



Allen-Bradley

Controllori Logix5000

(Num. cat.) 1756-L1, -L1Mx

## **Manuale di riferimento delle istruzioni generali**



## Informazioni importanti per gli utenti

A causa della varietà d'uso dei prodotti descritti in questa pubblicazione, i responsabili dell'applicazione ed utilizzo di questa apparecchiatura di controllo devono accertarsi che sia stato fatto il possibile per rendere l'applicazione ed ogni suo uso conforme a tutti i requisiti di sicurezza, compresi qualsiasi legge, regolamento, codice o norma applicabile.

Le figure, gli schemi, i programmi di esempio e gli esempi di configurazione riportati in questa guida hanno uno scopo esclusivamente illustrativo. Poiché vi sono molte variabili e requisiti associati ad ogni particolare installazione, l'Allen-Bradley non si assume alcuna responsabilità civile né penale (ivi compresa la responsabilità di proprietà intellettuale) per l'uso effettivo basato sugli esempi riportati in questa pubblicazione.

La pubblicazione Allen-Bradley SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (disponibile presso gli uffici Allen-Bradley di zona), descrive alcune importanti differenze tra le apparecchiature allo stato solido ed i dispositivi elettromagnetici, da tenere in considerazione quando si utilizzano prodotti come quelli descritti in questa pubblicazione.

È proibita la riproduzione totale o parziale del contenuto di questa pubblicazione protetta da copyright, senza il permesso scritto della Rockwell Automation.

Nel presente manuale vengono utilizzate delle note per richiamare l'attenzione su alcune considerazioni sulla sicurezza:



**ATTENZIONE:** identifica informazioni su pratiche o circostanze che possono causare infortuni o morte alle persone, danni alle proprietà o perdite economiche

---

I segnali di Attenzione permettono di:

- identificare un pericolo
- evitare un pericolo
- riconoscerne le conseguenze

**Importante:** Identifica informazioni fondamentali per un'applicazione e un funzionamento corretti del prodotto.



Segnala un consiglio o una nota

Allen-Bradley, ControlLogix, DH+, Logix5000, Logix5550, MicroLogix, PLC-2, PLC-3, PLC-5, Rockwell Software, RSLinx, RSNetWorx ed SLC sono marchi registrati della Rockwell Automation.

ControlNet è un marchio registrato di ControlNet International, Ltd.

DeviceNet è un marchio registrato della Open DeviceNet Vendor Association.

Ethernet è un marchio registrato della Digital Equipment Corporation, Intel e Xerox Corporation.

### Introduzione

Questo documento contiene informazioni nuove ed aggiornate. Alcune informazioni, inoltre, sono state cancellate o spostate in un altro manuale. Per facilitare la ricerca degli aggiornamenti, sono state inserite delle barre di revisione come quella mostrata accanto a questo paragrafo.

### Informazioni aggiornate

La seguente riporta i cambiamenti più significativi apportati al documento rispetto all'ultima versione:

<b>Per queste informazioni nuove o aggiornate:</b>	<b>Vedere il capitolo:</b>
Specificazione dei dettagli di comunicazione (Scheda Comunicazione) di un'istruzione di messaggio	3
Accesso all'oggetto TASK	3
Formattazione delle espressioni	4, 5 e 7
Uso degli operatori ABS, MOD e TRN	4, 5 e 7
Frazioni nel risultato di una DIV e SQR	5
Istruzioni di valore assoluto e modulo	5
Istruzione troncata	15
Valori immediati	Appendice A
Conversione di dati	Appendice A
Tempi di esecuzione delle istruzioni	Appendice C
Requisiti di memoria per l'estensione del segno	Appendice D
Requisiti di memoria per il rimpimento con zeri	Appendice D

### Informazioni eliminate

La seguente tabella riporta le informazioni che sono state eliminate da questo manuale ma che possono essere reperite in altri manuali:

<b>Per informazioni su:</b>	<b>Consultare questo manuale:</b>
Istruzioni di movimento	<i>Set di istruzioni di movimento per controllori Logix5000 – Manuale di riferimento</i> , pubblicazione 1756-6.4.3IT
Terminologia relativa al Logix5000	<i>Controllori Logix5000 – Manuale dell'utente</i> , pubblicazione 1756-6.5.12IT

**Nota:**

## Dove trovare un'istruzione

Utilizzare la seguente tabella per individuare le istruzioni contenute in questo manuale. La sigla **MIM** accanto ad un'istruzione rimanda alla pubblicazione *Istruzioni di movimento dei controllori Logix5000 – Manuale di riferimento* pubblicazione 1756-6.4.3IT.

Istruzione:	Pagina o manuale:
ABS	5-19
ACS	13-10
ADD	5-6
AFI	10-15
AND	6-9
ASN	13-8
ATN	13-12
AVE	7-35
BRK	11-5
BSL	8-2
BSR	8-5
BTD	6-5
BTR (tipo MSG)	3-2
BTW (tipo MSG)	3-2
CLR	6-8
CMP	4-2
COP	7-28
COS	13-4
CPT	5-2
CTD	2-14
CTU	2-11
DDT	12-9
DEG	15-2
DIV	5-12
DTR	12-16
EQU	4-6
FAL	7-7
FBC	12-2
FFL	8-8
FFU	8-14
FLL	7-32
FOR	11-2
FRD	15-6
FSC	7-18

Istruzione:	Pagina o manuale:
GEQ	4-8
GRT	4-10
GSV	3-27
JMP	10-2
JSR	10-4
LBL	10-2
LEQ	4-12
LES	4-14
LFL	8-20
LFU	8-26
LIM	4-16
LN	14-2
LOG	14-4
MAAT	<b>MIM</b>
MAFR	<b>MIM</b>
MAG	<b>MIM</b>
MAH	<b>MIM</b>
MAHD	<b>MIM</b>
MAJ	<b>MIM</b>
MAM	<b>MIM</b>
MAPC	<b>MIM</b>
MAR	<b>MIM</b>
MAS	<b>MIM</b>
MASD	<b>MIM</b>
MASR	<b>MIM</b>
MATC	<b>MIM</b>
MAW	<b>MIM</b>
MCCP	<b>MIM</b>
MCD	<b>MIM</b>
MCR	10-11
MDF	<b>MIM</b>
MDO	<b>MIM</b>
MDR	<b>MIM</b>
MDW	<b>MIM</b>

Istruzione:	Pagina o manuale:
MEQ	4-19
MGPS	<b>MIM</b>
MGS	<b>MIM</b>
MGSD	<b>MIM</b>
MGSP	<b>MIM</b>
MGSR	<b>MIM</b>
MOD	5-14
MOV	6-2
MRAT	<b>MIM</b>
MRHD	<b>MIM</b>
MRP	<b>MIM</b>
MSF	<b>MIM</b>
MSG	3-2
MSO	<b>MIM</b>
MUL	5-10
MVM	6-3
NEG	5-18
NEQ	4-22
NOP	10-16
NOT	6-15
ONS	1-9
OR	6-11
OSF	1-14
OSR	1-11
OTE	1-6
OTL	1-7
OTU	1-8
PID	12-19
RAD	15-3
RES	2-18
RET	10-4, 11-6
RTO	2-8
SBR	10-4
SIN	13-2

---

<b>Istruzione:</b>	<b>Pagina o manuale:</b>
SQI	9-2
SQL	9-11
SQO	9-6
SQR	5-16
SRT	7-39
SSV	3-27
STD	7-42
SUB	5-8
TAN	13-6
TND	10-10
TOD	15-4
TOF	2-5
TON	2-2
TRUN	15-8
UID	10-13
UIE	10-14
XIC	1-2
XIO	1-4
XOR	6-13
XPY	14-6

## Uso di questo manuale

### Introduzione

Questo manuale fa parte di una serie di pubblicazioni relative al ControlLogix.

<b>Obiettivo/Scopo:</b>	<b>Documenti:</b>
Installazione del controllore e dei suoi componenti	<i>Controllore Logix5550 – Messa in funzione rapida</i> , pubblicazione 1756-10.1IT <i>Scheda memoria Logix5550 – Istruzioni per l'installazione</i> , pubblicazione 1756-5.33IT
Uso del controllore	<i>Logix5000 Controllers User Manual</i> , publication 1756-6.5.12
Programmazione del controllore per applicazioni sequenziali	<i>Set di istruzioni generali per controllori Logix5000 – Manuale di riferimento</i> , pubblicazione 1756-6.4.1IT
<b>Voi siete qui</b> 	
Programmazione del controllore per applicazioni di movimento	<i>Set di istruzioni di movimento per controllori Logix5000 – Manuale di riferimento</i> , pubblicazione 1756-6.4.3IT
Configurazione e comunicazione con moduli I/O digitali	<i>Moduli digitali – Manuale dell'utente</i> , pubblicazione 1756-6.5.8IT
Configurazione dei moduli I/O analogici	<i>Moduli analogici – Manuale dell'utente</i> , pubblicazione 1756-6.5.9IT
Configurazione ed uso dei moduli di movimento	<i>Modulo di movimento ControlLogix – Manuale dell'utente</i> , pubblicazione 1756-6.5.16IT.
Scelta ed installazione di uno chassis	<i>Chassis ControlLogix – Istruzioni per l'installazione</i> , pubblicazione 1756-5.69IT
Scelta ed installazione di un alimentatore	<i>Alimentatore ControlLogix – Istruzioni per l'installazione</i> , pubblicazione 1756-5.11IT
Importazione di un file di testo o di tag in un progetto	<i>Logix5550 Controller Import/Export Reference Manual</i> , publication 1756-6.8.4
Esportazione di un progetto o di tag in un file di testo	
Conversione di un'applicazione per PLC-5 o SLC 500 in un'applicazione per Logix5000	<i>Logix5550 Controller Converting PLC-5 or SLC 500 Logic to Logix5550 Logic Reference Manual</i> , publication 1756-6.8.5

### A chi è rivolto questo manuale

Questo documento si rivolge ai programmatori per fornire loro dettagliate informazioni riguardo le istruzioni disponibili con il controllore Logix5550. È necessario essere già a conoscenza di come il controllore Logix5550 archivia e elabora i dati.

Si consiglia ai programmatori non esperti di leggere tutte le informazioni riguardanti un'istruzione prima di utilizzare la stessa. I programmatori esperti possono fare riferimento alle informazioni sulle istruzioni per verificare eventuali dettagli.

## Scopo di questo manuale

Il presente manuale fornisce informazioni su ciascuna istruzione supportata dai controllori Logix5000. Ogni descrizione segue il seguente formato.

<b>Questa sezione:</b>	<b>Fornisce questo tipo di informazioni:</b>
Nome dell'istruzione	identifica l'istruzione specifica se si tratta di un'istruzione di ingresso o di uscita
Operandi	elenca tutti gli operandi dell'istruzione
Struttura di controllo	elenca i bit ed i valori di stato del controllo dell'istruzione, se presenti
Descrizione	descrive l'uso dell'istruzione definisce le differenze tra quando un'istruzione abilitata e quando è disabilitata
Esecuzione	indica in che modo l'istruzione funziona: <ul style="list-style-type: none"> <li>• durante la prescansione</li> <li>• se la condizione del ramo di ingresso è falsa</li> <li>• se la condizione del ramo di ingresso è vera</li> </ul>
Indicatori di stato aritmetico	indica se l'istruzione influenza gli indicatori di stato aritmetico o meno vedere appendice A
Condizioni di errore	indica se l'istruzione genera errori gravi o minori in caso positivo, viene indicato il tipo ed il codice di errore
Esempio	fornisce almeno un esempio di programmazione comprende una descrizione esplicitiva per ogni esempio

## Informazioni comuni per tutte le istruzioni

Il set di istruzioni del Logix5000 presenta degli attributi comuni:

### Per queste informazioni: Vedere questa appendice:

attributi comuni	l'appendice A definisce: <ul style="list-style-type: none"> <li>• indicatori di stato aritmetico</li> <li>• tipi di dati (data type)</li> <li>• parole chiave</li> </ul>
matrici	l'appendice B definisce le matrici e spiega in che modo il controllore le manipola
strutture	l'appendice C illustra le strutture di controllo supportate dal controllore

## Convenzioni e termini correlati Impostato e azzerato

In questo manuale si utilizzano i termini impostato ed azzerato per indicare lo stato dei bit (booleani) e dei valori (non booleani):

### Questo termine: Significa:

impostato	il bit è impostato a 1 (ON) un valore è impostato su un numero diverso da zero
azzerato	il bit è a 0 (OFF) tutti i bit di un valore sono a 0

Nella sezione relativa agli operandi, i tipi di dati (data type) in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente e richiede meno memoria se tutti gli operandi dell'istruzione utilizzano il medesimo tipo di dati ottimali, in genere DINT o REAL.

### Condizione del ramo

Il controllore valuta le istruzioni ladder in base alla condizione del ramo precedente l'istruzione (condizione del ramo di ingresso). In base alla condizione di ingresso del ramo e all'istruzione, il controllore imposta la condizione del ramo seguente l'istruzione (condizione del ramo di uscita) che, a sua volta, determina la successiva istruzione.



Se la condizione di ingresso del ramo per un'istruzione di ingresso è vera, il controllore valuta l'istruzione ed imposta la condizione del ramo di uscita in base ai risultati dell'istruzione. Se l'istruzione è vera, la condizione del ramo di uscita è vera; se l'istruzione è falsa la condizione del ramo di uscita è falsa.

**Nota:**

	<b>Capitolo 1</b>	
<b>Istruzioni di bit</b>	Introduzione	1-1
<b>(XIC, XIO, OTE, OTL, OTU, ONS,</b>	Esamina se chiuso (XIC)	1-2
<b>OSR, OSF)</b>	Esamina se aperto (XIO)	1-4
	Eccita uscita (OTE)	1-6
	Aggancia uscita (OTL)	1-7
	Sgancia uscita (OTU)	1-8
	One Shot (ONS)	1-9
	One shot sul fronte di salita (OSR)	1-11
	One shot sul fronte di discesa (OSF)	1-14
	<b>Capitolo 2</b>	
<b>Istruzioni di timer e contatori</b>	Introduzione	2-1
<b>(TON, TOF, RTO, CTU, CTD, RES)</b>	Timer ritardato all'eccitazione (TON)	2-2
	Timer ritardato alla diseccitazione (TOF)	2-5
	Timer ritardato all'eccitazione ritentivo (RTO)	2-8
	Conteggio incrementale (CTU)	2-11
	Conteggio decrementale (CTU)	2-14
	Ripristino (RES)	2-18
	<b>Capitolo 3</b>	
<b>Istruzioni di Input/Output</b>	Introduzione	3-1
<b>(MSG, GSV, SSV)</b>	Messaggio (MSG)	3-2
	Codici di errore MSG	3-7
	Codici di errore ControlLogix (CIP)	3-7
	Codici di errore estesi ControlLogix	3-8
	Codici di errore di PLC e SLC (.ERR)	3-9
	Codici di errore estesi di PLC e SLC (.EXERR)	3-10
	Codici di errore dei trasferimenti a blocchi	3-11
	Codici di errore del Logix5550	3-12
	Codici di errore estesi del Logix5550	3-12
	Specificazione dei dettagli di comunicazione	
	(Scheda Configurazione)	3-13
	Specificazione di messaggi CIP	3-14
	Uso di messaggi CIP generici per ripristinare moduli I/O	3-15
	Specificazione di messaggi per PLC-5	3-16
	Specificazione di messaggi per SLC	3-17
	Specificazione dei messaggi a trasferimento a blocchi	3-17
	Specificazione di messaggi per PLC-3	3-18
	Specificazione di messaggi per PLC-2	3-19
	Esempi di configurazione MSG:	3-20
	Specificazione dei dettagli di comunicazione	
	(Scheda Comunicazione)	3-21
	Specificazione di un percorso di connessione	3-21
	Specificazione di un metodo di comunicazione:	3-25
	Selezione di un'opzione con cache:	3-26
	Otteni valore di sistema (GSV) ed Imposta valore	
	di sistema (SSV)	3-27

Oggetti GSV/SSV .....	3-29
Accesso all'oggetto AXIS .....	3-30
Accesso all'oggetto CONTROLLER .....	3-37
Accesso all'oggetto CONTROLLERDEVICE .....	3-37
Accesso all'oggetto CST .....	3-39
Accesso all'oggetto DF11 .....	3-40
Accesso all'oggetto FAULTLOG .....	3-43
Accesso all'oggetto MESSAGE .....	3-43
Accesso all'oggetto MODULE .....	3-45
Accesso all'oggetto MOTIONGROUP .....	3-46
Accesso all'oggetto PROGRAM .....	3-47
Accesso all'oggetto ROUTINE .....	3-47
Accesso all'oggetto SERIALPORT .....	3-48
Accesso all'oggetto TASK .....	3-49
Accesso all'oggetto WALLCLOCKTIME .....	3-50
Esempio di programmazione GSV/SSV .....	3-51
Ottenere informazioni sugli errori .....	3-51
Impostazione degli indicatori di abilitazione e disabilitazione .....	3-52

#### Capitolo 4

#### Istruzioni di confronto (CMP, EQU, GEQ, GRT, LEQ, LES, LIM, MEQ, NEQ)

Introduzione .....	4-1
Confronto (CMP) .....	4-2
Operatori validi .....	4-4
Formattazione delle espressioni .....	4-4
Determinazione dell'ordine delle operazioni .....	4-5
Uguale a (EQU) .....	4-6
Maggiore di o Uguale a (GEQ) .....	4-8
Maggiore di (GRT) .....	4-10
Minore di o Uguale a (LEQ) .....	4-12
Minore di (LES) .....	4-14
Limite (LIM) .....	4-16
Maschera uguale a (MEQ) .....	4-19
Immissione di un valore di maschera immediato .....	4-19
Diverso da (NEQ) .....	4-22

<b>Istruzioni di</b>	<b>Capitolo 5</b>	
<b>Calcolo/Matematiche</b>	Introduzione	5-1
<b>(CPT, ADD, SUB, MUL, DIV, MOD,</b>	Calcola (CPT)	5-2
<b>SQR, NEG, ABS)</b>	Operatori validi	5-4
	Formattazione delle espressioni	5-4
	Determinazione dell'ordine delle operazioni	5-5
	Somma (ADD)	5-6
	Sottrai (SUB)	5-8
	Moltiplica (MUL)	5-10
	Dividi (DIV)	5-12
	Modulo (MOD)	5-14
	Radice quadrata (SQR)	5-16
	Nega (NEG)	5-18
	Valore assoluto (ABS)	5-19
	<b>Capitolo 6</b>	
<b>Istruzioni di</b>	Introduzione	6-1
<b>Spostamento/Logica</b>	Sposta (MOV)	6-2
<b>(MOV, MVM, BTD, CLR,</b>	Sposta con maschera (MVM)	6-3
<b>AND, OR, XOR, NOT)</b>	Immissione di un valore di maschera immediato	6-3
	Distribuisce campo di bit (BTD)	6-5
	Azzerata (CLR)	6-8
	AND di bit (AND)	6-9
	OR di bit (OR)	6-11
	OR esclusivo di bit (XOR)	6-13
	NOT di bit (NOT)	6-15
	<b>Capitolo 7</b>	
<b>Istruzioni di Matrici</b>	Introduzione	7-1
<b>(File)/Varie Istruzioni</b>	Scelta della modalità operativa	7-1
<b>(FAL, FSC, COP, FLL, AVE,</b>	Modalità All (Continua)	7-2
<b>SRT, STD)</b>	Modalità discreta	7-3
	Modalità incrementale	7-4
	Aritmetica e logica di file (FAL)	7-6
	Operatori validi	7-15
	Formattazione delle espressioni	7-15
	Determinazione dell'ordine delle operazioni	7-16
	Ricerca e confronto file (FSC)	7-17
	Operatori validi	7-25
	Formattazione delle espressioni	7-25
	Determinazione dell'ordine delle operazioni	7-26
	Copia file (COP)	7-27
	Riempì file (FLL)	7-31
	Media file (AVE)	7-34
	Ordina file (SRT)	7-38
	Deviazione standard file (STD)	7-41

	<b>Capitolo 8</b>	
<b>Istruzioni di Matrice (File)/Scorrimento (BSL, BSR, FFL, FFU, LFL, LFU)</b>	Introduzione . . . . .	8-1
	Sposta bit a sinistra (BSL) . . . . .	8-2
	Sposta bit a destra (BSL) . . . . .	8-5
	Carica FIFO (FFL) . . . . .	8-8
	Scarica FIFO (FFU) . . . . .	8-14
	Carica LIFO (FFL) . . . . .	8-20
	Scarica LIFO (FFU) . . . . .	8-26
	<b>Capitolo 9</b>	
<b>Istruzioni sequenziatore (SQI, SQO, SQL)</b>	Introduzione . . . . .	9-1
	Sequenziatore di ingresso (SQI) . . . . .	9-2
	Immissione di un valore di maschera immediato . . . . .	9-3
	Uso di SQI senza SQO . . . . .	9-5
	Sequenziatore di uscita (SQO) . . . . .	9-6
	Immissione di un valore di maschera immediato . . . . .	9-7
	Uso di SQI con SQO . . . . .	9-9
	Ripristino della posizione dell'istruzione SQO . . . . .	9-10
	Carica sequenziatore (SQL) . . . . .	9-11
	<b>Capitolo 10</b>	
<b>Istruzioni di controllo programma (JMP, LBL, JSR, RET, SBR, TND, MCR, UID, UIE, AFI, NOP)</b>	Introduzione . . . . .	10-1
	Salta all'etichetta (JMP)	
	Etichetta (LBL) . . . . .	10-2
	Salta alla subroutine (JSR)	
	Subroutine (SBR)	
	Ritorno (RET) . . . . .	10-4
	Fine temporanea (TND) . . . . .	10-10
	Relè di controllo principale (MCR) . . . . .	10-11
	Disabilita interrupt utente (UID) . . . . .	10-13
	Abilita interrupt utente (UIE) . . . . .	10-14
	Istruzione sempre falso (AFI) . . . . .	10-15
	Nessuna operazione (NOP) . . . . .	10-16
	<b>Capitolo 11</b>	
<b>Istruzioni For/Break (FOR, BRK, RET)</b>	Introduzione . . . . .	11-1
	For (FOR) . . . . .	11-2
	Break (BRK) . . . . .	11-5
	Ritorno (RET) . . . . .	11-6

**Istruzioni speciali  
(FBC, DDT, DTR, PID)**

**Capitolo 12**

Introduzione	12-1
Confronto bit di file (FBC)	12-2
Scelta della modalità di ricerca	12-4
Rileva diagnostica (DDT)	12-9
Scelta della modalità di ricerca	12-11
Transizione dati (DTR)	12-16
Immissione di un valore di maschera immediato	12-16
Proporzionale, integrale e derivativo (PID)	12-19
Configurazione di un'istruzione PID	12-24
Calcolo parametri	12-24
Configurazione	12-25
Avvisi	12-25
Conversione in scala	12-26
Uso delle istruzioni PID	12-26
Antiaccumulo dell'integrale e trasferimento graduale da manuale ad automatico	12-28
Temporizzazione dell'istruzione PID	12-29
Riavviamento graduale	12-32
Attenuazione derivativa	12-34
Impostazione della banda morta	12-34
Uso del limite dell'uscita	12-35
Compensazione anticipata o bias dell'uscita	12-35
Anelli in cascata	12-35
Controllo di un rapporto	12-36

**Istruzioni trigonometriche  
(SIN, COS, TAN, ASN, ACS, ATN)**

**Capitolo 13**

Introduzione	13-1
Seno (SIN)	13-2
Coseno (COS)	13-4
Tangente (TAN)	13-6
Arcoseno (ASN)	13-8
Arcocoseno (ACS)	13-10
Arcotangente (ATN)	13-12

**Istruzioni di matematica  
avanzata (LN, LOG, XPY)**

**Capitolo 14**

Introduzione	14-1
Logaritmo naturale (LN)	14-2
Logaritmo in base 10 (LOG)	14-4
X alla potenza di Y (XPY)	14-6

**Istruzioni di conversione  
matematica  
(DEG, RAD, TOD, FRD, TRN)**

**Capitolo 15**

Introduzione	15-1
Gradi (DEG)	15-2
Radiani (RAD)	15-3
Conversione in BCD (TOD)	15-4
Conversione in interi (FRD)	15-6
Tronca (TRN)	15-8

**Attributi comuni****Appendice A**

Introduzione .....	A-1
Parole chiave di stato aritmetico .....	A-1
Se il tipo di dati è SINT .....	A-2
Se il tipo di dati è INT .....	A-2
Se il tipo di dati è DINT .....	A-2
Altre parole chiave .....	A-4
Tipi di dati .....	A-5
Valori immediati .....	A-6
Conversione di dati .....	A-6
Da SINT o INT in DINT .....	A-7
Da intero a REAL .....	A-9
Da DINT in SINT o INT .....	A-9
Da REAL in intero .....	A-10

**Concetti sulle matrici****Appendice B**

Visualizzazione di una matrice come insieme di elementi .....	B-1
Indicizzazione mediante matrici .....	B-3
Specificazione di bit all'interno di matrici .....	B-4
Visualizzazione di una matrice come blocco di memoria .....	B-4
Come il controllore memorizza i dati della matrice .....	B-5
Variazione di una dimensione .....	B-6
Allocazione di memoria per matrici .....	B-6

**Tempo d'esecuzione****Appendice C**

Introduzione .....	C-1
Tabelle di riferimento .....	C-3

**Utilizzo della memoria****Appendice D**

Introduzione .....	D-1
Conversione dei dati .....	D-1
Requisiti di memoria per l'estensione del segno .....	D-2
Requisiti di memoria per il rimpimento con zeri .....	D-2
Istruzioni .....	D-3
Indici di matrice .....	D-7

## **Istruzioni di bit**

### **(XIC, XIO, OTE, OTL, OTU, ONS, OSR, OSF)**

#### **Introduzione**

Utilizzare le istruzioni di bit (tipo relè) per monitorare e controllare lo stato dei bit.

<b>Se si desidera:</b>	<b>Utilizzare questa istruzione:</b>	<b>Vedere pagina:</b>
abilitare le uscite quando un bit è impostato	XIC	1-2
abilitare le uscite quando un bit è azzerato	XIO	1-4
impostare un bit	OTE	1-6
impostare un bit (ritentivo)	OTL	1-7
azzerare un bit (ritentivo)	OTU	1-8
abilitare le uscite per una scansione ogni volta che un ramo diventa vero	ONS	1-9
impostare un bit per una scansione ogni volta che un ramo diventa vero	OSR	1-11
impostare un bit per una scansione ogni volta che il ramo diventa falso	OSF	1-14

### Esamina se chiuso (XIC)

L'istruzione XIC è un'istruzione di ingresso.

**Operandi:**

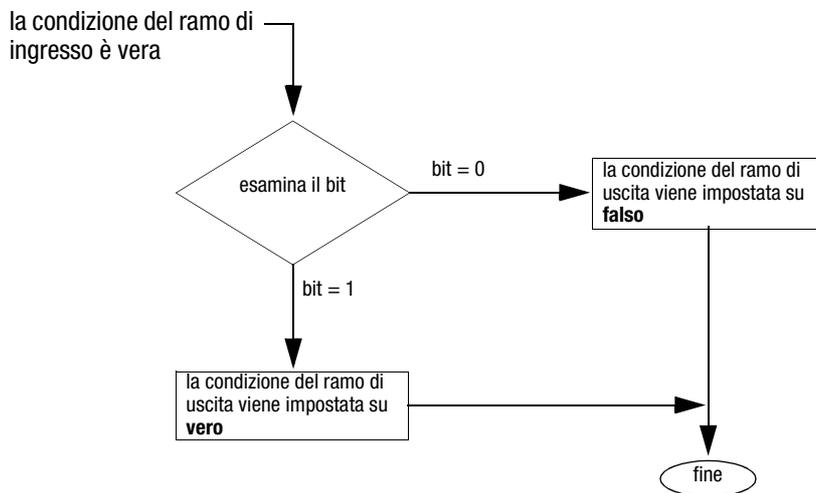


Operando:	Tipo:	Formato:	Descrizione:
data bit	BOOL	tag	bit da testare

**Descrizione:** L'istruzione XIC esamina il bit per verificare se è impostato.

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.

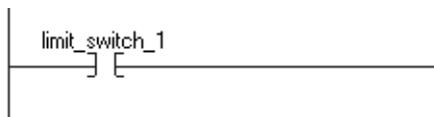


**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di XIC:**

**esempio 1**



Se *limit\_switch\_1* è impostato, viene abilitata la successiva istruzione (la condizione del ramo di uscita è vera).

**esempio 2**



Se *S:V* è impostato (indica che si è verificato un overflow), viene abilitata la successiva istruzione (la condizione del ramo di uscita è vera).

**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<code>XIC(<i>data_bit</i>);</code>
testo ASCII	<code>XIC <i>data_bit</i></code>

**Istruzioni correlate:** XIO

### Esamina se aperto (XIO)

L'istruzione XIO è un'istruzione di ingresso.



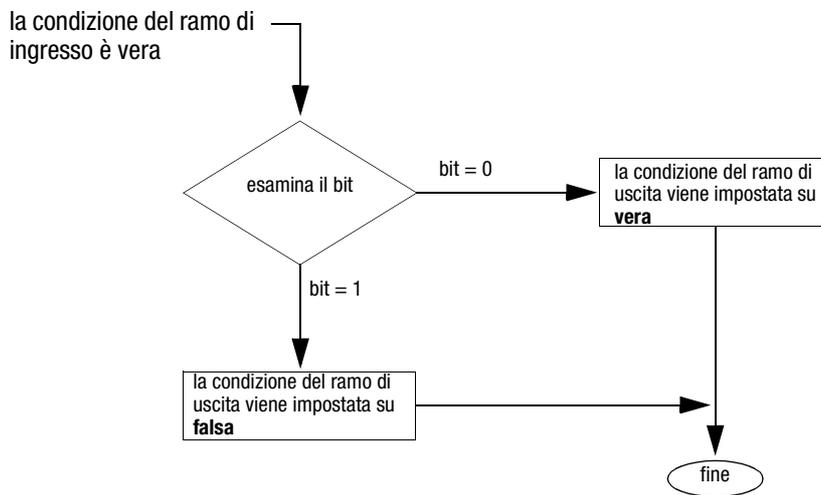
**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
data bit	BOOL	tag	bit da testare

**Descrizione:** L'istruzione XIO esamina il bit per verificare se è azzerato.

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.

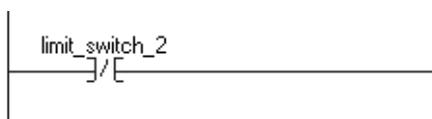


**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di XIO:**

**esempio 1**



Se *limit\_switch\_2* è azzerato, viene abilitata la successiva istruzione (la condizione del ramo di uscita è vera).

**esempio 2**



Se *S:V* è azzerato (indica che non si è verificato un overflow), viene abilitata la successiva istruzione (la condizione del ramo di uscita è vera).

**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<code>XIO(<i>data_bit</i>);</code>
testo ASCII	<code>XIO <i>data_bit</i></code>

**Istruzioni correlate:** XIC

**Eccita uscita (OTE)**

L'istruzione OTE è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
data bit	BOOL	tag	bit da impostare o da azzerare

**Descrizione:** L'istruzione OTE imposta o azzerata il bit.

Se l'istruzione OTE è abilitata, il controllore imposta il bit. Se l'istruzione OTE è disabilitata, il controllore azzerata il bit.

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit viene azzerato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit viene azzerato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il bit è impostato. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati**Condizioni di errore:** nessuna**Esempio di OTE:**

Se è abilitata, l'istruzione OTE imposta (accende) la *light\_1*.  
Se è disabilitata, l'istruzione OTE azzerata (spegne) la *light\_1*.

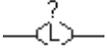
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>OTE (data_bit) ;</code>
testo ASCII	<code>OTE data_bit</code>

**Istruzioni correlate:** OTL, OTU

**Aggancia uscita (OTL)**

L'istruzione OTL è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
data bit	BOOL	tag	bit da impostare

**Descrizione:** L'istruzione OTL imposta (blocca) il bit.

Se è abilitata, l'istruzione OTL imposta il bit. Il bit rimane impostato fino a quando non viene azzerato, generalmente mediante un'istruzione OTU. Se è disabilitata, l'istruzione OTL non modifica lo stato del bit dati.

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit non è modificato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit non è modificato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il bit è impostato. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di OTL:**

Se è abilitata, l'istruzione OTL imposta la *light\_2*. Questo bit rimane impostato fino a quando non viene azzerato, generalmente mediante un'istruzione OTU.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>OTL(<i>data_bit</i>);</code>
testo ASCII	<code>OTL <i>data_bit</i></code>

**Istruzioni correlate:** OTU, OTE

**Sgancia uscita (OTU)**

L'istruzione OTU è un'istruzione di uscita.

**Operandi:**

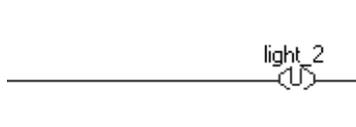
Operando:	Tipo:	Formato:	Descrizione:
data bit	BOOL	tag	bit da azzerare

**Descrizione:** L'istruzione OTU azzerata (sblocca) il bit.

Se è abilitata, l'istruzione OTU azzerata il bit. Se è disabilitata, l'istruzione OTU non modifica lo stato del bit dati.

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit non è modificato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit non è modificato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il bit viene azzerato. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati**Condizioni di errore:** nessuna**Esempio di OTU:**Se è abilitata, l'istruzione OTU azzerata la *luce\_2*.**Altri formati:**

Formato:	Sintassi:
testo neutro	OTU( <i>data_bit</i> );
testo ASCII	OTU <i>data_bit</i>

**Istruzioni correlate:** OTL, OTE

**One Shot (ONS)**

L'istruzione ONS è un'istruzione di ingresso.

—<sup>?</sup>[ONS]—

**Operandi:**

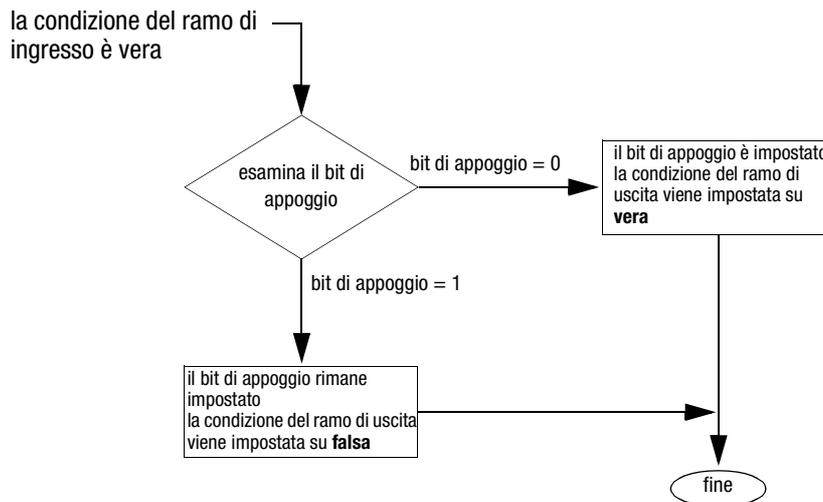
Operando:	Tipo:	Formato:	Descrizione:
storage bit	BOOL	tag	bit di appoggio interno memorizza la condizione del ramo di ingresso dall'ultima volta che l'istruzione è stata eseguita

**Descrizione:** L'istruzione ONS abilita o disabilita la restante parte del ramo, a seconda dello stato del bit di appoggio.

Se è abilitata e il bit di appoggio è azzerato, l'istruzione ONS abilita la rimanente parte del ramo. Se è disabilitata e se il bit di appoggio è impostato, l'istruzione ONS disabilita la rimanente parte del ramo.

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit di appoggio è impostato per evitare un intervento non valido durante la prima scansione. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit di appoggio viene azzerato. La condizione del ramo di uscita viene impostata su falso.



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di ONS:** in genere l'istruzione ONS va fatta precedere da un'istruzione di ingresso poiché per funzionare correttamente l'istruzione ONS viene scandita quando viene abilitata e quando viene disabilitata. Dopo che l'istruzione ONS è stata abilitata, la condizione del ramo di ingresso deve diventare falsa oppure il bit di appoggio deve essere azzerato affinché l'istruzione ONS venga di nuovo abilitata.



Questo ramo non ha alcun effetto in una scansione in cui *limit\_switch\_1* è azzerato o *storage\_1* è impostato. Nelle scansioni in cui *limit\_switch\_1* è impostato e *storage\_1* è azzerato, l'istruzione ONS imposta *storage\_1* e l'istruzione ADD fa incrementare il valore *sum* di 1. Fino a quando *limit\_switch\_1* rimane impostato, il valore *sum* rimane lo stesso. Il *limit\_switch\_1* deve nuovamente passare da zero ad uno affinché il valore *sum* possa essere incrementato di nuovo.

**Altri formati:**

Formato:	Sintassi:
testo neutro	ONS( <i>storage_bit</i> );
testo ASCII	ONS( <i>storage_bit</i> );

**Istruzioni correlate:** OSR, OSF

**One shot sul fronte di salita (OSR)**

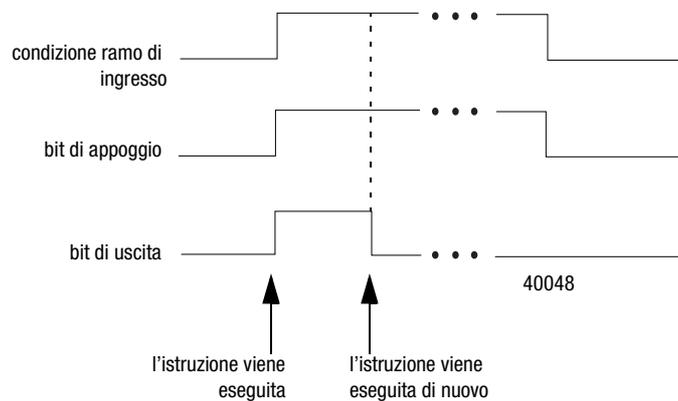
L'istruzione OSR è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
storage bit	BOOL	tag	bit di appoggio interno memorizza la condizione del ramo di ingresso dall'ultima volta che l'istruzione è stata eseguita
bit di uscita	BOOL	tag	bit da impostare

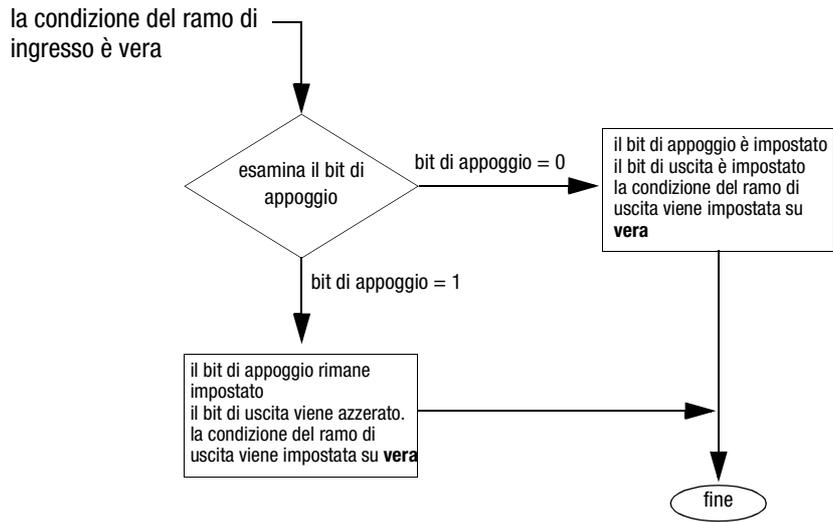
**Descrizione:** L'istruzione OSR imposta o azzerà il bit di uscita a seconda dello stato del bit di appoggio.

Se è abilitata e il bit di appoggio è azzerato, l'istruzione OSR imposta il bit di uscita. Se è abilitata e il bit di appoggio è impostato oppure se è disabilitata, l'istruzione OSR azzerà il bit di uscita.



**Esecuzione:**

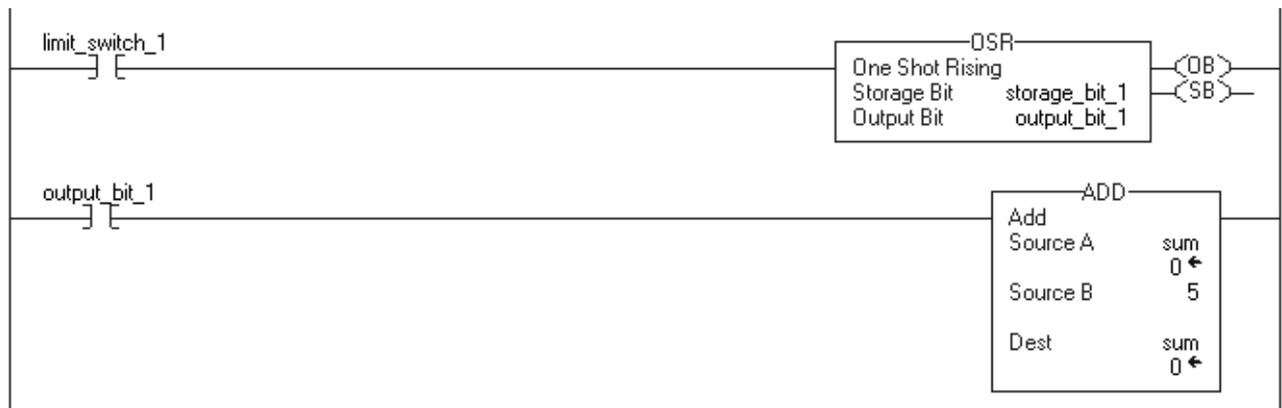
Condizione:	Azione:
prescansione	Il bit di appoggio è impostato per evitare un intervento non valido durante la prima scansione. Il bit di uscita viene azzerato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit di appoggio viene azzerato. Il bit di uscita non è modificato. La condizione del ramo di uscita viene impostata su falso.



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di OSR:**



Ogni volta che *limit\_switch\_1* passa da azzerato ad impostato, l'istruzione OSR imposta *output\_bit\_1* e l'istruzione ADD fa incrementare il valore *sum* di 5. Fino a quando *limit\_switch\_1* rimane impostato, il valore *sum* rimane lo stesso. Il *limit\_switch\_1* deve nuovamente passare da zero ad uno affinché il valore *sum* possa essere incrementato di nuovo. È possibile utilizzare *output\_bit\_1* su più rami per attivare altre operazioni

**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<code>OSR(<i>storage_bit</i>, <i>output_bit</i>);</code>
testo ASCII	<code>OSR(<i>storage_bit</i>, <i>output_bit</i>);</code>

**Istruzioni correlate:** OSF, ONS

### One shot sul fronte di discesa (OSF)

L'istruzione OSF è un'istruzione di uscita.

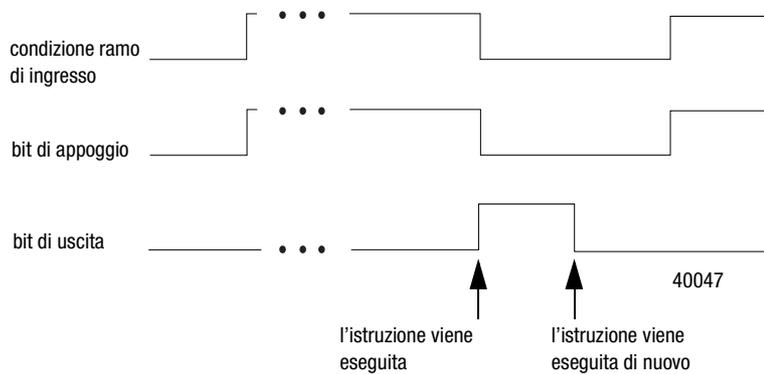
**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
storage bit	BOOL	tag	bit di appoggio interno memorizza la condizione del ramo di ingresso dall'ultima volta che l'istruzione è stata eseguita
output bit	BOOL	tag	bit da impostare

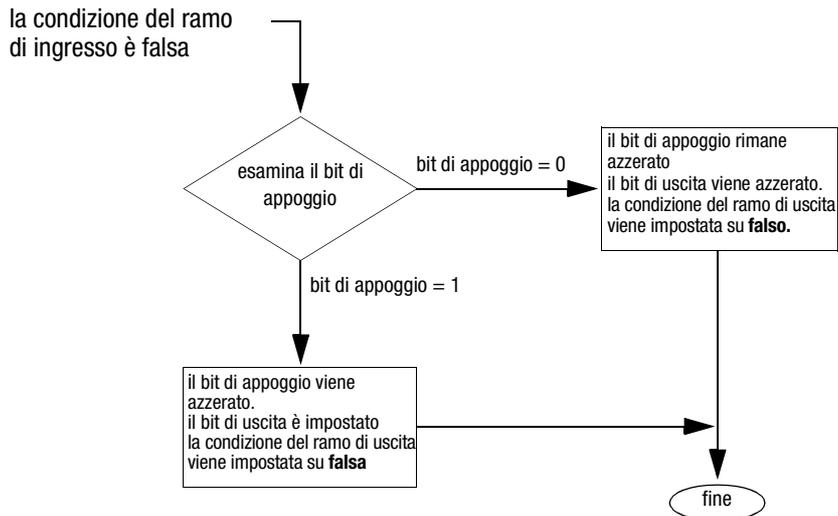
**Descrizione:** L'istruzione OSF imposta o azzerà il bit di uscita a seconda dello stato del bit di appoggio.

Se è disabilitata e il bit di appoggio è impostato, l'istruzione OSF imposta il bit di uscita. Se è disabilitata e il bit di appoggio è azzerato oppure se è abilitata, l'istruzione OSF azzerà il bit di uscita.

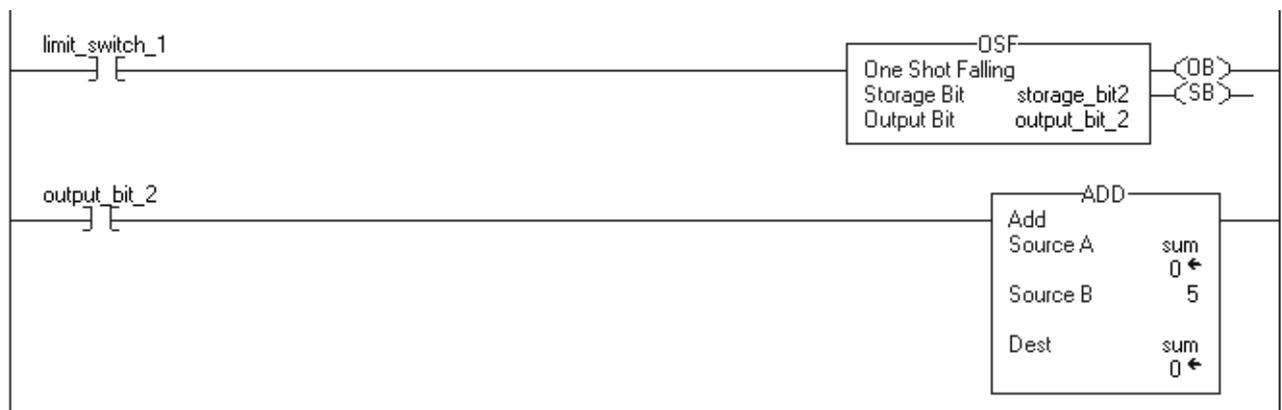


**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit di appoggio viene azzerato per evitare un intervento non valido durante la prima scansione. Il bit di uscita viene azzerato. La condizione del ramo di uscita viene impostata su falso.



la condizione del ramo di ingresso è vera	Il bit di appoggio viene impostato. Il bit di uscita viene azzerato. La condizione del ramo di uscita viene impostata su vero.
---	--

**Indicatori di stato aritmetico:** non influenzati**Condizioni di errore:** nessuna**Esempio di OSF:**

Ogni volta che *limit\_switch\_1* passa da impostato ad azzerato, l'istruzione OSF imposta *output\_bit\_2* e l'istruzione ADD fa incrementare il valore *sum* di 5. Finché *limit\_switch\_1* rimane azzerato, il valore *sum* rimane lo stesso. Il *limit\_switch\_1* deve nuovamente passare da uno a zero affinché il valore *sum* possa essere incrementato di nuovo. È possibile utilizzare *output\_bit\_2* su più rami per attivare altre operazioni

**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>OSF(storage_bit, output_bit);</i>
testo ASCII	<i>OSF(storage_bit, output_bit);</i>

**Istruzioni correlate:** OSR, ONS

---

## **Istruzioni di timer e contatori (TON, TOF, RTO, CTU, CTD, RES)**

### **Introduzione**

I timer ed i contatori controllano le operazioni in base al tempo o al numero di eventi.

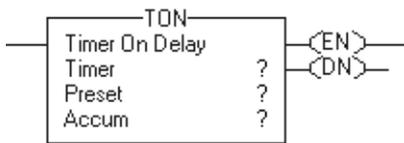
<b>Se si desidera:</b>	<b>Utilizzare questa istruzione:</b>	<b>Vedere pagina:</b>
calcolare per quanto tempo un timer rimane abilitato	TON	2-2
calcolare per quanto tempo un timer rimane disabilitato	TOF	2-5
conteggiare il tempo	RTO	2-8
contare in modo incrementale	CTU	2-11
contare in modo decrementale	CTD	2-14
azzerare un timer o un contatore	RES	2-18

La base tempo di tutti i timer è 1 msec.

