

# SIEMENS

## SIMATIC

### Schema logico (FUP) per S7-300/400

Manuale di riferimento

Il presente manuale fa parte del pacchetto di documentazione con il numero di ordinazione:

6ES7810-4CA10-8EW1




05/2010  
A5E02790134-01

Operazioni logiche combinatorie a bit	1
Operazioni di confronto	2
Operazioni di conversione	3
Operazioni di conteggio	4
Operazioni di blocchi dati	5
Operazioni di salto	6
Operazioni matematiche con i numeri interi	7
Operazioni con numeri 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 combinatorie a parola	14
Sommario di tutte le operazioni FUP	A
Esempi di programmazione	B
Struttura ed elementi in FUP	C

## Avvertenze di legge

### Concetto di segnaletica di avvertimento

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 decrescente i diversi livelli di rischio.

 <b>PERICOLO</b>
questo simbolo indica che la mancata osservanza delle opportune misure di sicurezza <b>provoca</b> la morte o gravi lesioni fisiche.
 <b>AVVERTENZA</b>
il simbolo indica che la mancata osservanza delle relative misure di sicurezza <b>può causare</b> la morte o gravi lesioni fisiche.
 <b>CAUTELA</b>
con il triangolo di pericolo indica che la mancata osservanza delle relative misure di sicurezza può causare lesioni fisiche non gravi.
<b>CAUTELA</b>
senza triangolo di pericolo indica che la mancata osservanza delle relative misure di sicurezza può causare danni materiali.
<b>ATTENZIONE</b>
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

Il prodotto/sistema oggetto di questa documentazione può essere adoperato solo da **personale qualificato** per il rispettivo compito assegnato nel rispetto della documentazione relativa al compito, specialmente delle avvertenze di sicurezza e delle precauzioni in essa contenute. Il personale qualificato, in virtù della sua formazione ed esperienza, è in grado di riconoscere i rischi legati all'impiego di questi prodotti/sistemi e di evitare possibili pericoli.

### Uso conforme alle prescrizioni di prodotti Siemens

Si prega di tener presente quanto segue:

 <b>AVVERTENZA</b>
I prodotti Siemens devono essere utilizzati solo per i casi d' impiego previsti nel catalogo e nella rispettiva documentazione tecnica. Qualora vengano impiegati prodotti o componenti di terzi, questi devono essere consigliati oppure approvati da Siemens. Il funzionamento corretto e sicuro dei prodotti presuppone un trasporto, un magazzinaggio, un' installazione, un montaggio, una messa in servizio, un utilizzo e una manutenzione appropriati e a regola d' arte. Devono essere rispettate le condizioni ambientali consentite. Devono essere osservate le avvertenze contenute nella rispettiva documentazione.

### Marchio di prodotto

Tutti 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.

# Prefazione

## Scopo del manuale

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

Esso descrive gli elementi del linguaggio di programmazione FUP, 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 MS Windows XP, MS Windows Server 2003 o MS Windows 7.

## Validità del manuale

Il presente manuale ha validità per il pacchetto software STEP 7 V5.5.

## 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.RTF di STEP 7.

## Presupposti

Il presente manuale di FUP 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 <ul style="list-style-type: none"> <li>• Primi passi ed esercitazioni con STEP 7</li> <li>• Programmazione con STEP 7</li> <li>• Configurazione dell'hardware e progettazione di collegamenti con STEP 7</li> <li>• Manuale di conversione: STEP 7, da S5 a S7</li> </ul>	Conoscenze di base per il personale tecnico: procedure per la realizzazione di compiti di controllo con STEP 7 e S7-300/400	6ES7810-4CA10-8EW0
Nozioni di riferimento di STEP 7 con <ul style="list-style-type: none"> <li>• Manuali KOP/FUP/AWL per S7-300/400</li> <li>• Funzioni standard e di sistema per S7-300/400 Volume 1 e Volume 2</li> </ul>	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.	6ES7810-4CA10-8EW1

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
Guida 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 FUP. 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.

Internet: <http://www.sitrain.com>

## Technical Support

Per tutti i prodotti Industry Automation and Drive Technology è possibile rivolgersi al Technical Support

- mediante il modulo Web per la Support Request  
<http://www.siemens.com/automation/support-request>

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 Industry Automation and Drive Technology;
- informazioni su Riparazioni, pezzi di ricambio e consulenza.

# Indice

<b>1</b>	<b>Operazioni logiche combinatorie a bit</b> .....	<b>11</b>
1.1	Sommario delle operazioni logiche combinatorie di bit.....	11
1.2	>=1: Combinazione OR .....	12
1.3	&: Combinazione logica AND.....	13
1.4	Combinazione logica AND prima di OR o combinazione logica OR prima di AND.....	14
1.5	XOR: Combinazione logica OR esclusivo .....	16
1.6	Inserimento di un ingresso binario.....	17
1.7	Negazione di un ingresso binario.....	18
1.8	=: Assegnazione .....	19
1.9	#: Connettore .....	21
1.10	R: Resettaggio della bobina.....	23
1.11	S: Impostazione della bobina.....	24
1.12	RS: Resettaggio impostazione flip flop.....	25
1.13	SR: Impostazione resettaggio flip flop .....	26
1.14	N: Rilevamento del fronte RLC di discesa .....	28
1.15	P: Rilevamento del fronte RLC di salita .....	29
1.16	SAVE: Memorizzazione dell'RLC nel registro BIE.....	30
1.17	NEG: Interrogazione del rilevamento di fronte di discesa .....	31
1.18	POS: Interrogazione del rilevamento del fronte RLC di salita .....	32
<b>2</b>	<b>Operazioni di confronto</b> .....	<b>33</b>
2.1	Sommario delle operazioni di confronto .....	33
2.2	CMP ? I: Confronto di numeri interi (16 bit).....	34
2.3	CMP ? D: Confronto di numeri interi (32 bit).....	35
2.4	CMP ? R: Confronto di numeri in virgola mobile.....	36
<b>3</b>	<b>Operazioni di conversione</b> .....	<b>37</b>
3.1	Sommario delle operazioni di conversione .....	37
3.2	BCD_I: Conversione di un numero BCD in un numero intero (16 bit).....	38
3.3	I_BCD: Conversione di un numero intero (16 bit) in un numero BCD .....	39
3.4	BCD_DI: Conversione di un numero BCD in un numero intero (32 bit) .....	40
3.5	I_DI: Conversione di un numero intero (16 bit) in un numero intero (32 bit) .....	41
3.6	DI_BCD: Conversione di un numero intero (32 bit) in un BCD.....	42
3.7	DI_R: Conversione di un numero intero (32 bit) in un numero in virgola mobile.....	43
3.8	INV_I: Complemento a 1 di numero intero (16 bit) .....	44
3.9	INV_DI: Complemento a 1 di numero intero (32 bit).....	45
3.10	NEG_I: Complemento a 2 di numero intero (16 bit) .....	46
3.11	NEG_DI: Complemento a 2 di numero intero (32 bit).....	47
3.12	NEG_R: Negazione del numero in virgola mobile .....	48
3.13	ROUND: Arrotondamento a numero intero.....	49
3.14	TRUNC: Arrotondamento senza resto di un numero intero.....	50
3.15	CEIL: Generazione di un numero intero superiore da un numero in virgola mobile.....	51
3.16	FLOOR: Generazione di un numero intero inferiore da un numero in virgola mobile .....	52

<b>4</b>	<b>Operazioni di conteggio.....</b>	<b>53</b>
4.1	Sommario delle operazioni di conteggio .....	53
4.2	ZAEHLER: Parametrizzazione e conteggio in avanti / all'indietro .....	55
4.3	Z_VORW: Parametrizzazione e conteggio in avanti.....	57
4.4	Z_RUECK: Parametrizzazione e conteggio all'indietro.....	59
4.5	SZ: Impostazione del valore di conteggio.....	61
4.6	ZV: Conta in avanti.....	63
4.7	ZR: Conta all'indietro.....	64
<b>5</b>	<b>Operazioni di blocchi dati.....</b>	<b>65</b>
5.1	OPN: Apertura di un blocco dati .....	65
<b>6</b>	<b>Operazioni di salto .....</b>	<b>67</b>
6.1	Sommario delle operazioni di salto .....	67
6.2	JMP: Salto nel blocco assoluto .....	68
6.3	JMP: Salto nel blocco se 1 (condizionato).....	69
6.4	JMPN: Salto nel blocco se 0 (condizionato) .....	70
6.5	LABEL: Etichetta di salto.....	71
<b>7</b>	<b>Operazioni matematiche con i numeri interi.....</b>	<b>73</b>
7.1	Sommario delle operazioni matematiche con i numeri interi .....	73
7.2	Valutazione dei bit della parola di stato nelle operazioni in virgola fissa .....	74
7.3	ADD_I: Somma di numeri interi (16 bit) .....	75
7.4	SUB_I: Sottrazione di numeri interi (16 bit).....	76
7.5	MUL_I: Moltiplicazione di numeri interi (16 bit).....	77
7.6	DIV_I: Divisione di numeri interi (16 bit).....	78
7.7	ADD_DI: Somma di numeri interi (32 bit).....	79
7.8	SUB_DI: Sottrazione di numeri interi (32 bit).....	80
7.9	MUL_DI: Moltiplicazione di numeri interi (32 bit) .....	81
7.10	DIV_DI: Divisione di numeri interi (32 bit).....	82
7.11	MOD_DI: Ricavo del resto della divisione (32 bit) .....	83
<b>8</b>	<b>Operazioni con numeri in virgola mobile.....</b>	<b>85</b>
8.1	Sommario delle operazioni matematiche con i numeri mobile .....	85
8.2	Valutazione dei bit della parola di stato nelle operazioni in virgola mobile .....	86
8.3	Operazioni di base .....	87
8.3.1	ADD_R: Somma di numeri in virgola mobile.....	87
8.3.2	SUB_R: Sottrazione di numeri in virgola mobile .....	88
8.3.3	MUL_R: Moltiplicazione di numeri in virgola mobile .....	89
8.3.4	DIV_R: Divisione di numeri in virgola mobile .....	90
8.3.5	ABS: Formazione del valore assoluto di un numero in virgola mobile .....	91
8.4	Operazioni avanzati .....	92
8.4.1	SQR: Formazione del quadrato di un numero in virgola mobile .....	92
8.4.2	SQRT: Formazione della radice quadrata di un numero in virgola mobile .....	93
8.4.3	EXP: Formazione del valore esponenziale di un numero in virgola mobile.....	94
8.4.4	LN: Formazione del logaritmo naturale di un numero in virgola mobile .....	95
8.4.5	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile .....	96
<b>9</b>	<b>Operazioni di trasferimento.....</b>	<b>99</b>
9.1	MOVE: Assegnazione di un valore .....	99



<b>10</b>	<b>Operazioni di controllo del programma .....</b>	<b>101</b>
10.1	Sommario delle operazioni di comando del programma .....	101
10.2	CALL: Richiamo di FC/SFC senza parametri .....	102
10.3	CALL_FB Richiama FB dal box .....	104
10.4	CALL_FC Richiama FC dal box .....	106
10.5	CALL_SFB Richiama SFB dal box .....	108
10.6	CALL_SFC Richiama SFC dal box .....	110
10.7	Richiamo di una multi-istanza .....	112
10.8	Richiamo di blocchi da una biblioteca .....	112
10.9	Funzioni del relè master control (MCR) .....	113
10.10	Avvertenze importanti sulle funzionalità MCR .....	114
10.11	MCR< / MCR>: Attivazione/disattivazione di una zona di relè master control .....	115
10.12	MCRA/ MCRD: Inizio/fine della zona relè master control .....	118
10.13	RET: Salto indietro .....	121
<b>11</b>	<b>Operazioni di scorrimento e rotazione.....</b>	<b>123</b>
11.1	Operazioni di scorrimento .....	123
11.1.1	Sommario delle operazioni di scorrimento .....	123
11.1.2	SHR_I: Fai scorrere numero intero a destra (a 16 bit) .....	124
11.1.3	SHR_DI: Fai scorrere numero intero a destra (a 32 bit) .....	126
11.1.4	SHL_W: Fai scorrere parola a sinistra (a 16 bit) .....	127
11.1.5	SHR_W: Fai scorrere parola a destra .....	129
11.1.6	SHL_DW: Fai scorrere doppia parola a sinistra .....	130
11.1.7	SHR_DW: Fai scorrere doppia parola a destra .....	131
11.2	Operazioni di rotazione .....	133
11.2.1	Sommario delle operazioni di rotazione .....	133
11.2.2	ROL_DW: Fai ruotare doppia parola a sinistra (a 16 bit) .....	133
11.2.3	ROR_DW: Fai ruotare doppia parola a destra (a 32 bit) .....	135
<b>12</b>	<b>Operazioni di bit di stato .....</b>	<b>137</b>
12.1	Sommario delle operazioni di bit di stato .....	137
12.2	OV: Bit di anomalia: overflow .....	138
12.3	OS: Bit di anomalia: overflow con memoria .....	140
12.4	UO: Bit di anomalia: operazione non ammessa .....	142
12.5	BIE: Registro BIE bit di anomalia .....	143
12.6	<> 0: Bit di risultato .....	144
<b>13</b>	<b>Operazioni di temporizzazione.....</b>	<b>147</b>
13.1	Sommario delle operazioni di temporizzazione .....	147
13.2	Aree di memoria e componenti di un temporizzatore .....	148
13.3	S_IMPULS: Avviamento e parametrizzazione del temporizzatore come impulso .....	152
13.4	S_VIMP: Avviamento e parametrizzazione del temporizzatore come impulso prolungato .....	154
13.5	S_EVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo all'inserzione .....	156
13.6	S_SEVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo all'inserzione con memoria .....	158
13.7	S_AVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo alla disinserzione .....	160
13.8	SI: Avvia temporizzatore come impulso .....	162
13.9	SV: Avvia temporizzatore come impulso prolungato .....	164
13.10	SE: Avvia temporizzatore come ritardo all'inserzione .....	166
13.11	SS: Avvia temporizzatore come ritardo all'inserzione con memoria .....	168
13.12	SA: Avvia temporizzatore come ritardo alla disinserzione .....	170

<b>14</b>	<b>Operazioni logiche combinatorie a parola.....</b>	<b>173</b>
14.1	Sommario delle operazioni logiche combinatorie a parola .....	173
14.2	WAND_W: Combinazione AND parola .....	174
14.3	WOR_W: Combinazione OR parola .....	175
14.4	WXOR_W: Combinazione OR esclusivo parola .....	176
14.5	WAND_DW: Combinazione AND doppia parola.....	177
14.6	WOR_DW: Combinazione OR doppia parola .....	178
14.7	WXOR_DW: Combinazione OR esclusivo doppia parola.....	179
<b>A</b>	<b>Sommario di tutte le operazioni FUP.....</b>	<b>181</b>
A.1	Operazioni FUP ordinate secondo il set mnemonico tedesco (SIMATIC).....	181
A.2	Operazioni FUP ordinate secondo il set mnemonico inglese (internazionale) .....	185
<b>B</b>	<b>Esempi di programmazione.....</b>	<b>189</b>
B.1	Sommario .....	189
B.2	Esempi: Operazioni logiche combinatorie a bit.....	190
B.3	Esempio: Operazioni di temporizzazione.....	193
B.4	Esempio: Operazioni di conteggio e confronto .....	197
B.5	Esempio: Operazioni matematiche con i numeri interi .....	200
B.6	Esempio: Operazioni logiche combinatorie a parola .....	201
<b>C</b>	<b>Struttura ed elementi in FUP .....</b>	<b>203</b>
C.1	Meccanismo EN/ENO .....	203
C.1.1	Addendo con collegamento EN e collegamento ENO .....	205
C.1.2	Addendo con collegamento EN e senza collegamento ENO .....	206
C.1.3	Addendo senza collegamento EN e con collegamento ENO .....	206
C.1.4	Addendo senza collegamento EN e senza collegamento ENO.....	207
14.8	Assegnazione dei parametri.....	208
	<b>Indice analitico .....</b>	<b>209</b>

# 1 Operazioni logiche combinatorie a 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 (sistema binario) e vengono denominate "cifre binarie" o semplicemente "bit". Nell'ambito delle istruzioni AND, OR, XOR e delle uscite, 1 equivale ad un "Sì logico" e 0 ad un "NO logico".

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).

Sono disponibili le seguenti operazioni logiche combinatorie a bit:

- AND, OR e XOR: interrogano lo stato di segnale e producono un risultato che può essere copiato nel bit RLC oppure combinato con esso.
- Combinazione logica AND prima di OR o combinazione logica OR prima di AND
- Le operazioni Assegnazione e Connettore assegnano l'RLC, o lo memorizzano, temporaneamente.

Le seguenti operazioni rispondono ad un RLC di 1:

- S: Impostazione della bobina
- R: Resettaggio della bobina
- SR: Impostazione resettaggio flip flop
- RS: Resettaggio impostazione flip flop

Altre operazioni rispondono ad una transizione nel RLC:

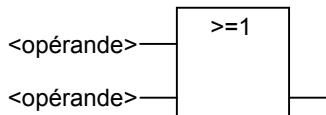
- N: Rilevamento del fronte RLC di discesa
- P: Rilevamento del fronte RLC di salita
- NEG: Interrogazione del rilevamento di fronte di discesa
- POS: Interrogazione del rilevamento del fronte RLC di salita

Le restanti operazioni influenzano direttamente l'RLC nel seguente modo:

- Inserimento di un ingresso binario
- Negazione di un ingresso binario
- SAVE: Memorizzazione dell'RLC nel registro BIE

## 1.2 >=1: Combinazione OR

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, T, Z, D, L	L'operando indica il bit di cui viene interrogato lo stato di segnale.

### Descrizione

L'utente può utilizzare l'operazione **OR** per interrogare gli stati di segnale di due o più operandi specificati negli ingressi di un box OR.

Se lo stato di segnale di uno degli operandi è 1, la condizione posta viene soddisfatta e l'operazione dà il risultato 1. Se lo stato di segnale degli operandi è 0, la condizione posta non viene soddisfatta e l'operazione dà il risultato 0.

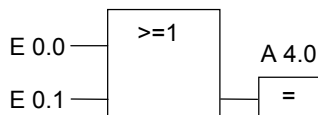
Se l'operazione **OR** rappresenta la prima operazione in una stringa logica, l'operazione memorizza il risultato della sua interrogazione dello segnale nel bit RLC.

Qualsiasi operazione **OR**, che non sia la prima operazione in una stringa logica, combina il risultato della sua interrogazione dello stato di segnale con il valore che è memorizzato nel bit RLC. Questa combinazione viene eseguita secondo la tabella della verità OR.

### Parola di stato

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

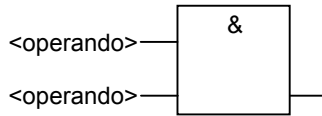
### Esempio



L'uscita A 4.0 è impostata se lo stato di segnale degli ingressi E 0.0 OR E 0.1 è 1.

## 1.3 &: Combinazione logica AND

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, T, Z, D, L	L'operando indica il bit di cui viene interrogato lo stato di segnale.

### Descrizione

L'utente può utilizzare l'operazione **AND** per interrogare lo stato di segnale di due o più operandi specificati negli ingressi di un box AND.

Se lo stato di segnale di tutti gli operandi è 1, la condizione viene soddisfatta e l'operazione dà il risultato 1. Se lo stato di segnale di un operando è 0, la condizione non viene soddisfatta e l'operazione dà il risultato 0.

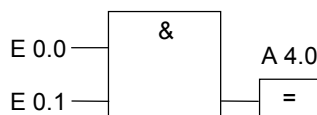
Se l'operazione **AND** rappresenta la prima operazione in una stringa logica, essa memorizza il risultato della propria interrogazione di segnale nel bit RLC.

Tutte le operazioni **AND**, che non siano le prime di una stringa logica, combinano il risultato della propria interrogazione dello stato di segnale con il valore memorizzato nel bit RLC. Esse vengono eseguite in base alla tabella della verità AND.

### Parola di stato

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

### Esempio



L'uscita A 4.0 è impostata se lo stato di segnale degli ingressi E 0.0 AND E 0.1 è 1.

## 1.4 Combinazione logica AND prima di OR o combinazione logica OR prima di AND

### Descrizione

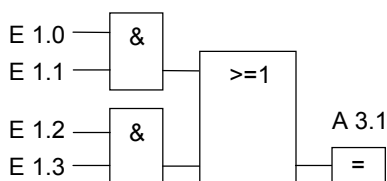
L'operazione **AND prima di OR** può essere utilizzata per interrogare il risultato di un'interrogazione di segnale secondo la tabella della verità OR.

In un'operazione **AND prima di OR** lo stato di segnale è 1 quando viene soddisfatta almeno una combinazione logica AND.

### Parola di stato

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

### Esempio



L'uscita A 3.1 ha lo stato di segnale 1 se viene soddisfatta almeno una combinazione logica AND.

L'uscita A 3.1 ha lo stato di segnale 0 se non viene soddisfatta alcuna combinazione logica AND.

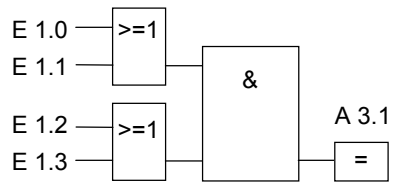
### Descrizione

L'operazione **OR prima di AND** può essere utilizzata per interrogare il risultato di un'interrogazione di segnale secondo la tabella della verità AND.

In un'operazione **OR prima di AND** lo stato di segnale è 1 quando viene soddisfatta almeno una combinazione logica OR.

### Parola di stato

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

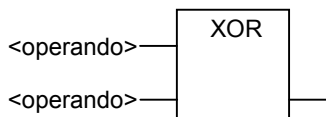
**Esempio**

L'uscita A 3.1 ha lo stato di segnale 1 se vengono soddisfatte entrambi le combinazioni logiche OR.

L'uscita A 3.1 ha lo stato di segnale 0 se non viene soddisfatta almeno una delle combinazioni logiche OR.

## 1.5 XOR: Combinazione logica OR esclusivo

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, T, Z, D, L	L'operando indica il bit di cui viene interrogato lo stato di segnale.

### Descrizione

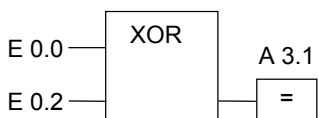
L'operazione **OR esclusivo** può essere utilizzata per interrogare il risultato di un'interrogazione di segnale secondo la tabella della verità OR esclusivo.

In una combinazione logica OR esclusivo lo stato di segnale è 1 quando lo stato di segnale di uno dei due operandi specificati è 1. Con gli elementi XOR per l'interrogazione di più di due operandi il risultato logico combinatorio condiviso è "1", se un numero dispari di operandi interrogati dà come risultato dell'interrogazione "1".

### Parola di stato

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

### Esempio



L'uscita A 3.1 ha lo stato di segnale 1 se "esclusivamente" l'ingresso E 0.1 OR l'ingresso E 0.2 hanno lo stato di segnale 1.



## 1.6 Inserimento di un ingresso binario

### Simbolo

<operando>



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, T, Z, D, L	L'operando indica il bit di cui viene interrogato lo stato di segnale.

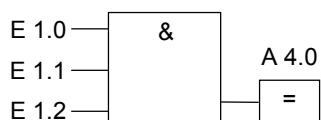
### Descrizione

L'operazione **Ingresso binario** inserisce un box AND, OR o XOR dopo l'indicazione di un ulteriore ingresso binario.

### Parola di stato

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

### Esempio



L'uscita A 4.0 ha lo stato di segnale 1 quando E 1.0 AND E 1.1 AND E 1.2 hanno lo stato di segnale 1.

## 1.7 Negazione di un ingresso binario

### Simbolo



### Descrizione

L'operazione **Nega ingresso binario** nega l'RLC.

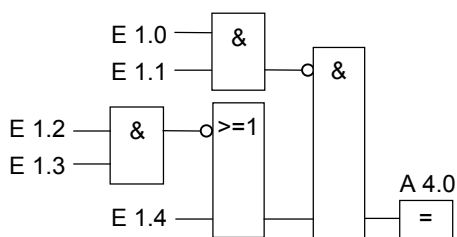
Nella negazione del risultato logico combinatorio è necessario tener conto di quanto segue:

- Se il risultato logico combinatorio viene negato nel primo ingresso di un box AND o OR, non vengono messe le parentesi.
- Se il risultato logico combinatorio non viene negato nel primo ingresso di un box OR, l'intera combinazione binaria viene inserita nella combinazione OR prima dell'ingresso.
- Se il risultato logico combinatorio non viene negato nel primo ingresso di un box AND, l'intera combinazione binaria viene inserita nella combinazione AND prima dell'ingresso.

### Parola di stato

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

### Esempio

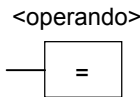


L'uscita A 4.0 è 1 quando:

- lo stato di segnale di E 1.0 AND E 1.1 NON è 1.
- AND lo stato di segnale di E 1.2 AND E 1.3 NON è 1
- OR lo stato di segnale di E 1.4 NON è 1.

## 1.8 =: Assegnazione

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, D, L	L'operando indica il bit al quale è assegnato lo stato di segnale della stringa logica.

### Descrizione

L'operazione **Assegnazione** fornisce il risultato logico combinatorio. Il box alla fine della combinazione trasmette il segnale 1 o 0 in base ai seguenti criteri:

- L'uscita trasmette il segnale 1 se le condizioni della combinazione logica vengono soddisfatte prima del box di uscita.
- L'uscita trasmette il segnale 0 se le condizioni della combinazione logica non vengono soddisfatte prima del box di uscita.

La combinazione logica FUP assegna lo stato di segnale all'uscita indirizzata dall'operazione (ciò è equivalente all'assegnazione dello stato di segnale del bit RLC all'operando). Se le condizioni delle combinazioni logiche FUP vengono soddisfatte lo stato di segnale del box di uscita è 1; altrimenti è 0.

L'operazione **Assegnazione** è influenzata dall'MCR (Master Control Relay).

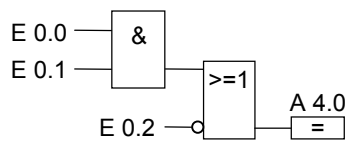
Si può posizionare il box **Assegnazione** soltanto all'estremità destra di una stringa logica. È tuttavia possibile utilizzare più box.

Per creare un'assegnazione negata utilizzare l'operazione **Nega ingresso**.

### Parola di stato

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

### Esempio



Lo stato di segnale dell'uscita A 4.0 è 1 quando:

- le uscite E 0.0 AND E 0.1 hanno lo stato di segnale 1.
- OR E 0.2 = 0.

## 1.9 #: Connettore

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, D, *L	L'operando indica il bit al quale è assegnato l'RLC.

\* Un operando nello stack di dati locali può essere utilizzato solo se è stato dichiarato nella tabella di dichiarazione delle variabili nella zona TEMP di un blocco codice (FC, FB, OB).

### Descrizione

L'operazione **Connettore** è un elemento di assegnazione intermedia che memorizza l'RLC. Tale elemento di assegnazione intermedia salva l'operazione logica combinatoria a bit dell'ultima diramazione aperta, prima dell'elemento di assegnazione.

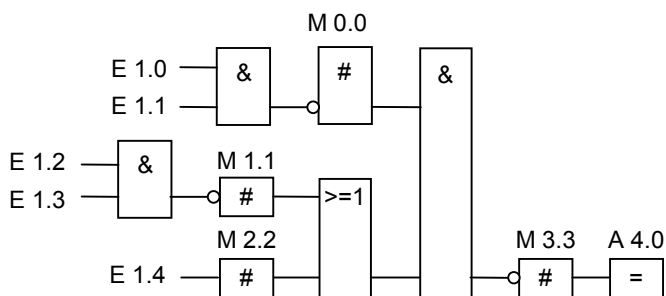
L'operazione **Connettore** è influenzata dall'MCR (Master Control Relay).

Per creare un connettore negato, si deve negare il relativo ingresso.

### Parola di stato

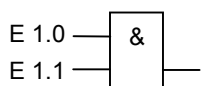
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	0	X	-	1

**Esempio**

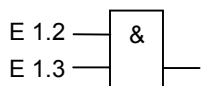


I connettori memorizzano i seguenti risultati logici combinatori:

M 0.0 memorizza l'RLC negato di



M 1.1 negato di l'RLC di

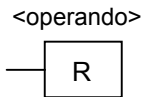


M 2.2 l'RLC di E1.4

M 3.3 l'RLC negato di operazione logica combinatoria.

## 1.10 R: Resettaggio della bobina

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL TIMER COUNTER	E, A, M, T, Z, D, L	L'operando indica il bit da resettare.

### Descrizione

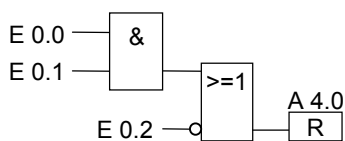
L'operazione **Resetta bobina** viene eseguita soltanto se l'RLC = 1. Se l'RLC = 1, questa operazione resetta a 0 l'operando specificato. Se l'RLC = 0, l'operazione non ha alcun effetto sull'operando specificato. L'operando rimane inalterato.

L'operazione **Resetta bobina** è influenzata dall'MCR (Master Control Relay).

### Parola di stato

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

### Esempio



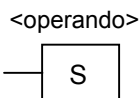
Lo stato di segnale della bobina A 4.0 viene resettato a 0 quando:

- le uscite E 0.0 AND E 0.1 hanno lo stato di segnale 1
- OR lo stato di segnale dell'uscita E 0.2 = 0.

Se l'RLC della diramazione = 0 lo stato di segnale di A 4.0 non viene modificato.

## 1.11 S: Impostazione della bobina

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, D, L	L'operando indica il bit da impostare.

### Descrizione

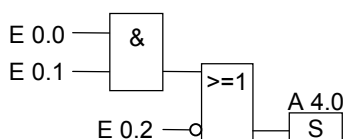
L'operazione **Imposta bobina** viene eseguita soltanto se l'RLC = 1. Se l'RLC = 1, questa operazione imposta a 1 l'operando specificato. Se l'RLC è 0, l'operazione non ha alcun effetto sull'operando specificato. L'operando rimane inalterato.

L'operazione **Imposta bobina** è influenzata dall'MCR (Master Control Relay).

### Parola di stato

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

### Esempio



Lo stato di segnale della bobina A 4.0 viene impostato su 1 quando:

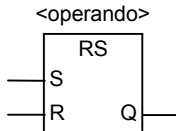
- le uscite E 0.0 AND E 0.1 hanno lo stato di segnale 1
- OR lo stato di segnale dell'uscita E 0.2 = 0.

Se l'RLC della diramazione = 0 lo stato di segnale di A 4.0 non viene modificato.



## 1.12 RS: Resettaggio impostazione flip flop

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, D, L	L'operando indica il bit da impostare o resettare.
S	BOOL	E, A, M, D, L, T, Z	Operazione Resetta abilitata
R	BOOL	E, A, M, D, L, T, Z	Operazione Imposta abilitata
Q	BOOL	E, A, M, D, L	Stato di segnale dell'<Operando>

### Descrizione

L'operazione **Resetta imposta flip flop** esegue le operazioni Imposta (S) e Resetta (R) soltanto quando l'RLC è 1. Un RLC uguale a 0 non influenza tali operazioni; l'operando specificato nell'operazione rimane immutato.

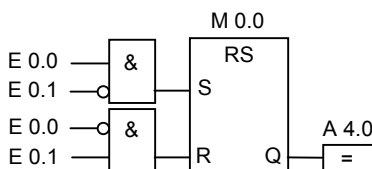
Un flip flop viene resettato se lo stato di segnale è 1 all'ingresso R, e 0 all'ingresso S. Contrariamente, se lo stato di segnale è 0 all'ingresso R, ed 1 all'ingresso S, il flip flop viene impostato. Se l'RLC è 1 presso entrambi gli ingressi, il flip flop viene impostato.

L'operazione **Resetta imposta flip flop** è influenzata dall'MCR (Master Control Relay).

### Parola di stato

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

### Esempio



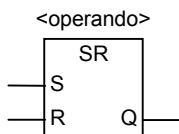
Se E 0.0 = 1 e E 0.1 = 0, il merker M 0.0 viene resettato e l'uscita A 4.0 è 0.

Se E 0.0 = 0 e E 0.1 = 1, il merker M 0.0 viene impostato e l'uscita A 4.0 è 1.

Se entrambi i segnali di stato sono 0, non viene effettuata alcuna modifica. Se tali segnali valgono 1, in base all'ordine ha la precedenza l'operazione Imposta. M 0.0 viene resettato e A 4.0 è 1.

