da 0 a 15 dell'ingresso IN. Non sono interessati i bit da 16 a 31. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. Se N è maggiore di 16, il comando all'uscita OUT scrive uno "0", ed imposta a "0" i bit A0 e OV della parola di stato. Da sinistra viene fatto scorrere lo stesso numero (N) di zeri per occupare le posizioni libere. Il risultato dell'operazione di scorrimento può essere rilevato all'uscita OUT. L'operazione SHR_W imposta a "0" i bit A0 e OV se N è diverso da 0.

ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

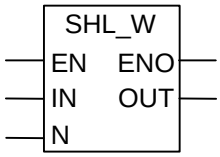
Esempio



Il box SHR_W viene attivato se E 0.0 = 1. MW0 viene caricato e fatto scorrere a destra nella misura del numero di bit specificato in MW2. La parola di risultato viene scritta in MW4. A 4.0 viene impostata.

11.1.5 SHL_W Fai scorrere parola verso sinistra (a 16 bit)

Simbolo

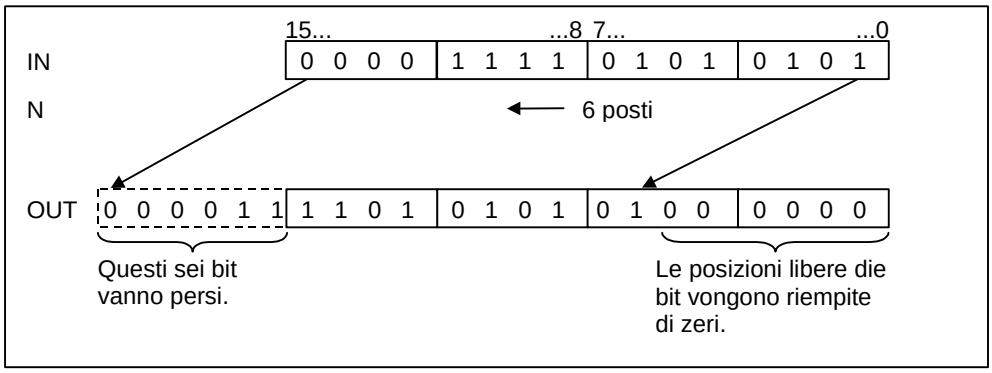


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	WORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire lo scorrimento
OUT	WORD	E, A, M, L, D	Risultato dell'operazione di scorrimento

Descrizione dell'operazione

SHL_W (Fai scorrere parola verso sinistra (a 16 bit)) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione SHL_W fa scorrere a sinistra, bit per bit, i bit da 0 a 15 dall'ingresso IN. Non sono interessati i bit da 16 a 31. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. Se N è maggiore di 16, il comando all'uscita OUT scrive uno "0", ed imposta a "0" i bit A0 e OV della parola di stato. Da destra viene fatto scorrere lo stesso numero (N) di zeri per occupare le posizioni libere. Il risultato dell'operazione di scorrimento può essere rilevato all'uscita O. L'operazione SHL_W imposta a "0" i bit A0 e OV se N è diverso da 0.

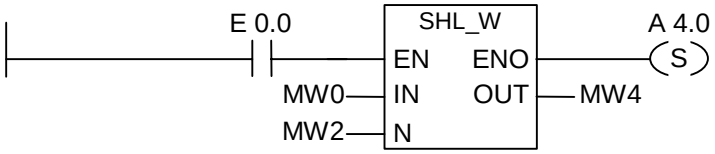
ENO ha lo stesso stato di segnale di EN.



Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

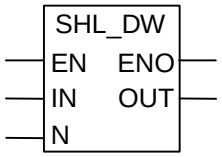
Esempio



Il box SHL_W viene attivato se E 0.0 = 1. MW0 viene caricato, e fatto scorrere a sinistra nella misura del numero di bit specificato in MW2. La parola di risultato viene scritta in MW4. A 4.0 viene impostata.

11.1.6 SHL_DW Fai scorrere doppia parola verso sinistra (a 32 bit)

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DWORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire lo scorrimento
OUT	DWORD	E, A, M, L, D	Doppia parola di risultato dell'operazione di scorrimento

Descrizione dell'operazione

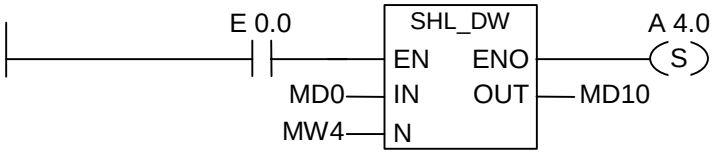
SHL_DW (Fai scorrere doppia parola verso sinistra (a 32 bit)) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione SHL_DW fa scorrere a sinistra, bit per bit, i bit da 0 a 31 dall'ingresso IN. L'ingresso N sta a specificare il valore dell'operazione di scorrimento che deve essere caricato nell'ACCU 1. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. N può variare tra 0 e 31. Sulla destra viene fatto scorrere nell'accumulatore lo stesso numero di zeri (N) per occupare le posizioni libere. La doppia parola di risultato dell'operazione di scorrimento può essere rilevata all'uscita OUT. L'operazione imposta sempre a "0" il bit OV.

ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

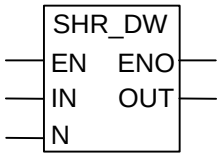
Esempio



Il box SHL_DW viene attivato se E 0.0 = 1. MD0 viene caricato e fatto scorrere a sinistra nella misura del numero di bit specificato in MW4. La doppia parola di risultato viene scritta in MD10. A 4.0 viene impostata.

11.1.7 SHR_DW Fai scorrere doppia parola verso destra (a 32 bit)

Simbolo

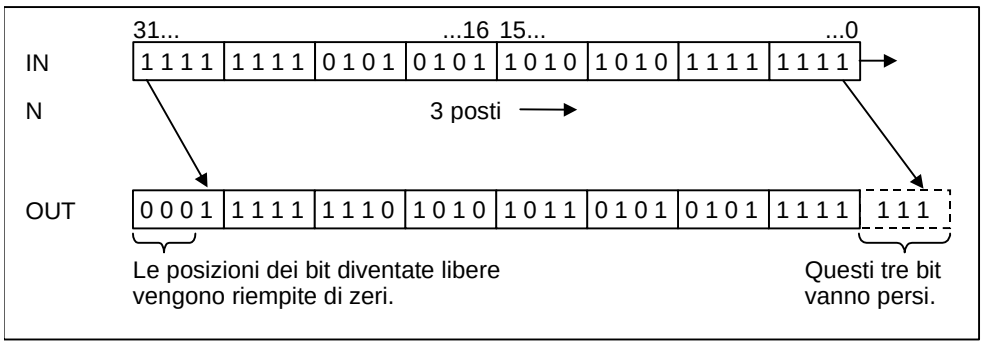


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DWORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire lo scorrimento
OUT	DWORD	E, A, M, L, D	Doppia parola di risultato dell'operazione di scorrimento

Descrizione dell'operazione

SHR_DW (Fai scorrere doppia parola verso destra (a 32 bit)) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione SHR_DW fa scorrere a destra, bit per bit, i bit da 0 a 31 dall'ingresso IN. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. Se N è maggiore di 32, il comando all'uscita OUT scrive uno "0", ed imposta a "0" i bit A0 e OV della parola di stato. Da sinistra viene fatto scorrere lo stesso numero (N) di zeri per occupare le posizioni libere. La doppia parola di risultato dell'operazione di scorrimento può essere rilevato all'uscita OUT. L'operazione SHR_DW imposta a "0" i bit A0 e OV se N è diverso da 0.

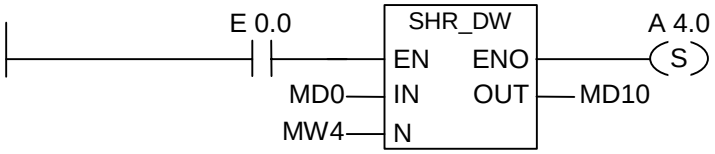
ENO ha lo stesso stato di segnale di EN.



Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

Esempio



Il box SHR_DW viene attivato se E 0.0 = 1. MD0 viene caricato e fatto scorrere a destra nella misura del numero di bit specificato in MW4. La doppia parola di risultato viene scritta in MD10. A 4.0 viene impostata.

11.2 Operazioni di rotazione

11.2.1 Sommario delle operazioni di rotazione

Descrizione

L’utente può usare le operazioni di rotazione per far ruotare l’intero contenuto dell’ingresso IN a sinistra o a destra, bit per bit (vedere Registri CPU). Tuttavia, i posti bit lasciati liberi vengono colmati con gli stati di segnale dei bit che vengono traslati fuori dall’ingresso IN.

Il numero che l’utente fornisce per il parametro di ingresso N sta a specificare il numero dei bit nella misura in cui eseguire la rotazione.

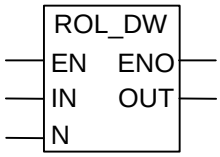
A seconda dall’operazione, la rotazione avviene tramite il bit A1. Il bit A0 della parola di stato viene resettato a 0.

Sono disponibili le seguenti operazioni di rotazione:

- ROL_DW Fai ruotare doppia parola verso sinistra (a 32 bit)
- ROR_DW Fai ruotare doppia parola verso destra (a 32 bit)

11.2.2 ROL_DW Fai ruotare doppia parola verso sinistra (a 32 bit)

Simbolo

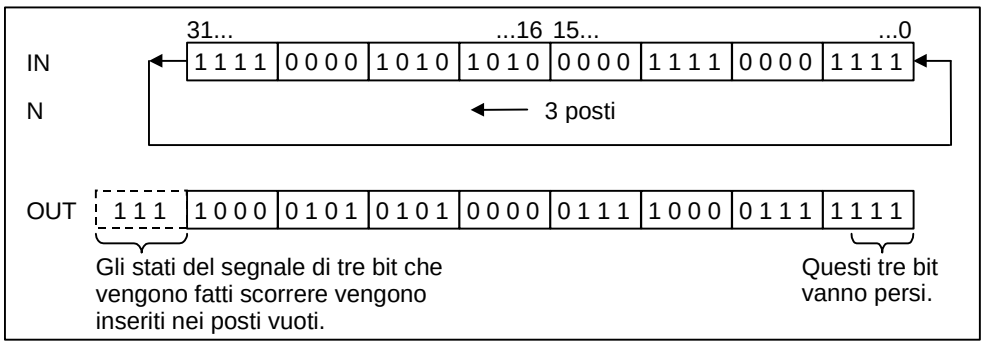


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DWORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire lo scorrimento
OUT	DWORD	E, A, M, L, D	Doppia parola di risultato dell'operazione di rotazione

Descrizione dell'operazione

ROL_DW (Fai ruotare doppia parola verso sinistra (a 32 bit)) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione ROL_DW fa ruotare a sinistra, bit per bit, l'intero contenuto dell'ingresso IN. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. Se N è maggiore di 32, la doppia parola IN viene fatta ruotare di ((N-1) modulo 32)+1 posizioni. Le posizioni di bit che vengono fatte scorrere da destra vengono occupate con lo stato di segnale dei bit che sono stati fatti ruotare a sinistra (rotazione a sinistra). La doppia parola di risultato dell'operazione di trasferimento può essere rilevato all'uscita OUT. L'operazione ROL_DW imposta a "0" i bit A0 e OV se N è diverso da 0.

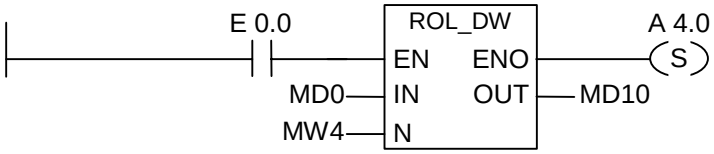
ENO ha lo stesso stato di segnale di EN.



Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

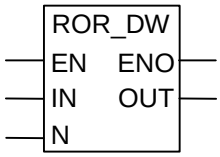
Esempio



Il box ROL_DW viene attivato se E 0.0 = 1. MD0 viene caricato e fatto ruotare a sinistra nella misura del numero di bit specificato in MW4. La doppia parola di risultato viene scritta a MD10. A 4.0 viene impostata.

11.2.3 ROR_DW Fai ruotare doppia parola verso destra (a 32 bit)

Simbolo

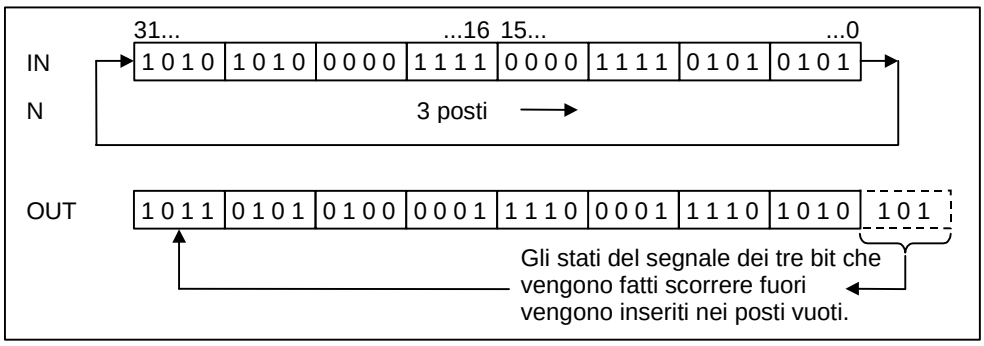


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN	DWORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit nella misura in cui eseguire la rotazione
OUT	DWORD	E, A, M, L, D	Doppia parola di risultato dell'operazione di rotazione

Descrizione dell'operazione

ROR_DW (Fai ruotare doppia parola verso destra (a 32 bit)) viene attivata se lo stato di segnale all'ingresso di abilitazione (EN) è "1". L'operazione ROR_DW fa ruotare a destra, bit per bit, l'intero contenuto dell'ingresso IN. L'ingresso N specifica il numero di posizioni di bit nella misura in cui va effettuato lo scorrimento. Se N è maggiore di 32, la doppia parola IN viene fatta ruotare di ((N-1) modulo 32)+1 posizioni. Le posizioni di bit che vengono fatte scorrere da sinistra vengono occupate con lo stato di segnale dei bit che sono stati fatti ruotare a destra (rotazione a destra). La doppia parola di risultato dell'operazione di trasferimento può essere rilevato all'uscita OUT. L'operazione ROR_DW imposta a "0" i bit A0 e OV se N è diverso da 0.

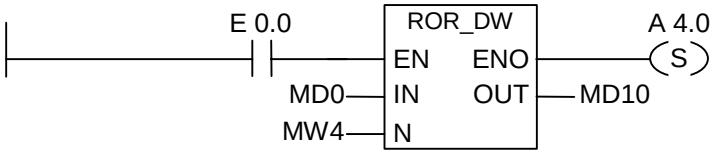
ENO ha lo stesso stato di segnale di EN.



Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	x	x	x	x	-	x	x	x	1

Esempio



Il box ROR_DW viene attivato se E 0.0 = 1. MD0 viene caricato e fatto ruotare a destra nella misura del numero di bit specificato in MW4. La doppia parola di risultato viene scritta in MD10. A 4.0 viene impostata.

12 Operazioni di bit di stato

12.1 Sommario delle operazioni di bit di stato

Descrizione

Le operazioni di bit di stato sono operazioni logiche combinatorie di bit che operano con i bit della parola di stato. Ognuna di queste operazioni reagisce ad una delle seguenti condizioni indicate da uno o più bit della parola di stato:

- Il bit di risultato binario (BIE ---I I---) viene impostato (ovvero, il suo stato di segnale è 1).
- In un'operazione matematica si verifica un overflow (OV ---I I---) o un overflow con memoria (OS ---I I---).
- Il risultato di un'operazione non è amesso (UO ---I I---).
- Il risultato di una funzione matematica rispetto allo 0 può essere:
  == 0, <> 0, > 0, < 0, >= 0, <= 0.

Le operazioni di bit di stato combinano il risultato della loro interrogazione di stato di segnale con il precedente risultato dell'operazione logica in conformità alla tabella della verità AND, se connesse in serie. Quando un'operazione di bit di stato viene collegata in parallelo, essa combina il suo risultato con il precedente RLC secondo la tabella della verità OR .

Parola di stato

La parola di stato è un registro nella memoria della CPU che contiene bit a cui si può fare riferimento nell'operando di operazioni logiche di bit e a parola.Struttura della parola di stato:

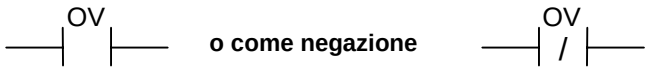
2^{15} ... ...2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER

E´ possibile valutare i bit nella parola di stato

- con operazioni con i numeri in virgola fissa,
- con operazioni con i numeri in virgola mobile.

12.2 OV ---| |--- Bit di anomalia: overflow

Simbolo



Descrizione dell'operazione

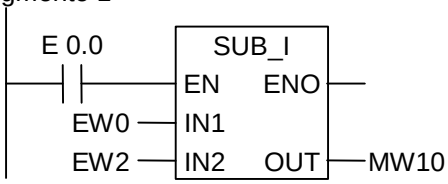
OV ---| |--- (Bit di anomalia overflow) e **OV ---| / |---** (Bit di anomalia negato overflow) servono per riconoscere un'eccedenza (overflow) nella operazione matematica appena eseguita. Ciò significa che il risultato dell'operazione si trova al di fuori del campo positivo o negativo permesso. Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo invece, il risultato viene combinato con RLC tramite OR.

Parola di stato

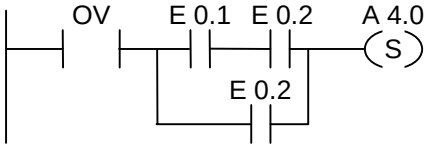
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio

Segmento 1



Segmento 2



Il box viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Se il risultato dell'operazione matematica EW0 - EW2 è al di fuori del campo permesso per un numero intero, viene impostato il bit OV.

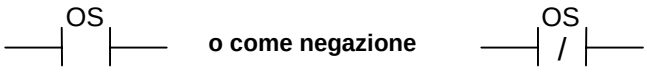
Il risultato dell'interrogazione di OV (per verificare un'eccedenza) è = 1. A 4.0 viene impostata se l'interrogazione per la verifica dell'eccedenza è = 1, e RLC del percorso di corrente 2 è = 1.

Nota

L'interrogazione per verificare un'eccedenza si rende necessaria solo a causa della presenza di due percorsi di corrente separati all'interno della rete. In caso contrario, se il risultato si trova al di fuori del campo permesso, è possibile utilizzare l'uscita ENO dell'operazione matematica che ha come stato di segnale "0".

12.3 OS ---| |--- Bit di anomalia: overflow con memoria

Simbolo



Descrizione dell'operazione

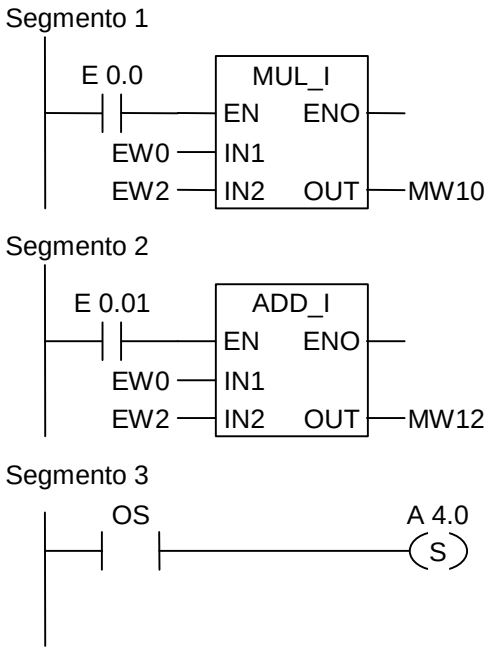
OS ---| |--- (Bit di anomalia overflow con memoria) e **OS ---| / |---** (Bit di anomalia negato overflow con memoria) servono per riconoscere un'eccedenza (overflow) in un'operazione matematica, che viene poi memorizzata. Se il risultato dell'operazione si trova al di fuori del campo positivo o negativo permesso, viene impostato il bit OS della parola di stato. Al contrario del bit OV, che viene riscritto durante le operazioni matematiche successive, il bit OS salva l'eccedenza (overflow) che si è verificata. Il bit OS rimane impostato fin quando permane nel blocco.

Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo, invece, il risultato viene combinato con RLC tramite OR.

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



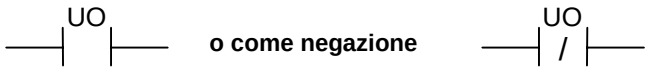
Il box MUL_I viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Il box ADD_I viene attivato dallo stato di segnale "1" all'ingresso E 0.1. Se il risultato di una delle operazioni matematiche si trova al di fuori del campo permesso per un numero intero, il bit OS della parola di stato viene impostato ad "1". A 4.0 viene impostata se l'interrogazione per la verifica dell'eccedenza con memoria è "1".