## Timer ritardato all'eccitazione (TON)

L'istruzione TON è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Timer	TIMER	tag	struttura timer
Preset	DINT	immediato	quanto ritardare (tempo accumulato)
Accum	DINT	immediato	msec. totali calcolati dal timer il valore iniziale generalmente è 0

### Struttura TIMER:

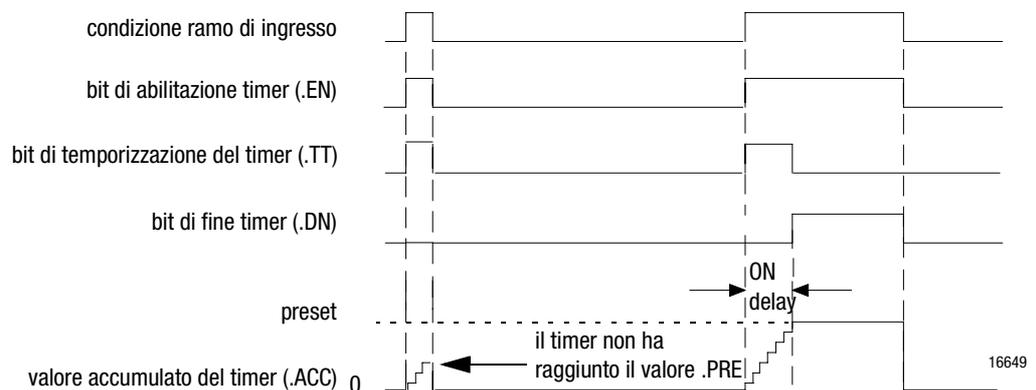
Mnemonico:	Tipo dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione TON è abilitata.
.TT	BOOL	Il bit di temporizzazione indica che è in corso un'operazione di temporizzazione
.DN	BOOL	Il bit di fine viene impostato quando $.ACC \geq .PRE$ .
.PRE	DINT	Il valore preimpostato specifica il valore (unità di 1 msec.) che il valore accumulato deve raggiungere prima che l'istruzione imposti il bit .DN.
.ACC	DINT	Il valore accumulato specifica il numero di millisecondi trascorsi dal momento dell'abilitazione dell'istruzione TON.

**Descrizione:** L'istruzione TON è un timer non ritentivo che calcola il tempo quando l'istruzione è abilitata (condizione del ramo di entrata è vera). La base tempo è sempre 1 msec. Per esempio, per un timer a 2 secondi, inserire 2000 come valore .PRE.

Se è abilitata, l'istruzione TON calcola il tempo fin quando:

- l'istruzione TON non viene disabilitata
- $.ACC \geq .PRE$

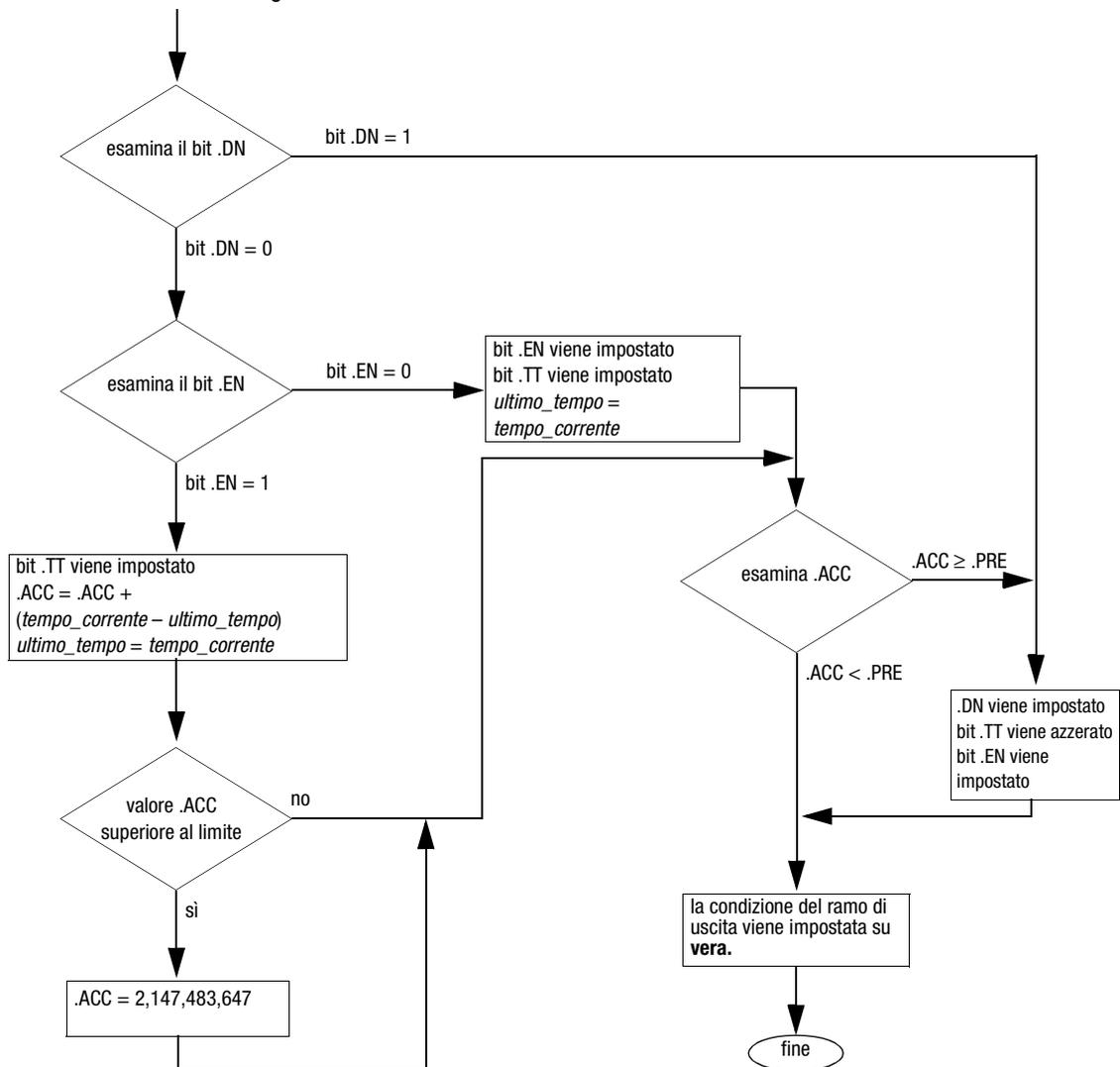
Se l'istruzione TON è disabilitata, il valore di .ACC è azzerato.



**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN è azzerato. Il bit .TT viene azzerato. Il bit .DN viene azzerato. Il valore .ACC viene azzerato. La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è falsa	Il bit .EN è azzerato. Il bit .TT viene azzerato. Il bit .DN viene azzerato. Il valore .ACC viene azzerato. La condizione del ramo di uscita viene impostata su falsa.

la condizione del ramo di ingresso è vera

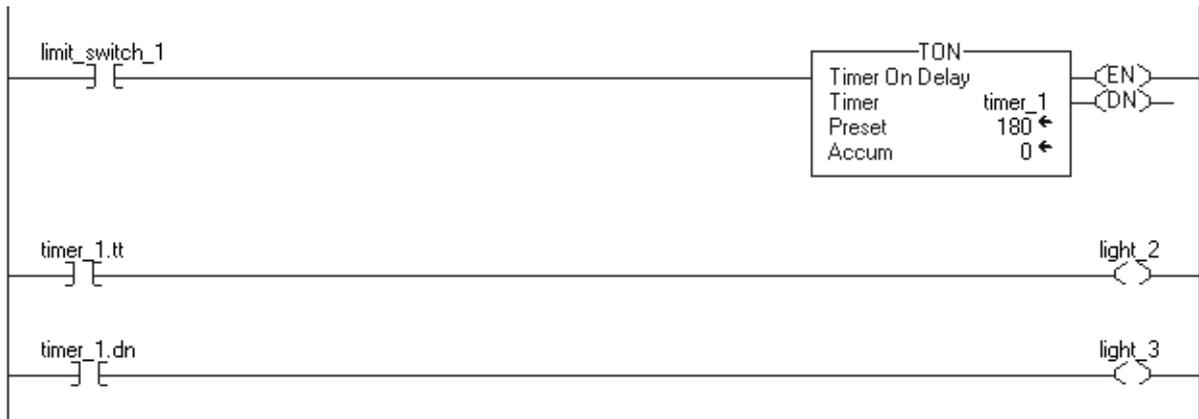


**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
.PRE < 0	4	34
.ACC < 0	4	34

**Esempio di TON:**



Se *limit\_switch\_1* è impostato, *light\_2* è accesa per 180 msec (*timer\_1* è in fase di conteggio). Se *timer\_1.acc* raggiunge 180, *light\_2* si spegne e *light\_3* si accende. La *light\_3* rimane accesa fino a quando l'istruzione TON non viene disabilitata. Se *limit\_switch\_1* viene azzerato mentre *timer\_1* è in fase di conteggio, *light\_2* si spegne.

**Altri formati:**

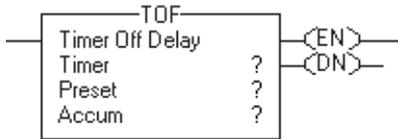
Formato:	Sintassi:
testo neutro	<code>TON(<i>timer</i>,<i>preset</i>,<i>accum</i>);</code>
testo ASCII	<code>TON <i>timer preset accum</i></code>

**Istruzioni correlate:** TOF, RTO

## Timer ritardato alla diseccitazione (TOF)

L'istruzione TOF è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Timer	TIMER	tag	struttura timer
Preset	DINT	immediato	quanto ritardare (tempo accumulato)
Accum	DINT	immediato	msec. totali calcolati dal timer il valore iniziale generalmente è 0

### Struttura TIMER:

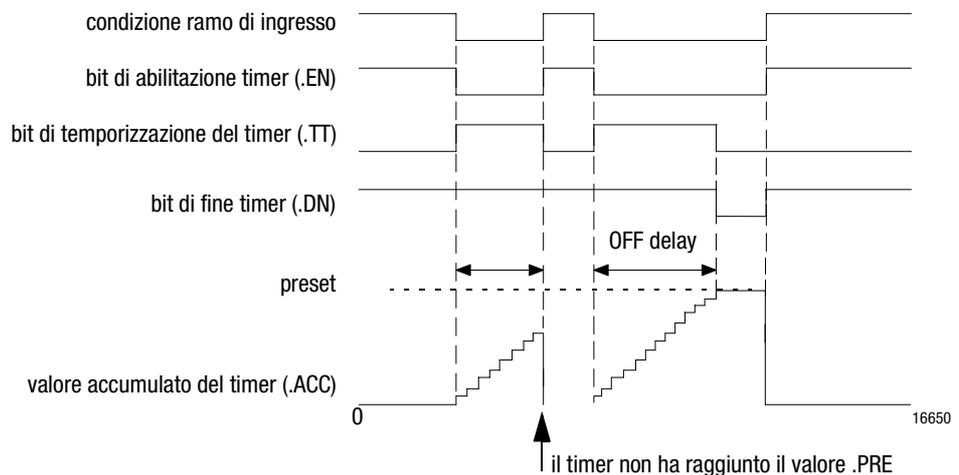
Mnemonico:	Tipo dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione TOF è abilitata.
.TT	BOOL	Il bit di temporizzazione indica che è in corso un'operazione di temporizzazione
.DN	BOOL	Il bit di fine viene azzerato quando $.ACC \geq .PRE$ .
.PRE	DINT	Il valore preimpostato specifica il valore (unità di 1 msec.) che il valore accumulato deve raggiungere prima che l'istruzione azzeri il bit .DN.
.ACC	DINT	Il valore accumulato specifica il numero di millisecondi trascorsi dal momento dell'abilitazione dell'istruzione TOF.

**Descrizione:** L'istruzione TOF è un timer non ritentivo che misura il tempo quando l'istruzione è abilitata (la condizione del ramo di ingresso è falsa). La base tempo è sempre 1 msec. Per esempio, per un timer a 2 secondi, inserire 2000 come valore .PRE.

Se è abilitata, l'istruzione TOF calcola il tempo fin quando:

- l'istruzione TOF non viene disabilitata
- $.ACC \geq .PRE$

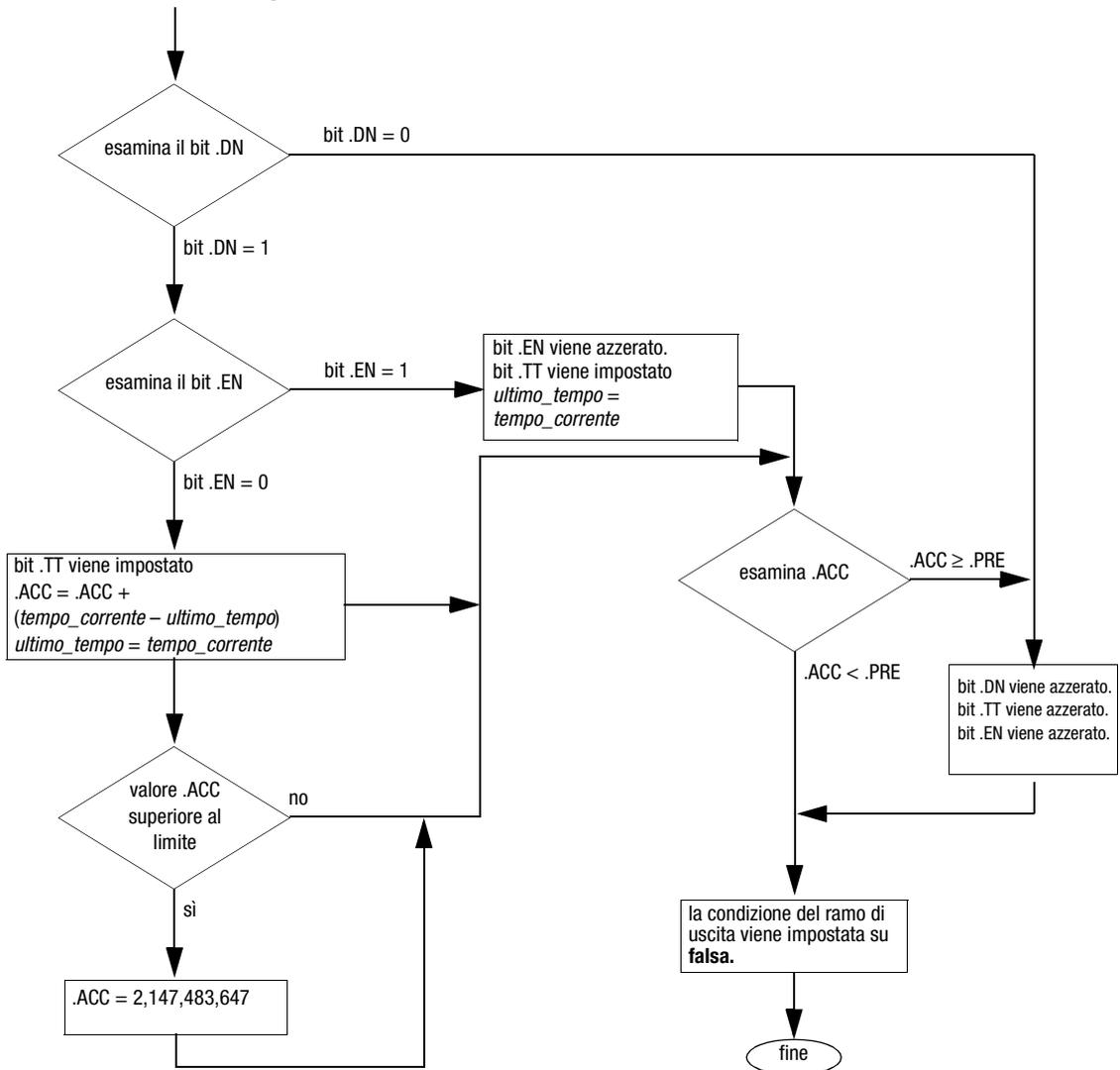
Se l'istruzione TOF è disabilitata, il valore di .ACC viene azzerato.



**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN è azzerato. Il bit .TT viene azzerato. Il bit .DN viene azzerato. Il valore .ACC viene impostato uguale al valore .PRE. La condizione del ramo di uscita viene impostata su falsa.

la condizione del ramo di ingresso è falsa

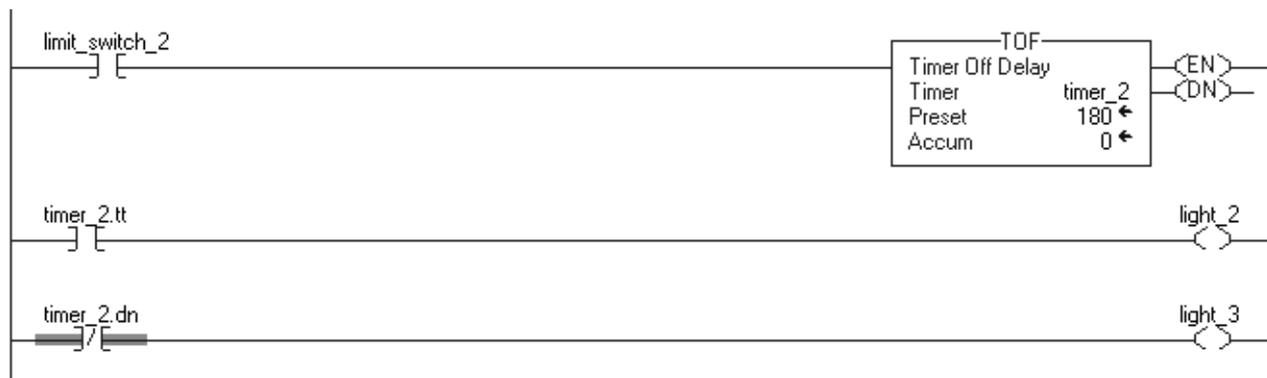


la condizione del ramo di ingresso è vera	Il bit .EN viene impostato. Il bit .TT viene azzerato. Il bit .DN viene impostato. Il valore .ACC viene azzerato. La condizione del ramo di uscita viene impostata su vero.
---	---

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
.PRE < 0	4	34
.ACC < 0	4	34

**Esempio di TOF:**

Se *limit\_switch\_2* viene azzerato, *light\_2* è accesa per 180 msec (*timer\_2* è in fase di conteggio). Se *timer\_2.acc* raggiunge 180, *light\_2* si spegne e *light\_3* si accende. La *light\_3* rimane accesa fino a quando l'istruzione TOF non viene abilitata. Se *limit\_switch\_2* è impostato mentre *timer\_2* è in fase di conteggio, *light\_2* si spegne.

**Altri formati:**

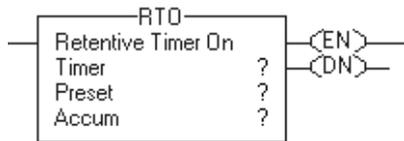
Formato:	Sintassi:
testo neutro	TOF( <i>timer</i> , <i>preset</i> , <i>accum</i> );
testo ASCII	TOF <i>timer preset accum</i>

**Istruzioni correlate:** TON, RTO

### Timer ritardato all'eccitazione ritentivo (RTO)

L'istruzione RTO è un'istruzione di uscita.

**Operandi:**



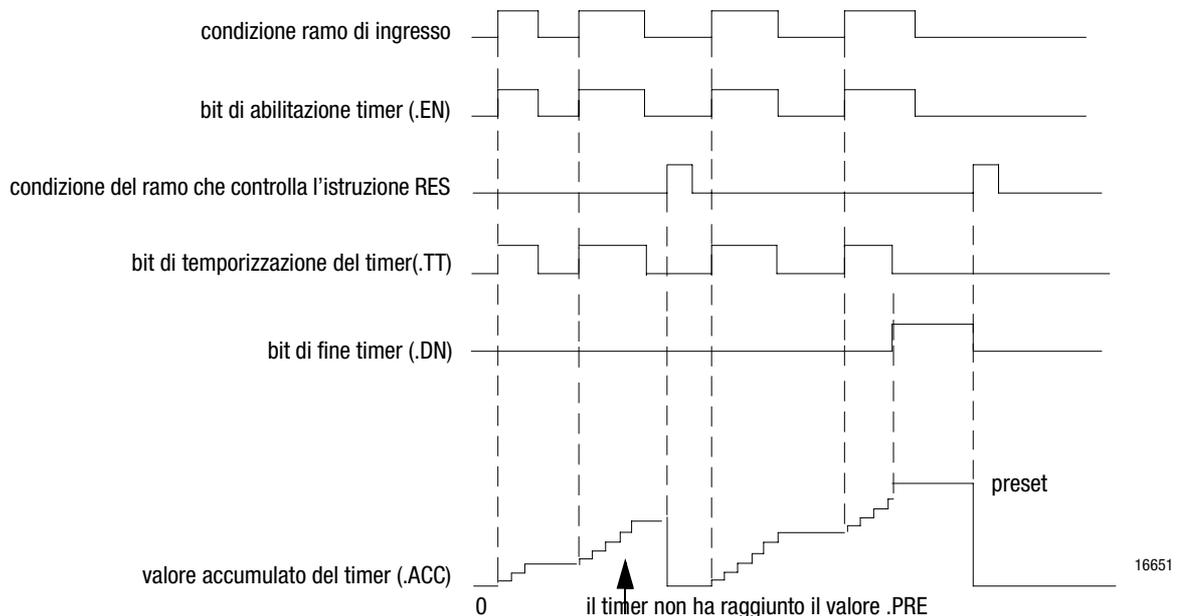
Operando:	Tipo:	Formato:	Descrizione:
Timer	TIMER	tag	struttura timer
Preset	DINT	immediato	quanto ritardare (tempo accumulato)
Accum	DINT	immediato	numero di msec misurati dal timer il valore iniziale generalmente è 0

**Struttura TIMER:**

Mnemonico:	Tipo dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione RTO è abilitata.
.TT	BOOL	Il bit di temporizzazione indica che è in corso un'operazione di temporizzazione
.DN	BOOL	Il bit di fine indica che $.ACC \geq .PRE$ .
.PRE	DINT	Il valore preimpostato specifica il valore (unità di 1 msec.) che il valore accumulato deve raggiungere prima che l'istruzione imposti il bit .DN.
.ACC	DINT	Il valore accumulato specifica il numero di millisecondi trascorsi dal momento dell'abilitazione dell'istruzione RTO.

**Descrizione:** L'istruzione RTO è un timer ritentivo che misura il tempo quando l'istruzione è abilitata. La base tempo è sempre 1 msec. Per esempio, per un timer a 2 secondi, inserire 2000 come valore .PRE.

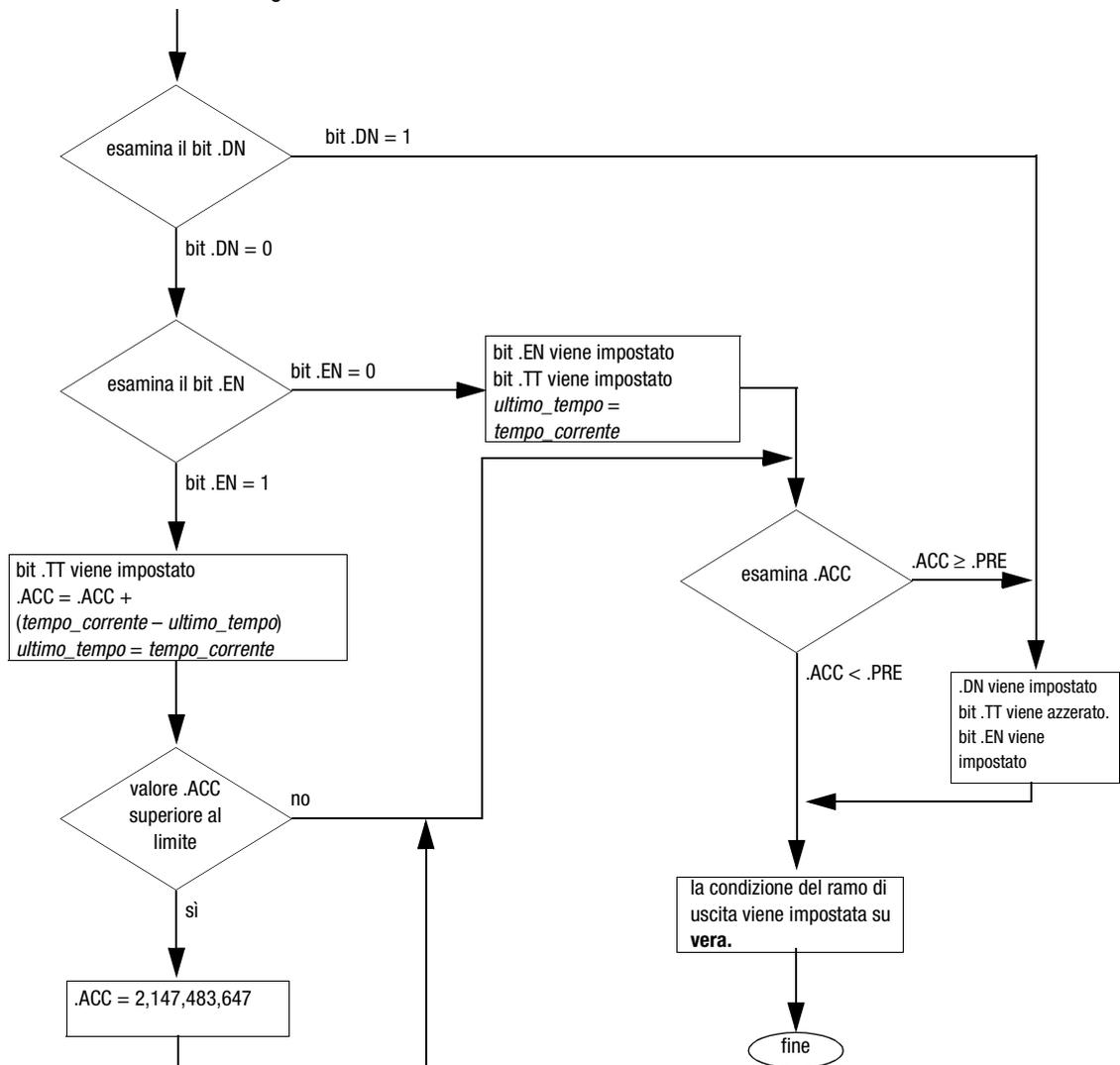
Se è abilitata, l'istruzione RTO accumula il tempo fino a quando non viene disabilitata. Se l'istruzione RTO viene disabilitata, essa conserva il suo valore .ACC. È necessario azzerare il valore .ACC, in genere mediante un'istruzione RES che faccia riferimento alla stessa struttura TIMER.



**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN è azzerato. Il bit .TT viene azzerato. Il bit .DN viene azzerato. Il valore .ACC non viene modificato. La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è falsa	Il bit .EN è azzerato. Il bit .TT viene azzerato. Il bit .DN non viene modificato. Il valore .ACC non viene modificato. La condizione del ramo di uscita viene impostata su falsa.

la condizione del ramo di ingresso è vera

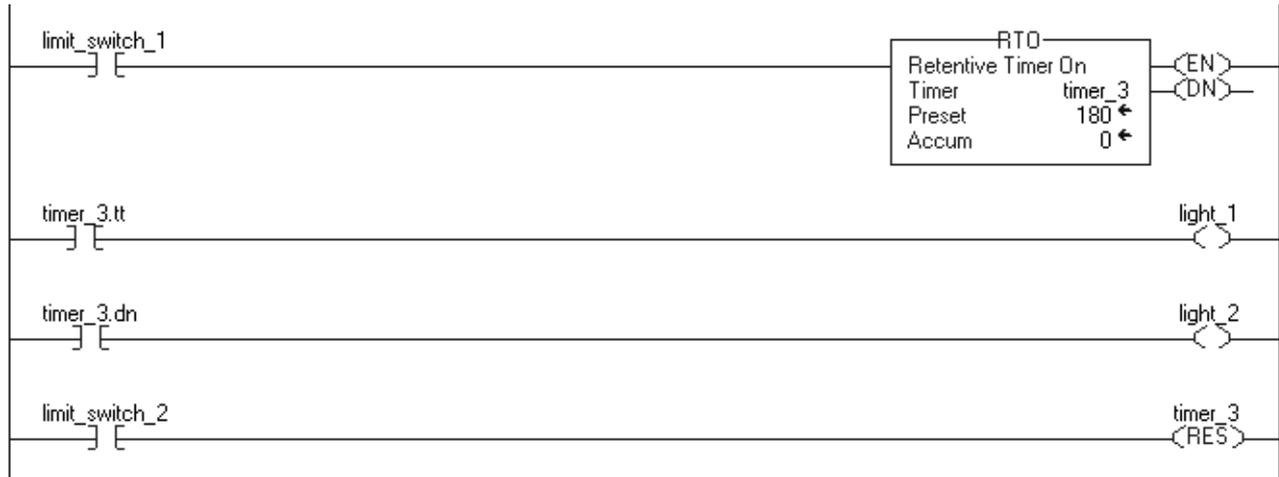


**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
.PRE < 0	4	34
.ACC < 0	4	34

**Esempio di RTO:**

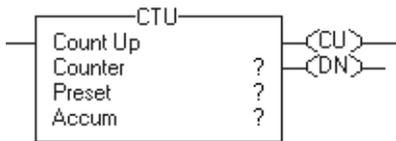


Se *limit\_switch\_1* è impostato, *light\_1* è accesa per 180 msec (*timer\_2* è in fase di conteggio). Se *timer\_3.acc* raggiunge 180, *light\_1* si spegne e *light\_2* si accende. La *light\_2* rimane accesa fino a quando *timer\_3* non viene azzerato. Se *limit\_switch\_2* viene azzerato mentre *timer\_3* è in fase di conteggio, *light\_1* rimane accesa. Se *limit\_switch\_2* è impostato, l'istruzione RES ripristina *timer\_3* (azzerata i bit di stato ed il valore .ACC).

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>RTO(timer, preset, accum);</code>
testo ASCII	<code>RTO timer preset accum</code>

**Istruzioni correlate:** TON, TOF, RES

**Conteggio incrementale (CTU)** L'istruzione CTU è un'istruzione di uscita.**Operandi:**

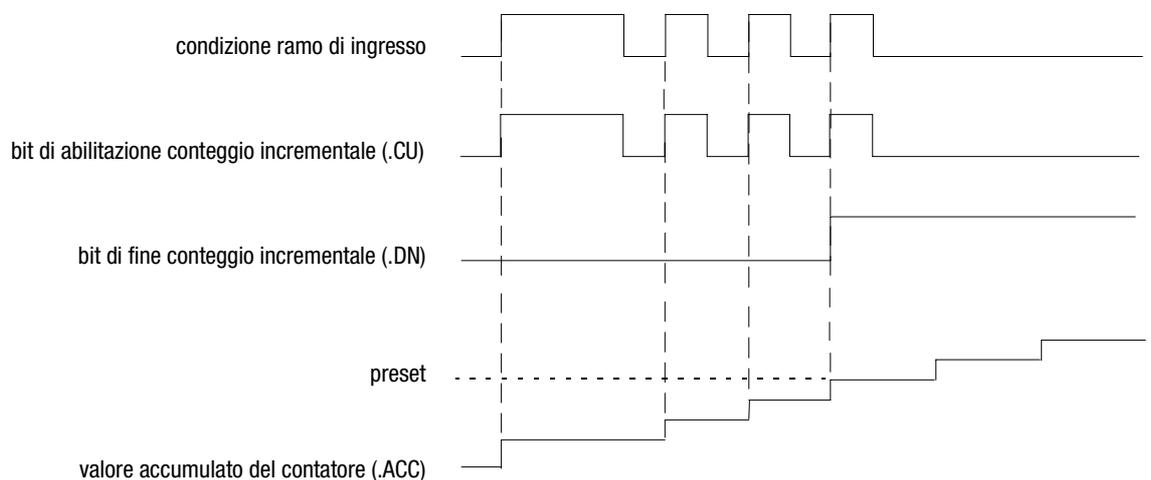
Operando:	Tipo:	Formato:	Descrizione:
Counter	COUNTER	tag	struttura contatore
Preset	DINT	immediato	limite superiore del conteggio
Accum	DINT	immediato	numero di volte calcolate dal contatore il valore iniziale generalmente è 0

**Struttura COUNTER:**

Mnemonico:	Tipo dati:	Descrizione:
.CU	BOOL	Il bit di abilitazione del conteggio incrementale indica che l'istruzione CTU è abilitata.
.DN	BOOL	Il bit di fine indica che $.ACC \geq .PRE$ .
.OV	BOOL	Il bit di overflow indica che il contatore ha superato il limite massimo di 2.147.483.647. Il contatore quindi continua a contare in modo incrementale a partire da -2.147.483.648.
.UN	BOOL	Il bit di underflow indica che il contatore ha superato il limite massimo di -2.147.483.648. Il contatore quindi continua a contare in modo decrementale a partire da 2.147.483.648.
.PRE	DINT	Il valore preimpostato specifica il valore che il valore accumulato deve raggiungere prima che l'istruzione imposti il bit .DN.
.ACC	DINT	Il valore accumulato specifica il numero di transizioni che l'istruzione ha contato.

**Descrizione:** L'istruzione CTU conteggia in modo incrementale.

Se è abilitata ed il bit .CU è azzerato, l'istruzione CTU fa incrementare il contatore di una unità. Se è abilitata e il bit .CU è impostato, oppure se è disabilitata, l'istruzione CTU conserva il valore .ACC.



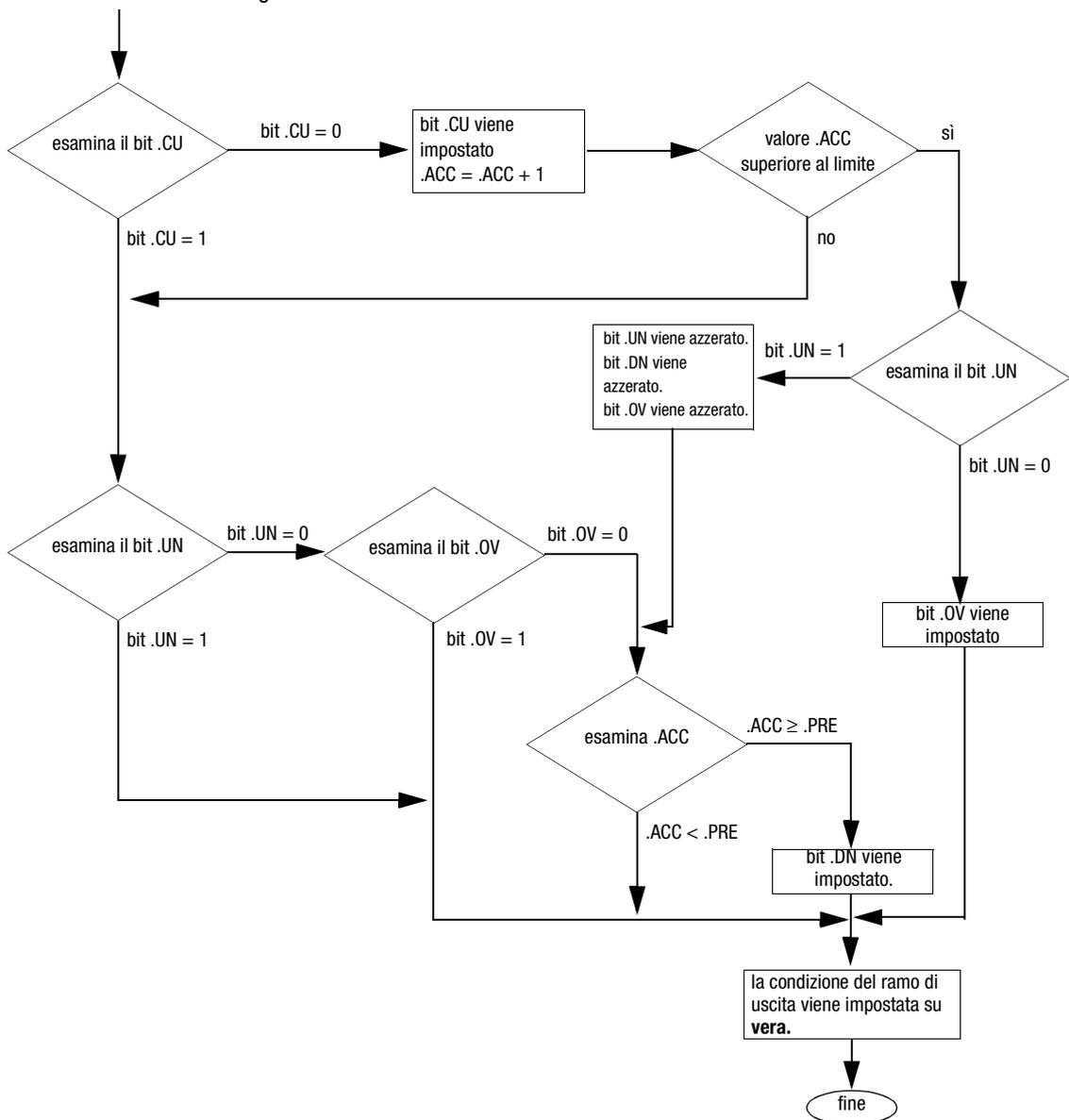
16636

Il valore accumulato continua ad aumentare, anche dopo che il bit .DN è stato impostato. Per azzerare il valore accumulato, utilizzare un'istruzione RES che faccia riferimento alla struttura del contatore o che scriva 0 nel valore accumulato.

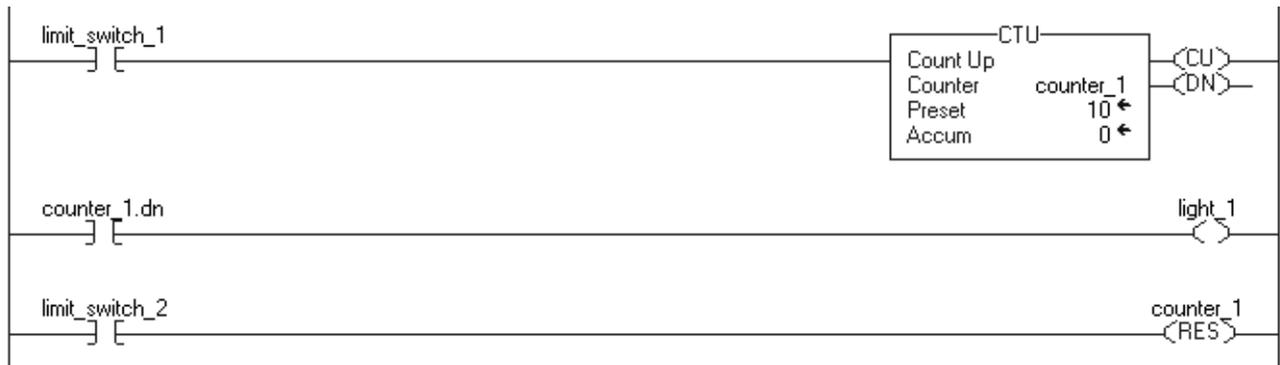
**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .CU viene impostato per evitare incrementi non validi durante la prima scansione del programma. La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è falsa	Il bit .CU viene azzerato. La condizione del ramo di uscita viene impostata su falsa.

la condizione del ramo di ingresso è vera



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna**Esempio di CTU:**

Dopo che *limit\_switch\_1* è passato da disabilitato ad abilitato per 10 volte, viene impostato il bit .DN e *light\_1* si accende. Se *limit\_switch\_1* continua a passare da disabilitato ad abilitato, *counter\_1* continua ad aumentare il suo conteggio ed il bit .DN rimane impostato. Se *limit\_switch\_2* è abilitato, l'istruzione RES ripristina *counter\_1* (azzerà i bit di stato ed il valore .ACC) e *light\_1* si spegne.

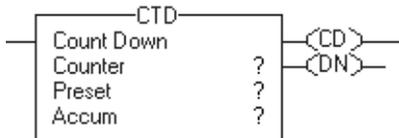
**Altri formati:**

Formato:	Sintassi:
testo neutro	CTU( <i>counter</i> , <i>preset</i> , <i>accum</i> ) ;
testo ASCII	CTU <i>contatore preset accum</i>

**Istruzioni correlate:** CTD, RES

**Conteggio decrementale (CTU)** L'istruzione CTD è un'istruzione di uscita.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
Counter	COUNTER	tag	struttura contatore
Preset	DINT	immediato	limite inferiore del conteggio
Accum	DINT	immediato	numero di volte calcolate dal contatore il valore iniziale generalmente è 0

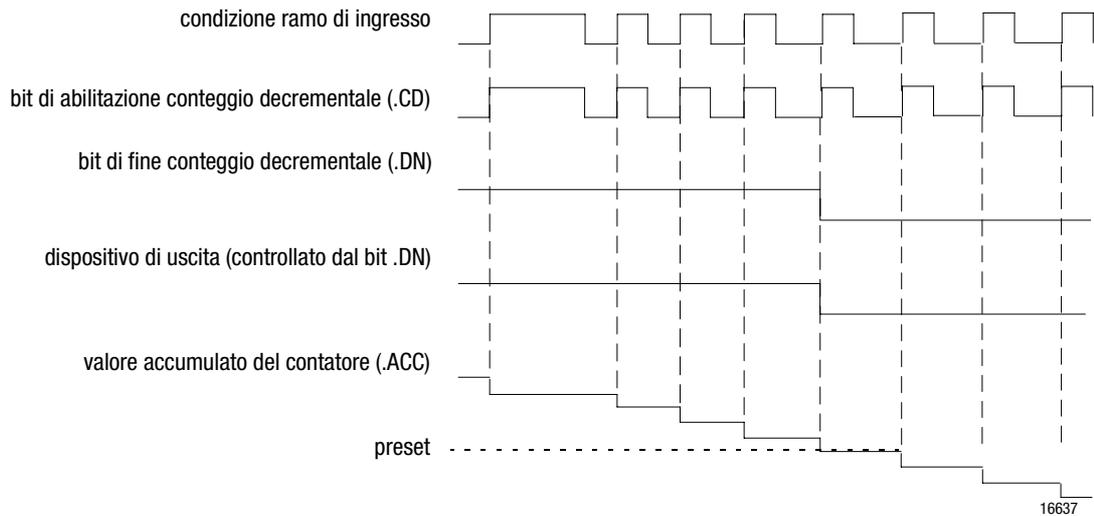
**Struttura COUNTER:**

Mnemonico:	Tipo dati:	Descrizione:
.CD	BOOL	Il bit di abilitazione del conteggio decrementale indica che l'istruzione CTD è abilitata.
.DN	BOOL	Il bit di fine indica che $.ACC \geq .PRE$ .
.OV	BOOL	Il bit di overflow indica che il contatore ha superato il limite massimo di 2.147.483.647. Il contatore quindi continua a contare in modo incrementale a partire da -2.147.483.648.
.UN	BOOL	Il bit di underflow indica che il contatore ha superato il limite massimo di -2.147.483.648. Il contatore quindi continua a contare in modo decrementale a partire da 2.147.483.648.
.PRE	DINT	Il valore preimpostato specifica il valore che il valore accumulato deve raggiungere prima che l'istruzione imposti il bit .DN.
.ACC	DINT	Il valore accumulato specifica il numero di transizioni che l'istruzione ha contato.

**Descrizione:** L'istruzione CTD conta in modo decrescente .

L'istruzione CTD generalmente viene utilizzata con un'istruzione CTU facente riferimento alla stessa struttura di contatore.

Se è abilitata ed il bit .CD viene azzerato, l'istruzione CTD fa decrescere il contatore di una unità. Se è abilitata e il bit .CD è impostato, oppure se è disabilitata, l'istruzione CTD conserva il valore .ACC.

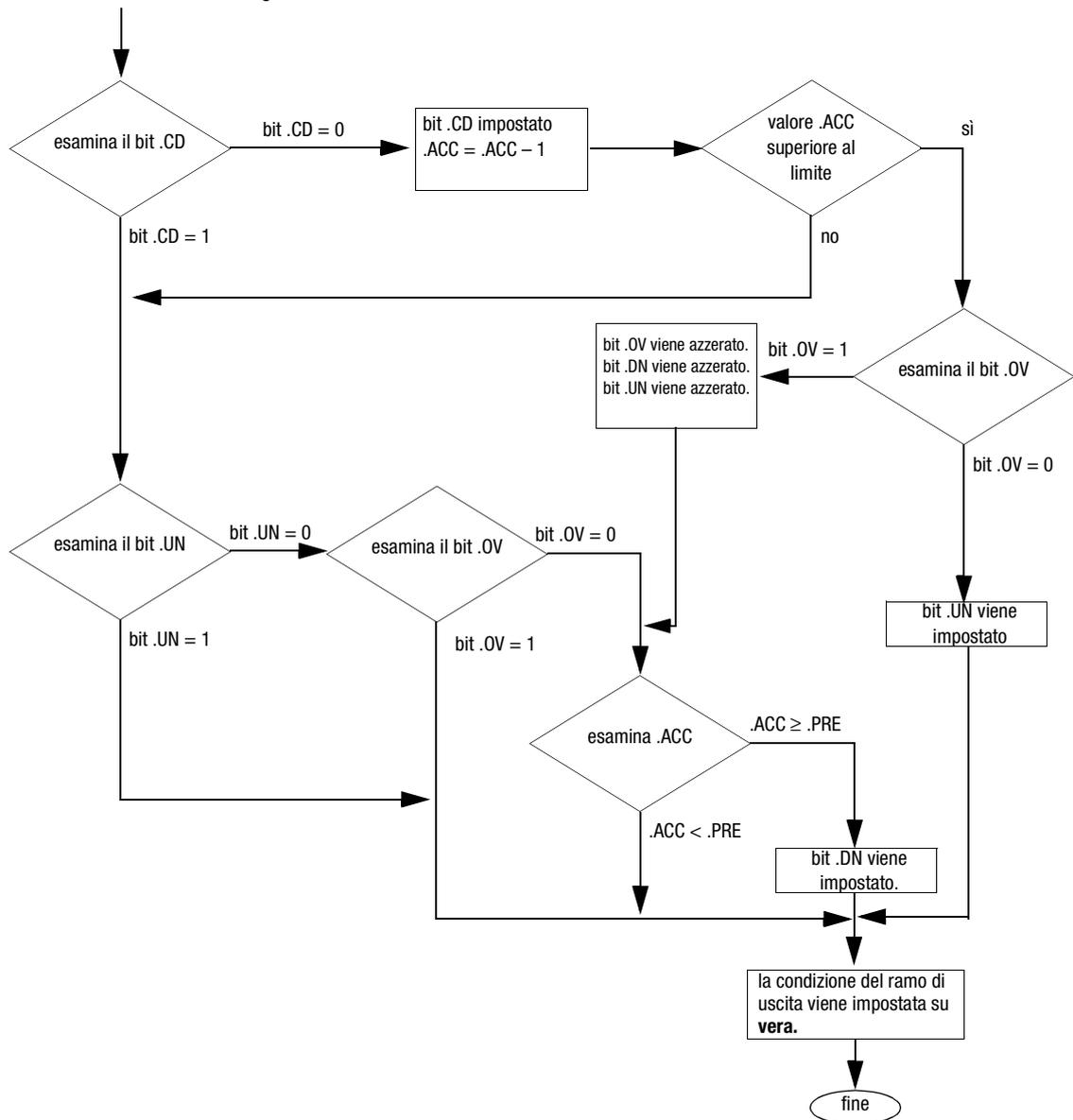


Il valore accumulato continua a decrescere, anche dopo che il bit .DN è stato impostato. Per azzerare il valore accumulato, utilizzare un'istruzione RES che faccia riferimento alla struttura del contatore o che scriva 0 nel valore accumulato.

**Esecuzione:**

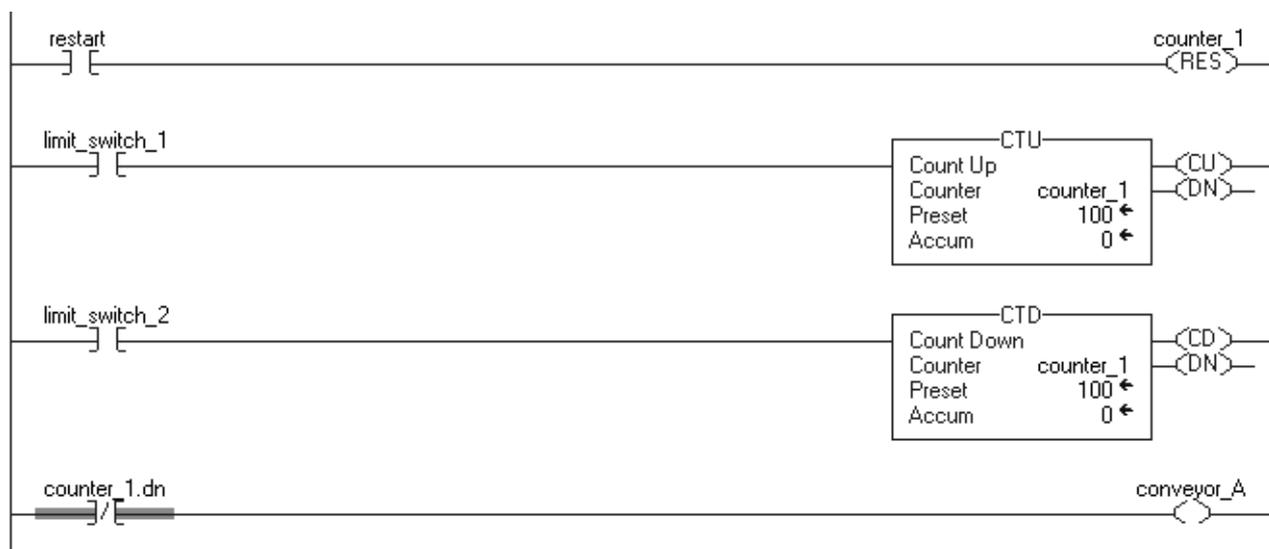
Condizione:	Azione:
prescansione	Il bit .CD è impostato per evitare decrementi non validi durante la prima scansione del programma. La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è falsa	Il bit .CD viene azzerato. La condizione del ramo di uscita viene impostata su falsa.

la condizione del ramo di ingresso è vera



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di CTD:**

Un nastro trasportatore porta dei pezzi in una zona di stoccaggio. Ogni volta che viene immesso un pezzo, *limit\_switch\_1* viene abilitato ed *counter\_1* incrementa di 1. Ogni volta che un pezzo esce dal trasportatore, *limit\_switch\_2* viene abilitato ed *contatore\_1* decrementa di 1. Se nell'area di accumulo vi sono 100 pezzi (*contatore\_1.dn* è impostato), il *nastro\_a* viene attivato e impedisce al nastro trasportatore di ricevere ulteriori pezzi finché nell'area di accumulo non si è creato ulteriore spazio per altri pezzi.