## 1.13 SR: Impostazione resettaggio flip flop

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	E, A, M, D, L	L'operando indica il bit da impostare o resettare.
S	BOOL	E, A, M, D, L, T, Z	Operazione Imposta abilitata
R	BOOL	E, A, M, D, L, T, Z	Operazione Resetta abilitata
Q	BOOL	E, A, M, D, L	Stato di segnale dell'<Operando>

### Descrizione

L'operazione **Imposta resetta flip flop** esegue le operazioni Imposta (S) e Resetta (R), soltanto quando l'RLC è 1. Un RLC uguale a 0 non influenza tali operazioni; l'operando specificato nella operazione rimane immutato.

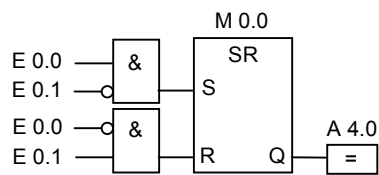
Un flip flop viene impostato se lo stato di segnale è 1 all'ingresso S, e 0 all'ingresso R. Contrariamente, se lo stato di segnale è 0 all'ingresso S, e 1 all'ingresso R, il flip flop viene resettato. Se l'RLC è 1 presso entrambi gli ingressi, il flip flop viene resettato.

L'operazione **Imposta resetta flip flop** è influenzata dall'MCR (Master Control Relay).

### Parola di stato

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

### Esempio



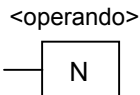
Se  $E\ 0.0 = 1$  e  $E\ 0.1 = 0$ , il merker M 0.0 viene impostato e A 4.0 è 1.

Se  $E\ 0.0 = 0$  e  $E\ 0.1 = 1$ , il merker M 0.0 viene resettato e A 4.0 è 0.

Se entrambi i segnali di stato sono 0, non viene effettuata alcuna modifica. Se tali segnali valgono 1, in base all'ordine ha la precedenza l'operazione Imposta. M 0.0 viene resettato e A 4.0 è 0.

## 1.14 N: Rilevamento del fronte RLC di discesa

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	A, M, D, L	L'operando indica il merker di fronte che memorizza il precedente RLC.

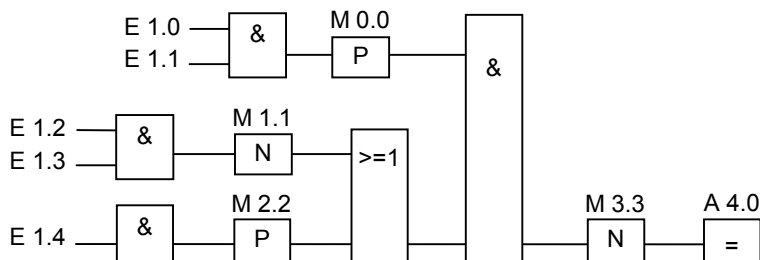
### Descrizione

L'operazione **Rileva fronte RLC di discesa** riconosce un cambiamento di segnale da 1 a 0 (fronte di discesa) nell'operando specificato; dopo l'operazione, tale cambiamento viene segnalato dall'RLC = 1. L'attuale stato di segnale nell'RLC viene messo a confronto con quello dell'operando, cioè il merker di fronte. Se lo stato di segnale dell'operando è 0 e l'RLC prima dell'operazione è 1, l'RLC dopo l'operazione è 1 (impulso), altrimenti 0. L'RLC prima dell'operazione viene salvato nell'operando.

### Parola di stato

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

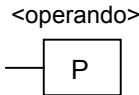
### Esempio



Il merker di fronte M 3.3 memorizza lo stato di segnale del precedente RLC.

## 1.15 P: Rilevamento del fronte RLC di salita

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando>	BOOL	A, M, D, L	L'operando indica il merker di fronte che memorizza il precedente RLC.

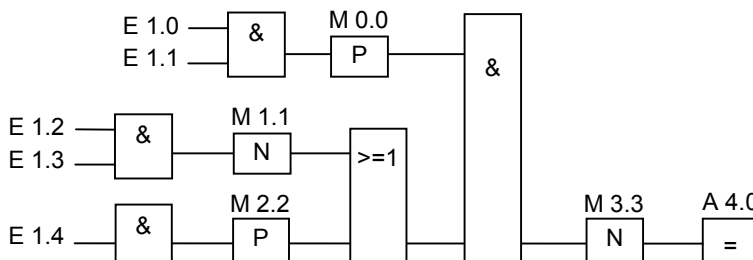
### Descrizione

L'operazione **Rileva fronte RLC di salita** riconosce un cambiamento di segnale da 0 a 1 (fronte di salita) nell'operando specificato; dopo l'operazione, tale cambiamento viene segnalato dall'RLC = 1. L'attuale stato di segnale nell'RLC viene messo a confronto con quello dell'operando, cioè il merker di fronte. Se lo stato di segnale dell'operando è 0 e l'RLC prima dell'operazione è 1, l'RLC dopo l'operazione è 1 (impulso), altrimenti 0. L'RLC prima dell'operazione viene salvato nell'operando.

### Parola di stato

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

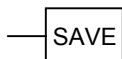
### Esempio



Il merker di fronte M 3.3 memorizza lo stato di segnale del precedente RLC.

## 1.16 SAVE: Memorizzazione dell'RLC nel registro BIE

### Simbolo



### Descrizione

L'operazione **Salva RLC nel registro BIE**, memorizza il risultato logico combinatorio (RLC) nel bit BIE della parola di stato. Il bit della prima interrogazione \ER non viene resettato.

Per questo motivo, nella combinazione logica **AND**, viene combinato nel segmento successivo anche lo stato del bit BIE.

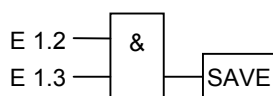
Per quanto concerne l'operazione **SAVE** (KOP, FUP, AWL), leggere le regole seguenti, ignorando la descrizione riportata nei manuali e nelle Guide online.

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 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

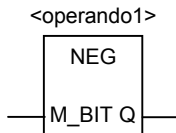


Il risultato logico combinatorio (RLC) viene memorizzato nel bit BIE.

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

## 1.17 NEG: Interrogazione del rilevamento di fronte di discesa

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando1>	BOOL	E, A, M, D, L	Segnale da interrogare per una transizione fronte di discesa
M_BIT	BOOL	A, M, D	L'operando M_BIT indica il merker di fronte che memorizza il precedente stato di segnale di NEG. Utilizzare l'area di memoria dell'immagine di processo degli ingressi (E) per l'M_BIT solo se l'operando non è occupato da alcuna unità di ingresso.
Q	BOOL	E, A, M, D, L	Uscita dell'unico cambio di segnale.

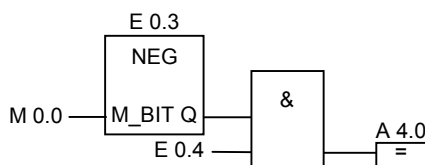
### Descrizione

L'operazione **Interroga rilevamento di fronte di discesa** confronta lo stato di segnale dell'<Operando1> con lo stato di segnale della precedente interrogazione dello stato di segnale memorizzata nel parametro M\_BIT. Se si presenta un cambiamento da 1 a 0, l'uscita Q è 1, in tutti gli altri casi 0.

### Parola di stato

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

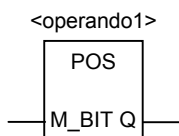
### Esempio



L'uscita A 4.0 è 1 quando:  
 l'ingresso E 3.0 ha un fronte di discesa AND  
 l'ingresso E 0.4 ha lo stato di segnale 1.

## 1.18 POS: Interrogazione del rilevamento del fronte RLC di salita

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
<Operando1>	BOOL	E, A, M, D, L	Segnale da interrogare per transizione fronte di salita.
M_BIT	BOOL	A, M, D	L'operando M_BIT indica il merker di fronte che memorizza il precedente stato di segnale di POS. Utilizzare l'area di memoria dell'immagine di processo degli ingressi E per l'M_BIT solo se tale operando non è occupato da alcuna unità di ingresso.
Q	BOOL	E, A, M, D, L	Uscita dell'unico cambio di segnale.

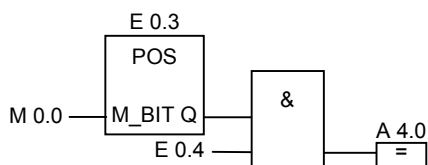
### Descrizione

L'operazione **Interroga rilevamento di fronte di salita** confronta lo stato di segnale dell'<Operando1> con lo stato di segnale della precedente interrogazione dello stato di segnale memorizzata nel parametro M\_BIT. Se si presenta un cambiamento da 0 a 1, l'uscita Q è 1, in tutti gli altri casi 0.

### Parola di stato

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

### Esempio



L'uscita A 4.0 è 1 quando:  
 l'ingresso E 3.0 ha un fronte di salita  
 AND l'ingresso E 0.4 ha lo stato di segnale 1.



## 2 Operazioni di confronto

### 2.1 Sommario delle operazioni di confronto

#### Descrizione

Le operazioni di confronto confrontano gli ingressi IN1 e IN2 secondo 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

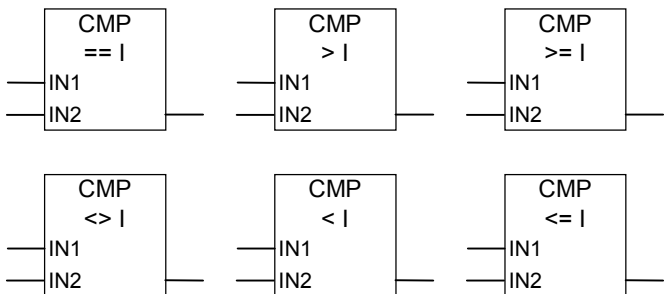
Se il confronto è vero, il risultato logico combinatorio (RLC) dell'operazione è 1, altrimenti 0. Non vi è negazione del risultato di confronto, in quanto questa modalità logica può essere trattata anche dalla funzione di confronto inversa.

Sono disponibili le seguenti operazioni di confronto:

- CMP ? I Confronto di numeri interi (16 bit)
- CMP ? D Confronto di numeri interi (32 bit)
- CMP ? R Confronto di numeri in virgola mobile

## 2.2 CMP ? I: Confronto di numeri interi (16 bit)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
IN1	INT	E, A, M, D, L o costante	Primo valore da confrontare
IN2	INT	E, A, M, D, L o costante	Secondo valore da confrontare
Uscita box	BOOL	E, A, M, D, L	Risultato del confronto

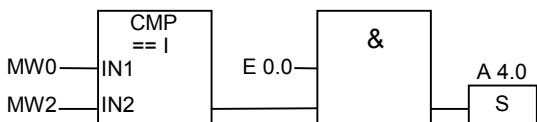
### Descrizione

L'operazione **Confronta numeri interi a 16 bit** esegue un'operazione di confronto sulla base di numeri interi a 16 bit. Tale operazione confronta gli ingressi IN1 e IN2, a seconda del tipo di confronto che l'utente ha scelto dal box di selezione.

### Parola di stato

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

### Esempio

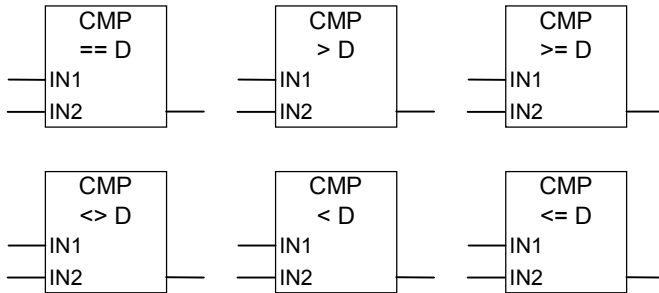


A 4.0 viene impostata se:

- MW0 = MW2
- AND l'ingresso E 0.0 ha lo stato di segnale 1.

## 2.3 CMP ? D: Confronto di numeri interi (32 bit)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
IN1	DINT	E, A, M, D, L o costante	Primo valore da confrontare
IN2	DINT	E, A, M, D, L o costante	Secondo valore da confrontare
Uscita box	BOOL	E, A, M, D, L	Risultato del confronto

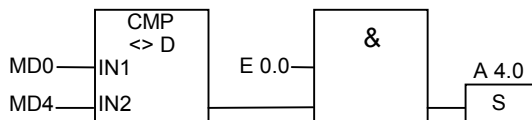
### Descrizione

L'operazione **Confronta numeri interi a 32 bit** esegue un'operazione di confronto sulla base di numeri interi a 32 bit. Tale operazione confronta gli ingressi IN1 e IN2, a seconda del tipo di confronto che l'utente ha scelto dal box di selezione.

### Parola di stato

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

### Esempio

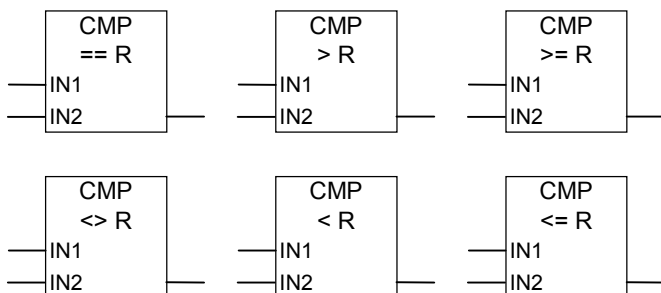


A 4.0 viene impostata se:

- MD0 diverso da MD4
- AND l'ingresso E 0.0 ha lo stato di segnale 1.

## 2.4 CMP ? R: Confronto di numeri in virgola mobile

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
IN1	REAL	E, A, M, D, L o costante	Primo valore da confrontare
IN2	REAL	E, A, M, D, L o costante	Secondo valore da confrontare
Uscita box	BOOL	E, A, M, D, L	Risultato del confronto

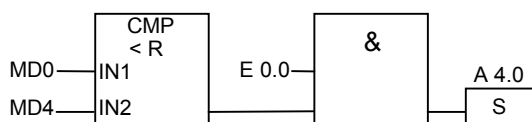
### Descrizione

L'operazione **Confronta numeri in virgola mobile** esegue un'operazione di confronto sulla base di numeri in virgola mobile. Tale operazione confronta gli ingressi IN1 e IN2, a seconda del tipo di confronto che l'utente ha scelto dal box di selezione.

### Parola di stato

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

### Esempio



A 4.0 viene impostata se:

- MD0 < MD4
- AND l'ingresso E 0.0 ha lo stato di segnale 1.

## 3 Operazioni di conversione

### 3.1 Sommario delle operazioni di conversione

#### Descrizione

Per convertire numeri decimali in codice binario (BCD) e numeri interi in altri tipi di numeri si possono adoperare le seguenti operazioni:

- BCD\_I: Conversione di un numero BCD in un numero intero (16 bit)
- I\_BCD: Conversione di un numero intero (16 bit) in un numero BCD
- BCD\_DI: Conversione di un numero BCD in un numero intero (32 bit)
- I\_DI: Conversione di un numero intero (16 bit) in un numero intero (32 bit)
- DI\_BCD: Conversione di un numero intero (32 bit) in un BCD
- DI\_R: Conversione di un numero intero (32 bit) in un numero in virgola mobile

Si possono adoperare le seguenti operazioni per formare i complementi dei numeri interi o per cambiare il segno di un numero in virgola mobile:

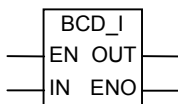
- INV\_I: Complemento a 1 di numero intero (16 bit)
- INV\_DI: Complemento a 1 di numero intero (32 bit)
- NEG\_I: Complemento a 2 di numero intero (16 bit)
- NEG\_DI: Complemento a 2 di numero intero (32 bit)
- NEG\_R: Negazione del numero in virgola mobile

Per convertire un numero IEEE 754 in virgola mobile a 32 bit in ACCU 1 in un numero intero a 32 bit (doppio numero intero) si può adoperare una delle seguenti operazioni. Le singole operazioni differiscono l'una dall'altra nel metodo di arrotondamento:

- ROUND: Arrotondamento a numero intero
- TRUNC: Arrotondamento senza resto di un numero intero
- CEIL: Generazione di un numero intero superiore da un numero in virgola mobile
- FLOOR: Generazione di un numero intero inferiore da un numero in virgola mobile

### 3.2 BCD\_I: Conversione di un numero BCD in un numero intero (16 bit)

**Simbolo**



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

**Descrizione**

L'operazione **Converti un numero BCD in un numero intero (16 bit)** legge il contenuto specificato nel parametro di ingresso IN sotto forma di numero decimale a tre cifre in formato di cifra decimale in codice binario (BCD, ± 999) e converte tale numero in un valore di numero intero. Il parametro di uscita OUT fornisce il risultato.

ENO e EN hanno sempre il medesimo stato di segnale.

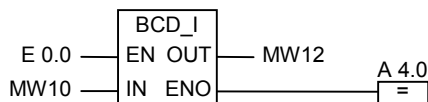
Se la posizione di un BCD si trova nel campo errato compreso tra 10 e 15, si verifica un errore BCDF durante un tentativo di conversione:

- La CPU si pone in modalità STOP. Viene inserito nel buffer di diagnostica "Errore di conversione BCD" con il numero ID di evento 2521.
- Se è programmato OB121, esso viene richiamato.

**Parola di stato**

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

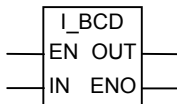
**Esempio**



La conversione viene eseguita se E 0.0 = 1. Il contenuto della parola di merker MW10 viene letto come numero BCD di tre cifre e convertito in numero intero (a 16 bit). Il risultato viene memorizzato in MW12. Dopo la conversione A 4.0 = 1 (ENO = EN).

### 3.3 I\_BCD: Conversione di un numero intero (16 bit) in un numero BCD

#### Simbolo



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

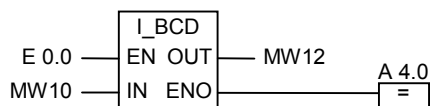
#### Descrizione

L'operazione **Converti numero intero (16 bit) in numero BCD** legge il contenuto specificato nel parametro di ingresso IN sotto forma di numero intero (a 16 bit) e lo converte in un numero a tre cifre in formato di cifra decimale in codice binario (BCD,  $\pm 999$ ). Il parametro di uscita OUT fornisce il risultato. Se si verifica un overflow, ENO corrisponde a 0.

#### Parola di stato

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

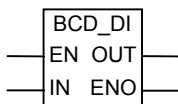
#### Esempio



La conversione viene eseguita se  $E\ 0.0 = 1$ . Il contenuto della parola di merker MW10 viene copiato sotto forma di numero intero (a 16 bit) e convertito in un numero a tre cifre in formato BCD. Il risultato viene memorizzato in MW12. Se si verifica un overflow,  $A\ 4.0 = 0$ . Se lo stato di segnale dell'ingresso  $EN = 0$  (ovvero la conversione non viene eseguita), sarà 0 anche lo stato di segnale dell'uscita  $A\ 4.0$ .

### 3.4 BCD\_DI: Conversione di un numero BCD in un numero intero (32 bit)

#### Simbolo



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

#### Descrizione

L'operazione **Converti un BCD in un numero intero (32 bit)** legge il contenuto specificato nel parametro di ingresso IN sotto forma di numero a sette cifre con formato di cifra decimale in codice binario (BCD, ± 9 999 999) e converte tale numero in numero intero (a 32 bit). Il parametro di uscita OUT fornisce il risultato.

ENO e EN hanno sempre il medesimo stato di segnale.

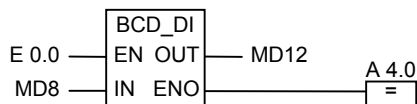
Se la posizione di un BCD si trova nel campo errato compreso tra 10 e 15, si verifica un errore BCDF durante un tentativo di conversione:

- La CPU va in modalità STOP. Viene inserito nel buffer di diagnostica "Errore di conversione BCD" con il numero di evento 2521.
- Se è programmato OB121, esso viene richiamato.

#### Parola di stato

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

#### Esempio

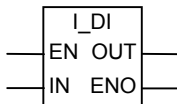


La conversione viene eseguita se E 0.0 = 1. Il contenuto della doppia parola di merker MD8 viene letto sotto forma di numero BCD a sette cifre con formato di cifra decimale in codice binario e convertito in numero intero (a 32 bit). Il risultato viene memorizzato in MD12. Dopo la conversione A 4.0 = 1 (ENO = EN).



### 3.5 I\_DI: Conversione di un numero intero (16 bit) in un numero intero (32 bit)

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	INT	E, A, M, D, L o costante	Valore da convertire
OUT	DINT	E, A, M, D, L	Risultato
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

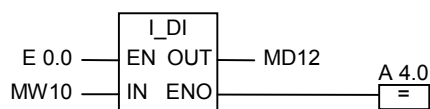
#### Descrizione

L'operazione **Converti numero intero (16 bit) in numero intero (32 bit)** legge il contenuto specificato nel parametro di ingresso IN sotto forma di numero intero (a 16 bit) e lo converte in un numero intero (a 32 bit). Il parametro di uscita OUT fornisce il risultato. ENO e EN hanno sempre il medesimo stato di segnale.

#### Parola di stato

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

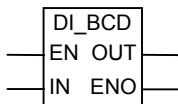
#### Esempio



La conversione viene eseguita se  $E\ 0.0 = 1$ . Il contenuto della parola di merker MW10 viene copiato sotto forma di numero intero (a 16 bit) e convertito in un numero intero (a 32 bit). Il risultato viene memorizzato in M12. Se l'operazione viene eseguita  $A\ 4.0 = 1$  ( $ENO = EN$ ).

### 3.6 DI\_BCD: Conversione di un numero intero (32 bit) in un BCD

#### Simbolo



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

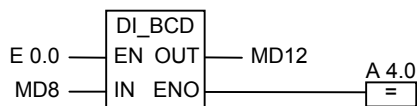
#### Descrizione

L'operazione **Converti un numero intero (32 bit) in un BCD** legge il contenuto specificato nel parametro di ingresso IN come un valore di numero intero, e converte questo valore in un numero a sette cifre dal formato di cifra decimale in codice binario (BCD, ± 9 999 999). Il parametro di uscita OUT fornisce il risultato. Se si verifica un overflow, ENO è uguale a 0.

#### Parola di stato

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

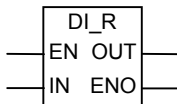
#### Esempio



La conversione viene eseguita se E 0.0 = 1. Il contenuto della doppia parola di merker MD8 viene copiato sotto forma di numero intero (a 32 bit) e convertito in un numero a sette cifre in formato BCD. Il risultato viene memorizzato in MD12. Se si verifica un overflow, A 4.0 = 0. Se lo stato di segnale dell'ingresso EN = 0 (ovvero la conversione non viene eseguita), sarà 0 anche lo stato di segnale dell'uscita A 4.0.

### 3.7 DI\_R: Conversione di un numero intero (32 bit) in un numero in virgola mobile

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	DINT	E, A, M, D, L o costante	Valore da convertire
OUT	REAL	E, A, M, D, L	Risultato
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

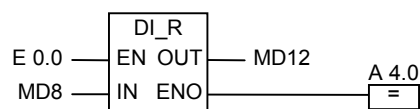
#### Descrizione

L'operazione **Converti un numero intero (32 bit) in un numero in virgola mobile** legge il contenuto specificato nel parametro di ingresso IN come un valore di numero intero (a 32 bit), e converte questo valore in un numero in virgola mobile. Il parametro di uscita OUT fornisce il risultato. ENO e EN hanno sempre il medesimo stato di segnale.

#### Parola di stato

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

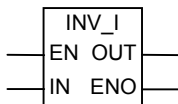
#### Esempio



La conversione viene eseguita se  $E\ 0.0 = 1$ . Il contenuto della doppia parola di merker MD8 viene letto sotto forma di numero intero (a 32 bit) e convertito in un numero in virgola mobile. Il risultato dell'operazione viene memorizzato in MD12. Se l'operazione non viene eseguita  $A\ 4.0 = 0$  ( $ENO = EN$ ).

### 3.8 INV\_I: Complemento a 1 di numero intero (16 bit)

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	INT	E, A, M, D, L o costante	Valore di ingresso
OUT	INT	E, A, M, D, L	Complemento a 1 di numero intero (a 16 bit)
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

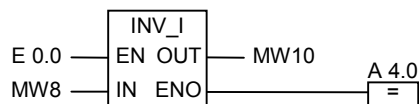
#### Descrizione

L'operazione **Complemento a 1 di numero intero (16 bit)** legge il contenuto specificato nel parametro di ingresso IN, ed esegue l'operazione logica combinatoria booleana OR esclusivo parola, mascherata da FFFFH, in modo tale che ogni bit viene convertito nel suo valore opposto. Il parametro di uscita OUT fornisce il risultato. ENO e EN hanno sempre il medesimo stato di segnale.

#### Parola di stato

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

#### Esempio



La conversione viene eseguita se E 0.0 = 1. Tutti i bit di MW8 vengono invertiti:

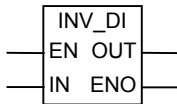
MW8 = 01000001 10000001 →

MW10 = 10111110 01111110

La conversione non viene eseguita se E 0.0 = 0 e A 4.0 = 0 (ENO = EN).

### 3.9 INV\_DI: Complemento a 1 di numero intero (32 bit)

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	DINT	E, A, M, D, L o costante	Valore di ingresso
OUT	DINT	E, A, M, D, L	Complemento a 1 di numero intero (a 32 bit)
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

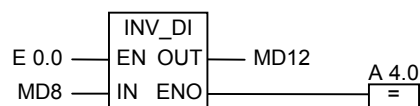
#### Descrizione

L'operazione **Complemento a 1 di numero intero (32 bit)** legge il contenuto specificato nel parametro di ingresso IN ed esegue l'operazione logica combinatoria a parola Combina OR esclusivo con la maschera esadecimale FFFF FFFFH. In questo modo vengono invertiti i valori di tutti i bit. Il parametro di uscita OUT fornisce il risultato. ENO e EN hanno sempre il medesimo stato di segnale.

#### Parola di stato

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

#### Esempio



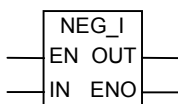
La conversione viene eseguita se E 0.0 = 1. Tutti i bit della doppia parola di merker MD8 vengono invertiti:

MD8 = F0FF FFF0 → MD12 = 0F00 000F

Se la conversione non viene eseguita A 4.0 = 0 (ENO = EN).

### 3.10 NEG\_I: Complemento a 2 di numero intero (16 bit)

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	INT	E, A, M, D, L o costante	Valore di ingresso
OUT	INT	E, A, M, D, L	Complemento a 2 di numero intero (a 16 bit)
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

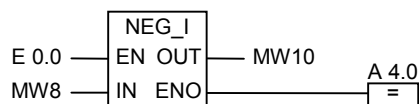
#### Descrizione

L'operazione **Complemento a 2 di numero intero (16 bit)** legge il contenuto specificato nel parametro di ingresso IN, e cambia il segno (per esempio, da un valore positivo ad un valore negativo). Il parametro di uscita OUT fornisce il risultato. Se lo stato del segnale di EN è 0, lo stato del segnale di ENO è 0. Se lo stato del segnale EN è 1 e si verifica un overflow, lo stato del segnale di ENO è 0.

#### Parola di stato

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

#### Esempio



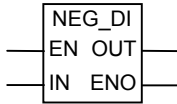
La conversione viene eseguita se E 0.0 = 1. Il contenuto della parola di merker MW8 viene emesso con il segno opposto nel parametro OUT di MW10.

Ad esempio: MW8 = +10 → MW10 = - 10

Se EN = 1 e si verifica un overflow, ENO = 0 e lo stato di segnale di A 4.0 è 0. Se la conversione non viene eseguita A 4.0 = 0 (ENO = EN).

### 3.11 NEG\_DI: Complemento a 2 di numero intero (32 bit)

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	DINT	E, A, M, D, L o costante	Valore di ingresso
OUT	DINT	E, A, M, D, L	Complemento a 2 di numero intero (a 32 bit)
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

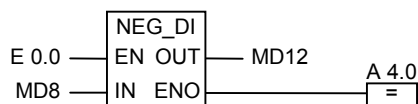
#### Descrizione

L'operazione **Complemento a 2 di numero intero (32 bit)** legge il contenuto specificato nel parametro di ingresso IN, e cambia il segno (per esempio, da un valore positivo ad un valore negativo). Il parametro di uscita OUT fornisce il risultato. Se lo stato del segnale di EN è 0, lo stato del segnale di ENO è 0. Se lo stato del segnale di EN è 1 e si verifica un overflow, lo stato del segnale di ENO è 0.

#### Parola di stato

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

#### Esempio



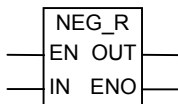
La conversione viene eseguita se E 0.0 = 1. Il contenuto della doppia parola di merker MD8 viene emesso con il segno opposto nel parametro OUT di MD12. Ad esempio:

MD8 = + 60.000 → MD12 = - 60.000.

Se EN = 1 e si verifica un overflow, ENO = 0 e lo stato di segnale di A 4.0 è 0. Se la conversione non viene eseguita A 4.0 = 0 (ENO = EN).

### 3.12 NEG\_R: Negazione del numero in virgola mobile

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Valore di ingresso
OUT	REAL	E, A, M, D, L	Il risultato è la forma negata del valore di ingresso.
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

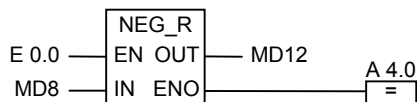
#### Descrizione

L'operazione **Nega numero in virgola mobile** legge il contenuto specificato nel parametro di ingresso IN e inverte il bit di segno, l'operazione cioè modifica il segno del numero (ad esempio da 0 per il segno più a 1 per il segno meno). I bit dell'esponente e della mantissa rimangono invariati. Il parametro di uscita OUT fornisce il risultato. ENO e EN hanno sempre lo stesso stato del segnale con la seguente eccezione: se lo stato di segnale di EN è uguale a 1 e si verifica un overflow, lo stato di segnale di ENO diventa 0.

#### Parola di stato

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

#### Esempio



La conversione viene eseguita se E 0.0 = 1. Il contenuto della doppia parola di merker MD8 viene emesso con il segno opposto nel parametro OUT di MD12. Ad esempio:

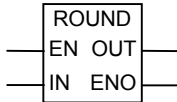
MD8 = + 6,234 → MD12 = - 6,234

Se la conversione non viene eseguita A 4.0 = 0 (ENO = EN).



### 3.13 ROUND: Arrotondamento a numero intero

#### Simbolo



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

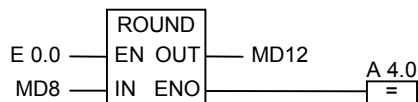
#### Descrizione

L'operazione **Arrotonda a numero intero (32 bit)** legge il contenuto specificato nel parametro di ingresso IN come un numero in virgola mobile e converte questo numero in un numero intero (a 32 bit). Il risultato corrisponde al successivo numero intero che viene emesso dal parametro di uscita OUT. Se la parte interrotta = x,54, il numero pari viene arrotondato per difetto (ad es.: 2,5 -> 2, 1,5 -> 2). Se si verifica un overflow, ENO = 0. Se l'ingresso non è un numero in virgola mobile, i bit OV e OS hanno il valore 1 e ENO il valore 0.

#### Parola di stato

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

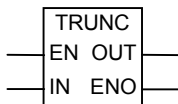
#### Esempio



La conversione viene eseguita se E 0.0 = 1. Il contenuto della doppia parola di merker MD8 viene letto come numero in virgola mobile e convertito in un numero intero (a 32 bit) secondo il principio "round to nearest". Il risultato dell'operazione viene memorizzato in MD12. Se si verifica un overflow, A 4.0 = 0. Se lo stato di segnale dell'ingresso EN = 0 (ovvero la conversione non viene eseguita), sarà 0 anche lo stato di segnale dell'uscita A 4.0.

### 3.14 TRUNC: Arrotondamento senza resto di un numero intero

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Valore da arrotondare
OUT	DINT	E, A, M, D, L	Parte del numero intero del valore IN
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

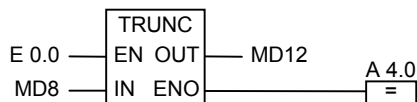
#### Descrizione

L'operazione **Arrotonda senza resto di un numero intero (32 bit)** legge il contenuto specificato nel parametro di ingresso IN come un numero in virgola mobile e converte questo numero in un numero intero (a 32 bit) (ad es. 1,5 diventa 1). Il risultato corrisponde alla parte intera del numero in virgola mobile che viene emessa nel parametro di uscita OUT. Se si verifica un overflow, ENO corrisponde a 0. Se l'ingresso non è un numero in virgola mobile, i bit OV e OS hanno il valore 1 e ENO il valore 0. Se l'ingresso non è un numero in virgola mobile, i bit OV e OS hanno il valore 1 e ENO il valore 0.