Nota

L'interrogazione per verificare un'eccedenza (overflow) con memoria si rende necessaria solo a causa della presenza di due percorsi di corrente separati all'interno della rete. Altrimenti è anche possibile collegare l'uscita ENO della prima operazione matematica all'ingresso EN della seconda operazione matematica (collegamento in cascata).

12.4 UO ---| |--- Bit di anomalia: operazione non ammessa

Simbolo



Descrizione dell'operazione

UO ---| |--- (Bit di anomalia operazione non ammessa) e **UO ---| / |---** (Bit di anomalia negato operazione non ammessa) vengono utilizzate per stabilire se il risultato di un'operazione matematica è ”non ammesso” (ovvero, se uno dei valori di una operazione matematica non sia un numero in virgola mobile valido).

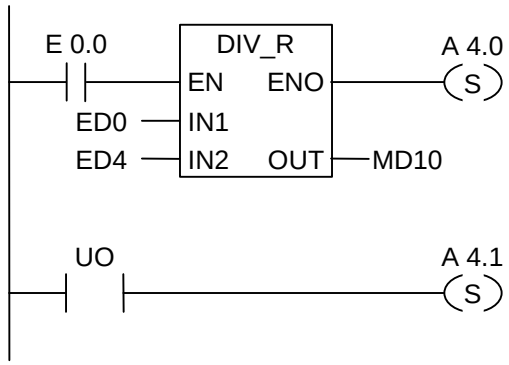
Se il risultato di un'operazione matematica con numeri in virgola mobile (UO) non è ammesso, l'interrogazione di stato di segnale produce 1. Se la combinazione A1 e A0 indica "non ammesso", il risultato dell'interrogazione di stato di segnale sarà "0".

Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo invece, il risultato viene combinato con RLC tramite OR.

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempio

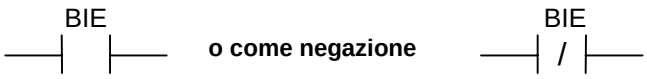


Il box viene attivato dallo di segnale "1" all'ingresso E 0.0. Se il valore di ED0 o ED4 non è un numero in virgola mobile valido, la operazione matematica è non ammessa. Se lo stato di segnale di EN è "1" (attivo), e si verifica un errore durante la esecuzione della funzione DIV_R; sarà "0" lo stato di segnale di ENO.

L'uscita A 4.1 viene impostata se l'operazione DIV_R viene eseguita, ma uno dei valori non risulta essere un **numero in virgola mobile** valido.

12.5 BIE ---| |--- Bit di anomalia: registro BIE

Simbolo



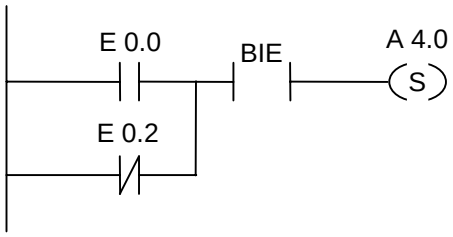
Descrizione dell'operazione

BIE ---| |--- (Bit di anomalia registro BIE) e **BIE ---| / |---** (Bit di anomalia negato registro BIE) interrogano lo stato di segnale del bit BIE dell parola di stato. Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo, invece, il risultato viene combinato con RLC tramite OR. Il bit BIE viene adoperato come collegamento tra l'elaborazione di bit e quella a parole.

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempio

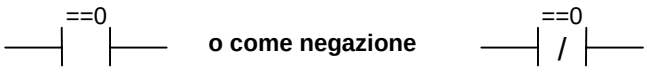


A 4.0 viene impostata se E 0.0 = 1 o E 0.2 = 0, e se, oltre a questo RLC, il bit BIE = 1.

(o = operazione OR, e = operazione AND)

12.6 ==0 ---| |--- Bit di risultato uguale a 0

Simbolo



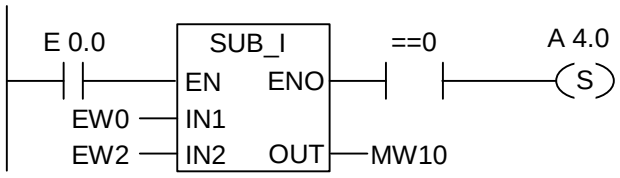
Descrizione dell'operazione

==0 ---| |--- (Bit di risultato uguale a 0) e **==0 ---| / |---** (Bit di risultato negato uguale a 0) stabiliscono se il risultato di un'operazioni matematica è uguale a 0. Queste operazioni interrogano i bit di visualizzazione A1 e A0 nella parola di stato, per stabilire il rapporto del risultato con "0". Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo, il risultato è invece combinato con RLC tramite OR.

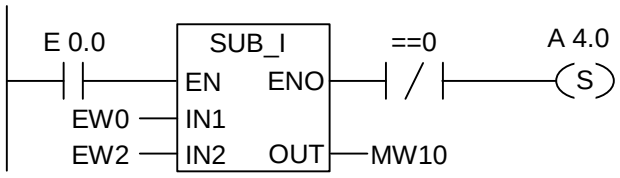
Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempi



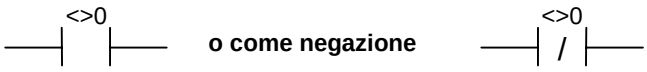
Il box viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Se il valore di EW0 è uguale al valore di EW2, il risultato dell'operazione matematica EW0 - EW2 è uguale a "0". A 4.0 viene impostata se l'operazione è stata eseguita senza errori e il risultato è uguale a "0".



A 4.0 viene impostata se l'operazione è stata eseguita correttamente, e il risultato non è uguale a "0".

12.7 <>0 ---| |--- Bit di risulato diverso da 0

Simbolo



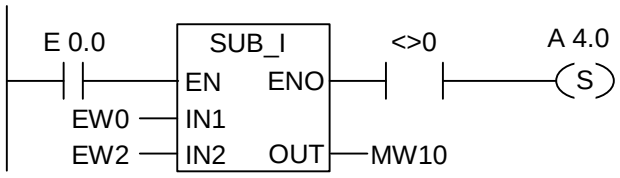
Descrizione dell'operazione

<>0 ---| |--- (Bit di risultato diverso da 0) e **<>0 ---| / |---** (Bit di risultato negato diverso da 0) stabiliscono se il risultato di un'operazioni matematica è diverso da 0. Queste operazioni interrogano i bit di visualizzazione A1 e A0 nella parola di stato, per stabilire il rapporto del risultato con "0". Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo, il risultato è invece combinato con RLC tramite OR.

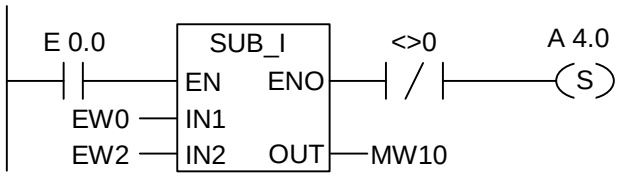
Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempi



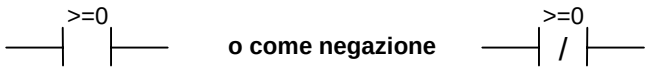
Il box viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Se il valore di EW0 è diverso dal valore di EW2, il risultato dell'operazione matematica EW0 - EW2 è diverso da "0". A 4.0 viene impostata se l'operazione è stata eseguita senza errori, e il risultato è diverso da "0".



A 4.0 viene impostata se l'operazione è stata eseguita correttamente, e il risultato è uguale a "0".

12.8 >=0 ---| |--- Bit di risultato maggiore o uguale a 0

Simbolo



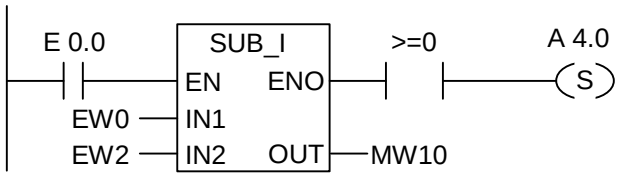
Descrizione dell'operazione

>=0 ---| |--- (Bit di risultato maggiore o uguale a 0) e **>=0 ---| / |---** (Bit di risultato negato maggiore uguale a 0) stabiliscono se il risultato di un'operazioni matematica è maggiore uguale a 0. Queste operazioni interrogano i bit di visualizzazione A1 e A0 nella parola di stato, per stabilire il rapporto del risultato con "0". Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con il RLC tramite AND; nei collegamenti in parallelo, il risultato è invece combinato con il RLC tramite OR.

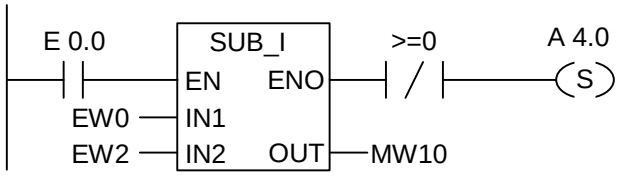
Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempi



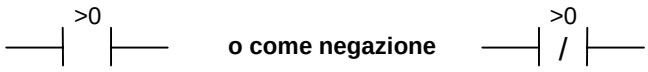
Il box viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Se il valore di EW0 è maggiore o uguale al valore di EW2, il risultato dell'operazione matematica EW0 - EW2 è maggiore o uguale a "0". A 4.0 viene impostata se l'operazione è stata eseguita senza errori, e il risultato è maggiore o uguale a "0".



A 4.0 viene impostata se l'operazione è stata eseguita correttamente e il risultato non è maggiore o uguale a "0".

12.9 >0 ---| |--- Bit di risultato maggiore di 0

Simbolo



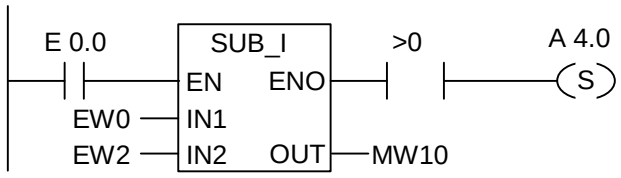
Descrizione dell'operazione

>0 ---| |--- (Bit di risultato maggiore di 0) e **>0 ---| / |---** (Bit di risultato negato maggiore di 0) stabiliscono se il risultato di un'operazioni matematica è maggiore di 0. Queste operazioni interrogano i bit di visualizzazione A1 e A0 nella parola di stato, per stabilire il rapporto del risultato con "0". Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo, il risultato è invece combinato con RLC tramite OR.

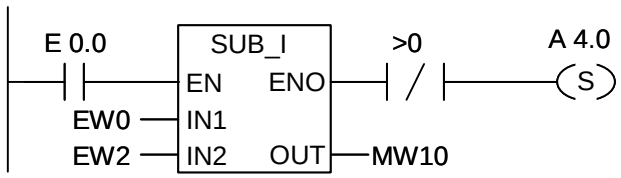
Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



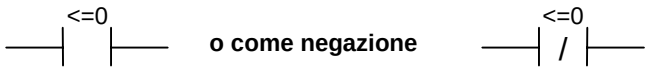
Il box viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Se il valore di EW0 è maggiore del valore di EW2, il risultato dell'operazione matematica EW0 - EW2 è maggiore di "0". A 4.0 viene impostata se l'operazione è stata eseguita senza errori, e il risultato è maggiore di "0".



A 4.0 viene impostata se l'operazione è stata eseguita correttamente e il risultato non è maggiore di "0".

12.10 <=0 ---| |--- Bit di risultato minore o uguale a 0

Simbolo



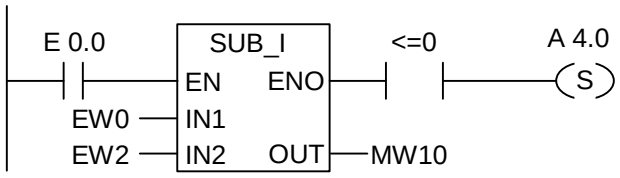
Descrizione dell'operazione

<=0 ---| |--- (Bit di risultato minore o uguale a 0) e **<=0 ---| / |---** (Bit di risultato negato minore o uguale a 0) stabiliscono se il risultato di un'operazioni matematica è minore o uguale a 0. Queste operazioni interrogano i bit di visualizzazione A1 e A0 nella parola di stato, per stabilire il rapporto del risultato con "0". Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo, il risultato è invece combinato con RLC tramite OR.

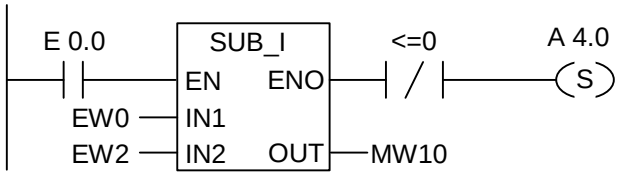
Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempi



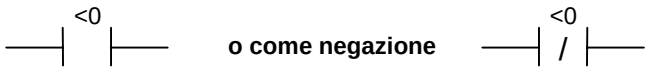
Il box viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Se il valore di EW0 è minore o uguale al valore di EW2, il risultato dell'operazione matematica EW0 - EW2 è minore o uguale a "0". A 4.0 viene impostata se l'operazione è stata eseguita senza errori, e il risultato è minore o uguale a "0".



A 4.0 viene impostata se l'operazione è stata eseguita correttamente e il risultato non è minore o uguale a "0".

12.11 <0 ---| |--- Bit di risultato minore di 0

Simbolo



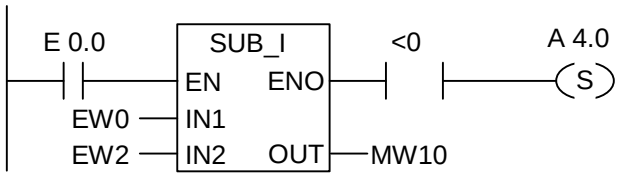
Descrizione dell'operazione

<0 ---| |--- (Bit di risultato minore di 0) e **<0 ---| / |---** (Bit di risultato negato minore di 0) stabiliscono se il risultato di un'operazioni matematica è minore di 0. Queste operazioni interrogano i bit di visualizzazione A1 e A0 nella parola di stato, per stabilire il rapporto del risultato con "0". Nei collegamenti in serie, il risultato dell'interrogazione viene combinato con RLC tramite AND; nei collegamenti in parallelo, il risultato è invece combinato con RLC tramite OR.

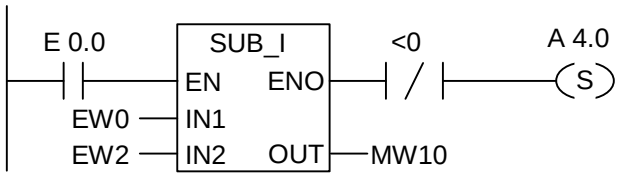
Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempi



Il box viene attivato dallo stato di segnale "1" all'ingresso E 0.0. Se il valore di EW0 è minore del valore di EW2, il risultato dell'operazione matematica EW0 - EW2 è minore di "0". A 4.0 viene impostata se l'operazione è stata eseguita senza errori, e il risultato è minore di "0".



A 4.0 viene impostata se l'operazione è stata eseguita correttamente, e il risultato non è minore di "0".

13 Operazioni di temporizzazione

13.1 Sommario delle operazioni di temporizzazione

Descrizione

Al paragrafo "Aree di memoria e componenti di un temporizzatore" si trovano informazioni e la selezione dei temporizzatori.

Sono disponibili le seguenti operazioni di temporizzatore:

- S_IMPULS Parametrizza e avvia bobina temporizzatore come impulso
- S_VIMP Parametrizza e avvia bobina temporizzatore come impulso prolungato
- S_EVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione
- S_SEVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione con memoria
- S_AVERZ Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione
- ---(SI) Avvia bobina temporizzatore come impulso
- ---(SV) Avvia bobina temporizzatore come impulso prolungato
- ---(SE) Avvia bobina temporizzatore come ritardo all'inserzione
- ---(SS) Avvia bobina temporizzatore come ritardo all'inserzione con memoria
- ---(SA) Avvia bobina temporizzatore come ritardo alla disinserzione

13.2 Aree di memoria e componenti di un temporizzatore

Area di memoria

I temporizzatori hanno un'area riservata nella memoria della CPU. Quest'area di memoria riserva una parola a 16 bit per ogni operando del temporizzatore. Il set di operazioni logiche KOP supporta 256 temporizzatori. Le parole di tempo rizzazione a disposizione per la CPU utilizzata sono riportate nei relativi dati tecnici.

Le seguenti funzioni hanno accesso all'area di memoria del temporizzatore:

- Operazioni di temporizzazione
- Aggiornamento di parole di temporizzazione mediante generatore di clock. Questa funzione della CPU nello stato di funzionamento RUN decrementa un determinato valore di un'unità in intervalli definiti dalla base di tempo finché il valore temporale non è uguale a zero.

Valore di tempo

I bit da 0 a 9 della parola di temporizzazione contengono il valore temporale in codice binario. Questo valore specifica un numero di unità. L'aggiornamento del tempo decrementa il valore di un'unità in intervalli definiti dalla base di tempo. Il decremento continua finché il valore temporale non è uguale a zero. Si può caricare un valore di tempo in formato binario, esadecimale o decimale codificato in binario (BCD).

Si può precaricare un valore di tempo in uno dei seguenti formati:

- W#16#wxyz
  - laddove w = base di tempo (ossia, l'intervallo di tempo o risoluzione)
  - laddove xyz = valore in formato BCD

- S5T#a**H**_b**M**_c**S**_d**MS**
  - laddove H = ore, M = minuti, S = secondi, MS = millisecondi;
    a, b, c, d vengono definiti dall'utente
  - La base di tempo viene selezionata automaticamente e il valore viene arrotondato al numero immediatamente inferiore rispetto ad essa.

Il valore di tempo massimo che si può immettere è 9.990 secondi, o 2H_46M_30S. Esempi:

S5TIME#4S --> 4 secondi
s5t#2h_15m --> 2 ore e 15 minuti
S5T#1H_12M_18S --> 1 ora, 12 minuti e 18 secondi

Base di tempo

I bit 12 e 13 della parola di temporizzazione contengono la base di tempo in codice binario. La base di tempo definisce l'intervallo di decremento di un'unità. La base di tempo più piccola è 10 ms; la più grande è 10 s.

Base di tempo	Codice binario per la base di tempo
10 ms	00
100 ms	01
1 s	10
10 s	11

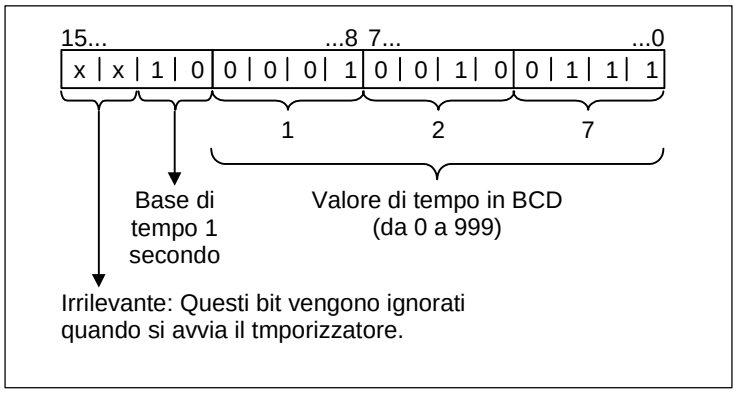
I valori non devono superare 2H_46M_30S. I valori con un'area o una risoluzione troppo grande (ad es. 2H_10MS) vengono arrotondati così da corrispondere alla tabella per l'area e la risoluzione. Il formato generale per il tipo di dati S5TIME ha i seguenti valori limite per l'area e la risoluzione:

Risoluzione	Area
0,01 secondi	Da 10MS a 9S_990MS
0,1 secondi	Da 100MS a 1M_39S_900MS
1 secondo	Da 1S a 16M_39S
10 secondi	Da 10S a 2H_46M_30S

Configurazione dei bit nella cella di tempo

Quando si avvia un temporizzatore, il contenuto della cella di tempo viene utilizzato come valore di tempo. I bit da 0 a 11 della cella di tempo registrano il valore di tempo in formato decimale codificato in binario (formato BCD: ogni serie di quattro bit contiene il codice binario di un valore decimale). I bit 12 e 13 registrano la base di tempo in codice binario.