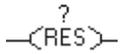
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>CTD( counter , preset , accum ) ;</code>
testo ASCII	<code>CTD counter preset accum</code>

**Istruzioni correlate:** CTU, RES

**Ripristino (RES)**

L'istruzione RES è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
structure	TIMER CONTROL COUNTER	tag	struttura da ripristinare

**Descrizione:** L'istruzione RES ripristina (azzerà) una struttura TIMER, COUNTER o CONTROL.

Se è abilitata, l'istruzione RES azzerà questi elementi:

**Se si usa un'istruzione RES per un:****L'istruzione azzerà:**

TIMER	il valore .ACC i bit di stato di controllo
COUNTER	il valore .ACC i bit di stato di controllo
CONTROL	il valore .POS i bit di stato di controllo

**ATTENZIONE:** poiché l'istruzione RES azzerà il valore .ACC, il bit .DN ed il bit .TT, non utilizzate l'istruzione RES per ripristinare un timer TOF.

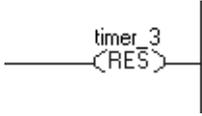
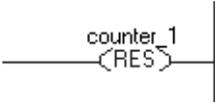
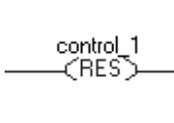
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è vera	L'istruzione RES azzerà la struttura specificata. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di RES:**

<b>Esempio:</b>	<b>Descrizione:</b>
	Se abilitata, azzera <i>timer_3</i> .
	Se abilitata, azzera <i>counter_1</i> .
	Se abilitata, azzera <i>control_1</i> .

**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<code>RES( <i>structure</i> );</code>
testo ASCII	<code>RES( <i>structure</i> );</code>

**Nota:**

## **Istruzioni di Input/Output (MSG, GSV, SSV)**

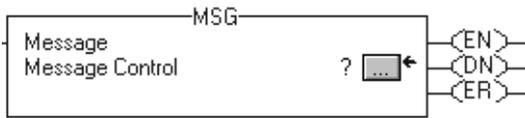
### **Introduzione**

Le istruzioni di input/output leggono o scrivono dati al/dal controllore o blocchi di dati a/da un altro modulo su un'altra rete.

<b>Se si desidera:</b>	<b>Utilizzare questa istruzione:</b>	<b>Vedere pagina:</b>
inviare/ricevere dati a/da un altro modulo	MSG	3-2
ottenere informazioni sullo stato del controllore	GSV	3-27
impostare informazioni sullo stato del controllore	SSV	3-27

## Messaggio (MSG)

L'istruzione MSG è un'istruzione di uscita.



### Operandi:

Operando:	Tipo:	Formato:	Descrizione:
Message control	Message	tag	struttura messaggio

### Struttura MSG:

Mnemonico:	Tipo di dati:	Descrizione:
.FLAGS	INT	Il membro .FLAGS permette di accedere ai membri di stato (bit) in una parola a 16 bit. <b>Quest bit: È questo membro:</b>
		2     .EW
		4     .ER
		5     .DN
		6     .ST
		7     .EN
		8     .TO
		9     .EN_CC
		<b>Importante:</b> l'azzeramento dei bit di stato MSG con un MSG abilitato può interrompere la comunicazione.
.ERR	INT	Se il bit .ER è impostato, la parola codice di errore segnala i codici di errore dell'istruzione MSG.
.EXERR	INT	La parola codice di errore esteso segnala informazioni aggiuntive per alcuni codici di errore.
.REQ_LEN	INT	La lunghezza richiesta indica quante parole l'istruzione di messaggio tenterà di trasferire.
.DN_LEN	INT	La lunghezza completa indica quante parole sono state effettivamente trasferite.
.EW	BOOL	Il bit di abilitazione attesa viene impostato quando il controllore rileva che è stata messa in coda una richiesta di messaggio. Il controllore azzerato il bit .EW quando il viene impostato il bit .ST.
.ER	BOOL	Il bit di errore viene impostato quando il controllore rileva un trasferimento errato. Il bit .ER viene azzerato la volta successiva che la condizione del ramo di ingresso passa da falsa a vera.
.DN	BOOL	Il bit di fine viene impostato quando l'ultimo pacchetto del messaggio viene trasferito senza errori. Il bit .DN viene azzerato la volta successiva che la condizione del ramo di ingresso passa da falsa a vera.
.ST	BOOL	Il bit di inizio viene impostato quando il controllore inizia ad eseguire l'istruzione MSG. Il bit .ST viene azzerato quando viene impostato il bit .DN o il bit .ER.

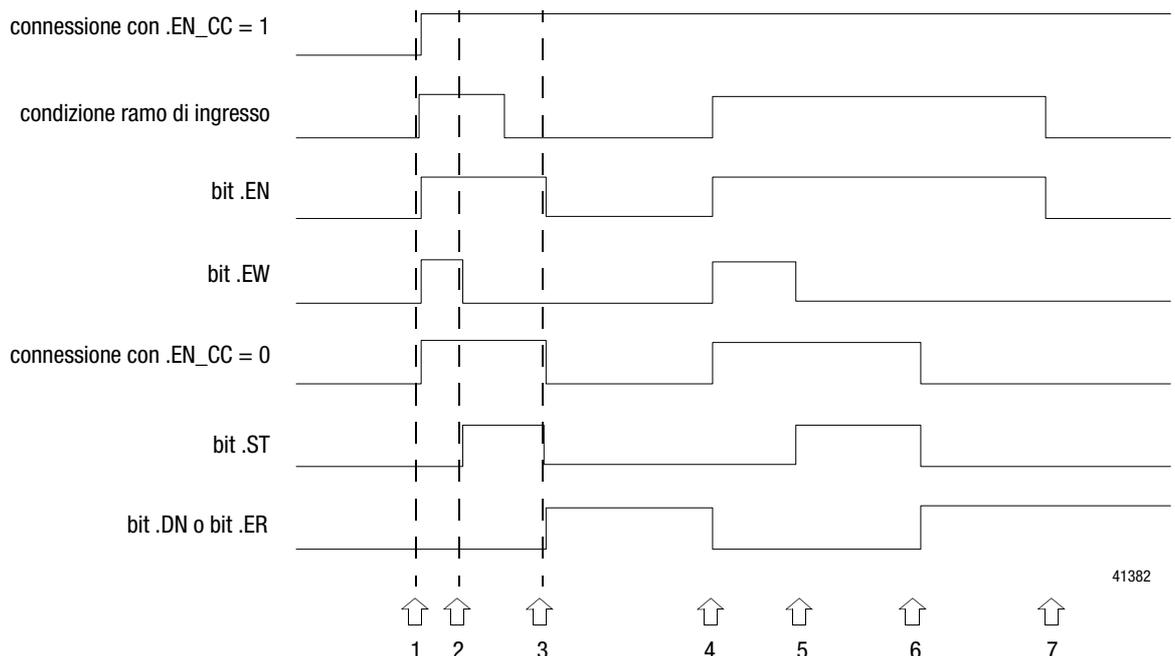
Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione viene impostato quando la condizione del ramo di ingresso diventa vera e rimane impostato fino a quando non viene impostato il bit .DN o il bit .ER e la condizione del ramo di ingresso è falsa. Se la condizione del ramo di ingresso diventa falsa ma il bit .DN ed il bit .ER sono azzerati, il bit .EN rimane impostato.
.TO	BOOL	Se si imposta manualmente il bit .TO, il controllore smette di elaborare il messaggio ed imposta il bit .ER.
.EN_CC	BOOL	Il bit di abilitazione cache stabilisce come gestire la connessione MSG. Fare riferimento a "Selezione di un'opzione con cache:" a pag. 3-26. Le connessioni delle istruzioni MSG in uscita dalla porta seriale sono senza cache, anchese il bit .EN_CC è impostato.

**ATTENTION**

**ATTENZIONE:** Il controllore elabora i bit .ST ed .EW in modo asincrono rispetto alla scansione del programma. Per esaminare questi bit nella logica ladder, copiare la parola .FLAGS in una tag INT e controllare i bit da lì. In caso contrario, problemi di temporizzazione potrebbero danneggiare l'applicazione con possibili danni alle apparecchiature o alle persone.

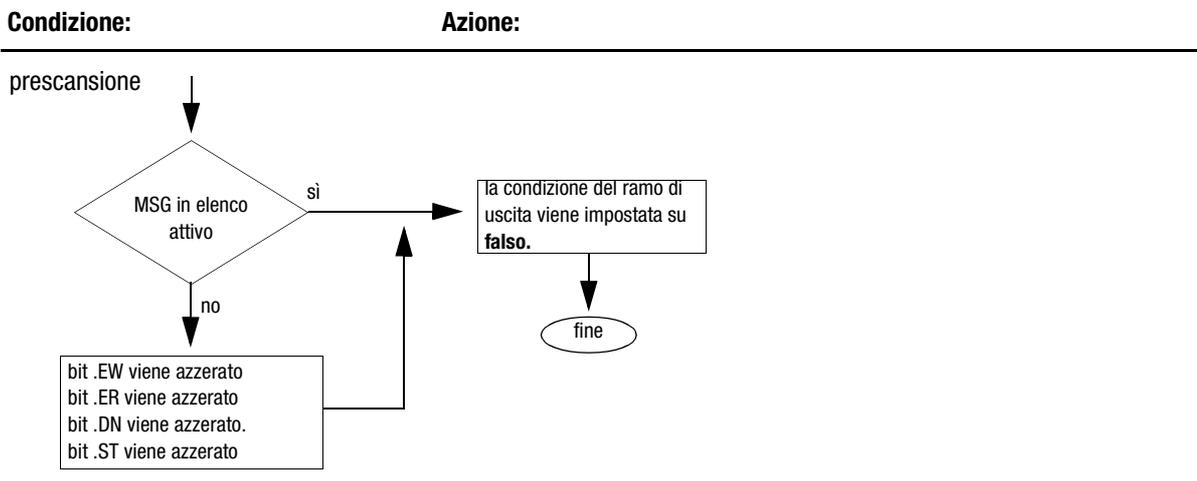
**Descrizione:** L'istruzione MSG legge o scrive un blocco di dati ad un altro modulo di una rete in modo asincrono.

L'istruzione MSG trasferisce elementi di dati. La dimensione di ciascun elemento dipende dal tipo di dati specificato e dal tipo di comando di messaggio utilizzato.

**Diagramma temporale dell'istruzione MSG**

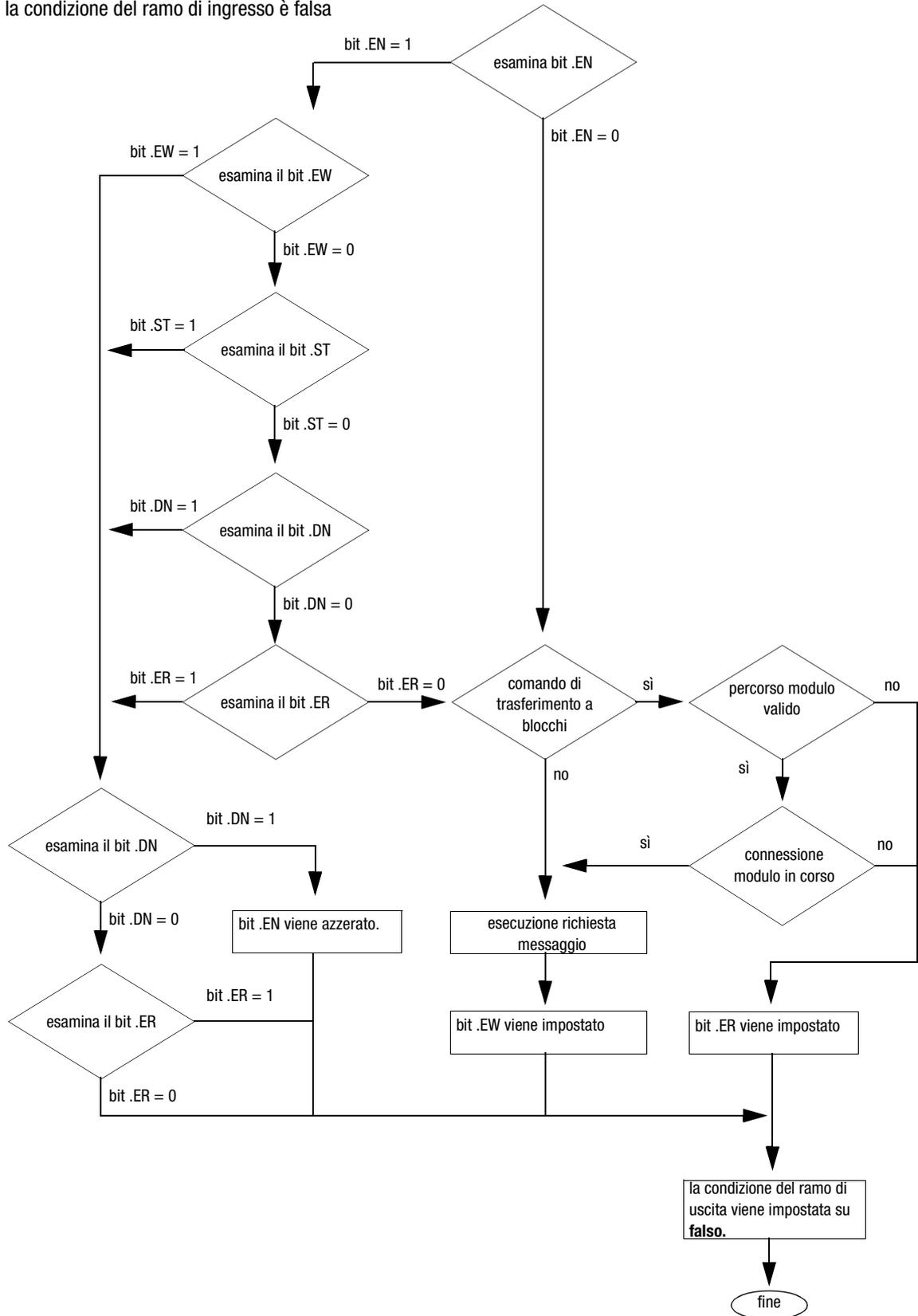
Dove:	Descrizione:
1	La condizione del ramo di ingresso è vera .EN è impostato .EW è impostato la connessione è aperta
2	il messaggio è inviato .ST è impostato .EW è azzerato
3	il messaggio è completato o in errore e la condizione del ramo di ingresso è falsa .DN o .ER impostato .ST è azzerato la connessione è chiusa (se .EN_CC = 0) .EN è azzerato (perchè la condizione del ramo di ingresso è falsa)
4	la condizione del ramo di ingresso è vera e .DN o .ER erano precedentemente impostati .EN è impostato .EW è impostato la connessione è aperta .DN o .ER è azzerato
5	il messaggio è inviato .ST è impostato .EW è azzerato
6	il messaggio è completato o in errore e la condizione del ramo di ingresso è ancora vera .DN o .ER impostato .ST è azzerato la connessione è chiusa (se .EN_CC = 0)
7	la condizione del ramo di ingresso diventa falsa e .DN o .ER è impostato .EN è azzerato.

**Esecuzione:**



**Condizione:****Azione:**

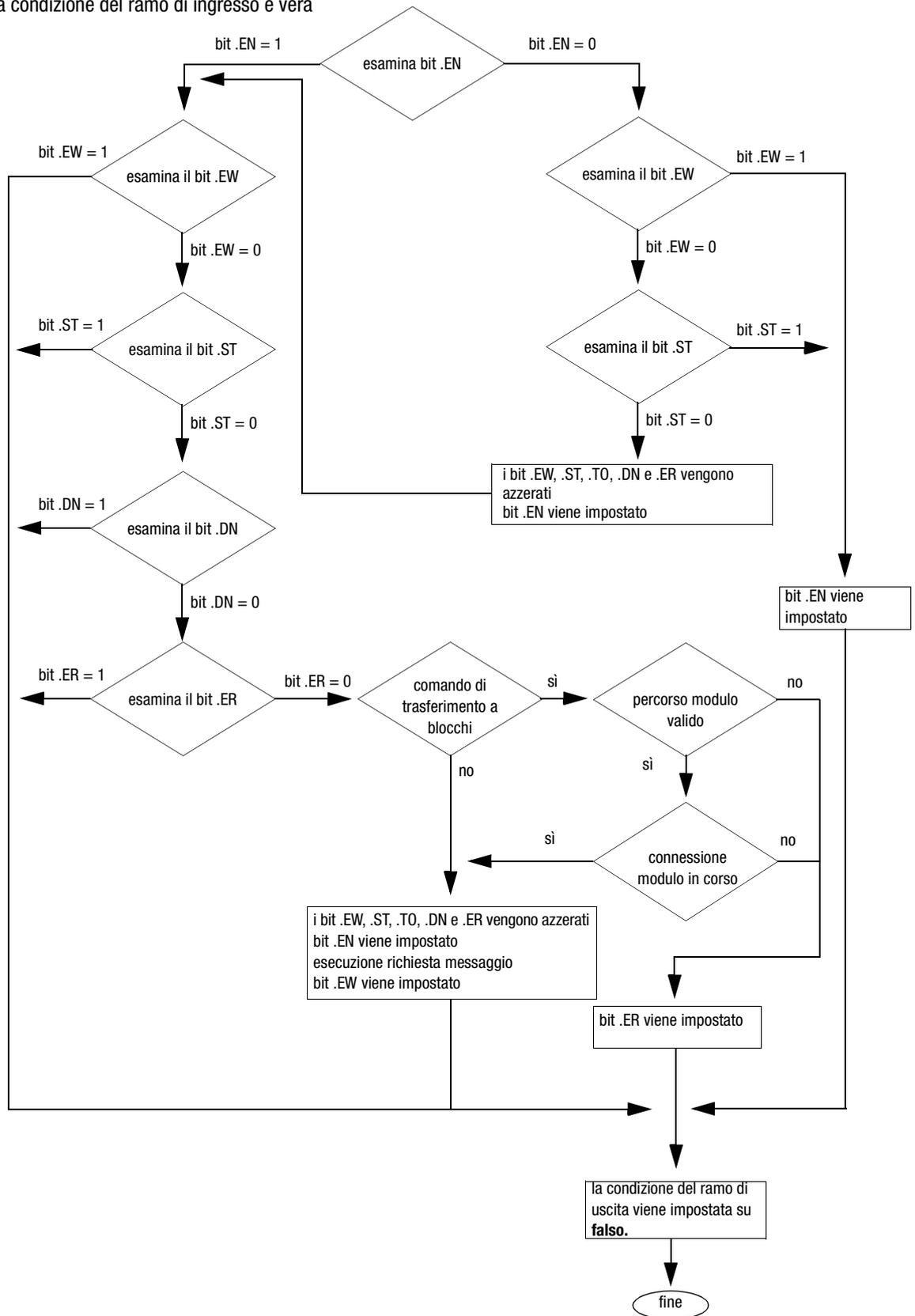
la condizione del ramo di ingresso è falsa



**Condizione:**

**Azione:**

La condizione del ramo di ingresso è vera



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>MSG(message_control);</code>
testo ASCII	<code>MSG message_control</code>

## Codici di errore MSG

I codici di errore dipendono dal tipo di istruzione MSG.

### Codici di errore ControlLogix (CIP)

Per i codici di errore ControlLogix (CIP), il software di programmazione non visualizza sempre l'intera descrizione.

Codice di errore (Esad.):	Descrizione:	Display del software:
0001	Collegamento interrotto (vedere codici di errore estesi)	come nella descrizione
0002	Risorse insufficienti	come nella descrizione
0003	Valore non valido	come nella descrizione
0004	Errore di sintassi IOI (vedere codici di errore estesi)	come nella descrizione
0005	Destinazione sconosciuta, classe non supportata, istanza non definita o elemento struttura non definito (vedere codici di errore estesi)	come nella descrizione
0006	Spazio pacchetto insufficiente	come nella descrizione
0007	Connessione interrotta	come nella descrizione
0008	Servizio non supportato	come nella descrizione
0009	Errore nel segmento dati o valore attributo non valido	come nella descrizione
000A	Errore elenco attributi	come nella descrizione
000B	Stato già esistente	come nella descrizione
000C	Conflitto modello oggetto	come nella descrizione
000D	Oggetto già esistente	come nella descrizione
000E	Impossibile impostare l'attributo	come nella descrizione
000F	Permesso negato	come nella descrizione
0010	Conflitto di stato dispositivo	come nella descrizione
0011	Risposta non adatta	come nella descrizione
0012	Frammento primitivo	come nella descrizione
0013	Dati di comando insufficienti	come nella descrizione
0014	Attributo non supportato	come nella descrizione

<b>Codice di errore (Esad.):</b>	<b>Descrizione:</b>	<b>Display del software:</b>
0015	Troppi dati	come nella descrizione
001A	Richiesta ponte troppo grande	come nella descrizione
001B	Risposta ponte troppo grande	come nella descrizione
001C	Elenco attributi vuoto	come nella descrizione
001D	Elenco attributi non valido	come nella descrizione
001E	Errore di servizio integrato	come nella descrizione
001F	Errore di collegamento (vedere codici di errore estesi)	come nella descrizione
0022	È stata ricevuta una risposta non valida	come nella descrizione
0025	Errore segmento chiave	come nella descrizione
0026	Errore di IOI non valido	come nella descrizione
0027	Attributo non previsto in elenco	come nella descrizione
0028	Errore DeviceNet – ID membro non valida	come nella descrizione
0029	Errore DeviceNet – impossibile impostare il membro	come nella descrizione

### Codici di errore estesi ControlLogix

Questi che seguono sono i codici di errore estesi di ControlLogix (CIP). Il software non visualizza testo per i codici di errore estesi. Questi sono i codici di errore estesi per il codice d'errore **0001**.

<b>Codice d'errore estesi (Esad.):</b>	<b>Descrizione:</b>	<b>Codice d'errore estesi (Esad.):</b>	<b>Descrizione:</b>
0100	Collegamento in uso	0203	Timeout connessione
0103	Trasporto non supportato	0204	Timeout di messaggio non collegato
0106	Conflitto di proprietà	0205	Errore parametro di invio non collegato
0107	Collegamento non trovato	0206	Messaggio troppo grande
0108	Tipo di collegamento non valido	0301	Memoria buffer insufficiente
0109	Dimensione collegamento non valida	0302	Larghezza di banda non disponibile
0110	Modulo non configurato	0303	Nessun filtro disponibile
0111	EPR non supportato	0305	Corrisponenza firma
0114	Modulo errato	0311	Porta non disponibile
0115	Tipo di dispositivo errato	0312	Indirizzo collegamento non disponibile
0116	Revisione errata	0315	Tipo di segmento non valido
0118	Formato di configurazione non valido	0317	Collegamento non pianificato
011A	Collegamenti non disponibili per l'applicazione		

Questi sono i codici di errore estesi per il codice d'errore **001F**.

<b>Codice d'errore estesi (Esad.):</b>	<b>Descrizione:</b>
0203	Timeout connessione

Questi sono i codici di errore estesi per i codici d'errore **0004** e **0005**.

<b>Codice d'errore estesi (Esad.):</b>	<b>Descrizione:</b>
0000	memoria insufficiente per lo stato esteso
0001	istanze insufficienti per lo stato esteso

### **Codici di errore di PLC e SLC (.ERR)**

Per i codici di errore dei PLC ed SLC, il software di programmazione non visualizza sempre l'intera descrizione.

<b>Codice di errore (Esad.):</b>	<b>Descrizione:</b>	<b>Display del software:</b>
0010	comando o formato del processore locale non valido	Conflitto di stato dispositivo
0020	Il modulo di comunicazione non funziona	Errore sconosciuto
0030	Nodo remoto mancante, scollegato o spento	Errore sconosciuto
0040	Processore collegato ma in errore (hardware)	Errore sconosciuto
0050	Numero di stazione errato	Errore sconosciuto
0060	La funzione richiesta non è disponibile	Errore sconosciuto
0070	Processore in modalità Programmazione	Errore sconosciuto
0080	Il file di compatibilità del processore non esiste	Errore sconosciuto
0090	Il nodo remoto non può inserire in memoria il comando	Errore sconosciuto
00B0	Il processore sta eseguendo uno scaricamento e non è accessibile	Errore sconosciuto
00F0	Errore PCCC (vedere codici di errore estesi)	Errore sconosciuto

### Codici di errore estesi di PLC e SLC (.EXERR)

Il software non visualizza testo per i codici di errore estesi. Questi sono i codici di errore estesi per il codice d'errore **00F0**.

Codice d'errore estesi (Esad.):	Descrizione:	Codice d'errore estesi (Esad.):	Descrizione:
0001	Il processore ha convertito l'indirizzo in modo errato	0010	Nessun accesso
0002	Indirizzo incompleto	0011	Il data type (tipo di dati) richiesto non corrisponde ai dati disponibili
0003	Indirizzo errato	0012	Parametri di comando non corretti
0004	Formato indirizzo non valido – simbolo non trovato	0013	Riferimento di indirizzo presente nell'area cancellata
0005	Formato indirizzo non valido – il simbolo non riporta alcun carattere o riporta un numero di caratteri superiore a quello massimo supportato dal dispositivo	0014	Errore di esecuzione del comando per cause sconosciute Overflow istogramma del PLC-3
0006	Il file di indirizzo non esiste nel processore di destinazione	0015	Errore di conversione dati
0007	Il file di destinazione è troppo piccolo per il numero di parole richiesto	0016	Lo scanner non è disponibile per comunicare con un adattatore rack 1771
0008	Impossibile completare la richiesta La situazione è cambiata durante il funzionamento multipacchetto	0017	L'adattatore non è disponibile per comunicare con il modulo
0009	Dati o file troppo grandi Memoria non disponibile	0018	La risposta del modulo 1771 non era valida
000A	Il processore di destinazione non può inserire le informazioni richieste in pacchetti	0019	Etichetta duplicata
000B	Errore di privilegio; accesso negato	001A	Proprietario file attivo – il file è in uso
000C	La funzione richiesta non è disponibile	001B	Proprietario programma attivo – qualcuno sta eseguendo uno scaricamento o delle modifiche online
000D	La richiesta è ridondante	001C	Il file del disco è protetto contro la scrittura o non è accessibile (solo offline)
000E	Il comando non può essere eseguito	001D	Il file del disco è utilizzato da un'altra applicazione Aggiornamento non eseguito (solo offline)
000F	Overflow; overflow istogramma		

## Codici di errore dei trasferimenti a blocchi

Questi sono i codici di errore specifici per i trasferimenti a blocchi ControlLogix.

<b>Codice di errore (Esad.):</b>	<b>Descrizione:</b>	<b>Display del software:</b>
00D0	Lo scanner non ha ricevuto una risposta di trasferimento a blocchi dal modulo a trasferimenti a blocchi entro 3,5 secondi dalla richiesta	Errore sconosciuto
00D1	Il checksum dalla risposta di lettura non corrisponde con il checksum del flusso dati	Errore sconosciuto
00D2	Lo scanner ha richiesto una lettura o una scrittura ma il modulo a trasferimento a blocchi ha risposto con l'opposto	Errore sconosciuto
00D3	Lo scanner ha richiesto una lunghezza ed il modulo a trasferimento a blocchi ha risposto con una lunghezza diversa	Errore sconosciuto
00D6	Lo scanner ha ricevuto una risposta dal modulo a trasferimento a blocchi indicante un errore nella richiesta di scrittura	Errore sconosciuto
00EA	Lo scanner non è stato configurato per comunicare con il rack contenente questo modulo a trasferimento a blocchi	Errore sconosciuto
00EB	Lo slot logico specificato non è disponibile per la dimensione del rack	Errore sconosciuto
00EC	Esiste una richiesta di trasferimento a blocchi in corso che richiede una risposta prima che possa avere inizio un'altra richiesta	Errore sconosciuto
00ED	La dimensione della richiesta di trasferimento a blocchi non è compatibile con le richieste di trasferimenti a blocchi di dimensioni valide	Errore sconosciuto
00EE	Il tipo di richiesta di trasferimento a blocchi non è compatibile con il previsto BT_READ o BT_WRITE	Errore sconosciuto
00EF	Lo scanner non è riuscito a trovare nella tabella dei trasferimenti a blocchi uno slot disponibile per la richiesta di trasferimento a blocchi	Errore sconosciuto
00F0	Lo scanner ha ricevuto la richiesta di ripristinare i canali I/O remoti con trasferimenti a blocchi in sospenso	Errore sconosciuto
00F3	Le code per i trasferimenti a blocchi remoti sono piene	Errore sconosciuto
00F5	Nessun canale di comunicazione configurato per il rack o lo slot richiesto	Errore sconosciuto
00F6	Nessun canale di comunicazione configurato per I/O remoto	Errore sconosciuto
00F7	Il timeout del trasferimento a blocchi, impostato nell'istruzione, è scaduto prima del completamento	Errore sconosciuto
00F8	Errore nel protocollo del trasferimento a blocchi – trasferimento a blocchi non richiesto	Errore sconosciuto
00F9	I dati del trasferimento a blocchi sono andati persi a causa di un canale di comunicazione guasto	Errore sconosciuto
00FA	Il modulo a trasferimento a blocchi ha richiesto una lunghezza diversa da quella associata all'istruzione di trasferimento a blocchi	Errore sconosciuto
00FB	Il checksum dei dati di trasferimento a blocchi di lettura era errato	Errore sconosciuto
00FC	Trasferimento non valido dei dati di trasferimento a blocchi di scrittura tra l'adattatore ed il modulo a trasferimento a blocchi	Errore sconosciuto
00FD	La dimensione del trasferimento a blocchi più la dimensione dell'indice nella tabella dati dei trasferimenti a blocchi è maggiore della dimensione del file della tabella dati dei trasferimenti a blocchi	Errore sconosciuto

## Codici di errore del Logix5550

Questi sono i codici di errore del Logix5550.

<b>Codice di errore (Esad.):</b>	<b>Descrizione:</b>	<b>Display del software:</b>
00D0	Istanza mappa non definita	Errore sconosciuto
00D1	Il modulo non è in stato di esecuzione	Errore sconosciuto
00FB	Porta messaggi non supportata	Errore sconosciuto
00FC	Il messaggio non supporta il data type	Errore sconosciuto
00FD	Messaggio non inizializzato	Errore sconosciuto
00FE	Timeout del messaggio	Errore sconosciuto
00FF	Errore generale (vedere codici di errore estesi)	Errore sconosciuto

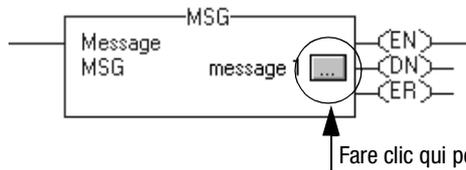
## Codici di errore estesi del Logix5550

Questi che seguono sono i codici di errore estesi del Logix5550. Il software non visualizza testo per i codici di errore estesi. Questi sono i codici di errore estesi per il codice d'errore **00FF**.

<b>Codice d'errore estesi (Esad.):</b>	<b>Descrizione:</b>	<b>Codice d'errore estesi (Esad.):</b>	<b>Descrizione:</b>
2001	IOI eccessivo	2108	Controllore in modalità di caricamento o di scaricamento
2002	Valore parametro errato	2109	Si è tentato di modificare il numero di dimensioni della matrice
2018	Rifiuto semaforo	210A	Nome simbolo non valido
201B	Dimensioni troppo piccole	210B	Il simbolo non esiste
201C	Dimensione non valida	210E	Ricerca nulla
2100	Errore di privilegio	210F	La task non può iniziare
2101	Posizione del selettore a chiave non valida	2110	Impossibile scrivere
2102	Password non valida	2111	Impossibile leggere
2103	Nessuna password configurata	2112	Routine condivisa non modificabile
2104	Indirizzo fuori gamma	2113	Controllore in modalità di errore
2105	Indirizzo e quantità fuori gamma	2114	Modalità di esecuzione inibita
2106	Dati in uso		
2107	Tipo non valido o non supportato		

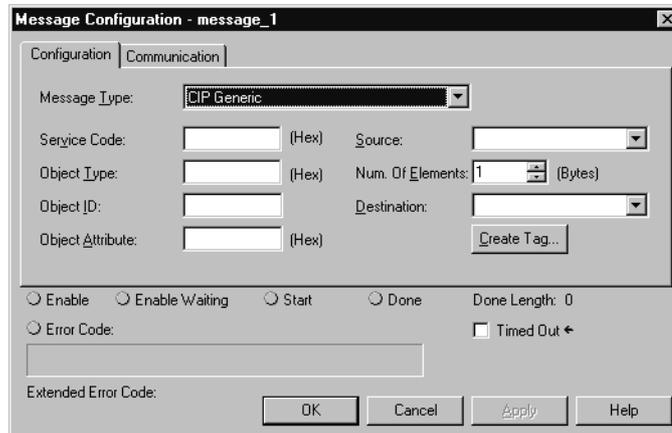
## Specificazione dei dettagli di comunicazione (Scheda Configurazione)

Dopo avere immesso l'istruzione MSG e specificato la struttura del MESSAGE, utilizzare la scheda Configurazione del software di programmazione per specificare i dettagli del messaggio.



Fare clic qui per configurare l'istruzione MSG

Le informazioni da configurare dipendono dal tipo di messaggio che si seleziona.



Se il dispositivo di destinazione è un:	Selezionare uno di questi tipi di messaggio:	Vedere pagina:
Dispositivo ControlLogix o modulo I/O 1756	Tabella dati CIP di lettura	3-14
	Tabella dati CIP di scrittura	
	CIP generico	
Processore PLC-5	Letture tipizzate per PLC5	3-16
	Scrittura tipizzata per PLC5	
	Letture a parole per PLC5	
	Scrittura a parole per PLC5	
Controllore SLC	Letture tipizzate per SLC	3-17
	Scrittura tipizzata per SLC	
Modulo a trasferimento a blocchi su una rete I/O remoto universale	Trasferimento a blocchi di lettura	3-17
	Trasferimento a blocchi di scrittura	
Processore PLC-3	Letture tipizzate per PLC3	3-18
	Scrittura tipizzata per PLC3	
	Letture a parole per PLC3	
	Scrittura a parole per PLC3	
Processore PLC-2	Letture non protetta per PLC2	3-19
	Scrittura non protetta per PLC2	

È necessario specificare queste informazioni di configurazione:

In questo campo:	Specificare:
Source Element/Tag	Se si seleziona un tipo di messaggio di lettura, l'elemento di origine è l'indirizzo dei dati che si desidera leggere nel dispositivo di destinazione. Utilizzare la sintassi di indirizzamento del dispositivo di destinazione. Se si seleziona un tipo di messaggio di scrittura, la tag di origine è la tag dei dati del controllore Logix5550 che si desidera inviare al dispositivo di destinazione.
Number of Elements	Il numero di elementi che si leggono/scrivono dipende dal tipo di dati utilizzati. Un elemento si riferisce ad un "gruppo" di dati correlati. Per esempio, la tag <i>timer1</i> è un elemento costituito da una struttura di controllo timer.
Destination Element/Tag	Se si seleziona un tipo di messaggio di lettura, la tag di destinazione è la tag del controllore Logix5550 in cui si desidera memorizzare i dati letti dal dispositivo di destinazione. Se si seleziona un tipo di messaggio di scrittura, l'elemento di destinazione è l'indirizzo del luogo del dispositivo di destinazione in cui si desidera scrivere i dati.

Se si indica una tag matrice di Logix5550 per l'origine o per la destinazione, indicare solamente il nome della tag matrice. Non includere le parentesi o il deponente di posizione.

### Specificazione di messaggi CIP

I messaggi CIP servono a trasferire dati a/da altri dispositivi ControlLogix, come ad esempio l'invio di un messaggio da un controllore Logix5550 ad un altro.

Selezionare questo comando:	Se si desidera:
Tabella dati CIP di lettura	leggere dati da un altro controllore. I tipi di Source e di Destination devono corrispondere.
Tabella dati CIP di scrittura	scrivere dati ad un altro controllore. I tipi di Source e di Destination devono corrispondere.
CIP generico	configurare un messaggio personalizzato per inviare dati di configurazione ad un modulo I/O

## Uso di messaggi CIP generici per ripristinare moduli I/O

Per creare un messaggio personalizzato utilizzando il tipo di messaggio CIP generico, specificare queste informazioni.

Se si desidera:	In questo campo:	Digitare:	
Eseguire una prova di impulso su un modulo di uscita digitale	Service Code	4c	
	Object Type	1e	
	Object ID	1	
	Object attribute	lasciare vuoto	
	Source	<i>nome_tag</i> di tipo INT [5] Questa matrice contiene:	
		<i>nome_tag</i> [0]	maschera bit dei punti da testare (testare solo un punto alla volta)
		<i>nome_tag</i> [1]	riservato, lasciare 0
		<i>nome_tag</i> [2]	ampiezza di impulso (centinaia di $\mu$ sec, generalmente 20)
		<i>nome_tag</i> [3]	ritardo dell'attraversamento dello zero per l'I/O ControlLogix (centinaia di $\mu$ sec, generalmente 40)
		<i>nome_tag</i> [4]	verifica ritardo
	Number of Elements	10	
	Destination	lasciare vuoto	
Ripristinare i fusibili elettronici di un modulo di uscita digitale	Service Code	4d	
	Object Type	1e	
	Object attribute	lasciare vuoto	
	Object ID	1	
	Source	<i>nome_tag</i> di tipo DINT Questa tag rappresenta una maschera a bit dei punti su cui ripristinare i fusibili.	
	Number of Elements	4	
	Destination	lasciare vuoto	
Ripristinare la diagnostica agganciata di un modulo I/O digitale	Service Code	4b	
	Object Type	per un modulo di ingresso 1d per un modulo di uscita 1e	
	Object attribute	lasciare vuoto	
	Object ID	1	
	Source	<i>nome_tag</i> di tipo DINT Questa tag rappresenta una maschera a bit dei punti su cui ripristinare la diagnostica.	
		Number of Elements	4
		Destination	lasciare vuoto

<b>Se si desidera:</b>	<b>In questo campo:</b>	<b>Digitare:</b>
Ripristinare lo stato agganciato di un modulo analogico	Service Code	4b
	Object Type	a
	Object attribute	inserire il codice dell'attributo richiesto
	Object ID	0
	Source	lasciare vuoto
	Number of Elements	0
	Destination	lasciare vuoto

### Specificazione di messaggi per PLC-5

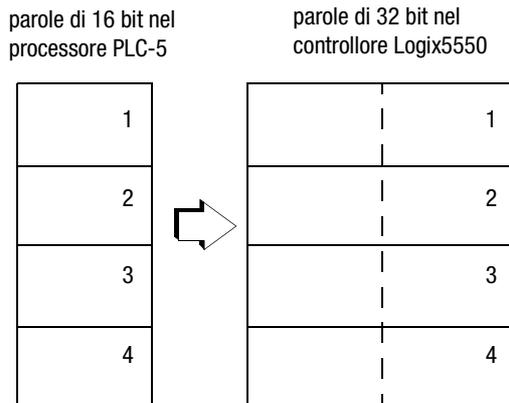
I messaggi per PLC-5 sono rivolti ai processori PLC-5.

<b>Selezionare questo comando:</b>	<b>Se si desidera:</b>
Lettura tipizzata per PLC5	leggere dati di tipo intero o REAL. Per i numeri interi, questo comando legge interi a 16 bit dal processore PLC-5 (tipi di file S, B ed N) che memorizza in matrici di dati SINT, INT o DINT del controllore Logix5550, e gestisce l'integrità dei dati. Questo comando, inoltre, legge i dati a virgola mobile dal processore PLC-5 (tipo di file F) e li memorizza in una tag di dati di tipo REAL del controllore Logix5550.
Scrittura tipizzata per PLC5	scrivere dati di tipo intero o REAL. Questo comando scrive dati SINT o INT al processore PLC-5 (tipi di file S, B ed N) e gestisce l'integrità dei dati. È possibile scrivere dati DINT purché essi rientrino nel tipo di dati INT ( $-32.768 \geq \text{dati} \leq 32.767$ ). Questo comando, inoltre, scrive dati di tipo REAL dal processore Logix5550 ad un file a virgola mobile per PLC-5 (tipo di file F).
Lettura a parole per PLC5	leggere una serie contigua di parole a 16 bit nella memoria del PLC-5 a prescindere dal tipo di dati. Questo comando inizia dall'indirizzo specificato come Elemento Source e legge in sequenza il numero delle parole a 16 bit richieste. I dati dall'Elemento Source vengono memorizzati, a partire dall'indirizzo specificato come tag di destinazione.
Scrittura a parole per PLC5	scrivere parole contigue di 16 bit dalla memoria del Logix5550 alla memoria del PLC-5, a prescindere dal tipo di dati. Questo comando inizia dall'indirizzo specificato come Tag Source e legge in sequenza il numero delle parole a 16 bit richieste. I dati dalla Tag Source vengono memorizzati a partire dall'indirizzo specificato come Elemento Destination nel processore PLC-5.

I comandi di Lettura tipizzata e di Scrittura tipizzata funzionano anche con i processori SLC 5/03 (OS303 e superiori), i processori SLC 5/04 (OS402 e superiori) e i processori SLC 5/05.

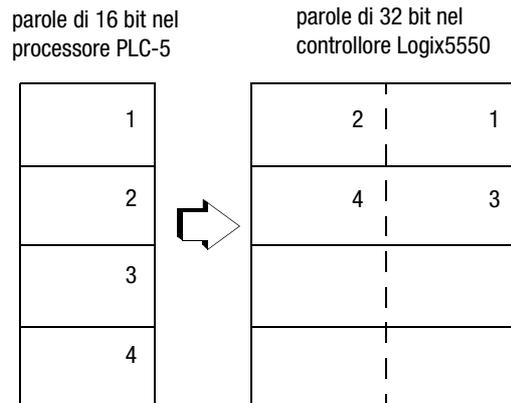
I seguenti schemi mostrano le differenze tra i comandi tipizzati e a parole. L'esempio utilizza comandi di lettura da un processore PLC-5 ad un controllore Logix5550.

#### Comando di lettura tipizzata



I comandi tipizzati mantengono la struttura ed il valore dei dati.

#### Comando di lettura a parole



I comandi a parole occupano le tag di destinazione in modo contiguo. La struttura ed il valore dei dati cambiano a seconda del tipo di dati di destinazione.

## Specificazione di messaggi per SLC

I messaggi per SLC sono rivolti agli SLC e ai controllori MicroLogix1000.

**Selezionare questo comando:**

**Se si desidera:**

Letture tipizzate per SLC leggere dati INT o DINT.

Scrittura tipizzata per SLC scrivere dati INT o DINT.

Il tipo di tag del Logix5550 deve corrispondere con il tipo di dati dell'SLC. È possibile solamente trasferire dati DINT (che indirizzano al tipo di dati a bit per SLC) o dati INT (che indirizzano al tipo di dati interi per SLC).

## Specificazione dei messaggi a trasferimento a blocchi

I messaggi a trasferimento a blocchi sono usati per comunicare con moduli a trasferimento a blocchi su una rete I/O remoto universale.

**Selezionare questo comando:**

**Se si desidera:**

Trasferimento a blocchi di lettura leggere dati da un modulo a trasferimento a blocchi. Questo tipo di messaggio sostituisce l'istruzione BTR.

Trasferimento a blocchi di scrittura scrivere dati ad un modulo a trasferimento a blocchi. Questo tipo di messaggio sostituisce l'istruzione BTW.

Le tag di origine (per i BTW) e di destinazione (per i BTR) devono essere abbastanza grandi da accettare i dati richiesti, eccetto le strutture MESSAGE, AXIS e MODULE.

È necessario inoltre specificare quanti interi a 16 bit (INT) inviare o ricevere. È possibile specificare da 0 a 64 interi. Se si specifica 0 per un messaggio BTR, il modulo a trasferimento a blocchi stabilisce quanti interi a 16 bit inviare. Se si specifica 0 per un messaggio BTW, il controllore invia 64 interi.

Il modulo I/O che deve ricevere il trasferimento a blocchi deve essere identificato nell'organizer del controllore.

Se si seleziona un messaggio di trasferimento a blocchi, non bisogna selezionare alcun metodo di comunicazione nella scheda Configurazione. Le voci CIP e DH+ sono disattivate.

### Specificazione di messaggi per PLC-3

I messaggi per PLC-3 sono rivolti ai processori PLC-3.

<b>Selezionare questo comando:</b>	<b>Se si desidera:</b>
Letture tipizzata per PLC3	leggere dati di tipo intero o REAL. Per i numeri interi, questo comando legge interi a 16 bit dal processore PLC-3 che memorizza in matrici di dati SINT, INT o DINT del controllore Logix5550, e gestisce l'integrità dei dati. Questo comando, inoltre, legge i dati a virgola mobile dal PLC-3 e li memorizza in una tag di tipo di dati REAL del controllore Logix5550.
Scrittura tipizzata per PLC3	scrivere dati di tipo intero o REAL. Questo comando scrive dati SINT o INT nel file intero del PLC-3 e gestisce l'integrità dei dati. È possibile scrivere dati DINT purché essi rientrino nel tipo di dati INT ( $-32.768 \geq \text{dati} \leq 32.767$ ). Questo comando, inoltre, scrive dati di tipo REAL dal processore Logix5550 in un file a virgola mobile di PLC-3.
Letture a parole per PLC3	leggere una serie contigua di parole a 16 bit nella memoria del PLC-3 a prescindere dal tipo di dati. Questo comando inizia dall'indirizzo specificato come Elemento Source e legge in sequenza il numero delle parole a 16 bit richieste. I dati dall'Elemento Source vengono memorizzati, a partire dall'indirizzo specificato come tag di destinazione.
Scrittura a parole per PLC3	scrivere parole contigue di 16 bit dalla memoria del Logix5550 alla memoria del PLC-3, a prescindere dal tipo di dati. Questo comando inizia dall'indirizzo specificato come Tag Source e legge in sequenza il numero delle parole a 16 bit richieste. I dati dalla Tag Source vengono memorizzati a partire dall'indirizzo specificato come Elemento di destinazione nel processore PLC-3.

I seguenti schemi mostrano le differenze tra i comandi tipizzati e a parola. L'esempio utilizza comandi di lettura da un processore PLC-3 ad un controllore Logix5550.

#### Comando di lettura tipizzata

parole di 16 bit nel processore PLC-3

1
2
3
4



parole di 32 bit nel controllore Logix5550

1
2
3
4

I comandi tipizzati mantengono la struttura ed il valore dei dati.

#### Comando di lettura a parole

parole di 16 bit nel processore PLC-3

1
2
3
4



parole di 32 bit nel controllore Logix5550

2	1
4	3

I comandi a parole occupano le tag di destinazione in modo contiguo. La struttura ed il valore dei dati cambiano a seconda del tipo di dati di destinazione.

## Specificazione di messaggi per PLC-2

I messaggi per PLC-2 sono rivolti ai processori PLC-2.

#### Selezionare questo comando:

#### Se si desidera:

Letture non protetta per PLC2

leggere parole a 16 bit da qualsiasi area della tabella dati del PLC-2 o dal file di compatibilità con il PLC-2 di un altro processore.

Scrittura non protetta per PLC2

scrivere parole a 16 bit in qualsiasi area della tabella dati del PLC-2 o dal file di compatibilità con il PLC-2 di un altro processore.

Questo trasferimento a messaggi utilizza parole a 16 bit, per cui accertatevi che la tag del Logix5550 memorizzi correttamente i dati trasferiti (in genere come una matrice INT).

**Esempi di configurazione MSG:** I seguenti esempi mostrano delle tag di origine e di destinazione ed elementi esemplificativi per varie combinazioni di controllori.