#### Parola di stato

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

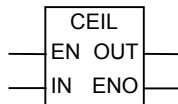
#### Esempio



La conversione viene eseguita se E 0.0 = 1. Il contenuto della doppia parola di merker MD8 viene letto come numero in virgola mobile e convertito in un numero intero (a 32 bit) secondo il principio "round to zero". Il risultato corrisponde alla parte intera del numero in virgola mobile che viene memorizzata in MD12. Se si verifica un overflow, A 4.0 = 0. Se lo stato di segnale dell'ingresso EN = 0 (ovvero la conversione non viene eseguita), sarà 0 anche lo stato di segnale dell'uscita A 4.0.

### 3.15 CEIL: Generazione di un numero intero superiore da un numero in virgola mobile

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Valore da convertire
OUT	DINT	E, A, M, D, L	Risultato
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

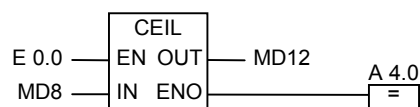
#### Descrizione

L'operazione di conversione **Genera da un numero in virgola mobile un numero intero superiore** legge il contenuto specificato nel parametro di ingresso IN come un numero in virgola mobile e lo converte in un numero intero (a 32 bit) (ad es.: +1,2 -> +2; -1,5 -> -1). Il risultato corrisponde al più piccolo numero intero maggiore o uguale al numero in virgola mobile specificato. Il parametro di uscita OUT fornisce il risultato. Se si verifica un overflow, ENO corrisponde a 0. Se l'ingresso non è un numero in virgola mobile, i bit OV e OS hanno il valore 1 e ENO il valore 0.

#### Parola di stato

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

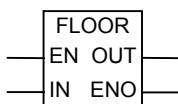
#### Esempio



La conversione viene eseguita se  $E\ 0.0 = 1$ . Il contenuto della doppia parola di merker MD8 viene letto come numero in virgola mobile e convertito in un numero intero (a 32 bit) secondo il principio "round to + infinity". Il risultato dell'operazione viene memorizzato in MD12. Se si verifica un overflow,  $A\ 4.0 = 0$ . Se lo stato di segnale dell'ingresso  $EN = 0$  (ovvero la conversione non viene eseguita), sarà 0 anche lo stato di segnale dell'uscita  $A\ 4.0$ .

### 3.16 FLOOR: Generazione di un numero intero inferiore da un numero in virgola mobile

**Simbolo**



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Valore da convertire
OUT	DINT	E, A, M, D, L	Risultato
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

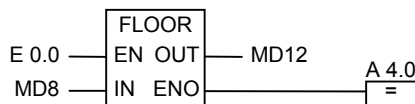
**Descrizione**

L'operazione di conversione **Genera da un numero in virgola mobile un numero intero inferiore** legge il contenuto specificato nel parametro di ingresso IN come un numero in virgola mobile, e converte questo numero in un numero intero (a 32 bit) (ad es.: +1,5 -> +1; -1,5 -> -2). Il risultato corrisponde al più piccolo numero intero minore o uguale al numero in virgola mobile specificato. Il parametro di uscita OUT fornisce il risultato. Se si verifica un overflow, ENO corrisponde a 0. Se l'ingresso non è un numero in virgola mobile, i bit OV e OS hanno il valore 1 e ENO il valore 0.

**Parola di stato**

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

**Esempio**



La conversione viene eseguita se E 0.0 = 1. Il contenuto della doppia parola di merker MD8 viene letto come numero in virgola mobile e convertito in un numero intero (a 32 bit) secondo il principio "round to - infinity". Il risultato dell'operazione viene memorizzato in MD12. Se si verifica un overflow, A 4.0 = 0. Se lo stato di segnale dell'ingresso EN = 0 (ovvero la conversione non viene eseguita), sarà 0 anche lo stato di segnale dell'uscita A 4.0.

## 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 FUP 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:

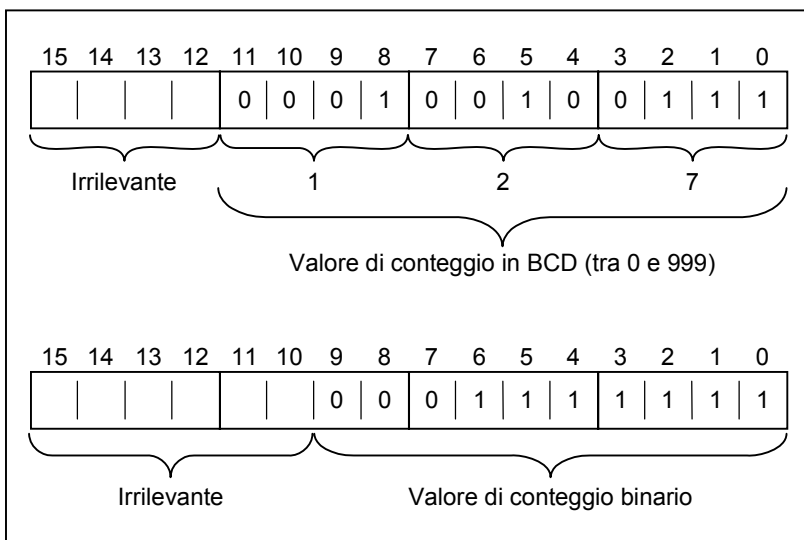
- ZAEHLER: Parametrizzazione e conteggio in avanti / all'indietro
- Z\_VORW: Parametrizzazione e conteggio in avanti
- Z\_RUECK: Parametrizzazione e conteggio all'indietro
- SZ: Impostazione del valore di conteggio
- ZV: Conta in avanti
- ZR: Conta all'indietro

### 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.

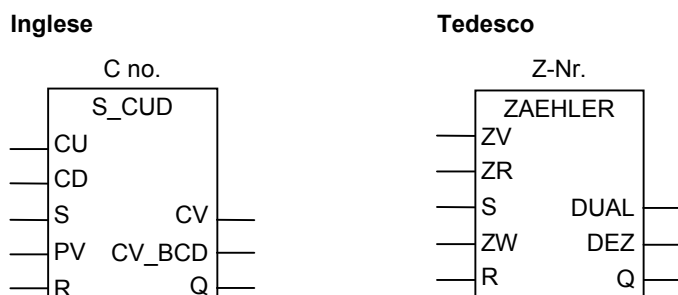
I bit da 0 a 11 del contatore contengono il valore di conteggio nel formato decimale in codice binario (formato BCD: ogni set di quattro bit contiene il codice binario per un valore decimale).

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 ZAEHLER: Parametrizzazione e conteggio in avanti / all'indietro

### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
C n°	Z Nr.	COUNTER	Z	Numero di identificazione del contatore. Il campo dipende dalla CPU
CU	ZV	BOOL	E, A, M, D, L, T, Z	Ingresso ZV: conteggio in avanti
CD	ZR	BOOL	E, A, M, D, L	Ingresso ZR: conteggio all'indietro
S	S	BOOL	E, A, M, D, L	Ingresso per preimpostare il contatore
PV	ZW	WORD	E, A, M, D, L o costante	Valore compreso tra 0 e 999 oppure valore di conteggio specificato come C#<Valore> in formato BCD
R	R	BOOL	E, A, M, D, L, T, Z	Ingresso di resettaggio
CV	DUAL	WORD	E, A, M, D, L	Valore del conteggio attuale, in codice esadecimale
CV_BCD	DEZ	WORD	E, A, M, D, L	Valore del conteggio attuale, codificato in formato BCD
Q	Q	BOOL	E, A, M, D, L	Stato del contatore

### Descrizione

Il cambiamento del fronte da 0 a 1 all'ingresso S dell'operazione **Parametrizza e conta in avanti/all'indietro** preimposta il valore di conteggio ZW nel contatore. Il valore del contatore viene incrementato di 1 se vi è un fronte di salita nell'ingresso ZV e se il valore del contatore è inferiore a 999. Il valore del contatore viene decrementato di 1 se vi è un fronte di salita nell'ingresso ZR e se il valore del contatore è maggiore di 0. Se vi è un fronte di salita in entrambi gli ingressi di conteggio, entrambe le operazioni vengono eseguite e il valore di conteggio resta uguale.

Se viene impostato il contatore e agli ingressi ZV/ZR l'RLC è = 1, il contatore conterà una volta nel ciclo **successivo** anche se non vi era stato un cambio di fronte.

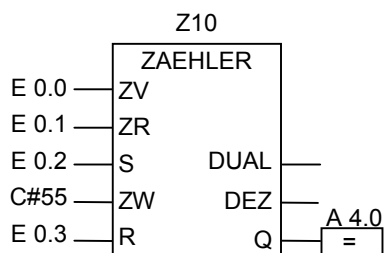
Il contatore viene resettato se all'ingresso R si trova il valore 1. Il valore di conteggio viene impostato su 0.

Un'interrogazione dello stato di segnale all'uscita Q per verificare se questo corrisponde a 1, produce un risultato di 1 se il valore di conteggio è maggiore di 0. L'interrogazione produce un risultato di 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

**Esempio**



Se lo stato di segnale dell'ingresso E 0.2 passa da 0 a 1, il temporizzatore Z10 viene impostato sul valore 55. Se lo stato di segnale di E 0.0 passa da 0 a 1, il valore del contatore Z10 viene incrementato di 1, tranne quando il suo valore è uguale a 999. Se l'ingresso E 0.1 passa da 0 a 1, il contatore Z10 viene decrementato di 1, tranne quando il suo valore è uguale a 0. Se E 0.3 passa da 0 a 1, il valore di conteggio di Z10 viene settato a 0. Lo stato di segnale dell'uscita A 4.0 è 1 a meno che Z10 non sia uguale a 0.

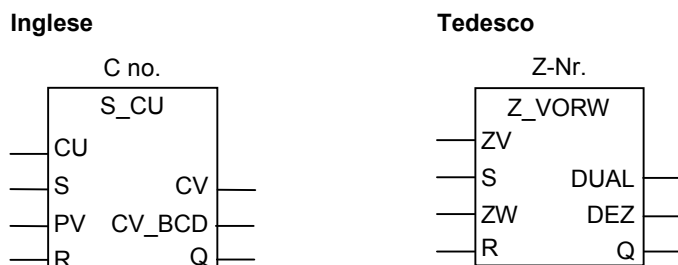
**Nota**

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



## 4.3 Z\_VORW: Parametrizzazione e conteggio in avanti

### Simbolo



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, D, L, T, Z	Ingresso ZV: conteggio in avanti
S	S	BOOL	E, A, M, D, L	Ingresso per preimpostare il contatore
PV	ZW	WORD	E, A, M, D, L	Valore compreso tra 0 e 999 oppure valore di conteggio specificato come C#<Valore> in formato BCD
R	R	BOOL	E, A, M, D, L o costante	Ingresso di resettaggio
CV	DUAL	WORD	E, A, M, D, L, T, Z	Valore del conteggio attuale, in codice esadecimale
CV_BCD	DEZ	WORD	E, A, M, D, L	Valore del conteggio attuale, codificato in formato BCD
Q	Q	BOOL	E, A, M, D, L	Stato del contatore

### Descrizione

Il cambiamento del fronte da 0 a 1 all'ingresso S dell'operazione **Parametrizza e conta in avanti** preimposta il valore di conteggio ZW nel contatore. Il valore del contatore viene incrementato di 1 se vi è un fronte di salita nell'ingresso ZV e se il valore del contatore è inferiore a 999.

Se il contatore viene impostato e all'ingresso ZV l'R/LC è = 1, il contatore conterà una volta nel ciclo **successivo**, anche se non vi era stato un cambio di fronte.

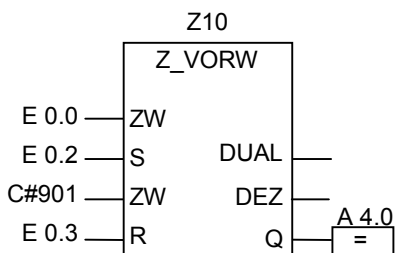
Il contatore viene resettato se all'ingresso R si trova il valore 1. Il valore di conteggio viene impostato su 0.

Un'interrogazione dello stato di segnale all'uscita Q per verificare se questo corrisponde a 1, produce un risultato di 1 se il valore di conteggio è maggiore di 0. L'interrogazione produce un risultato di 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

**Esempio**



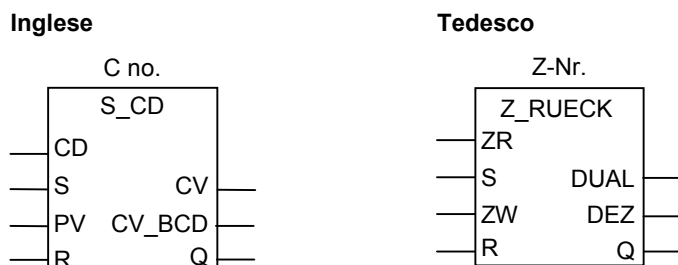
Se lo stato di segnale dell'ingresso E 0.2 passa da 0 a 1, il temporizzatore Z10 viene impostato sul valore 901. Se lo stato di segnale di E 0.0 passa da 0 a 1, il valore del contatore Z10 viene incrementato di 1, tranne quando il suo valore è uguale a 999. Se E 0.3 passa da 0 a 1, il valore di conteggio di Z10 viene settato a 0. Lo stato di segnale dell'uscita A 4.0 è 1 quando Z10 è diverso da 0.

**Nota**

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

## 4.4 Z\_RUECK: Parametrizzazione e conteggio all'indietro

### Simbolo



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, D, L	Ingresso ZR: conteggio all'indietro
S	S	BOOL	E, A, M, D, L, T, Z	Ingresso per preimpostare il contatore
PV	ZW	WORD	E, A, M, D, L o costante	Valore compreso tra 0 e 999 oppure valore di conteggio specificato come C#<Valore> in formato BCD
R	R	BOOL	E, A, M, D, L, T, Z	Ingresso di resettaggio
CV	DUAL	WORD	E, A, M, D, L	Valore del conteggio attuale, in codice esadecimale
CV_BCD	DEZ	WORD	E, A, M, D, L	Valore del conteggio attuale, codificato in formato BCD
Q	Q	BOOL	E, A, M, D, L	Stato del contatore

### Descrizione

Il cambiamento del fronte da 0 a 1 all'ingresso S dell'operazione **Parametrizza e conta all'indietro** preimposta il valore di conteggio ZW nel contatore. Il valore del contatore viene decrementato di 1 se vi è un fronte di salita nell'ingresso ZR e se il valore del contatore è maggiore di 0.

Se il contatore viene impostato e all'ingresso ZR l'RLC è = 1, il contatore conterà una volta nel ciclo **successivo**, anche se non vi era stato un cambio di fronte.

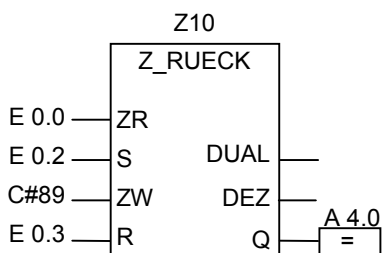
Il contatore viene resettato se all'ingresso R si trova il valore 1. Il valore di conteggio viene impostato su 0.

Un'interrogazione dello stato di segnale all'uscita Q per verificare se questo corrisponde a 1, produce un risultato di 1 se il valore di conteggio è maggiore di 0. L'interrogazione produce un risultato di 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

**Esempio**



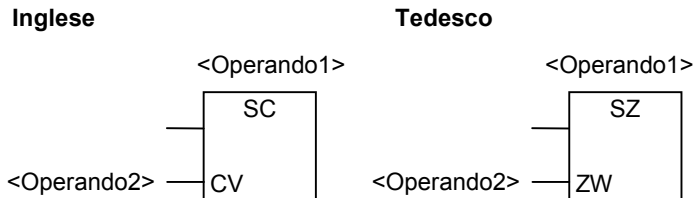
Se lo stato di segnale dell'ingresso E 0.2 passa da 0 a 1, il temporizzatore Z10 viene impostato sul valore 89. Se lo stato di segnale di E 0.0 passa da 0 a 1, il valore del contatore Z10 viene incrementato di 1, tranne quando il suo valore è uguale a 0. Se l'ingresso E 0.1 passa da 0 a 1, il contatore Z10 viene decrementato di 1, tranne quando il suo valore è uguale a 0. Se E 0.3 passa da 0 a 1, il valore di conteggio di Z10 viene settato a 0.

**Nota**

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

## 4.5 SZ: Impostazione del valore di conteggio

### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
Numero del contatore	Numero del contatore	COUNTER	Z	L'operando 1 indica il numero del contatore a cui assegnare un valore di preimpostazione
CV	ZW	WORD	E, A, M, D, L o costante	Il valore da preimpostare (operando 2) può essere compreso tra 0 e 999. Quando si immette una costante, prima del valore che indica il formato BCD, si deve specificare C#, ad esempio C#100

### Descrizione

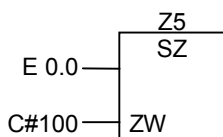
Si può utilizzare l'operazione **Imposta valore di conteggio** per assegnare un valore predefinito al contatore specificato. L'operazione Imposta valore di conteggio viene eseguita soltanto se l'RLC ha un fronte di salita (vale a dire che ha luogo una transizione da 0 a 1 nell'RLC).

Il box **Imposta valore di conteggio** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

### Parola di stato

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

**Esempio**



Il contatore Z5 viene preimpostato con il valore 100 quando lo stato di segnale di E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC). C# indica che si deve immettere un valore BCD.

Se il fronte di salita non è presente, il valore del contatore Z5 non viene modificato.

## 4.6 ZV: Conta in avanti

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
Numero del contatore	COUNTER	Z	L'operando indica il numero del contatore da incrementare

### Descrizione

L'operazione **Conta in avanti** incrementa di uno il valore di un contatore specificato se l'RLC ha un fronte di salita (ovvero, se nell'RLC ha luogo una transizione da 0 a 1), e il valore del contatore è minore di 999. Se l'RLC non ha un fronte di salita, oppure se il contatore è già a 999, il valore del contatore rimane inalterato.

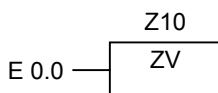
L'operazione **Imposta valore di conteggio** setta il valore del contatore.

Il box **Conta in avanti** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

### 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 (ovvero se vi è un fronte di salita nell'RLC), il valore del contatore Z10 viene incrementato di 1 (a meno che tale valore non sia uguale a 999).

Se non è presente un fronte di salita, il valore di Z10 non viene modificato.

## 4.7 ZR: Conta all'indietro

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
Numero del contatore	COUNTER	Z	L'operando indica il numero del contatore da decrementare

### Descrizione

L'operazione **Conta all'indietro** decrementa di uno il valore di un contatore specifico se l'RLC ha un fronte di salita (ovvero, nell'RLC ha luogo una transizione da 0 a 1), e il valore del contatore è maggiore di 0. Se l'RLC non ha un fronte di salita, oppure se il contatore è già a 0, il valore del contatore rimane inalterato.

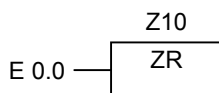
L'operazione **Imposta valore di conteggio** setta il valore del contatore.

Il box **Conta all'indietro** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

### 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 (ovvero se vi è un fronte di salita nell'RLC), il valore del contatore Z10 viene decrementato di 1 (a meno che tale valore non sia uguale a 0).

Se non è presente un fronte di salita, il valore di Z10 non viene modificato.



## 5 Operazioni di blocchi dati

### 5.1 OPN: Apertura di un blocco dati

#### Simbolo

<Numero DB> o  
<Numero DI >

OPN

Parametro	Tipo di dati	Area di memoria	Descrizione
Numero del DB o del DI	BLOCK_DB	-	Numero del DB o del DI; il campo numerico dipende dalla CPU

#### Descrizione

L'operazione **Apri blocco dati** apre un blocco dati globale (DB) o un blocco dati di istanza (DI) come blocco dati. Si tratta di un richiamo incondizionato di un blocco dati. Il numero del blocco dati viene trasferito nel registro DB e/o DI. I successivi comandi DB e/o DI hanno accesso ai blocchi dati corrispondenti a seconda del contenuto dei registri.

#### Parola di stato

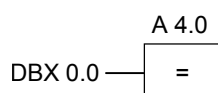
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	-	-	-	-

## Esempio

Segmento 1



Segmento 2



DB10 è il blocco dati attualmente aperto. L'interrogazione di DBX 0.0 si riferisce quindi al bit 0 del byte di dati 0 del DB10. Lo stato di segnale di tale bit viene assegnato all'uscita A 4.0.

# 6 Operazioni di salto

## 6.1 Sommario delle operazioni di salto

### Descrizione

Quest'operazione può essere utilizzata in tutti i blocchi di codice, ad es. nei blocchi organizzativi (OB), nei blocchi funzionali (FB) e nelle funzioni (FC).

Sono disponibili le seguenti operazioni di salto:

- JMP: Salto nel blocco assoluto
- JMP: Salto nel blocco se 1 (condizionato)
- JMPN: Salto nel blocco se 0 (condizionato)

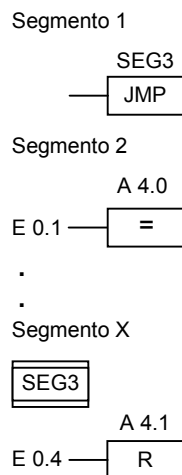
### Etichetta come operando

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 va specificata sopra il box JMP. 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).

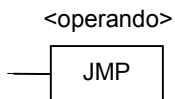
### Etichetta come destinazione

L'etichetta di destinazione deve trovarsi all'inizio di un segmento. Essa viene inserita all'inizio del segmento selezionando LABEL dalla casella di selezione di FUP. Comparirà un box vuoto in cui si potrà specificare il nome dell'etichetta.



## 6.2 JMP: Salto nel blocco assoluto

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
Nome di una etichetta di salto	-	-	L'operando indica l'etichetta in cui si esegue il salto assoluto.

### Descrizione

L'operazione **Salto assoluto nel blocco** corrisponde ad un'operazione **Vai all'etichetta**. Nessuna delle operazioni che si trovano tra l'operazione di salto e l'etichetta vengono eseguite.

Questa operazione può venire utilizzata in tutti i blocchi di codice, ad es. blocchi organizzativi (OB), blocchi funzionali (FB) e funzioni (FC).

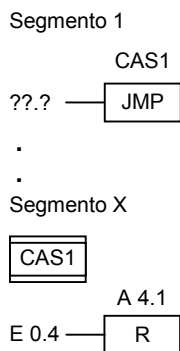
Il box **Salto assoluto nel blocco** non deve essere preceduto da operazioni logiche combinatorie.

### Parola di stato

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

Questa operazione non modifica i bit della parola di stato.

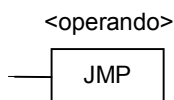
### Esempio



Il salto viene eseguito sempre. Non viene effettuata nessuna delle operazioni che si trovano tra l'operazione di salto e l'etichetta.

## 6.3 JMP: Salto nel blocco se 1 (condizionato)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
Nome di una etichetta di salto	-	-	L'operando indica l'etichetta a cui si salta se RLC = 1.

### Descrizione

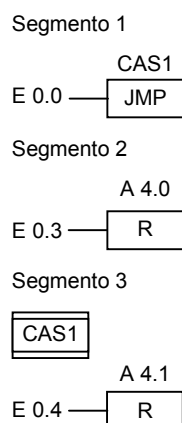
**Salta nel blocco se 1** corrisponde all'operazione **Vai all'etichetta** se RLC = 1. Utilizzare per questa operazione l'elemento FUP del **Salto assoluto** ma preceduto da una combinazione. Il salto condizionato viene eseguito solo se l'RLC di questa combinazione è 1. Nessuna delle operazioni che si trovano tra l'operazione di salto e l'etichetta di salto viene eseguita.

Quest'operazione può essere utilizzata in tutti i blocchi di codice, ad es. nei blocchi organizzativi (OB), nei blocchi funzionali (FB) e nelle funzioni (FC).

### Parola di stato

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

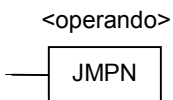
### Esempio



Il salto nell'etichetta CAS1 viene effettuato se E 0.0 = 1. L'operazione, che resetta l'uscita A 4.0, non viene eseguita neppure se E 0.3 = 1.

## 6.4 JMPN: Salto nel blocco se 0 (condizionato)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
Nome di una etichetta di salto	-	-	L'operando indica l'etichetta a cui si salta se RLC = 0.

### Descrizione

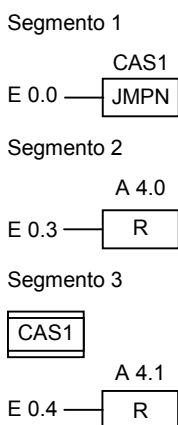
L'operazione **Salta nel blocco se 0** corrisponde ad un'operazione **Vai all'etichetta** che quale viene eseguita se l'RLC è 0.

Questa operazione può venire eseguita in tutti i blocchi di codice, ad es. blocchi organizzativi (OB), blocchi funzionali (FB) e funzioni (FC).

### Parola di stato

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

### Esempio



Il salto nell'etichetta CAS1 viene effettuato se E 0.0 = 0. L'operazione, che resetta l'uscita A 4.0, non viene eseguita neppure se E 0.3 = 1.

Non viene effettuata nessuna delle operazioni che si trovano tra l'operazione di salto e l'etichetta.

## 6.5 LABEL: Etichetta di salto

### Simbolo

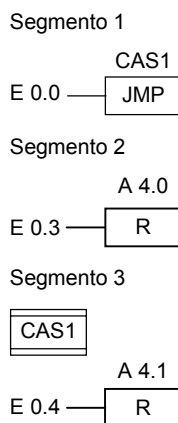


### Descrizione

L'etichetta di salto contrassegna la destinazione di un'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).

Per ogni operazione di salto condizionato (**JMP** o **JMPN**) deve essere presente un'etichetta di salto (**LABEL**).

### Esempio



Il salto nell'etichetta CAS1 viene effettuato se  $E 0.0 = 1$ .

In seguito al salto, l'operazione "Resetta bobina" non viene eseguita in A 4.0, neppure 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 di numeri interi (16 bit)
- SUB\_I: Sottrazione di numeri interi (16 bit)
- MUL\_I: Moltiplicazione di numeri interi (16 bit)
- DIV\_I: Divisione di numeri interi (16 bit)
- ADD\_DI: Somma di numeri interi (32 bit)
- SUB\_DI: Sottrazione di numeri interi (32 bit)
- MUL\_DI: Moltiplicazione di numeri interi (32 bit)
- DIV\_DI: Divisione di numeri interi (32 bit)
- MOD\_DI: Ricavo del resto della divisione (32 bit)

Valutazione dei bit della parola di stato

## 7.2 Valutazione dei bit della parola di stato nelle operazioni in virgola fissa

### Descrizione

Le operazioni di calcolo fondamentali influiscono sui seguenti bit della 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: $-32\,768 \leq \text{risultato} < 0$ (numero negativo) 32 bit: $-2\,147\,483\,648 \leq \text{risultato} < 0$ (numero negativo)	0	1	0	*
16 bit: $32\,767 \geq \text{risultato} > 0$ (numero positivo) 32 bit: $2\,147\,483\,647 \geq \text{risultato} > 0$ (numero positivo)	1	0	0	*

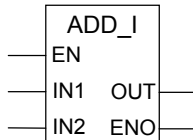
\* Il bit OS non è influenzato dal risultato dell'operazione.

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 di numeri interi (16 bit)

### Simbolo



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

### Descrizione

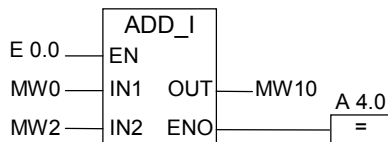
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Somma numeri interi (16 bit)**. L'operazione somma gli ingressi IN1 e IN2. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero intero (16 bit), il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

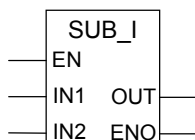
### Esempio



Se E 0.0 = 1, il box ADD\_I viene attivato. Il risultato della somma MW0 + MW2 viene memorizzato nella parola di merker MW10. Se tale risultato non rientra nel campo ammesso per i numeri interi (a 16 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

## 7.4 SUB\_I: Sottrazione di numeri interi (16 bit)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	INT	E, A, M, D, L o costante	Minuendo
IN2	INT	E, A, M, D, L o costante	Sottraendo
OUT	INT	E, A, M, D, L	Risultato della sottrazione
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

### Descrizione

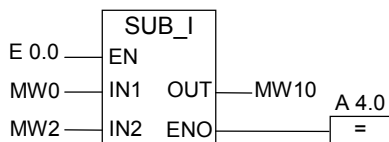
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Sottrai numeri interi (16 bit)**. L'operazione sottrae l'ingresso IN2 da IN1. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero intero (16 bit), il bit OV e il bit OS sono 1 e ENO è 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

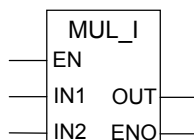
### Esempio



Se E 0.0 = 1, il box SUB\_I viene attivato. Il risultato della sottrazione MW0 - MW2 viene memorizzato nella parola di merker MW10. Se tale risultato non rientra nel campo ammesso per i numeri interi (a 16 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

## 7.5 MUL\_I: Moltiplicazione di numeri interi (16 bit)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	INT	E, A, M, D, L o costante	Moltiplicando
IN2	INT	E, A, M, D, L o costante	Moltiplicatore
OUT	INT	E, A, M, D, L	Risultato della moltiplicazione
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

### Descrizione

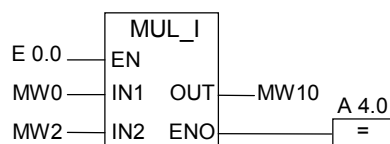
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Moltiplica numeri interi (16 bit)**. L'operazione moltiplica l'ingresso IN1 per IN2. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero intero a 16 bit, il bit OV e il bit OS hanno come valore 1, mentre ENO è 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

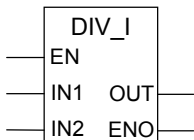
### Esempio



Se E 0.0 = 1, il box MUL\_I viene attivato. Il risultato della moltiplicazione MW0 x MW2 viene memorizzato nella parola di merker MW10. Se tale risultato non rientra nel campo ammesso per i numeri interi (a 16 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

## 7.6 DIV\_I: Divisione di numeri interi (16 bit)

### Simbolo



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

### Descrizione

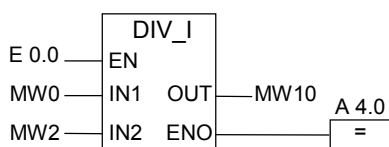
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Dividi numeri interi (16 bit)**. L'operazione divide l'ingresso IN1 per IN2. Il quoziente di questa divisione (parte intera) può essere interrogato all'uscita OUT. Il resto della divisione non può essere letto. Se il quoziente non rientra nel campo permesso per un numero intero (16 bit), il bit OV e il bit OS hanno come valore 1, ENO ha invece il valore 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

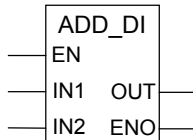
### Esempio



Se E 0.0 = 1, il box DIV\_I viene attivato. Il quoziente della divisione MW0 per MW2 viene memorizzato nella parola di merker MW10. Se tale quoziente non rientra nel campo ammesso per i numeri interi (a 16 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

## 7.7 ADD\_DI: Somma di numeri interi (32 bit)

### Simbolo



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

### Descrizione

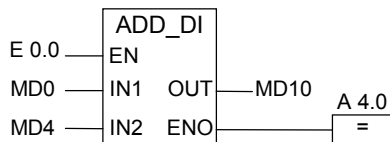
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Somma numeri interi (32 bit)**. L'operazione somma gli ingressi IN1 e IN2. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero intero (32 bit), il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

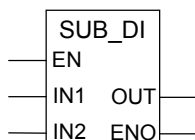
### Esempio



Se  $E\ 0.0 = 1$ , il box ADD\_DI viene attivato. Il risultato della somma  $MD0 + MD4$  viene memorizzato nella doppia parola di merker MD10. Se tale risultato non rientra nel campo ammesso per i numeri interi (a 32 bit) o se lo stato del segnale dell'ingresso  $E\ 0.0 = 0$ , viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

## 7.8 SUB\_DI: Sottrazione di numeri interi (32 bit)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	DINT	E, A, M, D, L o costante	Minuendo
IN2	DINT	E, A, M, D, L o costante	Sottraendo
OUT	DINT	E, A, M, D, L	Risultato della sottrazione
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

### Descrizione

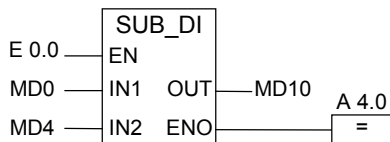
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Sottrai numeri interi (32 bit)**. L'operazione sottrae l'ingresso IN2 da IN1. Il risultato può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero intero a 32 bit, il bit OV e il bit OS sono 1, mentre ENO è 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

### Esempio

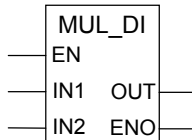


Se E 0.0 = 1, il box SUB\_DI viene attivato. Il risultato della sottrazione MD0 - MD4 viene memorizzato nella doppia parola di merker MD10. Se tale risultato non rientra nel campo ammesso per i numeri interi (a 32 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.