La figura mostra il contenuto della cella di tempo caricato con il valore di temporizzazione 127 e una base di tempo di 1 secondo:

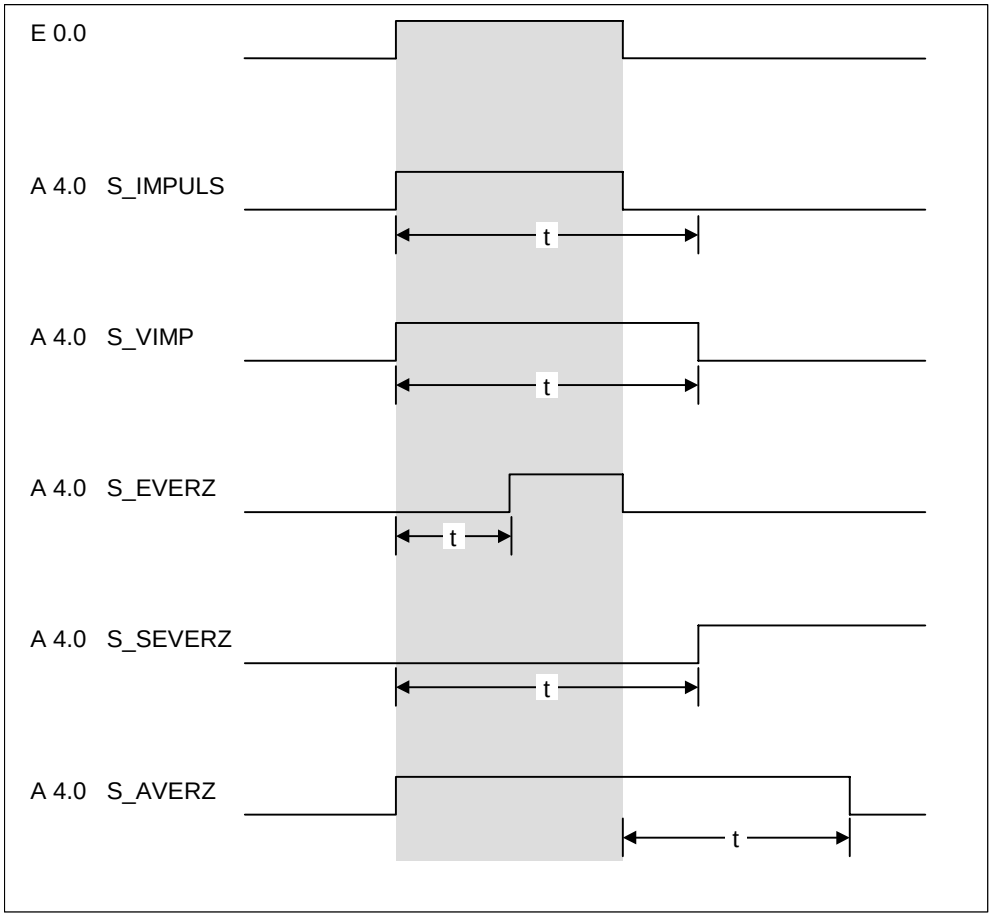


Lettura dell’ora e della base di tempo

Ogni box di temporizzatore contiene due uscite, DUAL e DEZ, per cui si può indicare un’indirizzo di parola. L’uscita DUAL fornisce il valore di tempo in formato binario. L’uscita DEZ fornisce la base di tempo e il valore di tempo in formato decimale codificato in binario (BCD).

Scelta del giusto temporizzatore

La figura illustra cinque tipi di temporizzatori che sono stati descritti in questo capitolo. Questo riepilogo vuole essere d’aiuto all’utente nella scelta del temporizzatore più adeguato ai suoi fini di utilizzo.

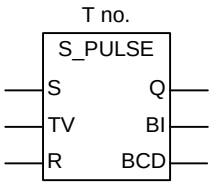


Temporizzatore	Descrizione
S_IMPULS Avvia temporizzatore come impulso	Il tempo massimo in cui il segnale di uscita resta a 1, è uguale al valore di tempo programmato t. Il segnale di uscita resta a 1 per un tempo più breve se il segnale di ingresso passa a 0.
S_VIMP Avvia temporizzatore come impulso prolungato	Il segnale di uscita resta a 1 per la durata programmata, indipendentemente dal tempo che il segnale di ingresso resta a 1.
S_EVERZ Avvia temporizzatore come ritardo all'inserzione	Il segnale di uscita è 1 solo quando è trascorso il tempo programmato e il segnale di ingresso è ancora 1.
S_SEVERZ Avvia temporizzatore come ritardo all'inserzione con memoria	Il segnale di uscita passa da 0 a 1 solo quando è trascorso il tempo programmato, indipendentemente dal tempo in cui il segnale di ingresso resta a 1.
S_AVERZ Avvia temporizzatore come ritardo alla disinserzione	Il segnale di uscita è 1 quando il segnale di ingresso è 1. Il segnale di uscita resta a 1 per la durata programmata. Il tempo viene avviato quando il segnale di ingresso cambia da 1 a 0.

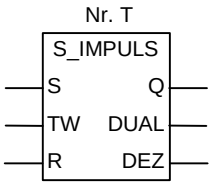
13.3 S_IMPULS Parametrizza e avvia bobina temporizzatore come impulso (SI)

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
T no.	Nr. T	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU
S	S	BOOL	E, A, M, L, D	Ingresso di avviamento
TV	TW	S5TIME	E, A, M, L, D	Valore di tempo di default
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, L, D	Valore di tempo attuale, formato binario
BCD	DEZ	WORD	E, A, M, L, D	Valore di tempo attuale, formato BCD
Q	Q	BOOL	E, A, M, L, D	Stato del temporizzatore

Descrizione dell'operazione

S_IMPULS (Parametrizza e avvia bobina temporizzatore come impulso) avvia un temporizzatore specificato se all'ingresso di avviamento S si trova un fronte di salita. Per abilitare un temporizzatore è sempre necessaria una modifica di segnale. Il temporizzatore continua ad operare finchè lo stato di segnale all'ingresso S è "1", e almeno per la durata del valore di tempo indicato all'ingresso TV/TW . Lo stato di segnale all'uscita Q è "1" fin quando il temporizzatore è attivo. Se ha luogo un cambiamento di segnale da "1" a "0" all'ingresso S prima che il tempo sia trascorso, il temporizzatore viene arrestato. In questo caso, lo stato di segnale dell'uscita Q è "0".

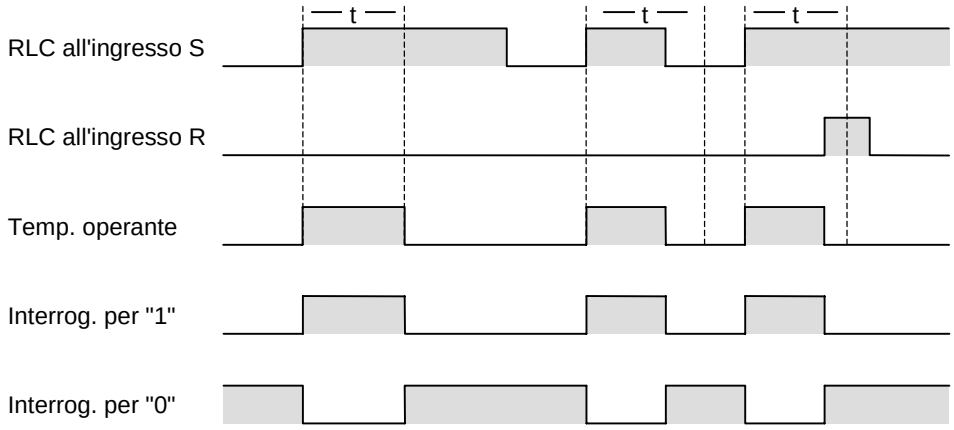
Mentre il temporizzatore è in funzionamento, un cambiamento da "0" a "1" all'ingresso di resettaggio R del temporizzatore resetta il temporizzatore. Anche il valore di tempo attuale e la base di tempo vengono impostati a zero. Lo stato di segnale di "1" all'ingresso R del temporizzatore non ha alcuna incidenza se il temporizzatore non è operante.

Il valore di tempo attuale può essere letto alle uscite BI/DUAL e BCD/DEZ. Il valore di tempo all'uscita BI/DUAL è in formato di cifra in codice binario, il valore all'uscita BCD/DEZ è in formato di cifra decimale in codice binario. Il valore di tempo attuale corrisponde al valore iniziale di TV/TW, dal quale viene sottratto il valore di tempo trascorso dal momento di avvio del temporizzatore.

Vedere anche "Aree di memoria e componenti di un temporizzatore".

Diagramma degli impulsi

Caratteristiche del temporizzatore come impulso:

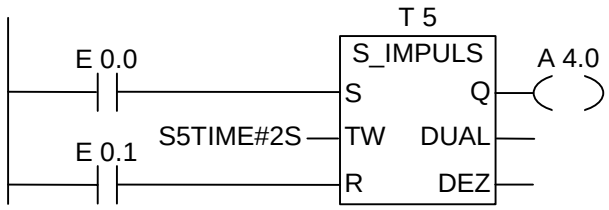


t = tempo programmato

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



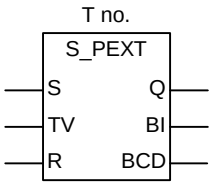
Se lo stato di segnale dell'ingresso E 0.0 cambia da "0" a "1" (ovvero se c'è un fronte di salita nel RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua ad essere attivo con il tempo specificato di due secondi (2 s) finché l'ingresso E 0.0 = 1. Se lo stato di segnale di E 0.0 cambia da "1" a "0" prima che il tempo sia trascorso, il temporizzatore viene arrestato. Se lo stato di segnale di E 0.0 cambia da "1" a "0" prima che il tempo sia trascorso, il temporizzatore viene resettato Se lo stato di segnale di E 0.1 cambia da "0" a "1" mentre il temporizzatore è attivo, il temporizzatore viene resettato.

L'uscita A 4.0 è "1" finché il temporizzatore è attivo, e "0" se il tempo è trascorso od è stato resettato.

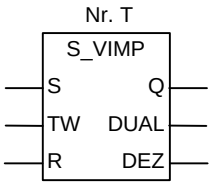
13.4 S_VIMP Parametrizza e avvia bobina temporizzatore come impulso prolungato (SV)

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
T no.	Nr. T	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU
S	S	BOOL	E, A, M, L, D	Ingresso di avviamento
TV	TW	S5TIME	E, A, M, L, D	Valore di tempo di default
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, L, D	Valore di tempo attuale, formato binario
BCD	DEZ	WORD	E, A, M, L, D	Valore di tempo attuale, formato BCD
Q	Q	BOOL	E, A, M, L, D	Stato del temporizzatore

Descrizione dell'operazione

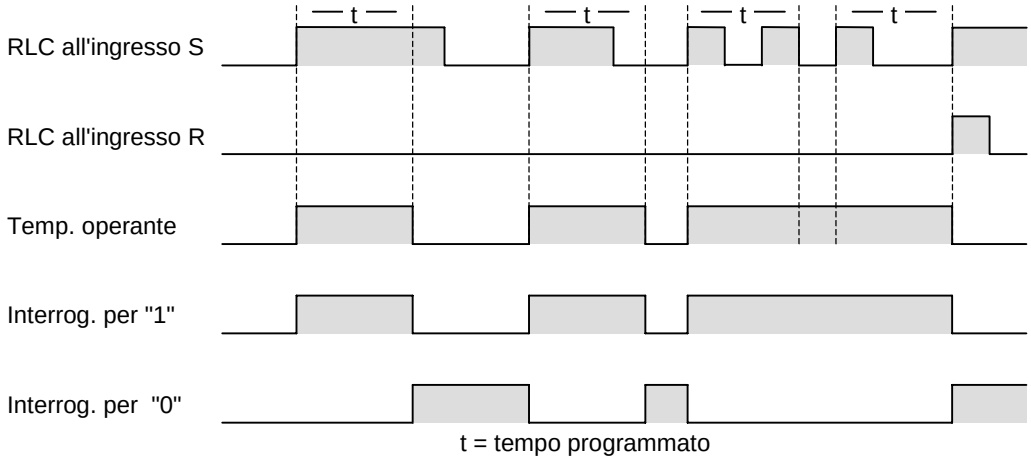
S_VIMP (Parametrizza e avvia bobina temporizzatore come impulso prolungato) avvia un temporizzatore specificato se all'ingresso di avviamento S si trova un fronte di salita. Per abilitare un temporizzatore è sempre necessaria una modifica di segnale. Il temporizzatore continua ad operare per tutta la durata del valore di tempo indicato all'ingresso TV/TW, anche se lo stato di segnale all'ingresso S diventa "0" prima che il tempo sia trascorso. Lo stato di segnale dell'uscita Q è "1" finché è attivo il temporizzatore.Il temporizzatore viene riavviato con il tempo specificato se lo stato di segnale all'ingresso S cambia da "0" a "1" mentre il temporizzatore è in funzione.

Il valore di tempo attuale può essere letto alle uscite BI/DUAL e BCD/DEZ. Il valore di tempo all'uscita BI/DUAL è in formato di cifra in codice binario; il valore all'uscita BCD/DEZ è in formato di cifra decimale in codice binario. Il valore di tempo attuale corrisponde al valore iniziale di TV/TW dal quale viene sottratto il valore di tempo trascorso dal momento di avvio del temporizzatore.

Vedere anche "Aree di memoria e componenti di un temporizzatore".

Diagramma degli impulsi

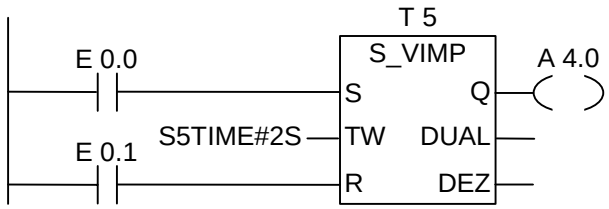
Caratteristiche del temporizzatore come impulso prolungato:



Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio

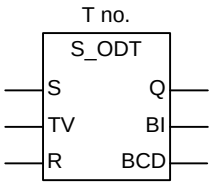


Se lo stato di segnale dell'ingresso E 0.0 cambia da "0" a "1" (ovvero se c'è un fronte di salita nel RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua ad essere attivo con il tempo specificato di due secondi (2 s) senza considerazione del fronte di discesa all'ingresso S. Se lo stato di segnale di E 0.0 cambia da "0" a "1" prima che il tempo sia trascorso, il temporizzatore viene riavviato. Se lo stato di segnale di E 0.1 cambia da "0" a "1" mentre il temporizzatore scorre, il temporizzatore viene resettato. L'uscita A 4.0 è "1" finché è attivo il temporizzatore.

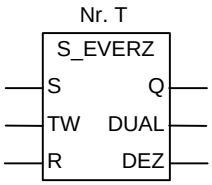
13.5 S_EVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione (SE)

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
T no.	Nr. T	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU
S	S	BOOL	E, A, M, L, D	Ingresso di avviamento
TV	TW	S5TIME	E, A, M, L, D	Valore di tempo di default
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, L, D	Valore di tempo attuale, formato binario
BCD	DEZ	WORD	E, A, M, L, D	Valore di tempo attuale, formato BCD
Q	Q	BOOL	E, A, M, L, D	Stato del temporizzatore

Descrizione dell'operazione

S_EVERZ (Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione) avvia un temporizzatore specificato se all'ingresso di avviamento S si trova un fronte di salita. Per abilitare un temporizzatore è sempre necessaria una modifica di segnale. Il temporizzatore continua ad operare per tutta la durata del valore di tempo indicato all'ingresso TV/TW, fino a quando lo stato di segnale all'ingresso S rimane positivo. Lo stato di segnale dell'uscita Q è "1" se il tempo è trascorso senza errori e lo stato di segnale dell'ingresso S è "1". Se lo stato di segnale all'ingresso S cambia da "1" a "0" mentre il temporizzatore è attivo, il temporizzatore viene arrestato. In questo caso, lo stato di segnale all'uscita Q è "0".

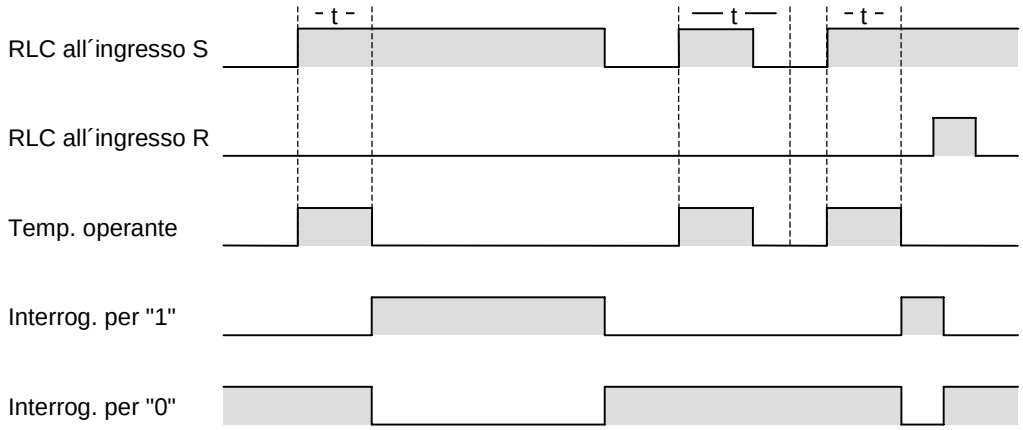
Mentre il temporizzatore è attivo, esso viene resettato se vi è un cambiamento da "0" a "1" all'ingresso di resettaggio R. Anche il valore di tempo attuale e la base di tempo vengono impostati a zero. Lo stato di segnale all'uscita Q è quindi "0". Inoltre, il temporizzatore viene resettato anche quando lo stato di segnale all'ingresso di resettaggio R è "1", mentre il temporizzatore non è attivo, ed RLC è "1" all'ingresso S.

Il valore di tempo attuale può essere letto alle uscite BI/DUAL e BCD/DEZ. Il valore di tempo all'uscita BI/DUAL è in formato di cifra in codice binario, il valore all'uscita BCD/DEZ è in formato di cifra decimale in codice binario. Il valore di tempo attuale corrisponde al valore iniziale di TV/TW, dal quale viene sottratto il valore di tempo trascorso dal momento di avvio del temporizzatore.

Vedere anche "Aree di memoria e componenti di un temporizzatore".

Diagramma degli impulsi

Caratteristiche del temporizzatore come ritardo all'inserzione:

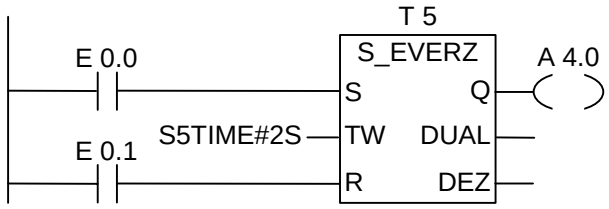


t = tempo programmato

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio

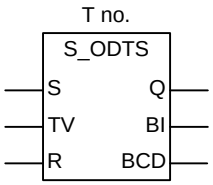


Se lo stato di segnale dell'ingresso E 0.0 cambia da "0" a "1" (ovvero se c'è un fronte di salita nel RLC), il temporizzatore T5 viene avviato. Se il tempo specificato di due secondi (2 sec) è trascorso e lo stato di segnale all'ingresso E 0.0 è sempre "1", l'uscita A 4.0 è "1". Se lo stato di segnale di E 0.0 cambia da "1" a "0", il temporizzatore viene arrestato e l'uscita A 4.0 è "0". (Se lo stato di segnale dell'ingresso E 0.1 cambia da "0" a "1", il temporizzatore viene resettato, a prescindere dal fatto se il temporizzatore sia o no attivo).

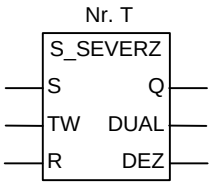
13.6 S_SEVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inerzione con memoria (SS)

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
T no.	Nr. T	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU
S	S	BOOL	E, A, M, L, D	Ingresso di avviamento
TV	TW	S5TIME	E, A, M, L, D	Valore di tempo di default
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, L, D	Valore di tempo attuale, formato binario
BCD	DEZ	WORD	E, A, M, L, D	Valore di tempo attuale, formato BCD
Q	Q	BOOL	E, A, M, L, D	Stato del temporizzatore

Descrizione dell'operazione

S_SEVERZ (Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione con memoria) avvia un temporizzatore specificato se all'ingresso di avviamento S si trova un fronte di salita. Per abilitare un temporizzatore è sempre necessaria una modifica di segnale. Il temporizzatore continua ad operare per la durata del valore di tempo indicato all'ingresso TV/TW, anche se lo stato di segnale all'ingresso S diventa "0" prima che il tempo sia trascorso. Lo stato di segnale dell'uscita Q è "1" quando il tempo è trascorso, e indipendentemente dallo stato di segnale all'ingresso S. Il temporizzatore viene riavviato con il tempo specificato se lo stato di segnale all'ingresso S cambia da "0" a "1" mentre il temporizzatore è in funzione.

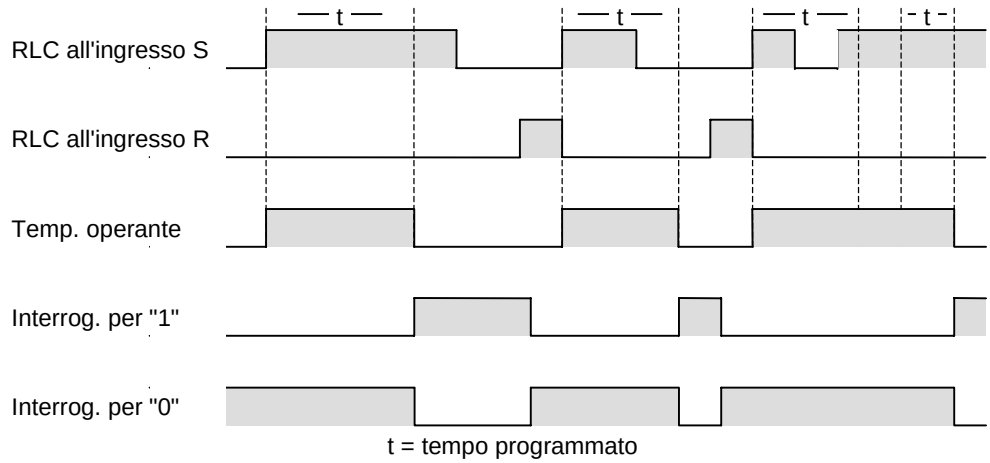
Il temporizzatore viene resettato se vi è un cambiamento da "0" a "1" all'ingresso di resettaggio R del temporizzatore, indipendentemente dal RLC all'ingresso S. Lo stato di segnale all'uscita Q è quindi "0".