Per le istruzioni MSG che hanno origine da un controllore Logix5550 e che scrivono in un altro controllore:

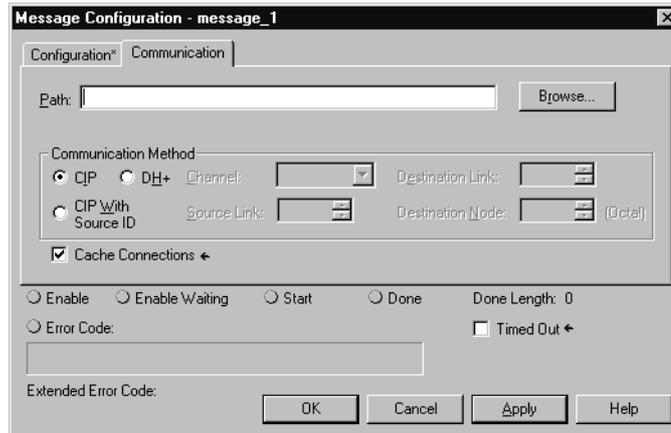
<b>Percorso del messaggio:</b>	<b>Origine e Destinazione di esempio:</b>	
Logix5550 → Logix5550	tag di origine	<i>matrice_1</i>
	tag di destinazione	<i>matrice_2</i>
	<p>È possibile utilizzare una tag alias al posto della tag di origine (nel controllore Logix5550 di partenza). Se si vuole iniziare da un offset all'interno di una matrice, utilizzare un alias che si indirizzi all'offset.</p> <p><b>Non è possibile</b> utilizzare un alias al posto della tag di destinazione. La destinazione deve essere una tag base.</p>	
Logix5550 → PLC-5 Logix5550 → SLC	tag di origine	<i>matrice_1</i>
	elemento di destinazione	<i>N7:10</i>
	<p>È possibile utilizzare una tag alias al posto della tag di origine (nel controllore Logix5550 di partenza). Se si vuole iniziare da un offset all'interno di una matrice, utilizzare un alias che si indirizzi all'offset.</p>	
Logix5550 → PLC-2	tag di origine	<i>matrice_1</i>
	elemento di destinazione	<i>010</i>

Per le istruzioni MSG che hanno origine da un controllore Logix5550 e che leggono da un altro controllore:

<b>Percorso del messaggio:</b>	<b>Origine e Destinazione di esempio:</b>	
Logix5550 → Logix5550	tag di origine	<i>matrice_1</i>
	tag di destinazione	<i>matrice_2</i>
	<p><b>Non è possibile</b> utilizzare una tag alias al posto della tag di origine. L'origine deve essere una tag base.</p> <p>È possibile utilizzare una tag alias al posto della tag di destinazione (nel controllore Logix5550 di partenza). Se si vuole iniziare da un offset all'interno di una matrice, utilizzare un alias che si indirizzi all'offset.</p>	
Logix5550 → PLC-5 Logix5550 → SLC	elemento di origine	<i>N7:10</i>
	tag di destinazione	<i>matrice_1</i>
	<p>È possibile utilizzare una tag alias al posto della tag di destinazione (nel controllore Logix5550 di partenza). Se si vuole iniziare da un offset all'interno di una matrice, utilizzare un alias che si indirizzi all'offset.</p>	
Logix5550 → PLC-2	elemento di origine	<i>010</i>
	tag di destinazione	<i>matrice_1</i>

## Specificazione dei dettagli di comunicazione (Scheda Comunicazione)

Quando si configura un'istruzione MSG, è necessario specificare le seguenti informazioni nella scheda Comunicazione.

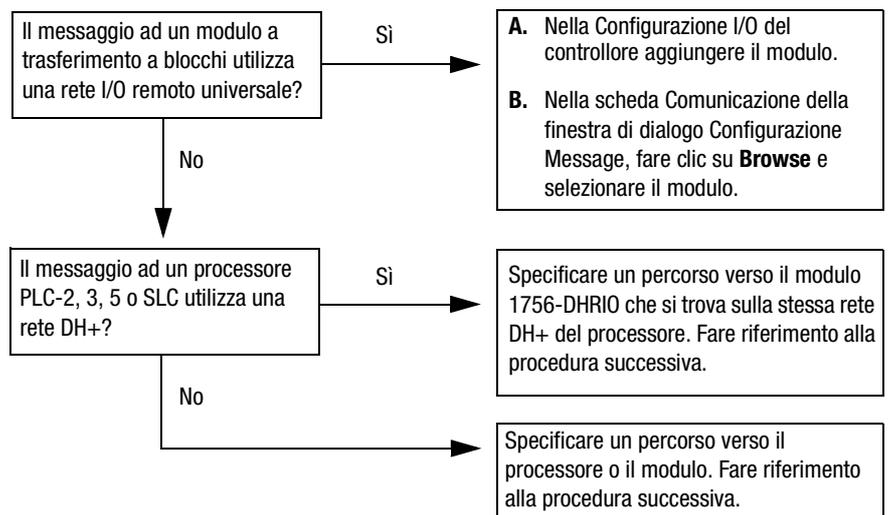


## Specificazione di un percorso di connessione

Il percorso di connessione descrive il percorso del messaggio per raggiungere il processore o il modulo di destinazione. Il percorso utilizza dei numeri, dove ogni coppia di numeri rappresenta il passaggio da un modulo all'altro mediante il backplane ControlLogix o mediante una rete di comunicazione.

**TI** Se la Configurazione I/O contiene il processore o il modulo di destinazione, invece di specificare un percorso, fare clic su **Browse** e selezionare il dispositivo.

Utilizzare il seguente diagramma di flusso per determinare se bisogna inserire un percorso.



**Per specificare un percorso:**

1. Nella casella di testo Percorso, digitare quanto segue, separando ogni numero con una virgola [,]:

A. Specificare la porta di uscita del messaggio. (Il primo numero del percorso, generalmente "1," per indicare la porta del backplane del controllore Logix5550 che invia il messaggio.):

<b>Per il/la:</b>	<b>Specificare:</b>
backplane di un qualsiasi processore o modulo 1756	1
porta DF1 di un controllore 1756-L1	
porta ControlNet di un modulo 1756-CNB	
porta Ethernet di un modulo 1756-ENET	2
porta DH+ sul canale A di un modulo 1756-DHRIO	
porta DH+ sul canale B di un modulo 1756-DHRIO	3

B. Specificare il modulo successivo:

<b>Per un modulo su un/una:</b>	<b>Specificare:</b>
backplane ControlBus	il numero di slot
rete DF1	l'indirizzo della stazione (0-254)
rete ControlNet	il numero di nodo (1-99 decimale)
rete DH+	il numero di nodo (1-77 ottale)
rete Ethernet	l'indirizzo IP (quattro numeri decimali separati da punti)

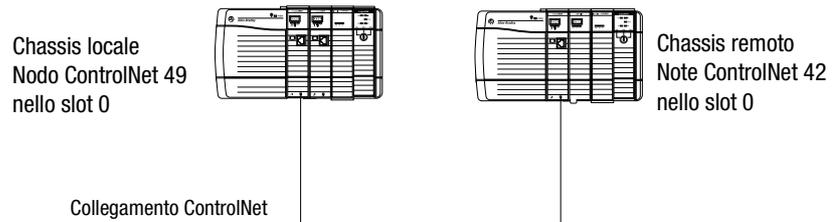
2. Ripetere il passo 1. fino a quando non si specifica il processore o il modulo di destinazione.

Le seguenti pagine mostrano degli esempi di percorsi di connessione:

- su ControlNet, pagina 3-23
- su Ethernet, pagina 3-24
- per un messaggio DH+, pagina 3-24

I due esempi che seguono mostrano i percorsi di connessione tra controllori Logix5550 su reti ControlNet ed Ethernet. Il controllore che invia il messaggio si trova nello chassis locale mentre il controllore che lo riceve si trova nello chassis remoto.

### Esempio 1: Specificazione di un percorso su ControlNet

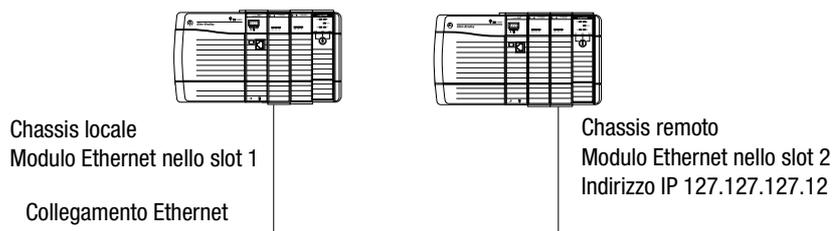


42042

Percorso: 1, 0, 2, 42, 1, 3

<b>Dove:</b>	<b>Indica:</b>
1	la porta del backplane del controllore Logix5550 nello chassis locale
0	il numero di slot del modulo 1756-CNB nello chassis locale
2	la porta ControlNet del modulo 1756-CNB nello slot 0 dello chassis locale
42	il nodo ControlNet del modulo 1756-CNB nello slot 0 dello chassis remoto
1	la porta del backplane del modulo 1756-CNB nello chassis remoto
3	il numero di slot del controllore nello chassis remoto

### Esempio 2: Specificazione di un percorso su ControlNet



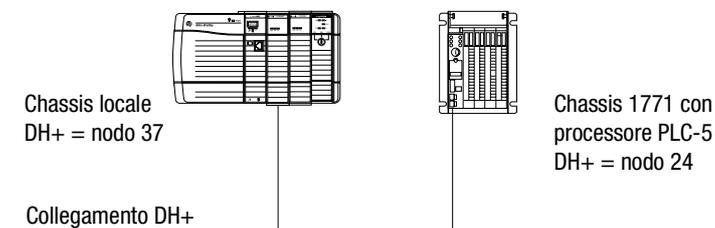
42043

Percorso: 1, 1, 2, 127.127.127.12, 1, 3

Dove:	Indica:
1	la porta del backplane del controllore Logix5550 nello chassis locale
1	il numero di slot del modulo 1756-ENET nello chassis locale
2	la porta Ethernet del modulo 1756-ENET nello chassis locale
127.127.127.12	l'indirizzo IP del modulo 1756-ENET nello chassis remoto
1	la porta del backplane del modulo 1756-ENET nello chassis remoto
3	il numero di slot del controllore nello chassis remoto

Il seguente esempio mostra il percorso di connessione utilizzato in un messaggio ad un processore PLC-5. In questo caso, il percorso termina con il modulo 1756-DHRIO che si trova sulla stessa rete DH+ del processore PLC-5.

### Esempio 3: Specificazione di un percorso per un messaggio DH+



42044

Percorso: 1, 1

Dove:	Indica:
1	la porta del backplane del controllore Logix5550 nello chassis locale
1	il numero di slot del modulo 1756-DHRIO nello chassis locale

### Specificazione di un metodo di comunicazione:

Utilizzare la seguente tabella per selezionare un metodo di comunicazione per il messaggio.

**TI** Per i messaggi CIP generico o a trasferimento a blocchi, *non* selezionare alcun metodo di comunicazione. (Questa sezione non è disponibile.)

Se il dispositivo di destinazione è un:	Selezionare:	Specificare:	
Controllore ControlLogix Processore PLC-5 per ControlNet	<b>CIP</b>	non sono necessarie altre informazioni	
Processore PLC-5	<b>DH+</b>	<b>Canale:</b>	Canale A o B del modulo 1756-DHRIO collegato alla rete DH+
Processore SLC		<b>Collegamento di origine:</b>	ID del collegamento assegnato al backplane del controllore nella tabella di instradamento del modulo 1756-DHRIO. (Il nodo di origine della tabella di instradamento è automaticamente il numero di slot del controllore.)
Processore PLC-3		<b>Collegamento di destinazione</b>	ID del collegamento DH+ remoto dove risiede il dispositivo di destinazione
Processore PLC-2		<b>Nodo di destinazione:</b>	Indirizzo della stazione del dispositivo di destinazione
Se c'è un solo collegamento DH+ e non si è utilizzato il software Gateway per configurare il modulo DH/RIO per collegamenti remoti, specificare 0 sia per il Collegamento di origine che per il Collegamento di destinazione.			
Applicazione su una workstation che riceve un messaggio non richiesto instradato su una rete Ethernet o una rete ControlNet tramite RSLinx.	<b>CIP con ID origine</b>  (Ciò consente all'applicazione di ricevere i dati da un controllore.)	<b>Collegamento di origine:</b>	ID del collegamento dei dati (0-65535) forniti dall'applicazione ad RSLinx
		<b>Collegamento di destinazione:</b>	Impostazione dell'ID di collegamento virtuale in RSLinx (0-65535)
		<b>Nodo di destinazione:</b>	ID di destinazione (0-77 ottale) fornito dall'applicazione ad RSLinx
Il numero di slot del controllore Logix5550 viene utilizzato come Nodo di origine.			

### Selezione di un'opzione con cache:

Alcuni tipi di messaggi utilizzano una connessione per inviare o ricevere dati. Alcuni consentono anche di scegliere se lasciare la connessione aperta (cache) o di chiuderla quando il messaggio è stato trasmesso. La seguente tabella mostra quali messaggi utilizzano una connessione e se è possibile eseguire o meno la cache della connessione:

Un tipo di messaggio:	Che utilizza questo metodo di comunicazione:	Utilizza una connessione:	Di cui è possibile eseguire la cache:
lettura o scrittura di tabella dati CIP	CIP	✓	✓
PLC2, PLC3, PLC5 o SLC (tutti i tipi)	CIP		
	CIP con ID origine		
	DH+	✓	
CIP generico	n/a		
Trasferimento a blocchi di lettura o di scrittura	n/a	✓	✓

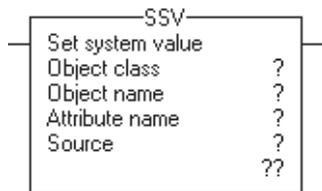
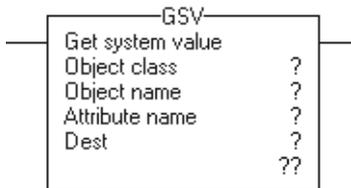
Utilizzare la seguente tabella per selezionare un'opzione cache di un messaggio.

Se il messaggio viene ripetuto:	Allora:	Perché:
ripetutamente	Selezionare la casella di controllo Collegamenti cache	Ciò permette di mantenere la connessione aperta e di ottimizzare il tempo d'esecuzione. L'apertura di una connessione ogni volta che un'istruzione deve essere eseguita comporta un aumento del tempo d'esecuzione.
raramente	Deselezionare la casella di controllo Collegamenti cache	Ciò farà chiudere la connessione al completamento e la renderà disponibile per altri scopi.

## Otteni valore di sistema (GSV) ed Imposta valore di sistema (SSV)

L'istruzione GSV è un'istruzione di uscita.  
L'istruzione SSV è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
<b>Istruzione GSV</b>			
Object class		nome	nome della classe oggetto
Object name		nome	nome dell'oggetto specifico, quando l'oggetto richiede il nome
Attribute name		nome	attributo dell'oggetto il tipo di dati (data type) dipende dall'attributo selezionato
Dest	SINT INT DINT REAL	tag	destinazione dei dati dell'attributo
<b>Istruzione SSV</b>			
Object class		nome	nome della classe oggetto
Object name		nome	nome dell'oggetto specifico, quando l'oggetto richiede il nome
Attribute name		nome	attributo dell'oggetto
Source	SINT INT DINT REAL	tag	tag contenente i dati che si desidera copiare nell'attributo

**Descrizione:** Le istruzioni GSV/SSV servono per ottenere ed impostare i dati di sistema del controllore memorizzati negli oggetti. Il controllore memorizza i dati di sistema in oggetti. Come nel processore PLC-5, non ci sono file di stato.

Quando è abilitata, l'istruzione GSV recupera le informazioni specifiche e le inserisce nella destinazione. Quando è abilitata, l'istruzione SSV imposta l'attributo specifico con i dati dell'origine.

Quando si immette un'istruzione GSV/SSV, il software di programmazione visualizza le classi di oggetto valide, i nomi degli oggetti e i nomi degli attributi di ciascuna istruzione. Per l'istruzione GSV è possibile ottenere i valori per tutti gli attributi disponibili. Nel caso dell'istruzione SSV, il software visualizza solo quegli attributi che è possibile impostare (SSV).

**ATTENZIONE:** utilizzare le istruzioni GSV/SSV con cautela. Apportare modifiche agli oggetti può causare un funzionamento imprevisto del controllore o rischi per le persone.



Se le dimensioni di Source o di Destination sono troppo piccole, l'istruzione non viene eseguita e viene registrato un errore minore (minor fault). Il paragrafo successivo, *Oggetti GSV/SSV*, definisce gli attributi di ciascun oggetto ed i relativi tipi di dati. Ad esempio, l'attributo MajorFaultRecord dell'oggetto Program richiede un tipo di dati DINT[11].

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita è impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita è impostata su falso.
La condizione del ramo di ingresso è vera	Richiede o imposta il valore specificato. La condizione del ramo di uscita è impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

Si verifica un errore minore se:	Tipo errore:	Codice errore:
l'indirizzo dell'oggetto non è valido	4	5
si è specificato un oggetto che non supporta GSV/SSV	4	6
l'attributo non è valido	4	6
non si sono fornite informazioni sufficienti per un'istruzione SSV	4	6
la destinazione GSV non è sufficientemente grande da mantenere i dati richiesti	4	7

**Altri formati:**

Formato:	Sintassi:
testo neutro	<i>GSV(object_class, object_name, attribute_name, destination);</i> <i>SSV(object_class, object_name, attribute_name, destination);</i>
testo ASCII	<i>GSV(object_class, object_name, attribute_name, destination);</i> <i>SSV(object_class, object_name, attribute_name, destination);</i>

**Oggetti GSV/SSV**

Quando si immette un'istruzione GSV/SSV, si specifica l'oggetto e l'attributo a cui si vuole accedere. In alcuni casi vi sarà più di una presenza dello stesso tipo di oggetto, per cui è necessario specificare anche il nome dell'oggetto. Per esempio vi possono essere vari task nell'applicazione. Ogni task ha il proprio oggetto TASK a cui si può accedere con il nome del task.

---

**ATTENZIONE:** Nel caso dell'istruzione GSV, solo la dimensione specificata di dati viene copiata nella destinazione. Se, per esempio, l'attributo viene definito come SINT e la destinazione è un DINT, solo gli 8 bit meno significativi della destinazione DINT verranno aggiornati mentre i rimanenti 24 bit resteranno immutati.



È possibile accedere a questi oggetti:

<b>Per informazioni su questo oggetto:</b>	<b>Vedere pagina:</b>
AXIS	3-30
CONTROLLER	3-37
CONTROLLERDEVICE	3-37
CST	3-39
DF1	3-40
FAULTLOG	3-43
MESSAGE	3-43
MODULE	3-45
MOTIONGROUP	3-46
PROGRAM	3-47
ROUTINE	3-47
SERIALPORT	3-48
TASK	3-49
WALLCLOCKTIME	3-50

## Accesso all'oggetto AXIS

L'oggetto AXIS fornisce informazioni sullo stato del asse di un servo modulo. Specificare il nome della tag dell'asse per selezionare l'oggetto AXIS richiesto.

Per ulteriori informazioni sull'oggetto AXIS, consultare *Logix5550 Modulo controllo assi – Manuale dell'utente*, pubblicazione 1756-6.5.16IT.

Quando un'attributo è segnato con un asterisco (\*), ciò vuol dire che l'attributo si trova sia nel controllore ControlLogix che nel modulo controllo assi. Quando si utilizza un'istruzione SSV per scrivere uno di questi valori, il controllore aggiorna automaticamente la copia nel modulo. Questo processo però non è immediato. Per essere certi che il nuovo valore nel modulo sia stato aggiornato, utilizzare un meccanismo di interblocco tramite i bit booleani dell'UpdateStatus della tag Axis.

Se, ad esempio, si esegue una SSV alla PositionLockTolerance, viene impostato il PositionTolStatus della tag Axis fino a quando il modulo non viene aggiornato. Pertanto, la logica che segue l'SSV potrebbe attendere l'azzeramento di questo bit prima di continuare il programma.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
* AccelerationFeedforwardGain	REAL	GSV SSV	Il valore utilizzato per fornire l'uscita del comando di coppia per generare l'accelerazione di comando.
ActualPosition	REAL	GSV	La posizione effettiva dell'asse.
ActualVelocity	REAL	GSV	La velocità effettiva dell'asse.
AverageVelocity	REAL	GSV	La velocità media dell'asse.
AverageVelocityTimebase	REAL	GSV SSV	La base tempo della velocità media dell'asse.
AxisConfigurationState	SINT	GSV	Lo stato di configurazione dell'asse.
AxisState	SINT	GSV	Lo stato operativo dell'asse. <b>Valore: Significato:</b> 0 asse pronto 1 controllo azionamento diretto 2 controllo servo 3 asse in errore 4 blocco asse
*AxisType	INT	GSV SSV	Il tipo di asse utilizzato. <b>Valore: Significato:</b> 0 asse inutilizzato 1 asse di sola posizione 2 servo asse
C2CConnectionInstance	DINT	GSV	L'istanza di connessione del controllore che produce i dati dell'asse.
C2CMapTableInstance	DINT	GSV	L'istanza della mappa del controllore che produce i dati dell'asse.
CommandPosition	REAL	GSV	La posizione di comando dell'asse.
CommandVelocity	REAL	GSV	La velocità di comando dell'asse.
ConversionConstant	REAL	GSV SSV	Il fattore di conversione usato per convertire le unità scelte in conteggi di retroazione.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
DampingFactor	REAL	GSV SSV	Il valore usato per calcolare la larghezza di banda massima del servo posizionamento durante l'esecuzione dell'istruzione Movimento calcola parametri dell'asse (MRAT).
*DriveFaultAction	SINT	GSV SSV	L'operazione eseguita quando si verifica un errore dell'azionamento. <b>Valore: Significato:</b> 0 blocco asse 1 disabilitazione dell'azionamento 2 blocco del movimento comandato 3 solo modifica del bit di stato
Effectivelnertia	REAL	GSV	Il valore di inerzia dell'asse calcolato in base alle misurazioni eseguite dal controllore durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).
*EncoderLossFaultAction	SINT	GSV SSV	L'operazione eseguita quando si verifica un errore di perdita dell'encoder. <b>Valore: Significato:</b> 0 blocco asse 1 disabilitazione dell'azionamento 2 blocco del movimento comandato 3 solo modifica del bit di stato
*EncoderNoiseFaultAction	SINT	GSV SSV	L'operazione eseguita quando si verifica un errore di disturbo dell'encoder. <b>Valore: Significato:</b> 0 blocco asse 1 disabilitazione dell'azionamento 2 blocco del movimento comandato 3 solo modifica del bit di stato
*FrictionCompensation	REAL	GSV SSV	Il livello di uscita fisso usato per compensare la frizione statica.
GroupInstance	DINT	GSV	Il numero di istanza del gruppo di movimento contenente l'asse.
HomeMode	SINT	GSV SSV	La modalità di impostazione del punto di riferimento dell'asse. <b>Valore: Significato:</b> 0 impostazione punto di riferimento passivo 1 impostazione punto di riferimento attivo (default)
HomePosition	REAL	GSV SSV	Ricerca del punto di riferimento dell'asse.
HomeReturnSpeed	REAL	GSV SSV	La velocità di ritorno al punto di riferimento dell'asse.
HomeSequenceType	SINT	GSV SSV	Il tipo di sequenza di impostazione punto di riferimento dell'asse. <b>Valore: Significato:</b> 0 impostazione punto di riferimento immediata 1 impostazione mediante interruttore di riferimento 2 impostazione punto di riferimento mediante marker 3 impostazione punto di riferimento mediante interruttore-marker (default)
HomeSpeed	REAL	GSV SSV	La velocità di impostazione punto di riferimento dell'asse.
Instance	DINT	GSV	Il numero di istanza dell'asse.
MapTableInstance	DINT	GSV	L'istanza della mappatura I/O del modulo servo.
MaximumAcceleration	REAL	GSV SSV	L'accelerazione massima dell'asse.
MaximumDeceleration	REAL	GSV SSV	La decelerazione massima dell'asse.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
*MaximumNegativeTravel	REAL	GSV SSV	Il limite massimo di corsa negativo.
*MaximumPositiveTravel	REAL	GSV SSV	Il limite massimo di corsa positivo.
MaximumSpeed	REAL	GSV SSV	La velocità massima dell'asse.
ModuleChannel	SINT	GSV	Il canale del modulo del servo modulo.
MotionConfigurationBits	DINT	GSV SSV	I bit di configurazione del movimento dell'asse. <b>Bit:</b> 0 1 2 <b>Significato:</b> direzione ricerca riferimento all'indietro interruttore di riferimento normalmente chiuso fronte di discesa del marker di posizione di riferimento
MotionFaultBits	DINT	GSV	I bit di errore del movimento dell'asse. (Nella struttura AXIS, è il membro MotionFault.) <b>Bit:</b> 0 1 <b>Nome bit:</b> ACAsyncConnFault ACSyncConnFault <b>Significato:</b> errore di connessione asincrona errore di connessione sincrona
MotionStatusBits	DINT	GSV	I bit di stato del movimento dell'asse. (Nella struttura AXIS, è il membro MotionStatus.) <b>Bit:</b> 0 1 2 3 4 5 6 7 <b>Nome bit:</b> AccelStatus DecelStatus MoveStatus JogStatus GearingStatus HomingStatus ClutchStatus AxisHomedStatus <b>Significato:</b> accelerazione decelerazione spostamento avanzamento a velocità costante accoppiamento ritorno al punto di riferimento frizione asse in posizione di riferimento
MotorEncoderTestIncrement	REAL	GSV SSV	Il movimento necessario ad iniziare il test di Movimento Esegui diagnostica schema collegamenti (MRHD).
*OutputFilterBandwidth	REAL	GSV SSV	La larghezza di banda del filtro digitale di uscita passa basso dell'asservimento.
*OutputLimit	REAL	GSV SSV	Il valore della tensione di uscita asservimento massima dell'asse.
*OutputOffset	REAL	GSV SSV	Il valore utilizzato per compensare gli effetti degli offset cumulativi dell'uscita DAC dal modulo servo e dell'ingresso dell'azionamento servo.
*OutputScaling	REAL	GSV SSV	Il valore utilizzato per convertire l'uscita dell'anello di controllo nella tensione equivalente per l'azionamento.
PositionError	REAL	GSV	La differenza tra la posizione effettiva e la posizione di comando dell'asse.
*PositionErrorFaultAction	SINT	GSV SSV	L'operazione eseguita quando si verifica un errore di posizione. <b>Valore:</b> 0 1 2 3 <b>Significato:</b> blocco dell'asse disabilitazione dell'azionamento blocco del movimento comandato solo modifica del bit di stato
*PositionErrorTolerance	REAL	GSV SSV	L'errore di posizione tollerato dall'asservimento prima di segnalare un errore di posizione.

<b>Attributo:</b>	<b>Tipo di dati:</b>	<b>Istruzione:</b>	<b>Descrizione:</b>
*PositionIntegralGain	REAL	GSV SSV	Il valore utilizzato per ottenere un posizionamento preciso dell'asse nonostante disturbi quali la frizione statica e la gravità.
PositionIntegratorError	REAL	GSV	La somma degli errori di posizionamento di un asse.
PositionLockTolerance	REAL	GSV SSV	L'errore di posizione tollerato dal modulo servo quando segnala uno stato vero di posizione bloccata.
*PositionProportionalGain	REAL	GSV SSV	Il valore per cui il controllore moltiplica l'errore di posizione per correggerlo.
PositionServoBandwidth	REAL	GSV SSV	La larghezza di banda del guadagno dell'unità che il controllore utilizza per calcolare i guadagni dell'istruzione Movimento Applica i parametri dell'asse (MAAT).
*PositionUnwind	DINT	GSV SSV	Il valore utilizzato per eseguire lo svolgimento automatico dell'asse rotante.
ProgrammedStopMode	SINT	GSV SSV	Il tipo di stop da eseguire sull'asse. <b>Valore:   Significato:</b> 0       stop rapido 1       blocco rapido 2       blocco hardware
RegistrationPosition	REAL	GSV	La posizione di registrazione dell'asse.
*ServoConfigurationBits	DINT	GSV SSV	I bit di configurazione dell'asservimento per l'anello di controllo. <b>Bit:       Significato:</b> 0       asse rotante 1       velocità esterna dell'azionamento servo 2       polarità encoder negativa 3       polarità asservimento negativa 4       controllo oltrecorsa mediante software 5       controllo errore di posizione 6       controllo errore perdita encoder 7       controllo errore causato da disturbi dell'encoder 8       controllo errore dell'azionamento 9       errore dell'azionamento normalmente chiuso

Attributo:	Tipo di dati:	Istruzione:	Descrizione:																																																																					
ServoConfigurationUpdateBits	DINT	GSV	<p>I bit di stato di configurazione dell'asservimento per l'anello di controllo. (Nella struttura AXIS, è il membro UpdateStatus.)</p> <table border="1"> <thead> <tr> <th>Bit:</th> <th>Nome bit:</th> <th>Significato:</th> </tr> </thead> <tbody> <tr><td>0</td><td>AxisTypeStatus</td><td>tipo di asse</td></tr> <tr><td>1</td><td>PosUnwndStatus</td><td>svolgimento posizione</td></tr> <tr><td>2</td><td>MaxPTrvlStatus</td><td>corsa positiva massima</td></tr> <tr><td>3</td><td>MaxNTrvlStatus</td><td>corsa negativa massima</td></tr> <tr><td>4</td><td>PosErrorTolStatus</td><td>tolleranza errore di posizione</td></tr> <tr><td>5</td><td>PosLockTolStatus</td><td>tolleranza blocco posizione</td></tr> <tr><td>6</td><td>PosPGainStatus</td><td>guadagno proporzionale della posizione</td></tr> <tr><td>7</td><td>PosIGainStatus</td><td>guadagno integrale dell'apozione</td></tr> <tr><td>8</td><td>VelFFGainStatus</td><td>guadagno compensazione anticipata della velocità</td></tr> <tr><td>9</td><td>AccFfGainStatus</td><td>guadagno compensazione anticipata dell'accelerazione</td></tr> <tr><td>10</td><td>VelPGainStatus</td><td>guadagno proporzionale della velocità</td></tr> <tr><td>11</td><td>VelIGainStatus</td><td>guadagno integrale della velocità</td></tr> <tr><td>12</td><td>OutFiltBwStatus</td><td>larghezza di banda del filtro di uscita</td></tr> <tr><td>13</td><td>OutScaleStatus</td><td>conversione in scala dell'uscita</td></tr> <tr><td>14</td><td>OutLimitStatus</td><td>limite uscita</td></tr> <tr><td>15</td><td>OutOffsetStatus</td><td>offset dell'uscita</td></tr> <tr><td>16</td><td>FricCompStatus</td><td>compensazione della frizione</td></tr> <tr><td>17</td><td>POtrvlFaultActStatus</td><td>risposta software all'errore di oltrecorsa</td></tr> <tr><td>18</td><td>PosErrorFaultActStatus</td><td>risposta all'errore di posizione</td></tr> <tr><td>19</td><td>EncLossFaultActStatus</td><td>risposta ad un errore di perdita encoder</td></tr> <tr><td>20</td><td>EncNsFaultActStatus</td><td>risposta ad un errore di disturbi dell'encoder</td></tr> <tr><td>21</td><td>DriveFaultActStatus</td><td>risposta all'errore dell'azionamento</td></tr> </tbody> </table>	Bit:	Nome bit:	Significato:	0	AxisTypeStatus	tipo di asse	1	PosUnwndStatus	svolgimento posizione	2	MaxPTrvlStatus	corsa positiva massima	3	MaxNTrvlStatus	corsa negativa massima	4	PosErrorTolStatus	tolleranza errore di posizione	5	PosLockTolStatus	tolleranza blocco posizione	6	PosPGainStatus	guadagno proporzionale della posizione	7	PosIGainStatus	guadagno integrale dell'apozione	8	VelFFGainStatus	guadagno compensazione anticipata della velocità	9	AccFfGainStatus	guadagno compensazione anticipata dell'accelerazione	10	VelPGainStatus	guadagno proporzionale della velocità	11	VelIGainStatus	guadagno integrale della velocità	12	OutFiltBwStatus	larghezza di banda del filtro di uscita	13	OutScaleStatus	conversione in scala dell'uscita	14	OutLimitStatus	limite uscita	15	OutOffsetStatus	offset dell'uscita	16	FricCompStatus	compensazione della frizione	17	POtrvlFaultActStatus	risposta software all'errore di oltrecorsa	18	PosErrorFaultActStatus	risposta all'errore di posizione	19	EncLossFaultActStatus	risposta ad un errore di perdita encoder	20	EncNsFaultActStatus	risposta ad un errore di disturbi dell'encoder	21	DriveFaultActStatus	risposta all'errore dell'azionamento
Bit:	Nome bit:	Significato:																																																																						
0	AxisTypeStatus	tipo di asse																																																																						
1	PosUnwndStatus	svolgimento posizione																																																																						
2	MaxPTrvlStatus	corsa positiva massima																																																																						
3	MaxNTrvlStatus	corsa negativa massima																																																																						
4	PosErrorTolStatus	tolleranza errore di posizione																																																																						
5	PosLockTolStatus	tolleranza blocco posizione																																																																						
6	PosPGainStatus	guadagno proporzionale della posizione																																																																						
7	PosIGainStatus	guadagno integrale dell'apozione																																																																						
8	VelFFGainStatus	guadagno compensazione anticipata della velocità																																																																						
9	AccFfGainStatus	guadagno compensazione anticipata dell'accelerazione																																																																						
10	VelPGainStatus	guadagno proporzionale della velocità																																																																						
11	VelIGainStatus	guadagno integrale della velocità																																																																						
12	OutFiltBwStatus	larghezza di banda del filtro di uscita																																																																						
13	OutScaleStatus	conversione in scala dell'uscita																																																																						
14	OutLimitStatus	limite uscita																																																																						
15	OutOffsetStatus	offset dell'uscita																																																																						
16	FricCompStatus	compensazione della frizione																																																																						
17	POtrvlFaultActStatus	risposta software all'errore di oltrecorsa																																																																						
18	PosErrorFaultActStatus	risposta all'errore di posizione																																																																						
19	EncLossFaultActStatus	risposta ad un errore di perdita encoder																																																																						
20	EncNsFaultActStatus	risposta ad un errore di disturbi dell'encoder																																																																						
21	DriveFaultActStatus	risposta all'errore dell'azionamento																																																																						
ServoEventBits	DINT	GSV	<p>I bit di evento dell'asservimento per l'anello di controllo. (Nella struttura AXIS, è il membro EventStatus.)</p> <table border="1"> <thead> <tr> <th>Bit:</th> <th>Nome bit:</th> <th>Significato:</th> </tr> </thead> <tbody> <tr><td>0</td><td>WatchEvArmStatus</td><td>evento osservato armato</td></tr> <tr><td>1</td><td>WatchEvStatus</td><td>evento osservato</td></tr> <tr><td>2</td><td>RegEvArmStatus</td><td>evento di registrazione armato</td></tr> <tr><td>3</td><td>RegEvStatus</td><td>evento di registrazione</td></tr> <tr><td>4</td><td>HomeEvArmStatus</td><td>evento di posizione di riferimento armato</td></tr> <tr><td>5</td><td>HomeEvStatus</td><td>evento di posizione di riferimento</td></tr> </tbody> </table>	Bit:	Nome bit:	Significato:	0	WatchEvArmStatus	evento osservato armato	1	WatchEvStatus	evento osservato	2	RegEvArmStatus	evento di registrazione armato	3	RegEvStatus	evento di registrazione	4	HomeEvArmStatus	evento di posizione di riferimento armato	5	HomeEvStatus	evento di posizione di riferimento																																																
Bit:	Nome bit:	Significato:																																																																						
0	WatchEvArmStatus	evento osservato armato																																																																						
1	WatchEvStatus	evento osservato																																																																						
2	RegEvArmStatus	evento di registrazione armato																																																																						
3	RegEvStatus	evento di registrazione																																																																						
4	HomeEvArmStatus	evento di posizione di riferimento armato																																																																						
5	HomeEvStatus	evento di posizione di riferimento																																																																						
ServoFaultBits	DINT	GSV	<p>I bit di errore dell'asservimento per l'anello di controllo. (Nella struttura AXIS, è il membro ServoFault.)</p> <table border="1"> <thead> <tr> <th>Bit:</th> <th>Nome bit:</th> <th>Significato:</th> </tr> </thead> <tbody> <tr><td>0</td><td>POtrvlFault</td><td>errore di oltrecorsa positivo</td></tr> <tr><td>1</td><td>NOtrvlFault</td><td>errore di oltrecorsa negativo</td></tr> <tr><td>2</td><td>PosErrorFault</td><td>errore di posizione</td></tr> <tr><td>3</td><td>EncCHALossFault</td><td>errore di perdita canale A dell'encoder</td></tr> <tr><td>4</td><td>EncCHBLossFault</td><td>errore di perdita canale B dell'encoder</td></tr> <tr><td>5</td><td>EncCHZLossFault</td><td>errore perdita canale Z dell'encoder</td></tr> <tr><td>6</td><td>EncNsFault</td><td>errore di disturbi per l'encoder</td></tr> <tr><td>7</td><td>DriveFault</td><td>errore dell'azionamento</td></tr> <tr><td>8</td><td>SyncConnFault</td><td>errore di connessione sincrona</td></tr> <tr><td>9</td><td>HardFault</td><td>errore hardware dell'asservimento</td></tr> </tbody> </table>	Bit:	Nome bit:	Significato:	0	POtrvlFault	errore di oltrecorsa positivo	1	NOtrvlFault	errore di oltrecorsa negativo	2	PosErrorFault	errore di posizione	3	EncCHALossFault	errore di perdita canale A dell'encoder	4	EncCHBLossFault	errore di perdita canale B dell'encoder	5	EncCHZLossFault	errore perdita canale Z dell'encoder	6	EncNsFault	errore di disturbi per l'encoder	7	DriveFault	errore dell'azionamento	8	SyncConnFault	errore di connessione sincrona	9	HardFault	errore hardware dell'asservimento																																				
Bit:	Nome bit:	Significato:																																																																						
0	POtrvlFault	errore di oltrecorsa positivo																																																																						
1	NOtrvlFault	errore di oltrecorsa negativo																																																																						
2	PosErrorFault	errore di posizione																																																																						
3	EncCHALossFault	errore di perdita canale A dell'encoder																																																																						
4	EncCHBLossFault	errore di perdita canale B dell'encoder																																																																						
5	EncCHZLossFault	errore perdita canale Z dell'encoder																																																																						
6	EncNsFault	errore di disturbi per l'encoder																																																																						
7	DriveFault	errore dell'azionamento																																																																						
8	SyncConnFault	errore di connessione sincrona																																																																						
9	HardFault	errore hardware dell'asservimento																																																																						
ServoOutputLevel	REAL	GSV	Il livello della tensione d'uscita dell'anello di controllo dell'asse.																																																																					

<b>Attributo:</b>	<b>Tipo di dati:</b>	<b>Istruzione:</b>	<b>Descrizione:</b>																					
ServoStatusBits	DINT	GSV	<p>1 bit di stato dell'anello di controllo. (Nella struttura AXIS, è il membro ServoStatus.)</p> <p><b>Bit:</b></p> <table> <tr> <td>0</td> <td>ServoActStatus</td> <td>azione dell'asservimento</td> </tr> <tr> <td>1</td> <td>DriveEnableStatus</td> <td>abilitazione azionamento</td> </tr> <tr> <td>2</td> <td>OutLmtStatus</td> <td>limite uscita</td> </tr> <tr> <td>3</td> <td>PosLockStatus</td> <td>blocco posizione</td> </tr> <tr> <td>13</td> <td>TuneStatus</td> <td>processo di calcolo parametri</td> </tr> <tr> <td>14</td> <td>TestStatus</td> <td>diagnostica di prova</td> </tr> <tr> <td>15</td> <td>ShutdownStatus</td> <td>blocco asse</td> </tr> </table>	0	ServoActStatus	azione dell'asservimento	1	DriveEnableStatus	abilitazione azionamento	2	OutLmtStatus	limite uscita	3	PosLockStatus	blocco posizione	13	TuneStatus	processo di calcolo parametri	14	TestStatus	diagnostica di prova	15	ShutdownStatus	blocco asse
0	ServoActStatus	azione dell'asservimento																						
1	DriveEnableStatus	abilitazione azionamento																						
2	OutLmtStatus	limite uscita																						
3	PosLockStatus	blocco posizione																						
13	TuneStatus	processo di calcolo parametri																						
14	TestStatus	diagnostica di prova																						
15	ShutdownStatus	blocco asse																						
ServoStatusUpdateBits	DINT	GSV	<p>1 bit di aggiornamento stato dell'asservimento dell'asse.</p> <p><b>Bit:</b></p> <table> <tr> <td>0</td> <td>aggiornamento errore di posizione</td> </tr> <tr> <td>1</td> <td>aggiornamento errore integratore di posizione</td> </tr> <tr> <td>2</td> <td>aggiornamento errore di velocità</td> </tr> <tr> <td>3</td> <td>aggiornamento errore integratore di velocità</td> </tr> <tr> <td>4</td> <td>aggiornamento comando velocità</td> </tr> <tr> <td>5</td> <td>aggiornamento feedback velocità</td> </tr> <tr> <td>6</td> <td>aggiornamento livello uscita asservimento</td> </tr> </table>	0	aggiornamento errore di posizione	1	aggiornamento errore integratore di posizione	2	aggiornamento errore di velocità	3	aggiornamento errore integratore di velocità	4	aggiornamento comando velocità	5	aggiornamento feedback velocità	6	aggiornamento livello uscita asservimento							
0	aggiornamento errore di posizione																							
1	aggiornamento errore integratore di posizione																							
2	aggiornamento errore di velocità																							
3	aggiornamento errore integratore di velocità																							
4	aggiornamento comando velocità																							
5	aggiornamento feedback velocità																							
6	aggiornamento livello uscita asservimento																							
*SoftOvertravelFaultAction	SINT	GSV SSV	<p>L'operazione eseguita quando si verifica un errore di oltrecorsa software.</p> <p><b>Valore:</b></p> <table> <tr> <td>0</td> <td>blocco asse</td> </tr> <tr> <td>1</td> <td>disabilitazione dell'azionamento</td> </tr> <tr> <td>2</td> <td>blocco del movimento comandato</td> </tr> <tr> <td>3</td> <td>solo modifica del bit di stato</td> </tr> </table>	0	blocco asse	1	disabilitazione dell'azionamento	2	blocco del movimento comandato	3	solo modifica del bit di stato													
0	blocco asse																							
1	disabilitazione dell'azionamento																							
2	blocco del movimento comandato																							
3	solo modifica del bit di stato																							
StartActualPosition	REAL	GSV	La posizione effettiva dell'asse quando inizia il nuovo movimento comandato dell'asse.																					
StartCommandPosition	REAL	GSV	La posizione di comando dell'asse quando inizia il nuovo movimento comandato dell'asse.																					
StrobeActualPosition	REAL	GSV	La posizione effettiva di un asse quando viene eseguita l'istruzione Movimento Posizione traccia gruppo (MGSP).																					
StrobeCommandPosition	REAL	GSV	La posizione di comando di un asse quando viene eseguita l'istruzione Movimento Posizione traccia gruppo (MGSP).																					
TestDirectionForward	SINT	GSV	<p>La direzione di corsa dell'asse durante l'istruzione di Movimento Esegui diagnostica schema collegamenti (MRHD) vista dal modulo servo.</p> <p><b>Valore:</b></p> <table> <tr> <td>0</td> <td>direzione negativa (indietro)</td> </tr> <tr> <td>1 (ff)</td> <td>direzione positiva (avanti)</td> </tr> </table>	0	direzione negativa (indietro)	1 (ff)	direzione positiva (avanti)																	
0	direzione negativa (indietro)																							
1 (ff)	direzione positiva (avanti)																							
TestStatus	INT	GSV	<p>Lo stato dell'ultima istruzione di Movimento Esegui diagnostica schema collegamenti (MRHD).</p> <p><b>Valore:</b></p> <table> <tr> <td>0</td> <td>test completato</td> </tr> <tr> <td>1</td> <td>test in corso</td> </tr> <tr> <td>2</td> <td>test interrotto dall'utente</td> </tr> <tr> <td>3</td> <td>test 2 secondi oltre il timeout</td> </tr> <tr> <td>4</td> <td>test non superato per errore asservimento</td> </tr> <tr> <td>5</td> <td>incremento test non sufficiente</td> </tr> </table>	0	test completato	1	test in corso	2	test interrotto dall'utente	3	test 2 secondi oltre il timeout	4	test non superato per errore asservimento	5	incremento test non sufficiente									
0	test completato																							
1	test in corso																							
2	test interrotto dall'utente																							
3	test 2 secondi oltre il timeout																							
4	test non superato per errore asservimento																							
5	incremento test non sufficiente																							
TuneAcceleration	REAL	GSV	Il valore di accelerazione misurato durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).																					
TuneAccelerationTime	REAL	GSV	Il tempo di accelerazione misurato durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).																					
TuneDeceleration	REAL	GSV	Il valore di decelerazione misurato durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).																					

<b>Attributo:</b>	<b>Tipo di dati:</b>	<b>Istruzione:</b>	<b>Descrizione:</b>
TuneDecelerationTime	REAL	GSV	Il tempo di decelerazione misurato durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).
TuneRiseTime	REAL	GSV	Il tempo di risposta dell'asse misurato durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).
TuneSpeedScaling	REAL	GSV	Il fattore di conversione in scala dell'azionamento dell'asse misurato durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).
TuneStatus	INT	GSV	Lo stato dell'ultima istruzione Movimento Calcola parametri dell'asse (MRAT). <b>Valore: Significato:</b> 0 calcolo parametri completato 1 calcolo parametri in corso 2 calcolo parametri interrotto dall'utente 3 calcolo parametri 2 secondi oltre il timeout 4 calcolo parametri non superato a causa di un errore dell'asservimento 5 l'asse ha raggiunto il limite corsa per il calcolo parametri 6 polarità asse errata 7 velocità per calcolo parametri troppo bassa per eseguire misurazioni
TuneVelocityBandwidth	REAL	GSV	La larghezza di banda dell'azionamento calcolata in base alle misurazioni eseguite dal controllore durante l'ultima istruzione Movimento Calcola parametri dell'asse (MRAT).
TuningConfigurationBits	DINT	GSV SSV	I bit di configurazione calcolo parametri dell'asse. <b>Bit: Significato:</b> 0 direzione calcolo parametri (=avanti, 1=indietro) 1 calcolo parametri dell'integratore errore di posizione 2 calcolo parametri dell'integratore errore di velocità 3 calcolo parametri bit di compensazione anticipata della velocità 4 compensazione anticipata accelerazione 5 calcolo parametri filtro passa basso della velocità
TuningSpeed	REAL	GSV SSV	La velocità massima iniziata dall'istruzione Movimento Calcola parametri dell'asse (MRAT).
TuningTravelLimit	REAL	GSV SSV	Il limite di corsa usato dall'istruzione Movimento Calcola parametri dell'asse (MRAT) per limitare l'azione della corsa durante il calcolo parametri.
VelocityCommand	REAL	GSV	Il riferimento della velocità corrente rispetto all'anello di controllo della velocità dell'asse.
VelocityError	REAL	GSV	La differenza tra la velocità comandata e la velocità effettiva di un asse servo.
VelocityFeedback	REAL	GSV	La velocità effettiva dell'asse per come è stimata dal modulo servo.
*VelocityFeedforwardGain	REAL	GSV SSV	L'uscita del comando di velocità necessaria a generare la velocità comandata.
*VelocityIntegralGain	REAL	GSV SSV	Il valore per il quale il controllore moltiplica il valore VelocityIntegratorError per correggere l'errore di velocità.
VelocityIntegratorError	REAL	GSV	L'errore di velocità totale di un determinato asse.
*VelocityProportionalGain	REAL	GSV SSV	Il valore per il quale il controllore moltiplica il valore VelocityError per correggere l'errore di velocità.
WatchPosition	REAL	GSV	La posizione osservata dell'asse.

### Accesso all'oggetto CONTROLLER

L'oggetto CONTROLLER fornisce informazioni di stato riguardo l'esecuzione di un controllore.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
TimeSlice	INT	GSV SSV	Percentuale di CPU disponibile assegnata alla comunicazione. Valori validi da 10 a 90. Questo valore non può essere modificato quando il selettore a chiave del controllore è nella posizione di esecuzione.

### Accesso all'oggetto CONTROLLERDEVICE

L'oggetto CONTROLLERDEVICE identifica l'hardware fisico del controllore.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
DeviceName	SINT[33]	GSV	nome stringa in ASCII del controllore. Il primo byte contiene un conteggio del numero di caratteri ASCII riportati nella stringa matrice.
ProductCode	INT	GSV	Identifica il tipo di controllore Logix5550 = 3
ProductRev	INT	GSV	Indica la revisione corrente del prodotto. Il display deve essere esadecimale. Il byte basso contiene la revisione principale; il byte alto contiene la revisione secondaria.
SerialNumber	DINT	GSV	Numero seriale del dispositivo. Il numero seriale viene assegnato al dispositivo al momento della sua costruzione.

<b>Attributo:</b>	<b>Tipo di dati:</b>	<b>Istruzione:</b>	<b>Descrizione:</b>
Status	INT	GSV	<p>Bit che indicano lo stato: I bit 3-0 sono riservati</p> <p><b>Bit di stato del dispositivo</b>  <b>Bit 7-4:      Significato:</b>            0000      riservato            0001      aggiornamento della memoria flash in corso            0010      riservato            0011      riservato            0100      la memoria flash è difettosa            0101      in errore            0110      esecuzione            0111      programmazione</p> <p><b>Bit di stato di errore</b>  <b>Bit 11-8:      Significato:</b>            0001      errore minore recuperabile            0010      errore minore non recuperabile            0100      errore grave recuperabile            1000      errore grave non recuperabile</p> <p><b>Bit di stato specifici del Logix5550</b>  <b>Bit 13-12:      Significato:</b>            01      selettore a chiave in posizione esecuzione            10      selettore a chiave in posizione programmazione            11      selettore a chiave in posizione remoto</p> <p><b>Bit 15-14:      Significato:</b>            01      il controllore sta cambiando modalità            10      modalità debug se il controllore è in modalità esecuzione</p>
Type	INT	GSV	<p>Identifica il dispositivo come controllore. Controllore = 14</p>
Vendor	INT	GSV	<p>Identifica il produttore del dispositivo. Allen-Bradley = 0001</p>

## Accesso all'oggetto CST

L'oggetto CST (tempo di sistema coordinato) fornisce il tempo di sistema coordinato dei dispositivi di uno chassis.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
CurrentStatus	INT	GSV	Stato corrente del tempo di sistema coordinato. I bit indicano:
			<b>Bit:</b> <b>Significato:</b>
			0 <b>errore hardware del timer:</b> l'hardware del timer interno del dispositivo è in stato di errore
			1 <b>rampa abilitata:</b> il valore corrente dei bit 16+ meno significativi del timer sale fino al valore richiesto invece di passare al valore più basso. Questi bit vengono manipolati tramite il metodo di sincronizzazione specifico della rete
			2 <b>tempo di sistema master:</b> l'oggetto CST è una sorgente di tempo master nel sistema ControlLogix
			3 <b>sincronizzato:</b> il CurrentValue a 64 bit dell'oggetto CST è sincronizzato da un oggetto CST master tramite un aggiornamento tempo di sistema
			4 <b>rete locale master:</b> l'oggetto CST è la sorgente di tempo master della rete locale
			5 <b>in modalità relè:</b> l'oggetto CST sta funzionando in modalità relè a tempo
			6 <b>rilevato master duplicato:</b> è stato rilevato un master di tempo della rete locale duplicato. Questo bit è sempre 0 per i nodi a tempo.
			7            non usato
8-9        00 = nodo a tempo dipendente 01 = nodo a tempo master 10 = nodo a relè a tempo 11 = non usato			
10-15     non usato			
CurrentValue	DINT[2]	GSV	Valore corrente del timer. DINT[0] contiene i 32 bit più bassi; DINT[1] contiene i 32 bit più alti. La sorgente del timer viene regolata in modo da combaciare con il valore fornito nei servizi di aggiornamento e tramite la sincronizzazione della rete di comunicazione locale. La regolazione viene fatta o salendo fino al valore richiesto o con un'impostazione immediata sul valore richiesto, così come riportato nell'attributo CurrentStatus.