## 7.9 MUL\_DI: Moltiplicazione di numeri interi (32 bit)

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	DINT	E, A, M, D, L o costante	Moltiplicando
IN2	DINT	E, A, M, D, L o costante	Moltiplicatore
OUT	DINT	E, A, M, D, L	Risultato della moltiplicazione
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

### Descrizione

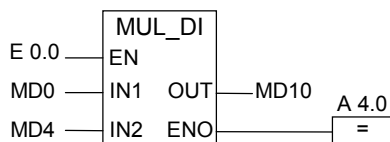
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Moltiplica numeri interi (32 bit)**. L'operazione moltiplica l'ingresso IN1 per IN2. Il risultato può essere letto all'uscita 0. Se il risultato non rientra nel campo permesso per un numero intero (32 bit), il bit OV e il bit OS hanno come valore 1, ENO è invece 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

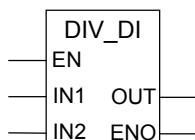
### Esempio



Se E 0.0 = 1, il box MUL\_DI viene attivato. Il risultato della moltiplicazione MD0 x MD4 viene memorizzato nella doppia parola di merker MD10. Se tale risultato non rientra nel campo ammesso per i numeri interi (a 32 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

## 7.10 DIV\_DI: Divisione di numeri interi (32 bit)

### Simbolo



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

### Descrizione

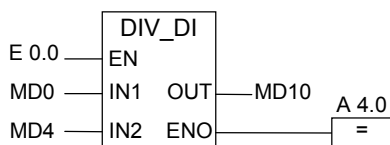
Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Dividi numeri interi (32 bit)**. L'operazione divide l'ingresso IN1 per IN2. Il quoziente (parte intera) può essere letto all'uscita OUT. L'operazione Dividi numeri interi (32 bit) memorizza il quoziente come un singolo valore a 32 bit nel formato DINT senza produrre il resto. Se il quoziente non rientra nel campo permesso per un numero intero a 32 bit, il bit OV e il bit OS sono 1, e ENO è 0.

Vedere anche Valutazione dei bit della parola di stato.

### Parola di stato

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

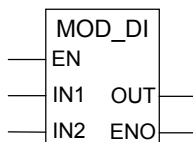
### Esempio



Se E 0.0 = 1, il box DIV\_DI viene attivato. Il quoziente della divisione MD0 per MD4 viene memorizzato nella doppia parola di merker MD10. Se tale quoziente non rientra nel campo ammesso per i numeri interi (a 32 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

## 7.11 MOD\_DI: Ricavo del resto della divisione (32 bit)

### Simbolo



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

### Descrizione

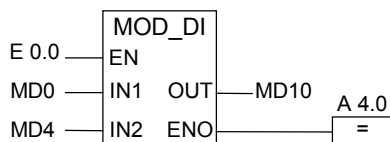
Uno stato di segnale di 1 all'ingresso di abilitazione (EN) attiva l'operazione **Ricava resto della divisione (32 bit)**. L'operazione divide l'ingresso IN1 per IN2. Il resto (frazione) può essere letto all'uscita OUT. Se il risultato non rientra nel campo permesso per un numero intero a 32 bit, il bit OV è 1, e ENO è 0.

Vedere anche Valutazione dei bit della parola di stato.

### 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 box MOD\_DI viene attivato. Il ricavo della divisione MD0 per MD4 viene memorizzato nella parola di merker MD10. Se il risultato non rientra nel campo ammesso per i numeri interi (a 32 bit) o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.



## 8 Operazioni con numeri in virgola mobile

### 8.1 Sommario delle operazioni matematiche con i numeri mobile

#### Descrizione

I numeri IEEE 754 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 754 in virgola mobile a 32 bit:

- ADD\_R: Somma di numeri in virgola mobile
- SUB\_R: Sottrazione di numeri in virgola mobile
- MUL\_R: Moltiplicazione di numeri in virgola mobile
- DIV\_R: Divisione di numeri in virgola mobile

Con l'aritmetica in virgola mobile, le seguenti funzioni possono essere eseguite con **un** numero in virgola mobile (32 bit, IEEE 754):

- ABS: Formazione del valore assoluto di un numero in virgola mobile
- SQR: Formazione del quadrato di un numero in virgola mobile
- SQRT: Formazione della radice quadrata di un numero in virgola mobile
- EXP: Formazione del valore esponenziale di un numero in virgola mobile
- LN: Formazione del logaritmo naturale di un numero in virgola mobile
- Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
  - seno (SIN) ed arcoseno (ASIN)
  - coseno (COS) ed arcocoseno (ACOS)
  - tangente (TAN) ed arcotangente (ATAN)

Valutazione dei bit della parola di stato nelle operazioni in virgola mobile

## 8.2 Valutazione dei bit della parola di stato nelle operazioni in virgola mobile

### Descrizione

Le operazioni matematiche in virgola mobile 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).

<b>Campo di validità</b>	<b>A1</b>	<b>A0</b>	<b>OV</b>	<b>OS</b>
+0, -0 (zero)	0	0	0	*
$-3.402823E+38 < \text{risultato} < -1.175494E-38$ (numero negativo)	0	1	0	*
$+1.175494E-38 < \text{risultato} < +3.402823E+38$ (numero positivo)	1	0	0	*

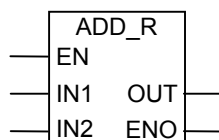
\* Il bit OS non è influenzato dal risultato dell'operazione.

<b>Campo di non validità</b>	<b>A1</b>	<b>A0</b>	<b>OV</b>	<b>OS</b>
superamento negativo di capacità $-1.175494E-38 < \text{risultato} < -1.401298E-45$ (numero negativo)	0	0	1	1
superamento negativo di capacità $+1.401298E-45 < \text{risultato} < +1.175494E-38$ (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 di numeri in virgola mobile

#### Simbolo



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

#### Descrizione

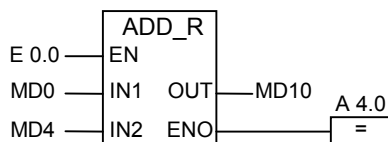
Se lo stato del segnale all'ingresso di abilitazione (EN) è 1, l'operazione **Somma numeri in virgola mobile** viene attivata. Essa somma gli ingressi IN1 e IN2. Il risultato può essere letto all'uscita OUT. Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

Vedere anche Valutazione dei bit della parola di stato.

#### Parola di stato

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

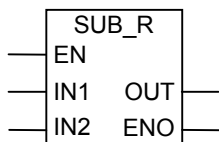
#### Esempio



Se E 0.0 = 1, il box ADD\_R viene attivato. Il risultato della somma MD0 + MD4 viene memorizzato nella doppia parola di merker MD10. Se uno degli ingressi o il risultato non sono un numero in virgola mobile o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

### 8.3.2 SUB\_R: Sottrazione di numeri in virgola mobile

#### Simbolo



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

#### Descrizione

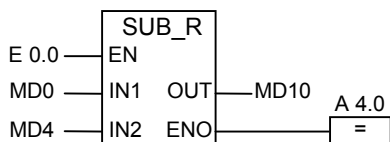
Se lo stato del segnale all'ingresso di abilitazione (EN) è 1, l'operazione **Sottrai numeri in virgola mobile** viene attivata. Essa sottrae l'ingresso IN2 da IN1. Il risultato può essere letto all'uscita OUT. Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

Vedere anche Valutazione dei bit della parola di stato.

#### Parola di stato

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

#### Esempio

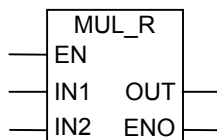


Se E 0.0 = 1, il box SUB\_R viene attivato. Il risultato della sottrazione MD0 - MD4 viene memorizzato nella doppia parola di merker MD10. Se uno degli ingressi o il risultato non sono un numero in virgola mobile o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.



### 8.3.3 MUL\_R: Moltiplicazione di numeri in virgola mobile

#### Simbolo



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

#### Descrizione

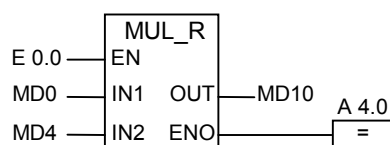
Se lo stato del segnale all'ingresso di abilitazione (EN) è 1, l'operazione **Moltiplica numeri in virgola mobile** viene attivata. Essa moltiplica l'ingresso IN1 per IN2. Il risultato può essere letto all'uscita OUT. Il risultato può essere letto all'uscita OUT. Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

Vedere anche Valutazione dei bit della parola di stato.

#### Parola di stato

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

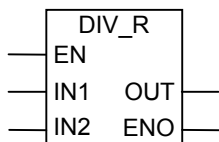
#### Esempio



Se E 0.0 = 1, il box MUL\_R viene attivato. Il risultato della moltiplicazione MD0 x MD4 viene memorizzato nella doppia parola di merker MD10. Se uno degli ingressi o il risultato non sono un numero in virgola mobile o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

### 8.3.4 DIV\_R: Divisione di numeri in virgola mobile

#### Simbolo



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

#### Descrizione

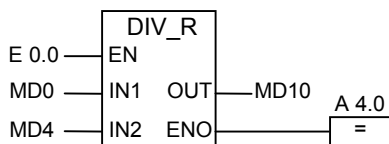
Se lo stato del segnale all'ingresso di abilitazione (EN) è 1, l'operazione **Dividi numeri in virgola mobile** viene attivata. Essa divide l'ingresso IN1 per IN2. Il risultato può essere letto all'uscita OUT. Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

Vedere anche Valutazione dei bit della parola di stato.

#### Parola di stato

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

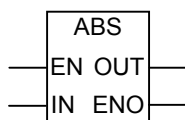
#### Esempio



Se E 0.0 = 1, il box DIV\_R viene attivato. Il risultato della divisione di MD0 per MD4 viene memorizzato nella doppia parola di merker MD10. Se uno degli ingressi o il risultato non sono un numero in virgola mobile o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0 e l'operazione non viene eseguita.

### 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, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Valore di ingresso: numero in virgola mobile
OUT	REAL	E, A, M, D, L	Valore di uscita: valore assoluto di numero in virgola mobile
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

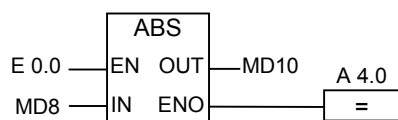
#### Descrizione

L'operazione **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	X	-	-	-	-	0	X	X	1

#### Esempio



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

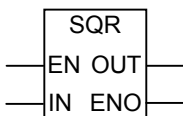
MD8 = -6,234 dà MD12 = 6,234.

L'uscita A 4.0 è 0 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, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Quadrato del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

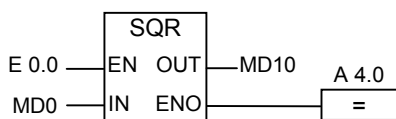
#### Descrizione

Con l'operazione **Forma quadrato di un numero in virgola mobile** è possibile elevare al quadrato un numero in virgola mobile. Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

#### Parola di stato

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

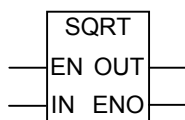
#### Esempio



Se E 0.0 = 1, il box SQR viene attivato. Il risultato della divisione di MD0 per MD4 viene memorizzato nella doppia parola di merker MD10. Se uno degli ingressi o il risultato non sono un numero in virgola mobile o se lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0.

## 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, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Radice quadrata del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

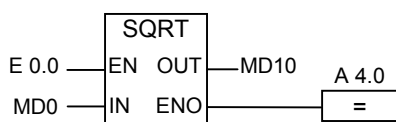
### Descrizione

Con l'operazione **Forma radice quadrata di un numero in virgola mobile** è possibile ricavare la radice quadrata di un numero in virgola mobile. Essa dà un risultato positivo se l'operando è maggiore di 0. Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO ha il valore 0.

### Parola di stato

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

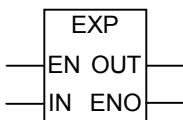
### Esempio



Quando  $E\ 0.0 = 1$ , il box SQRT viene attivato. Il risultato di SQRT (MD0) viene memorizzato nella doppia parola di merker MD10. Se  $MD0 < 0$  oppure una delle uscite o il risultato non sono un numero in virgola mobile e se lo stato del segnale dell'ingresso  $E\ 0.0 = 0$ , viene attribuito all'uscita A 4.0 lo stato di segnale 0.

### 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, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Esponente del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

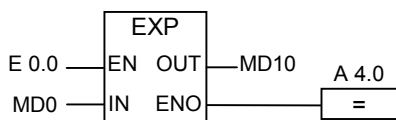
#### Descrizione

Con l'operazione **Forma valore esponenziale di un numero in virgola mobile** è possibile ricavare il valore esponenziale di un numero in virgola mobile sulla base e (= 2,71828...). Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

#### Parola di stato

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

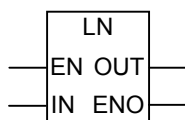
#### Esempio



Quando E 0.0 = 1, il box EXP viene attivato. Il risultato di EXP (MD0) viene memorizzato nella doppia parola di merker MD10. Se uno degli ingressi o il risultato non sono un numero in virgola mobile e lo stato del segnale dell'ingresso E 0.0 = 0, viene attribuito all'uscita A 4.0 lo stato di segnale 0.

### 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, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Logaritmo naturale del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

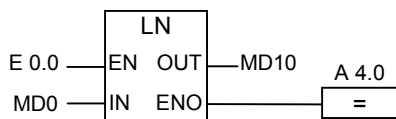
#### Descrizione

Con l'operazione **Forma logaritmo naturale di un numero in virgola mobile** è possibile ricavare il logaritmo naturale di un numero in virgola mobile. Se uno degli ingressi o il risultato non sono un numero in virgola mobile, il bit OV e il bit OS hanno il valore 1 e ENO il valore 0.

#### Parola di stato

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

#### Esempio



Quando  $E 0.0 = 1$ , il box LN viene attivato. Il risultato di LN (MD0) viene memorizzato nella doppia parola di merker MD10. Se  $MD0 < 0$  oppure uno degli ingressi o il risultato non sono un numero in virgola mobile e lo stato del segnale dell'ingresso  $E 0.0 = 0$ , viene attribuito all'uscita A 4.0 lo stato di segnale 0.

### 8.4.5 Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile

#### Descrizione

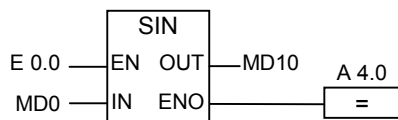
Con le operazioni seguenti è possibile ricavare funzioni trigonometriche di angoli rappresentati sotto forma di angoli (32 bit, IEEE 754):

Operazione	Significato
SIN	Formazione del seno di un numero in virgola mobile di un angolo indicato in radianti.
ASIN	Formazione dell'arcoseno di un numero in virgola mobile. Il risultato è un angolo indicato in radianti. Il valore è compreso nel seguente campo: $-\pi / 2 \leq \text{arcoseno} \leq + \pi / 2$ , dove $\pi = 3.14\dots$
COS	Formazione del coseno di un numero in virgola mobile di un angolo indicato in radianti.
ACOS	Formazione dell'arcocoseno di un numero in virgola mobile. Il risultato è un angolo indicato in radianti. Il valore è compreso nel seguente campo: $0 \leq \text{arcocoseno} \leq + \pi$ , dove $\pi = 3.14\dots$
TAN	Formazione della tangente di un numero in virgola mobile di un angolo indicato in radianti.
ATAN	Formazione dell'arcotangente di un numero in virgola mobile. Il risultato è un angolo indicato in radianti. Il valore è compreso nel seguente campo: $-\pi / 2 \leq \text{arcotangente} \leq + \pi / 2$ , dove $\pi = 3.14\dots$

#### Parola di stato

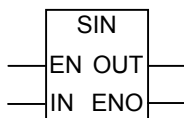
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	X	X	X	X	X	0	X	X	1

#### Ejemplo

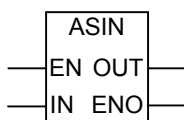


Quando  $E\ 0.0 = 1$ , il box SIN viene attivato. Il risultato di SIN (MD0) viene memorizzato nella doppia parola di merker MD10. Se uno degli ingressi o il risultato non sono un numero in virgola mobile e lo stato del segnale dell'ingresso  $E\ 0.0 = 0$ , viene attribuito all'uscita A 4.0 lo stato di segnale 0.

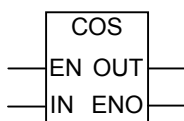


**Simbolo**

Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Seno del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

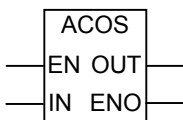
**Simbolo**

Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Arcoseno del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

**Simbolo**

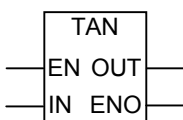
Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Coseno del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

**Simbolo**



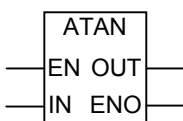
Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Arcocoseno del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

**Simbolo**



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Tangente del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

**Simbolo**

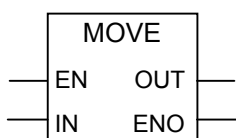


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	REAL	E, A, M, D, L o costante	Numero
OUT	REAL	E, A, M, D, L	Arcotangente del numero
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

## 9 Operazioni di trasferimento

### 9.1 MOVE: Assegnazione di un valore

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN	Tutti i tipi di dati semplici che hanno una lunghezza di 8, 16, e 32 bit	E, A, M, D, L o costante	Valore di origine
OUT	Tutti i tipi di dati semplici che hanno una lunghezza di 8, 16, e 32 bit	E, A, M, D, L	Indirizzo di arrivo
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

#### Descrizione

L'operazione **Assegna un valore** consente di preassegnare una variabile con un valore specifico.

Il valore specificato all'ingresso IN viene copiato nell'indirizzo specificato all'uscita OUT. Tale uscita ha il medesimo stato di segnale di EN.

Con il box MOVE, l'operazione **Assegna un valore** può copiare tutti i tipi di dati elementari che hanno una lunghezza di 8, 16, o 32 bit. I tipi di dati definiti dall'utente, quali i campi o le strutture, devono essere copiati con la funzione di sistema SFC 20 "BLKMOV".

L'operazione **Assegna un valore** è influenzata dall' MCR.

#### Parola di stato

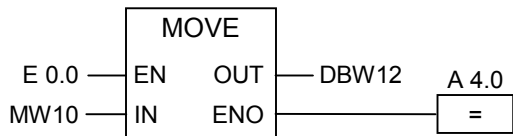
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	1	-	-	-	-	0	1	1	1

**Nota**

Con il trasferimento di un valore in un tipo di dati di lunghezza diversa, i byte con i valori più alti vengono abbreviati o integrati con zeri secondo necessità:

<b>Esempio: doppia parola</b>	<b>1111 1111</b>	<b>0000 1111</b>	<b>1111 0000</b>	<b>0101 0101</b>
<b>Trasferimento</b>	<b>Risultato</b>			
in una doppia parola:	1111 1111	0000 1111	1111 0000	0101 0101
in un byte:				0101 0101
in una parola:			1111 0000	0101 0101
<b>Esempio: byte:</b>				<b>1111 0000</b>
<b>Trasferimento</b>	<b>Risultato</b>			
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 copiato nella parola di dati 12 dell'operando attuale DB.

Se l'operazione viene eseguita A 4.0 = 1.

# 10 Operazioni di controllo del programma

## 10.1 Sommario delle operazioni di comando del programma

### Descrizione

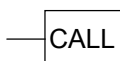
Sono disponibili le seguenti operazioni di comando del programma:

- CALL Richiamo di FC/SFC 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 una multi-istanza
- Richiamo di blocchi da una biblioteca
  
- Funzioni del relè master control (MCR)
- Avvertenze importanti sulle funzionalità MCR
- MCR< Attivazione di una zona di relè master control
- MCR> Disattivazione di una zona di relè master control
- MCRA Inizio della zona relè master control
- MCRD Fine della zona relè master control
  
- RET Salto indietro

## 10.2 CALL: Richiamo di FC/SFC senza parametri

### Simbolo

<numero>



Parametro	Tipo di dati	Area di memoria	Descrizione
Numero	BLOCK_FC	-	Numero di FC o SFC (per esempio, FC10 o SFC59). Le FC o SFC disponibili dipendono dalla propria CPU. Un richiamo condizionato con un parametro del tipo di dati BLOCK_FC come operando, è possibile solo in FB e non in FC.

### Descrizione

L'utente può adoperare l'operazione *Richiama FC/SFC* senza parametri per richiamare una funzione (FC) o una funzione di sistema (SFC). A seconda della combinazione precedente, il richiamo è condizionato o incondizionato (vedere l'esempio).

Nel caso di un richiamo condizionato, nella parte istruzioni di una funzione (FC), non è possibile indicare come operando un parametro del tipo dei dati BLOCK\_FC. Tuttavia, per un blocco funzionale (FB) è possibile indicare come operando un parametro del tipo BLOCK\_FC.

Un richiamo condizionato viene eseguito solamente se RLC è 1. Se non viene eseguito un richiamo condizionato, RLC, dopo l'operazione di richiamo, è 0. Quando l'operazione viene eseguita, essa effettua quanto segue:

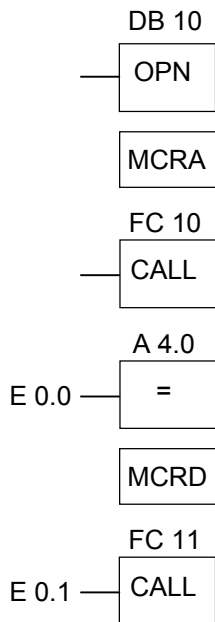
- memorizza l'indirizzo di cui c'è bisogno per ritornare al blocco richiamante
- memorizza i due registri di blocchi dati (blocco dati e blocco di istanza)
- cambia l'attuale campo di dati locali nel precedente campo di dati locali
- crea il nuovo campo di dati locali per le funzioni FC o SFC richiamate
- colloca il bit MA (bit attivo MCR) nello stack di blocco (B-stack)

Al termine di queste funzioni, l'elaborazione del programma continua nel blocco richiamato.

### Parola di stato

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

## Esempio



Se il richiamo incondizionato di FC10 viene eseguito, l'operazione CALL esegue le seguenti funzioni:

- memorizza l'indirizzo di ritorno dell'FB attuale
- memorizza i selettori di DB10 e del blocco di istanza FB
- sposta il bit MA impostato a 1 dall'operazione MCRA, nello stack di blocco (B-stack) e lo imposta a 0 per l'FC10 richiamata.

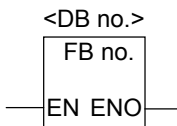
L'elaborazione del programma continua in FC10. Se si vuole utilizzare la funzione MCR in FC 10, la si deve attivare nuovamente in tale funzione. Quando FC10 finisce, l'elaborazione del programma passa nuovamente all'FB richiamante. Il bit MA viene ripristinato. DB 10 e il blocco dati di istanza dell'FB definito dall'utente tornano ad essere i DB attuali, a prescindere da quali DB sono stati usati da FC 10.

Dopo il salto all'indietro di FC 10, viene assegnato all'uscita A 4.0 lo stato di segnale dell'ingresso E 0.0. Il richiamo di FC 11 è condizionato. Esso viene eseguito solo se lo stato di segnale dell'ingresso E 0.1 è 1. Una volta eseguito il richiamo, la funzione è la stessa utilizzata nel richiamo di FC 10.

Si dovrebbero programmare queste operazioni alla fine di FB o FC in modo che queste siano le ultime operazioni eseguite nel blocco.

## 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.	BLOCK_FB	-	Numero del FB/DB; il campo dipende dalla CPU
DB no.	BLOCK_DB	-	

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

### Parola di stato

		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Condizionato	scrive	X	-	-	-	0	0	X	X	X
Absolute	scrive	-	-	-	-	0	0	X	X	X



## Esempio

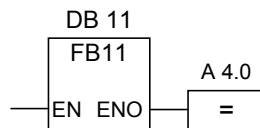
Segmento 1



Segmento 2



Segmento 3



Segmento 4



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 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 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.

---

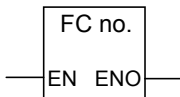
### Nota

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

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.

Richiamando una FC, se la tabella di dichiarazione delle variabili del blocco richiamato dispone di dichiarazioni dati del tipo IN, OUT e IN\_OUT, queste variabili verranno visualizzate nel programma del blocco chiamante in forma di lista dei parametri formali.

Con il richiamo di FC è **assolutamente indispensabile** assegnare parametri attuali ai parametri formali nella posizione del richiamo. Eventuali valori iniziali della dichiarazione FC sono irrilevanti.

### Parola di stato

		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
Condizionato	scrive	X	-	-	-	0	0	X	X	X
Absolute	scrive	-	-	-	-	0	0	X	X	X

## Esempio

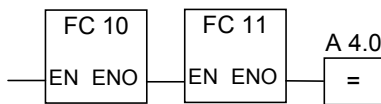
Segmento 1



Segmento 2



Segmento 3



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 quanto 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.

---

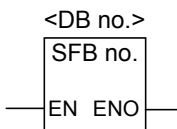
### Nota

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.	BLOCK_SFB	-	Numero del SFB/DB; il campo dipende dalla CPU
DB no.	BLOCK_DB	-	

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

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
Condizionato	scrive	X	-	-	-	0	0	X	X	X
Absolute	scrive	-	-	-	-	0	0	X	X	X

## Esempio

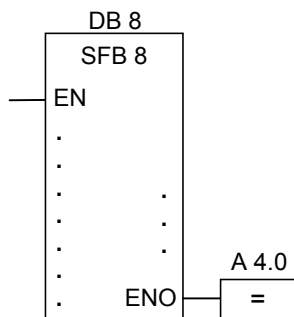
Segmento 1



Segmento 2



Segmento 3



Segmento 4



Nei percorsi di corrente sopraindicati, relativi a 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, ha luogo 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 correttamente, ENO = 1 e quindi A4.0 = 1.

---

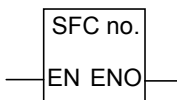
### Nota

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

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
Condizionato	scrive	X	-	-	-	0	0	X	X	X
Assoluto	scrive	-	-	-	-	0	0	X	X	X

## Esempio

Segmento 1

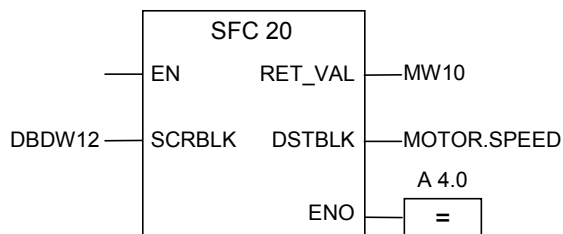
DB 10

OPN

Segmento 2

MCRA

Segmento 3



L'esempio soprariportato 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 quanto 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

---

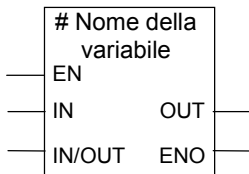
### Nota

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, D, L, T, Z	Ingresso di abilitazione
ENO	BOOL	E, A, M, D, L	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, ENO 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 Funzioni del relè master control (MCR)

Avvertenze importanti sulle funzionalità MCR

### Definizione del relè master control

Il relè master control (MCR, vedere anche Attivazione/disattivazione di una zona MCR) viene usato per eccitare e diseccitare il flusso segnali. Un flusso di segnale diseccitato corrisponde ad una sequenza di operazioni che scrive un valore zero invece del valore calcolato, o ad una sequenza di operazioni che lascia immutato il valore esistente della memoria.

Le operazioni *Assegnazione* e *Connettore* scrivono uno 0 in memoria se l'MCR è 0. Le operazioni *Imposta operando* e *Resetta operando* lasciano immutato il valore esistente.

Operazioni influenzati di zone MCR:

- Connettore
- Assegnazione
- Imposta operando
- Resetta operando
- Imposta resetta flip flop
- Resetta imposta flip flop
- Assegna un valore

### Operazioni che dipendono dal MCR e le sue reazioni sullo stato di segnale del MCR

Stato di segnale di MCR	Assegnazione, Connettore	Imposta o resetta operando	Assegna un valore
0 ("OFF ")	Scrive "0". (limita un relè che entra in stato di quiete quando viene tolta la tensione.)	Non scrive. (limita un relè che conserva lo stato attuale quando viene tolta la tensione.)	Scrive "0". (Imita una componente che in caso di caduta di tensione fornisce il valore "0".)
1 ("ON ")	Elaborazione normale.	Elaborazione normale.	Elaborazione normale.

## 10.10 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 l'attivazione di una zona di relè master control e la disattivazione di una zona di relè master control viene scritto il valore 0 attraverso tutte le assegnazioni! Ciò riguarda naturalmente anche **tutti** i box che contengono un'assegnazione, compresa l'assegnazione di parametri ai blocchi!
- L'MRC è disattivato esattamente se prima di un comando di attivazione di una zona di relè master control 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 successivi 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 trasferiti parametri.

#### KOP/FUP

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

#### Rimedio

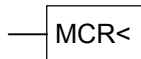
Sciogliere i comandi indicati dalla dipendenza MCR:

- 1° **disattivare** il relè master control con l'operazione Fine della zona relè master control **prima** dell'istruzione o del segmento in questione
- 2° **riattivare** il relè master control con l'operazione Inizio della zona relè master control **dopo** l'istruzione o il segmento in questione.

## 10.11 MCR< / MCR>: Attivazione/disattivazione di una zona di relè master control

Avvertenze importanti sulle funzionalità MCR

### Simbolo



### Attivazione di una zona relè master control

L'operazione *Attiva zona relè master control (MCR<)* avvia una funzione che memorizza l'RLC nello stack MCR. Tale azione apre una zona MCR. Le operazioni vengono influenzate dall'RLC memorizzato nello stack MCR quando viene aperta la zona MCR. Lo stack MCR funge da buffer LIFO. Sono possibili solo otto registrazioni. Se lo stack è già pieno, l'operazione *Attiva zona relè master control* determina un errore nello stack MCR (MCRF).

### Simbolo



### Disattivazione di una zona relè master control

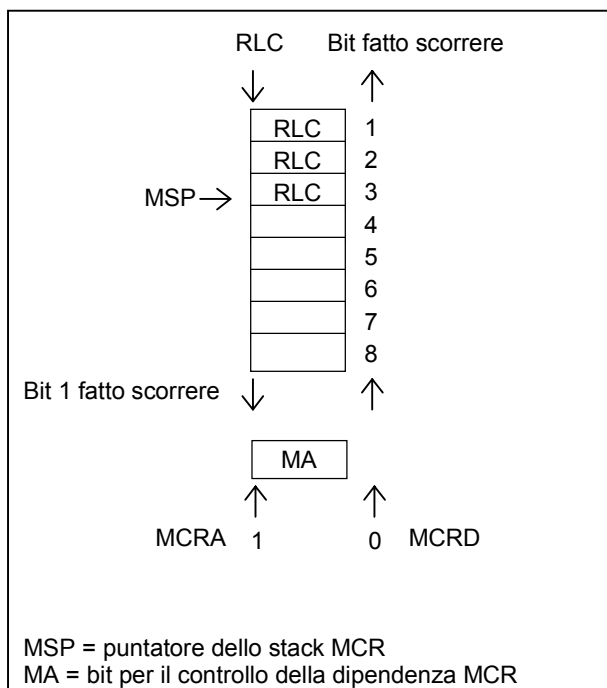
L'operazione *Disattiva zona relè master control (MCR>)* chiude la zona MCR aperta per ultima. L'operazione esegue questo compito cancellando la registrazione di RLC dallo stack MCR. L'RLC vi era stato collocato dall'operazione *Attiva zona relè master control*. Lo stack MCR funge da buffer LIFO (cioè last in, first out). La posizione che si libera alla fine di tale stack viene settata a 1. Se lo stack è già vuoto, l'operazione *Disattiva zona relè master control* produce un'avaria dello stack di MCR (MCRF).

### Stack MCR

L'MCR viene controllata da uno stack largo 1 bit e profondo 8 registrazioni. L'MCR viene attivata fino a quando tutte e otto le registrazioni dello stack sono 1. L'operazione MCR< copia il risultato logico combinatorio nello stack MCR. L'operazione MCR> rimuove l'ultima registrazione dallo stack e setta l'indirizzo dello stack liberatosi ad 1.

Se si verifica un errore, ad es. se si susseguono più di otto operazioni MCR> o si tenta di eseguire l'operazione MCR> con lo stack MCR vuoto, viene emesso il messaggio di errore MCRF.

Il controllo dello stack MCR segue il puntatore allo stack (MSP: 0 = vuoto, 1 = 1 registrazione, 2 = due registrazioni, ..., 8 = otto registrazioni).



L'operazione MCR< assume lo stato del segnale dell'RLC e lo copia nel bit MCR.