Il valore di tempo attuale può essere letto alle uscite BI/DUAL e BCD/DEZ. Il valore di tempo all'uscita BI/DUAL è in formato di cifra in codice binario, il valore all'uscita BCD/DEZ è in formato di cifra decimale in codice binario. Il valore di tempo attuale corrisponde al valore iniziale di TV/TW, dal quale viene sottratto il valore di tempo trascorso dal momento di avvio del temporizzatore.

Vedere anche "Aree di memoria e componenti di un temporizzatore".

Diagramma degli impulsi

Caratteristiche del temporizzatore come ritardo all'inserzione con memoria:

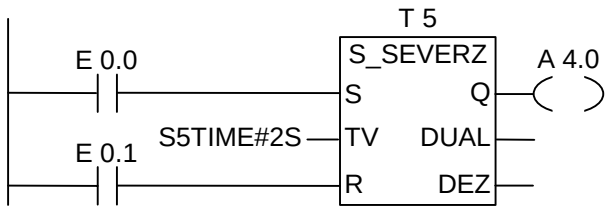


t = tempo programmato

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	x	x	x	1

Esempio

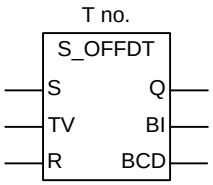


Se lo stato di segnale dell'ingresso E 0.0 cambia da "0" a "1" (ovvero se c'è un fronte di salita nel RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua ad essere attivo senza considerazione del cambiamento dello stato di segnale dell'ingresso E 0.0 da "1" a "0". Se lo stato di segnale di E 0.0 cambia da "0" a "1" prima che il tempo sia trascorso, il temporizzatore viene avviato nuovamente. L'uscita A 4.0 è "1" se il tempo è trascorso. (Se lo stato di segnale dell'ingresso E 0.1 cambia da "0" a "1", il temporizzatore viene resettato, a prescindere dal RLC in S).

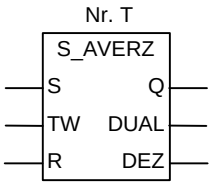
13.7 S_AVERZ Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione (SA)

Simbolo

Inglese



Tedesco



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
T no.	Nr. T	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU
S	S	BOOL	E, A, M, L, D	Ingresso di avviamento
TV	TW	S5TIME	E, A, M, L, D	Valore di tempo di default
R	R	BOOL	E, A, M, L, D	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, L, D	Valore di tempo attuale, formato binario
BCD	DEZ	WORD	E, A, M, L, D	Valore di tempo attuale, formato BCD
Q	Q	BOOL	E, A, M, L, D	Stato del temporizzatore

Descrizione dell'operazione

S_AVERZ (Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione) avvia un temporizzatore specificato se all'ingresso di avviamento S si trova un fronte di discesa. Per abilitare un temporizzatore è sempre necessaria una modifica di segnale. Il risultato di un'interrogazione di segnale all'uscita Q è "1" se lo stato di segnale all'ingresso S è "1", oppure se il temporizzatore è operante. Il temporizzatore viene arrestato quando lo stato di segnale all'ingresso S cambia da "0" a "1" mentre il temporizzatore è attivo. Il temporizzatore non viene riavviato finché lo stato di segnale all'ingresso S non cambia nuovamente da "1" a "0".

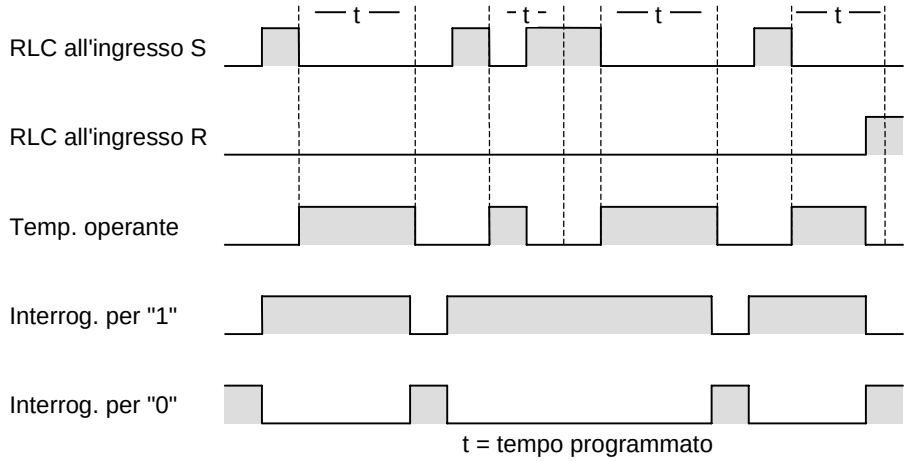
Mentre il temporizzatore è attivo, un cambiamento da "0" a "1" all'ingresso di resettaggio R del temporizzatore resetta il temporizzatore.

Il valore di tempo attuale può essere letto alle uscite BI/DUAL e BCD/DEZ. Il valore di tempo all'uscita BI/DUAL è in formato di cifra in codice binario, il valore all'uscita BCD/DEZ è in formato di cifra decimale in codice binario. Il valore di tempo attuale corrisponde al valore iniziale di TV/TW dal quale viene sottratto il valore di tempo trascorso dal momento di avvio del temporizzatore.

Vedere anche "Aree di memoria e componenti di un temporizzatore".

Diagramma degli impulsi

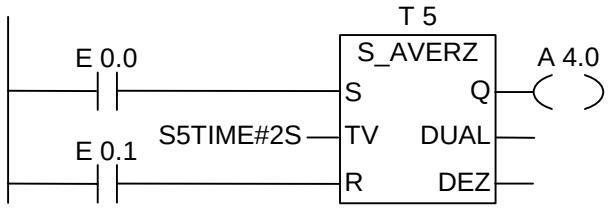
Caratteristiche del temporizzatore come ritardo alla disinserzione:



Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	x	x	x	1

Esempio



Se lo stato di segnale dell'ingresso E 0.0 cambia da "1" a "0", il temporizzatore viene avviato.

A 4.0 è "1" se lo stato di segnale di E 0.0 è "1", oppure se il temporizzatore è operante.
(Se lo stato di segnale dell'ingresso E 0.1 cambia da "0" a "1", mentre il tempo è attivo, il temporizzatore viene resettato).

13.8 ---(SI) Avvia bobina temporizzatore come impulso

Simbolo

Inglese	**Tedesco**
<T-Nr.>	<T-Nr.>
---(SP)	**---(SI)**
<Valore di tempo>	<Valore di tempo>

Parametro	**Tipo di dati**	**Area di memoria**	**Descrizione**
<T-Nr.>	TIMER	T	Numero di identificazione del temporizzatore; il campo dipende dalla CPU
<Valore di tempo>	S5TIME	E, A, M, L, D	Valore di tempo di default

Descrizione dell'operazione

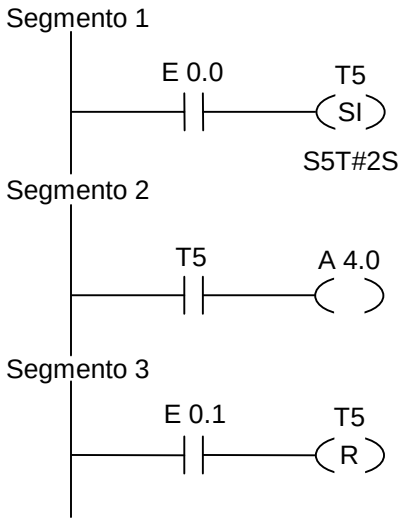
---(SI) (Avvia bobina temporizzatore come impulso) avvia il temporizzatore specificato con il **<Valore di tempo>,** se RLC ha un fronte di salita. Il temporizzatore continua a scorrere con il tempo specificato finché RLC è positivo ("1"). L'interrogazione di "1" dello stato di segnale da per risultato "1" finché scorre il temporizzatore. Se RLC passa da "1" a "0" prima che il tempo sia trascorso, il temporizzatore viene arrestato. In questo caso lo stato del contatore è "0".

Vedere anche "Aree di memoria e componenti di un temporizzatore" e S_IMPULS Parametrizza e avvia bobina temporizzatore come impulso (SI).

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	-	-	0

Esempio



Se lo stato di segnale di E 0.0 passa da "0" a "1" (fronte di salita in RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua a scorrere con il tempo specificato di 2 secondi, finché lo stato di segnale di E 0.0 è "1". Se lo stato di segnale all'ingresso E.0 0 passa da "1" a "0" prima che sia trascorso il tempo, il temporizzatore viene arrestato. L'uscita A 4.0 è 1 finché scorre il temporizzatore. Se lo stato di segnale all'ingresso E 0.1 passa da "0" a "1", il temporizzatore T5 viene resettato, ovvero viene arrestato, e il valore di tempo attuale viene impostato a "0".

13.9 ---(SV) Avvia bobina temporizzatore come impulso prolungato

Simbolo

Inglese	Tedesco
<T-Nr.>	<T-Nr.>
---(SE)	**---(SV)**
<Valore di tempo>	<Valore di tempo>

Parametro	Tipo di dati	Area di memoria	Descrizione
<Nr. T>	TIMER	T	Numero di identificazione del temporizzatore; il campo dipende dalla CPU
<Valore di tempo>	S5TIME	E, A, M, L, D	Valore di tempo di default

Descrizione dell'operazione

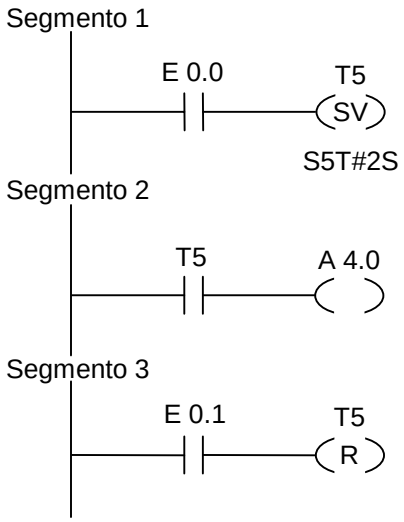
---(SV) (Avvia bobina temporizzatore come impulso prolungato) avvia il temporizzatore specificato con il **<valore di tempo>** se RLC ha un fronte di salita. Il temporizzatore continua a scorrere con il tempo specificato anche se RLC diventa "0" prima che il tempo sia trascorso. L'interrogazione di "1" dello stato di segnale produce "1" finché il temporizzatore scorre. Se RLC passa da "0" a "1" il temporizzatore viene riavviato mentre scorre.

Vedere anche "Aree di memoria e componenti di un temporizzatore" e S_VIMP Parametrizza e avvia bobina temporizzatore come impulso prolungato (SV).

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	-	-	0

Esempio



Se lo stato di segnale di E 0.0 passa da "0" a "1" (fronte di salita in RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua a scorrere con il tempo specificato, senza essere influenzato da un fronte di discesa in RLC. Se lo stato di segnale all'ingresso E 0 0 passa da "0" a "1" prima che sia trascorso il tempo, il temporizzatore viene riavviato. L'uscita A 4.0 è "1" finché scorre il temporizzatore. Se lo stato di segnale all'ingresso E 0.1 passa da "0" a "1", il temporizzatore T5 viene resettato, ovvero viene arrestato, e il valore di tempo attuale viene impostato a"0".

13.10 ---(SE) Avvia bobina temporizzatore come ritardo all'inserzione

Simbolo

Inglese	**Tedesco**
<T-Nr.>	<T-Nr.>
---(SD)	**---(SE)**
<Valore di tempo>	<Valore di tempo>

Parametro	**Tipo di dati**	**Area di memoria**	**Descrizione**
<Nr. T>	TIMER	T	Numero di identificazione del temporizzatore; il campo dipende dalla CPU
<Valore di tempo>	S5TIME	E, A, M, L, D	Valore di tempo di default

Descrizione dell'operazione

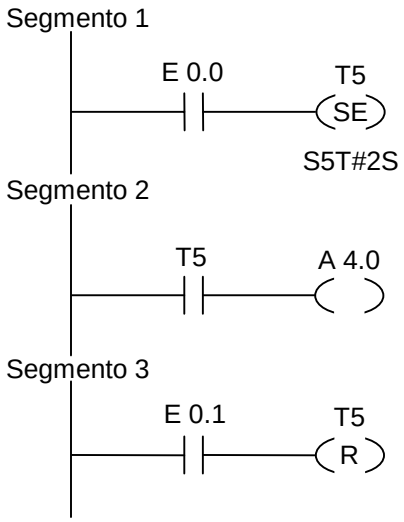
---(SE) (Avvia bobina temporizzatore come ritardo all'inserzione) avvia il temporizzatore specificato con il **<valore di tempo>**, se RLC ha un fronte di salita. Lo stato di segnale del temporizzatore è "1" se il **<valore di tempo>** è trascorso senza errori e RLC è sempre "1". Se RLC passa da "1" a "0" mentre il temporizzatore scorre, il temporizzatore viene resettato. In questo caso, l'interrogazione di "1" dello stato di segnale produce sempre il risultato "0".

Vedere anche "Aree di memoria e componenti di un temporizzatore" e S_EVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione (SE).

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	-	-	0

Esempio



Se lo stato di segnale di E 0.0 passa da "0" a "1" (fronte di salita in RLC), il temporizzatore T5 viene avviato. Se il tempo trascorre e lo stato di segnale all'ingresso E 0.0 è sempre "1", l'uscita A 4.0 è anche "1". Se lo stato di segnale all'ingresso E.0 0 passa da "1" a "0", il temporizzatore viene resettato, e A 4.0 è "0". Se lo stato di segnale all'ingresso E 0.1 passa da "0" a "1", il temporizzatore T5 viene resettato, ovvero viene arrestato, e il valore di tempo attuale viene impostato a "0".

13.11 ---(SS) Avvia bobina temporizzatore come ritardo all'inserzione con memoria (SS)

Simbolo

Inglese	**Tedesco**
<T-Nr.>	<T-Nr.>
---(SS)	**---(SS)**
<Valore di tempo>	<Valore di tempo>

Parametro	**Tipo di dati**	**Area di memoria**	**Descrizione**
<Nr. T>	TIMER	T	Numero di identificazione del temporizzatore; il campo dipende dalla CPU
<Valore di tempo>	S5TIME	E, A, M, L, D	Valore di tempo di default

Descrizione dell'operazione

---(SS) (Avvia bobina temporizzatore come ritardo all'inserzione con memoria) avvia il temporizzatore specificato se RLC ha un fronte di salita. Lo stato di segnale del temporizzatore è "1" se il tempo è trascorso. Un nuovo avvio del temporizzatore è possibile se questo è stato resettato in modo esplicito. Lo stato del temporizzatore può essere impostato a "0", solo se precedentemente resettato.

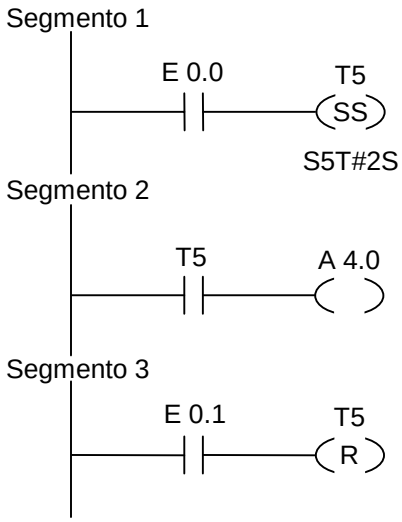
Il temporizzatore viene riavviato con il valore di tempo specificato se RLC passa da "0" a "1" mentre scorre il temporizzatore.

Vedere anche "Aree di memoria e componenti di un temporizzatore" e S_SEVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inerzione con memoria (SS).

Parola di stato

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLC**	**/ER**
Scrive:	-	-	-	-	-	0	-	-	0

Esempio



Se lo stato di segnale di E 0.0 passa da "0" a "1" (fronte di salita in RLC), il temporizzatore T5 viene avviato. Se lo stato di segnale all'ingresso E.0 0 passa da "0" a "1" prima che il tempo sia trascorso, il temporizzatore viene riavviato. L'uscita A 4.0 è "1" se il tempo è trascorso. Se lo stato di segnale all'ingresso E 0.1 è "1", il temporizzatore T5 viene resettato, ovvero viene arrestato, e il valore di tempo attuale viene impostato a "0".

13.12 ---(SA) Avvia bobina temporizzatore come ritardo alla disinserzione (SA)

Simbolo

Inglese	Tedesco
<T-Nr.>	<T-Nr.>
---(SF)	**---(SA)**
<Valore di tempo>	<Valore di tempo>

Parametro	Tipo di dati	Area di memoria	Descrizione
<Nr. T>	TIMER	T	Numero di identificazione del temporizzatore; il campo dipende dalla CPU
<Valore di tempo>	S5TIME	E, A, M, L, D	Valore di tempo di default

Descrizione dell'operazione

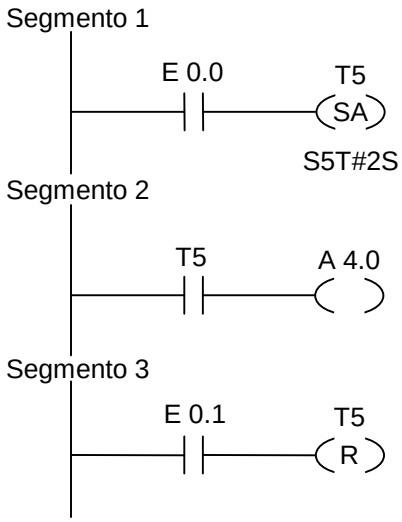
---(SA) (Avvia bobina temporizzatore come ritardo alla disinserzione) avvia il temporizzatore specificato se RLC ha un fronte di discesa. L'interrogazione di "1" dello stato di segnale produce "1" se RLC è "1" ovvero se il temporizzatore scorre con il **<valore di tempo>**. Il temporizzatore viene sempre riavviato se RLC passa da "1" a "0".

Vedere anche "Aree di memoria e componenti di un temporizzatore" e S_AVERZ Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione (SA).

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	-	-	-	-	-	0	-	-	0

Esempio



Se lo stato di segnale di E 0.0 passa da "1" a "0", il temporizzatore T5 viene avviato.

L'uscita 4.0 è "1" se E 0.0 è "1", o se scorre il temporizzatore. Se lo stato di segnale all'ingresso E 0.1 passa da "0" a "1", il temporizzatore T5 viene resettato, ovvero viene arrestato, e il valore di tempo attuale viene impostato a "0".

14 Operazioni logiche a parola

14.1 Sommario delle operazioni logiche combinatorie a parola

Descrizione

Le operazioni logiche combinatorie a parola confrontano coppie di parole (a 16 bit), e doppie parole (a 32 bit), bit per bit, in conformità alla logica booleana. Le operazioni vengono attivati dallo stato di segnale "1" all'ingresso di abilitazione (EN).

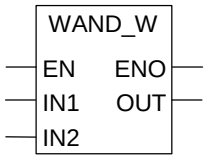
Se il risultato all'uscita OUT è diverso da 0, il bit A1 della parola di stato viene impostato a "1".
Se il risultato all'uscita OUT è uguale a 0, il bit A1 della parola di stato viene impostato a "0".

Per eseguire combinazioni logiche a parola sono disponibili le seguenti operazioni:

- WAND_W Combinazione AND parola
- WOR_W Combinazione OR parola
- WXOR_W Combinazione OR esclusivo parola
- WAND_DW Combinazione AND doppia parola
- WOR_DW Combinazione AND doppia parola
- WXOR_DW Combinazione OR esclusivo doppia parola

14.2 WAND_W Combinazione AND parola

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	WORD	E, A, M, L, D	Primo valore per l'operazione logica
IN2	WORD	E, A, M, L, D	Secondo valore per l'operazione logica
O	WORD	E, A, M, L, D	Risultato dell'operazione logica

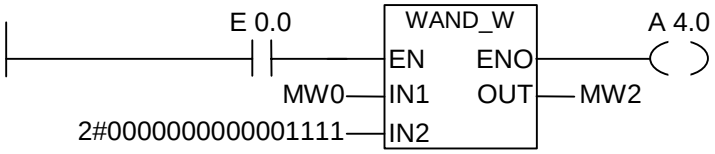
Descrizione dell'operazione

WAND_W (Combinazione AND parola) viene attivata dallo stato di segnale "1" all'ingresso di abilitazione (EN), e combina i due valori di parola IN1 e IN2 bit per bit tramite AND. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita O. ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	x	0	0	-	x	1	1	1

Esempio



L'operazione viene eseguita se E 0.0 = 1. Solo i bit da 0 a 3 di MW0 sono rilevanti; tutti gli altri bit vengono mascherati dal pattern di bit della parola in IN2:

MW0 = 01010101 01010101

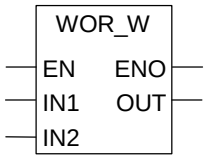
IN2 = 00000000 00001111

MW0 AND IN2 = MW2 = 00000000 00000101

Lo stato di segnale dell'uscita A 4.0 è "1", se l'operazione viene eseguita.