## Accesso all'oggetto DF11

L'oggetto DF1 fornisce un'interfaccia verso il driver di comunicazione DF1 che è possibile configurare per la porta seriale.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
ACKTimeout	DINT	GSV	Il tempo necessario per ottenere un riscontro della trasmissione di un messaggio (solo punto a punto e master). valore valido 0-32.767. Ritardo in conteggi di 20 msec. Il valore predefinito è 50 (1 secondo).
DiagnosticCounters	INT[19]	GSV	Matrice di contatori diagnostici per il driver di comunicazione DF1.
<b>offset parola</b>	<b>DF1 punto a punto</b>	<b>DF1 slave</b>	<b>master</b>
0	firma (0x0043)	firma (0x0042)	firma (0x0044)
1	bit modem	bit modem	bit modem
2	pacchetti inviati	pacchetti inviati	pacchetti inviati
3	pacchetti ricevuti	pacchetti ricevuti	pacchetti ricevuti
4	pacchetti non consegnati	pacchetti non consegnati	pacchetti non consegnati
5	non usato	messaggi rinviati	messaggi rinviati
6	NAK ricevuti	NAK ricevuti	non usato
7	ENQ ricevuti	pacchetti di interrogazione richiesti	non usato
8	pacchetti errati con NAK	pacchetti errati senza ACK	pacchetti errati senza ACK
9	NAK inviati senza memoria	senza memoria senza ACK	non usato
10	pacchetti duplicati ricevuti	pacchetti duplicati ricevuti	pacchetti duplicati ricevuti
11	caratteri errati ricevuti	non usato	non usato
12	conteggio recuperi DCD	conteggio recuperi DCD	conteggio recuperi DCD
13	conteggio interruzioni modem	conteggio interruzioni modem	conteggio interruzioni modem
14	non usato	non usato	tempo di scansione priorità massimo
15	non usato	non usato	ultimo tempo di scansione priorità
16	non usato	non usato	tempo di scansione normale massimo
17	non usato	non usato	ultimo tempo di scansione normale
18	ENQ inviati	non usato	non usato
DuplicateDetection	SINT	GSV	Abilita il rilevamento di messaggi duplicati. <b>Valore:</b> 0 non zero <b>Significato:</b> rilevamento messaggi duplicati disabilitato rilevamento messaggi duplicati abilitato
EmbeddedResponseEnable	SINT	GSV	Abilita la funzione di risposta integrata (solo punto a punto). <b>Valore:</b> 0 1 <b>Significato:</b> iniziato solo dopo averne ricevuto una (default) abilitato incondizionatamente
ENQTransmitLimit	SINT	GSV	Il numero di richieste (ENQ) da inviare dopo un timeout di ACK (solo punto a punto). Valore valido 0-127. Il valore di default è 3.
EOTSuppression	SINT	GSV	Abilita la soppressione delle trasmissioni EOT in risposta a tutti i pacchetti di interrogazione (solo slave). <b>Valore:</b> 0 non zero <b>Significato:</b> soppressione EOT disabilitata (disabilitata) soppressione EOT abilitata
ErrorDetection	SINT	GSV	Indica lo schema di rilevamento errori. <b>Valore:</b> 0 1 <b>Significato:</b> BCC (default) CRC

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
MasterMessageTransmit	SINT	GSV	<p>Il valore corrente della trasmissione messaggi master (solo master).</p> <p><b>Valore:</b> 0 1</p> <p><b>Significato:</b> tra interrogazioni delle stazioni in sequenza alle interrogazioni (al posto del numero di stazione del master)</p> <p>Il valore di default è 0.</p>
NAKReceiveLimit	SINT	GSV	<p>Il numero di NAK ricevuti in risposta ad un messaggio prima di chiudere la trasmissione (solo comunicazione punto a punto). Valore valido 0-127. Il valore di default è 3.</p>
NormalPollGroupSize	INT	GSV	<p>Il numero di stazioni da interrogare nella matrice dei nodi ad interrogazione normale dopo l'interrogazione di tutte le stazioni nella matrice dei nodi ad interrogazione con priorità (solo master). Valore valido 0-255. Il valore di default è 0.</p>
PollingMode	SINT	GSV	<p>Modalità di interrogazione corrente (solo master).</p> <p><b>Valore:</b> 0 1 2 3</p> <p><b>Significato:</b> basata su messaggi, ma non permette agli slave di iniziare messaggi basata su messaggi, ma permette agli slave di iniziare messaggi (default) trasferimento standard, con messaggio singolo per scansione di nodo trasferimento standard, con più messaggi per scansione di nodo</p> <p>Il valore di default è 1.</p>
ReplyMessageWait	DINT	GSV	<p>Il tempo (che si comporta da master) che bisogna attendere dopo il ricevimento di un ACK prima di interrogare lo slave per una risposta (solo slave). Valore valido 0-65,535. Ritardo in unità da 20 msec. Il valore di default è 5 unità (100 msec).</p>
StationAddress	INT	GSV	<p>L'indirizzo della stazione corrente della porta seriale. Valore valido 0-254. Il valore di default è 0.</p>
SlavePollTimeout	DINT	GSV	<p>Il tempo espresso in msec. che lo slave attende affinché il master esegua l'interrogazione prima che lo slave dichiarare che è impossibilitato a trasmettere poiché il master è inattivo (solo slave). valore valido 0-32,767. Ritardo in conteggi di 20 msec. Il valore di default è 3000 unità (1 minuto).</p>
TransmitRetries	SINT	GSV	<p>Il numero di tentativi di invio di un messaggio senza ottenere un riconoscimento (solo master e slave). Valore valido 0-127. Il valore di default è 3.</p>
PendingACKTimeout	DINT	SSV	<p>Valore in attesa per l'attributo ACKTimeout.</p>
PendingDuplicateDetection	SINT	SSV	<p>Valore in attesa per l'attributo DuplicateDetection.</p>
PendingEmbeddedResponse Enable	SINT	SSV	<p>Valore in attesa per l'attributo EmbeddedResponse.</p>
PendingENQTransmitLimit	SINT	SSV	<p>Valore in attesa per l'attributo ENQTransmitLimit.</p>
PendingEOTSuppression	SINT	SSV	<p>Valore in attesa per l'attributo EOTSuppression.</p>
PendingErrorDetection	SINT	SSV	<p>Valore in attesa per l'attributo ErrorDetection.</p>
PendingNormalPollGroupSize	INT	SSV	<p>Valore in attesa per l'attributo NormalPollGroupSize.</p>
PendingMasterMessage Transmit	SINT	SSV	<p>Valore in attesa per l'attributo MasterMessageTransmit.</p>

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
PendingNAKReceiveLimit	SINT	SSV	Valore in attesa per l'attributo NAKReceiveLimit.
PendingPollingMode	SINT	SSV	Valore in attesa per l'attributo PollingMode.
PendingReplyMessageWait	DINT	SSV	Valore in attesa per l'attributo ReplyMessageWait.
PendingStationAddress	INT	SSV	Valore in attesa per l'attributo StationAddress.
PendingSlavePollTimeout	DINT	SSV	Valore in attesa per l'attributo SlavePollTimeout.
PendingTransmitRetries	SINT	SSV	Valore in attesa per l'attributo TransmitRetries.

Per applicare i valori per ciascuno degli attributi di DF1 in attesa:

1. Utilizzare un'istruzione SSV per impostare il valore per l'attributo in attesa.

È possibile impostare tutti gli attributi in attesa che si desidera utilizzando un'istruzione SSV per ogni attributo in attesa.

2. Utilizzare un'istruzione MSG per applicare il valore. L'istruzione MSG si applica ad ogni attributo in attesa che si è impostato. Configurate l'istruzione MSG come segue:

Scheda di configurazione MSG:	Campo:	Valore:
Configuration	Message Type	CIP generico
	Service Code	0d esad.
	Object Type	a2
	Object ID	1
	Object attribute	lasciare vuoto
	Source	lasciare vuoto
	Number of Elements	0
	Destination	lasciare vuoto
Communication	Path	percorso di comunicazione verso se stesso (1,s dove s = numero di slot del controllore)

## Accesso all'oggetto FAULTLOG

L'oggetto FAULTLOG fornisce informazioni sugli errori del controllore.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
MajorEvents	INT	GSV SSV	Il numero di errori gravi che si sono verificati dall'ultima volta che questo contatore è stato azzerato.
MinorEvents	INT	GSV SSV	Il numero di errori minori che si sono verificati dall'ultima volta che questo contatore è stato azzerato.
MajorFaultBits	DINT	GSV SSV	I singoli bit indicano il motivo dell'errore grave corrente. <b>Bit:      Significato:</b> 1        perdita di alimentazione 3        I/O 4        esecuzione dell'istruzione (programma) 5        programma di gestione errori 6        watchdog 7        stack 8        cambiamento di modalità 11      movimento
MinorFaultBits	DINT	GSV SSV	I singoli bit indicano il motivo dell'errore minore corrente. <b>Bit:      Significato:</b> 4        esecuzione dell'istruzione (programma) 6        watchdog 9        porta seriale 10      batteria

## Accesso all'oggetto MESSAGE

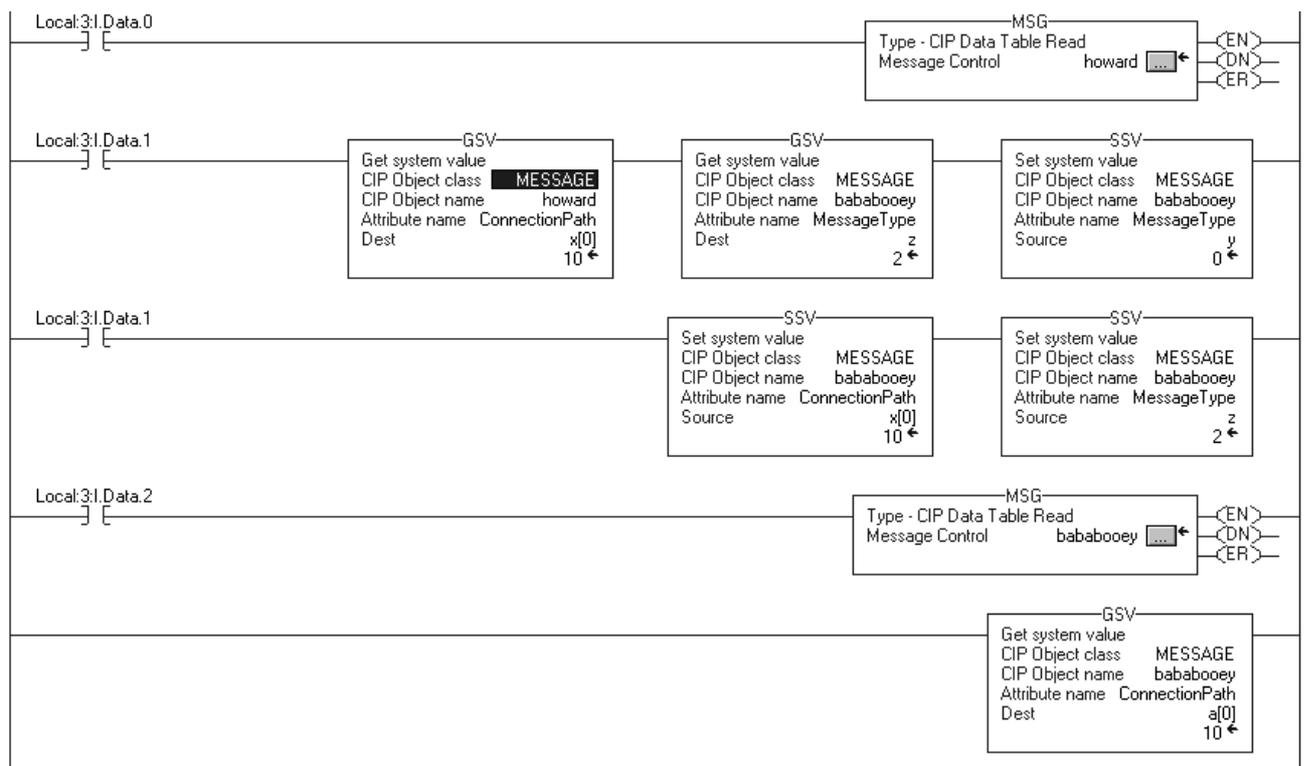
È possibile accedere all'oggetto MESSAGE mediante le istruzioni GSV/SSV. Specificare il nome della tag del messaggio per determinare l'oggetto MESSAGE desiderato. L'oggetto MESSAGE fornisce un'interfaccia per impostare ed attivare la comunicazione peer-to-peer. Questo oggetto sostituisce il tipo di dati MG del processore PLC-5.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
ConnectionPath	SINT[130]	GSV SSV	Dati per l'impostazione del percorso di connessione. I primi due byte (byte basso e byte alto) rappresentano la lunghezza in byte del percorso di connessione.
ConnectionRate	DINT	GSV SSV	Frequenza pacchetto richiesto della connessione.
MessageType	SINT	GSV SSV	Specifica il tipo di messaggio. <b>Valore:      Significato:</b> 0        non inizializzato
Port	SINT	GSV SSV	Indica su quale porta deve essere inviato il messaggio. <b>Valore:      Significato:</b> 1        backplane 2        porta seriale

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
TimeoutMultiplier	SINT	GSV SSV	Determina quando una connessione deve essere considerata scaduta e chiusa. <b>Valore: Significato:</b> 0 la connessione scadrà in un tempo pari a 4 volte la frequenza di aggiornamento (default) 1 la connessione scadrà in un tempo pari ad 8 volte la frequenza di aggiornamento 2 la connessione scadrà in un tempo pari ad 16 volte la frequenza di aggiornamento
UnconnectedTimeout	DINT	GSV SSV	Il periodo di timeout è espresso in microsecondi per tutti i messaggi non connessi. Il valore di default è 30.000.000 microsecondi (30 secondi).

Per modificare un attributo MESSAGE, procedere in questo modo:

1. Utilizzare un'istruzione GSV per ottenere l'attributo MessageType e salvarlo in una tag.
2. Utilizzare un'istruzione SSV per impostare il MessageType su 0.
3. Utilizzare un'istruzione SSV per impostare l'attributo MESSAGE che si desidera modificare.
4. Utilizzare un'istruzione SSV per impostare l'attributo MessageType sul valore originario ottenuto al punto 1.



## Accesso all'oggetto MODULE

L'oggetto MODULE fornisce informazioni sullo stato di un modulo. Per selezionare un determinato oggetto MODULE, impostare nell'operando Nome Oggetto dell'istruzione GSV/SSV il nome del modulo. Il modulo specificato deve essere presente nella sezione Configurazione I/O dell'organizer del controllore e deve avere un nome di dispositivo.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:			
EntryStatus	INT	GSV	Indica lo stato corrente dell'elemento di mappa specificato. Quando si esegue un'operazione di confronto, i 12 bit più bassi devono essere mascherati. Solo i bit 12-15 sono validi.			
			<b>Valore:</b> 16#0000 16#1000	<b>Significato:</b> <b>Standby:</b> il controllore è in fase di accensione. <b>In errore:</b> errore di una delle connessioni dell'oggetto MODULE associata al modulo. Questo valore non deve essere usato per stabilire se il modulo è in errore poiché l'oggetto MODULE esce da questo stato periodicamente quando tenta di riconnettersi al modulo. Si consiglia invece di verificare lo stato di Esecuzione (16#4000). Controllare che FaultCode sia diverso da 0 per determinare se un modulo è guasto. Se vi è un errore, gli attributi FaultCode e FaultInfo rimangono validi fino a quando la condizione di errore non viene corretta.		
			16#2000	<b>Validazione:</b> l'oggetto MODULE sta verificando l'integrità dell'oggetto MODULE prima di stabilire le connessioni con il modulo.		
			16#3000	<b>Connessione in corso:</b> l'oggetto MODULE sta iniziando le connessioni con il modulo.		
			16#4000	<b>In esecuzione:</b> tutte le connessioni con il modulo sono attive ed i dati vengono trasferiti correttamente.		
			16#5000	<b>Chiusura in corso:</b> l'oggetto MODULE sta eseguendo la chiusura di tutte le connessioni con il modulo.		
			16#6000	<b>Inibito:</b> l'oggetto MODULE è inibito (il bit di inibizione nell'attributo Modalità è impostato).		
			16#7000	<b>In attesa:</b> l'oggetto MODULE principale da cui dipende questo oggetto MODULE non è in esecuzione.		
			FaultCode	INT	GSV	Un numero che, in caso di errore del modulo, lo identifica.
			FaultInfo	DINT	GSV	Fornisce informazioni specifiche sul codice di errore dell'oggetto MODULE.
ForceStatus	INT	GSV	Indica lo stato delle forzature.			
			<b>Bit:</b> 0 1 2-15	<b>Significato:</b> forzature installate (1=sì, 0=no) forzature abilitate (1= sì, 0=no) non usati		

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
Instance	DINT	GSV	Fornisce il numero di istanza di questo oggetto MODULE.
LEDStatus	INT	GSV	<p>Indica lo stato corrente del LED I/O posto sul frontale del controllore.</p> <p><b>Valore: Significato:</b></p> <p>0 <b>LED spento:</b> Nessun oggetto MODULE configurato per il controllore (nella Configurazione I/O dell'organizer del controllore non è presente alcun modulo).</p> <p>1 Rosso lampeggiante: nessun oggetto MODULE in esecuzione.</p> <p>2 Verde lampeggiante: almeno un oggetto MODULE non è in esecuzione.</p> <p>3 Verde fisso: tutti gli oggetti MODULE sono in esecuzione.</p> <p><b>Nota:</b> non inserire un nome di oggetto con questo attributo in quanto questo attributo si riferisce a tutto il gruppo di moduli.</p>
Mode	INT	GSV SSV	<p>Indica la modalità corrente dell'oggetto MODULE.</p> <p><b>Bit: Significato:</b></p> <p>0 Se è impostato, provoca un errore grave qualora una delle connessioni dell'oggetto MODULE va in errore mentre il controllore è in modalità esecuzione (RUN).</p> <p>2 Se è impostato, fa sì che l'oggetto MODULE passi allo stato Inibito dopo la chiusura di tutte le connessioni con il modulo.</p>

### Accesso all'oggetto MOTIONGROUP

L'oggetto MOTIONGROUP fornisce informazioni sullo stato di un gruppo di assi del modulo servo. Specificare il nome della tag del gruppo assi per determinare l'oggetto MOTIONGROUP desiderato.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
Instance	DINT	GSV	Fornisce il numero di istanza di questo oggetto MOTION_GROUP.

## Accesso all'oggetto PROGRAM

L'oggetto PROGRAM fornisce informazioni sullo stato di un programma. Specificare il nome del programma per determinare l'oggetto PROGRAM desiderato.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
DisableFlag	SINT	GSV SSV	Controlla l'esecuzione di questo programma. <b>Valore: Significato:</b> 0 esecuzione abilitata 1 esecuzione disabilitata
Instance	DINT	GSV	Fornisce il numero di istanza di questo oggetto PROGRAM.
LastScanTime	DINT	GSV SSV	Il tempo impiegato per eseguire questo programma l'ultima volta che è stato eseguito. Tempo espresso in microsecondi.
MajorFaultRecord	DINT[11]	GSV SSV	Registra gli errori gravi di questo programma Si consiglia di creare una struttura definita dall'utente per semplificare l'accesso all'attributo MajorFaultRecord:
<b>Nome:</b> TimeLow	<b>Tipo dati</b> DINT	<b>Stile:</b> Decimale	<b>Descrizione:</b> i 32 bit più bassi del valore della registrazione cronologica dell'errore
TimeHigh	DINT	Decimale	i 32 bit più alti del valore della registrazione cronologica dell'errore
Type	INT	Decimale	tipo di errore (programma, I/O, ecc.)
Code	INT	Decimale	codice univoco per l'errore (dipende dal tipo di errore)
Info	DINT[8]	Esadecimale	informazioni specifiche sull'errore (in base al tipo e al codice errore)
MaxScanTime	DINT	GSV SSV	Tempo di esecuzione massimo registrato di questo programma. Tempo espresso in microsecondi.
MinorFaultRecord	DINT[11]	GSV SSV	Registra gli errori minori di questo programma Si consiglia di creare una struttura definita dall'utente per semplificare l'accesso all'attributo MinorFaultRecord:
<b>Nome:</b> TimeLow	<b>Tipo dati</b> DINT	<b>Stile:</b> Decimale	<b>Descrizione:</b> i 32 bit più bassi del valore della registrazione cronologica dell'errore
TimeHigh	DINT	Decimale	i 32 bit più alti del valore della registrazione cronologica dell'errore
Type	INT	Decimale	tipo di errore (programma, I/O, ecc.)
Code	INT	Decimale	codice univoco per l'errore (dipende dal tipo di errore)
Info	DINT[8]	Esadecimale	informazioni specifiche sull'errore (in base al tipo e al codice errore)
SFCRestart	INT	GSV SSV	non usato – riservato per uso futuro

## Accesso all'oggetto ROUTINE

L'oggetto ROUTINE fornisce informazioni sullo stato di una routine. Specificare il nome della routine per determinare l'oggetto ROUTINE desiderato.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
Instance	DINT	GSV	Fornisce il numero di istanza di questo oggetto ROUTINE. Valori validi sono 0-65.535.

## Accesso all'oggetto SERIALPORT

L'oggetto SERIALPORT fornisce un'interfaccia alla porta di comunicazione seriale.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
BaudRate	DINT	GSV	Specifica la velocità di trasmissione (baud rate). Valori validi sono 110, 300, 600, 1200, 2400, 4800, 9600 e 19200 (default).
DataBits	SINT	GSV	Indica il numero di bit di dati per carattere. <b>Valore: Significato:</b> 7      7 bit dati (solo ASCII) 8      8 bit dati (default)
Parity	SINT	GSV	Specifica la parità. <b>Valore: Significato:</b> 0      nessuna parità (no default) 1      parità dispari (solo ASCII) 2      parità pari
RTSOffDelay	INT	GSV	Il tempo di ritardo per la disattivazione della linea RTS (Request To Send) dopo che l'ultimo carattere è stato trasmesso. valore valido 0-32.767. Ritardo in conteggi di 20 msec. Il valore di default è 0 msec.
RTSSendDelay	INT	GSV	Il tempo di ritardo per la trasmissione del primo carattere di un messaggio dopo l'attivazione della linea RTS. valore valido 0-32.767. Ritardo in conteggi di 20 msec. Il valore di default è 0 msec.
StopBits	SINT	GSV	Indica il numero di bit di stop. <b>Valore: Significato:</b> 1      1 bit di stop (default) 2      2 bit stop (solo ASCII)
PendingBaudRate	DINT	SSV	Valore in attesa dell'attributo BaudRate.
PendingDataBits	SINT	SSV	Valore in attesa dell'attributo DataBits.
PendingParity	SINT	SSV	Valore in attesa dell'attributo Parity.
PendingRTSOffDelay	INT	SSV	Valore in attesa dell'attributo RTSOffDelay.
PendingRTSSendDelay	INT	SSV	Valore in attesa dell'attributo RTSSendDelay.
PendingStopBits	SINT	SSV	Valore in attesa dell'attributo StopBits.

Per applicare i valori per uno degli attributi di SERIALPORT in attesa:

1. Utilizzare un'istruzione SSV per impostare il valore per l'attributo in attesa.

È possibile impostare tutti gli attributi in attesa che si desidera utilizzando un'istruzione SSV per ogni attributo in attesa.

2. Utilizzare un'istruzione MSG per applicare il valore. L'istruzione MSG si applica ad ogni attributo in attesa che si è impostato. Configurare le istruzioni MSG come segue:

<b>Scheda di configurazione MSG:</b>	<b>Campo:</b>	<b>Valore:</b>
Configuration	Message Type	CIP generico
	Service Code	0d esad.
	Object Type	6f esad.
	Object ID	1
	Object attribute	lasciare vuoto
	Source	lasciare vuoto
	Number of Elements	0
	Destination	lasciare vuoto
Communication	Path	percorso di comunicazione verso se stesso (1, s dove s = numero di slot del controllore)

### Accesso all'oggetto TASK

L'oggetto TASK fornisce informazioni sullo stato di un task. Specificare il nome del task per determinare l'oggetto TASK desiderato.

<b>Attributo:</b>	<b>Tipo di dati:</b>	<b>Istruzione:</b>	<b>Descrizione:</b>
Instance	DINT	GSV	Fornisce il numero di istanza di questo oggetto TASK. Valori validi sono 0-31.
LastScanTime	DINT	GSV SSV	Il tempo impiegato per eseguire questo task l'ultima volta che è stato eseguito. Tempo espresso in microsecondi.
MaxInterval	DINT[2]	GSV SSV	L'intervallo di tempo massimo tra esecuzioni successive del task. DINT[0] contiene i 32 bit meno significativi del valore; DINT[1] contiene i 32 bit più significativi del valore. Un valore di 0 indica 1 o meno esecuzioni del task.
MaxScanTime	DINT	GSV SSV	Tempo di esecuzione massimo registrato di questo programma. Tempo espresso in microsecondi.
MinInterval	DINT[2]	GSV SSV	L'intervallo di tempo minimo tra esecuzioni successive del task. DINT[0] contiene i 32 bit meno significativi del valore; DINT[1] contiene i 32 bit più significativi del valore. Un valore di 0 indica 1 o meno esecuzioni del task.
Priority	INT	GSV	La priorità relativa di questo task in confronto agli altri task. Valori validi sono 0-15.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
Rate	DINT	GSV	L'intervallo di tempo tra le esecuzioni del task. Tempo espresso in microsecondi.
StartTime	DINT[2]	GSV SSV	Il valore del WALLCLOCKTIME quando l'ultima esecuzione del task ha avuto inizio. DINT[0] contiene i 32 bit meno significativi del valore; DINT[1] contiene i 32 bit più significativi del valore.
Watchdog	DINT	GSV SSV	Tempo limite per l'esecuzione di tutti i programmi associati a questo task. Tempo espresso in microsecondi. Se si inserisce 0, vengono assegnati questi valori: <b>Tempo: Tipo di task:</b> 0,5 sec periodico 5,0 sec continuo

### Accesso all'oggetto WALLCLOCKTIME

L'oggetto WALLCLOCKTIME fornisce una registrazione cronologica che il controllore utilizza per la schedulazione.

Attributo:	Tipo di dati:	Istruzione:	Descrizione:
CSTOffset	DINT[2]	GSV SSV	Offset positivo dal CurrentValue dell'oggetto CST (tempo di sistema coordinato, vedere pagina 3-39). DINT[0] contiene i 32 bit meno significativi del valore; DINT[1] contiene i 32 bit più significativi del valore. Il valore è espresso in $\mu$ sec. Il valore di default è 0.
CurrentValue	DINT[2]	GSV SSV	Il valore corrente dell'ora dell'orologio. DINT[0] contiene i 32 bit meno significativi del valore; DINT[1] contiene i 32 bit più significativi del valore. Il valore rappresenta il numero di microsecondi trascorsi dalle 0000 del 1 Gennaio 1972. Gli oggetti CST e WALLCLOCKTIME sono matematicamente correlati nel controllore. Ad esempio, se si sommano il CurrentValue del CST e il CSTOffset del WALLCLOCKTIME, il risultato è il CurrentValue del WALLCLOCKTIME.
DateTime	DINT[7]	GSV SSV	La data e l'ora in formato leggibile. DINT[0] anno DINT[1] intero rappresentante il mese (1-12) DINT[2] intero rappresentante il giorno (1-31) DINT[3] ora (0-23) DINT[4] minuto (0-59) DINT[5] secondi (0-59) DINT[6] microsecondi (0-999.999)

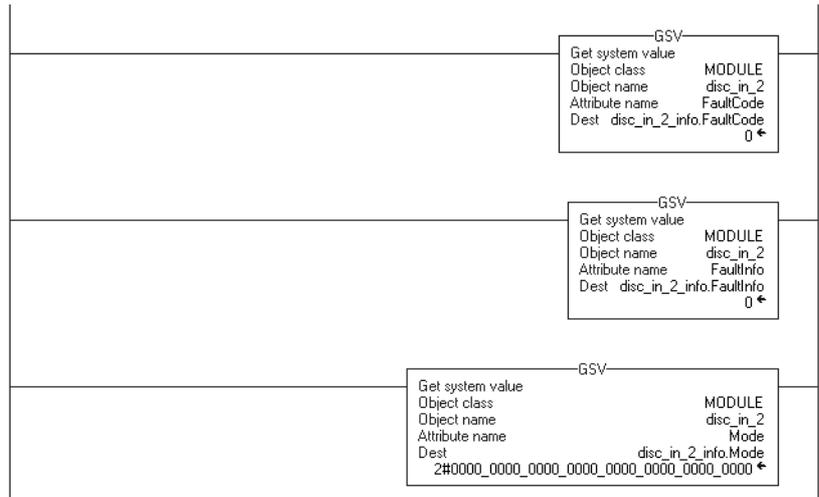
## Esempio di programmazione GSV/SSV

### Ottenere informazioni sugli errori

Nei seguenti esempi si utilizzano le istruzioni GSV per ottenere informazioni sugli errori.

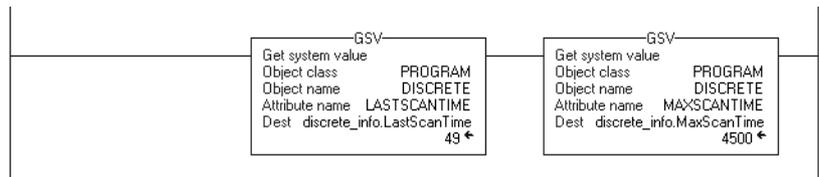
#### Ottenere informazioni sugli errori I/O

In questo esempio si mostra come ottenere informazioni sugli errori dal *disc\_in\_2* del modulo I/O ed inserire i dati in una struttura definita dall'utente *disc\_in\_2\_info*.



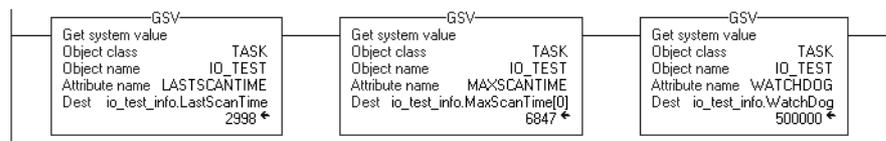
#### Ottenere informazioni sullo stato di un programma

In questo esempio si mostra come ottenere informazioni sullo stato di un programma *discrete* ed inserire i dati in una struttura definita dall'utente *discrete\_info*.



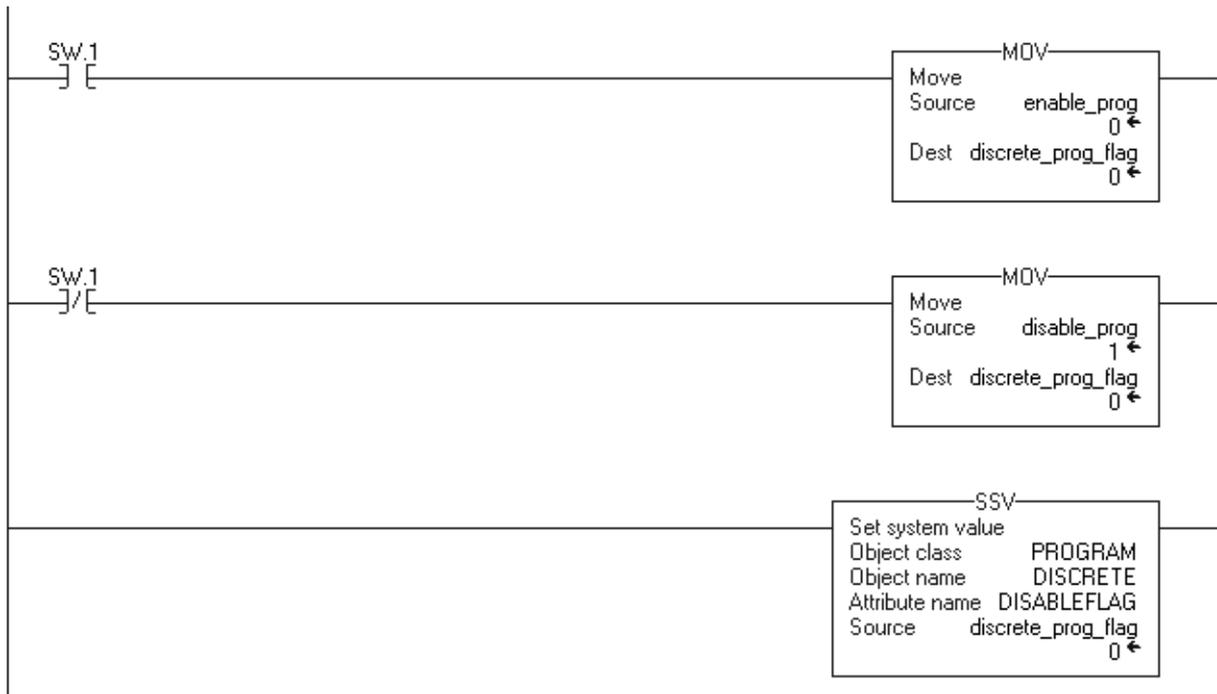
#### Ottenere informazioni sullo stato di un task

In questo esempio si mostra come ottenere informazioni sullo stato di un task *io\_test* ed inserire i dati in una struttura definita dall'utente *io\_test\_info*.



### Impostazione degli indicatori di abilitazione e disabilitazione

Nel seguente esempio si utilizza l'istruzione SSV per abilitare o disabilitare un programma. È possibile utilizzare questo metodo anche per abilitare o disabilitare un modulo I/O; una soluzione di programma simile a quella che prevede l'uso dei bit di inibizione con un processore PLC-5.



In base allo stato di *SW.1*, inserire il valore appropriato nell'attributo *disableflag* del programma *discrete*.

## Istruzioni di confronto (CMP, EQU, GEQ, GRT, LEQ, LES, LIM, MEQ, NEQ)

### Introduzione

Le istruzioni di confronto permettono di confrontare dei valori utilizzando un'espressione o un'istruzione di confronto specifica.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
confrontare dei valori con un'espressione	CMP	4-2
conoscere se due valori sono uguali	EQU	4-6
conoscere se un valore è maggiore o uguale ad un secondo valore	GEQ	4-8
conoscere se un valore è maggiore di un secondo valore	GRT	4-10
conoscere se un valore è minore o uguale ad un secondo valore	LEQ	4-12
conoscere se un valore è minore di un secondo valore	LES	4-14
conoscere se un valore è intermedio a due altri valori	LIM	4-16
passare due valori attraverso una maschera e vedere se sono uguali	MEQ	4-19
conoscere se un valore è diverso da un secondo valore	NEQ	4-22

È possibile confrontare valori di tipo diverso, ad esempio a virgola mobile ed interi.

I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente e richiede meno memoria se tutti gli operandi dell'istruzione utilizzano il medesimo tipo di dati ottimale, in genere DINT o REAL.

## Confronto (CMP)

L'istruzione CMP è un'istruzione di ingresso.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Expression	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	un'espressione consiste in tag e/o valori immediati separati da operatori

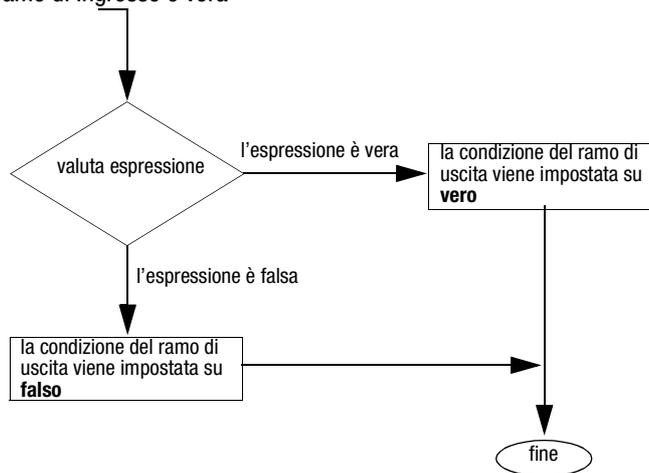
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione CMP esegue un confronto delle operazioni aritmetiche specificate nell'espressione. Definire l'espressione utilizzando operatori, tag e valori immediati. Utilizzare delle parentesi ( ) per definire parti di espressioni più complesse.

L'esecuzione di un'istruzione CMP è leggermente più lenta ed utilizza più memoria rispetto all'esecuzione di altre istruzioni di confronto. Il vantaggio di un'istruzione CMP consiste nel fatto che consente di inserire espressioni complesse in una istruzione.

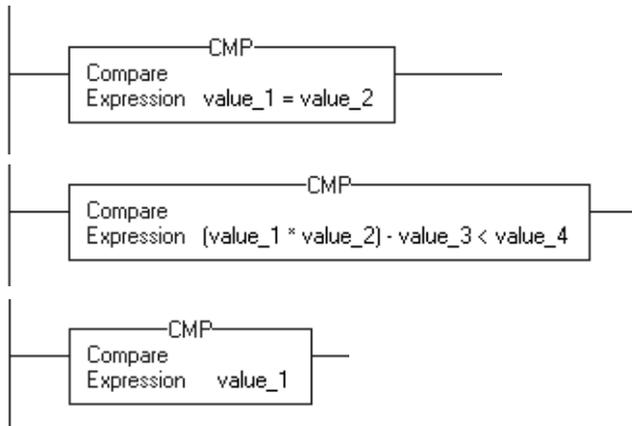
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di CMP:**

Se l'istruzione CMP trova che l'espressione è vera, la condizione del ramo di uscita viene impostata a vera.

Se si inserisce un'espressione senza un operatore di confronto, ad esempio  $value_1 + value_2$  oppure  $value_1$ , l'istruzione valuta l'espressione nel modo seguente:

Se l'espressione è:	La condizione del ramo di uscita è impostata su:
diversa zero	vera
zero	falsa

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>CMP ( expression ) ;</code>
testo ASCII	<code>CMP ( expression ) ;</code>

**Istruzioni correlate:** CPT, FAL, FSC

Le espressioni delle istruzioni CMP vanno programmate nello stesso modo delle espressioni delle istruzioni FSC. Per informazioni su operatori validi, formato ed ordine delle operazioni comuni ad entrambi le istruzioni, utilizzare le seguenti sezioni.

### Operatori validi

Operatore:	Descrizione:	Ottimale:	Operatore:	Descrizione:	Ottimale:
+	somma	DINT, REAL	COS	coseno	REAL
-	sottrai/nega	DINT, REAL	DEG	da radianti a gradi	DINT, REAL
*	moltiplica	DINT, REAL	FRD	da BCD a intero	DINT
/	dividi	DINT, REAL	LN	logaritmo naturale	REAL
=	uguale	DINT, REAL	LOG	logaritmo in base 10	REAL
<	minore di	DINT, REAL	MOD	dividi modulo	DINT, REAL
<=	minore di o uguale	DINT, REAL	NOT	complemento di bit	DINT
>	maggiore di	DINT, REAL	OR	OR di bit	DINT
>=	maggiore di o uguale a	DINT, REAL	RAD	da gradi a radianti	DINT, REAL
<>	diverso	DINT, REAL	SIN	seno	REAL
**	potenza (x alla y)	DINT, REAL	SQR	radice quadrata	DINT, REAL
ABS	valore assoluto	DINT, REAL	TAN	tangente	REAL
ACS	arcocoseno	REAL	TOD	da intero a BCD	DINT
AND	AND di bit	DINT	TRN	tronca	DINT, REAL
ASN	arcoseno	REAL	XOR	OR esclusivo di bit	DINT
ATN	arcotangente	REAL			

### Formattazione delle espressioni

Per ciascun operatore utilizzato in un'espressione, è necessario fornire uno o due operandi (tag o valori immediati). Per la formattazione degli operatori e degli operandi di un'espressione, utilizzare la seguente tabella:

Per operatori con:	Utilizzare questo formato:	Esempi:
un operando	operatore(operando)	<i>ABS(tag_a)</i>
due operandi	operando_a operatore operando_b	<ul style="list-style-type: none"> <li>• <i>tag_b + 5</i></li> <li>• <i>tag_c AND tag_d</i></li> <li>• <i>(tag_e ** 2) MOD (tag_f / tag_g)</i></li> </ul>

## Determinazione dell'ordine delle operazioni

Le operazioni che vengono scritte nell'espressione sono eseguite dall'istruzione in un determinato ordine, non necessariamente nell'ordine con cui sono state scritte. È possibile forzare l'ordine delle operazioni raggruppando i termini in parentesi e forzando l'istruzione ad eseguire un'operazione all'interno delle parentesi prima delle altre operazioni.

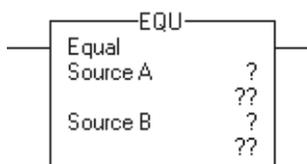
Operazioni di ordine uguale vengono eseguite da sinistra a destra.

<b>Ordine:</b>	<b>Funzionamento:</b>
1.	()
2.	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3.	**
4.	– (nega), NOT
5.	*, /, MOD
6.	<, <=, >, >=, =
7.	– (sottrai), +
8.	AND
9.	XOR
10.	OR

## Uguale a (EQU)

L'istruzione EQU è un'istruzione di ingresso.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source A

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

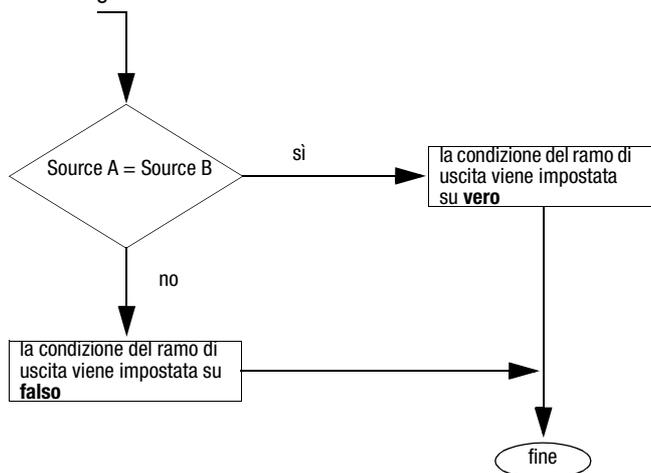
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione EQU verifica se Source A è uguale a Source B.

I valori REAL raramente sono assolutamente uguali. Se si desidera determinare l'uguaglianza di due valori REAL, utilizzare l'istruzione LIM.

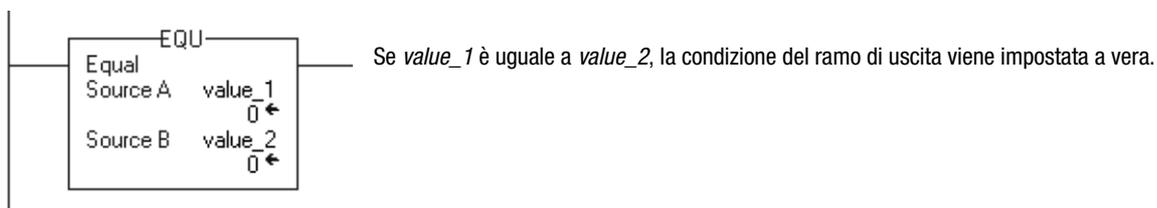
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di EQU:****Altri formati:****Formato:****Sintassi:**

testo neutro

 $\text{EQU}(\text{source\_A}, \text{source\_B}) ;$ 

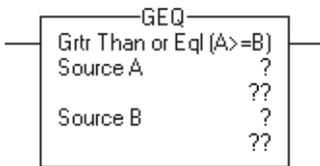
testo ASCII

 $\text{EQU}(\text{source\_A}, \text{source\_B}) ;$ **Istruzioni correlate:** CMP, GEQ, LEQ, MEQ, NEQ

### Maggiore di o Uguale a (GEQ)

L'istruzione GEQ è un'istruzione di ingresso.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source A

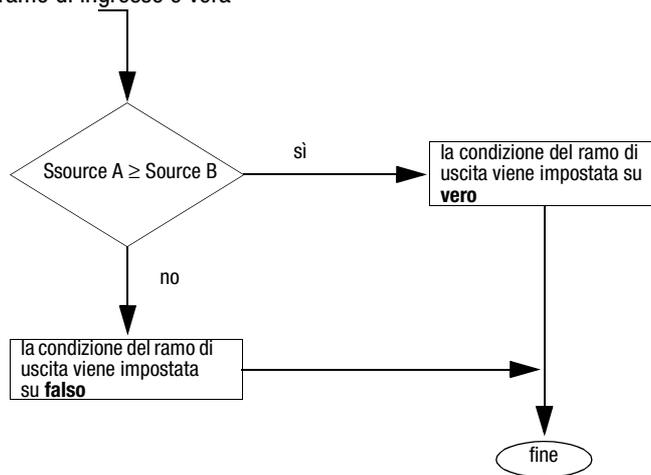
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione GEQ verifica se Source A è maggiore di o uguale a Source B.

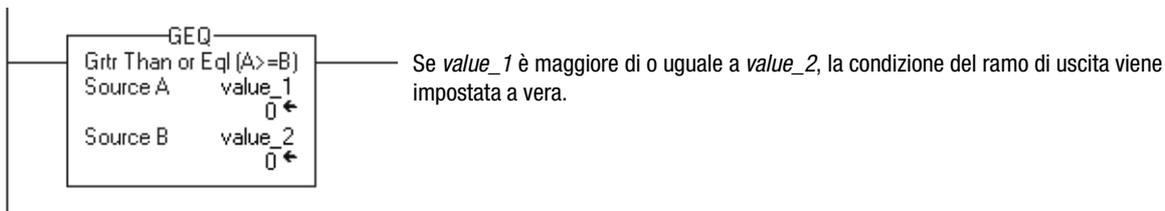
**Esecuzione:**

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di GEQ:****Altri formati:****Formato:****Sintassi:**

testo neutro

GEQ(*source\_A*,*source\_B*);

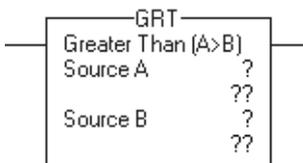
testo ASCII

GEQ(*source\_A*,*source\_B*);**Istruzioni correlate:** CMP, EQU, LEQ, MEQ, NEQ

### Maggiore di (GRT)

L'istruzione GRT è un'istruzione di ingresso.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source A

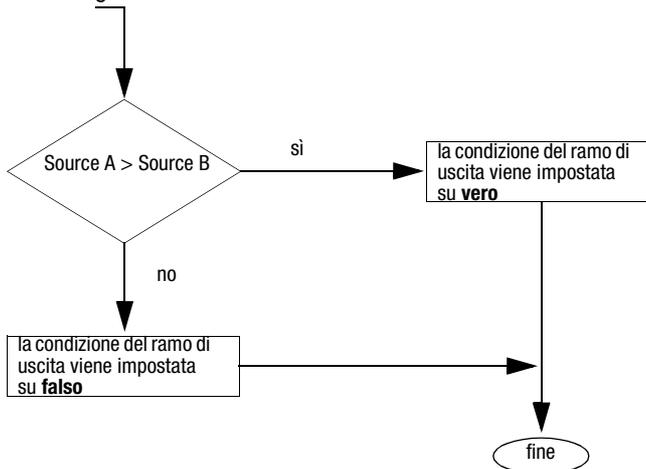
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione GRT verifica se Source A è maggiore di Source B.

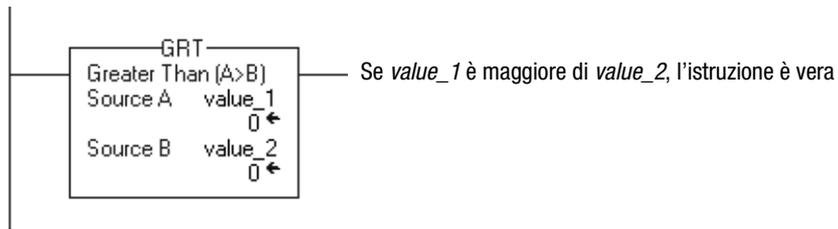
**Esecuzione:**

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di GRT:****Altri formati:**

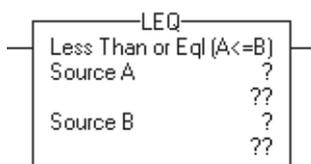
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<code>GRT ( <i>source_A</i>, <i>source_B</i> ) ;</code>
testo ASCII	<code>GRT ( <i>source_A</i>, <i>source_B</i> ) ;</code>

**Istruzioni correlate:** CMP, LES

### Minore di o Uguale a (LEQ)

L'istruzione LEQ è un'istruzione di ingresso.

#### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source A

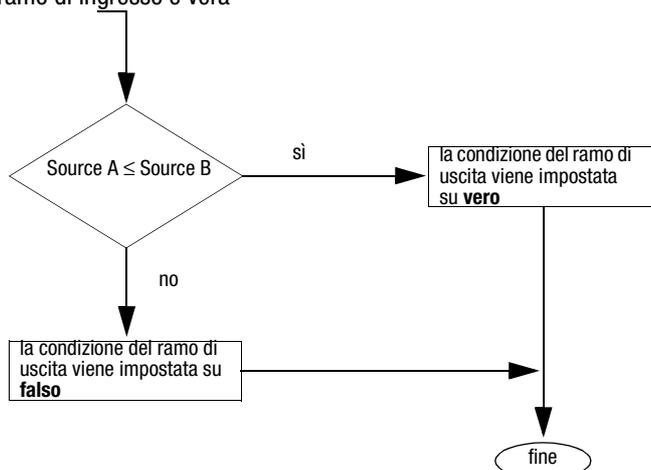
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione LEQ verifica se Source A è minore di o uguale a Source B.

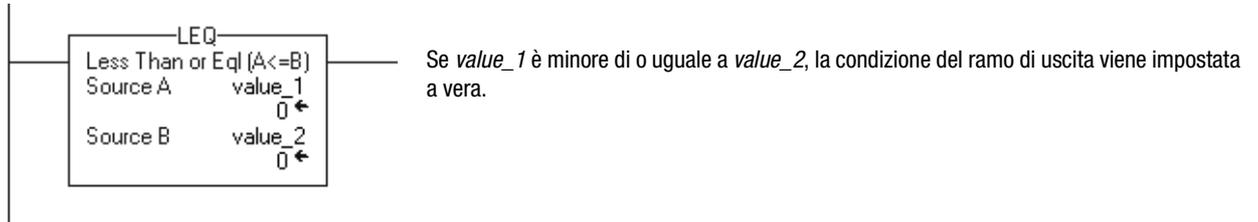
#### Esecuzione:

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di LEQ:****Altri formati:**

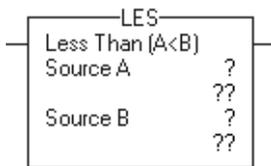
Formato:	Sintassi:
testo neutro	LEQ ( <i>source_A</i> , <i>source_B</i> ) ;
testo ASCII	LEQ ( <i>source_A</i> , <i>source_B</i> ) ;

**Istruzioni correlate:** CMP, EQU, GEQ, MEQ, NEQ

## Minore di (LES)

L'istruzione LES è un'istruzione di ingresso.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source A

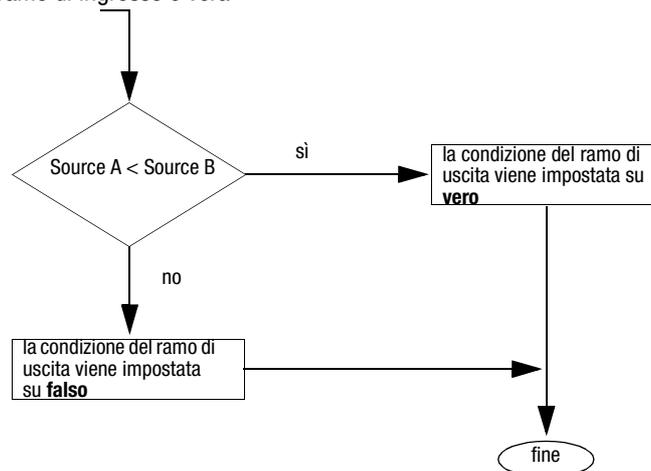
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione LES verifica se Source A è minore di Source B.

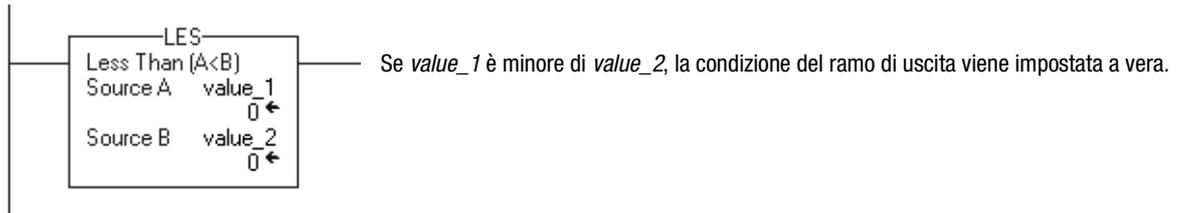
### Esecuzione:

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di LES:****Altri formati:**

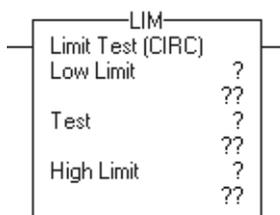
Formato:	Sintassi:
testo neutro	LES ( <i>source_A</i> , <i>source_B</i> ) ;
testo ASCII	LES ( <i>source_A</i> , <i>source_B</i> ) ;

**Istruzioni correlate:** CMP, GRT

## Limite (LIM)

L'istruzione LIM è un'istruzione di ingresso.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Low Limit	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	il valore del limite inferiore
Test	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	il valore da confrontare
High Limit	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	il valore del limite superiore

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

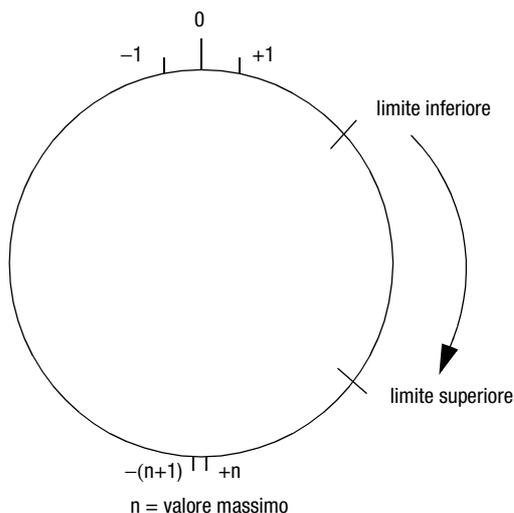
**Descrizione:** L'istruzione LIM verifica se il valore Test è all'interno dell'intervallo compreso tra il limite inferiore ed il limite superiore.

Gli interi con segno, quando il bit più significativo è ad 1, passano dal massimo positivo al massimo negativo. Per esempio, negli interi a 16 bit (tipo INT), il numero intero positivo massimo è 32.767, rappresentato in forma esadecimale come 16#7FFF (i bit da 0 a 14 sono tutti ad 1). Se questo numero aumenta di una unità, il risultato sarà di 16#8000 (il bit 15 è impostato). Nel caso di interi con segno, il numero esadecimale 16#8000 è uguale a -32.768 decimale. Se si continua ad incrementare fino a quando tutti i 16 bit sono ad 1, si arriva a 16#FFFF, che è pari a -1 decimale.

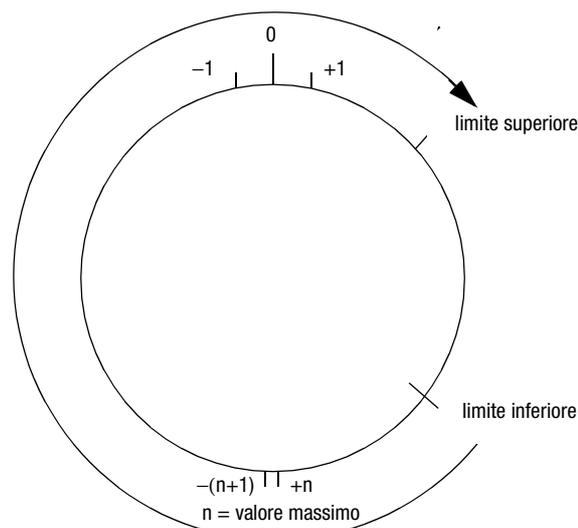
Questo processo può essere rappresentato come un diagramma circolare (vedere le figure seguenti). L'istruzione LIM inizia dal limite inferiore e cresce in senso orario fino a raggiungere il limite superiore. Qualsiasi valore di Test nell'intervallo in senso orario compreso tra il limite inferiore ed il limite superiore imposta la condizione del ramo di uscita a vera. Qualsiasi valore di Test nell'intervallo in senso orario compreso tra il limite superiore ed il limite inferiore imposta la condizione del ramo di uscita a falsa.

**Limite inferiore  $\leq$  Limite superiore**

L'istruzione è vera se il valore da Test è uguale a oppure è compreso tra il limite inferiore e superiore

**Limite inferiore  $\geq$  Limite superiore**

L'istruzione è vera se il valore Test è uguale a oppure è esterno al limite inferiore e superiore

**Esecuzione:****Condizione:****Azione:**

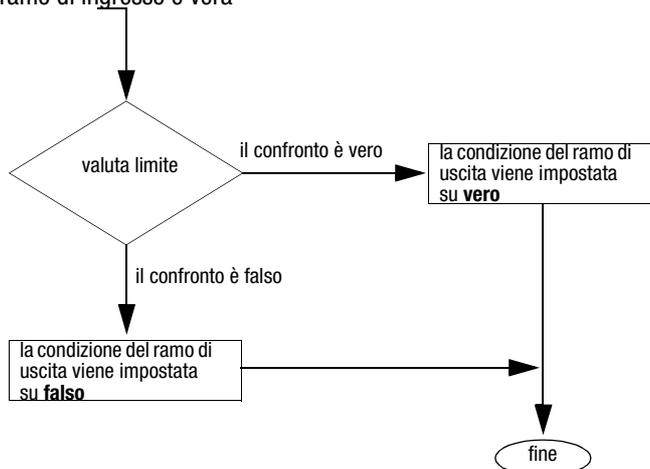
prescazione

La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è falsa

La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è vera



Se il Limite inferiore:	Ed il valore da testare è:	La condizione del ramo di uscita è:
$\leq$ Limite superiore	uguale ai limiti o entro di questi	vera
	diverso dai limiti o al di fuori di questi	falsa
$\geq$ Limite superiore	uguale ai limiti o al di fuori di questi	vera
	diverso dai limiti o entro questi	falsa

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di LIM:**

**esempio 1**



**Limite inferiore ≤ Limite superiore:**  
 Quando  $0 \leq \text{value} \leq 100$ , *light\_1* si accende.

**esempio 2**



**Limite inferiore ≥ Limite superiore:**  
 Quando  $\text{value} \geq 0$  o  $\text{value} \leq -100$ , *light\_1* si accende.

**Altri formati:**

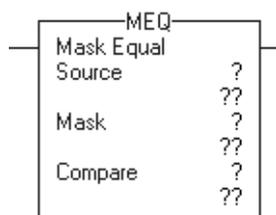
Formato:	Sintassi:
testo neutro	<code>LIM(<i>limite_inferiore</i>,<i>test</i>,<i>limite_superiore</i>);</code>
testo ASCII	<code>LIM(<i>limite_inferiore</i>,<i>test</i>,<i>limite_superiore</i>);</code>

**Istruzioni correlate:** CMP

## Maschera uguale a (MEQ)

L'istruzione MEQ è un'istruzione di ingresso.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b>	immediato tag	valore da confrontare con Confronta
			Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.
Mask	SINT INT <b>DINT</b>	immediato tag	quali bit bloccare o fare passare
			Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.
Compare	SINT INT <b>DINT</b>	immediato tag	valore da confrontare con Source
			Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione MEQ fa passare i valori di Source e Compare attraverso una maschera e confronta i risultati.

Un "1" nella maschera significa che il bit dati è passato. Uno "0" nella maschera significa che il bit dati è bloccato. In genere i valori Source, Mask e Compare sono tutti dello stesso tipo di dati.

Se si mischiano tipi di dati interi, l'istruzione riempie con degli 0 i bit più significativi dei tipi di dati interi più piccoli in modo che questi abbiano la stessa dimensione dei tipi di dati più grandi.

### Immissione di un valore di maschera immediato

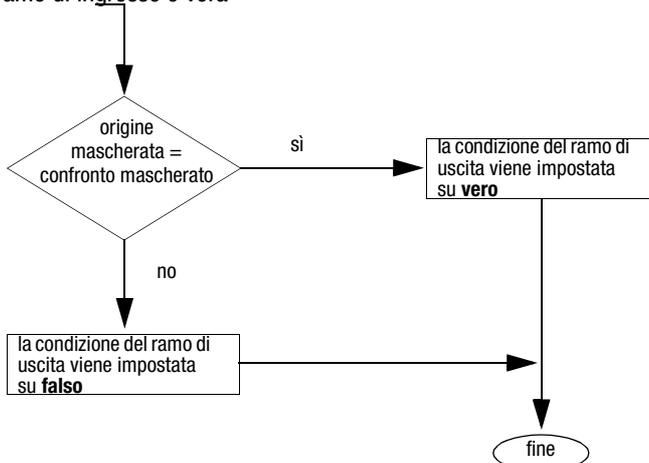
Quando si immette una maschera, il software di programmazione imposta automaticamente valori decimali. Se si desidera immettere una maschera utilizzando un altro formato, fare precedere il prefisso corretto al valore.

Prefisso:	Descrizione:
16#	Esadecimale per esempio; 16#0F0F
8#	Ottale per esempio; 8#16
2#	Binario per esempio; 2#00110011

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è vera



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di MEQ:**

esempio 1



<i>value_1</i>	0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1	<i>value_2</i>	0 1 0 1 0 1 0 1 0 1 1 1 1 1 0 0 0 0
<i>mask_1</i>	1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0	<i>mask_1</i>	1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
<i>value_1 mascherato</i>	0 1 0 1 0 1 0 1 0 1 1 1 1 1 x x x x	<i>value_2 mascherato</i>	0 1 0 1 0 1 0 1 0 1 1 1 1 1 x x x x

*Value\_1* mascherato è uguale a *value\_2* mascherato, quindi *light\_1* si accende. Uno 0 nella maschera impedisce all'istruzione di comparare quel bit (indicato dalla x nell'esempio).

## esempio 2



<i>value_1</i>	010101010111111111	<i>value_2</i>	0101010101111110000
<i>mask_1</i>	0000000000000001111	<i>mask_1</i>	0000000000000001111
<i>value_1</i> mascherato	x x x x x x x x x x 1 1 1 1	<i>Value_2</i> mascherato	x x x x x x x x x x 0 0 0 0

*Value\_1* mascherato è diverso da *value\_2* mascherato, quindi *light\_1* si spegne. Uno 0 nella maschera impedisce all'istruzione di comparare quel bit (indicato dalla x nell'esempio).

**Altri formati:****Formato:****Sintassi:**

testo neutro

MEQ(*source*, *mask*, *compare*);

testo ASCII

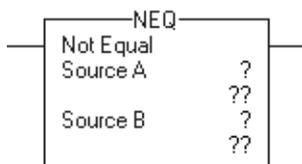
MEQ *source*, *mask*, *compare*

**Istruzioni correlate:** CMP, EQU, GEQ, LEQ, MEQ, NEQ

## Diverso da (NEQ)

L'istruzione NEQ è un'istruzione di ingresso.

### Operandi:

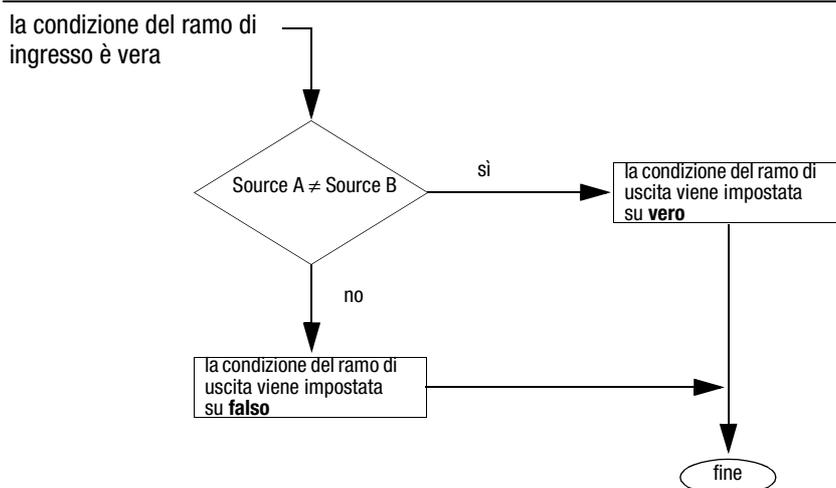


Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da confrontare con Source A

**Descrizione:** L'istruzione NEQ verifica se due valori sono diversi.

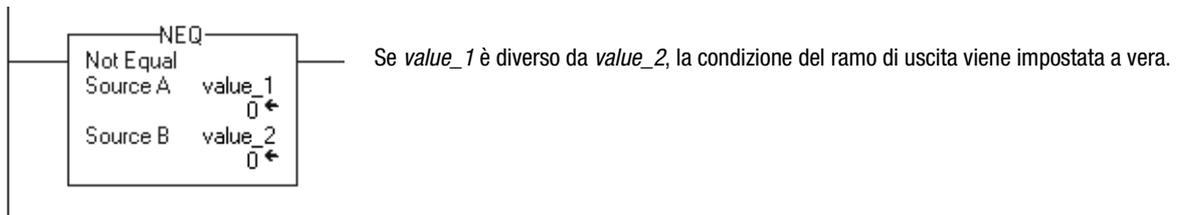
### Esecuzione:

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di NEQ:****Altri formati:**

Formato:	Sintassi:
testo neutro	<code>NEQ( source_A, source_B ) ;</code>
testo ASCII	<code>NEQ( source_A, source_B ) ;</code>

**Istruzioni correlate:** CMP, EQU, LEQ, GEQ, MEQ

**Nota:**

## Istruzioni di Calcolo/Matematiche (CPT, ADD, SUB, MUL, DIV, MOD, SQR, NEG, ABS)

### Introduzione

Le istruzioni di calcolo/matematiche eseguono le operazioni aritmetiche utilizzando un'espressione o una specifica istruzione aritmetica.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
calcolare un'espressione	CPT	5-2
sommare due valori	ADD	5-6
sottrarre due valori	SUB	5-8
moltiplicare due valori	MUL	5-10
dividere due valori	DIV	5-12
determinare il resto dopo la divisione di un valore per un altro	MOD	5-14
calcolare la radice quadrata di un valore	SQR	5-16
calcolare segno ad un valore	NEG	5-18
calcolare il valore assoluto di un valore	ABS	5-19

È possibile usare tipi di dati diversi, ma si potrebbero verificare errori di precisione e di arrotondamento ed inoltre l'istruzione potrebbe impiegare più tempo per essere eseguita. Controllare il bit S:V per verificare se il risultato è stato troncato.

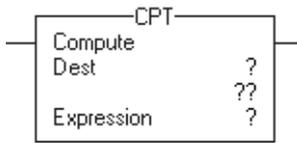
I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente e richiede meno memoria se tutti gli operandi dell'istruzione utilizzano il medesimo tipo di dati ottimali, in genere DINT o REAL.

Un'istruzione di calcolo/matematica viene eseguita ogni volta che l'istruzione viene scandita e per tutto il tempo che la condizione del ramo di entrata rimane vera. Se si desidera che l'espressione venga calcolata solo una volta, utilizzare un'istruzione one-shot per attivare l'istruzione.

## Calcola (CPT)

L'istruzione CPT è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato
Expression	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	un'espressione formata da tag e/o valori immediati separati da operatori

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione CPT esegue le operazioni aritmetiche definite nell'espressione. Quando è abilitata, l'istruzione CPT calcola l'espressione ed inserisce il risultato nella Destinazione.

L'esecuzione di un'istruzione CPT è leggermente più lenta ed utilizza più memoria rispetto all'esecuzione di altre istruzioni di calcolo/matematiche. Il vantaggio di un'istruzione CPT consiste nel fatto che consente di inserire espressioni complesse in una istruzione.

**TI** La lunghezza di un'espressione *non ha limite*.

### Esecu

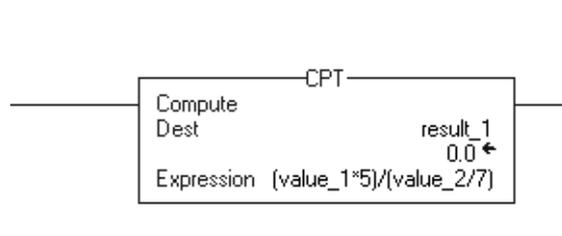
Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione calcola l'Espressione ed inserisce il risultato nella Destinazione. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

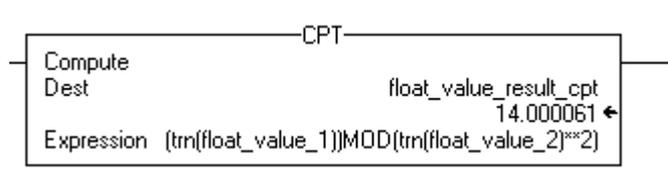
### Esempi di CPT:

#### esempio 1



Se è abilitata, l'istruzione CPT calcola *value\_1* moltiplicato 5, divide questo risultato per il risultato del *value\_2* diviso 7 ed inserisce il risultato finale in *result\_1*.

## esempio 2



Se è abilitata, l'istruzione CPT tronca *float\_value\_1* e *float\_value\_2*, eleva il valore troncato *float\_value\_2* alla potenza di due, divide il valore troncato *float\_value\_1* per questo risultato e memorizza il resto della divisione in *float\_value\_result\_cpt*.

**Altri formati:****Formato:****Sintassi:**

testo neutro	<code>CPT(destination, expression);</code>
testo ASCII	<code>CPT destination expression</code>

**Istruzioni correlate:** CMP, FAL, FSC

Le espressioni delle istruzioni CPT vanno programmate nello stesso modo delle espressioni delle istruzioni FAL. Per informazioni su operatori validi, formato ed ordine delle operazioni comuni ad entrambi le istruzioni, utilizzare le seguenti sezioni.

**Operatori validi**

Operatore:	Descrizione:	Ottimale:
+	somma	DINT, REAL
-	sottrai/nega	DINT, REAL
*	moltiplica	DINT, REAL
/	dividi	DINT, REAL
**	esponente (x alla y)	DINT, REAL
ABS	valore assoluto	DINT, REAL
ACS	arcocoseno	REAL
AND	AND di bit	DINT
ASN	arcoseno	REAL
ATN	arcotangente	REAL
COS	coseno	REAL
DEG	da radianti a gradi	DINT, REAL
FRD	da BCD a intero	DINT

Operatore:	Descrizione:	Ottimale:
LN	logaritmo naturale	REAL
LOG	logaritmo in base 10	REAL
MOD	dividi modulo	DINT, REAL
NOT	complemento di bit	DINT
OR	OR di bit	DINT
RAD	da gradi a radianti	DINT, REAL
SIN	seno	REAL
SQR	radice quadrata	DINT, REAL
TAN	tangente	REAL
TOD	da intero a BCD	DINT
TRN	tronca	DINT, REAL
XOR	OR di bit esclusivo	DINT

### Formattazione delle espressioni

Per ciascun operatore utilizzato in un'espressione, è necessario fornire uno o due operandi (tag o valori immediati). Per la formatazione degli operatori e degli operandi di un'espressione, utilizzare la seguente tabella:

<b>Per operatori con:</b>	<b>Utilizzare questo formato:</b>	<b>Esempi:</b>
un operando	operatore(operando)	$ABS(tag\_a)$
due operandi	operando_a operatore operando_b	<ul style="list-style-type: none"><li>• <math>tag\_b + 5</math></li><li>• <math>tag\_c \text{ AND } tag\_d</math></li><li>• <math>(tag\_e ** 2) \text{ MOD } (tag\_f / tag\_g)</math></li></ul>

## Determinazione dell'ordine delle operazioni

Le operazioni che vengono scritte nell'espressione sono eseguite dall'istruzione in un determinato ordine, non necessariamente nell'ordine con cui sono state scritte. È possibile forzare l'ordine delle operazioni raggruppando i termini all'interno di parentesi e forzando l'istruzione ad eseguire un'operazione all'interno delle parentesi prima delle altre operazioni.

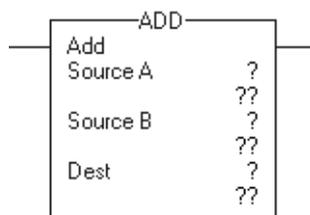
Operazioni di ordine uguale vengono eseguite da sinistra a destra.

<b>Ordine:</b>	<b>Funzionamento:</b>
1.	()
2.	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3.	**
4.	– (nega), NOT
5.	*, /, MOD
6.	– (sottrai), +
7.	AND
8.	XOR
9.	OR

## Somma (ADD)

L'istruzione ADD è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da sommare a Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da sommare a Source A
Destinazione	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione ADD somma Source A a Source B ed inserisce il risultato nella Destination.

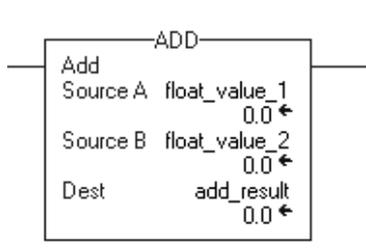
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Destination = Source A + Source B La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di ADD:



Se è abilitata, l'istruzione ADD somma *float\_value\_1* e *float\_value\_2* ed inserisce il risultato in *add\_result*.

**Altri formati:**

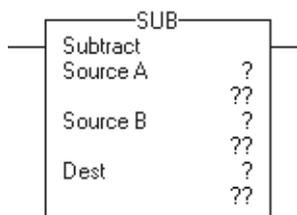
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>ADD( source_A, source_B, destination );</i>
testo ASCII	<i>ADD source_A source_B destination</i>

**Istruzioni correlate:** CPT, DIV, MUL, SUB

## Sottrai (SUB)

L'istruzione SUB è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da cui sottrarre Source B
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da sottrarre a Source A
Destinazione	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione SUB sottrae Source B da Source A ed inserisce il risultato nella destinazione.

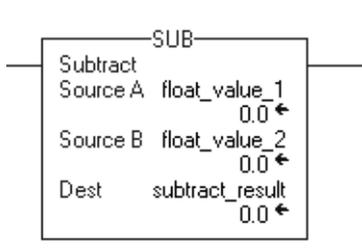
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Destination = Source A – Source B La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di SUB:



Se è abilitata, l'istruzione SUB sottrae *float\_value\_2* da *float\_value\_1* ed inserisce il risultato in *subtract\_result*.

**Altri formati:**

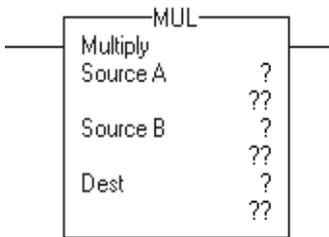
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>SUB ( source_A , source_B , destination ) ;</i>
testo ASCII	<i>SUB source_A source_B destination</i>

**Istruzioni correlate:** CPT, ADD DIV, MUL

## Moltiplica (MUL)

L'istruzione MUL è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore del moltiplicando
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore del moltiplicatore
Destinazione	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione MUL moltiplica Source A per Source B ed inserisce il risultato nella destinazione.

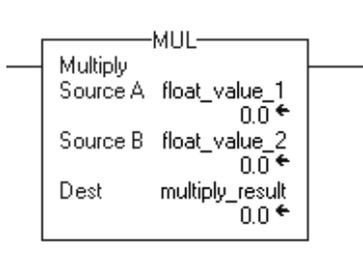
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Destination = Source A * Source B La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di MUL:



Se è abilitata, l'istruzione MUL moltiplica *float\_value\_1* per *float\_value\_2* ed inserisce il risultato in *multiply\_result*.

**Altri formati:**

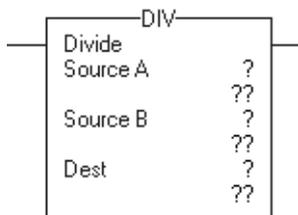
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	$MUL(source\_A, source\_B, destination);$
testo ASCII	$MUL source\_A source\_B destination$

**Istruzioni correlate:** CPT, ADD, DIV, SUB

## Dividi (DIV)

L'istruzione DIV è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore del dividendo
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore del divisore
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione DIV divide Source A per Source B ed inserisce il risultato nella destinazione. Se la destinazione *non* è un valore REAL, l'istruzione gestisce la parte frazionaria del risultato nel modo seguente:

Se Source A:	La parte frazionaria del risultato:	Esempio:
e Source B <i>non</i> sono REAL	viene troncata	Source A    DINT    5 Source B    DINT    3 Destination DINT    1
o Source B è un valore REAL	arrotonda	Source A    REAL    5.0 Source B    DINT    3 Destination DINT    2

Se Source B (il divisore) è zero, la destinazione viene posta uguale a Source A (il dividendo) e viene registrato un errore minore di overflow aritmetico. Monitorando il bit di errore minore (S:MINOR) è possibile rilevare un possibile caso di divisione per zero.

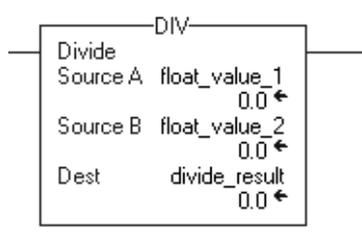
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Destination = Source A / Source B La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di DIV:**



Se è abilitata, l'istruzione DIV divide *float\_value\_1* per *float\_value\_2* ed inserisce il risultato in *divide\_result*.

**Altri formati:**

**Formato:**

**Sintassi:**

---

testo neutro	<code>DIV(<i>source_A</i>,<i>source_B</i>,<i>destination</i>);</code>
--------------	---

---

testo ASCII	<code>DIV <i>source_A</i> <i>source_B</i> <i>destination</i></code>
-------------	---

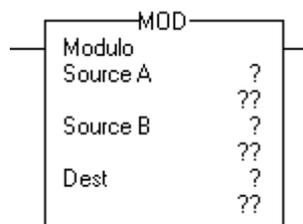
---

**Istruzioni correlate:** CPT, ADD, MUL, SUB

## Modulo (MOD)

L'istruzione MOD è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore del dividendo
Source B	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore del divisore
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione MOD divide Source A per Source B ed inserisce il resto in Destination.

Se Source B (il divisore) è zero, viene registrato un errore minore di overflow aritmetico e la destinazione è posta come indicato nella seguente tabella:

Se il divisore è zero e la destinazione è una tag di questo tipo:	La destinazione è impostata a:
SINT, INT o DINT	zero
REAL	infinito

**TI** Esaminando il bit di errore minore (S:MINOR) è possibile rilevare un possibile caso di divisione per zero.

**Esecu**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	$Destination = Source A - (TRN ( Source A / Source B ) * Source B)$ La condizione del ramo di uscita viene impostata su vero.

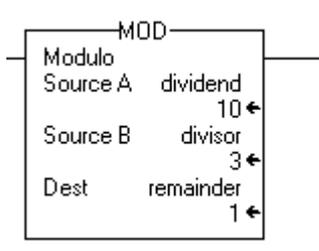
**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:****Si verifica un errore  
minore se:****Tipo errore:****Codice errore:**

il divisore è zero

4

4

**Esempio di MOD:**

Se è abilitata, l'istruzione MOD divide *dividend* per *divisor* ed inserisce il resto in *remainder*. In questo esempio, la divisione ha un resto di uno.

**Altri formati:****Formato:****Sintassi:**

testo neutro

MOD(*source\_A*, *source\_B*, *destination*);

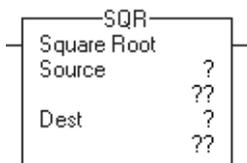
testo ASCII

MOD *source\_A* *source\_B* *destination***Istruzioni correlate:** CPT, ADD, MUL, SUB, DIV

## Radice quadrata (SQR)

L'istruzione SQR è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	trova la radice quadrata di questo valore
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione SQR calcola la radice quadrata di Source ed inserisce il risultato in Destination. Se la destinazione *non* è un valore REAL, l'istruzione gestisce la parte frazionaria del risultato nel modo seguente:

Se Source è:	La parte frazionaria del risultato:	Esempio:
<i>non</i> un valore REAL	viene troncata	Source     DINT     3 ----- Destination     DINT     1
un valore REAL	arrotonda	Source     REAL     3.0 ----- Destination     DINT     2

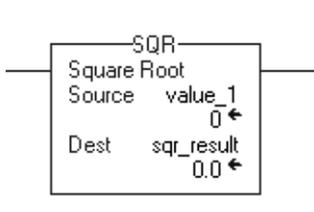
Se Source è negativa, l'istruzione, prima di calcolare la radice quadrata, cambia il valore assoluto di Source.

### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	$Destination = \sqrt{Source}$ La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di SQR:**

Se è abilitata, l'istruzione SQR calcola la radice quadrata di *value\_1* ed inserisce il risultato in *sqr\_result*.

**Altri formati:**

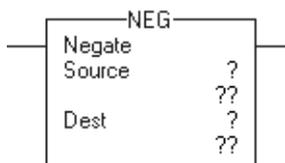
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>SQR(source,destination);</i>
testo ASCII	<i>SQR source destination</i>

**Istruzioni correlate:** CPT, NEG

## Nega (NEG)

L'istruzione NEG è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da negare
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione prima di calcolare la radice quadrata cambia il segno di Source ed inserisce il risultato in Destination. Se si nega un valore negativo, il risultato è positivo. Se si nega un valore positivo, il risultato è negativo.

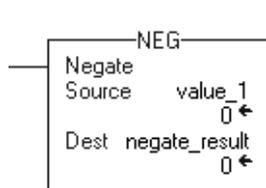
### Esecuzione:

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Destination = 0 – Source La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di NEG:



Se è abilitata, l'istruzione NEG cambia il segno del *value\_1* ed inserisce il risultato in *negate\_result*.

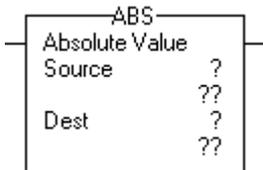
### Altri formati:

Formato:	Sintassi:
testo neutro	NEG( <i>source</i> , <i>destination</i> );
testo ASCII	NEG <i>source destination</i>

**Istruzioni correlate:** CPT, SQR

**Valore assoluto (ABS)**

L'istruzione ABS è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore di cui cambiare il valore assoluto
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

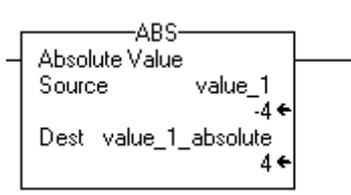
**Descrizione:** L'istruzione ABS cambia il valore assoluto di Source ed inserisce il risultato in Destination.

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Destination =   Source   La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di ABS:**

Se è abilitata, l'istruzione ABS inserisce il valore assoluto di *value\_1* in *value\_1\_absolute*. In questo esempio, il valore assoluto di -4 è +4.

**Altri formati:**

Formato:	Sintassi:
testo neutro	ABS( <i>source</i> , <i>destination</i> );
testo ASCII	ABS <i>source destination</i>

**Istruzioni correlate:** CPT, NEG, SQR, TRN

**Nota:**

## Istruzioni di Spostamento/Logica (MOV, MVM, BTD, CLR, AND, OR, XOR, NOT)

### Introduzione

Le istruzioni di spostamento modificano e spostano bit.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
copiare un valore	MOV	6-2
copiare una determinata parte di un intero	MVM	6-3
spostare bit all'interno di un intero o tra interi	BTD	6-5
azzerare un valore	CLR	6-8

Le istruzioni di logica permettono di eseguire operazioni logiche sui bit.

Se si desidera eseguire una:	Utilizzare questa istruzione:	Vedere pagina:
operazione AND di bit	AND	6-9
operazione OR di bit	OR	6-11
operazione OR esclusivo di bit	XOR	6-13
operazione NOT di bit	NOT	6-15

È possibile usare tipi di dati diversi, ma si potrebbero verificare errori di precisione e di arrotondamento ed inoltre l'istruzione potrebbe impiegare più tempo per essere eseguita. Controllare il bit S:V per verificare se il risultato è stato troncato.

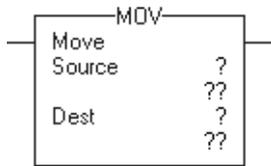
I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente e richiede meno memoria se tutti gli operandi dell'istruzione utilizzano il medesimo tipo di dati ottimali, in genere DINT o REAL.

Un'istruzione di spostamento/logica viene eseguita ogni volta che l'istruzione viene scandita e per tutto il tempo che la condizione del ramo di entrata rimane vera. Se si desidera che l'espressione venga calcolata solo una volta, utilizzare un'istruzione one-shot per attivare l'istruzione spostamento/logica.

## Sposta (MOV)

L'istruzione MOV è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	valore da spostare (copiare)
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione MOV copia Source in Destination. Il valore Source rimane invariato.

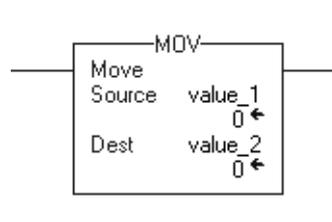
### Esecuzione:

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione copia Source in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di MOV:



Se è abilitata, l'istruzione MOV copia i dati di *value\_1* in *value\_2*.

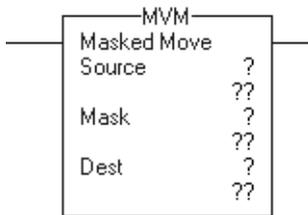
### Altri formati:

Formato:	Sintassi:
testo neutro	<code>MOV(<i>source</i>,<i>destination</i>);</code>
testo ASCII	<code>MOV <i>source</i> <i>destination</i></code>

**Istruzioni correlate:** BTM, CLR, MVM

**Sposta con maschera (MVM)**

L'istruzione MVM è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b>	immediato tag	valore da spostare
			Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.
Mask	SINT INT <b>DINT</b>	immediato tag	quali bit bloccare o passare
			Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.
Destination	SINT INT <b>DINT</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione MVM copia Source in una destinazione e permette che parte dei dati siano mascherati. Il valore Source rimane invariato.

Se è abilitata, l'istruzione MVM utilizza una maschera per fare passare o bloccare i bit dati di Source. Un "1" nella maschera significa che il bit dati viene fatto passare. Uno "0" nella maschera significa che il bit dati viene bloccato.

Se si mischiano tipi di dati interi, l'istruzione riempie con degli 0 i bit superiori dei tipi di dati interi più piccoli in modo che questi abbiano la stessa dimensione dei tipi di dati più grandi.

**Immissione di un valore di maschera immediato**

Quando si immette una maschera, il software di programmazione imposta automaticamente valori decimali. Se si desidera immettere una maschera utilizzando un altro formato, fare precedere il prefisso corretto al valore.

Prefisso:	Descrizione:
16#	Esadecimale per esempio; 16#0F0F
8#	Ottale per esempio; 8#16
2#	Binario per esempio; 2#00110011

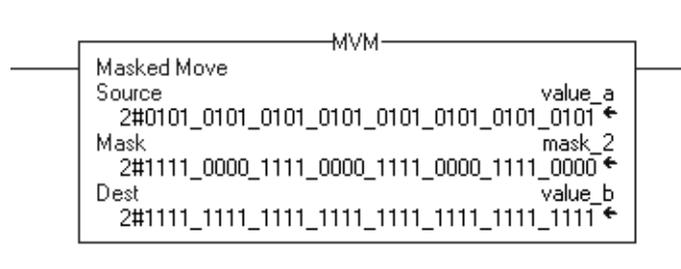
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione fa passare il valore di Source attraverso la maschera e copia i risultati nella destinazione. I bit non mascherati della destinazione rimangono immutati. La condizione del ramo di uscita viene impostata su vero.

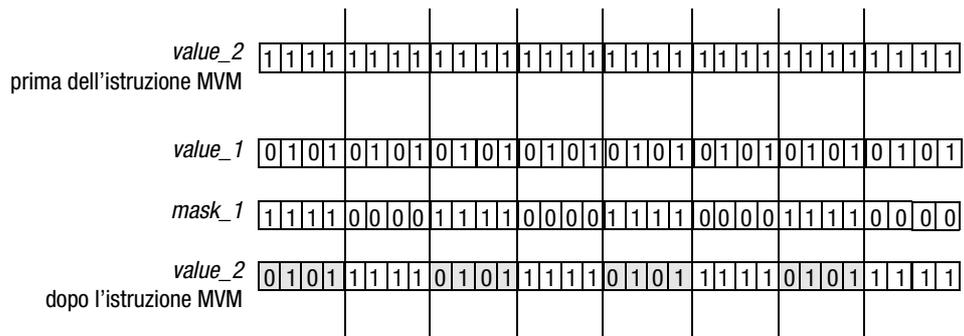
**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di MVM:**



Se è abilitata, l'istruzione MVM copia i dati da *value\_a* in *value\_b*, consentendo allo stesso tempo che i dati siano mascherati (uno 0 maschera i dati di *value\_a*).



Le caselle ombrate mostrano i bit di *value\_2* che sono cambiati.

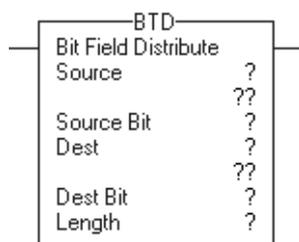
**Altri formati:**

Formato:	Sintassi:
testo neutro	<i>MVM(source,mask,destination);</i>
testo ASCII	<i>MVM source mask destination</i>

**Istruzioni correlate:** BTD, CLR, MOV

**Distribuisci campo di bit (BTD)** L'istruzione BTD è un'istruzione di uscita.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b>	immediato tag	tag contenente i bit da spostare
Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.			
Source bit	DINT	immediato (0-31 DINT) (0-15 INT) (0-7 SINT)	numero del bit (numero del bit più basso) da cui iniziare lo spostamento deve rientrare nell'intervallo valido del tipo di dati di Source
Destination	SINT INT <b>DINT</b>	tag	tag dove spostare i bit
Destination bit	DINT	immediato (0-31 DINT) (0-15 INT) (0-7 SINT)	il numero del bit (numero del bit più basso) da cui iniziare a copiare i bit di Source deve rientrare nell'intervallo valido del tipo di dati della Destination
Lenght	DINT	immediato (1-32)	numero di bit da spostare

**Descrizione:** L'istruzione BTD copia da Source i bit specificati, li sposta nella posizione appropriata e li scrive in Destination. La restante parte della destinazione rimane invariata.

Se è abilitata, l'istruzione BTD copia un gruppo di bit di Source in Destination. Il gruppo di bit è identificato da Source bit (il numero del bit più basso del gruppo) e da Lenght (numero di bit da copiare). Il Destination bit identifica il numero di bit più basso della Destination da cui iniziare. Il valore Source rimane invariato.

Se la lunghezza del campo di bit eccede la Destination, l'istruzione non salva i bit in eccesso. I bit in eccesso non vanno a capo nella parola successiva.

Se si mischiano tipi di dati interi, l'istruzione riempie con degli 0 i bit superiori dei tipi di dati interi più piccoli in modo che questi abbiano la stessa dimensione dei tipi di dati più grandi.

**Esecuzione:**

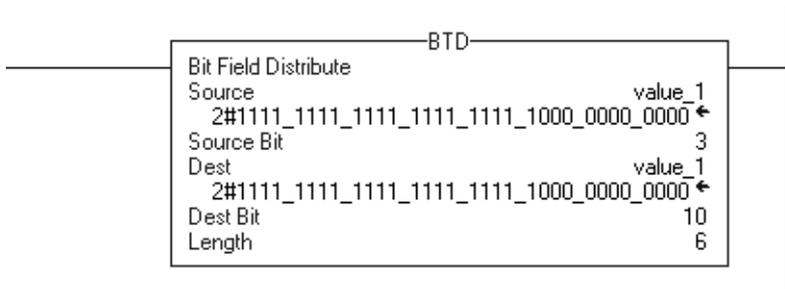
Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione copia e sposta i bit di Source in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

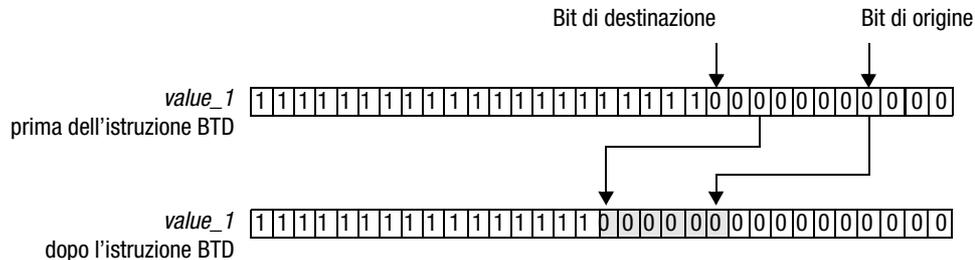
**Condizioni di errore:** nessuna

**Esempio di BTD:**

esempio 1

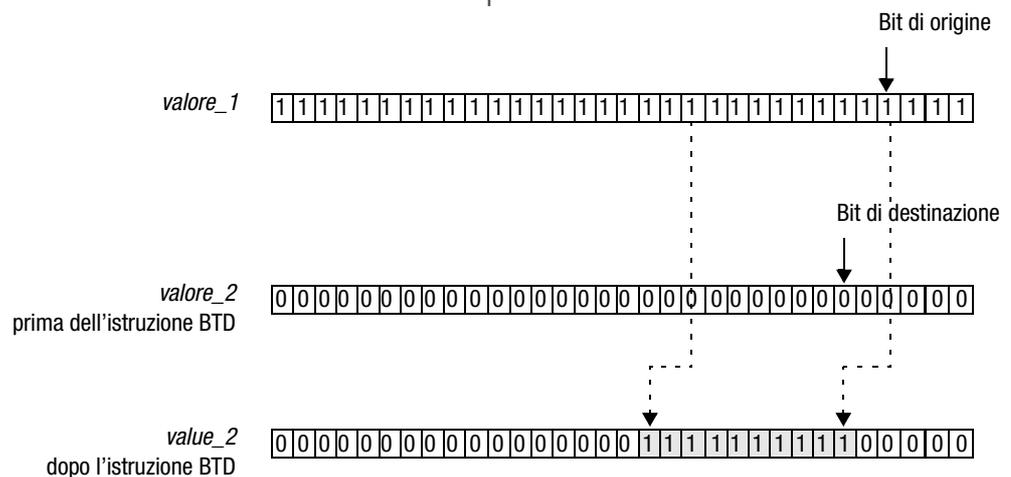
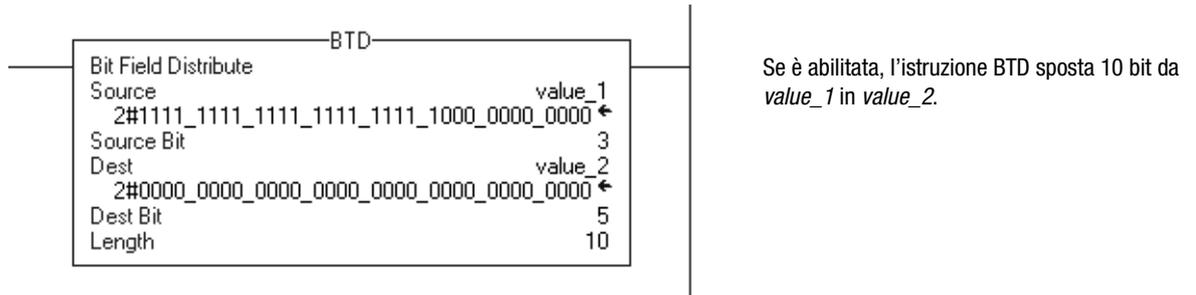


Se è abilitata, l'istruzione BTD sposta i bit all'interno di *value\_1*.



Le caselle ombrate mostrano i bit di *value\_1* che sono cambiati.

## esempio 2



Le caselle ombrate mostrano i bit di *value\_2* che sono cambiati.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>BTD(<i>source</i>, <i>source_bit</i>, <i>destination</i>, <i>destination_bit</i>, <i>length</i>);</code>
testo ASCII	<code>BTD <i>source</i> <i>source_bit</i> <i>destination</i> <i>destination_bit</i> <i>length</i></code>

**Istruzioni correlate:** CLR, MOV, MVM

## Azzera (CLR)

L'istruzione CLR è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Destination	SINT INT DINT REAL	tag	tag da azzera

**Descrizione:** L'istruzione CLR azzera tutti i bit di Destination.

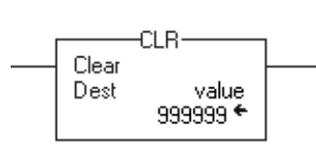
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione azzera la destinazione. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di CLR:



Se è abilitata, l'istruzione CLR imposta a 0 tutti i bit di *value\_1*.

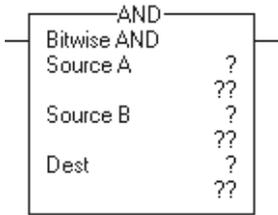
### Altri formati:

Formato:	Sintassi:
testo neutro	<code>CLR(destination);</code>
testo ASCII	<code>CLR destination</code>

**Istruzioni correlate:** MOV

**AND di bit (AND)**

L'istruzione AND è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b>	immediato tag	valore su cui eseguire AND con Source B
Source B	SINT INT <b>DINT</b>	immediato tag	valore su cui eseguire AND con Source A
Destination	SINT INT <b>DINT</b>	tag	memorizza il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.

Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione AND esegue un'operazione AND di bit utilizzando i bit di Source A e Source B ed inserendo il risultato nella destinazione.

Se è abilitata, l'istruzione calcola l'operazione AND:

Se il bit di Source A è:	E il bit di Source B è:	Il bit di Destination è:
0	0	0
0	1	0
1	0	0
1	1	1

Se si mischiano tipi di dati interi, l'istruzione riempie con degli 0 i bit più significativi dei tipi di dati interi più piccoli in modo che questi abbiano la stessa dimensione dei tipi di dati più grandi.