L'operazione MCR> setta in modo assoluto il bit MCR ad 1. Per questo motivo, ogni altra operazione tra le operazioni MCRA e MCRD opera indipendentemente dal bit MCR.

### Annidamento delle operazioni (MCR<) e (MCR>)

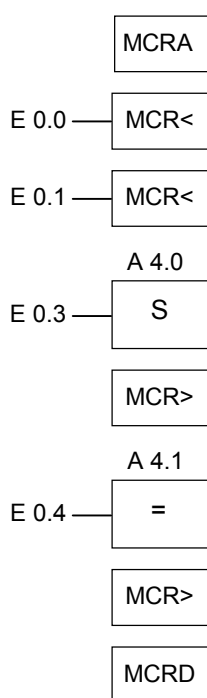
Le operazioni MCR< e MCR> possono essere annidate. La profondità di annidamento massima è di otto, cioè, è possibile scrivere di seguito un massimo di otto operazioni MCR< prima di poter inserire un'operazione MCR>. Si deve programmare sempre lo stesso numero di operazioni MCR< e MCR>.

Se le operazioni MCR< sono annidate, si viene a formare il bit MCR del livello di annidamento più basso. A questo punto, l'operazione MCR< combina l'RLC attuale con il bit MCR attuale in conformità alla tabella della verità AND.

Quando un'operazione MCR> termina un livello di annidamento, essa carica il bit MCR da un livello superiore.

**Parola di stato**

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

**Esempio**

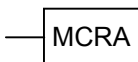
Quando l'operazione MCRA attiva la funzione MCR, è possibile creare fino a otto zone MCR annidate. Nell'esempio qui riportato vi sono due zone MCR. La prima operazione MCR> opera insieme alla seconda operazione MCR<. Tutte le operazioni tra il secondo set di parentesi di MCR (MCR< MCR>) appartengono alla seconda zona MCR. Le operazioni vengono effettuate nel seguente modo:

- E 0.0 = 1: lo stato di segnale di E 0.4 viene assegnato all'uscita A 4.1.
- E 0.0 = 0: l'uscita A 4.1 è 0 indipendentemente dallo stato di segnale di E 0.4. L'uscita A 4.0 non viene modificata, indipendentemente dallo stato di segnale di E 0.3.
- E 0.0 e E 0.1 = 1: l'uscita A 4.0 viene impostata su 1 se E 0.3 = 1 e A 4.1 = E 0.4.
- E 0.1 = 0: l'uscita A 4.0 non viene modificata, indipendentemente dallo stato di segnale di E 0.3 e E 0.0.

## 10.12 MCRA/ MCRD: Inizio/fine della zona relè master control

Avvertenze importanti sulle funzionalità MCR

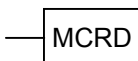
### Simbolo



### Inizio zona relè master control

L'operazione *Inizio zona relè master control* consente di attivare la dipendenza da MCR dei comandi successivi. Dopo questo comando, l'utente può adoperare le operazioni **Attivazione zona relè master control** e **Disattivazione zona relè master control** per programmare le zone MCR. Se il programma attiva una zona MCR, tutte le azioni MCR dipendono dal contenuto dello stack MCR.

### Simbolo



### Fine zona relè master control

L'operazione *Fine zona relè master control* disattiva la funzione del relè master control. In seguito all'operazione *Disattiva zona relè master control*, l'utente non può più programmare le zone MCR. Se il programma disattiva la zona MCR, l'MCR è sempre sotto tensione, indipendentemente dalle registrazioni presenti nello stack MCR.

Lo stack MCR e il bit che lo controlla (il bit MA) fanno riferimento al relativo livello e devono essere sempre memorizzati e ricaricati se si effettua una commutazione nell'ambito del livello di sequenza. All'inizio di ogni livello di sequenza essi vengono preimpostati (i bit di introduzione da 1 a 8 vengono settati a 1, il puntatore stack MCR a 0, e il bit MA a 0).

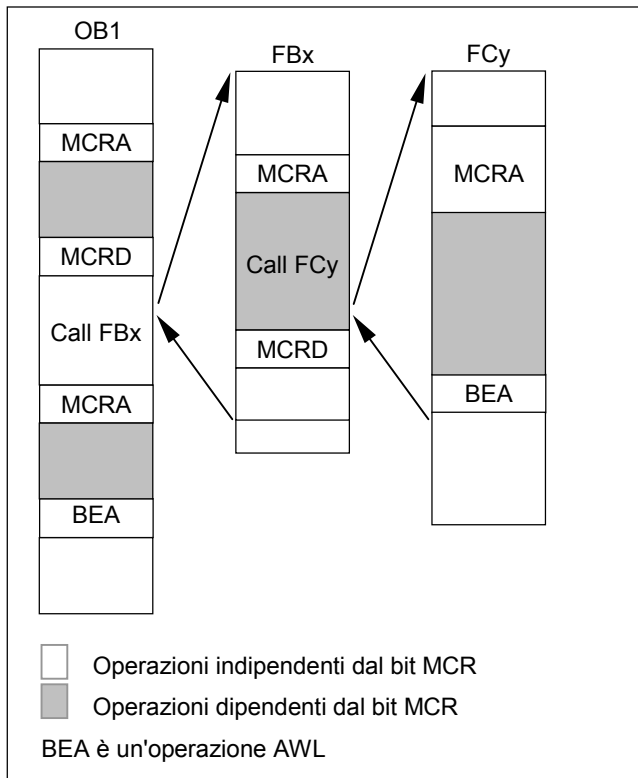
Lo stack MCR viene fatto passare da blocco a blocco, mentre il bit MA viene memorizzato ad ogni richiamo di blocco e settato a 0. Alla fine del blocco, questa operazione viene ripetuta.

L'MCR può essere implementato in modo da ottimizzare il tempo di esecuzione di CPU generatrici di codice. Il motivo è da ricercarsi nel fatto che il controllo di MCR non viene trasferito al blocco, ma questo deve essere attivato in modo esplicito da un'operazione MCRA. Una CPU generatrice di codice riconosce questa operazione e genera il codice supplementare necessario per l'analisi dello stack MCR fino a quando essa riconosce un'operazione MCRD o si raggiunge la fine del blocco. Per le operazioni al di fuori della zona MCRA/MCRD non si ha un aumento del tempo di esecuzione.

Nel programma, le operazioni *MCRA* e *MCRD* devono essere usate sempre in coppia.

### Attivazione e disattivazione di una zona MCR

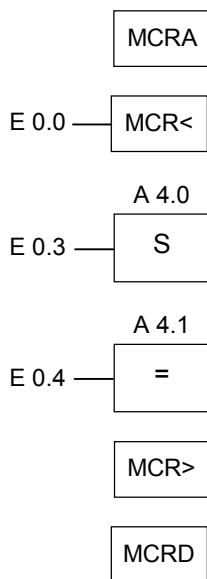
Le operazioni programmate tra MCRA e MCRD dipendono dallo stato del segnale del bit MCR. Le operazioni programma al di fuori di una sequenza MCRA-MCRD non dipendono invece dallo stato di segnale del bit MCR. Se manca un'operazione MCRD, le operazioni programmate tra le operazioni MCRA e BEA dipendono dal bit MCR.



### Parola di stato

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

### Esempio



L'operazione MCRA attiva la funzione MCR fino all'MCRD successivo. Le operazioni comprese tra MCR< e MCR> vengono eseguite in dipendenza del bit MA (in questo caso E 0.0):

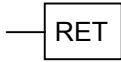
- Se E 0.0 = 1:
  - A 4.0 viene impostato su 1 se E 0.3 = 1
  - A 4.0 non viene impostato su 1 se E 0.3 = 0
  - Lo stato di segnale di E 0.4 viene assegnato all'uscita A 4.1.
- Se E 0.0 = 0:
  - A 4.0 non viene modificato indipendentemente dallo stato di segnale di E 0.3
  - A 4.0 ha valore 0 indipendentemente dallo stato di segnale di E 0.4

L'utente deve programmare nei blocchi lo stato di dipendenza delle funzioni (FC) e dei blocchi funzionali (FB) dall'MRC. Se la funzione o il blocco funzionale vengono richiamati da una sequenza MCRA-MCRD, le istruzioni all'interno di questa sequenza non dipendono tutte automaticamente dal bit MCR. Utilizzare a questo scopo l'operazione MCRA del blocco richiamato.



## 10.13 RET: Salto indietro

### Simbolo



### Descrizione

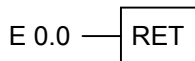
L'utente può adoperare l'operazione *Salta indietro* per uscire dai blocchi. Si può uscire da un blocco in modo condizionato.

### Parola di stato

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

\* L'operazione *RET* deriva internamente da una sequenza "SAVE; BEB;". Ciò significa che anche il bit BIE viene influenzato.

### Esempio



Si esce dal blocco 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 anche possibilità di utilizzare le operazioni di scorrimento per trasferire il contenuto dell'ingresso IN bit per bit, a sinistra o a destra (vedere anche 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.

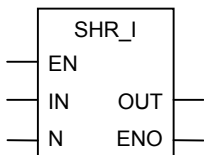
Il numero che l'utente fornisce per il parametro di ingresso N sta ad indicare il numero di bit nella misura in cui si deve effettuare lo scorrimento. I posti bit che vengono lasciati vuoti dall'operazione di scorrimento vengono colmati con degli zero, oppure con lo stato di segnale del bit del segno (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 destra (a 16 bit)
- SHR\_DI: Fai scorrere numero intero a destra (a 32 bit)
- SHL\_W: Fai scorrere parola a sinistra (a 16 bit)
- SHR\_W: Fai scorrere parola a destra
- SHL\_DW: Fai scorrere doppia parola a sinistra
- SHR\_DW: Fai scorrere doppia parola a destra

11.1.2 SHR\_I: Fai scorrere numero intero a destra (a 16 bit)

Simbolo

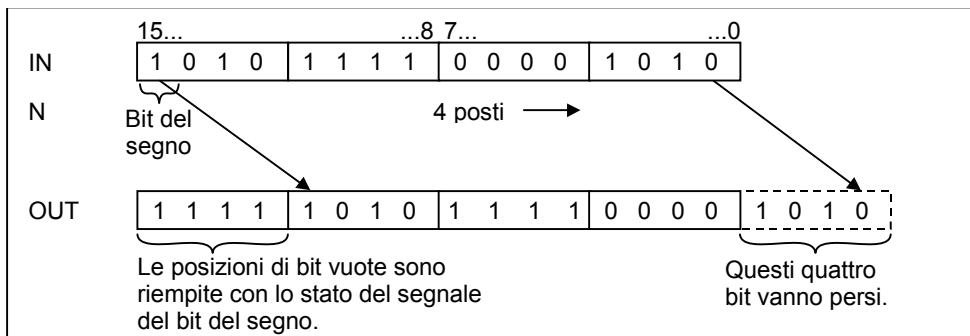


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso di abilitazione
IN	INT	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit da far scorrere
OUT	INT	E, A, M, L, D	Risultato dell'operazione di scorrimento
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione

Descrizione

Uno stato di segnale di 1 all'ingresso di abilitazione (EN) attiva l'operazione **Fai scorrere numero intero a 16 bit a destra**. Tale operazione fa scorrere a destra, bit per bit, i bit da 0 a 15 dell'ingresso IN. L'ingresso N specifica il numero di bit nella misura in cui bisogna effettuare lo scorrimento. Se N è maggiore di 16, il comando si comporta come se N fosse 16. Le posizioni dei bit sulla sinistra vengono riempite a seconda dello stato del segnale del bit 15 (il quale rappresenta il segno di un numero intero), vale a dire che esse vengono colmate di zero se il numero è positivo, e di uno se il numero è negativo. Il risultato dell'operazione di scorrimento può essere rilevato all'uscita OUT.

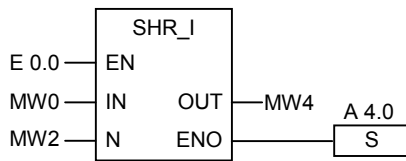
Con N diverso da 0, l'operazione avviata da tale istruzione resetta sempre a "0" i bit A0 e OV della parola di stato. ENO ha lo stesso stato di segnale di EN.



**Parola di stato**

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

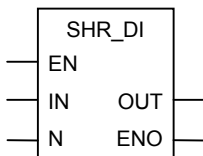
**Esempio**



L'operazione viene attivata se E 0.0 = 1.  
 La parola di merker MW0 viene fatta scorrere a destra del numero di bit indicato in MW2.  
 Il risultato viene memorizzato in MW4 L'uscita A 4.0 viene impostata a "1".

### 11.1.3 SHR\_DI: Fai scorrere numero intero a destra (a 32 bit)

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso 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 lo da far scorrere
OUT	DINT	E, A, M, L, D	Risultato dell'operazione di scorrimento
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione

#### Descrizione

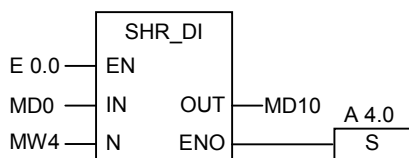
Uno stato di segnale di 1 all'ingresso di abilitazione (EN) attiva l'operazione **Fai scorrere numero intero a 32 bit a destra**. Tale operazione fa scorrere a destra, bit per bit, l'intero contenuto dell'ingresso IN. L'ingresso N specifica il numero di bit nella misura in cui bisogna effettuare lo scorrimento. Se N è maggiore di 32, il comando si comporta come se N fosse 32. Le posizioni di bit sulla sinistra vengono riempite secondo lo stato di segnale del bit 15 (il quale rappresenta il segno di un numero intero), vale a dire che esse vengono colmate di zero se il numero è positivo, e di uno se il numero è negativo. Il risultato dell'operazione di scorrimento può essere rilevato all'uscita OUT.

L'operazione avviata da tale istruzione resetta sempre a 0 il bit OV e il bit A0 della parola di stato. 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



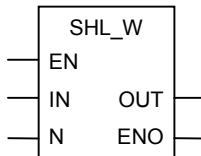
L'operazione viene attivata se E 0.0 = 1.

La doppia parola di merker MD0 viene fatta scorrere a destra del numero di bit indicato in MW4.

Il risultato viene memorizzato in MD10 L'uscita A 4.0 viene impostata a "1".

### 11.1.4 SHL\_W: Fai scorrere parola a sinistra (a 16 bit)

#### Simbolo



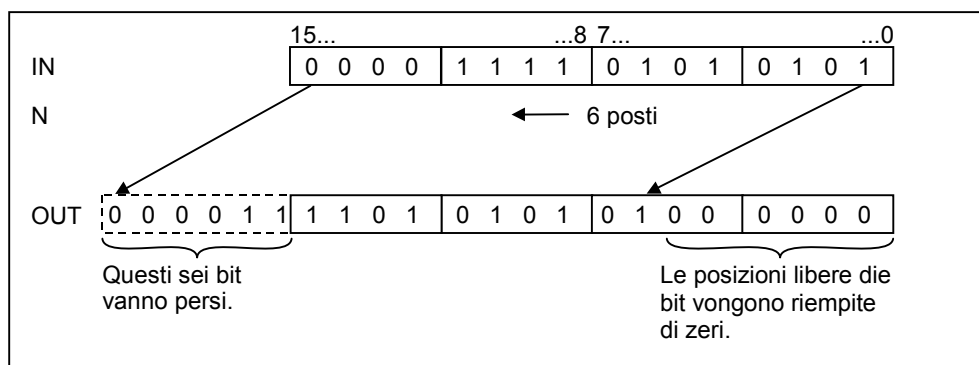
Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso di abilitazione
IN	WORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit da far scorrere
OUT	WORD	E, A, M, L, D	Risultato dell'operazione di scorrimento
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione

#### Descrizione

Uno stato di segnale di "1" all'ingresso di abilitazione (EN) attiva l'operazione **Fai scorrere parola a sinistra** che fa scorrere a sinistra, bit per bit, i bit da 0 a 15 dell'ingresso IN.

L'ingresso N specifica il numero di bit che vengono spostati. Se N è maggiore di 16, il comando scrive uno 0 nell'uscita OUT e resetta a "0" i bit A0 e OV della parola di stato. Le posizioni dei bit sulla destra vengono riempite di zero. Il risultato dell'operazione di scorrimento può essere rilevato all'uscita OUT.

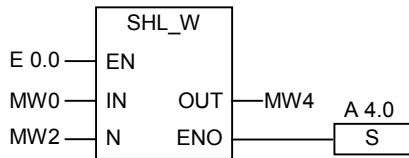
L'operazione avviata da tale istruzione resetta sempre a "0" i bit A0 e OV della parola di stato. ENO ha lo stesso stato di segnale di EN.



**Parola di stato**

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

**Esempio**



L'operazione viene attivata se E 0.0 = 1.

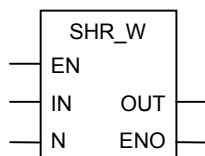
La parola di merker MW0 viene fatta scorrere a sinistra del numero di bit indicato in MW2.

Il risultato viene memorizzato in MW4. L'uscita A 4.0 viene impostata a "1".



### 11.1.5 SHR\_W: Fai scorrere parola a destra

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso 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 da far scorrere
OUT	WORD	E, A, M, L, D	Risultato dell'operazione di scorrimento
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione

#### Descrizione

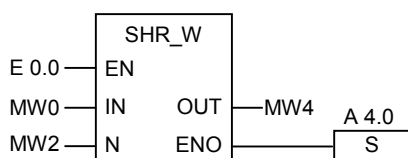
Uno stato di segnale di 1 all'ingresso di abilitazione (EN) attiva l'operazione **Fai scorrere parola a destra**. Tale operazione fa scorrere a destra, bit per bit, i bit da 0 a 15 dell'ingresso IN. Non sono interessati i bit da 16 a 31. L'ingresso IN specifica quanti bit si devono far scorrere. Se N è maggiore di 16, il comando scrive uno 0 nell'uscita OUT e resetta a 0 i bit A0 e OV. Le posizioni libere dei bit alla sinistra vengono riempite di zero. Il risultato dell'operazione di scorrimento può essere rilevato all'uscita OUT.

Con N diverso da 0, l'operazione avviata da tale istruzione resetta sempre a "0" i bit A0 e OV della parola di stato. 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



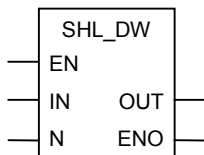
L'operazione viene attivata se E 0.0 = 1.

La parola di merker MW0 viene fatta scorrere a destra del numero di bit indicato in MW2.

Il risultato viene memorizzato in MW4. L'uscita A 4.0 viene impostata a "1".

### 11.1.6 SHL\_DW: Fai scorrere doppia parola a sinistra

#### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso 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 da far scorrere
OUT	DWORD	E, A, M, L, D	Risultato dell'operazione di scorrimento
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione

#### Descrizione

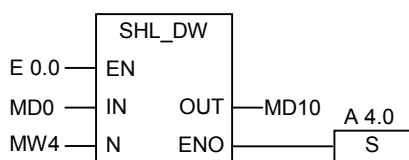
Uno stato di segnale di 1 all'ingresso di abilitazione (EN) attiva l'operazione **Fai scorrere doppia parola a sinistra**. Tale operazione fa scorrere a sinistra, bit per bit, i bit da 0 a 31 dell'ingresso IN. L'ingresso IN specifica quanti bit si devono far scorrere. Se N è maggiore di 32, il comando scrive uno 0 e resetta a 0 i bit A0 e OV. Le posizioni libere di bit sulla destra sono riempite di zeri. Il risultato dell'operazione di di scorrimento può essere rilevato all'uscita OUT.

Con N diverso da 0, l'operazione avviata da tale istruzione resetta sempre a "0" i bit A0 e OV della parola di stato. 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

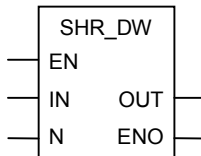


L'operazione viene attivata se E 0.0 = 1.

La doppia parola di merker MD0 viene fatta scorrere a sinistra del numero di bit indicato in MW4. Il risultato viene memorizzato in MD10. L'uscita A 4.0 viene impostata a "1".

### 11.1.7 SHR\_DW: Fai scorrere doppia parola a destra

#### Simbolo

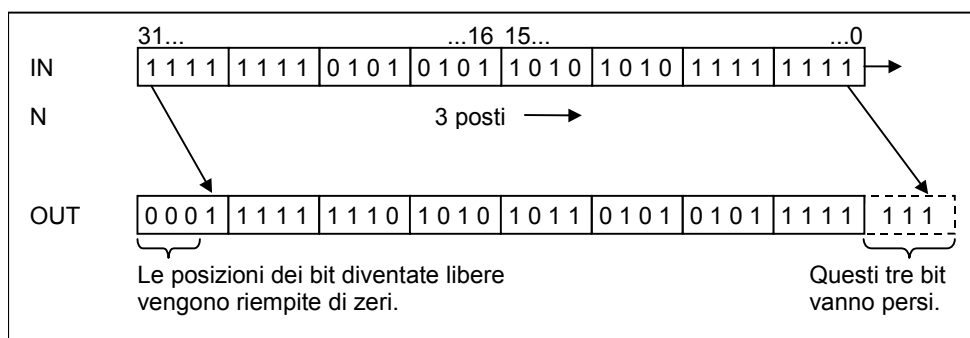


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso di abilitazione
IN	DWORD	E, A, M, L, D	Valore da far scorrere
N	WORD	E, A, M, L, D	Numero di posizioni di bit da far scorrere
OUT	DWORD	E, A, M, L, D	Risultato dell'operazione di scorrimento
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione

#### Descrizione

Uno stato di segnale di 1 all'ingresso di abilitazione (EN) attiva l'operazione **Fai scorrere doppia parola a destra**. Tale operazione fa scorrere a destra, bit per bit, i bit da 0 a 31 dell'ingresso IN. L'ingresso N specifica il numero di bit nella misura in cui bisogna effettuare lo scorrimento. Se N è maggiore di 32, il comando scrive uno 0 nell'uscita OUT e resetta a 0 i bit A0 e OV. Le posizioni dei bit sulla sinistra vengono riempite di zero. Il risultato dell'operazione di scorrimento può essere rilevato all'uscita OUT.

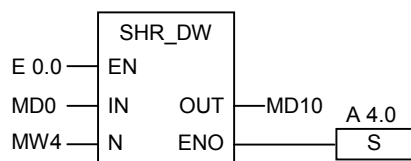
Con N diverso da 0, l'operazione avviata da tale istruzione resetta sempre a "0" i bit A0 e OV della parola di stato. ENO ha lo stesso stato di segnale di EN.



#### Parola di stato

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

**Esempio**



L'operazione viene attivata se E 0.0 = 1.

La parola di merker MD0 viene fatta scorrere a destra del numero di bit indicato in MW4.

Il risultato viene memorizzato in MD10 L'uscita A 4.0 viene impostata a "1".

## 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 il contenuto dell'ingresso IN a sinistra o a destra, bit per bit (vedere anche Registri CPU). Tuttavia, i posti bit liberi vengono occupati dagli stati di segnale dei bit spostati dall'ingresso IN.

Il numero che l'utente fornisce per il parametro di ingresso N sta a specificare il numero dei bit da far ruotare.

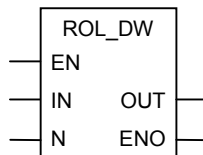
A seconda dall'operazione, la rotazione avviene tramite il bit A1. Il bit A0 della parola di stato viene resettato a 0.

Sono disponibili le seguenti operazioni di rotazione:

- ROL\_DW: Fai ruotare doppia parola a sinistra (a 16 bit)
- ROR\_DW: Fai ruotare doppia parola a destra (a 32 bit)

### 11.2.2 ROL\_DW: Fai ruotare doppia parola a sinistra (a 16 bit)

#### Simbolo

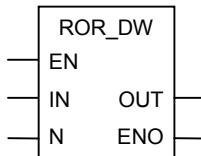


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso di abilitazione
IN	DWORD	E, A, M, L, D	Valore da far ruotare
N	WORD	E, A, M, L, D	Numero di posizioni di bit da far ruotare
OUT	DWORD	E, A, M, L, D	Risultato dell'operazione di rotazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione



### 11.2.3 ROR\_DW: Fai ruotare doppia parola a destra (a 32 bit)

#### Simbolo

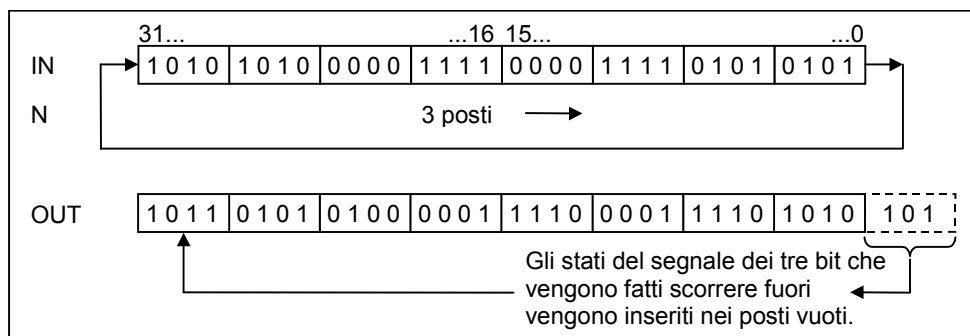


Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, L, D, T, Z	Ingresso di abilitazione
IN	DWORD	E, A, M, L, D	Valore da far ruotare
N	WORD	E, A, M, L, D	Numero di posizioni di bit da far ruotare
OUT	DWORD	E, A, M, L, D	Risultato dell'operazione di rotazione
ENO	BOOL	E, A, M, L, D	Uscita di abilitazione

#### Descrizione

Uno stato di segnale di 1 all'ingresso di abilitazione (EN) attiva l'operazione **Fai ruotare doppia parola a destra**. Tale operazione fa ruotare a destra, bit per bit, l'intero contenuto dell'ingresso IN. L'ingresso N specifica il numero di bit nella misura in cui operare la rotazione. Se N è maggiore di 32, la doppia parola viene ruotata di  $((N-1) \text{ modulo } 32) + 1$  posti. Le posizioni dei bit sulla sinistra vengono riempite con gli stati di segnale dei bit oggetto della rotazione. Il risultato dell'operazione di rotazione può essere rilevato all'uscita OUT.

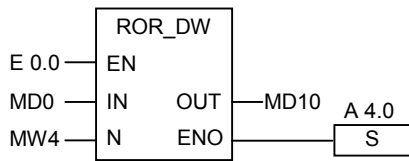
Con N diverso da 0, l'operazione avviata da tale istruzione resetta sempre a 0 il bit OV e il bit A0 della parola di stato. ENO ha lo stesso stato di segnale di EN.



**Parola di stato**

	<b>BIE</b>	<b>A1</b>	<b>A0</b>	<b>OV</b>	<b>OS</b>	<b>OR</b>	<b>STA</b>	<b>RLC</b>	<b>/ER</b>
scrive	X	X	X	X	-	X	X	X	1

**Esempio**



L'operazione viene attivata se E 0.0 = 1.

La doppia parola di merker MD0 viene fatta ruotare a destra del numero di bit indicato in MW4.

Il risultato viene memorizzato in MW10. L'uscita A 4.0 viene impostata a "1".



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

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 contenente i bit a cui si può fare riferimento nell'operando di operazioni logiche combinatorie a bit e a parola.

Struttura della parola di stato:

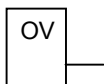
$2^{15}$ ...	...	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
		BIE	A1	A0	OV	OS	OR	STA	RLC	/ER	

È 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

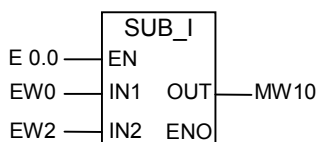
Si può adoperare l'operazione **Bit di anomalia: overflow** per capire se si è verificata un'eccedenza (OV) in un'operazione matematica. Se, una volta eseguita l'operazione matematica, si ottiene un risultato che non è compreso nel campo negativo o positivo ammesso, viene impostato il bit OV nella parola di stato (vedere anche Registri CPU). L'operazione interroga lo stato di segnale di questo bit che viene resettato quando le operazioni aritmetiche vengono eseguite senza errori.

### Parola di stato

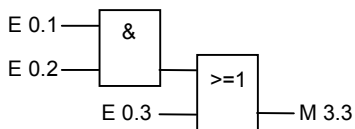
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	X	X	X	1

### Esempio

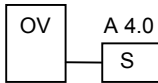
#### Segmento 1



#### Segmento 2



### Segmento 3

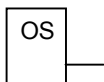


Il box SUB\_I viene attivato se  $E 0.0 = 1$ . Se il valore dell'operazione aritmetica EW0 - EW2 non è compreso nel campo ammesso per i numeri interi, viene impostato il bit OV. L'interrogazione dello stato di OV dà come risultato "1". L'uscita A 4.0 viene impostata se l'interrogazione di OV è "1" e l'RLC del segmento 2 è "1" (ovvero se l'RLC che precede l'uscita A 4.0 = 1).

Se lo stato di segnale dell'ingresso  $E 0.0 = 0$  (disattivato), lo stato di segnale di EN e ENO è 0. Se lo stato di segnale di EN = 1 (attivato) e il risultato dell'operazione non è compreso nel campo ammesso, lo stato di segnale di ENO = 0.

## 12.3 OS: Bit di anomalia: overflow con memoria

### Simbolo



### Descrizione

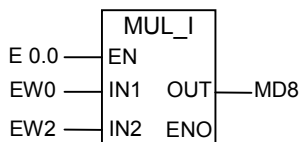
Si può adoperare un'operazione **Bit di anomalia: overflow con memoria** per riconoscere un'eccedenza a ritenzione OS in un'operazione matematica (eccedenza a ritenzione, OS). Se, dopo l'operazione matematica, il risultato è al di fuori del campo negativo o positivo ammesso, viene settato anche il bit OS della parola di stato (vedere il capitolo Registri CPU). L'operazione interroga lo stato di segnale di questo bit. A differenza del bit OV (overflow), il bit OS resta settato anche se l'operazione aritmetica viene eseguita senza errori.

### Parola di stato

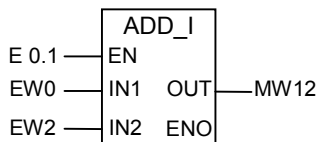
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	X	X	X	1

### Esempio

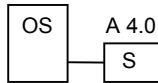
#### Segmento 1



#### Segmento 2



## Segmento 3



Il box MUL\_I viene attivato se E 0.0 = 1 e il box ADD\_I viene attivato se E 0.1 = 1. Se uno dei risultati non è compreso nel campo ammesso per i numeri interi, viene impostato il bit OS.

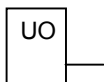
L'interrogazione dello stato di OS dà come risultato "1" e viene impostata l'uscita A 4.0.

Segmento 1: se lo stato di segnale dell'ingresso E 0.0 = 0 (disattivato), lo stato di segnale di EN e ENO è 0. Se lo stato di segnale di EN = 1 (attivato) e il risultato dell'operazione non è compreso nel campo ammesso, lo stato di segnale di ENO = 0.

Segmento 2: se lo stato di segnale dell'ingresso E 0.1 = 0 (disattivato), lo stato di segnale di EN e ENO è 0. Se lo stato di segnale di EN = 1 (attivato) e il risultato dell'operazione non è compreso nel campo ammesso, lo stato di segnale di ENO = 0.

## 12.4 UO: Bit di anomalia: operazione non ammessa

### Simbolo



### Descrizione

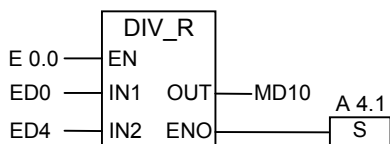
Si può adoperare l'operazione **Bit di anomalia: operazione non ammessa** per stabilire se il risultato di un'operazione matematica in virgola mobile è "non ammesso" (ovvero, se uno dei valori nella operazione matematica non è un numero in virgola mobile ammesso). A questo fine si valutano i bit di visualizzazione (A1 e A0, vedere anche Registri CPU) nella parola di stato. Se il risultato di un'operazione matematica è "non ammesso" (UO), il risultato dell'interrogazione dello stato del segnale è 1. Se la combinazione in A1 e A0 non indica "non ammesso" allora il risultato dell'interrogazione dello stato del segnale sarà 0.

### Parola di stato

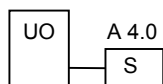
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	X	X	X	1

### Esempio

#### Segmento 1



#### Segmento 2



Il box DIV\_R viene attivato se E 0.0 = 1. Se il valore di ED0 o ED4 non è un numero in virgola mobile ammesso, l'operazione aritmetica non è valida.

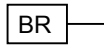
Se lo stato di segnale di EN = 1 (attivato) e durante l'elaborazione della funzione DIV\_R si verifica un errore, lo stato di segnale di ENO = 0.