14.3 WOR_W Combinazione OR parola

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	WORD	E, A, M, L, D	Primo valore per l'operazione logica
IN2	WORD	E, A, M, L, D	Secondo valore per l'operazione logica
O	WORD	E, A, M, L, D	Parola risultato dell'operazione logica

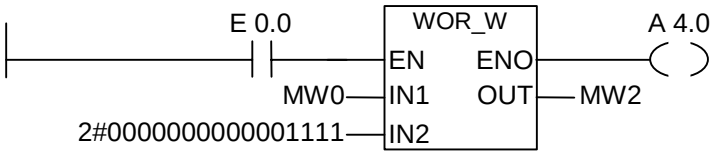
Descrizione dell'operazione

WOR_W (Combinazione OR parola) viene attivata dallo stato di segnale "1" all'ingresso di abilitazione (EN), e combina i due valori di parola IN1 e IN2 bit per bit tramite OR. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita O. ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	x	0	0	-	x	1	1	1

Esempio



L'operazione viene eseguita se E 0.0 = 1. I bit da 0 a 3 vengono impostati a "1"; tutti gli altri bit di MW0 non vengono modificati.

MW0 = 01010101 01010101

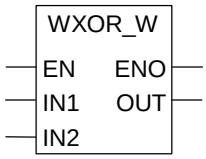
IN2 = 00000000 00001111

MW0 OR IN2 = MW2 = 01010101 01011111

Lo stato di segnale dell'uscita A 4.0 è "1" se l'operazione viene eseguita.

14.4 WXOR_W Combinazione OR esclusivo parola

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	WORD	E, A, M, L, D	Primo valore per l'operazione logica
IN2	WORD	E, A, M, L, D	Secondo valore per l'operazione logica
O	WORD	E, A, M, L, D	Parola risultato dell'operazione logica

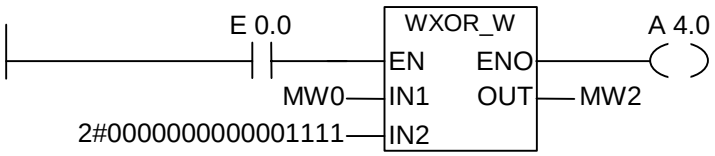
Descrizione dell'operazione

WXOR_W Combinazione OR esclusivo parola) viene attivata dallo stato di segnale "1" all'ingresso di abilitazione (EN), e combina i due valori di parola IN1 e IN2, bit per bit, tramite OR esclusivo. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita O. ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	x	0	0	-	x	1	1	1

Esempio



L'operazione viene eseguita se E 0.0 = 1:

MW0 = 01010101 01010101

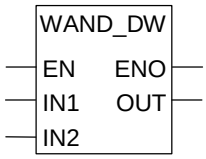
IN2 = 00000000 00001111

MW0 XOR IN2 = MW2 = 01010101 01011010

Lo stato di segnale dell'uscita A 4.0 è "1" se l'operazione viene eseguita.

14.5 WAND_DW Combinazione AND doppia parola

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DWORD	E, A, M, L, D	Primo valore per l'operazione logica
IN2	DWORD	E, A, M, L, D	Secondo valore per l'operazione logica
O	DWORD	E, A, M, L, D	Risultato di doppia parola dell'operazione logica

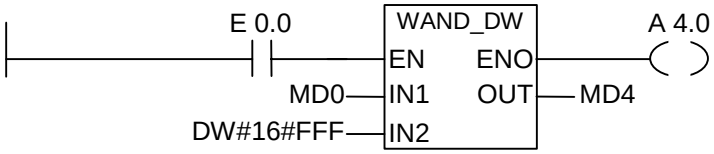
Descrizione dell'operazione

WAND_DW (Combinazione AND doppia parola) viene attivata dallo stato di segnale "1" all'ingresso di abilitazione (EN), e combina i due valori di parola IN1 e IN2 bit per bit tramite AND. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita O. ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	x	0	0	-	x	1	1	1

Esempio



L'operazione viene eseguita se E 0.0 = 1. Solo i bit 0 e 11 di MDO sono rilevanti; tutti gli altri bit vengono mascherati dal pattern di bit di IN2:

MD0 = 01010101 01010101 01010101 01010101

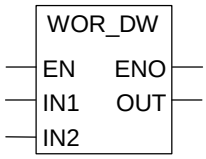
IN2 = 00000000 00000000 00001111 11111111

MD0 AND IN2 = MD4 = 00000000 00000000 00000101 01010101

Lo stato di segnale dell'uscita A 4.0 è "1" se l'operazione viene eseguita.

14.6 WOR_DW Combinazione OR doppia parola

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DWORD	E, A, M, L, D	Primo valore per l'operazione logica
IN2	DWORD	E, A, M, L, D	Secondo valore per l'operazione logica
O	DWORD	E, A, M, L, D	Doppia parola risultato dell'operazione logica

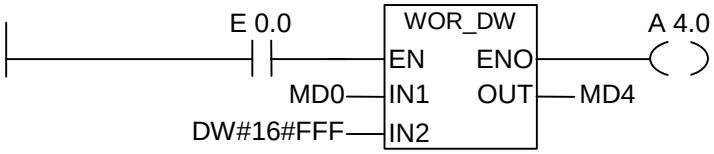
Descrizione dell'operazione

WOR_DW (Combinazione OR doppia parola) viene attivata dallo stato di segnale "1" all'ingresso di abilitazione (EN), e combina i due valori di parola IN1 e IN2, bit per bit, tramite OR. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita O. ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	x	0	0	-	x	1	1	1

Esempio



L'operazione viene eseguita se E 0.0 = 1. I bit da 0 a 11 vengono settati a "1". I rimanenti bit di MWZ non vengono modificati:

MD0 = 01010101 01010101 01010101 01010101

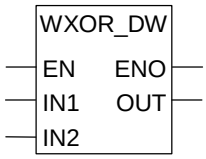
IN2 = 00000000 00000000 00001111 11111111

MD0 OR IN2 = MD4 = 01010101 01010101 01011111 11111111

Lo stato di segnale dell'uscita A 4.0 è "1" se l'operazione viene eseguita.

14.7 WXOR_DW Combinazione OR esclusivo doppia parola

Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D	Ingresso di abilitazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione
IN1	DWORD	E, A, M, L, D	Primo valore per l'operazione logica
IN2	DWORD	E, A, M, L, D	Secondo valore per l'operazione logica
O	DWORD	E, A, M, L, D	Doppia parola risultato dell'operazione logica

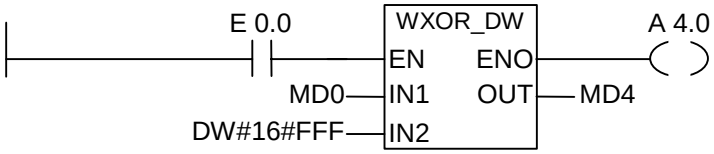
Descrizione dell'operazione

WXOR_DW Combinazione OR esclusivo doppia parola) viene attivata dallo stato di segnale "1" all'ingresso di abilitazione (EN), e combina i due valori di parola IN1 e IN2, bit per bit, tramite OR esclusivo. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita O. ENO ha lo stesso stato di segnale di EN.

Parola di stato

	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Scrive:	1	x	0	0	-	x	1	1	1

Esempio



L'operazione viene eseguita se E 0.0 = 1:

MD0 = 01010101 01010101 01010101 01010101

IN2 = 00000000 00000000 00001111 11111111

MD4 = MD0 XOR IN2 = 01010101 01010101 01011010 10101010

A 4.0 è "1", se l'operazione viene eseguita.

A Sommario di tutte le operazioni KOP

A.1 Operazioni KOP ordinate secondo il set mnemonico tedesco (SIMATIC)

Mnemonico tedesco	Mnemonico inglese	Catalogo elementi del programma	Descrizione
---| |---	---| |---	Operazione logica combinatoria a bit	Contatto normalmente aperto
---|/|---	---|/|---	Operazione logica combinatoria a bit	Contatto normalmente chiuso
---( )	---( )	Operazione logica combinatoria a bit	Bobina di uscita
---(#)---	---(#)---	Operazione logica combinatoria a bit	Connettore
==0 ---| |---	==0 ---| |---	Bit di stato	Bit di risultato uguale a 0
>0 ---| |---	>0 ---| |---	Bit di stato	Bit di risultato maggiore di 0
>=0 ---| |---	>=0 ---| |---	Bit di stato	Bit di risultato maggiore o uguale a 0
<=0 ---| |---	<=0 ---| |---	Bit di stato	Bit di risultato minore o uguale a 0
<0 ---| |---	<0 ---| |---	Bit di stato	Bit di risultato minore di 0
<>0 ---| |---	<>0 ---| |---	Bit di stato	Bit di risultato diverso da 0
ABS	ABS	Funzione in virgola mobile	Formazione del valore assoluto di un numero in virgola mobile
ACOS	ACOS	Funzione in virgola mobile	Formazione dell'arcoseno di un numero in virgola mobile (a 32 bit)
ADD_DI	ADD_DI	Funzione in virgola fissa	Somma numeri interi a 32 bit
ADD_I	ADD_I	Funzione in virgola fissa	Somma numeri interi a 16 bit
ADD_R	ADD_R	Funzione in virgola mobile	Somma numeri in virgola mobile
ASIN	ASIN	Funzione in virgola mobile	Formazione dell'arcoseno di un numero in virgola mobile (a 32 bit)
ATAN	ATAN	Funzione in virgola mobile	Formazione dell'arcotangente di un numero in virgola mobile (a 32 bit)
---(OPN)	---(OPN)	Richiamo DB	Apri un blocco dati
BCD_DI	BCD_DI	Convertitori	Converti un numero BCD in un numero intero (a 32 bit)
BCD_I	BCD_I	Convertitori	Converti un numero BCD in un numero intero (a 16 bit)
BIE ---| |---	BR ---| |---	Bit di stato	Bit di anomalia: registro BIE
----(CALL)	----(CALL)	Comando del programma	Richiama FC/SFC dalla bobina (senza parametri)

Mnemonico tedesco	**Mnemonico inglese**	**Catalogo elementi del programma**	**Descrizione**
CALL_FB	CALL_FB	Comando del programma	Richiama FB dal box
CALL_FC	CALL_FC	Comando del programma	Richiama FC dal box
CALL_SFB	CALL_SFB	Comando del programma	Richiama SFB dal box
CALL_SFC	CALL_SFC	Comando del programma	Richiama SFC dal box
CEIL	CEIL	Convertitori	Genera da un numero in virgola mobile un numero intero superiore
CMP ? D	CMP ? D	Comparatori	Confronta numeri interi a 32 bit
CMP ? I	CMP ? I	Comparatori	Confronta numeri interi a 16 bit
CMP ? R	CMP ? R	Comparatori	Confronta numeri in virgola mobile
COS	COS	Funzione in virgola mobile	Formazione del coseno di un angolo come numero in virgola mobile (a 32 bit)
DI_BCD	DI_BCD	Convertitori	Converti un numero intero (a 32 bit) in un numero BCD
DI_R	DI_R	Convertitori	Converti un numero intero (a 32 bit) in un numero in virgola mobile
DIV_DI	DIV_DI	Funzione in virgola fissa	Dividi numeri interi a32 bit
DIV_I	DIV_I	Funzione in virgola fissa	Dividi numeri interi a 16 bit
DIV_R	DIV_R	Funzione in virgola mobile	Dividi numeri in virgola mobile
EXP	EXP	Funzione in virgola mobile	Formazione del valore esponenziale di un numero in virgola mobile (a 32 bit)
FLOOR	FLOOR	Convertitori	Genera da un numero in virgola mobile un numero intero inferiore
I_BCD	I_BCD	Convertitori	Converti un numero intero (a 16 bit) in un numero BCD
I_DI	I_DI	Convertitori	Converti un numero intero (a 16 bit) in un numero intero (a 32 bit)
INV_I	INV_I	Convertitori	Complemento a uno di numero intero (a 16 bit)
INV_DI	INV_DI	Convertitori	Complemento a uno di numero intero (a 32 bit)
---(JMP)	---(JMP)	Salti	Salto assoluto
---(JMP)	---(JMP)	Salti	Se 1 salta in un FB (condizionato)
---(JMPN)	---(JMPN)	Salti	Se 0 salta in un FB (condizionato)
LABEL	LABEL	Salti	Etichetta di salto
LN	LN	Funzione in virgola mobile	Formazione del logaritmo naturale di un numeor in virgola mobile (a 32 bit)
---(MCR>)	---(MCR>)	Comando del programma	Disattiva zona relè master control
---(MCR<)	---(MCR<)	Comando del programma	Attiva zona relè master control
---(MCRA)	---(MCRA)	Comando del programma	Inizio zona relè master control
---(MCRD)	---(MCRD)	Comando del programma	Fine zona relè master control
MOD_DI	MOD_DI	Funzione in virgola fissa	Ricava resto della divisione (a 32 bit)
MOVE	MOVE	Spostamento	Assegna valore
MUL_DI	MUL_DI	Funzione in virgola fissa	Moltiplica numeri interi a 32 bit
MUL_I	MUL_I	Funzione in virgola fissa	Moltiplica numeri interi a 16 bit
MUL_R	MUL_R	Funzione in virgola mobile	Moltiplica numeri in virgola mobile

Mnemonico tedesco	**Mnemonico inglese**	**Catalogo elementi del programma**	**Descrizione**
---(N)---	---(N)---	Operazione logica combinatoria a bit	Rileva fronte RLC di discesa
NEG	NEG	Operazione logica combinatoria a bit	Interroga rilevamento di fronte di discesa
NEG_DI	NEG_DI	Convertitori	Complemento a due di numero intero (a 32 bit)
NEG_I	NEG_I	Convertitori	Complemento a due di numero intero (a 16 bit)
NEG_R	NEG_R	Convertitori	Complemento a due di numero in virgola mobile
---| NOT |---	---| NOT |---	Operazione logica combinatoria a bit	Inverti il risultato della combinazione
OS ---| |---	OS ---| |---	Bit di stato	Bit di anomalia: overflow con memoria
OV ---| |---	OV ---| |---	Bit di stato	Bit di anomalia: overflow
---(P)---	---(P)---	Operazione logica combinatoria a bit	Rileva fronte RLC di salita
POS	POS	Operazione logica combinatoria a bit	Interroga rilevamento di fronte di salita
---(R)	---(R)	Operazione logica combinatoria a bit	Resetta bobina
---(RET)	---(RET)	Comando del programma	Salta indietro
ROL_DW	ROL_DW	Scorrimento/rotazione	Fai ruotare doppia parola a sinistra
ROR_DW	ROR_DW	Scorrimento/rotazione	Fai ruotare doppia parola a destra
ROUND	ROUND	Convertitori	Arrotonda a numero intero (a 32 bit)
RS	RS	Operazione logica combinatoria a bit	Resetta imposta flip flop
---(S)	---(S)	Operazione logica combinatoria a bit	Imposta bobina
---(SA)	---(SF)	Temporizzatori	Avvia bobina temporizzatore come ritardo alla disinserzione (SA)
---(SAVE)	---(SAVE)	Operazione logica combinatoria a bit	Salva RLC nel registro BIE
S_AVERZ	S_OFFDT	Temporizzatori	Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione (SA)
---(SE)	---(SD)	Temporizzatori	Avvia bobina temporizzatore come ritardo all'inserzione (SE)
S_EVERZ	S_ODT	Temporizzatori	Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione (SE)
SHL_DW	SHL_DW	Scorrimento/rotazione	Fai scorrere doppia parola verso sinistra
SHL_W	SHL_W	Scorrimento/rotazione	Fai scorrere parola verso sinistra
SHR_DI	SHR_DI	Scorrimento/rotazione	Fai scorrere numero intero a 32 bit verso destra
SHR_DW	SHR_DW	Scorrimento/rotazione	Fai scorrere doppia parola verso destra
SHR_I	SHR_I	Scorrimento/rotazione	Fai scorrere numero intero a 16 bit verso destra
SHR_W	SHR_W	Scorrimento/rotazione	Fai scorrere parola verso destra
---(SI)	---(SP)	Temporizzatori	Avvia bobina temporizzatore come impulso (SI)
S_IMPULS	S_PULSE	Temporizzatori	Parametrizza e avvia bobina temporizzatore S5 come impulso (SI)

Mnemonico tedesco	**Mnemonico inglese**	**Catalogo elementi del programma**	**Descrizione**
SIN	SIN	Funzione in virgola mobile	Formazione del seno di un angolo come virgola mobile (a 32 bit)
SQR	SQR	Funzione in virgola mobile	Formazione del quadrato di un numero in virgola mobile (a 32 bit)
SQRT	SQRT	Funzione in virgola mobile	Formazione della radice quadrata di un numero in virgola mobile (a 32 bit)
SR	SR	Operazione logica combinatoria a bit	Imposta resetta flip flop
---(SS)	---(SS)	Temporizzatori	Avvia bobina temporizzatore come ritardo all'inserzione con memoria (SS)
S_SEVERZ	S_ODTS	Temporizzatori	Parametrizza e avvia bobina temporizzatore S5 come ritardo all'inserzione con memoria (SS)
SUB_DI	SUB_DI	Funzione in virgola fissa	Sottrai numeri interi a 32 bit
SUB_I	SUB_I	Funzione in virgola fissa	Sottrai numeri interi a 16 bit
SUB_R	SUB_R	Funzione in virgola mobile	Sottrai numeri in virgola mobile
---(SV)	---(SE)	Temporizzatori	Avvia bobina temporizzatore come impulso prolungato (SV)
S_VIMP	S_PEXT	Temporizzatori	Parametrizza e avvia bobina temporizzatore S5 come impulso prolungato (SV)
---(SZ)	---(SC)	Contatori	Imposta valore di conteggio
TAN	TAN	Funzione in virgola mobile	Formazione della tangente di un angolo come numero in virgola mobile (a 32 bit)
TRUNC	TRUNC	Convertitori	Arrotonda senza resto un numero intero (a 32 bit)
UO ---| |---	UO ---| |---	Bit di stato	Bit di anomalia: operazione non ammessa
WAND_DW	WAND_DW	Operazione logica combinatoria a parola	Combinazione AND doppia parola
WAND_W	WAND_W	Operazione logica combinatoria a parola	Combinazione AND parola
WOR_DW	WOR_DW	Operazione logica combinatoria a parola	Combinazione OR doppia parola
WOR_W	WOR_W	Operazione logica combinatoria a parola	Combinazione OR parola
WXOR_DW	WXOR_DW	Operazione logica combinatoria a parola	Combinazione OR esclusivo doppia parola
WXOR_W	WXOR_W	Operazione logica combinatoria a parola	Combinazione OR esclusivo parola
ZAEHLER	S_CUD	Contatori	Parametrizza e conta in avanti/all'indietro
----(ZR)	----(CD)	Contatori	Conta all'indietro
Z_RUECK	----(S_CD)	Contatori	Parametrizza e conta all'indietro
---(ZV)	----(CU)	Contatori	Conta in avanti
Z_VORW	S_CU	Contatori	Parametrizza e conta in avanti

A.2 Operazioni KOP ordinate secondo il set mnemonico inglese (internazionale)