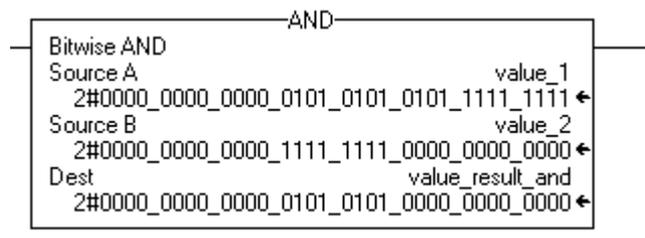
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione esegue un'operazione AND di bit. La condizione del ramo di uscita viene impostata su vero.

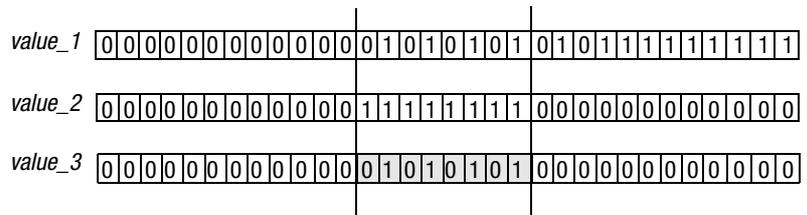
**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di AND:**



Se è abilitata, l'istruzione AND esegue un'operazione AND di bit su *value\_1* e *value\_2* ed inserisce il risultato in *value\_result\_and*.



Le caselle ombrate mostrano i bit che sono cambiati.

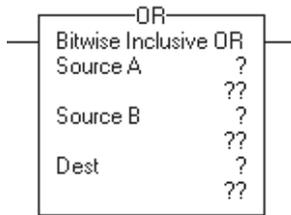
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>AND(<i>source_A</i>,<i>source_B</i>,<i>destination</i>);</code>
testo ASCII	<code>AND <i>source_A</i> <i>source_B</i> <i>destination</i></code>

**Istruzioni correlate:** NOT, OR, XOR

**OR di bit (OR)**

L'istruzione OR è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT <b>DINT</b>	immediato tag	valore su cui eseguire OR con Source B
			Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.
Source B	SINT INT <b>DINT</b>	immediato tag	valore su cui eseguire OR con Source A
			Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.
Destination	SINT INT <b>DINT</b>	tag	memorizza il risultato

**Descrizione:** L'istruzione OR esegue un'operazione OR di bit utilizzando i bit di Source A e Source B ed inserendo il risultato nella destinazione.

Se è abilitata, l'istruzione calcola l'operazione OR:

**Se il bit di Source A è:    E il bit di Source B è:    Il bit della Destination è:**

0	0	0
0	1	1
1	0	1
1	1	1

Se si mischiano tipi di dati interi, l'istruzione riempie con degli 0 i bit più significativi dei tipi di dati interi più piccoli in modo che questi abbiano la stessa dimensione dei tipi di dati più grandi.

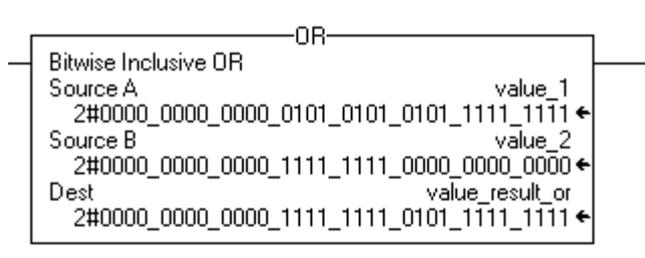
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione esegue un'operazione OR di bit. La condizione del ramo di uscita viene impostata su vero.

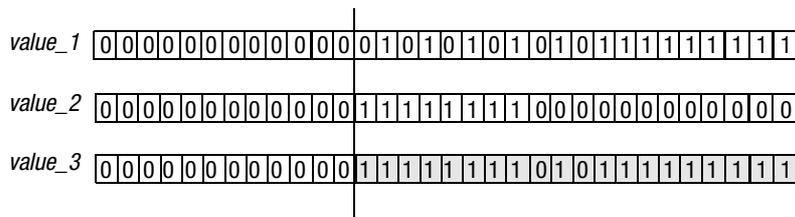
**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di OR:**



Se è abilitata, l'istruzione OR esegue un'operazione OR di bit su *value\_1* e *value\_2* ed inserisce il risultato in *value\_result\_or*.



Le caselle ombrate mostrano i bit che sono cambiati.

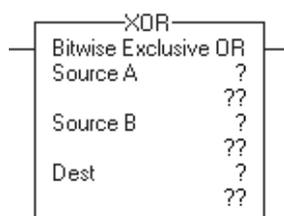
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>OR(source_A, source_B, destination);</code>
testo ASCII	<code>OR source_A source_B destination</code>

**Istruzioni correlate:** AND, OR, XOR

**OR esclusivo di bit (XOR)**

L'istruzione XOR è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT DINT	immediato tag	valore su cui eseguire XOR con Source B
Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.			
Source B	SINT INT DINT	immediato tag	valore su cui eseguire XOR con Source A
Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.			
Destination	SINT INT DINT	tag	memorizza il risultato

**Descrizione:** L'istruzione XOR esegue un'operazione XOR di bit utilizzando i bit di Source A e Source B ed inserendo il risultato nella destinazione.

Se è abilitata, l'istruzione calcola l'operazione XOR:

**Se il bit di Source A è: E il bit di Source B è: Il bit della destinazione è:**

0	0	0
0	1	1
1	0	1
1	1	0

Se si mischiano tipi di dati interi, l'istruzione riempie con degli 0 i bit più significativi dei tipi di dati interi più piccoli in modo che questi abbiano la stessa dimensione dei tipi di dati più grandi.

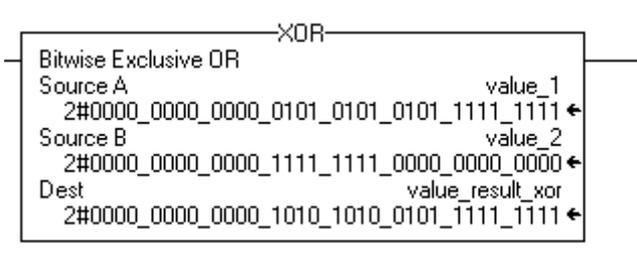
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione esegue un'operazione XOR di bit. La condizione del ramo di uscita viene impostata su vero.

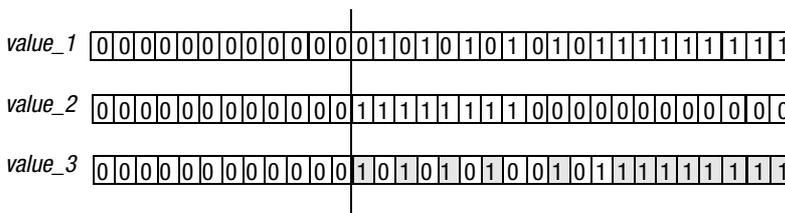
**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di XOR:**



Se è abilitata, l'istruzione XOR esegue un'operazione XOR di bit su *value\_1* e *value\_2* ed inserisce il risultato in *value\_result\_or*.



Le caselle ombrate mostrano i bit che sono cambiati.

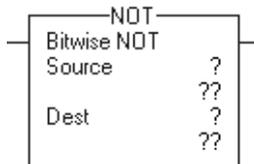
**Altri formati:**

Formato:	Sintassi:
testo neutro	XOR( <i>source_A</i> , <i>source_B</i> , <i>destination</i> );
testo ASCII	XOR <i>source_A</i> <i>source_B</i> <i>destination</i>

**Istruzioni correlate:** AND, NOT, OR

**NOT di bit (NOT)**

L'istruzione NOT è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT	immediato tag	valore su cui eseguire NOT
Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.			
Destination	SINT INT DINT	tag	memorizza il risultato

**Descrizione:** L'istruzione NOT esegue un'operazione NOT di bit utilizzando i bit di Source ed inserendo il risultato nella Destination.

Se è abilitata, l'istruzione calcola l'operazione NOT:

**Se il bit di Source A è: Il bit di Destination è:**

0	1
1	0

Se si mischiano tipi di dati interi, l'istruzione riempie con degli 0 i bit più significativi dei tipi di dati interi più piccoli in modo che questi abbiano la stessa dimensione dei tipi di dati più grandi.

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	L'istruzione esegue un'operazione NOT di bit. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna



## Istruzioni di Matrici (File)/Varie Istruzioni (FAL, FSC, COP, FLL, AVE, SRT, STD)

### Introduzione

Le istruzioni di file/varie operano su matrici di dati.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
eseguire operazioni aritmetiche, logiche, di spostamento e funzionali sui valori delle matrici	FAL	7-7
cercare e confrontare valori di matrici	FSC	7-18
copiare il contenuto di una matrice in un'altra matrice	COP	7-28
riempire una matrice con determinati dati	FLL	7-32
calcolare la media di una matrice di valori	AVE	7-35
ordinare una dimensione dei dati di una matrice in ordine crescente	SRT	7-39
calcolare la deviazione standard di una matrice di valori	STD	7-42

È possibile usare tipi di dati diversi, ma si potrebbe perdere in precisione ed in arrotondamento ed inoltre l'istruzione potrebbe impiegare più tempo per essere eseguita. Controllare il bit S:V per verificare se il risultato è stato troncato.

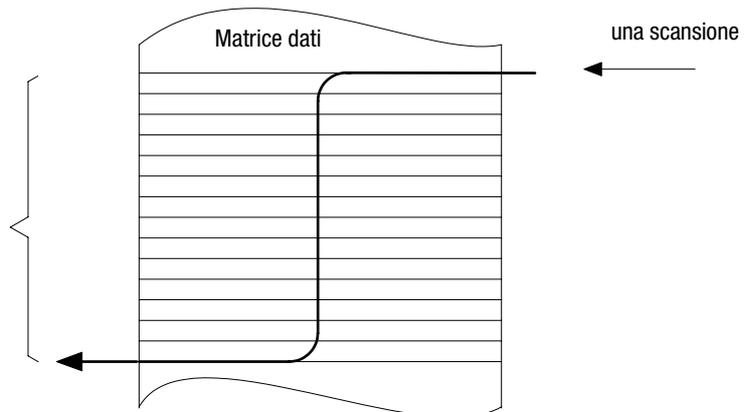
I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente ed utilizzando meno memoria se tutti gli operandi di un'istruzione utilizzano il medesimo tipo di dati ottimale, generalmente DINT o REAL.

**Scelta della modalità operativa** Nel caso delle istruzioni FAL e FSC, la modalità indica al controllore come distribuire le operazioni di matrice.

<b>Se si desidera:</b>	<b>Selezionare questa modalità:</b>
operare su tutti gli elementi specificati in una matrice prima di continuare con l'istruzione successiva	All (Tutti)
distribuire l'operazione sulla matrice su più scansioni immettere il numero di elementi per scansione su cui operare (1-2147483647)	Numerico
manipolare un elemento della matrice ogni volta che la condizione del ramo di ingresso passa da falsa a vera	Incrementale

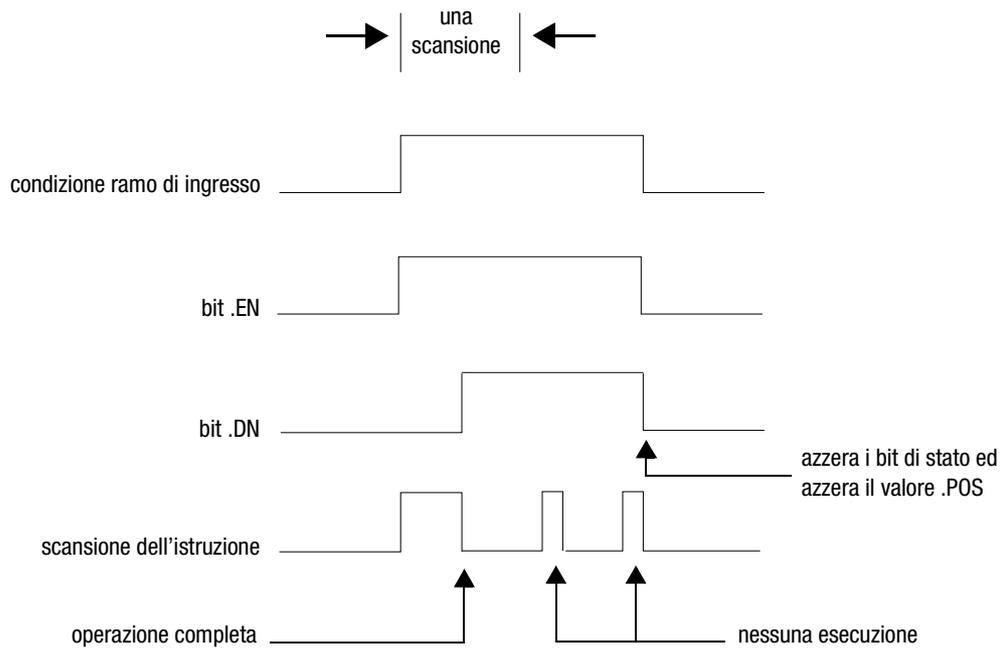
### Modalità All (Continua)

In modalità ALL, si opera su tutti gli elementi specificati nella matrice prima di continuare con l'istruzione successiva. L'operazione inizia quando la condizione del ramo di ingresso dell'istruzione passa da falsa a vera. Il valore posizione (.POS) della struttura di controllo indica l'elemento della matrice che l'istruzione sta utilizzando. L'operazione ha termine quando il valore .POS è uguale al valore .LEN.



16639

Il seguente schema di temporizzazione mostra la relazione tra i bit di stato e il funzionamento dell'istruzione. Quando l'esecuzione dell'istruzione è completa, viene impostato il bit .DN. Quando la condizione del ramo di ingresso è falsa, il bit .DN bit, il bit .EN ed il valore .POS vengono azzerati. Solo a questo punto una transizione della condizione del ramo di ingresso da falsa a vera può attivare un'altra esecuzione dell'istruzione.



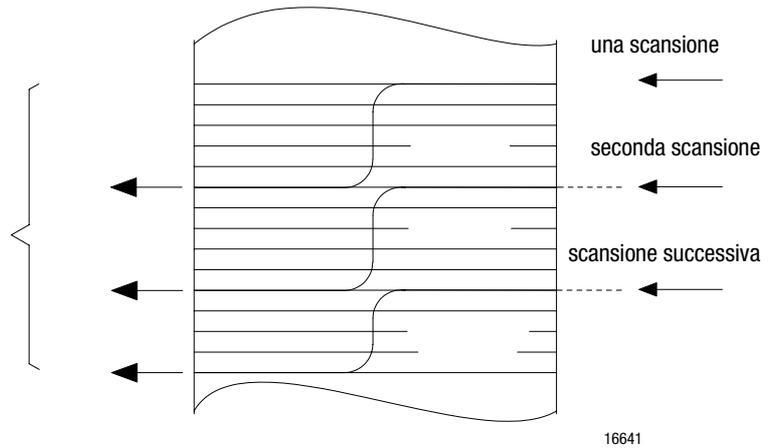
40010

### Modalità discreta

La modalità numerica distribuisce le operazioni sulla matrice in varie scansioni. Questa modalità è utile quando si lavora su dati non critici per tempo o su grandi quantità di dati. Si inserisce il numero di elementi su cui operare ad ogni scansione e così facendo si riducono i tempi di scansione.

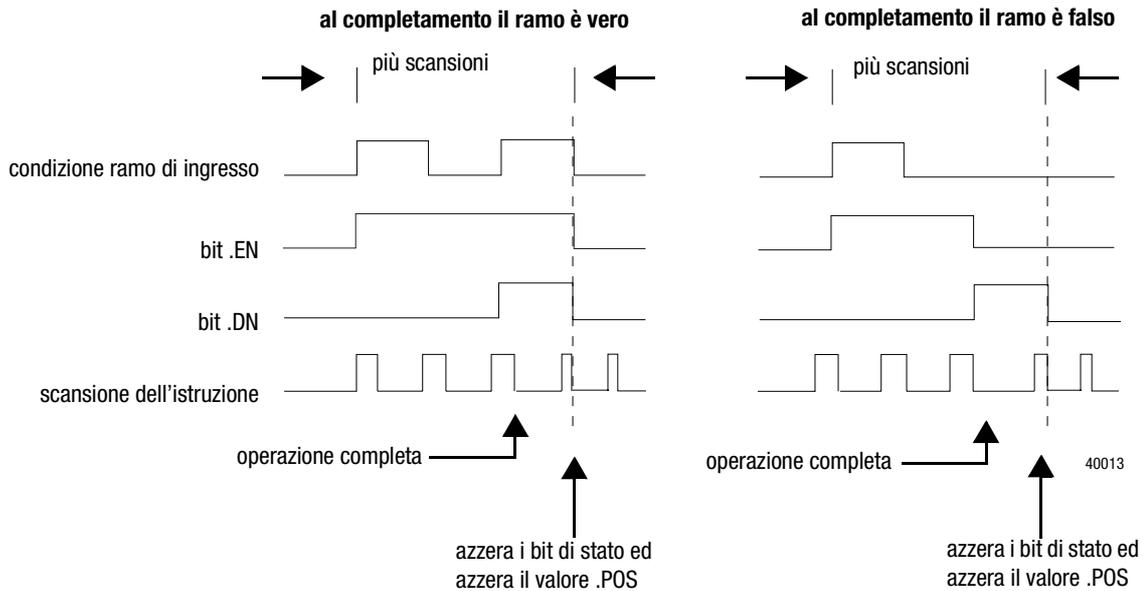
L'esecuzione viene attivata quando la condizione del ramo di ingresso passa da falsa a vera. Una volta attivata, l'istruzione viene eseguita ogni volta che viene scandita e per il numero di volte necessario a completare l'operazione su tutta la matrice. Dopo essere stata attivata, la condizione del ramo di

ingresso può cambiare continuamente senza interrompere l'esecuzione dell'istruzione.



**Importante:** evitare di utilizzare i risultati di un'istruzione di file funzionante in modalità numerica fino a quando non sia stato impostato il bit .DN.

Il seguente schema di temporizzazione mostra la relazione tra i bit di stato e il funzionamento dell'istruzione. Quando l'esecuzione dell'istruzione è completa, viene impostato il bit .DN.

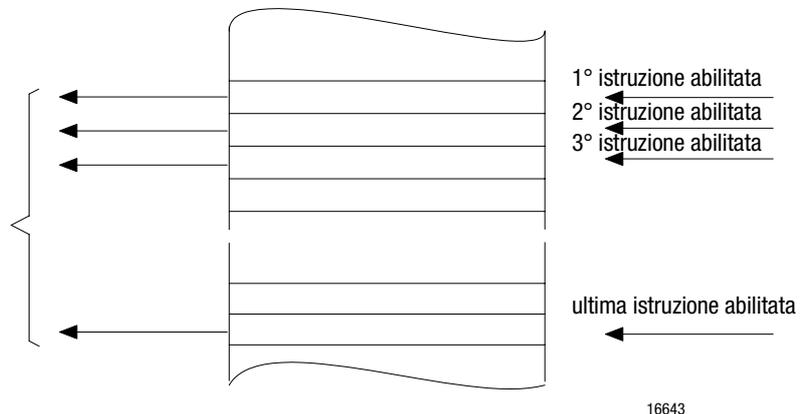


Se al completamento la condizione del ramo di ingresso è vera, i bit .DN ed .ER sono impostati fino a quando la condizione del ramo di ingresso diventa falsa. Quando la condizione del ramo di ingresso diventa falsa, questi bit vengono azzerati così come il valore .POS.

Se al completamento la condizione del ramo di ingresso è falsa, il bit .EN viene azzerato immediatamente. Una scansione dopo l'azzeramento del bit .EN, vengono azzerati il bit .DN ed il valore .POS.

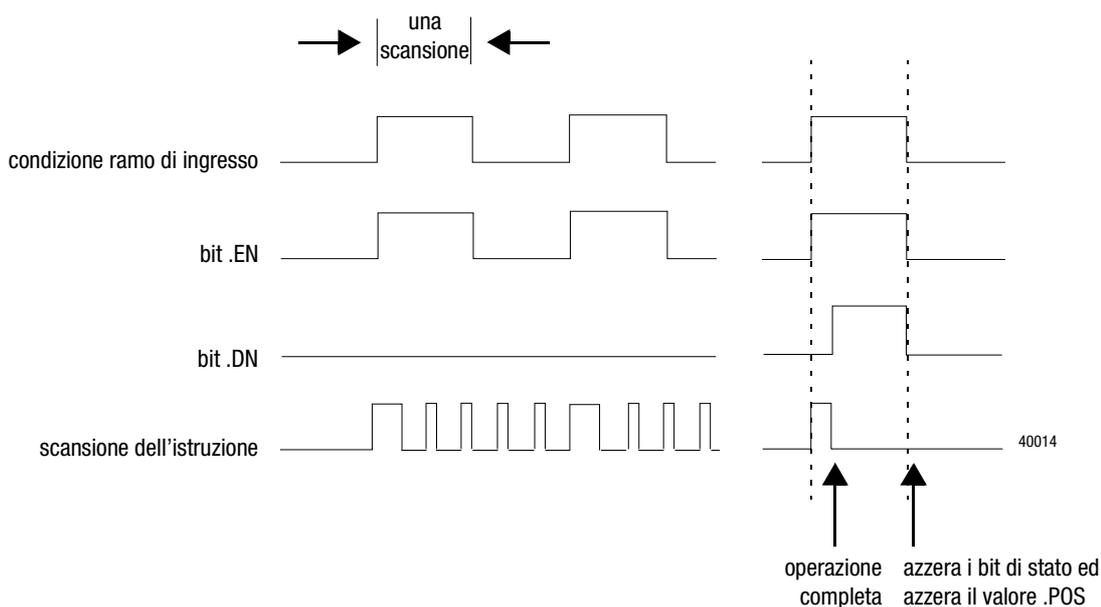
### Modalità incrementale

La modalità incrementale manipola un elemento della matrice ogni volta che la condizione del ramo di ingresso dell'istruzione passa da falsa a vera.



16643

Il seguente schema di temporizzazione mostra la relazione tra i bit di stato e il funzionamento dell'istruzione. L'esecuzione avviene solo in una scansione in cui la condizione del ramo di ingresso passa da falsa a vera. Ogni volta che ciò si verifica viene manipolato un solo elemento della matrice. Se la condizione del ramo di ingresso rimane vera per più di una scansione, l'istruzione viene eseguita solamente durante la prima scansione.



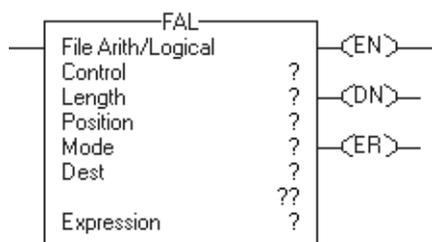
Quando la condizione del ramo di ingresso è vera, viene impostato il bit .EN. Quando è stato manipolato l'ultimo elemento della matrice, viene impostato il bit .DN. Quando l'ultimo elemento è stato manipolato e la condizione del ramo di ingresso diventa falsa, il bit .EN bit, il bit .DN bit ed il valore .POS vengono azzerati.

La differenza tra la modalità incrementale e la modalità numerica alla frequenza di un elemento per scansione è che:

- per iniziare l'esecuzione, la modalità numerica con un qualsiasi numero di elementi per scansione richiede solo una transizione da falsa a vera della condizione del ramo di ingresso. L'istruzione continua l'esecuzione del numero di elementi specificato per ogni scansione fino al suo completamento, a prescindere dallo stato della condizione del ramo di ingresso.
- La modalità incrementale richiede un cambiamento della condizione del ramo di ingresso da falsa a vera affinché sia possibile manipolare un elemento della matrice.

## Aritmetica e logica di file (FAL) L'istruzione FAL è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Control	CONTROL	tag	struttura di controllo per l'operazione
Length	DINT	immediato	numero di elementi della matrice da manipolare
Position	DINT	immediato	elemento corrente della matrice il valore iniziale generalmente è 0
Mode	DINT	immediato	come distribuire l'operazione selezionare INC, ALL oppure inserire un numero
Destination	SINT INT <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato
Expression	SINT INT <b>DINT</b> <b>REAL</b>	immediato tag	un'espressione formata da tag e/o valori immediati separati da operatori

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag.A-6.

### Struttura CONTROL:

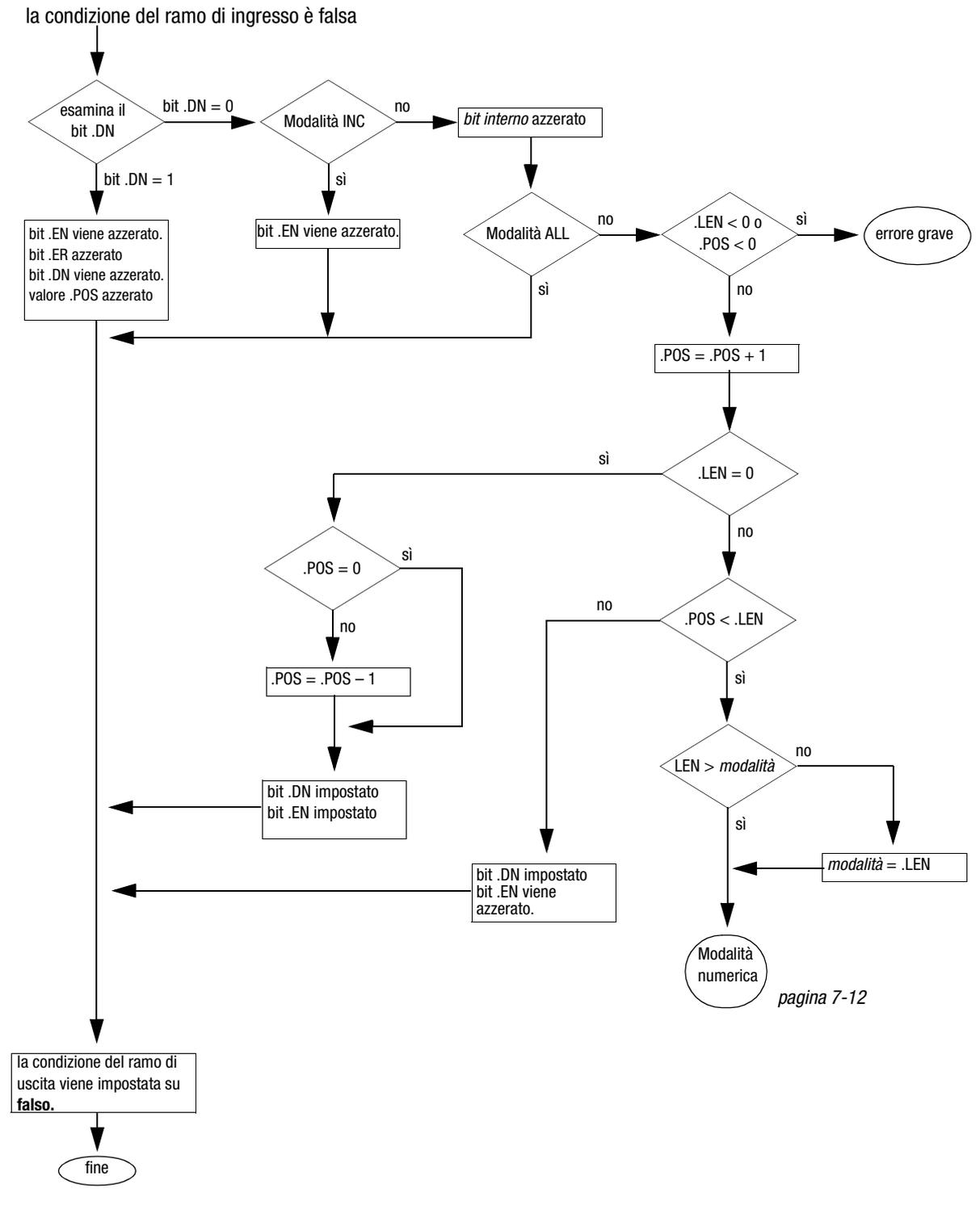
Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione FAL è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando l'istruzione ha operato sull'ultimo elemento (.POS = .LEN).
.ER	BOOL	Il bit di errore viene impostato se l'espressione genera un overflow (S:V è impostato). L'istruzione interrompe la sua esecuzione fino a quando il programma non azzeri il bit .ER. Il valore .POS contiene la posizione dell'elemento che ha causato un overflow.
.LEN	DINT	La lunghezza specifica il numero di elementi della matrice su cui opera l'istruzione FAL.
.POS	DINT	La posizione contiene la posizione dell'elemento corrente a cui l'istruzione accede.

**Descrizione:** L'istruzione FAL esegue operazioni di copia, aritmetiche, logiche e di funzione sui dati memorizzati in una matrice. L'istruzione FAL esegue sulle matrici le stesse operazioni che l'istruzione CPT esegue sugli elementi.

Gli esempi che iniziano a pagina 7-13 mostrano come utilizzare il valore .POS per avanzare nella matrice. Se un indice dell'espressione della destinazione è fuori gamma, l'istruzione FAL genera un errore grave (tipo 4, codice 20).

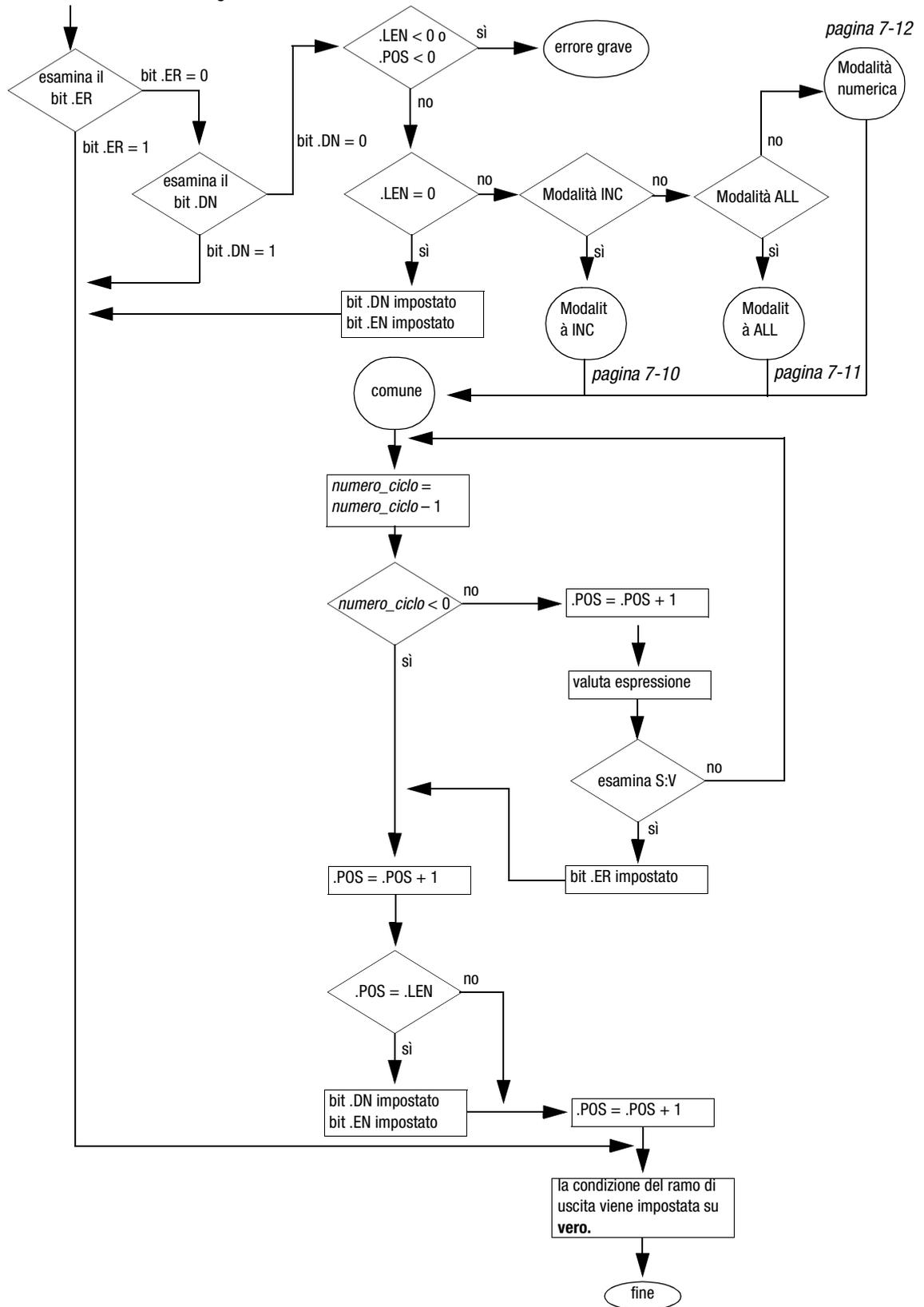
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.



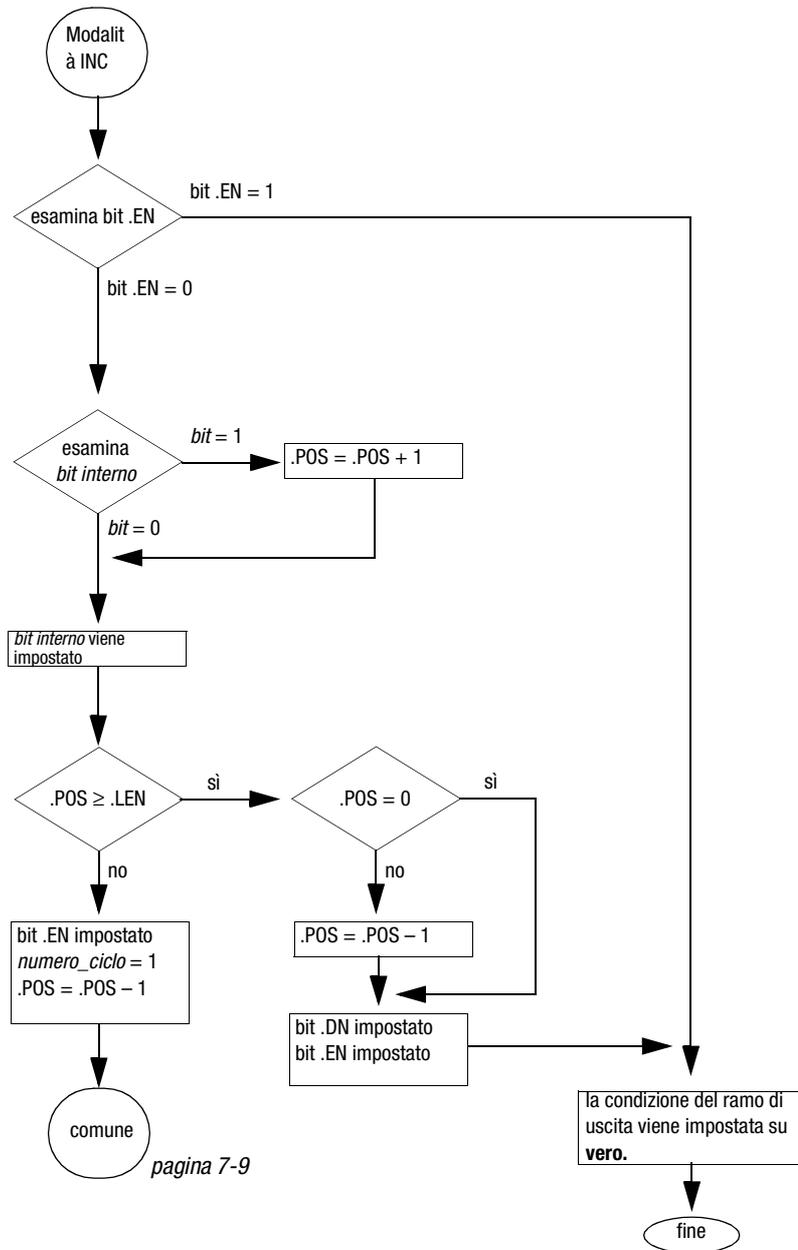
**Condizione:****Azione:**

la condizione del ramo di ingresso è vera

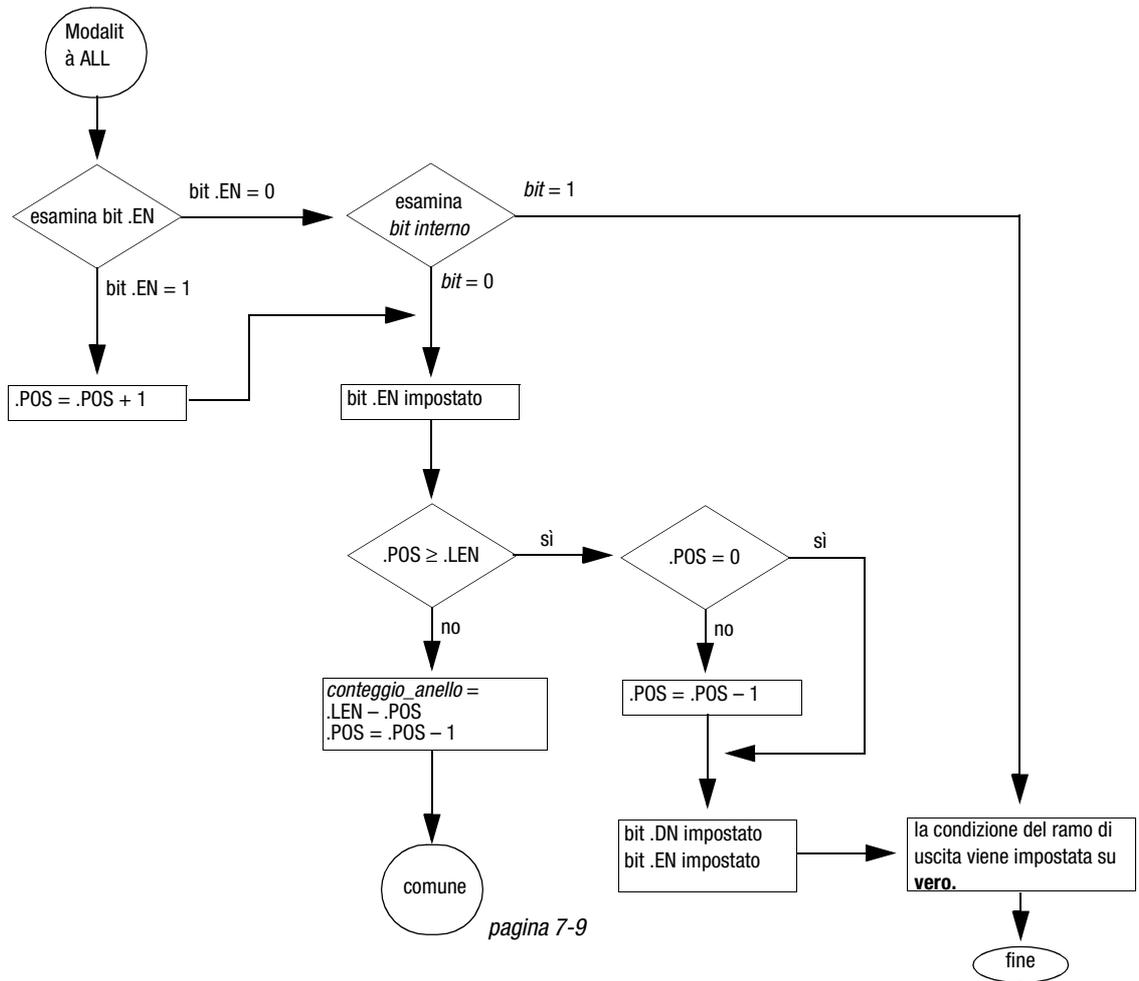


**Condizione:**

**Azione:**

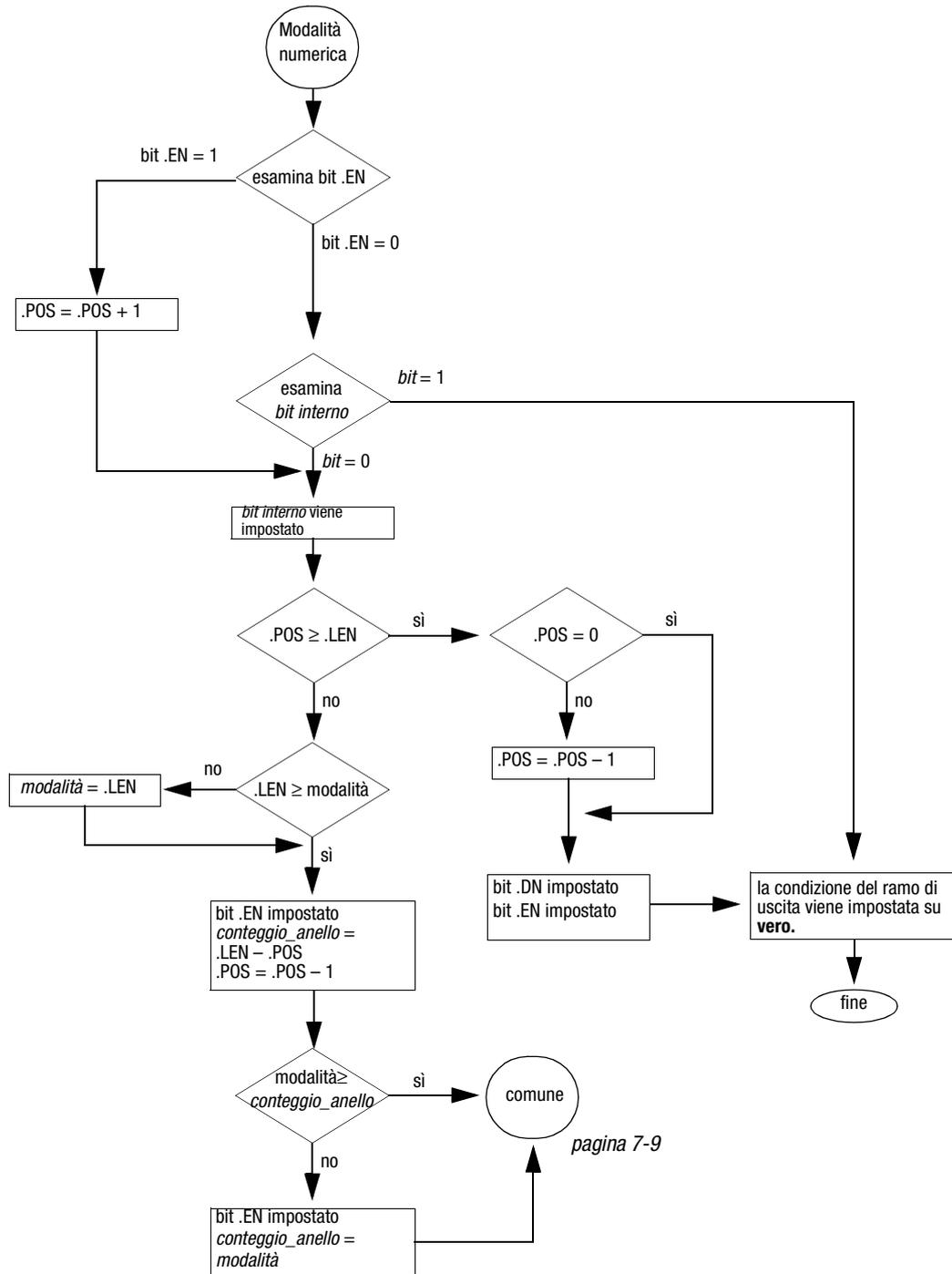


pagina 7-9

**Condizione:****Azione:**

**Condizione:**

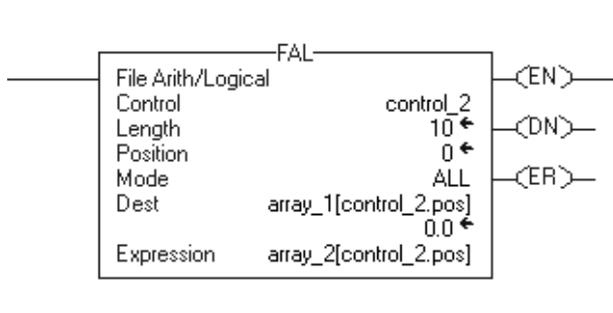
**Azione:**



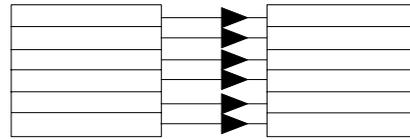
**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:**

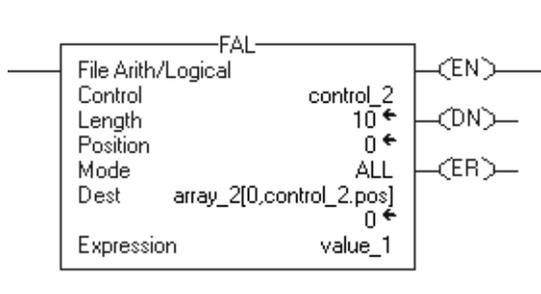
Si verifica un errore grave se:	Tipo errore:	Codice errore:
l'indice è fuori gamma	4	20
.POS < 0 oppure .LEN < 0	4	21

**Esempio di FAL:****copia da matrice a matrice**

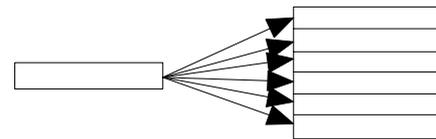
Se è abilitata, l'istruzione FAL copia ciascun elemento di *array\_2* nella stessa posizione di *array\_1*.



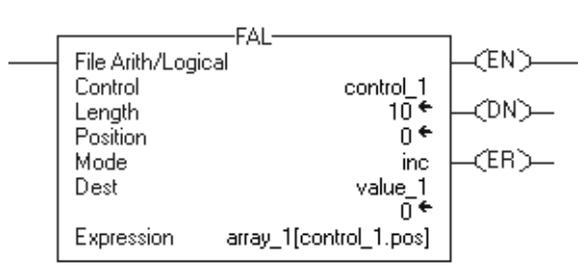
Espressione: *array\_2[control\_2.pos]*      Destinazione: *array\_1[control\_2.pos]*

**copia da elemento a matrice**

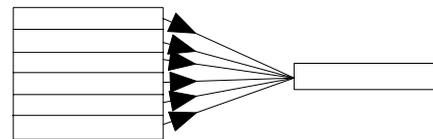
Se è abilitata, l'istruzione FAL copia *value\_1* nelle prime 10 posizioni della seconda dimensione di *array\_2*.



Espressione: *value\_1*      Destinazione: *array\_2[0,control\_2.pos]*

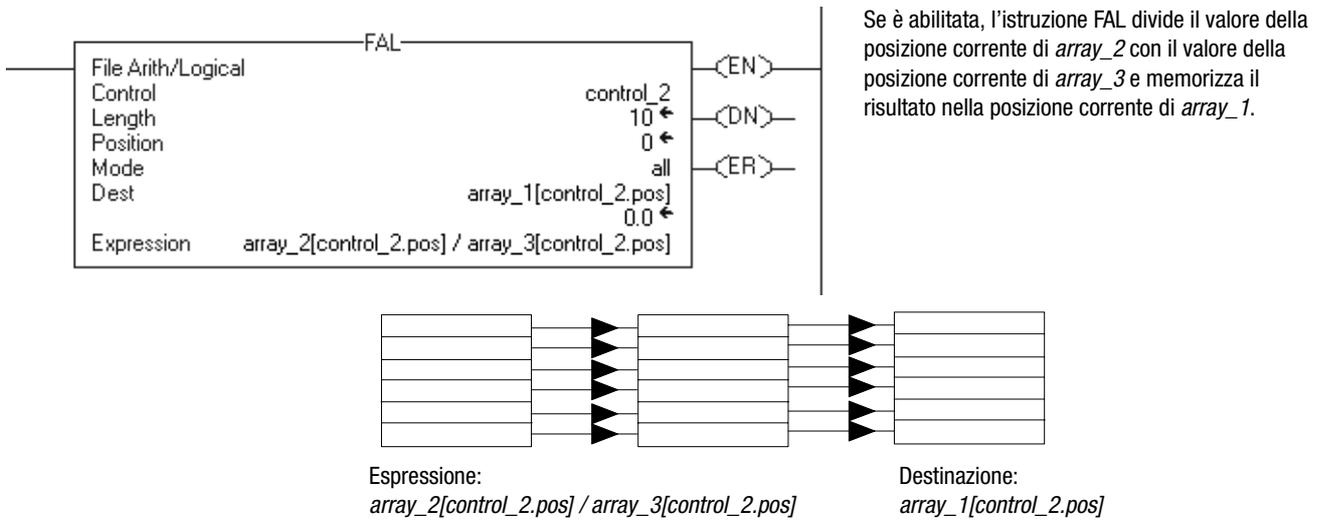
**copia da matrice ad elemento**

Ogni volta che l'istruzione FAL è abilitata, questa copia il valore corrente di *array\_1* in *value\_1*. L'istruzione FAL utilizza la modalità incrementale, pertanto, ogni volta che l'istruzione viene abilitata, viene copiato solo un elemento alla volta. La volta successiva che viene abilitata, l'istruzione sovrascrive *value\_1* con il valore successivo di *array\_1*.