L'uscita A 4.0 viene impostata quando viene eseguita la funzione DIV\_R e se uno dei valori dell'operazione aritmetica è un numero in virgola mobile non ammesso. Se lo stato di segnale dell'ingresso E 0.0 = 0 (disattivato), lo stato di segnale di EN e ENO è 0.

## 12.5 BIE: Registro BIE bit di anomalia

### Simbolo

Inglese



Tedesco



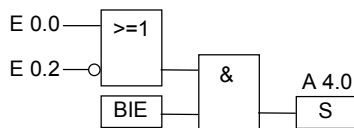
### Descrizione

Si può usare l'operazione **Registro BIE: bit di anomalia** per interrogare lo stato di segnale del bit BIE (risultato binario) (vedere anche Registri CPU).

### Parola di stato

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

### Esempio



L'uscita A 4.0 viene impostata se  $E 0.0 = 1$  OR  $E 0.2 = 0$  e, oltre a tale RLC, lo stato di segnale del bit BIE è uguale a 1.

## 12.6 <> 0: Bit di risultato

### Simboli

<code>== 0</code>	L'operazione Bit di risultato <b>uguale a 0</b> determina se il risultato di un'operazione aritmetica è uguale a 0.
<code>&lt;&gt; 0</code>	L'operazione Bit di risultato <b>diverso da 0</b> determina se il risultato di un'operazione aritmetica è diverso da 0.
<code>&gt; 0</code>	L'operazione Bit di risultato <b>maggiore di 0</b> determina se il risultato di un'operazione aritmetica è maggiore di 0.
<code>&lt; 0</code>	L'operazione Bit di risultato <b>minore di 0</b> determina se il risultato di un'operazione aritmetica è minore di 0.
<code>&gt;= 0</code>	L'operazione Bit di risultato <b>maggiore o uguale a 0</b> determina se il risultato di un'operazione aritmetica è maggiore o uguale a 0.
<code>&lt;= 0</code>	L'operazione Bit di risultato <b>minore o uguale a 0</b> determina se il risultato di un'operazione aritmetica è minore o uguale a 0.

### Descrizione

Si possono adoperare le operazioni di **Bit di risultato** per determinare come si comporta il risultato di un'operazione matematica rispetto allo zero, ovvero se è `==0`, `<>0`, `>0`, `<0`, `>=0`, `<=0`. A questo scopo vengono valutati i bit di visualizzazione della parola di stato (A1 e A0, vedere anche Registri CPU). Se la condizione di confronto indicata nell'operando viene soddisfatta, il risultato dell'interrogazione dello stato di segnale è 1.

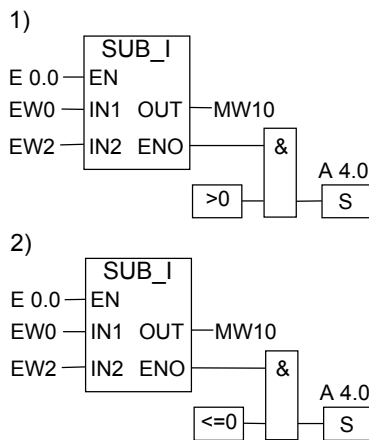
In un'operazione **AND**, le operazioni di bit di stato combinano il risultato della propria interrogazione con il precedente risultato logico combinatorio in base alla tabella della verità AND. In un'operazione **OR**, ciò avviene in base alla tabella della verità OR.

### Parola di stato

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



## Esempio



Il box SUB\_I viene attivato se E 0.0 = 1. Se il valore di EW0 è maggiore del valore di EW2, il risultato dell'operazione aritmetica EW0 - EW2 è superiore a 0.

Se lo stato di segnale di EN = 1 e durante l'elaborazione della funzione SUB\_1 si verifica un errore, lo stato di segnale di ENO = 0.

- 1) L'uscita A 4.0 viene impostata se la funzione viene eseguita senza errori e il risultato è maggiore di 0. Se lo stato di segnale dell'ingresso E 0.0 = 0 (disattivato), lo stato di segnale di EN e ENO è 0.
- 2) L'uscita A 4.0 viene impostata se la funzione viene eseguita senza errori e il risultato è inferiore o uguale a 0. Se lo stato di segnale dell'ingresso E 0.0 = 0 (disattivato), lo stato di segnale di EN e ENO è 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: Avviamento e parametrizzazione del temporizzatore come impulso
  - S\_VIMP: Avviamento e parametrizzazione del temporizzatore come impulso prolungato
  - S\_EVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo all'inserzione
  - S\_SEVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo all'inserzione con memoria
  - S\_AVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo alla disinserzione
- 
- SI: Avvia temporizzatore come impulso
  - SV: Avvia temporizzatore come impulso prolungato
  - SE: Avvia temporizzatore come ritardo all'inserzione
  - SS: Avvia temporizzatore come ritardo all'inserzione con memoria
  - SA: Avvia 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 FUP supporta 256 temporizzatori. Le parole di temporizzazione 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 valore viene decrementato finché non diventa uguale a zero.

Si può precaricare un valore di tempo con la seguente sintassi:

- S5T#aH\_bbM\_ccS\_dddMS
  - laddove a = ore, bb = minuti, cc = secondi, e ddd = millisecondi
  - 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.

## 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 in cui il valore di tempo viene decrementato 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

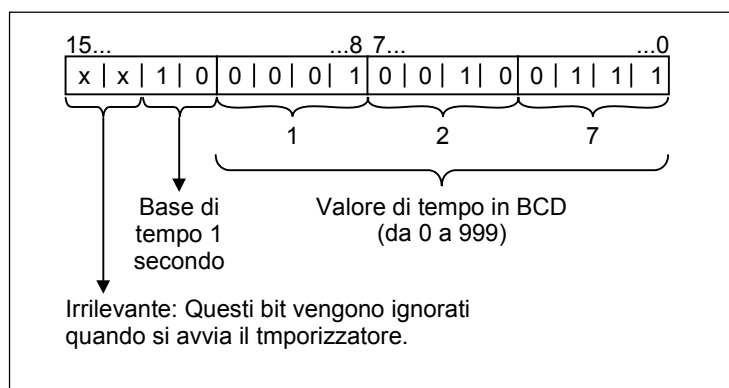
Siccome i valori di tempo vengono salvati con un solo intervallo di tempo, i valori che non sono multipli esatti di un intervallo di tempo vengono arrotondati per difetto. I valori con una risoluzione eccessiva rispetto al campo desiderato vengono arrotondati per difetto per ottenere il campo desiderato ma non la risoluzione desiderata. La seguente tabella mostra le risoluzioni possibili e i loro campi corrispondenti.

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 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.

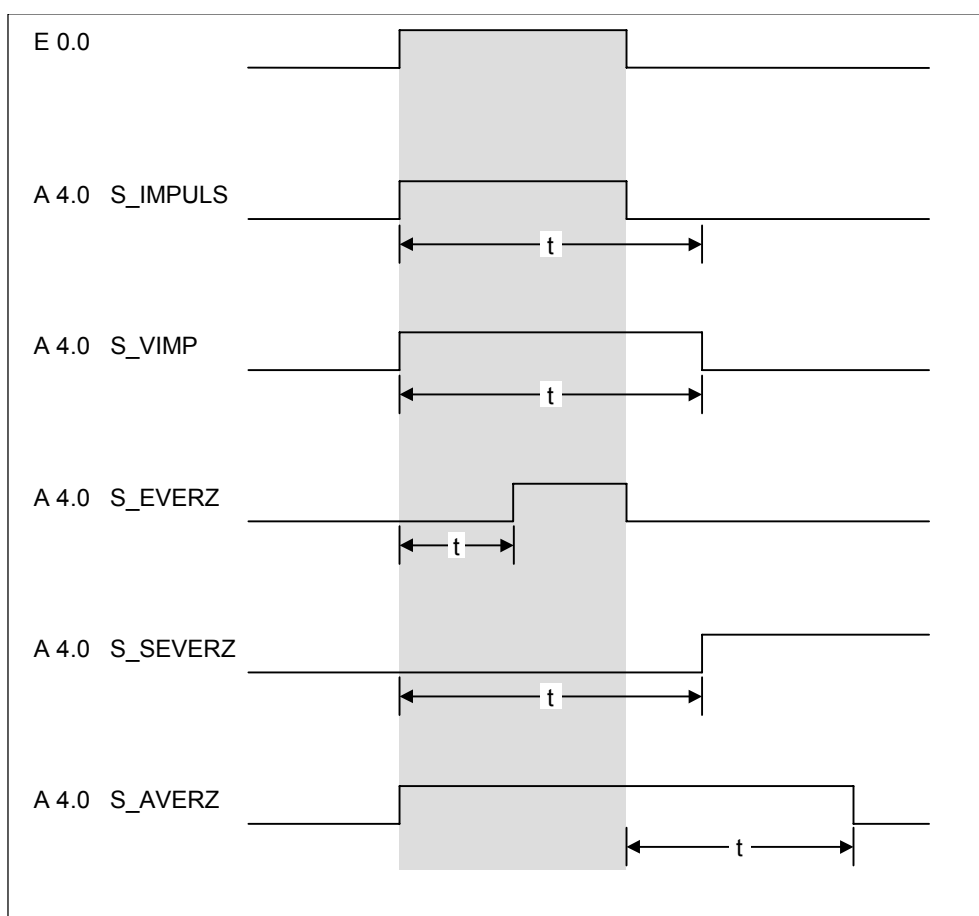


### Letture dell'ora e della base di tempo

Ogni box di temporizzatore contiene due uscite, DUAL e DEZ, per cui si può indicare un indirizzo di parola. Nell'uscita DUAL il valore di tempo è codificato in formato binario, la base di tempo non viene visualizzata. Nell'uscita DEZ la base di tempo e il valore di tempo sono in formato decimale codificato in binario (BCD).

### Scelta del giusto temporizzatore

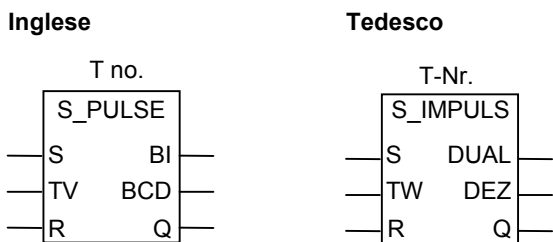
La figura illustra cinque tipi di temporizzatori che sono stati descritti in questo capitolo. Questo riepilogo vuole essere d'aiuto all'utente nella scelta del temporizzatore più adeguato ai suoi fini di utilizzo.



Temporizzatore	Descrizione
<b>S_IMPULS</b> 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.
<b>S_VIMP</b> 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.
<b>S_EVERZ</b> 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.
<b>S_SEVERZ</b> 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.
<b>S_AVERZ</b> 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: Avviamento e parametrizzazione del temporizzatore come impulso

**Simbolo**



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
no.	Nr.	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU.
S	S	BOOL	E, A, M, D, L, T, Z	Ingresso di avvio
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo preimpostato (campo 0 - 9999)
R	R	BOOL	E, A, M, D, L, T, Z	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, D, L	Valore di tempo rimanente (formato di num. int.)
BCD	DEZ	WORD	E, A, M, D, L	Valore di tempo rimanente (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Stato del temporizzatore

**Descrizione**

L'operazione **Avvia e parametrizza temporizzatore come impulso** avvia un temporizzatore specificato, se all'ingresso di avviamento (S) si trova un fronte di salita (ossia se lo stato del segnale passa da 0 a 1). Per abilitare un temporizzatore è necessario che lo stato di segnale cambi. Il temporizzatore continua ad operare con il valore specificato all'ingresso Valore di tempo (TW), finché non trascorre il tempo programmato. Mentre il temporizzatore è attivo, l'interrogazione dello stato di segnale all'uscita Q produce un risultato di 1. Se avviene un cambiamento da 1 a 0 all'ingresso S prima che il tempo sia trascorso, il temporizzatore si arresta. In questo caso, un'interrogazione dello stato di segnale all'uscita Q, per verificare se questo è uguale a 1, produce un risultato di 0.

Il temporizzatore viene resettato se si verifica un cambiamento da 0 a 1 all'ingresso di resettaggio (R) mentre esso è in funzionamento. Questo cambiamento, quindi, resetta a 0 il temporizzatore e la base di tempo. Lo stato di segnale 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 DUAL e DEZ. Il valore di tempo all'uscita DUAL è in formato di cifra in codice binario, e all'uscita DEZ è in formato decimale codificato in binario.

Vedere anche Aree di memoria e componenti di un temporizzatore.

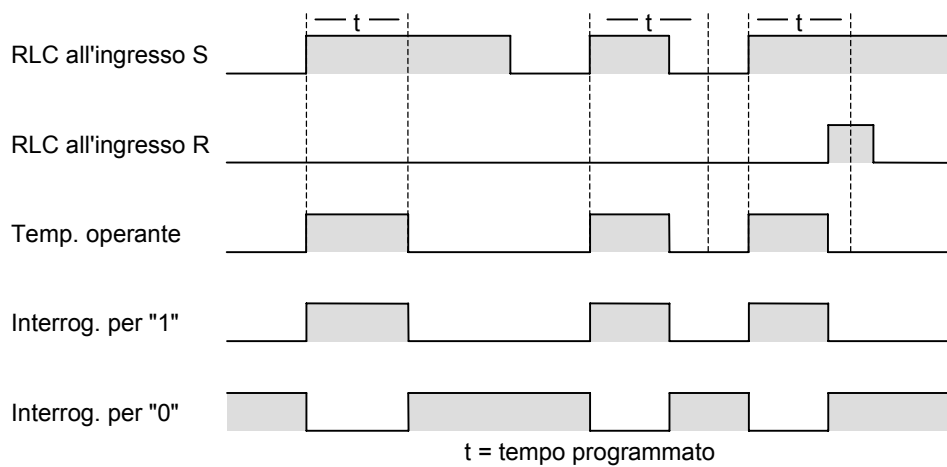


**Parola di stato**

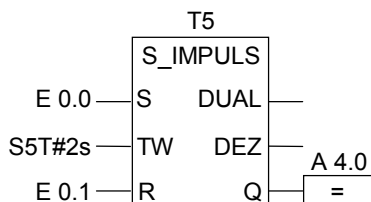
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	X	X	X	1

**Diagramma degli impulsi**

Caratteristiche del temporizzatore come impulso:



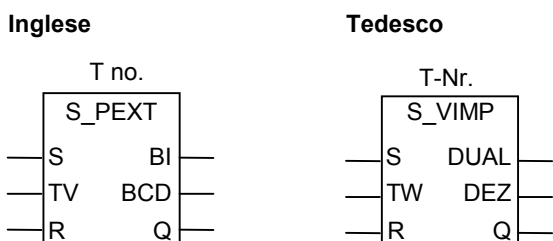
**Esempio**



Se lo stato di segnale dell'ingresso E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Quando E 0.0 = 1, il temporizzatore funziona con il valore indicato di 2 secondi (2 s). Se lo stato di segnale di E 0.0 passa da 1 a 0 prima che siano trascorsi 2 secondi, il temporizzatore viene arrestato. Se lo stato di segnale di E 0.1 passa da 0 a 1 durante il funzionamento del temporizzatore, quest'ultimo viene resettato. L'uscita A 4.0 resta 1 quando il temporizzatore è attivo.

### 13.4 S\_VIMP: Avviamento e parametrizzazione del temporizzatore come impulso prolungato

**Simbolo**



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
no.	Nr.	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU.
S	S	BOOL	E, A, M, D, L, T, Z	Ingresso di avvio
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo preimpostato (campo 0 - 9999)
R	R	BOOL	E, A, M, D, L, T, Z	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, D, L	Valore di tempo rimanente (formato di num. int.)
BCD	DEZ	WORD	E, A, M, D, L	Valore di tempo rimanente (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Stato del temporizzatore

**Descrizione**

L'operazione **Avvia e parametrizza temporizzatore come impulso prolungato** avvia un temporizzatore specificato, se all'ingresso di avviamento (S) si trova un fronte di salita (ossia lo stato del segnale passa da 0 a 1). Per abilitare un temporizzatore è necessario che lo stato di segnale cambi. Il temporizzatore continua ad operare con il valore specificato all'ingresso Valore di tempo (TW), anche se lo stato di segnale all'ingresso S diventa 0 prima che il tempo sia trascorso.

Un'interrogazione dello stato di segnale per verificare se questo è uguale a 1 all'uscita Q produce un risultato di 1, finché il temporizzatore è operante. 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 operante.

Il temporizzatore viene resettato se si verifica un cambiamento da 0 a 1 all'ingresso di resettaggio (R), mentre esso è attivo. Questo cambiamento, quindi, resetta a 0 il temporizzatore e la base di tempo.

Il valore di tempo attuale può essere letto alle uscite DUAL e DEZ. Il valore di tempo all'uscita DUAL è in formato di cifra in codice binario e all'uscita DEZ è in formato di cifra decimale in codice binario.

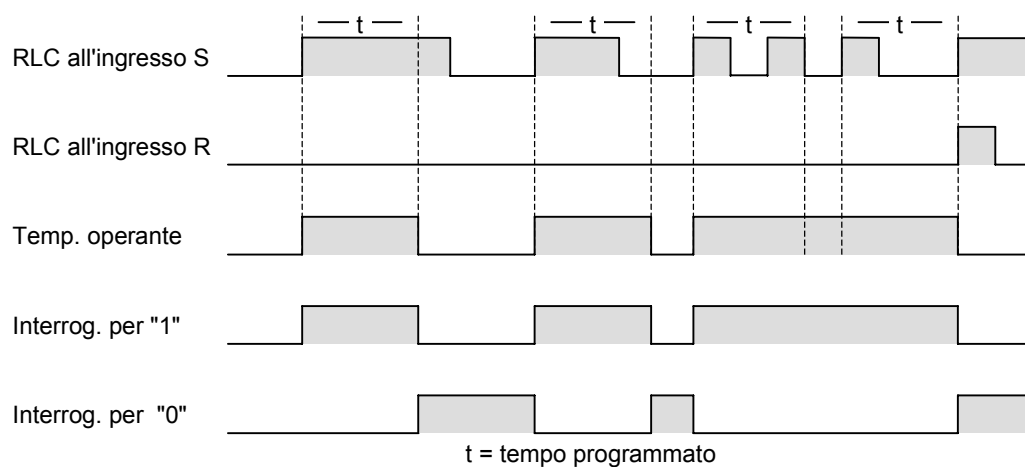
Vedere anche Aree di memoria e componenti di un temporizzatore.

## Parola di stato

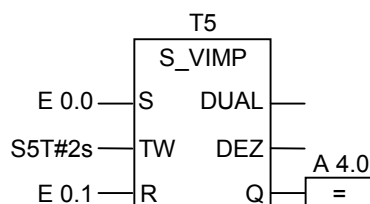
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	X	X	X	1

## Diagramma degli impulsi

Caratteristiche del temporizzatore come impulso prolungato:



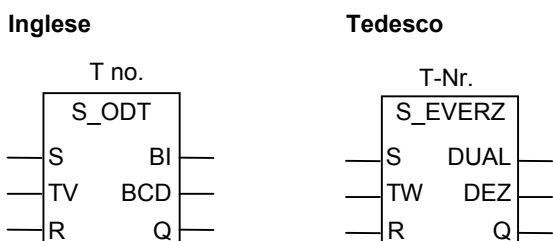
## Esempio



Se lo stato di segnale dell'ingresso E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Il temporizzatore funziona con il valore indicato di 2 secondi (2 s) indipendentemente dal fronte di discesa all'ingresso S. Se lo stato di segnale di E 0.0 passa da 0 a 1 prima che siano trascorsi 2 secondi, il temporizzatore viene riavviato. Se lo stato di segnale di E 0.0 passa da 0 a 1 durante il funzionamento del temporizzatore, quest'ultimo viene riavviato. L'uscita A 4.0 vale 1 quando il temporizzatore è attivo.

## 13.5 S\_EVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo all'inserzione

### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
no.	Nr.	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU.
S	S	BOOL	E, A, M, D, L, T, Z	Ingresso di avvio
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo preimpostato (campo 0 - 9999)
R	R	BOOL	E, A, M, D, L, T, Z	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, D, L	Valore di tempo rimanente (formato di num. int.)
BCD	DEZ	WORD	E, A, M, D, L	Valore di tempo rimanente (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Stato del temporizzatore

### Descrizione

L'operazione **Avvia e parametrizza temporizzatore come ritardo all'inserzione** avvia un temporizzatore specificato se all'ingresso di avvio (S) si trova un fronte di salita (ossia se lo stato del segnale passa da 0 a 1). Per avviare un temporizzatore è necessario che lo stato di segnale cambi. Il temporizzatore continua ad operare con il valore specificato all'ingresso Valore di tempo (TW), finché lo stato di segnale all'ingresso S rimane 1. Un'interrogazione dello stato di segnale per verificare se questo è uguale a 1 all'uscita Q produce un risultato di 1 se il tempo è trascorso e se lo stato di segnale all'ingresso S è sempre 1 e l'ingresso di resettaggio (R) rimane 0. 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, un'interrogazione di segnale per verificare se questo corrisponde a 1 produce sempre il risultato di 0.

Il temporizzatore viene resettato se si verifica un cambiamento da 0 a 1 all'ingresso di resettaggio (R), mentre esso è attivo. Questo cambiamento, quindi, resetta a 0 il temporizzatore e la base di tempo.

Il valore di tempo attuale può essere letto alle uscite DUAL e DEZ. Il valore di tempo all'uscita DUAL è in codice binario, il valore di tempo all'uscita DEZ è in formato decimale codificato in binario.

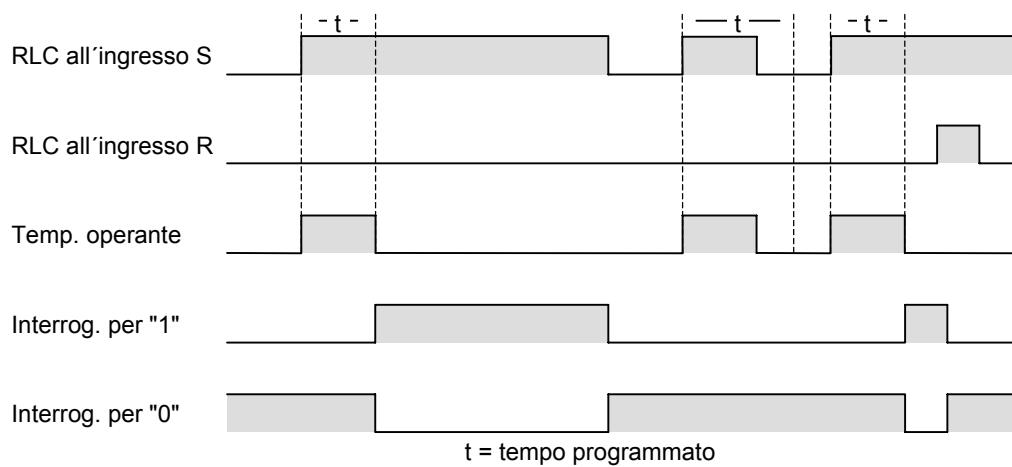
Vedere anche Aree di memoria e componenti di un temporizzatore.

**Parola di stato**

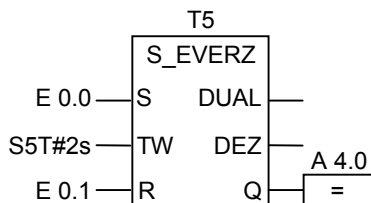
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	X	X	X	1

**Diagramma degli impulsi**

Caratteristiche del temporizzatore come ritardo all'inserzione:



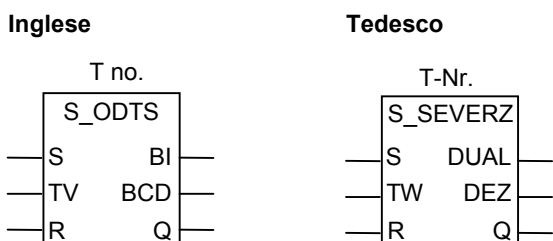
**Esempio**



Se lo stato di segnale dell'ingresso E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Se il tempo indicato di 2 secondi (2 s) è trascorso e lo stato di segnale di E 0.0 è ancora 1, lo stato del segnale dell'uscita A 0.4 = 1. Se lo stato di segnale di E 0.0 passa da 1 a 0, il temporizzatore viene arrestato e A 0.4 è 0. Se lo stato di segnale di E 0.0 passa da 0 a 1 durante il funzionamento del temporizzatore, quest'ultimo viene riavviato.

## 13.6 S\_SEVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo all'inserzione con memoria

### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
no.	Nr.	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU.
S	S	BOOL	E, A, M, D, L, T, Z	Ingresso di avvio
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo preimpostato (campo 0 - 9999)
R	R	BOOL	E, A, M, D, L, T, Z	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, D, L	Valore di tempo rimanente (formato di num. int.)
BCD	DEZ	WORD	E, A, M, D, L	Valore di tempo rimanente (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Stato del temporizzatore

### Descrizione

L'operazione **Avvia e parametrizza temporizzatore come ritardo all'inserzione con memoria** avvia un temporizzatore specificato se all'ingresso di avviamento (S) si trova un fronte di salita (ossia se lo stato del segnale passa da 0 a 1). Per avviare un temporizzatore è necessario che lo stato di segnale cambi. Il temporizzatore continua ad operare con il valore specificato all'ingresso Valore di tempo (TW), se lo stato di segnale all'ingresso S diventa 0 prima che sia trascorso il tempo.

Un'interrogazione dello stato di segnale per verificare se questo è uguale a 1 all'uscita Q, dopo che il tempo è trascorso, produce, indipendentemente dallo stato di segnale all'ingresso S, un risultato di 1, se l'ingresso di resettaggio (R) rimane 0. Il temporizzatore viene riavviato con il valore specificato se l'ingresso S passa da 0 a 1, mentre esso è attivo.

Il temporizzatore viene resettato se si verifica un cambiamento da 0 a 1 all'ingresso di resettaggio (R), mentre esso è attivo, senza considerare l'RLC all'ingresso.

Il valore di tempo attuale può essere letto alle uscite DUAL e DEZ. Il valore di tempo all'uscita DUAL è in codice binario, il valore di tempo all'uscita DEZ è in formato decimale codificato in binario.

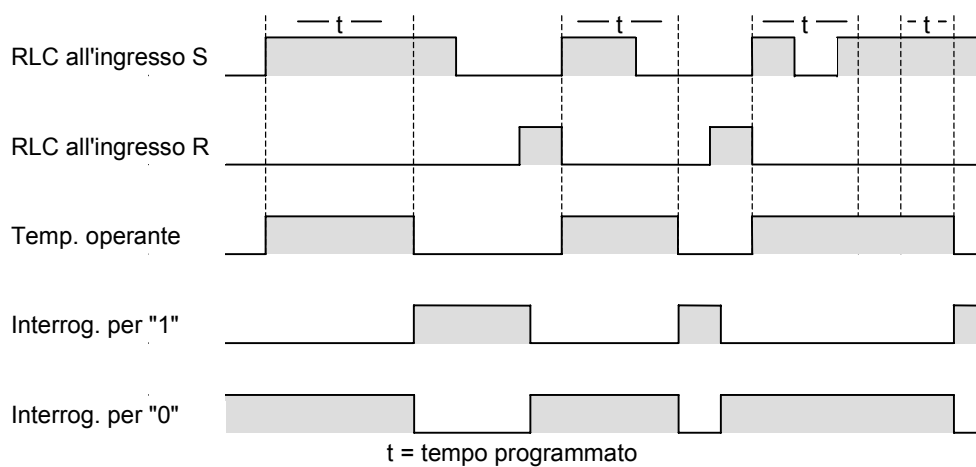
Vedere anche Aree di memoria e componenti di un temporizzatore.

## Parola di stato

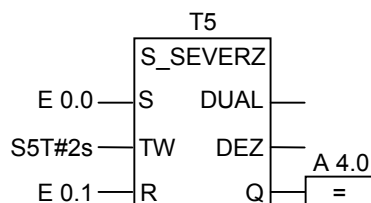
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	X	X	X	1

## Diagramma degli impulsi

Caratteristiche del temporizzatore come ritardo all'inserzione con memoria:



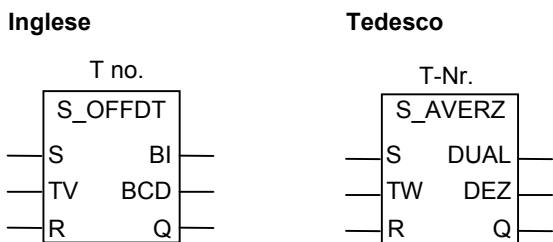
## Esempio



Se lo stato di segnale dell'ingresso E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Il temporizzatore resta attivo indipendentemente dal passaggio di E 0.0 da 1 a 0. Se lo stato di segnale di E 0.0 passa da 0 a 1 prima che sia trascorso il tempo indicato, il temporizzatore viene riavviato. Se lo stato di segnale di E 0.0 passa da 0 a 1 durante il funzionamento del temporizzatore, quest'ultimo viene riavviato. L'uscita A 4.0 è 1 se il tempo è trascorso ed E 0.1 resta su 0.

## 13.7 S\_AVERZ: Avviamento e parametrizzazione del temporizzatore come ritardo alla disinserzione

### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
no.	Nr.	TIMER	T	Numero di identificazione del temporizzatore. Il campo dipende dalla CPU.
S	S	BOOL	E, A, M, D, L, T, Z	Ingresso di avvio
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo preimpostato (campo 0 - 9999)
R	R	BOOL	E, A, M, D, L, T, Z	Ingresso di resettaggio
BI	DUAL	WORD	E, A, M, D, L	Valore di tempo rimanente (formato di num. int.)
BCD	DEZ	WORD	E, A, M, D, L	Valore di tempo rimanente (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Stato del temporizzatore

### Descrizione

L'operazione **Avvia e parametrizza temporizzatore come ritardo alla disinserzione** avvia un temporizzatore specificato, se all'ingresso di avviamento (S) si trova un fronte di discesa (ossia se lo stato del segnale è passato da 1 a 0). Per avviare un temporizzatore è necessario che lo stato di segnale cambi. Il risultato di una interrogazione di segnale per verificare se questo è 1, all'uscita Q è 1, se lo stato di segnale all'ingresso S è 1, oppure se il temporizzatore è operante. Il temporizzatore viene resettato quando lo stato di segnale all'ingresso S cambia da 0 a 1, mentre il temporizzatore è operante. Il temporizzatore non viene riavviato finché lo stato di segnale all'ingresso S non cambia nuovamente da 1 a 0.

Il temporizzatore viene resettato se si verifica un cambiamento da 0 a 1 all'ingresso di resettaggio (R), mentre esso è attivo.

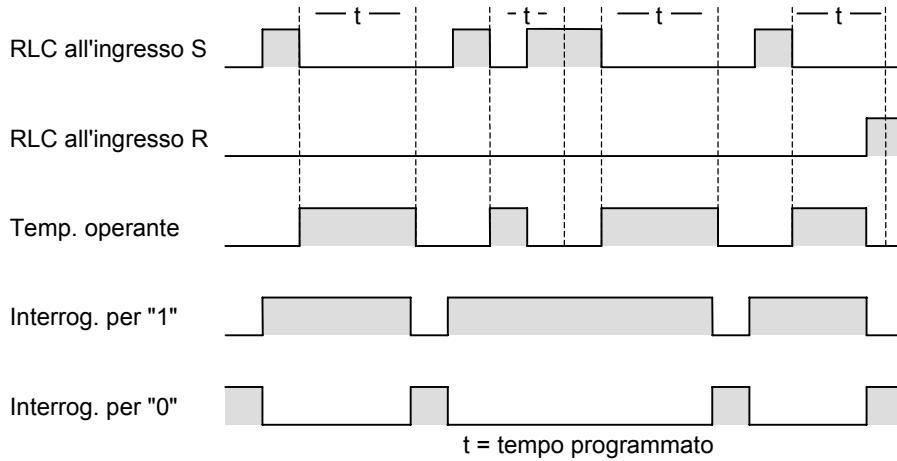
Il valore di tempo attuale può essere letto alle uscite DUAL e DEZ. Il valore di tempo all'uscita DUAL è in codice binario, il valore di tempo all'uscita DEZ è in formato decimale codificato in binario.

Vedere anche Aree di memoria e componenti di un temporizzatore.