Mnemonico inglese	Mnemonico tedesco	Catalogo elementi del programma	Descrizione
---|/|---	---|/|---	Operazione logica combinatoria a bit	Contatto normalmente aperto
---| |---	---| |---	Operazione logica combinatoria a bit	Contatto normalmente chiuso
---( )	---( )	Operazione logica combinatoria a bit	Bobina di uscita
---(#)---	---(#)---	Operazione logica combinatoria a bit	Connettore
==0 ---| |---	==0 ---| |---	Bit di stato	Bit di risultato uguale a 0
>0 ---| |---	>0 ---| |---	Bit di stato	Bit di risultato maggiore di 0
>=0 ---| |---	>=0 ---| |---	Bit di stato	Bit di risultato maggiore o uguale a 0
<=0 ---| |---	<=0 ---| |---	Bit di stato	Bit di risultato minore o uguale a 0
<0 ---| |---	<0 ---| |---	Bit di stato	Bit di risultato minore di 0
<>0 ---| |---	<>0 ---| |---	Bit di stato	Bit di risultato diverso da 0
ABS	ABS	Funzione in virgola mobile	Formazione del valore assoluto di un numero in virgola mobile
ACOS	ACOS	Funzione in virgola mobile	Formazione dell'arcoseno di un numero in virgola mobile (a 32 bit)
ADD_DI	ADD_DI	Funzione in virgola fissa	Somma numeri interi a 32 bit
ADD_I	ADD_I	Funzione in virgola fissa	Somma numeri interi a 16 bit
ADD_R	ADD_R	Funzione in virgola mobile	Somma numeri in virgola mobile
ASIN	ASIN	Funzione in virgola mobile	Formazione dell'arcoseno di un numero in virgola mobile (a 32 bit)
ATAN	ATAN	Funzione in virgola mobile	Formazione dell'arcotangente di un numero in virgola mobile (a 32 bit)
BCD_DI	BCD_DI	Convertitori	Converti un numero BCD in un numero intero (a 32 bit)
BCD_I	BCD_I	Convertitori	Converti un numero BCD in un numero intero (a 16 bit)
BR ---| |---	BIE ---| |---	Bit di stato	Bit di anomalia: registro BIE
----(CALL)	----(CALL)	Comando del programma	Richiama FC/SFC dalla bobina (senza parametri)
CALL_FB	CALL_FB	Comando del programma	Richiama FB dal box
CALL_FC	CALL_FC	Comando del programma	Richiama FC dal box
CALL_SFB	CALL_SFB	Comando del programma	Richiama SFB dal box
CALL_SFC	CALL_SFC	Comando del programma	Richiama SFC dal box
----(CD)	----(ZR)	Contatori	Conta all'indietro
CEIL	CEIL	Convertitori	Genera da un numero in virgola mobile un numero intero superiore
CMP ? D	CMP ? D	Comparatori	Confronta numeri interi a 32 bit

Mnemonico inglese	**Mnemonico tedesco**	**Catalogo elementi del programma**	**Descrizione**
CMP ? I	CMP ? I	Comparatori	Confronta numeri interi a 16 bit
CMP ? R	CMP ? R	Comparatori	Confronta numeri in virgola mobile
COS	COS	Funzione in virgola mobile	Formazione del coseno di un angolo come numero in virgola mobile (a 32 bit)
----(CU)	---(ZV)	Contatori	Conta in avanti
DI_BCD	DI_BCD	Convertitori	Converti un numero intero (a 32 bit) in un numeroBCD
DI_R	DI_R	Convertitori	Converti un numero intero (a 32 bit) in un numero in virgola mobile
DIV_DI	DIV_DI	Funzione in virgola fissa	Dividi numeri interi a32 bit
DIV_I	DIV_I	Funzione in virgola fissa	Dividi numeri interi a 16 bit
DIV_R	DIV_R	Funzione in virgola mobile	Dividi numeri in virgola mobile
EXP	EXP	Funzione in virgola mobile	Formazione del valore esponenziale di un numero in virgola mobile (a 32 bit)
FLOOR	FLOOR	Convertitori	Genera da un numero in virgola mobile un numero intero inferiore
I_BCD	I_BCD	Convertitori	Converti un numero intero (a 16 bit) in un numero BCD
I_DI	I_DI	Convertitori	Converti un numero intero (a 16 bit) in un numero intero (a 32 bit)
INV_I	INV_I	Convertitori	Complemento a uno di numero intero (a 16 bit)
INV_DI	INV_DI	Convertitori	Complemento a uno di numero intero (a 32 bit)
---(JMP)	---(JMP)	Salti	Salto assoluto
---(JMP)	---(JMP)	Salti	Se 1 salta in un FB (condizionato)
---(JMPN)	---(JMPN)	Salti	Se 0 salta in un FB (condizionato)
LABEL	LABEL	Salti	Etichetta di salto
LN	LN	Funzione in virgola mobile	Formazione del logaritmo naturale di un numero in virgola mobile (a 32 bit)
---(MCR>)	---(MCR>)	Comando del programma	Disattiva zona relè master control
---(MCR<)	---(MCR<)	Comando del programma	Attiva zona relè master control
---(MCRA)	---(MCRA)	Comando del programma	Inizio zona relè master control
---(MCRD)	---(MCRD)	Comando del programma	Fine zona relè master control
MOD_DI	MOD_DI	Funzione in virgola fissa	Ricava resto della divisione (a 32 bit
MOVE	MOVE	Spostamento	Assegna valore
MUL_DI	MUL_DI	Funzione in virgola fissa	Moltiplica numeri interi a 32 bit
MUL_I	MUL_I	Funzione in virgola fissa	Moltiplica numeri interi a 16 bit
MUL_R	MUL_R	Funzione in virgola mobile	Moltiplica numeri in virgola mobile
---(N)---	---(N)---	Operazione logica combinatoria a bit	Rileva fronte RLC di discesa
NEG	NEG	Operazione logica combinatoria a bit	Interroga rilevamento di fronte di discesa
NEG_DI	NEG_DI	Convertitori	Complemento a due di numero intero (a 32 bit)
NEG_I	NEG_I	Convertitori	Complemento a due di numero intero (a 16 bit)
NEG_R	NEG_R	Convertitori	Complemento a due di numero in virgola mobile

Mnemonico inglese	**Mnemonico tedesco**	**Catalogo elementi del programma**	**Descrizione**
---| NOT |---	---| NOT |---	Operazione logica combinatoria a bit	Inverti il risultato della combinazione
---(OPN)	---(OPN)	Richiamo DB	Apri blocco dati
OS ---| |---	OS ---| |---	Bit di stato	Bit di anomalia: overflow con memoria
OV ---| |---	OV ---| |---	Bit di stato	Bit di anomalia: overflow
---(P)---	---(P)---	Operazione logica combinatoria a bit	Rileva fronte RLC di salita
POS	POS	Operazione logica combinatoria a bit	Interroga rilevamento di fronte di salita
---(R)	---(R)	Operazione logica combinatoria a bit	Resetta bobina
---(RET)	---(RET)	Comando del programma	Salta indietro
ROL_DW	ROL_DW	Scorrimento/rotazione	Fai ruotare doppia parola a sinistra
ROR_DW	ROR_DW	Scorrimento/rotazione	Fai ruotare doppia parola a destra
ROUND	ROUND	Convertitori	Arrotonda a numero intero (a 32 bit)
RS	RS	Operazione logica combinatoria a bit	Resetta imposta flip flop
---(S)	---(S)	Operazione logica combinatoria a bit	Imposta bobina
---(SAVE)	---(SAVE)	Operazione logica combinatoria a bit	Salva RLC nel registro BIE
---(SC)	---(SZ)	Contatori	Imposta valore di conteggio
----(S_CD)	Z_RUECK	Contatori	Parametrizza e conta all'indietro
S_CU	Z_VORW	Contatori	Parametrizza e conta in avanti
S_CUD	ZAEHLER	Contatori	Parametrizza e conta in avanti/all'indietro
---(SD)	---(SE)	Temporizzatori	Avvia bobina temporizzatore come ritardo all'inserzione
---(SE)	---(SV)	Temporizzatori	Avvia bobina temporizzatore come impulso prolungato
---(SF)	---(SA)	Temporizzatori	Avvia bobina temporizzatore come ritardo alla disinserzione
SHL_DW	SHL_DW	Scorrimento/rotazione	Fai scorrere doppia parola verso sinistra
SHL_W	SHL_W	Scorrimento/rotazione	Fai scorrere parola verso sinistra
SHR_DI	SHR_DI	Scorrimento/rotazione	Fai scorrere numero intero a 32 bit verso destra
SHR_DW	SHR_DW	Scorrimento/rotazione	Fai scorrere doppia parola verso destra
SHR_I	SHR_I	Scorrimento/rotazione	Fai scorrere numero intero a 16 bit verso destra
SHR_W	SHR_W	Scorrimento/rotazione	Fai scorrere parola verso destra
SIN	SIN	Funzione in virgola mobile	Formazione del seno di un angolo come virgola mobile
S_ODT	S_EVERZ	Temporizzatori	Parametrizza e avvia bobina temporizzatore S5 come ritardo all'inserzione (SE)
S_ODTS	S_SEVERZ	Temporizzatori	Parametrizza e avvia bobina temporizzatore S5 come ritardo all'inserzione con memoria (SS)
S_OFFDT	S_AVERZ	Temporizzatori	Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione (SA)

Mnemonico inglese	**Mnemonico tedesco**	**Catalogo elementi del programma**	**Descrizione**
---(SP)	---(SI)	Temporizzatori	Avvia bobina temporizzatore come impulso (SI)
S_PEXT	S_VIMP	Temporizzatori	Parametrizza e avvia bobina temporizzatore S5 come impulso prolungato (SV)
S_PULSE	S_IMPULS	Temporizzatori	Parametrizza e avvia bobina temporizzatore S5 come impulso (SI)
SQR	SQR	Funzione in virgola mobile	Formazione del quadrato di un numero in virgola mobile
SQRT	SQRT	Funzione in virgola mobile	Formazione della radice quadrata di un numero in virgola mobile
SR	SR	Operazione logica combinatoria a bit	Resetta imposta flip flop
---(SS)	---(SS)	Temporizzatori	Avvia bobina temporizzatore come ritardo all'inserzione con memoria (SS)
SUB_DI	SUB_DI	Funzione in virgola fissa	Sottrai numeri interi a 32 bit
SUB_I	SUB_I	Funzione in virgola fissa	Sottrai numeri interi a 16 bit
SUB_R	SUB_R	Funzione in virgola mobile	Sottrai numeri in virgola mobile
TAN	TAN	Funzione in virgola mobile	Formazione della tangente di un angolo come numero in virgola mobile (a 32 bit)
TRUNC	TRUNC	Convertitori	Arrotonda senza resto un numero intero (a 32 bit)
UO ---| |---	UO ---| |---	Bit di stato	Bit di anomalia: operazione non ammessa
WAND_DW	WAND_DW	Operazione logica combinatoria a parola	Combinazione AND doppia parola
WAND_W	WAND_W	Operazione logica combinatoria a parola	Combinazione AND parola
WOR_DW	WOR_DW	Operazione logica combinatoria a parola	Combinazione OR doppia parola
WOR_W	WOR_W	Operazione logica combinatoria a parola	Combinazione OR parola
WXOR_DW	WXOR_DW	Operazione logica combinatoria a parola	Combinazione OR esclusivo doppia parola
WXOR_W	WXOR_W	Operazione logica combinatoria a parola	Combinazione OR esclusivo parola

B Esempi di programmazione

B.1 Sommario

Applicazione pratiche

Tutte le operazioni KOP inizializzano un'operazione specifica. Combinando queste operazioni in un programma, è possibile eseguire numerose e diversificate operazioni di automazione. Questa appendice contiene i seguenti esempi di applicazioni pratiche delle istruzioni KOP:

- Controllo di un nastro trasportatore utilizzando le operazioni logiche combinatorie a bit
- Rilevazione della direzione di movimento di un nastro trasportatore utilizzando le operazioni logiche combinatorie a bit
- Generazione di un impulso di clock utilizzando le operazioni di temporizzazione
- Registrazione dello spazio di memoria avvalendosi delle operazioni di conteggio e confronto
- Soluzione di un problema utilizzando le operazioni matematiche con numeri interi
- Impostazione della durata di riscaldamento di un forno

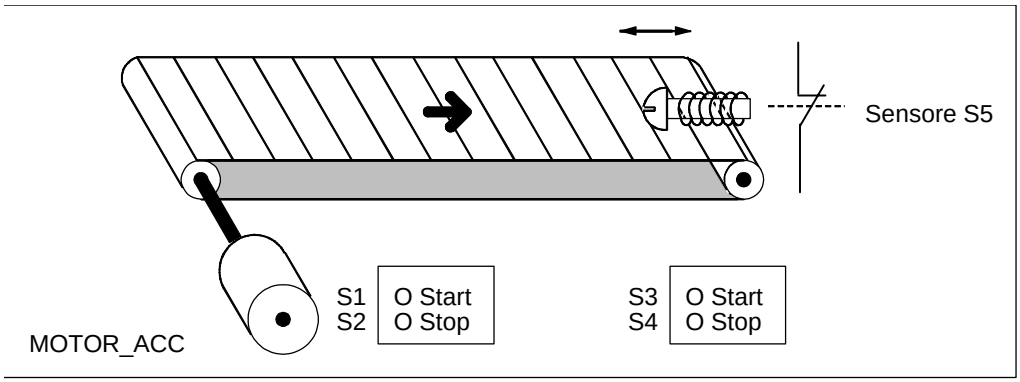
Operazioni utilizzate

Mnemonico	Operazione	Descrizione
WAND_W	Operazione logica combinatoria a parola	Combinazione AND parola
WOR_W	Operazione logica combinatoria a parola	Combinazione OR parola
Z_RUECK	Contatori	Conta all'indietro
Z_VORW	Contatori	Conta in avanti
---(R)	Operazione logica combinatoria a bit	Resetta bobina
---(S)	Operazione logica combinatoria a bit	Imposta bobina
---(P)	Operazione logica combinatoria a bit	Rileva fronte RLC di salita
ADD_I	Funzione in virgola fissa	Somma numeri interi
DIV_I	Funzione in virgola fissa	Dividi numeri interi
MUL_I	Funzione in virgola fissa	Moltiplica numeri interi
CMP >=I	Comparatori	Confronta numeri interi
CMP <=I	Comparatori	Confronta numeri interi
—| |—	Operazione logica combinatoria a bit	Contatto normalmente aperto
—| / |—	Operazione logica combinatoria a bit	Contatto normalmente chiuso
—( )	Operazione logica combinatoria a bit	Bobina di uscita
---(JMPN)	Salti	Se 0 salta in un FB (condizionato)
---(RET)	Comando del programma	Salta indietro
MOVE	Spostamento	Assegna un valore
---(SV)	Temporizzatori	Avvia temporizzatore come impulso prolungato

B.2 Esempi: Operazioni logiche combinatorie a bit

Esempio 1: Controllo di un nastro trasportatore

La figura mostra un nastro trasportatore che può essere attivato elettricamente. Alla partenza del nastro sono presenti due interruttori a pulsante: S1 per START e S2 per STOP. Anche alla fine del nastro sono presenti due interruttori: S3 per START e S4 per STOP. È quindi possibile avviare e arrestare il nastro da entrambi i suoi capi. La presenza di un sensore S5 permette di arrestare il nastro quando un elemento trasportato raggiunge il punto finale.



Programmazione con valori assoluti e a simboli

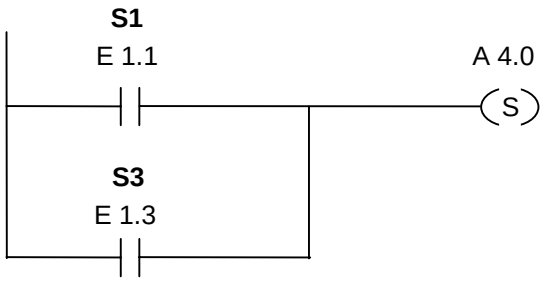
È possibile scrivere un programma destinato al controllo di un nastro trasportatore, avvalendosi di **valori assoluti** oppure di **simboli** che rappresentano i diversi componenti del sistema di trasporto.

I simboli scelti vengono correlati nella tabella dei simboli con i valori assoluti (consultare la Guida online di STEP 7).

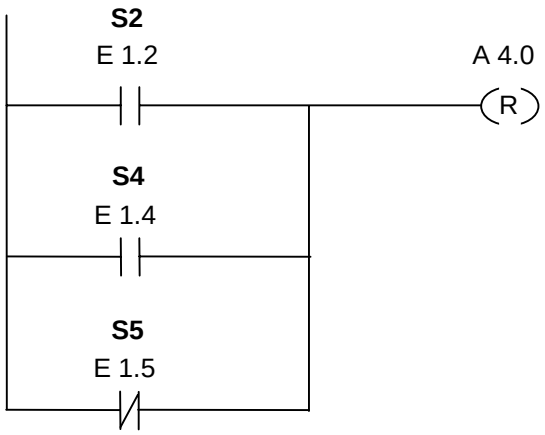
Componente del sistema	Indirizzo assoluto	Simbolo	Tabella dei simboli
Pulsante Start	E 1.1	S1	E 1.1 S1
Pulsante Stop	E 1.2	S2	E 1.2 S2
Pulsante Start	E 1.3	S3	E 1.3 S3
Pulsante Stop	E 1.4	S4	E 1.4 S4
Sensore	E 1.5	S5	E 1.5 S5
Motore	A 4.0	MOTORE_ACC	A 4.0 MOTORE_ACC

Schema a contatti KOP per il controllo del nastro trasportatore

Segmento 1: premendo uno dei due pulsanti Start si aziona il motore.

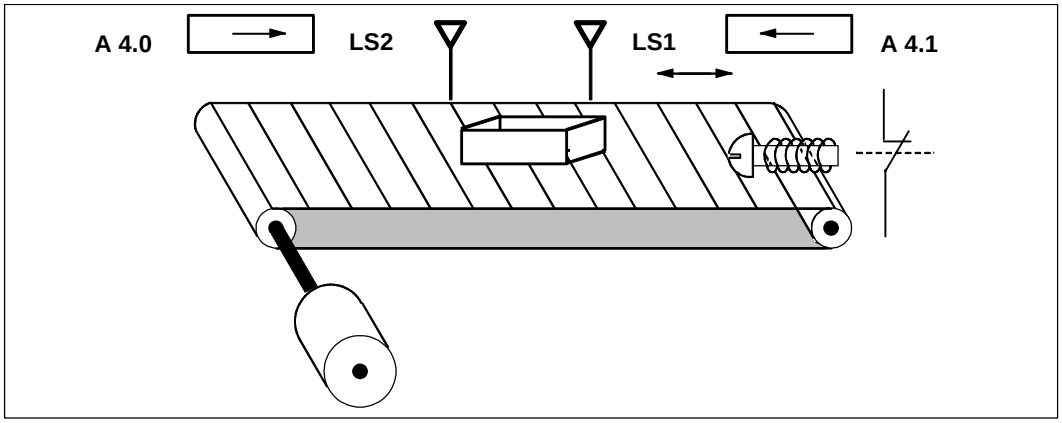


Segmento 2: premendo uno dei due pulsanti Stop o aprendo il contatto normalmente chiuso posto alla fine del nastro trasportatore si spegne il motore.



Esempio 2: Rilevamento della direzione di marcia di un nastro trasportatore

La seguenta figura mostra un nastro trasportatore che dispone di due barriere a fotocellula (LS1 e LS2) il cui scopo è quello di rilevare la direzione di movimento del materiale sul nastro. Ognuna delle due barriere a fotocellula elettrica funziona come un contatto normalmente aperto.



Programmazione con valori assoluti e a simboli

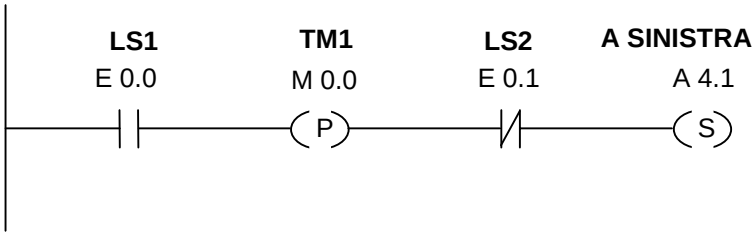
È possibile scrivere un programma destinato al controllo di un nastro trasportatore, avvalendosi di **valori assoluti** oppure di **simboli** che rappresentano i diversi componenti del sistema di trasporto.

I simboli scelti vengono correlati nella tabella dei simboli con i valori assoluti (consultare la Guida online di STEP 7).

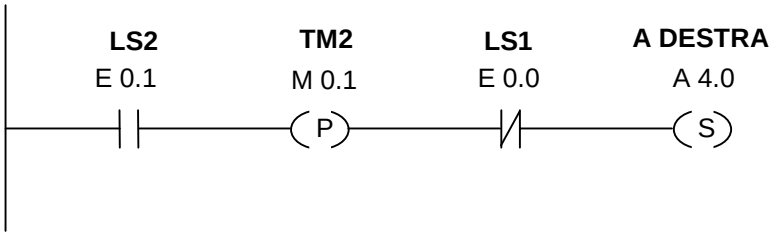
Componente del sistema	Indirizzo assoluto	Simbolo	Tabella dei simboli
Barriera a fotocellula elettrica 1	E 0.0	LS1	E 0.0 LS1
Barriera a fotocellula elettrica 2	E 0.1	LS2	E 0.1 LS2
Visualizzatore per il senso di marcia a destra	A 4.0	DESTRA	A 4.0 DESTRA
Visualizzatore per il senso di marcia a sinistra	A 4.1	SINISTRA	A 4.1 SINISTRA
Bit 1 della memoria d'impulso	M 0.0	TM1	M 0.0 TM1
Bit 2 della memoria d'impulso	M 0.1	TM2	M 0.1 TM2

Schema a contatti KOP per il rilevamento della direzione di marcia di un nastro trasportatore

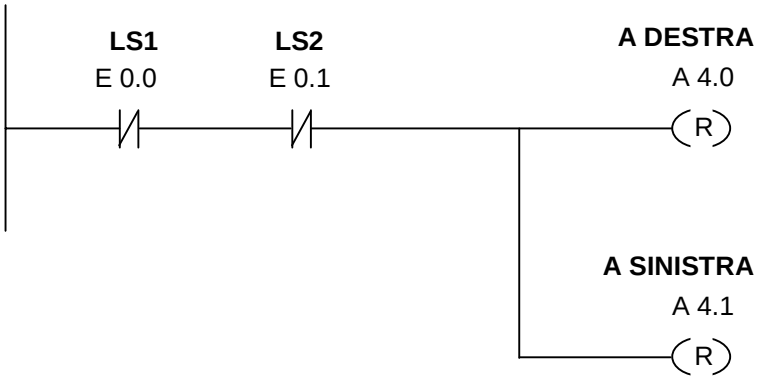
Segmento 1: se lo stato del segnale di E 0.0 passa da 0 a 1 (fronte di salita) e allo stesso tempo, lo stato di segnale di E 0.1 è 0, il pacco sul nastro si sposta verso sinistra.