### Diagramma degli impulsi

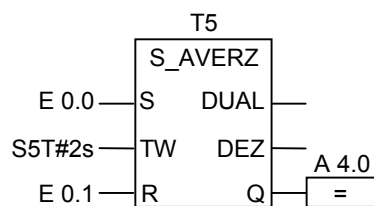
Caratteristiche del temporizzatore come ritardo alla disinserione:



### Parola di stato

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

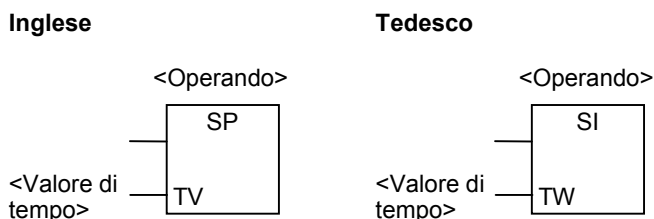
### Esempio



Se lo stato di segnale dell'ingresso E 0.0 passa da 0 a 1, il temporizzatore T5 viene avviato. L'uscita A 4.0 è 1 quando E 0.0 = 1 o il temporizzatore è attivo. Se lo stato di segnale di E 0.1 passa da 0 a 1 durante il funzionamento del temporizzatore, quest'ultimo viene riavviato.

### 13.8 SI: Avvia temporizzatore come impulso

#### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
Numero del temporizzatore	Numero del temporizzatore	TIMER	T	L'operando indica il temporizzatore da avviare.
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo (formato S5TIME)

#### Descrizione

L'operazione **Avvia temporizzatore come impulso** avvia un temporizzatore specificato con un valore di tempo dato se l'RLC ha un fronte di salita (ovvero, se nell'RLC ha luogo una transizione da 0 a 1). Il temporizzatore continua a funzionare con il tempo specificato finché l'RLC è positivo.

Un'interrogazione dello stato di segnale del numero del temporizzatore per verificare se questo corrisponde a 1, produce un risultato di 1, finché il temporizzatore è funzionante. Se l'RLC cambia da 1 a 0 prima che sia trascorso il tempo, il temporizzatore viene arrestato. In questo caso, l'interrogazione dello stato di segnale per verificare se questo è 1, produce un risultato di 0.

Il box **Avvia temporizzatore come impulso** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

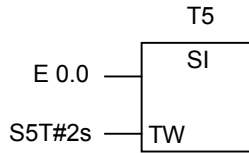
Vedere anche Aree di memoria e componenti di un temporizzatore.

#### Parola di stato

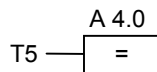
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	0	-	-	0

### Esempio

Segmento 1



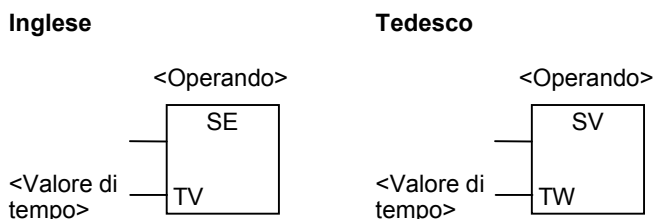
Segmento 2



Se lo stato di segnale di E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Quando lo stato di segnale è = 1, il temporizzatore funziona con il valore indicato di 2 secondi. Se lo stato di segnale di E 0.0 cambia da 1 a 0 prima che sia trascorso tale tempo, il temporizzatore viene arrestato.

### 13.9 SV: Avvia temporizzatore come impulso prolungato

#### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
Numero del temporizzatore	Numero del temporizzatore	TIMER	T	L'operando indica il temporizzatore da avviare.
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo (formato S5TIME)

#### Descrizione

L'operazione **Avvia temporizzatore come impulso prolungato** avvia un temporizzatore con un valore di tempo dato, se l'RLC ha un fronte di salita (ovvero, se nell'RLC ha luogo una transizione da 0 a 1). Il temporizzatore continua a funzionare con il tempo specificato anche se l'RLC diventa 0 prima che sia trascorso il tempo. Un'interrogazione dello stato di segnale del numero del temporizzatore per verificare se questo corrisponde a 1, produce un risultato di 1 finché il temporizzatore è funzionante. Il temporizzatore viene riavviato con il tempo specificato se l'RLC cambia da 0 a 1, mentre il temporizzatore sta funzionando.

Il box **Avvia temporizzatore come impulso prolungato** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

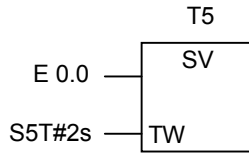
Vedere anche Aree di memoria e componenti di un temporizzatore.

#### Parola di stato

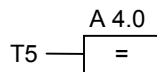
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	0	-	-	0

**Esempio**

Segmento 1



Segmento 2

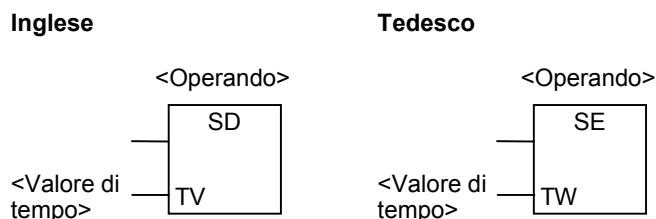


Se lo stato di segnale di E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua a funzionare senza essere influenzato da un fronte di discesa nell'RLC. Se lo stato di segnale dell'ingresso E 0.0 passa da 0 a 1 prima che sia trascorso il tempo specificato, il temporizzatore viene riavviato.

Quando il temporizzatore è in funzione, lo stato di segnale dell'uscita A 4.0 = 1.

### 13.10 SE: Avvia temporizzatore come ritardo all'inserzione

#### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
Numero del temporizzatore	Numero del temporizzatore	TIMER	T	L'operando indica il temporizzatore da avviare.
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo (formato S5TIME)

#### Descrizione

L'operazione **Avvia temporizzatore come ritardo all'inserzione** avvia un temporizzatore specificato se l'RLC ha un fronte di salita (ovvero, se nell'RLC ha luogo una transizione da 0 a 1).

Un'interrogazione dello stato di segnale del numero del temporizzatore per verificare se questo corrisponde a 1, produce un risultato di 1, quando il tempo specificato è trascorso senza errori, e l'RLC è ancora 1. Quando l'RLC cambia da 1 a 0 mentre il temporizzatore sta funzionando, il temporizzatore viene arrestato. In questo caso, l'interrogazione dello stato di segnale per verificare se questo corrisponde a 1 produce sempre il risultato 0.

Il box **Avvia temporizzatore come ritardo all'inserzione** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

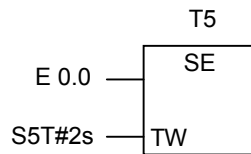
Vedere anche Aree di memoria e componenti di un temporizzatore.

#### Parola di stato

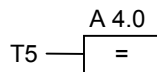
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	0	-	-	0

**Esempio**

Segmento 1



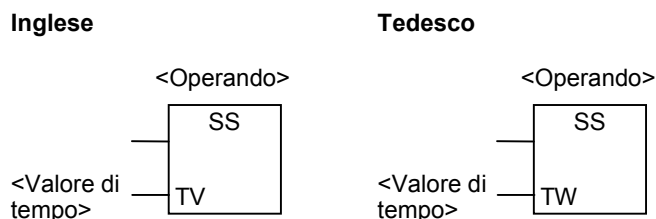
Segmento 2



Se lo stato di segnale di E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Se il temporizzatore è in funzione e lo stato di segnale continua ad essere 1, l'uscita A 4.0 = 1. Se lo stato di segnale passa da 1 a 0 il temporizzatore viene arrestato.

### 13.11 SS: Avvia temporizzatore come ritardo all'inserzione con memoria

#### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
Numero del temporizzatore	Numero del temporizzatore	TIMER	T	L'operando indica il temporizzatore da avviare.
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo (formato S5TIME)

#### Descrizione

L'operazione **Avvia temporizzatore come ritardo all'inserzione con memoria** avvia un temporizzatore specifico se l'RLC ha un fronte di salita (ovvero, nell'RLC ha luogo una transizione da 0 a 1). Il temporizzatore continua a funzionare con il tempo specificato, anche se l'RLC diventa 0 prima che il tempo sia trascorso. Un'interrogazione dello stato di segnale del numero del temporizzatore per verificare se questo corrisponde a 1, produce, quando il tempo è trascorso, un risultato di 1, indipendentemente dall'RLC. Il temporizzatore viene riavviato con il tempo specificato se l'RLC cambia da 0 a 1 mentre il temporizzatore sta funzionando.

Il box **Avvia temporizzatore come ritardo all'inserzione con memoria** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

Vedere anche Aree di memoria e componenti di un temporizzatore.

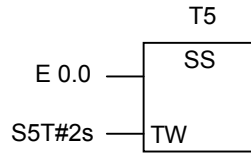
#### Parola di stato

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

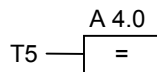


**Esempio**

Segmento 1



Segmento 2



Se lo stato di segnale di E 0.0 passa da 0 a 1 (ovvero se vi è un fronte di salita nell'RLC), il temporizzatore T5 viene avviato. Se il tempo trascorre e lo stato di segnale continua ad essere 1, l'uscita A 4.0 =1.

Se lo stato di segnale dell'ingresso E 0.0 passa da 1 a 0 prima che sia trascorso il tempo specificato, il temporizzatore viene arrestato.

### 13.12 SA: Avvia temporizzatore come ritardo alla disinserzione

#### Simbolo



Parametro Inglese	Parametro Tedesco	Tipo di dati	Area di memoria	Descrizione
Numero del temporizzatore	Numero del temporizzatore	TIMER	T	L'operando indica il temporizzatore da avviare.
TV	TW	S5TIME	E, A, M, D, L o costante	Valore di tempo (formato S5TIME)

#### Descrizione

L'operazione **Avvia temporizzatore come ritardo alla disinserzione** avvia un temporizzatore specificato se l'RLC ha un fronte di discesa (ovvero, nell'RLC ha luogo una transizione da 1 a 0). Un'interrogazione dello stato di segnale del numero del temporizzatore per verificare se questo è 1 produce un risultato di 1, quando l'RLC è 1, oppure quando il temporizzatore è in funzionamento. Il temporizzatore non viene riavviato, fino a quando l'RLC non cambia da 0 a 1.

Il box **Avvia temporizzatore come ritardo alla disinserzione** può essere collocato solo all'estremità destra della stringa logica. È tuttavia possibile utilizzare più box.

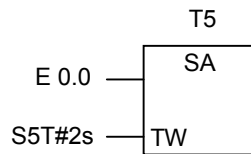
Vedere anche Aree di memoria e componenti di un temporizzatore.

#### Parola di stato

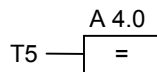
	BIE	A1	A0	OV	OS	OR	STA	RLC	/ER
scrive	-	-	-	-	-	0	-	-	0

**Esempio**

Segmento 1



Segmento 2



Quando lo stato di segnale di E 0.0 passa da 0 a 1 il temporizzatore T5 viene avviato. Il temporizzatore continua a funzionare indipendentemente dal passaggio dello stato di segnale da 1 a 0.

Se lo stato di segnale passa da 0 a 1 prima che il tempo sia trascorso, il temporizzatore viene riavviato.

Quando il tempo è trascorso l'uscita A 4.0 ha lo stato di segnale 1.



## 14 Operazioni logiche combinatorie 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. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita OUT. ENO ha il medesimo stato di segnale di EN.

La relazione rispetto allo 0 del risultato all'uscita OUT influisce sul bit A1 della parola di stato, nel seguente modo:

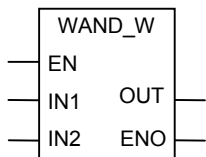
- Se il risultato dell'uscita OUT è diverso da 0, il bit A1 della parola di stato viene impostato a 1.
- Se il risultato dell'uscita OUT è uguale a 0, il bit A1 della parola di stato viene impostato a 0.

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 OR 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, D, L, T, Z	Ingresso di abilitazione
IN1	WORD	E, A, M, D, L o costante	Primo valore per l'operazione logica
IN2	WORD	E, A, M, D, L o costante	Secondo valore per l'operazione logica
OUT	WORD	E, A, M, D, L	Risultato dell'operazione logica
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

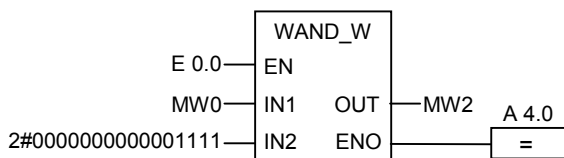
### Descrizione

Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Combinazione AND** parola che combina i due valori digitali indicati negli ingressi IN1 e IN2, bit per bit, secondo la tabella della verità AND. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita OUT. ENO ha il medesimo 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



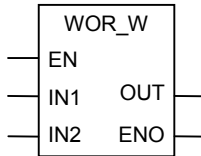
L'operazione viene attivata se E 0.0 = 1. Sono rilevanti solo i bit da 0 a 3, gli altri bit di MW0 sono mascherati.

IN1 = 0101010101010101  
 IN2 = 0000000000001111  
 OUT = 000000000000101

Se l'operazione logica viene eseguita, A 4.0 = 1

## 14.3 WOR\_W: Combinazione OR parola

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	WORD	E, A, M, D, L o costante	Primo valore per l'operazione logica
IN2	WORD	E, A, M, D, L o costante	Secondo valore per l'operazione logica
OUT	WORD	E, A, M, D, L	Risultato dell'operazione logica
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

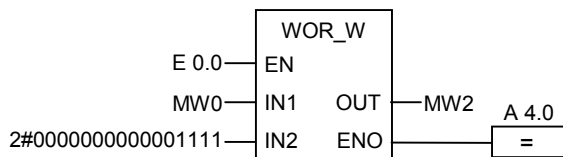
### Descrizione

Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Combinazione OR parola** che combina i due valori digitali indicati negli ingressi IN1 e IN2, bit per bit, secondo la tabella della verità OR. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita OUT. ENO ha il medesimo 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 attivata se E 0.0 = 1. I bit di MW0 e della costante vengono combinati con OR. I bit da 0 a 3 vengono impostati su 1, mentre gli altri bit di MW0 vengono inseriti in MW2 senza alcuna modifica.

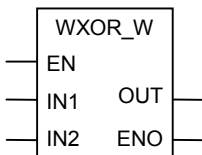
```

IN1      =    0101010101010101
IN2      =    0000000000001111
OUT      =    0101010101111111
    
```

Se l'operazione logica viene eseguita, A 4.0 = 1.

## 14.4 WXOR\_W: Combinazione OR esclusivo parola

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	WORD	E, A, M, D, L o costante	Primo valore per l'operazione logica
IN2	WORD	E, A, M, D, L o costante	Secondo valore per l'operazione logica
OUT	WORD	E, A, M, D, L	Risultato dell'operazione logica
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

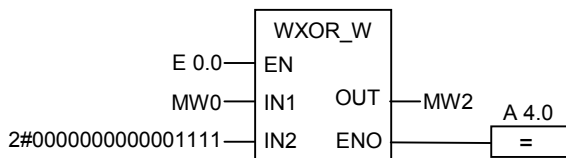
### Descrizione

Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Combinazione OR esclusivo parola** che combina i due valori digitali indicati negli ingressi IN1 e IN2, bit per bit, secondo la tabella della verità OR esclusivo. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita OUT. ENO ha il medesimo 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 attivata se E 0.0 = 1.

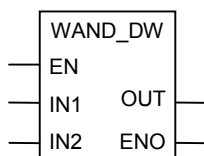
IN1 = 01010101010101  
 IN2 = 0000000000001111  
 OUT = 01010101011010

Se l'operazione logica viene eseguita, A 4.0 = 1.



## 14.5 WAND\_DW: Combinazione AND doppia parola

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	DWORD	E, A, M, D, L o costante	Primo valore per l'operazione logica
IN2	DWORD	E, A, M, D, L o costante	Secondo valore per l'operazione logica
OUT	DWORD	E, A, M, D, L	Risultato dell'operazione logica
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

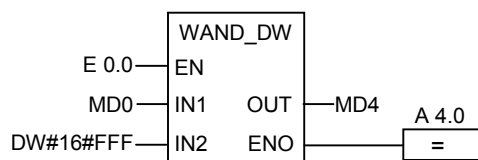
### Descrizione

Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Combinazione AND doppia parola** che combina i due valori digitali indicati negli ingressi IN1 e IN2, bit per bit, secondo la tabella della verità AND. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita OUT. ENO ha il medesimo 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 attivata se E 0.0 = 1. Sono rilevanti solo i bit da 0 a 11, gli altri bit di MD4 sono mascherati.

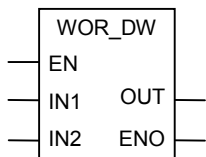
```

IN1 = 0101010101010101 0101010101010101
IN2 = 0000000000000000 0000111111111111
OUT = 0000000000000000 0000010101010101
    
```

Se l'operazione logica viene eseguita, A 4.0 = 1.

## 14.6 WOR\_DW: Combinazione OR doppia parola

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	DWORD	E, A, M, D, L o costante	Primo valore per l'operazione logica
IN2	DWORD	E, A, M, D, L o costante	Secondo valore per l'operazione logica
OUT	DWORD	E, A, M, D, L	Risultato dell'operazione logica
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

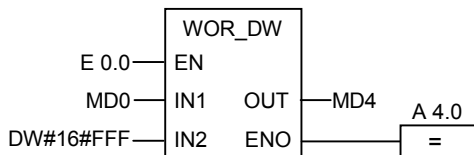
### Descrizione

Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Combinazione OR doppia parola** che combina i due valori digitali indicati negli ingressi IN1 e IN2, bit per bit, secondo la tabella della verità OR. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita OUT. ENO ha il medesimo 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 attivata se E 0.0 = 1. I bit di MD0 e della costante vengono combinati con OR. I bit da 0 a 11 vengono impostati su 1, mentre gli altri bit di MD0 vengono inseriti in MD4 senza alcuna modifica.

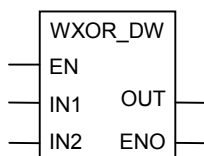
```

IN1 = 0101010101010101 0101010101010101
IN2 = 0000000000000000 0000111111111111
OUT = 0101010101010101 0101111111111111
    
```

Se l'operazione logica viene eseguita, A 4.0 = 1.

## 14.7 WXOR\_DW: Combinazione OR esclusivo doppia parola

### Simbolo



Parametro	Tipo di dati	Area di memoria	Descrizione
EN	BOOL	E, A, M, D, L, T, Z	Ingresso di abilitazione
IN1	DWORD	E, A, M, D, L o costante	Primo valore per l'operazione logica
IN2	DWORD	E, A, M, D, L o costante	Secondo valore per l'operazione logica
OUT	DWORD	E, A, M, D, L	Risultato dell'operazione logica
ENO	BOOL	E, A, M, D, L	Uscita di abilitazione

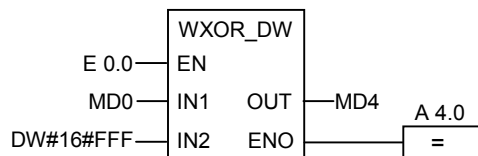
### Descrizione

Uno stato di segnale uguale a 1 all'ingresso di abilitazione (EN) attiva l'operazione **Combinazione OR esclusivo doppia parola** che combina i due valori digitali indicati negli ingressi IN1 e IN2, bit per bit, secondo la tabella della verità OR esclusivo. I valori vengono interpretati come puri pattern di bit. Il risultato può essere letto all'uscita OUT. ENO ha il medesimo 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 attivata se E 0.0 = 1.

```

IN1 = 0101010101010101 0101010101010101
IN2 = 0000000000000000 0000111111111111
OUT = 0101010101010101 0101101010101010
    
```

Se l'operazione logica viene eseguita, A 4.0 = 1.



## A Sommario di tutte le operazioni FUP

### A.1 Operazioni FUP ordinate secondo il set mnemonico tedesco (SIMATIC)

Mnemonico tedesco	Mnemonico inglese	Catalogo elementi del programma	Descrizione
&	&	Operazione logica combinatoria a bit	AND
>=1	>=1	Operazione logica combinatoria a bit	OR
=	=	Operazione logica combinatoria a bit	Assegnazione
#	#	Operazione logica combinatoria a bit	Connettore
---	---	Operazione logica combinatoria a bit	Ingresso binario
---o	---o	Operazione logica combinatoria a bit	Nega ingresso binario
==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 di 0
<=0	<=0	Bit di stato	Bit di risultato minore o uguale a 0
<>0	<>0	Bit di stato	Bit di risultato diverso da 0
ABS	ABS	Funzione in virgola mobile	Forma valore assoluto di un numero in virgola mobile
ACOS	ACOS	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
ADD_DI	ADD_DI	Funzione in virgola fissa	Somma numeri interi (32 bit)
ADD_I	ADD_I	Funzione in virgola fissa	Somma numeri interi (16 bit)
ADD_R	ADD_R	Funzione in virgola mobile	Somma numeri in virgola mobile
ASIN	ASIN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
ATAN	ATAN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
BCD_DI	BCD_DI	Convertitori	Converti un BCD in un numero intero (32 bit)
BCD_I	BCD_I	Convertitori	Converti un BCD in un numero intero (16 bit)
BIE	BR	Bit di stato	Registro BIE: bit di anomalia
CALL	CALL	Comando del programma	Richiama FC/SFC senza parametri
CALL_FB	CALL_FB	Comando del programma	Richiama FB dal box

Mnemonico tedesco	Mnemonico inglese	Catalogo elementi del programma	Descrizione
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 di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
DI_BCD	DI_BCD	Convertitori	Converti un numero intero (32 bit) in un numero BCD
DI_R	DI_R	Convertitori	Converti un numero intero (32 bit) in un numero in virgola mobile
DIV_DI	DIV_DI	Funzione in virgola fissa	Dividi numeri interi (32 bit)
DIV_I	DIV_I	Funzione in virgola fissa	Dividi numeri interi (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
FLOOR	FLOOR	Convertitori	Genera da un numero in virgola mobile un numero intero
I_BCD	I_BCD	Convertitori	Converti numero intero (16 bit) in numero BCD
I_DI	I_DI	Convertitori	Converti numero intero (16 bit) in numero intero (32 bit)
INV_I	INV_I	Convertitori	Complemento a 1 di numero intero (16 bit)
INV_DI	INV_DI	Convertitori	Complemento a 1 di numero intero (32 bit)
JMP	JMP	Salti	Salto assoluto nel blocco
JMP	JMP	Salti	Salta nel blocco se 1
JMPN	JMPN	Salti	Salta nel blocco se 0
LABEL	LABEL	Salti	Etichetta di salto
LN	LN	Funzione in virgola mobile	Formazione del logaritmo naturale di un numero in virgola mobile
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 (32 bit)
MOVE	MOVE	Spostamento	Assegna un valore
MUL_DI	MUL_DI	Funzione in virgola fissa	Moltiplica numeri interi (32 bit)
MUL_I	MUL_I	Funzione in virgola fissa	Moltiplica numeri interi (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 2 di numero intero (32 bit)

## A.1 Operazioni FUP ordinate secondo il set mnemonico tedesco (SIMATIC)

Mnemonico tedesco	Mnemonico inglese	Catalogo elementi del programma	Descrizione
NEG_I	NEG_I	Convertitori	Complemento a 2 di numero intero (16 bit)
NEG_R	NEG_R	Convertitori	Nega numero in virgola mobile
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	Salto indietro
ROL_DW	ROL_DW	Scorrimento/rotazione	Fai ruotare doppia parola a sinistra
ROUND	ROUND	Convertitori	Arrotonda a numero intero (32 bit)
ROR_DW	ROR_DW	Scorrimento/rotazione	Fai ruotare doppia parola a destra
RS	RS	Operazione logica combinatoria a bit	Imposta resetta flip flop
S	S	Operazione logica combinatoria a bit	Imposta bobina
SA	SF	Temporizzatori	Avvia temporizzatore come ritardo alla disinserzione
SAVE	SAVE	Operazione logica combinatoria a bit	Salva RLC nel registro BIE
S_AVERZ	S_OFFDT	Temporizzatori	Avvia e parametrizza temporizzatore come ritardo alla disinserzione
SE	SD	Temporizzatori	Avvia temporizzatore come ritardo all'inserzione
S_EVERZ	S_ODT	Temporizzatori	Avvia e parametrizza temporizzatore come ritardo all'inserzione
SHL_DW	SHL_DW	Scorrimento/rotazione	Fai scorrere doppia parola a sinistra
SHL_W	SHL_W	Scorrimento/rotazione	Fai scorrere parola a sinistra
SHR_DI	SHR_DI	Scorrimento/rotazione	Fai scorrere numero intero a 32 bit a destra
SHR_DW	SHR_DW	Scorrimento/rotazione	Fai scorrere doppia parola a destra
SHR_I	SHR_I	Scorrimento/rotazione	Fai scorrere numero intero a 16 bit a destra
SHR_W	SHR_W	Scorrimento/rotazione	Fai scorrere parola a destra
SI	SP	Temporizzatori	Avvia temporizzatore come impulso
S_IMPULS	S_PULSE	Temporizzatori	Avvia e parametrizza temporizzatore come impulso
SIN	SIN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
SQR	SQR	Funzione in virgola mobile	Forma quadrato di un numero in virgola mobile
SQRT	SQRT	Funzione in virgola mobile	Forma radice quadrata di un numero in virgola mobile
SR	SR	Operazione logica combinatoria a bit	Resetta imposta flip flop
SS	SS	Temporizzatori	Avvia temporizzatore come ritardo all'inserzione con memoria

Sommario di tutte le operazioni FUP

A.1 Operazioni FUP ordinate secondo il set mnemonico tedesco (SIMATIC)

Mnemonico tedesco	Mnemonico inglese	Catalogo elementi del programma	Descrizione
S_SEVERZ	S_ODTS	Temporizzatori	Avvia e parametrizza temporizzatore come ritardo all'inserzione con memoria
SUB_DI	SUB_DI	Funzione in virgola fissa	Sottrai numeri interi (32 bit)
SUB_I	SUB_I	Funzione in virgola fissa	Sottrai numeri interi (16 bit)
SUB_R	SUB_R	Funzione in virgola mobile	Sottrai numeri in virgola mobile
SV	SE	Temporizzatori	Avvia temporizzatore come impulso prolungato
S_VIMP	S_PEXT	Temporizzatori	Avvia e parametrizza temporizzatore come impulso prolungato
SZ	SC	Contatori	Imposta valore di conteggio
TAN	TAN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
TRUNC	TRUNC	Convertitori	Arrotonda senza resto di un numero intero (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
XOR	XOR	Operazione logica combinatoria a bit	OR esclusivo
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 FUP ordinate secondo il set mnemonico inglese (internazionale)

Menmonico inglese	Mnemonico tedesco	Catalogo elementi del programma	Descrizione
&	&	Operazione logica combinatoria a bit	AND
>=1	>=1	Operazione logica combinatoria a bit	OR
=	=	Operazione logica combinatoria a bit	Assegnazione
#	#	Operazione logica combinatoria a bit	Connettore
---	---	Operazione logica combinatoria a bit	Ingresso binario
---o	---o	Operazione logica combinatoria a bit	Nega ingresso binario
==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 di 0
<=0	<=0	Bit di stato	Bit di risultato minore o uguale a 0
<>0	<>0	Bit di stato	Bit di risultato diverso da 0
ABS	ABS	Funzione in virgola mobile	Forma valore assoluto di un numero in virgola mobile
ACOS	ACOS	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
ADD_DI	ADD_DI	Funzione in virgola fissa	Somma numeri interi (32 bit)
ADD_I	ADD_I	Funzione in virgola fissa	Somma numeri interi (16 bit)
ADD_R	ADD_R	Funzione in virgola mobile	Somma numeri in virgola mobile
ASIN	ASIN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
ATAN	ATAN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
BCD_DI	BCD_DI	Convertitori	Converti un BCD in un numero intero (32 bit)
BCD_I	BCD_I	Convertitori	Converti un BCD in un numero intero (16 bit)
BR	BIE	Bit di stato	Registro BIE: bit di anomalia
CALL	CALL	Comando del programma	Richiama FC/SFC 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

Menmonico 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 di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
CU	ZV	Contatori	Conta in avanti
DI_BCD	DI_BCD	Convertitori	Converti un numero intero (32 bit) in un BCD
DI_R	DI_R	Convertitori	Converti un numero intero (32 bit) in un numero in virgola mobile
DIV_DI	DIV_DI	Funzione in virgola fissa	Dividi numeri interi (32 bit)
DIV_I	DIV_I	Funzione in virgola fissa	Dividi numeri interi (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
FLOOR	FLOOR	Convertitori	Genera da un numero in virgola mobile un numero intero
I_BCD	I_BCD	Convertitori	Converti numero intero (16 bit) in numero BCD
I_DI	I_DI	Convertitori	Converti numero intero (16 bit) in numero intero (32 bit)
INV_I	INV_I	Convertitori	Complemento a 1 di numero intero (16 bit)
INV_DI	INV_DI	Convertitori	Complemento a 1 di numero intero (32 bit)
JMP	JMP	Salti	Salto assoluto nel blocco_
JMP	JMP	Salti	Salta nel blocco se 1
JMPN	JMPN	Salti	Salta nel blocco se 0
LABEL	LABEL	Salti	Etichetta di salto
LN	LN	Funzione in virgola mobile	Formazione del logaritmo naturale di un numero in virgola mobile
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 (32 bit)
MOVE	MOVE	Spostamento	Assegna un valore
MUL_DI	MUL_DI	Funzione in virgola fissa	Moltiplica numeri interi (32 bit)
MUL_I	MUL_I	Funzione in virgola fissa	Moltiplica numeri interi (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 2 di numero intero (32 bit)
NEG_I	NEG_I	Convertitori	Complemento a 2 di numero intero (16 bit)
NEG_R	NEG_R	Convertitori	Nega numero in virgola mobile
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

## A.2 Operazioni FUP ordinate secondo il set mnemonico inglese (internazionale)

Menmonico inglese	Mnemonico tedesco	Catalogo elementi del programma	Descrizione
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	Salto indietro
ROL_DW	ROL_DW	Scorrimento/rotazione	Fai ruotare doppia parola a sinistra
ROUND	ROUND	Convertitori	Arrotonda a numero intero (32 bit)
ROR_DW	ROR_DW	Scorrimento/rotazione	Fai ruotare doppia parola a destra
RS	RS	Operazione logica combinatoria a bit	Imposta resetta 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 temporizzatore come ritardo all'inserzione
SE	SV	Temporizzatori	Avvia temporizzatore come impulso prolungato
SF	SA	Temporizzatori	Avvia temporizzatore come ritardo alla disinserzione
SHL_DW	SHL_DW	Scorrimento/rotazione	Fai scorrere doppia parola a sinistra
SHL_W	SHL_W	Scorrimento/rotazione	Fai scorrere parola a sinistra
SHR_DI	SHR_DI	Scorrimento/rotazione	Fai scorrere numero intero a 32 bit a destra
SHR_DW	SHR_DW	Scorrimento/rotazione	Fai scorrere doppia parola a destra
SHR_I	SHR_I	Scorrimento/rotazione	Fai scorrere numero intero a 16 bit a destra
SHR_W	SHR_W	Scorrimento/rotazione	Fai scorrere parola a destra
SIN	SIN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
S_ODT	S_EVERZ	Temporizzatori	Avvia e parametrizza temporizzatore come ritardo all'inserzione
S_ODTS	S_SEVERZ	Temporizzatori	Avvia e parametrizza temporizzatore come ritardo all'inserzione con memoria
S_OFFDT	S_AVERZ	Temporizzatori	Avvia e parametrizza temporizzatore come ritardo alla disinserzione
SP	SI	Temporizzatori	Avvia temporizzatore come impulso
S_PEXT	S_VIMP	Temporizzatori	Avvia e parametrizza temporizzatore come impulso prolungato
S_PULSE	S_IMPULS	Temporizzatori	Avvia e parametrizza temporizzatore come impulso
SQR	SQR	Funzione in virgola mobile	Forma quadrato di un numero in virgola mobile
SQRT	SQRT	Funzione in virgola mobile	Forma radice quadrata di un numero in virgola mobile

Sommario di tutte le operazioni FUP

A.2 Operazioni FUP ordinate secondo il set mnemonico inglese (internazionale)

Menmonico inglese	Mnemonico tedesco	Catalogo elementi del programma	Descrizione
SR	SR	Operazione logica combinatoria a bit	Resetta imposta flip flop
SS	SS	Temporizzatori	Avvia temporizzatore come ritardo all'inserzione con memoria
SUB_DI	SUB_DI	Funzione in virgola fissa	Sottrai numeri interi (32 bit)
SUB_I	SUB_I	Funzione in virgola fissa	Sottrai numeri interi (16 bit)
SUB_R	SUB_R	Funzione in virgola mobile	Sottrai numeri in virgola mobile
TAN	TAN	Funzione in virgola mobile	Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile
TRUNC	TRUNC	Convertitori	Arrotonda senza resto di numero intero (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
XOR	XOR	Operazione logica combinatoria a bit	OR esclusivo

## B Esempi di programmazione

### B.1 Sommario

#### Applicazione pratiche

Tutte le operazioni FUP 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 operazioni FUP:

- 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

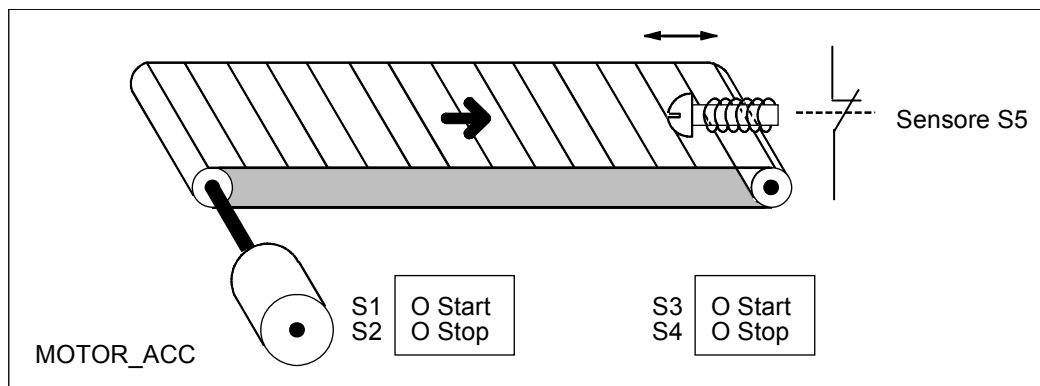
#### Operazioni utilizzate

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	Rilevamento di 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	Combinazione
>=1	Operazione logica combinatoria a bit	Combinazione logica AND
=	Operazione logica combinatoria a bit	Assegnazione logica OR
JMPN	Salti	Salta nel blocco se 0 (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

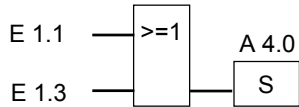
È possibile scrivere un programma destinato al controllo di un nastro trasportatore, avvalendosi di **valori assoluti** oppure di **simboli** che rappresentano i diversi componenti del sistema di trasporto.

I simboli scelti vengono correlati nella tabella dei simboli con i valori assoluti (consultare la Guida online di STEP 7).

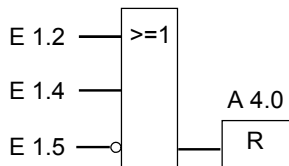
Componente del sistema	Indirizzo assoluto	Simbolo	Tabella dei simboli
Pulsante Start	E 1.1	S1	E 1.1 S1
Pulsante Stop	E 1.2	S2	E 1.2 S2
Pulsante Start	E 1.3	S3	E 1.3 S3
Pulsante Stop	E 1.4	S4	E 1.4 S4
Sensore	E 1.5	S5	E 1.5 S5
Motore	A 4.0	MOTORE_ACC	A 4.0 MOTORE_ACC

### Schema logico per il controllo del nastro trasportatore

Segmento 1: premendo uno dei due pulsanti Start si aziona il motore.

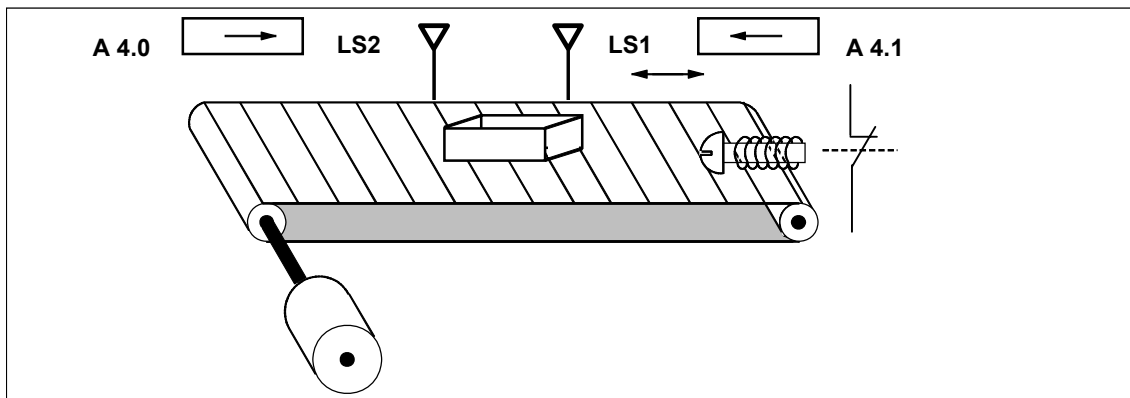


Segmento 2: premendo uno dei due pulsanti Stop o aprendo il contatto normalmente chiuso posto alla fine del nastro trasportatore si spegne il motore.



### Esempio 2: Rilevamento della direzione di marcia di un nastro trasportatore

La seguente 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**

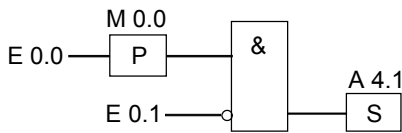
È possibile scrivere un programma destinato al controllo di un nastro trasportatore, avvalendosi di **valori assoluti** oppure di **simboli** che rappresentano i diversi componenti del sistema di trasporto.

I simboli scelti vengono correlati nella tabella dei simboli con i valori assoluti (consultare la Guida online di STEP 7).

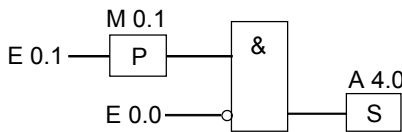
Componente del sistema	Indirizzo assoluto	Simbolo	Tabella dei simboli
Barriera a fotocellula elettrica 1	E 0.0	LS1	E 0.0 LS1
Barriera a fotocellula elettrica 2	E 0.1	LS2	E 0.1 LS2
Visualizzatore per il senso di marcia a destra	A 4.0	DESTRA	A 4.0 DESTRA
Visualizzatore per il senso di marcia a sinistra	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 logico per il rilevamento della direzione di marcia di un nastro trasportatore**

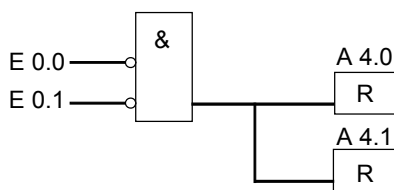
Segmento 1: se lo stato del segnale di E 0.0 passa da 0 a 1 (fronte di salita) e allo stesso tempo, lo stato di segnale di E 0.1 è 0, il pacco sul nastro si sposta verso sinistra.



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

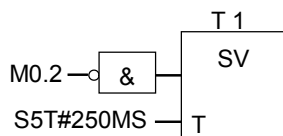
### Generatore d'impulso di clock

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.

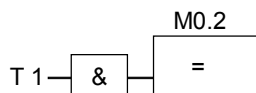
Quando si utilizza S7-300, si può implementare la funzione di generazione d'impulsi avvalendosi di un'elaborazione temporizzata in blocchi di organizzazione speciali.

### Schema logico per l'attivazione delle spie su un pannello visualizzatore (fattore di impulso 1:1)

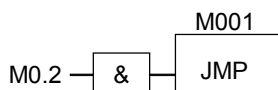
Segmento 1: se lo stato del segnale del temporizzatore T1 è 0, caricare il valore 250 ms in T1 e avviare T1 come temporizzatore di impulso prolungato.



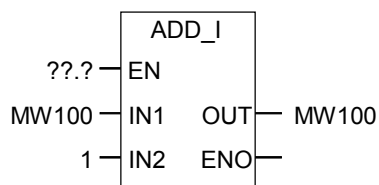
Segmento 2: lo stato del temporizzatore viene depositato in un merker di appoggio.



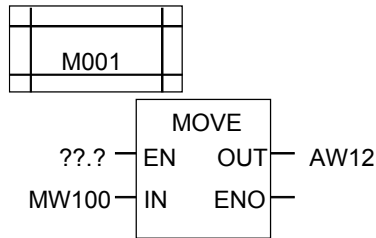
Segmento 3: se lo stato del segnale del temporizzatore T1 è 1, saltare all'etichetta M001.



Segmento 4: ogni volta che il tempo del temporizzatore T1 è trascorso, la parola di merker 100 viene incrementata di 1.

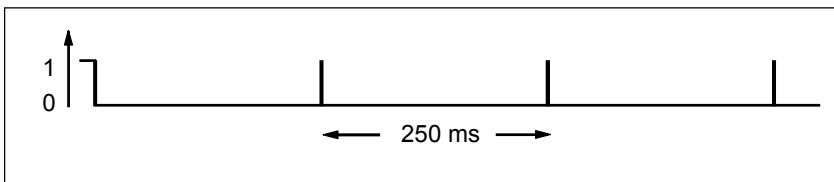


Segmento 5: l'operazione **MOVE** consente di vedere le diverse frequenze di clock sulle uscite da A 12.0 a A 13.7.



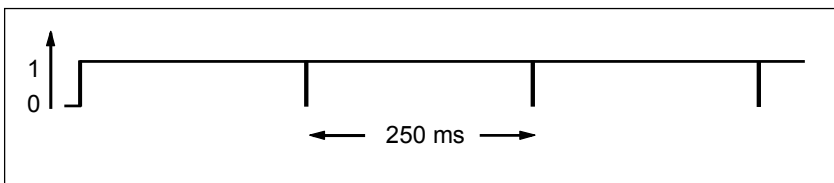
### Interrogazione di segnale

L'interrogazione di segnale del temporizzatore T1 produce per il parametro di ingresso negato della combinazione AND (M0.2) il seguente risultato combinatorio:



Appena il tempo è scaduto, il temporizzatore viene riavviato. Per questo motivo, il controllo di segnale svolto dal parametro di ingresso negato della combinazione AND (M0.2) produce uno stato del segnale uguale a 1 solo per un istante.

Il bit di negazione RLC (invertito):



Ogni 250 ms il bit RLC è 0. Il salto viene ignorato e il contenuto della parola di merker MW100 viene incrementato di 1.

### Ottenimento di una frequenza specifica

Con i bit dei merker MB101 e MB100 è possibile ottenere le seguenti frequenze:

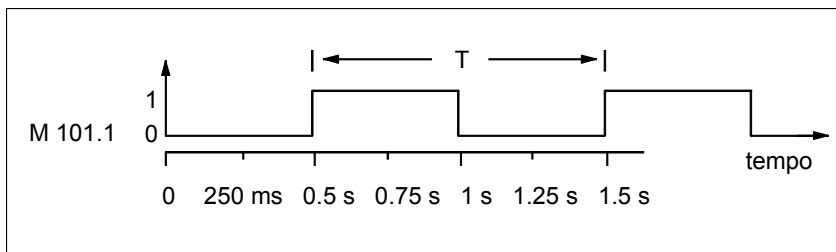
Bit di MB101, MB100	Frequenza in Hertz	Durata
M 101.0	2.0	0.5 s (250 ms on / 250 ms off )
M 101.1	1.0	1 s (0.5 s on / 0.5 s off )
M 101.2	0.5	2 s (1 s on / 1 s off )
M 101.3	0.25	4 s (2 s on / 2 s off )
M 101.4	0.125	8 s (4 s on / 4 s off )
M 101.5	0.0625	16 s (8 s on / 8 s off )
M 101.6	0.03125	32 s (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	<b>0</b>	0	250
1	0	0	0	0	0	0	<b>0</b>	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	<b>0</b>	0	250
5	0	0	0	0	0	1	<b>0</b>	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	<b>0</b>	0	250
9	0	0	0	0	1	0	<b>0</b>	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	<b>0</b>	0	250

**Stato del segnale del bit 1 di MB101 (M 101.1)**

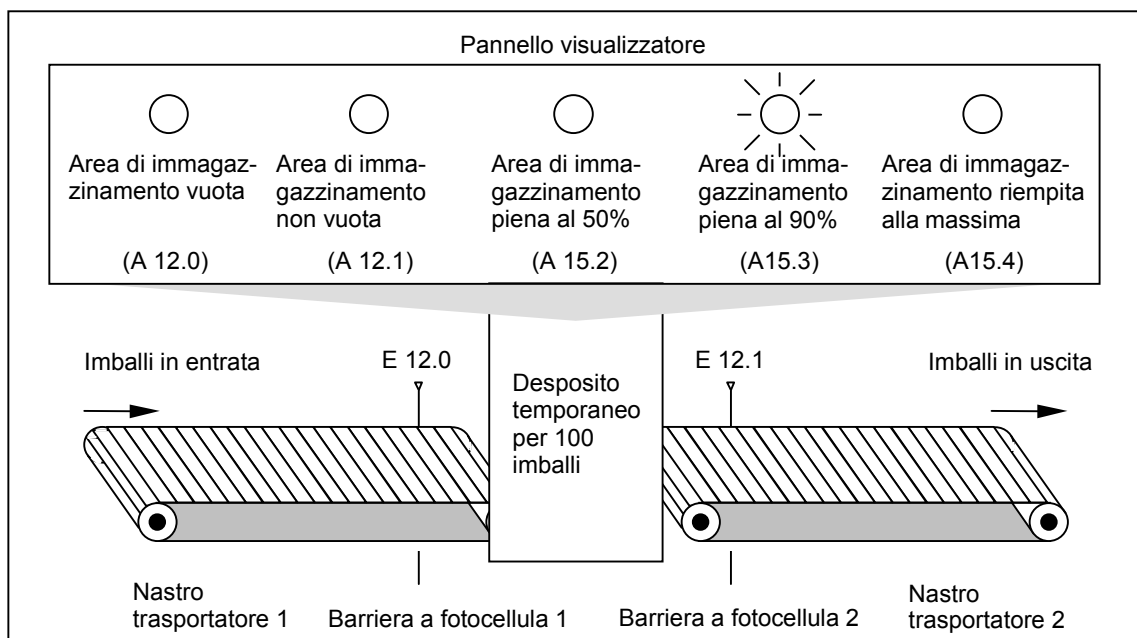
Frequenza =  $1/T = 1/1 \text{ s} = 1 \text{ Hz}$



## B.4 Esempio: Operazioni di conteggio e confronto

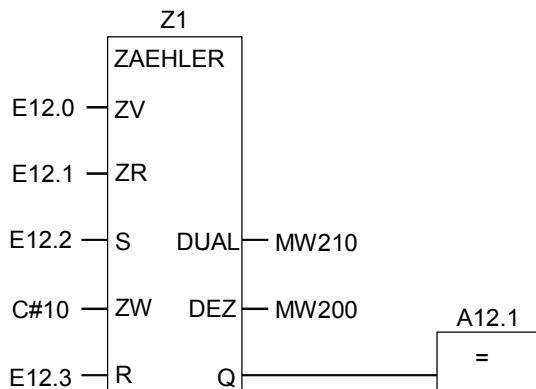
### Area di immagazzinamento con Contatore e Confrontatore

La seguente 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.

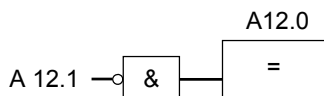


**Schema logico per l'attivazione delle spie su un pannello visualizzatore**

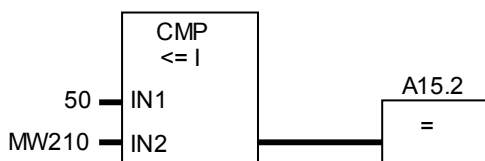
Segmento 1: il contatore Z1 conta in avanti con un fronte di segnale da "0" a "1" all'ingresso ZV e conta all'indietro con un fronte di segnale da "0" a "1" all'ingresso ZR. Con un fronte di segnale da "0" a "1" all'ingresso S, il valore di conteggio viene settato sul valore di ZW. Con un fronte di segnale 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".



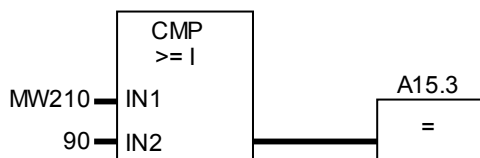
Segmento 2: A12.0 indica "Area di immagazzinamento vuota".



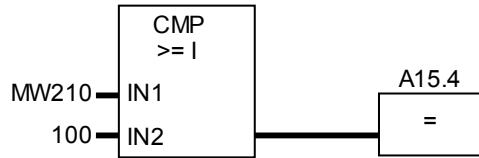
Segmento 3: se 50 è un valore inferiore o pari al valore di conteggio (se cioè lo stato del contatore attuale è maggiore o uguale a 50), si accende la spia di visualizzazione per il messaggio "Area di immagazzinamento piena al 50%".



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:

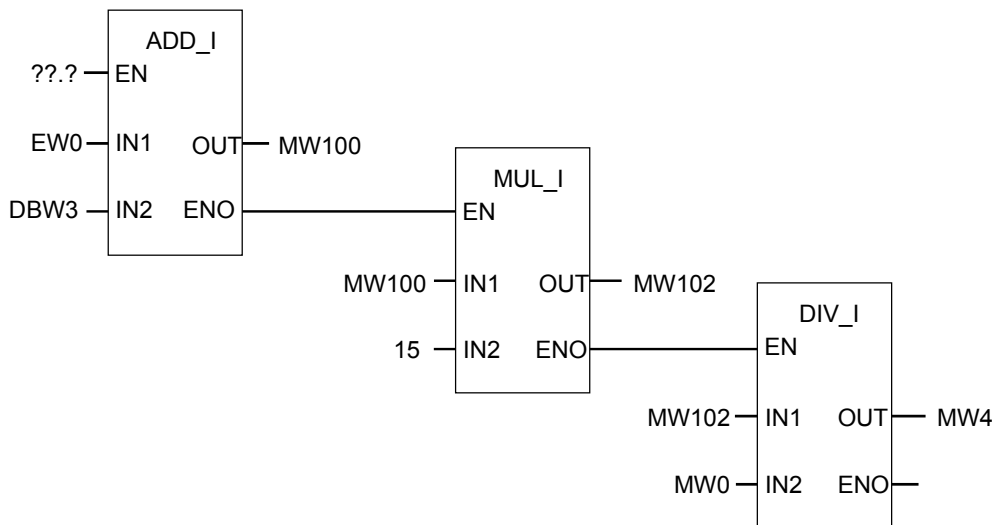
$$MW4 = ((EW0 + DBW3) \times 15) / MW0$$

### Schema logico

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.

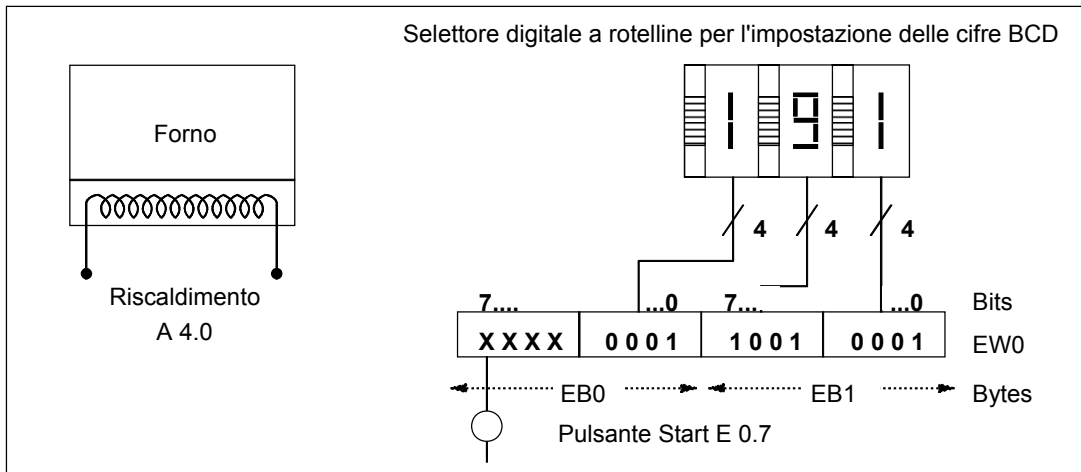




## B.6 Esempio: Operazioni logiche combinatorie a parola

### Riscaldamento di un forno

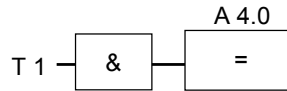
L'operatore di un forno avvia il riscaldamento del forno premendo il pulsante Start. L'operatore può impostare la durata del tempo di riscaldamento avvalendosi di selettori a rotella. Il valore che l'operatore imposta indica i secondi in formato BCD (decimali codificati in binario).



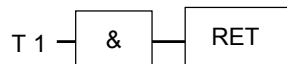
Componente del sistema	Indirizzo assoluto
Pulsante Start	E 0.7
Rotellina delle unità	da E 1.0 a E 1.3
Rotellina delle decine	da E 1.4 a E 1.7
Rotellina delle centinaia	da E 0.0 a E 0.3
Avvio riscaldamento	A 4.0

**Schema logico**

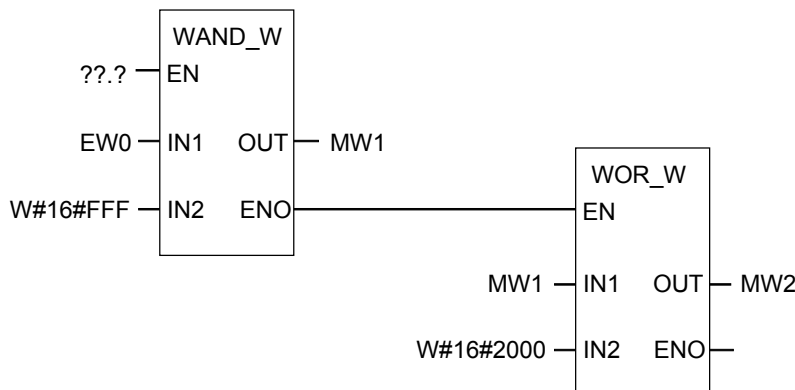
Segmento 1: se il temporizzatore è in funzione, accendere il riscaldatore.



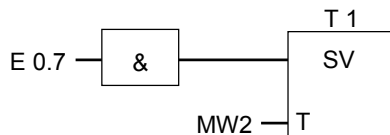
Segmento 2: se il temporizzatore è attivo, l'istruzione **Return** termina qui l'elaborazione.



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 Struttura ed elementi in FUP

### 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         =   A   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 riuscita.

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

## #

# 21

## &

& 13

## <

<=0 144

<>0 144

<0 144

## =

= 20

==0 144

## >

>=0 144

>=1 12

>0 144

## A

ABS 91

ACOS 96

ADD\_DI 79

ADD\_I 75

ADD\_R 87

Addendo con collegamento EN e collegamento ENO  
205

Addendo con collegamento EN e senza collegamento  
ENO 206

Addendo senza collegamento EN e con collegamento  
ENO 206

Addendo senza collegamento EN e senza collegamento  
ENO 207

Apertura di un blocco dati 65

Applicazione pratiche 189

Aree di memoria e componenti di un temporizzatore  
148

Arrotondamento a numero intero 49

Arrotondamento senza resto di un numero intero (32 bit)  
50

ASIN 96

Assegnazione 19

Assegnazione dei parametri 208

Assegnazione di un valore 99

ATAN 96

Attivazione/disattivazione di una zona di relè master  
control 115

Avvertenze importanti sulle funzionalità MCR 114

Avvia temporizzatore come impulso 162

Avvia temporizzatore come impulso prolungato 164

Avvia temporizzatore come ritardo alla disinserzione  
170

Avvia temporizzatore come ritardo all'inserzione 166

Avvia temporizzatore come ritardo all'inserzione con  
memoria 168

Avviamento e parametrizzazione del temporizzatore come  
impulso 152

Avviamento e parametrizzazione del temporizzatore  
come impulso prolungato 154

Avviamento e parametrizzazione del temporizzatore  
come ritardo alla disinserzione 160

Avviamento e parametrizzazione del temporizzatore  
come ritardo all'inserzione 156

Avviamento e parametrizzazione del temporizzatore  
come ritardo all'inserzione con memoria 158

## B

BCD\_DI 40

BCD\_I 38

BIE 143

Bit di anomalia

operazione non ammessa 142

overflow 138

overflow con memoria 140

Bit di risultato 144

BR 143

## C

CALL 103

CALL\_FB Richiama FB dal box 104

CALL\_FC Richiama FC dal box 106

CALL\_SFB Richiama SFB dal box 108

CALL\_SFC Richiama SFC dal box 110

CD 64

CEIL 51

CMP <>I 34

CMP<=D 35

CMP<=I 34

CMP<=R 36

CMP<>D 35

CMP<>R 36

CMP<D 35

CMP<I 34

CMP<R 36

CMP==D 35

CMP==I 34

CMP==R 36  
CMP>=D 35  
CMP>=R 36  
CMP>D 35  
CMP>I 34  
CMP>R 36  
Combinazione AND doppia parola 177  
Combinazione AND parola 174  
Combinazione logica AND 13  
Combinazione logica AND prima di OR o combinazione logica OR prima di AND 14  
Combinazione logica OR esclusivo 16  
Combinazione OR 12  
Combinazione OR doppia parola 178  
Combinazione OR esclusivo doppia parola 179  
Combinazione OR esclusivo parola 176  
Combinazione OR parola 175  
Complemento a 1 di numero intero (16 bit) 44  
Complemento a 1 di numero intero (32 bit) 45  
Complemento a 2 di numero intero (16 bit) 46  
Complemento a 2 di numero intero (32 bit) 47  
Confronto di numeri in virgola mobile 36  
Confronto di numeri interi (16 bit) 34  
Confronto di numeri interi (32 bit) 35  
Connettore 21  
Conta all'indietro 64  
Conta in avanti 63  
Conversione di un numero BCD in un numero intero (16 bit) 38  
Conversione di un numero BCD in un numero intero (32 bit) 40  
Conversione di un numero intero (16 bit) in un numero BCD 39  
Conversione di un numero intero (16 bit) in un numero intero (32 bit) 41  
Conversione di un numero intero (32 bit) in un BCD 42  
Conversione di un numero intero (32 bit) in un numero in virgola mobile 43  
COS 96  
CU 63

## D

DI\_BCD 42  
DI\_R 43  
DIV\_DI 82  
DIV\_I 78  
DIV\_R 90  
Divisione di numeri in virgola mobile 90  
Divisione di numeri interi (16 bit) 78  
Divisione di numeri interi (32 bit) 82

## E

Esempi  
Operazioni logiche combinatorie a bit 190  
Esempi di programmazione 189  
Esempio  
Operazioni di conteggio e confronto 197  
Operazioni di temporizzazione 193  
Operazioni logiche combinatorie a parola 201  
Operazioni matematiche con i numeri interi 200  
Etichetta di salto 71  
EXP 94

## F

Fai ruotare doppia parola a destra (a 32 bit) 135  
Fai ruotare doppia parola a sinistra 133  
Fai scorrere doppia parola a destra 131  
Fai scorrere doppia parola a sinistra 130  
Fai scorrere numero intero a 16 bit a destra 124  
Fai scorrere numero intero a 32 bit a destra 126  
Fai scorrere parola a destra 129  
Fai scorrere parola a sinistra (a 16 bit) 127  
FLOOR 52  
Formazione del logaritmo naturale di un numero in virgola mobile 95  
Formazione del quadrato di un numero in virgola mobile 92  
Formazione del valore assoluto di un numero in virgola mobile 91  
Formazione del valore esponenziale di un numero in virgola mobile 94  
Formazione della radice quadrata di un numero in virgola mobile 93  
Formazione di funzioni trigonometriche di angoli sotto forma di numeri in virgola mobile 96  
Funzioni del relè master control (MCR) 113

## G

Generazione di un numero intero inferiore da un numero in virgola mobile 52  
Generazione di un numero intero superiore da un numero in virgola mobile 51  
Guida online 5

## I

I\_BCD 39  
I\_DI 41  
Impostazione del valore di conteggio 61  
Impostazione della bobina 24  
Impostazione resettaggio flip flop 26  
Inizio/fine della zona relè master control 118  
Inserimento di un ingresso binario 17  
Interrogazione del rilevamento del fronte RLC di salita 32  
Interrogazione del rilevamento di fronte di discesa 31

INV\_DI 45  
INV\_I 44

**J**

JMP 68, 69  
JMPN 70

**L**

LABEL 71  
LN 95

**M**

MCR< 116, 117  
MCR> 116, 117  
MCRA 118, 120  
MCRD 118, 119  
Meccanismo EN/ENO 203, 204  
Memorizzazione dell'RLC nel registro BIE 30  
Mnemonico  
  inglese (internazionale) 185  
  tedesco (SIMATIC) 181  
MOD\_DI 83  
Moltiplicazione di numeri in virgola mobile 89  
Moltiplicazione di numeri interi (16 bit) 77  
Moltiplicazione di numeri interi (32 bit) 81  
MOVE 99  
MUL\_DI 81  
MUL\_I 77  
MUL\_R 89

**N**

N 28  
NEG 31  
NEG\_DI 47  
NEG\_I 46  
NEG\_R 48  
Negazione del numero in virgola mobile 48  
Negazione di un ingresso binario 18

**O**

Operazioni di bit di stato 137  
Operazioni di comando del programma 101  
Operazioni di confronto 33  
Operazioni di conteggio 53  
Operazioni di conversione 37  
Operazioni di rotazione 133  
Operazioni di salto 67  
Operazioni di scorrimento 123  
Operazioni di temporizzazione 147  
Operazioni FUP ordinate secondo il set mnemonico  
  inglese 185  
Operazioni FUP ordinate secondo il set mnemonico  
  tedesco 181  
Operazioni logiche combinatorie a parola 173

Operazioni logiche combinatorie di bit 11  
Operazioni matematiche con i numeri interi 73  
Operazioni matematiche con i numeri mobile 85  
OPN 65  
OS 140, 141  
OV 138, 139

**P**

P 29  
Parametrizzazione e conteggio all'indietro 59  
Parametrizzazione e conteggio in avanti 57  
Parametrizzazione e conteggio in avanti / all'indietro 55  
POS 32

**R**

R 23  
Registro BIE  
  bit di anomalia 143  
Resettaggio della bobina 23  
Resettaggio impostazione flip flop 25  
RET 121  
Ricavo del resto della divisione (32 bit) 83  
Richiamo di blocchi da una biblioteca 112  
Richiamo di FC/SFC senza parametri 102  
Richiamo di una multi-istanza 112  
Rilevamento del fronte RLC di discesa 28  
Rilevamento del fronte RLC di salita 29  
ROL\_DW 133  
ROR\_DW 135  
ROUND 49  
RS 25

**S**

S 24  
S\_AVERZ 160  
S\_CD 59  
S\_CU 57  
S\_CUD 55  
S\_EVERZ 156  
S\_IMPULS 152  
S\_ODT 156  
S\_ODTS 158  
S\_OFFDT 160  
S\_PEXT 154  
S\_PULSE 152  
S\_SEVERZ 158  
S\_VIMP 154  
SA 170  
Salto indietro 121  
Salto nel blocco assoluto 68  
Salto nel blocco se 0 (condizionato) 70  
Salto nel blocco se 1 (condizionato) 69  
SAVE 30  
SC 61  
SD 166  
SE 164, 165, 166, 167  
SF 170

SHL\_DW 130  
SHL\_W 127  
SHR\_DI 126  
SHR\_DW 131  
SHR\_I 124  
SHR\_W 129  
SI 162  
SIN 96  
Somma di numeri in virgola mobile 87  
Somma di numeri interi (16 bit) 75  
Somma di numeri interi (32 bit) 79  
Sommaro 11, 37, 53, 67, 73, 85, 101, 137, 147, 173  
Sommaro delle operazioni di bit di stato 137  
Sommaro delle operazioni di comando del programma 101  
Sommaro delle operazioni di confronto 33  
Sommaro delle operazioni di conteggio 53  
Sommaro delle operazioni di conversione 37  
Sommaro delle operazioni di rotazione 133  
Sommaro delle operazioni di salto 67  
Sommaro delle operazioni di scorrimento 123  
Sommaro delle operazioni di temporizzazione 147  
Sommaro delle operazioni logiche combinatorie a parola 173  
Sommaro delle operazioni logiche combinatorie di bit 11  
Sommaro delle operazioni matematiche con i numeri interi 73  
Sommaro delle operazioni matematiche con i numeri mobile 85  
Sottrazione di numeri in virgola mobile 88  
Sottrazione di numeri interi (16 bit) 76  
Sottrazione di numeri interi (32 bit) 80  
SP 162  
SQR 92  
SQRT 93  
SR 26  
SS 168  
SUB\_DI 80  
SUB\_I 76  
SUB\_R 88  
SV 164

SZ 61

## T

TAN 96  
Tipi di confronto 33  
TRUNC 50

## U

UO 142

## V

Valutazione dei bit della parola di stato nelle operazioni in virgola fissa 74  
Valutazione dei bit della parola di stato nelle operazioni in virgola mobile 86

## W

WAND\_DW 177  
WAND\_W 174  
WOR\_DW 178  
WOR\_W 175  
WXOR\_DW 179  
WXOR\_W 176

## X

XOR 16

## Z

Z\_RUECK 59  
Z\_VORW 57  
ZAEHLER 55  
ZR 64  
ZV 63