Segmento 2: se lo stato del segnale di E 0.1 passa da 0 a 1 (fronte di salita)e allo stesso tempo, lo stato di segnale di E 0.0 è 0, il pacco sul nastro si sposta verso destra. Se una delle barriere a fotocellula è interrotta, vuol dire che un pacco si trova tra le fotocellule.



Segmento 3: Se una delle barriere a fotocellula è interrotta, significa che un pacco è situato tra le fotocellule. Il visualizzatore del senso di marcia viene disattivato.



B.3 Esempio: Operazioni di temporizzazione

Generatore d'impulso di clock

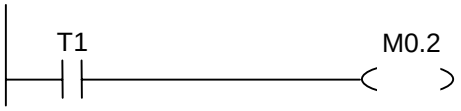
È possibile utilizzare un generatore di impulsi di clock o un relè di lampeggio per poter produrre un segnale che si ripete periodicamente. Un generatore di impulsi di clock è alquanto comune in un sistema di segnalazione che controlla il lampeggio delle spie.

Quando si utilizza S7-300, si può implementare la funzione di generazione d'impulsi avvalendosi di un'elaborazione temporizzata in blocchi di organizzazione speciali.

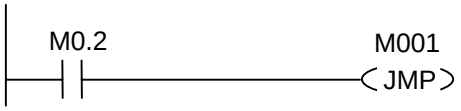
Schema a contatti KOP per l'attivazione delle spie su un pannello visualizzatore (fattore di impulso 1:1)

Segmento 1: se lo stato del segnale del temporizzatore T1 è 0, caricare il valore 250 ms in T1 e avviare T 1 come temporizzatore di impulso prolungato.

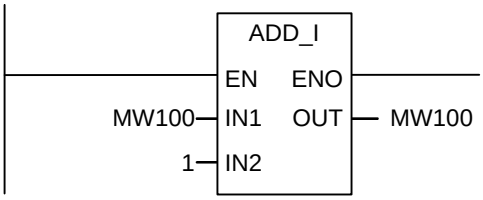
Segmento 2: lo stato del temporizzatore viene depositato in un merker di appoggio.



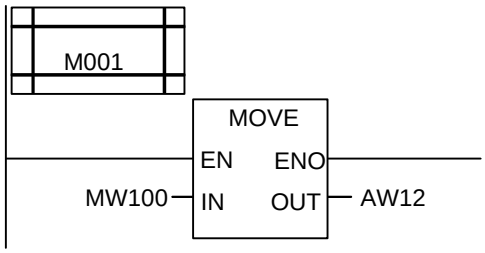
Segmento 3: se lo stato del segnale del temporizzatore T1 è 1, saltare all'etichetta M001.



Segmento 4: ogni volta che il tempo del temporizzatore T1 è trascorso, la parola di merker 100 viene incrementata di 1.

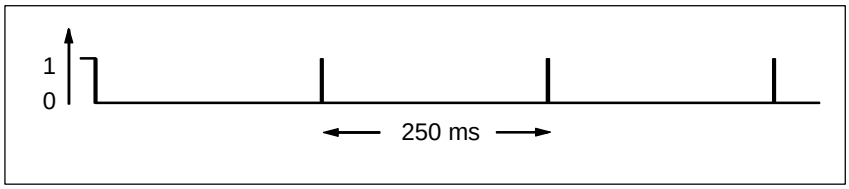


Segmento 5: l'operazione **MOVE** consente di vedere le diverse frequenze di clock sulle uscite da A 12.0 a A 13.7.



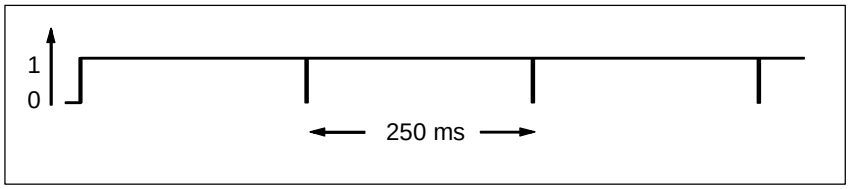
Interrogazione di segnale

L'interrogazione di segnale del temporizzatore T1 produce per il contatto normalmente chiuso M0.2 il seguente risultato combinatorio:



Appena il tempo è scaduto, il temporizzatore viene riavviato. Per questo motivo, il controllo di segnale svolto dal contatto normalmente chiuso —| / |— T1 produce uno stato del segnale uguale a 1 solo per un istante.

Il bit di negazione RLC (invertito):



Ogni 250 ms il bit di RLO è 0. Il salto viene ignorato e il contenuto della parola di merker MW100 viene incrementato di 1.

Ottenimento di una frequenza spezifica

Con i bit dei merker MB101 e MB100 è possibile ottenere le seguenti frequenze:

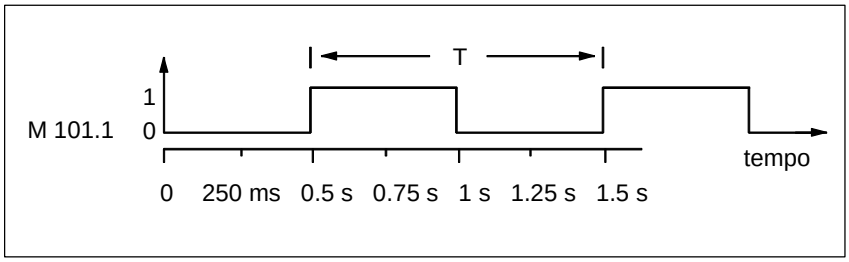
Bit di MB101, MB100	Frequenza in Hertz	Durata	
M 101.0	2.0	0.5 s	(250 ms on / 250 ms off)
M 101.1	1.0	1 s	(0.5 s on / 0.5 s off)
M 101.2	0.5	2 s	(1 s on / 1 s off)
M 101.3	0.25	4 s	(2 s on / 2 s off)
M 101.4	0.125	8 s	(4 s on / 4 s off)
M 101.5	0.0625	16 s	(8 s on / 8 s off)
M 101.6	0.03125	32 s	(16 s on / 16 s off)
M 101.7	0.015625	64 s	(32 s on / 32 s off)
M 100.0	0.0078125	128 s	(64 s on / 64 s off)
M 100.1	0.0039062	256 s	(128 s on / 128 s off)
M 100.2	0.0019531	512 s	(256 s on / 256 s off)
M 100.3	0.0009765	1024 s	(512 s on / 512 s off)
M 100.4	0.0004882	2048 s	(1024 s on / 1024 s off)
M 100.5	0.0002441	4096 s	(2048 s on / 2048 s off)
M 100.6	0.000122	8192 s	(4096 s on / 4096 s off)
M 100.7	0.000061	16384 s	(8192 s on / 8192 s off)

Stati dei segnali dei bit del byte di merker MB101

Ciclo	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valore di tempo in ms
0	0	0	0	0	0	0	**0**	0	250
1	0	0	0	0	0	0	**0**	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	**0**	0	250
5	0	0	0	0	0	1	**0**	1	250
6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	**0**	0	250
9	0	0	0	0	1	0	**0**	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	**0**	0	250

Stato del segnale del bit 1 di MB101 (M 101.1)

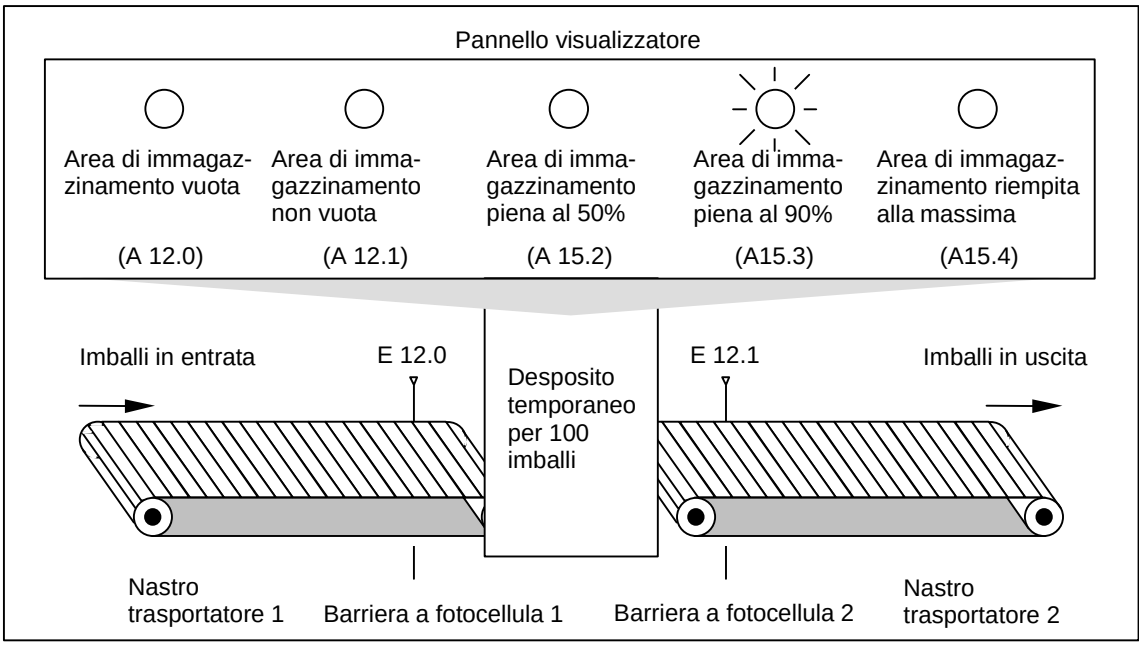
Frequenza = 1/T = 1/1 s = 1 Hz



B.4 Esempio: Operazioni di conteggio e confronto

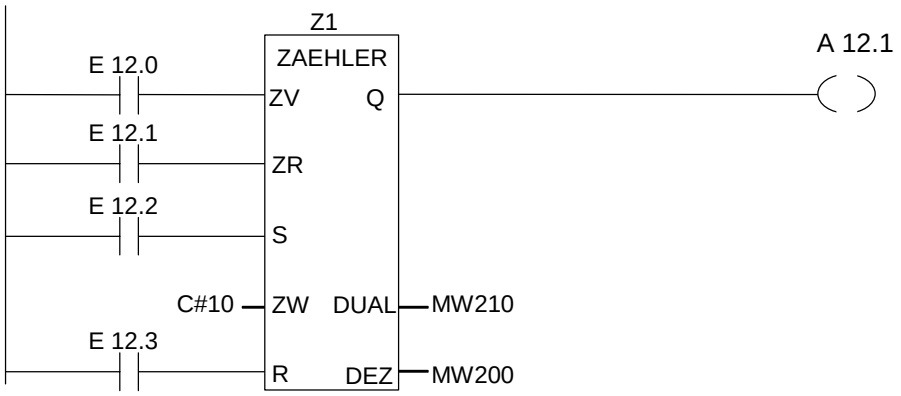
Area di immagazzinamento con Contatore e Confrontatore

La seguenta figura mostra un sistema con due nastri trasportatori e un'area di immagazzinamento temporaneo tra i due sistemi di trasporto. Il nastro trasportatore 1 invia il materiale nell'area di immagazzinamento. Una barriera a fotocellula alla fine del nastro 1 in prossimità dell'area di immagazzinamento determina quanti imballi sono stati trasportati nell'area di immagazzinamento. Il nastro 2 trasporta gli imballi dall'area di immagazzinamento temporaneo fino ad una piattaforma di carico dove degli autocarri sono pronti a ricevere il materiale da consegnare al cliente. Una barriera a fotocellula alla fine del nastro 2 in prossimità dell'area di immagazzinamento registra il numero degli imballi che escono dall'area di immagazzinamento per essere trasportati verso la piattaforma di carico. Un pannello visualizzatore dispone di cinque spie che segnalano il livello di riempimento dell'area di immagazzinamento.



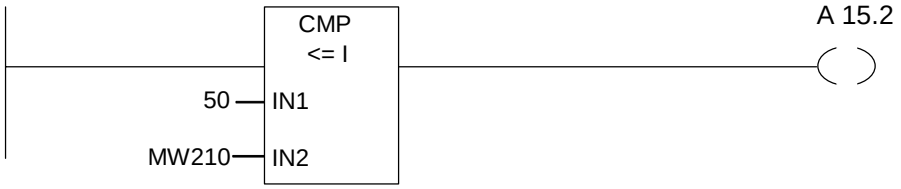
Schema a contatti KOP per l’attivazione delle spie su un pannello visualizzatore

Segmento 1: il contatore Z1 conta in avanti con un fronte di segnale da ”0” a ”1” all’ingresso ZV e conta all’indietro con un fronte di segnale da ”0” a ”1” all’ingresso ZR. Con un fronte di segnale da ”0” a ”1” all’ingresso S, il valore di conteggio viene settato sul valore di ZW. Con un fronte di segnale da ”0” a ”1” all’ingresso R, il valore di conteggio viene settato a ”0”. Nella MW 200 è sempre disponibile il valore di conteggio attuale di Z1. A12.1 indica ”Area di immagazzinamento non vuota”.

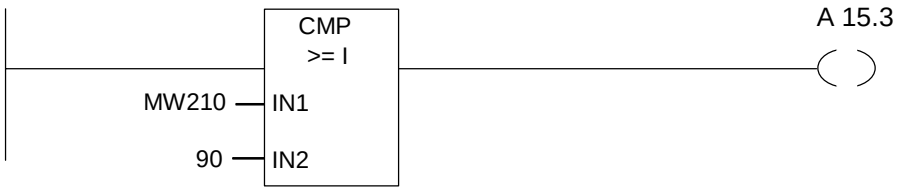


Segmento 2: A12.0 indica ”Area di immagazzinamento vuota”.

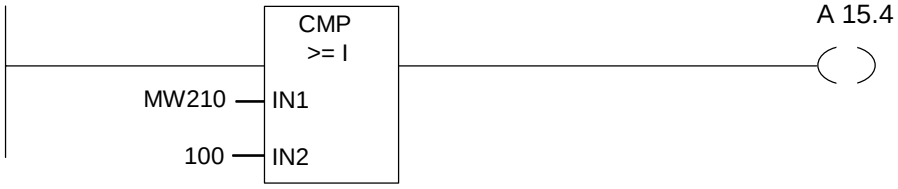
Segmento 3: se 50 è un valore inferiore o pari al valore di conteggio (se cioè lo stato del contatore attuale è maggiore o uguale a 50), si accende la spia di visualizzazione per il messaggio ”Area di immagazzinamento piena al 50%”.



Segmento 4: Se il valore di conteggio è maggiore o uguale a 90, si accende la spia di visualizzazione per il messaggio ”Area di immagazzinamento piena al 90%”.



Segmento 5: Se il valore di conteggio è maggiore o uguale a 100, si accende la spia di visualizzazione per il messaggio ”Area di immagazzinamento piena”.



B.5 Esempio: Operazioni matematiche con i numeri interi

Soluzione di un problema matematico

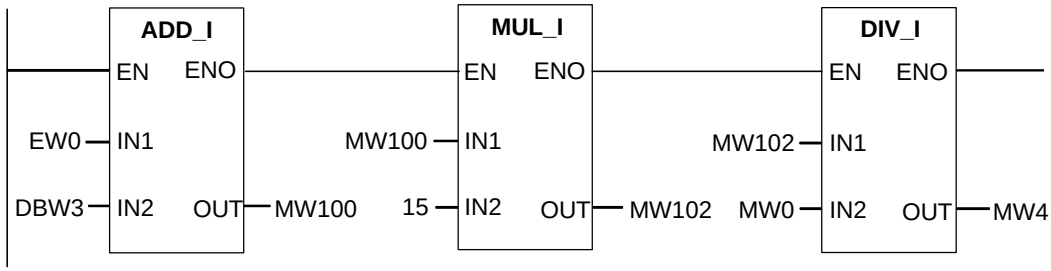
Il seguente programma di esempio mostra come utilizzare tre operazioni matematiche con numeri interi per conseguire lo stesso risultato che si ottiene dalla seguente equazione:

MW4 = ((EW0 + DBW3) x 15) / MW0

Schema a contatti KOP

Segmento 1: Apri blocco dati DB1

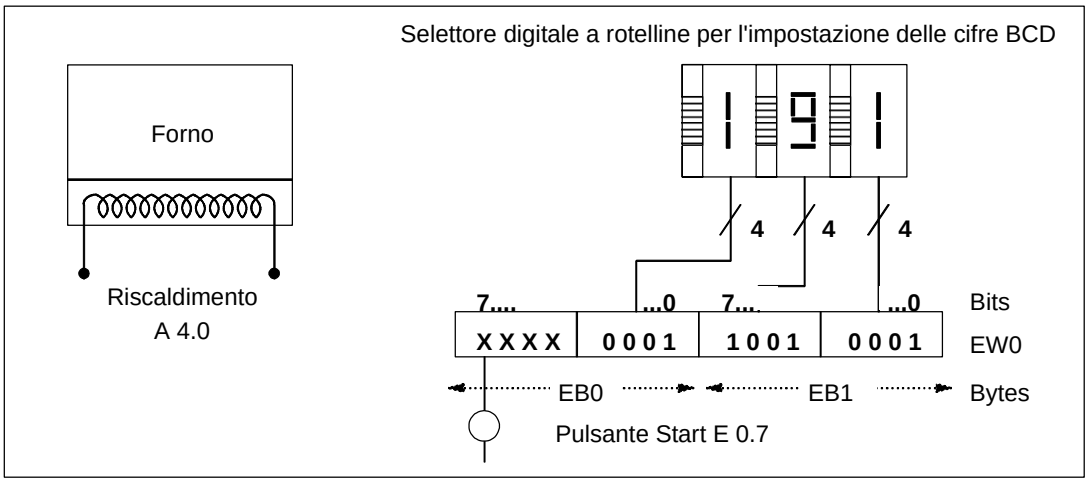
Segmento 2: viene aggiunta la parola d'ingresso EW0 e la parola di dati DBW3 (il blocco dati deve essere definito e aperto) e la somma caricata nella parola di memoria MW100. MW100 viene quindi moltiplicata per 15 e il risultato memorizzato nella parola di merker MW102. MW102 viene diviso per MW0 e il risultato memorizzato in MW4.



B.6 Esempio: Operazioni logiche combinatorie a parola

Riscaldamento di un forno

L'operatore di un forno avvia il riscaldamento del forno premendo il pulsante Start. L'operatore può impostare la durata del tempo di riscaldamento avvalendosi di selettori a rotella. Il valore che l'operatore imposta indica i secondi in formato BCD (decimali codificati in binario).



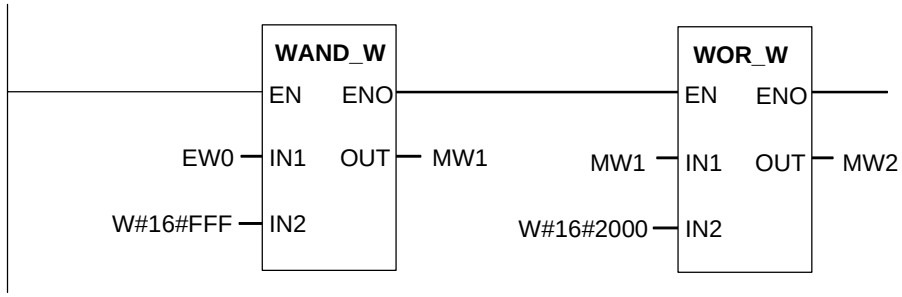
Componente del sistema	Indirizzo assoluto
Pulsante Start	E 0.7
Rotellina delle unità	da E 1.0 a E 1.3
Rotellina delle decine	da E 1.4 a E 1.7
Rotellina delle centinaia	da E 0.0 a E 0.3
Avvio riscaldamento	A 4.0

Schema a contatti KOP

Segmento 1: se il temporizzatore è in funzione, accendere il riscaldatore.

Segmento 2: se il temporizzatore è attivo, l'istruzione **Return** termina qui l'elaborazione.

Segmento 3: mascheramento del bit d'ingresso da E 0.4 a E 0.7 (ossia, resettarli a 0). Questi bit degli ingressi del selettore a rotelline non vengono utilizzati. I 16 bit di tali ingressi vengono combinati con W#16#0FFF secondo l'operazione **Combinazione AND a parola**. Il risultato è caricato nella parola di merker MW1. Per poter impostare la base di tempo in secondi, il valore predefinito viene combinato con l'operazione W#16#2000 **Combinazione OR a parola**, impostando il bit 13 a 1 e resettando il bit 12 a 0.



Segmento 4: quando viene premuto il pulsante Start, avviare il temporizzatore T 1 come un temporizzatore ad impulso prolungato caricandolo così come un valore predefinito della parola di merker MW2 (derivata dalla logica precedente).

C Come operare con KOP

C.1 Meccanismo EN/ENO

L'abilitazione (EN) e l'uscita di abilitazione (ENO) dei box FUP/KOP vengono realizzate tramite il bit BIE.

Se EN e ENO sono collegati:

ENO = EN AND NOT (errore di box)

Se non si verificano errori (errore di box = 0) si avrà ENO = EN.

Il meccanismo EN/ENO viene utilizzato per i seguenti scopi:

- operazioni matematiche
- operazioni di trasferimento e conversione
- operazioni di scorrimento e rotazione
- richiami di blocchi.

Questo meccanismo **non** viene utilizzato nei casi seguenti:

- confronti
- contatori
- temporizzatori.

Intorno ai comandi effettivi dei box vengono generati, per il meccanismo EN/ENO ulteriori comandi AWL a seconda delle operazioni logiche precedenti o successive esistenti. I quattro casi possibili vengono mostrati sull'esempio di un addendo:

- Addendo con collegamento EN e collegamento ENO
- Addendo con collegamento EN e senza collegamento ENO
- Addendo senza collegamento EN e con collegamento ENO
- Addendo senza collegamento EN e senza collegamento ENO

Avvertenze per la creazione di blocchi personali

Per scrivere blocchi da richiamare in FUP/KOP, è necessario fare in modo che uscendo dal blocco sia impostato il bit BIE. Il quarto esempio mostra che questa operazione non è automatica. Il BIE non può essere utilizzato come merker in quanto viene costantemente sovrascritto dal meccanismo EN/ENO. Utilizzare perciò una variabile temporanea nella quale memorizzare gli errori che si sono verificati. Inizializzare la variabile con 0. Impostare quindi questa variabile, con l'aiuto del meccanismo EN/ENO, in ogni posizione del blocco nella quale si pensa che un'operazione fallita possa rappresentare un errore per l'intero blocco. Sarà sufficiente un comando NOT e Imposta bobina. Programmare quindi un segmento alla fine del blocco.

```
fine:    errore UN
                SAVE
```

Osservare che il segmento verrà eseguito in ogni caso, il che significa che non è ammesso utilizzare un'operazione BEB all'interno del blocco e che il segmento non può essere ignorato.

C.1.1 Addendo con collegamento EN e collegamento ENO

Se l'addendo ha sia un collegamento EN che un collegamento ENO saranno rilevanti i seguenti comandi AWL:

```
1    U   E   0.0   // Collegamento EN
2    SPBNB _001    // Scorrimento di RLC in BIE e salto se RLC == 0
3    L     in1     // Parametro box
4    L     in2     // Parametro box
5    +I            // Addizione effettiva
6    T     out     // Parametro box
7    UN    OV      // Rilevamento errori
8    SAVE          // Memorizzazione errore in BIE
9    CLR           // Prima interrogazione
10   _001:     U   BIE     // Scorrimento di BIE in RLC
11   =      4.0
```

Dopo la riga 1, l'RLC contiene il risultato dell'operazione logica combinatoria precedente. Il comando SPBNB copia il risultato logico combinatorio nel bit BIE e imposta il bit della prima interrogazione.

- Se l'RLC è 0 si salta nella riga 10 e si continua con U BIE. L'addizione non viene eseguita. Nella riga 10 il BIE viene nuovamente copiato nell'RLC e quindi all'uscita viene assegnato il valore 0.
- Se l'RLC è 1 non si salta, il che significa che l'addizione viene eseguita. La riga 7 permette di determinare se con l'addizione si è verificato un errore, memorizzato nella riga 8 nel BIE. La riga 9 imposta il bit della prima interrogazione. A questo punto nella riga 10 il bit BIE viene copiato nuovamente nell'RLC, quindi nell'uscita viene visualizzato se l'addizione è riuscita.
  Il bit BIE non viene più modificato nelle righe 10 e 11 e quindi mostra allo stesso modo se la somma eseguita è corretta.

C.1.2 Addendo con collegamento EN e senza collegamento ENO

Se l'addendo ha un collegamento EN e non un collegamento ENO, sono rilevanti i seguenti comandi AWL:

```
1    U     E    0.0  // Collegamento EN
2    SPBNB _001      // Scorrimento RLC in BIE e salto se RLC == 0
3    L     in1       // Parametro box
4    L     in2       // Parametro box
5    +I              // Addizione effettiva
6    T     out       // Parametro box
7    _001: NOP   0
```

Dopo la riga 1 l'RLC contiene il risultato dell'operazione logica combinatoria. Il comando SPBNB copia l'RLC nel BIE e imposta il bit della prima interrogazione.

- Se l'RLC è 0 si salta nella riga 7, l'addizione non viene eseguita, RLC e BIE sono 0
- Se l'RLC era 1 non si salta, cioè l'addizione viene eseguita. Eventuali errori verificatisi nell'addizione non vengono determinati. RLC e BIE sono 1.

C.1.3 Addendo senza collegamento EN e con collegamento ENO

Se l'addendo non ha un collegamento EN ma un collegamento ENO, sono rilevanti i seguenti comandi AWL:

```
1    L     in1     // Parametro box
2    L     in2     // Parametro box
3    +I            // Addizione effettiva
4    T     out     // Parametro box
5    UN    OV      // Rilevamento errori
6    SAVE          // Memorizzazione errore in BIE
7    CLR           // Prima interrogazione
8    U     BIE     // Scorrimento di BIE in RLC
9    =     A      4.0
```

L'addizione viene eseguita in ogni caso. Nella riga 5 si determina se durante l'addizione si è verificato un errore, il che viene memorizzato nella riga 6 nel BIE. La riga 7 imposta il bit della prima interrogazione. A questo punto nella riga 8 il bit BIE viene copiato nuovamente nell'RLC e nell'uscita viene visualizzato se l'addizione è ruscita.

Il bit BIE non viene più modificato con le righe 8 e 9 e quindi mostra allo stesso modo se l'addizione eseguita era corretta o meno.

C.1.4 Addendo senza collegamento EN e senza collegamento ENO

Se l'addendo non ha collegamenti EN né ENO, sono rilevanti i seguenti comandi AWL:

```
1    L     in1     // Parametro box
2    L     in2     // Parametro box
3    +I            // Addizione effettiva
4    T     out     // Parametro box
5    NOP 0
```

L'addizione viene eseguita. L'RLC e il bit BIE restano invariati.

C.2 Assegnazione dei parametri

I parametri vengono assegnati a un blocco in forma di valori. Per quanto riguarda i blocchi funzionali viene utilizzata, all'interno del blocco richiamato, una copia del valore del parametro attuale nel DB di istanza. Nel caso delle funzioni, invece, si ha una copia del valore del parametro attuale nello stack dei dati locali. I puntatori non vengono copiati. Prima del richiamo i valori di INPUT vengono copiati nel DB di istanza o nello stack L. Dopo il richiamo i valori di OUTPUT vengono ricopiati nelle variabili. Nell'ambito del blocco richiamato si lavora soltanto su una copia. I comandi AWL necessari sono contenuti nel blocco richiamante e restano nascosti all'utente.

Nota

Se si utilizzano merker, ingressi, uscite, ingressi o uscite periferiche come operandi attuali di una funzione, tali operandi vengono trattati in modo diverso rispetto agli altri. L'aggiornamento avviene in questo caso direttamente e non tramite stack L.

Eccezione:

Se il relativo parametro formale è un parametro di ingresso del tipo di dati BOOL, il parametro attuale viene aggiornato mediante L-Stack.

Attenzione

Durante la programmazione del blocco richiamato, fare in modo che i parametri dichiarati come OUTPUT vengano anche descritti, altrimenti i valori emessi sarebbero casuali! Nel caso dei blocchi funzionali viene fornito all'utente il valore rilevato con l'ultimo richiamo dal DB di istanza, per le funzioni, invece, il valore contenuto casualmente nello stack L.

Tenere conto dei seguenti punti:

- inizializzare possibilmente tutti i parametri OUTPUT
- evitare se possibile di utilizzare i comandi di impostazione e resettaggio in quanto dipendono dall'RLC. Se l'RLC ha il valore 0, verrà mantenuto il valore casuale!
- saltando all'interno del blocco fare attenzione a non saltare posizioni nelle quali sono descritti parametri OUTPUT. Non dimenticare l'operazione BEB e le conseguenze dei comandi MCR.

Indice analitico

(

---( ) 1-6
---(#)--- 1-8
---(CD) 4-11
---(CU) 4-10
---(JMPN) 6-4
---(N)--- 1-18
---(P)--- 1-19
---(R) 1-10
---(S) 1-12
---(SA) 13-24
---(SC) 4-9
---(SD) 13-20
---(SE) 13-18, 13-20
---(SF) 13-24
---(SI) 13-16
---(SP) 13-16
---(SS) 13-22
---(SV) 13-18
---(SZ) 4-9
---(ZR) 4-11
---(ZV) 4-10
---(Call) 10-2
---(JMP)--- Salto condizionato 6-3
---(JMP)--- Salto assoluto 6-2
(LABEL Etichetta di salto 6-5
---(MCR<) 10-14
---(MCR>) 10-16, 10-17
---(MCRA) 10-18
---(MCRD) 10-19
---(OPN) 5-1
---(RET) 10-20
---(SAVE) 1-20

|

---| |--- 1-2
---| / |--- 1-3
--|NOT|-- 1-5

<

<=0 ---| |--- 12-11
<=0 ---| / |--- 12-11
<>0 ---| |--- 12-8
<>0 ---| / |--- 12-8
<0 ---| |--- 12-12
<0 ---| / |--- 12-12

=

==0 ---| |--- 12-7
==0 ---| / |--- 12-7

>

>=0 ---| |--- 12-9
>=0 ---| / |--- 12-9
>0 ---| |--- 12-10
>0 ---| / |--- 12-10

A

ABS 8-8
ACOS Formazione dell'arcocoseno di un numero in virgola mobile 8-17
ADD_DI 7-7
ADD_I 7-3
ADD_R 8-4
Addendo con collegamento EN e collegamento ENO C-3
Addendo con collegamento EN e senza collegamento ENO C-4
Addendo senza collegamento EN e con collegamento ENO C-5
Addendo senza collegamento EN e senza collegamento ENO C-6
Applicazione pratiche B-1
Apri un blocco dati 5-1
Area di memoria e componenti di un temporizzatore 13-2
Arrontonda a numero intero (a 32 bit) .. 3-13
Arrotonda senza resto un numero intero (a 32 bit) 3-14
ASIN Formazione dell'arcoseno di un numero in virgola mobile 8-16
Assegna valore 9-1
Assegnazione dei parametri C-7
ATAN Formazione dell'arcotangente di un numero in virgola mobile 8-18
Attiva zona relè master control 10-14
Avvertenze importanti sulle funzionalità MCR 10-13
Avvia bobina temporizzatore come impulso 13-16
Avvia bobina temporizzatore come impulso prolungato 13-18
Avvia bobina temporizzatore come ritardo alla disinserzione (SA) 13-24

Avvia bobina temporizzatore
come ritardo all'inserzione 13-20
Avvia bobina temporizzatore
come ritardo all'inserzione
con memoria (SS) 13-22

B

BCD_DI 3-5
BCD_I 3-2
BIE ---| |--- 12-6
BIE ---| / |--- 12-6
Bit di anomalia
operazione non ammessa 12-5
overflow 12-2
overflow con memoria 12-3
registro BIE 12-6
Bit di anomalia negato
operazione non ammessa 12-5
overflow 12-2
overflow con memoria 12-3
registro BIE 12-6
Bit di risulato diverso da 0 12-8
Bit di risulato negato diverso da 0 12-8
Bit di risultato maggiore di 0 12-10
Bit di risultato maggiore o uguale a 0 ... 12-9
Bit di risultato minore di 0 12-12
Bit di risultato minore o uguale a 0 12-11
Bit di risultato negato maggiore di 0 ... 12-10
Bit di risultato negato maggiore
o uguale a 0 12-9
Bit di risultato negato minore di 0 12-12
Bit di risultato negato minore
o uguale a 0 12-11
Bit di risultato negato uguale a 0 12-7
Bit di risultato uguale a 0 12-7
Bobina di uscita 1-6

C

CALL_FB 10-4
CALL_FC 10-6
CALL_SFB 10-8
CALL_SFC 10-10
CEIL 3-15
CMP ? D 2-3
CMP ? I 2-2
CMP ? R 2-4
Combinazione AND doppia parola 14-5
Combinazione AND parola 14-2
Combinazione esclusivo parola 14-4
Combinazione OR doppia parola 14-6
Combinazione OR esclusivo 1-4
Combinazione OR esclusivo
doppia parola 14-7
Combinazione OR parola 14-3
Complemento a due di numero in
virgola mobile 3-12
Complemento a due di numero intero
(a 16 bit) 3-10
Complemento a due di numero intero
(a 32 bit) 3-11
Complemento a uno di numero intero
(a 16 bit) 3-8
Complemento a uno di numero intero
(a 32 bit) 3-9
Componenti di un temporizzatore 13-2
Confronta numeri in virgola mobile
(== <> > < >= <=) 2-4
Confronta numeri interi a 16 bit
(== <> > < >= <=) 2-4
Confronta numeri interi a 32 bit
(== <> > < >= <=) 2-4
Connettore 1-8
Conta all'indietro 4-11
Conta in avanti 4-10
Contatto normalmente aperto 1-2
Contatto normalmente chiuso 1-3
Converti un numero BCD in
un numero intero (a 16 bit) 3-2
Converti un numero BCD in
un numero intero (a 32 bit) 3-5
Converti un numero intero (a 16 bit) in
un numero BCD 3-3
Converti un numero intero (a 16 bit) in
un numero intero (a 32 bit) 3-4
Converti un numero intero (a 32 bit) in
un numero BCD 3-6
Converti un numero intero (a 32 bit) in
un numero in virgola mobile 3-7
COS Formazione del coseno di
un angolo come numero
in virgola mobile 8-14

D

DI_BCD 3-6
DI_R 3-7
Disattiva zona relè master control 10-16
DIV_DI 7-10
DIV_I 7-6
DIV_R 8-7
Dividi numeri in virgola mobile 8-7
Dividi numeri interi a 16 bit 7-6
Dividi numeri interi a 32 bit 7-10

E

Esempi
Operazioni logiche combinatorie
a bit B-2
Esempi di programmazione B-1

Esempio
Operazioni di conteggio e confronto . B-10
Operazioni di temporizzazione B-6
Operazioni logiche combinatorie a parola..... B-14
Operazioni matematiche con i numeri interi..... B-13
Etichetta di salto 6-5
EXP Formazione del valore esponenziale di un numero in virgola mobile 8-11

F

Fai ruotare doppia parola verso destra (a 32 bit)..... 11-16
Fai ruotare doppia parola verso sinistra (a 32 bit)..... 11-14
Fai scorrere doppia parola verso destra (a 32 bit)..... 11-11
Fai scorrere doppia parola verso sinistra (a 32 bit)..... 11-9
Fai scorrere numero intero a 16 bit verso destra 11-2
Fai scorrere numero intero a 32 bit verso destra 11-4
Fai scorrere parola verso destra (a 16 bit)..... 11-5
Fai scorrere parola verso sinistra (a 16 bit)..... 11-7
Fine zona relè master control..... 10-19
FLOOR 3-16
Formazione del valore assoluto di un numero in virgola mobile..... 8-8

G

Genera da un numero in virgola mobile un numero intero inferiore..... 3-16
Genera da un numero in virgola mobile un numero intero superiore..... 3-15
Guida online v

I

I_BCD 3-3
I_DI 3-4
Imposta resetta flip flop 1-16
Imposta uscita 1-12
Imposta valore di conteggio..... 4-9
Inizio zona relè master control..... 10-18
Interroga rilevamento di fronte di discesa..... 1-21
Interroga rilevamento di fronte di salita..... 1-22
INV_D 3-9
INV_I..... 3-8
Inverti il risultato della combinazione..... 1-5

L

Leggi direttamente 1-23
LN Formazione del logaritmo naturale di un numero in virgola mobile 8-12

M

Meccanismo EN/ENO..... C-1
Mnemonico
inglese (internazionale) A-5
tedesco (SIMATIC).....A-1
MOD_DI 7-11
Moltiplica numeri in virgola mobile..... 8-6
Moltiplica numeri interi a 16 bit 7-5
Moltiplica numeri interi a 32 bit 7-9
MOVE 9-2
MUL_DI.....7-9
MUL_I 7-5
MUL_R.....8-6

N

NEG 1-21
NEG_DI.....3-11
NEG_I 3-10
NEG_R.....3-12

O

Operazioni di bit di stato 12-1
Operazioni di comando del programma10-1
Operazioni di confronto.....2-1
Operazioni di Conteggio 4-1
Operazioni di conversione 3-1
Operazioni di rotazione.....11-13
Operazioni di salto 6-1, 6-5
Operazioni di scorrimento.....11-1
Operazioni di temporizzazione 13-1
Operazioni KOP ordinate secondo il set mnemonico inglese (internazionale) ...A-5
Operazioni KOP ordinate secondo il set mnemonico tedesco (SIMATIC) A-1
Operazioni logiche combinatorie a parola 14-1
Operazioni logiche combinatorie di bit.... 1-1
Operazioni matematiche con i numeri interi.....7-1
Operazioni matematiche con i numeri mobile.....8-1
OS ---| |---.....12-3
OS ---| / |---.....12-3
OV ---| |---.....12-2
OV ---| / |---.....12-2

P

Parametrizza e avvia bobina temporizzatore come impulso (SI) 13-6
Parametrizza e avvia bobina temporizzatore come impulso prolungato (SV) 13-8
Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione (SA) 13-14
Parametrizza e avvia bobina temporizzatore come ritardo all'inerzione con memoria (SS) 13-12
Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione (SE) 13-10
Parametrizza e conta all'indietro 4-7
Parametrizza e conta in avanti 4-5
Parametrizza e conta in avanti/ all'indietro 4-3
POS 1-22

R

Resetta imposta flip flop 1-14
Resetta uscita 1-10
Ricava resto della divisione (a 32 bit) ... 7-11
Richiama FB dal box 10-4
Richiama FC dal box 10-6
Richiama FC/SFC dalla bobina (senza parametri) 10-2
Richiama SFB dal box 10-8
Richiama SFC dal box 10-10
Richiamo di blocchi da una biblioteca 10-13
Richiamo di una multi-istanza 10-12
Rileva fronte RLC di discesa 1-18
Rileva fronte RLC di salita 1-19
ROL_DW 11-14, 11-15
ROR_DW 11-16, 11-17
ROUND 3-13
RS 1-14

S

S_AVERZ 13-14
S_CD 4-7
S_CU 4-5
S_CUD 4-3
S_EVERZ 13-10
S_IMPULS 13-6
S_ODT 13-10
S_ODTS 13-12
S_OFFDT 13-14
S_PEXT 13-8
S_PULSE 13-6
S_SEVERZ 13-12
S_VIMP 13-8
Salta indietro 10-20
Salva RLC nel registro BIE 1-20
Scrivi direttamente 1-24
Se 0 salta in un FB 6-4
SHL_DW 11-9, 11-10
SHL_W 11-7, 11-8
SHR_DI 11-4
SHR_DW 11-11, 11-12
SHR_I 11-2, 11-3
SHR_W 11-5, 11-6
SIN Formazione del seno di un angolo come numero in virgola mobile 8-13
Somma numeri in virgola mobile 8-3
Somma numeri interi a 16 bit 7-3
Somma numeri interi a 32 bit 7-7
Sommario ..... 1-1, 2-1, 3-1, 4-1, 6-1, 7-1, .. 8-1, 10-1, 11-1, 11-13, 12-1, 13-1, 14-1
Sottrai numeri in virgola mobile 8-5
Sottrai numeri interi a 16 bit 7-4
Sottrai numeri interi a 32 bit 7-8
SQR Formazione del quadrato di un numero in virgola mobile 8-9
SQRT Formazione della radice quadrata di un numero in virgola mobile 8-10
SR 1-16
SUB_DI 7-8
SUB_I 7-4
SUB_R 8-5

T

TAN Formazione della tangente di un angolo come numero in virgola mobile 8-15
TRUNC 3-14

U

UO ---| |--- 12-5
UO ---| / |--- 12-5

V

Valutazione dei bit nella parola di stato con operazioni in virgola fissa 7-2
Valutazione dei bit nella parola di stato con operazioni in virgola mobile 8-2

W

WAND_DW 14-5
WAND_W 14-2
WOR_DW 14-6
WOR_W 14-3
WXOR_DW 14-7
WXOR_W 14-4

X

XOR 1-4

Z

Z_RUECK 4-7
Z_VORW 4-5
ZÄHLER 4-3