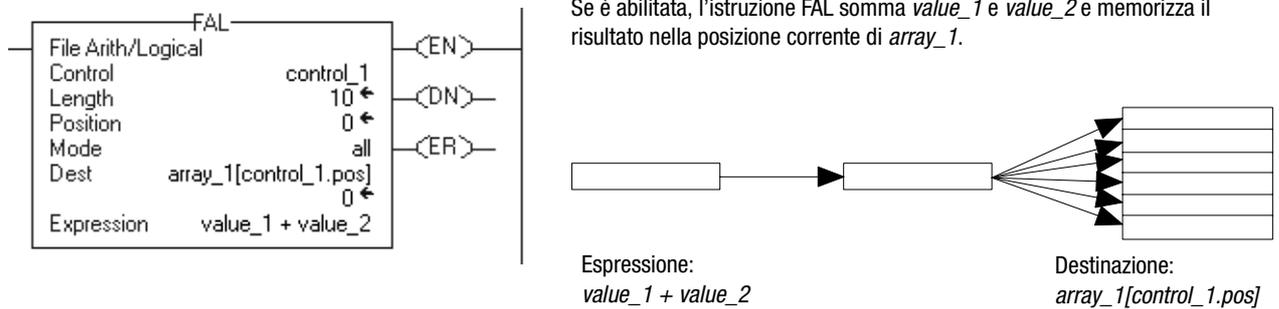


Espressione: *array\_1[control\_1.pos]*      Destinazione: *value\_1*

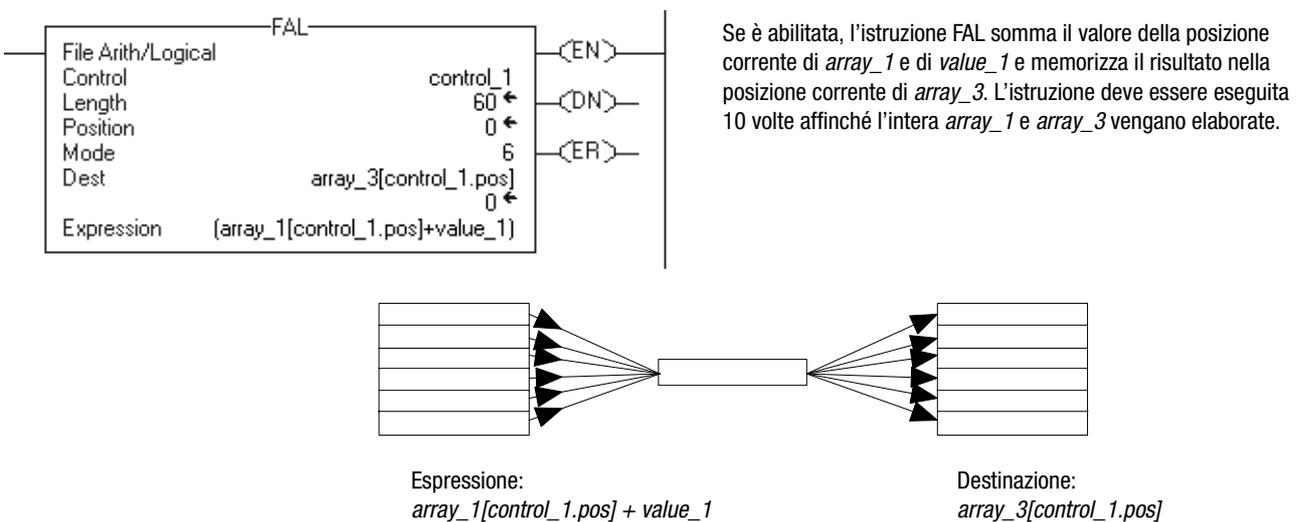
**operazione aritmetica: (matrice/matrice) a matrice**

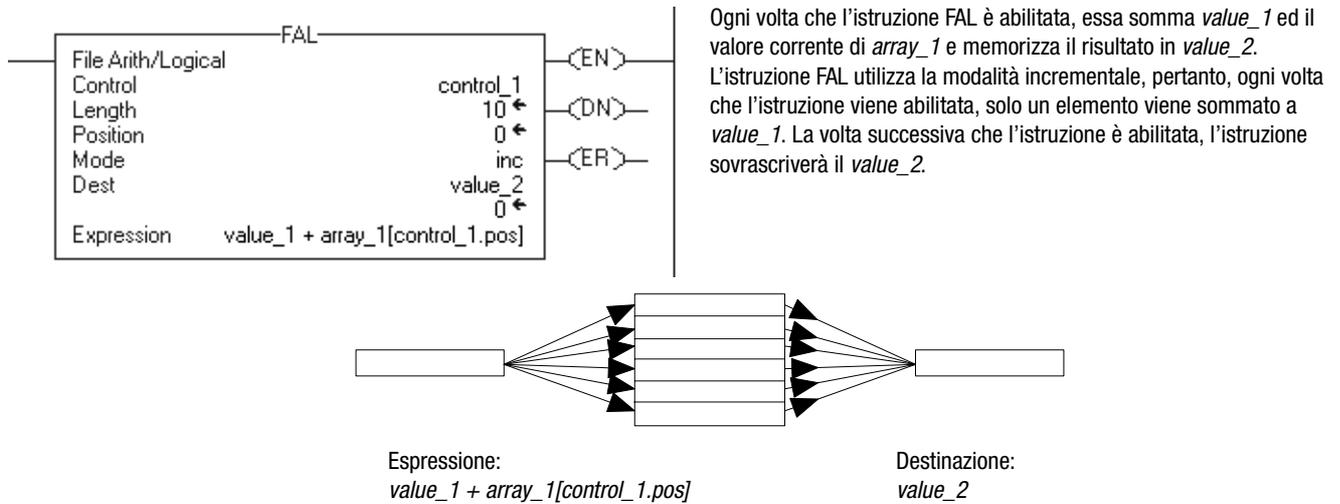
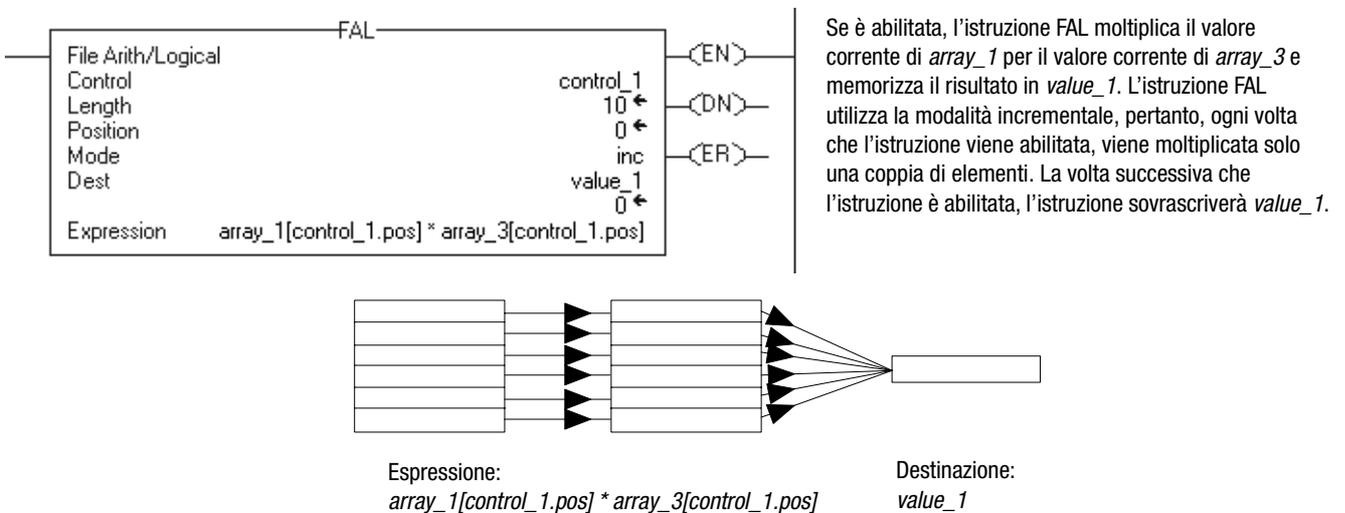


**operazione aritmetica: (elemento + elemento) a matrice**



**operazione aritmetica: (matrice + elemento) a matrice**



**operazione aritmetica: (elemento + matrice) ad elemento****operazione aritmetica: (array \* array) ad elemento****Altri formati:****Formato:**      **Sintassi:**

testo neutro	<code>FAL( control , length , position , mode , destination , expression ) ;</code>
testo ASCII	<code>FAL control length position mode destination expression</code>

**Istruzioni correlate:** CPT, CMP, FSC

Le espressioni delle istruzioni FAL vanno programmate nello stesso modo delle espressioni delle istruzioni CPT. Per informazioni su operatori validi, formato ed ordine delle operazioni comuni ad entrambi le istruzioni, utilizzare le seguenti sezioni.

## Operatori validi

Operatore:	Descrizione:	Ottimale:	Operatore:	Descrizione:	Ottimale:
+	somma	DINT, REAL	LN	logaritmo naturale	REAL
-	sottrai/nega	DINT, REAL	LOG	logaritmo in base 10	REAL
*	moltiplica	DINT, REAL	MOD	dividi modulo	DINT, REAL
/	dividi	DINT, REAL	NOT	complemento di bit	DINT
**	esponente (x alla y)	DINT, REAL	OR	OR di bit	DINT
ABS	valore assoluto	DINT, REAL	RAD	da gradi a radianti	DINT, REAL
ACS	arcocoseno	REAL	SIN	seno	REAL
AND	AND di bit	DINT	SQR	radice quadrata	DINT, REAL
ASN	arcoseno	REAL	TAN	tangente	REAL
ATN	arcotangente	REAL	TOD	da intero a BCD	DINT
COS	coseno	REAL	TRN	tronca	DINT, REAL
DEG	da radianti a gradi	DINT, REAL	XOR	OR di bit esclusivo	DINT
FRD	da BCD a intero	DINT			

## Formattazione delle espressioni

Per ciascun operatore utilizzato in un'espressione, è necessario fornire uno o due operandi (tag o valori immediati). Per la formattazione degli operatori e degli operandi di un'espressione, utilizzare la seguente tabella:

Per operatori con:	Utilizzare questo formato:	Esempi:
un operando	operatore(operando)	ABS(tag_a)
due operandi	operando_a operatore operando_b	<ul style="list-style-type: none"> <li>• tag_b + 5</li> <li>• tag_c AND tag_d</li> <li>• (tag_e ** 2) MOD (tag_f / tag_g)</li> </ul>

## Determinazione dell'ordine delle operazioni

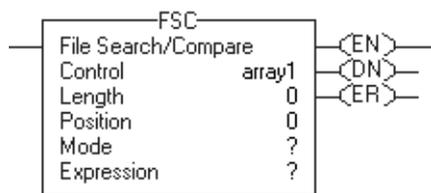
Le operazioni che vengono scritte nell'espressione sono eseguite dall'istruzione in un determinato ordine, non necessariamente nell'ordine con cui sono state scritte. È possibile forzare l'ordine delle operazioni raggruppando i termini all'interno di parentesi e forzando l'istruzione ad eseguire un'operazione all'interno delle parentesi prima delle altre operazioni.

Operazioni di ordine uguale vengono eseguite da sinistra a destra.

<b>Ordine:</b>	<b>Funzionamento:</b>
1.	()
2.	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3.	**
4.	– (nega), NOT
5.	*, /, MOD
6.	– (sottrai), +
7.	AND
8.	XOR
9.	OR

**Ricerca e confronto file (FSC)**

L'istruzione FSC è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Control	CONTROL	tag	struttura di controllo per l'operazione
Length	DINT	immediato	numero di elementi della matrice da manipolare
Position	DINT	immediato	offset nella matrice il valore iniziale generalmente è 0
Mode	DINT	immediato	come distribuire l'operazione selezionare INC, ALL oppure inserire un numero
Expression	SINT INT DINT REAL	immediato tag	un'espressione formata da tag e/o valori immediati separati da operatori

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag.A-6.

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione FSC è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando l'istruzione ha operato sull'ultimo elemento (.POS = .LEN).
.ER	BOOL	Il bit di errore non è modificato.
.IN	BOOL	Il bit di inibizione indica che l'istruzione FSC ha rilevato un confronto vero. Per continuare l'operazione, bisogna azzerare questo bit.
.FD	BOOL	Il bit di trovato indica che l'istruzione FSC ha rilevato un confronto vero.
.LEN	DINT	La lunghezza specifica il numero di elementi della matrice su cui opera l'istruzione.
.POS	DINT	Contiene la posizione dell'elemento corrente a cui l'istruzione accede.

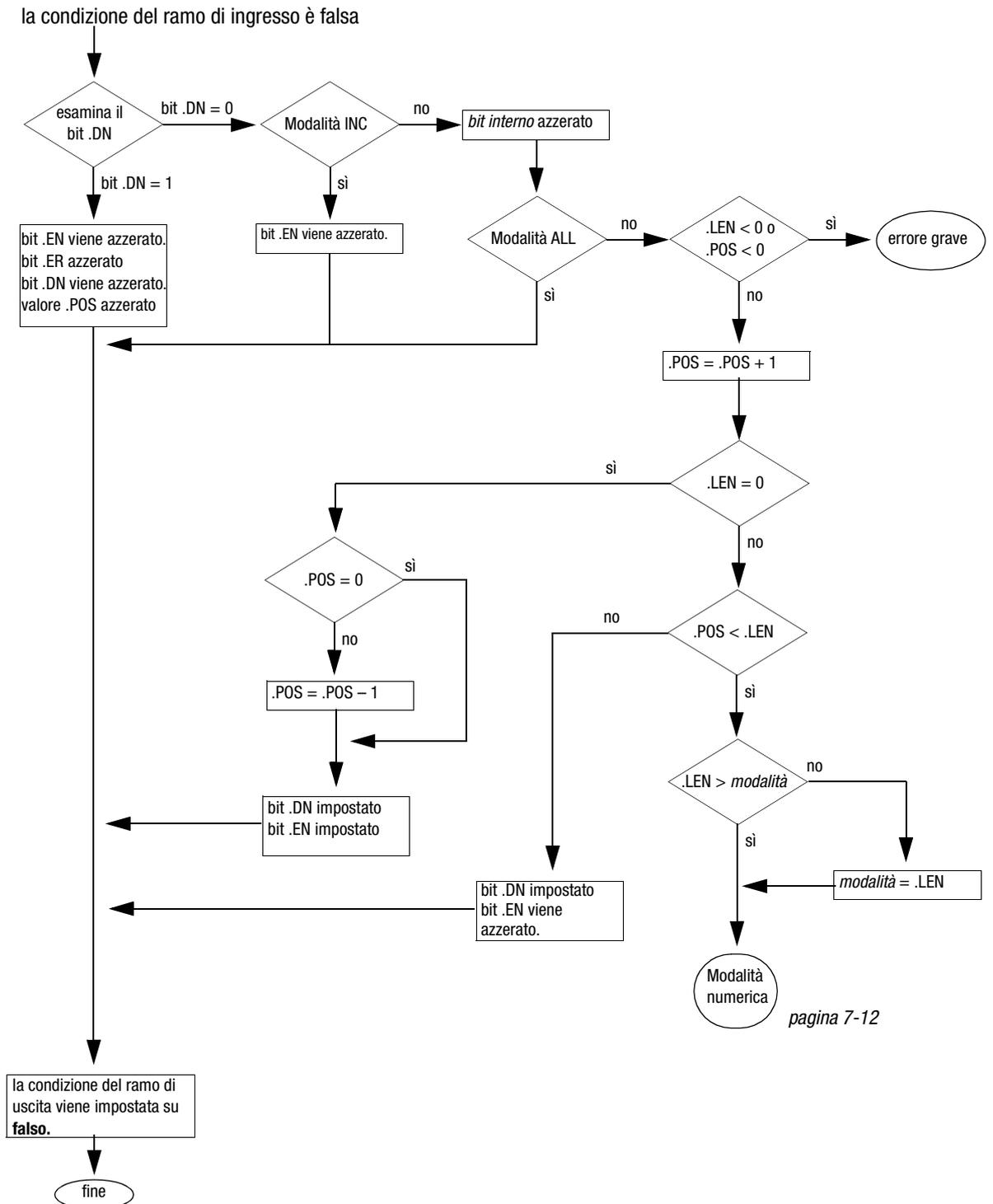
**Descrizione:** L'istruzione FSC confronta i valori di una matrice, elemento per elemento, per le operazioni di logica specificate nell'Espressione. Vedere *Visualizzazione di una matrice come insieme di elementi* a pagina B-1.

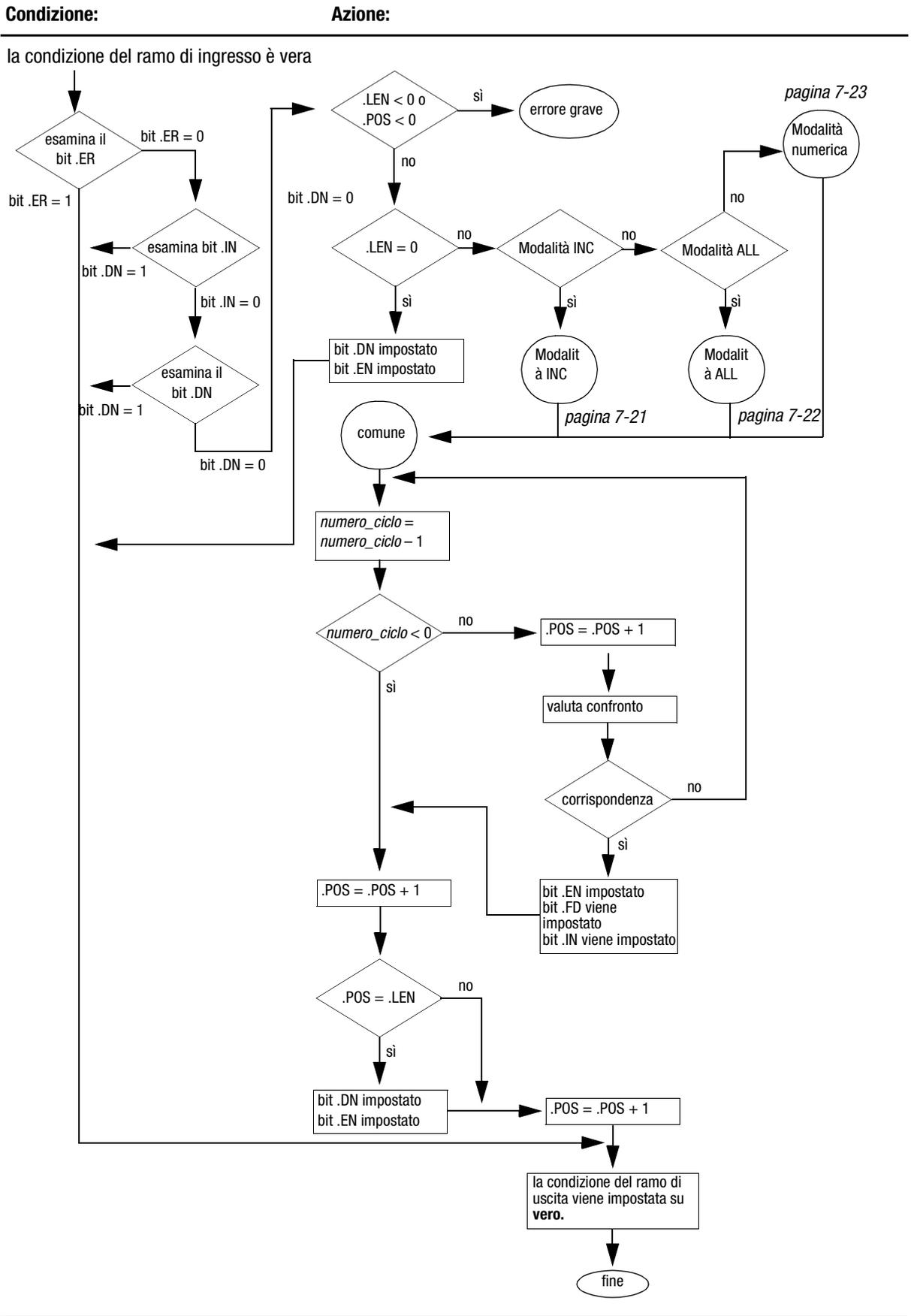
Quando l'istruzione FSC è abilitata ed il confronto risulta vero, l'istruzione imposta il bit .FD ed il bit .POS indica la posizione della matrice in cui l'istruzione ha trovato il confronto vero. L'istruzione imposta il bit .IN per interrompere la ricerca.

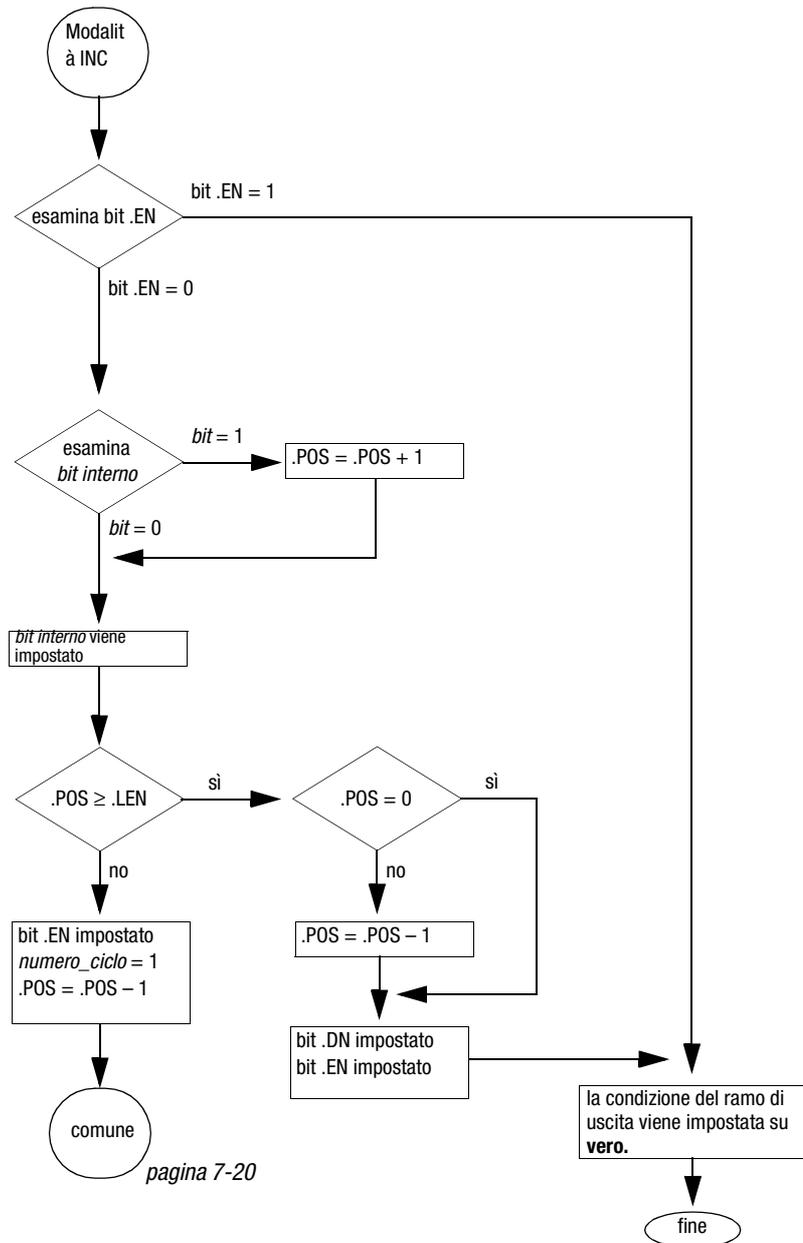
**Esecuzione:****Condizione:****Azione:**

prescansione

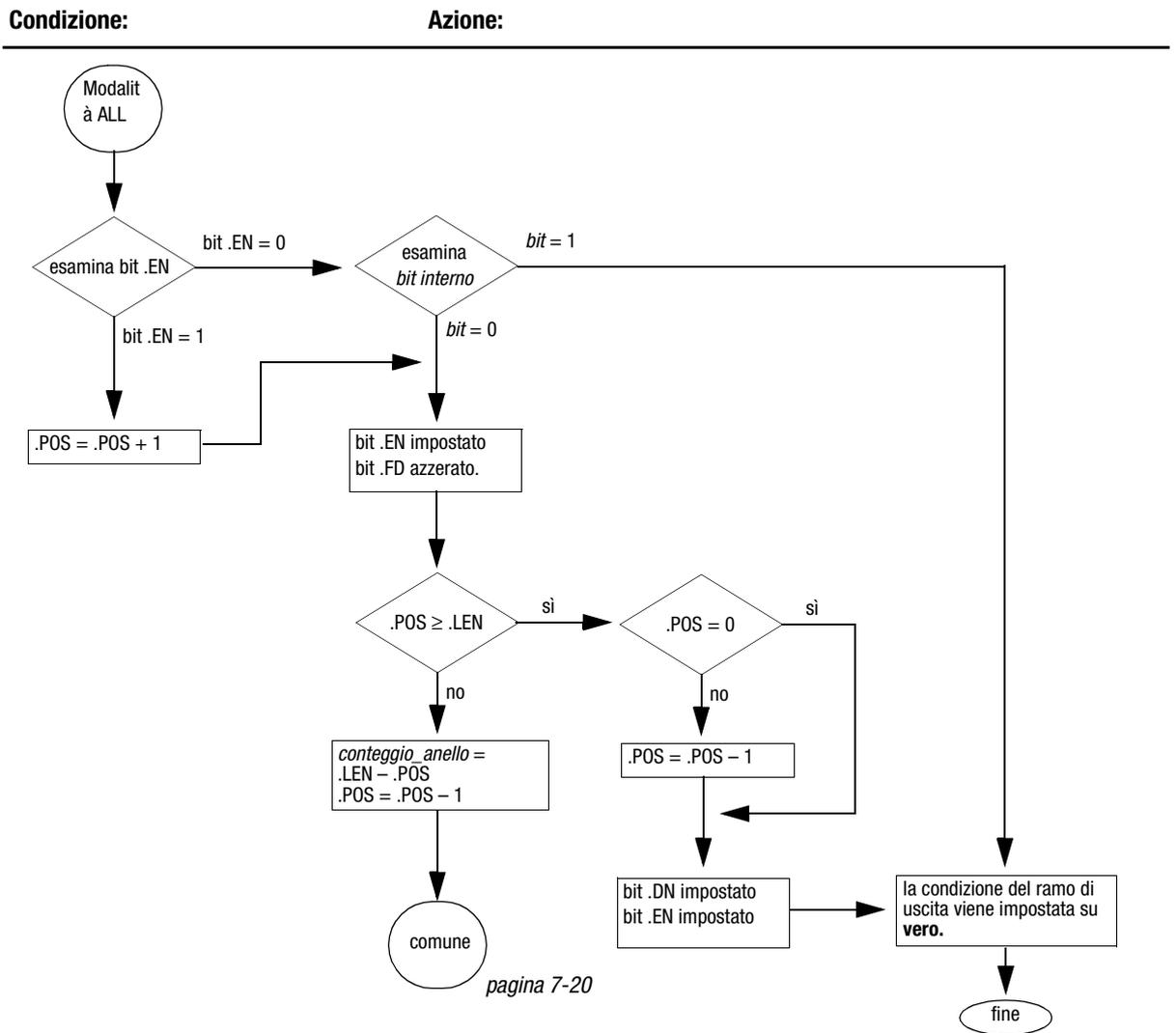
La condizione del ramo di uscita viene impostata su falso.

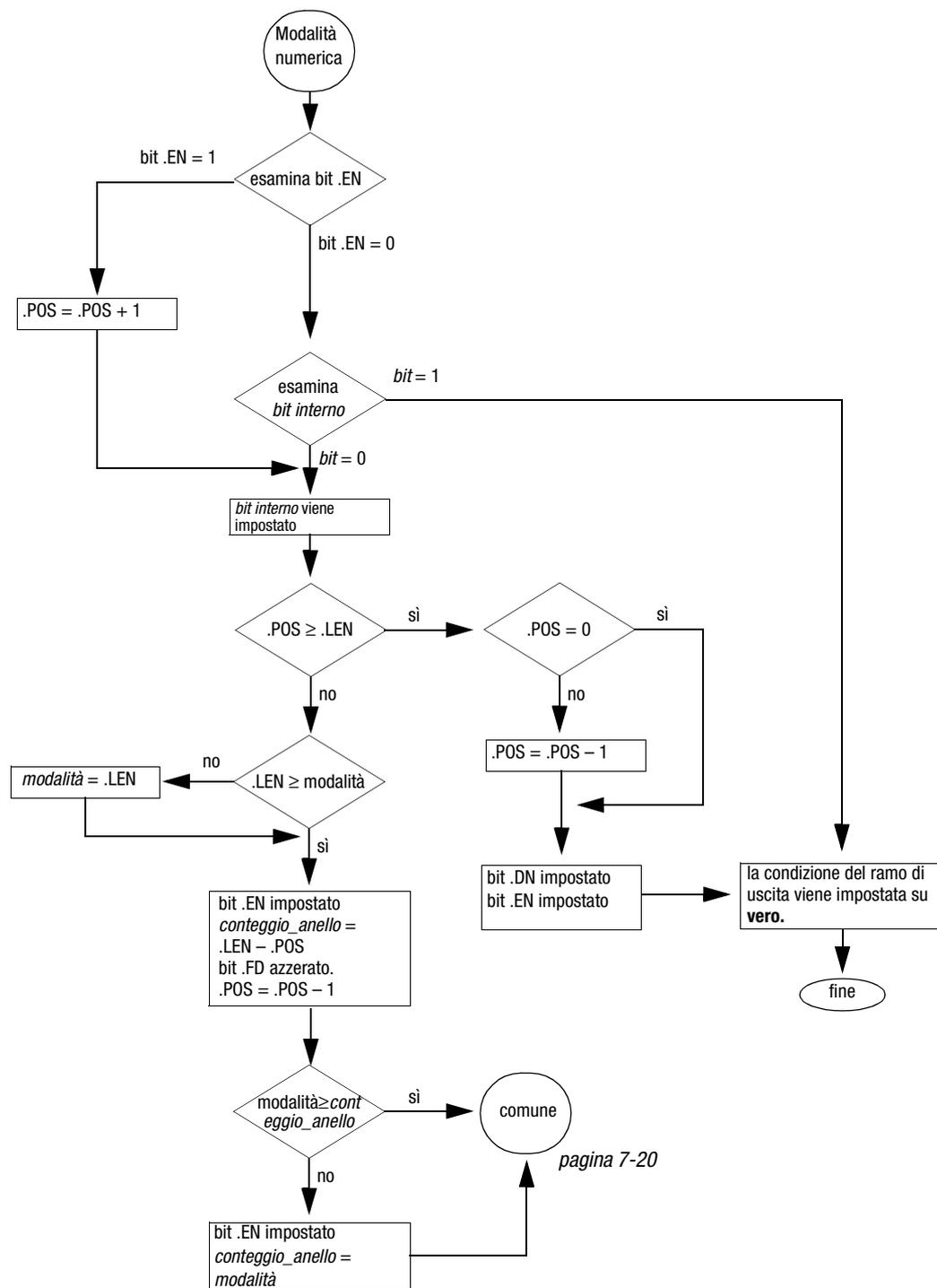




**Condizione:****Azione:**

pagina 7-20



**Condizione:****Azione:****Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.**Condizioni di errore:****Si verifica un errore grave se:    Tipo errore:    Codice errore:**

.POS &lt; 0 oppure .LEN &lt; 0

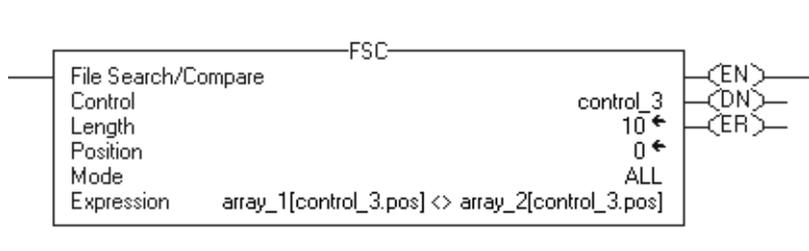
4

21

**Esempio di FSC:**

**esempio 1**

ricerca di una corrispondenza tra due matrici



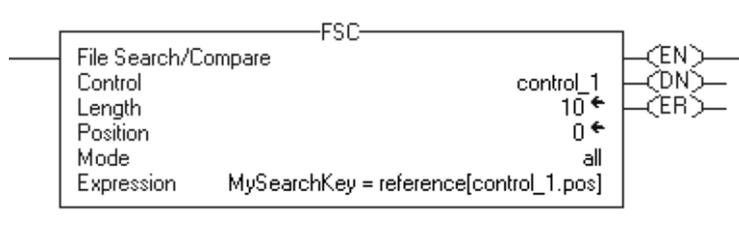
Quando è abilitata l'istruzione FSC confronta ognuno dei primi 10 elementi di *array\_1* con i corrispondenti elementi di *array\_2*.

<i>array_1</i>	<i>array_2</i>	<i>control_3.pos</i>
00000000000000000000000000000000	00000000000000000000000000000000	0
00000000000000000000000000000000	00000000000000000000000000000000	1
00000000000000000000000000000000	00000000000000000000000000000000	2
00000000000000000000000000000000	00000000000000000000000000000000	3
00000000000000001111111111111111	11111111111111110000000000000000	4
11111111111111111111111111111111	11111111111111111111111111111111	5
11111111111111111111111111111111	11111111111111111111111111111111	6
11111111111111111111111111111111	11111111111111111111111111111111	7
11111111111111111111111111111111	11111111111111111111111111111111	8
11111111111111111111111111111111	11111111111111111111111111111111	9

L'istruzione FSC rivela che questi elementi sono diversi. L'istruzione imposta i bit .FD e .IN. Il valore .POS (4) indica la posizione degli elementi che sono diversi. Per continuare a confrontare il resto della matrice, bisogna azzerare il bit .IN.

**esempio 2**

ricerca di una corrispondenza in una matrice



Quando è abilitata, l'istruzione FSC confronta *MySearchKey* con i 10 elementi di *array\_1*.

*MySearchKey*

11111111111111111000000000000000

reference	control_3.pos
00000000000000000000000000000000	0
00000000000000000000000000000000	1
00000000000000000000000000000000	2
00000000000000000000000000000000	3
11111111111111111000000000000000	4
11111111111111111111111111111111	5
11111111111111111111111111111111	6
11111111111111111111111111111111	7
11111111111111111111111111111111	8
11111111111111111111111111111111	9

L'istruzione FSC rivela che questo elemento di matrice è uguale a *MySearchKey*. L'istruzione imposta i bit .FD ed .IN. Il valore .POS (4) indica la posizione degli elementi uguali. Per continuare a confrontare il resto della matrice, bisogna azzerare il bit .IN.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>FSC( control , lenght , position , mode , expression ) ;</code>
testo ASCII	<code>FSC control lenght position mode expression</code>

**Istruzioni correlate:** CMP, CPT, FAL

Le espressioni delle istruzioni FSC vanno programmate nello stesso modo delle espressioni delle istruzioni CMP. Per informazioni su operatori validi, formato ed ordine delle operazioni comuni ad entrambi le istruzioni, utilizzare le seguenti sezioni.

## Operatori validi

Operatore:	Descrizione:	Ottimale:	Operatore:	Descrizione:	Ottimale:
+	somma	DINT, REAL	COS	coseno	REAL
-	sottrai/nega	DINT, REAL	DEG	da radianti a gradi	DINT, REAL
*	moltiplica	DINT, REAL	FRD	da BCD a intero	DINT
/	dividi	DINT, REAL	LN	logaritmo naturale	REAL
=	uguale	DINT, REAL	LOG	logaritmo in base 10	REAL
<	minore di	DINT, REAL	MOD	dividi modulo	DINT, REAL
<=	minore di o uguale	DINT, REAL	NOT	complemento di bit	DINT
>	maggiore di	DINT, REAL	OR	OR di bit	DINT
>=	maggiore di o uguale a	DINT, REAL	RAD	da gradi a radianti	DINT, REAL
<>	diverso	DINT, REAL	SIN	seno	REAL
**	potenza (x alla y)	DINT, REAL	SQR	radice quadrata	DINT, REAL
ABS	valore assoluto	DINT, REAL	TAN	tangente	REAL
ACS	arcocoseno	REAL	TOD	da intero a BCD	DINT
AND	AND di bit	DINT	TRN	tronca	DINT, REAL
ASN	arcoseno	REAL	XOR	OR esclusivo di bit	DINT
ATN	arcotangente	REAL			

## Formattazione delle espressioni

Per ciascun operatore utilizzato in un'espressione, è necessario fornire uno o due operandi (tag o valori immediati). Per la formattazione degli operatori e degli operandi di un'espressione, utilizzare la seguente tabella:

Per operatori con:	Utilizzare questo formato:	Esempi:
un operando	operatore(operando)	<i>ABS(tag_a)</i>
due operandi	operando_a operatore operando_b	<ul style="list-style-type: none"> <li>• <i>tag_b + 5</i></li> <li>• <i>tag_c AND tag_d</i></li> <li>• <i>(tag_e ** 2) MOD (tag_f / tag_g)</i></li> </ul>

## Determinazione dell'ordine delle operazioni

Le operazioni che vengono scritte nell'espressione sono eseguite dall'istruzione in un determinato ordine, non necessariamente nell'ordine con cui sono state scritte. È possibile forzare l'ordine delle operazioni raggruppando i termini in parentesi e forzando l'istruzione ad eseguire un'operazione all'interno delle parentesi prima delle altre operazioni.

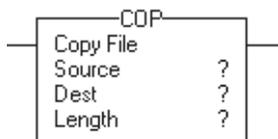
Operazioni di ordine uguale vengono eseguite da sinistra a destra.

<b>Ordine:</b>	<b>Funzionamento:</b>
1.	()
2.	ABS, ACS, ASN, ATN, COS, DEG, FRD, LN, LOG, RAD, SIN, SQR, TAN, TOD, TRN
3.	**
4.	– (nega), NOT
5.	*, /, MOD
6.	<, <=, >, >=, =
7.	– (sottrai), +
8.	AND
9.	XOR
10.	OR

## Copia file (COP)

L'istruzione COP è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> struttura	tag	elemento iniziale da copiare <b>Importante:</b> per evitare risultati inaspettati gli operandi Source e Destination devono essere dello stesso tipo di dati
Destination	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> struttura	tag	elemento iniziale che Source deve sovrascrivere <b>Importante:</b> per evitare risultati inaspettati gli operandi Source e Destination devono essere dello stesso tipo di dati
Lenght	DINT	immediato tag	numero di elementi di destinazione da copiare

**Descrizione:** L'istruzione COP copia i valori da Source in Destination. Il valore Source rimane invariato. Il numero di byte copiati è:

Numero byte = Lenght \* (numero di byte nel tipo di dati della destinazione)

**ATTENZIONE!**

**ATTENZIONE:** Se il numero di byte è superiore alla lunghezza di Source, per i rimanenti elementi verranno copiati dei dati imprevedibili.



COP opera su una memoria di dati contigui ed esegue una copia della memoria byte per byte, il che richiede una certa conoscenza della configurazione della memoria del controllore. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

L'istruzione COP non scrive oltre la fine della matrice. Se il valore Lenght è maggiore del numero totale di elementi della matrice di destinazione, l'istruzione COP si arresta alla fine della matrice stessa. Non viene generato alcun errore grave.

**Esecuzione:****Condizione:****Azione:**

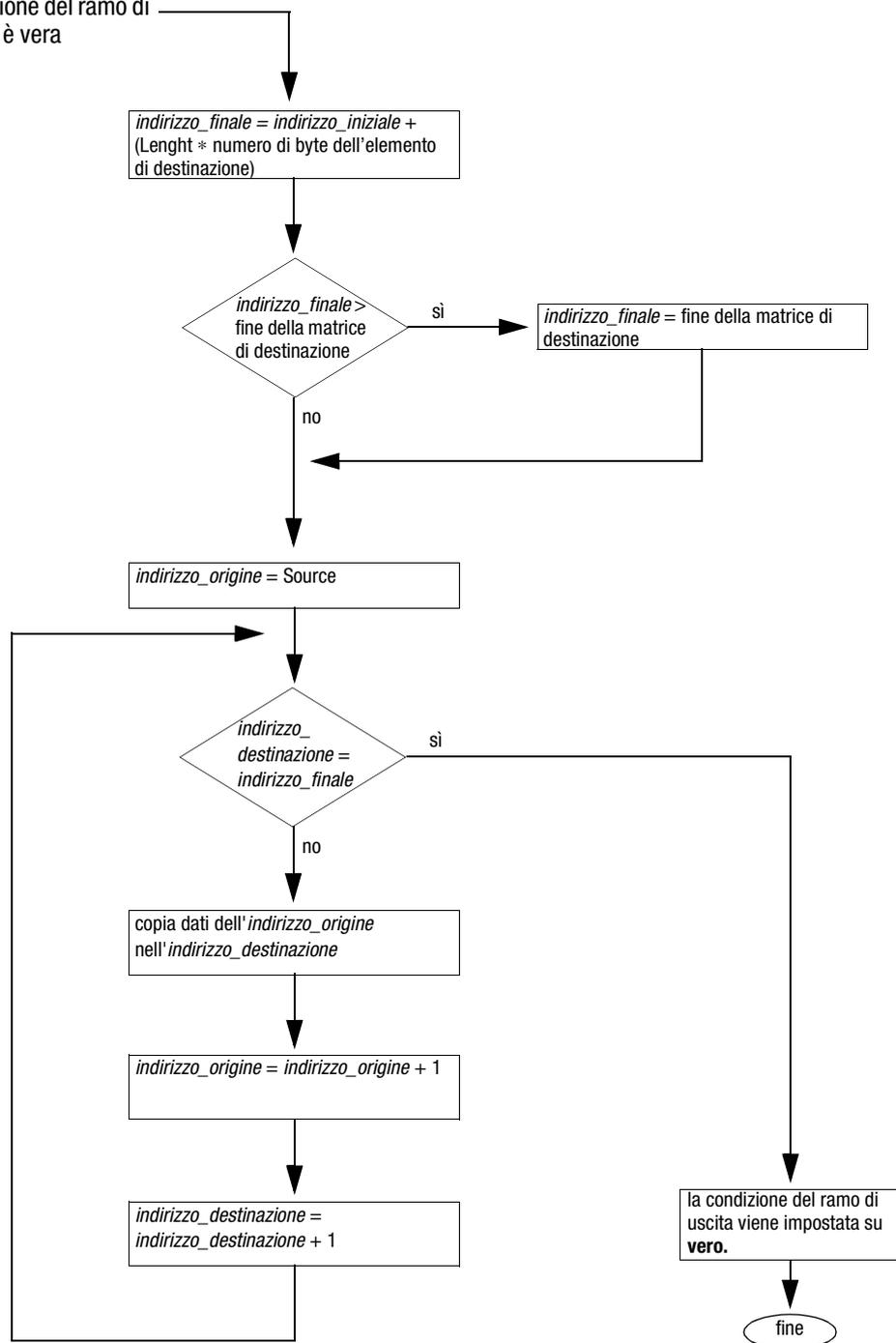
prescazione

La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è falsa

La condizione del ramo di uscita viene impostata su falso.

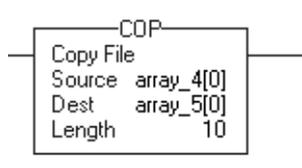
la condizione del ramo di ingresso è vera

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

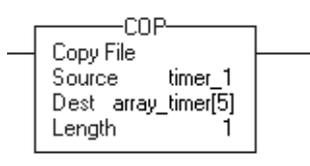
**Esempio di COP:**

**esempio 1**



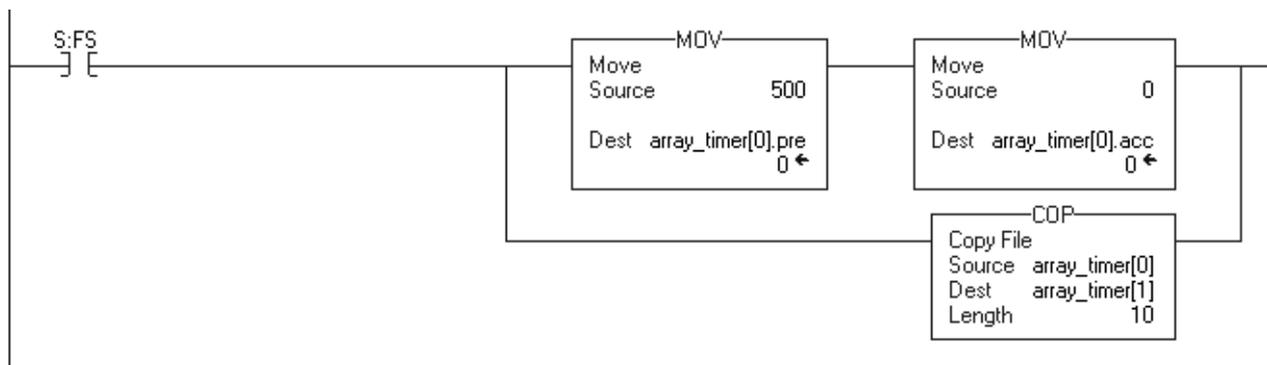
Sia *array\_4* che *array\_5* sono dello stesso tipo di dati. Se è abilitata, l'istruzione COP copia i primi 10 elementi di *array\_4* nei primi 10 elementi di *array\_5*.

**esempio 2**

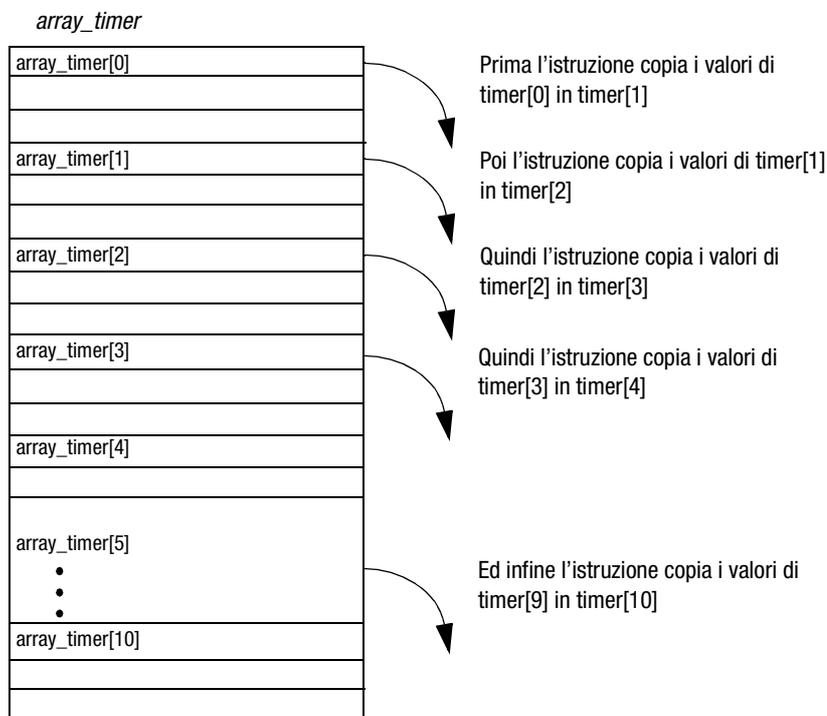


Se è abilitata, l'istruzione COP copia la struttura *timer\_1* nell'elemento 5 di *array\_timer*. L'istruzione copia solamente una struttura in un elemento della matrice.

## esempio 3



In questo esempio viene inizializzata una matrice di strutture timer. Se è abilitata, l'istruzione MOV inizializza i valori .PRE e .ACC del primo elemento di *array\_timer*. Se è abilitata, l'istruzione COP copia un blocco contiguo di byte, a partire da *array\_timer[0]*. La lunghezza è di nove strutture di timer.

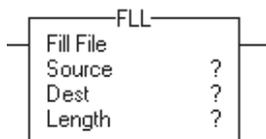
**Altri formati:**

Formato:	Sintassi:
testo neutro	<i>COP(source, destination, length);</i>
testo ASCII	<i>COP source destination length</i>

**Istruzioni correlate:** FAL, FLL, MOV

**Riempi file (FLL)**

L'istruzione FLL è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b>	immediato tag	elemento da copiare <b>Importante:</b> per evitare risultati inaspettati gli operandi Source e Destination devono essere dello stesso tipo di dati
Destination	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b>  struttura	tag	elemento iniziale che Source deve sovrascrivere <b>Importante:</b> per evitare risultati inaspettati gli operandi Source e Destination devono essere dello stesso tipo di dati Il modo migliore per inizializzare una struttura è di utilizzare un'istruzione COP.
Lenght	DINT	immediato	numero di elementi da riempire

**Descrizione:** L'istruzione FLL riempie gli elementi di una matrice con il valore Source. Il valore Source rimane invariato. Il numero di byte riempiti è:

Numero byte = Lenght \* (numero di byte nel tipo di dati della destinazione)

L'istruzione FLL opera su una memoria di dati contigui. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

L'istruzione FLL non scrive oltre la fine della matrice. Se il valore Lenght è maggiore del numero totale di elementi della matrice di destinazione, l'istruzione FLL si arresta alla fine della matrice stessa. Non viene generato alcun errore grave.

Per ottenere i migliori risultati i valori Source e Destination devono essere dello stesso tipo di dati. Se si desidera riempire una struttura, utilizzare l'istruzione COP (vedere esempio 3 a pagina 7-31). Se per Source e Destination si usano tipi di dati diversi, gli elementi di destinazione vengono riempiti con i valori Source convertiti.

Se il valore Source è:	Il valore Destination è:	Il valore Source viene convertito in:
SINT, INT, DINT o REAL	SINT	SINT
SINT, INT, DINT o REAL	INT	INT
SINT, INT, DINT o REAL	DINT	DINT
SINT, INT, DINT o REAL	REAL	REAL
SINT	struttura	SINT (non convertito)
INT	struttura	INT (non convertito)
DINT	struttura	DINT (non convertito)
REAL	struttura	REAL (non convertito)

**Esecuzione:****Condizione:****Azione:**

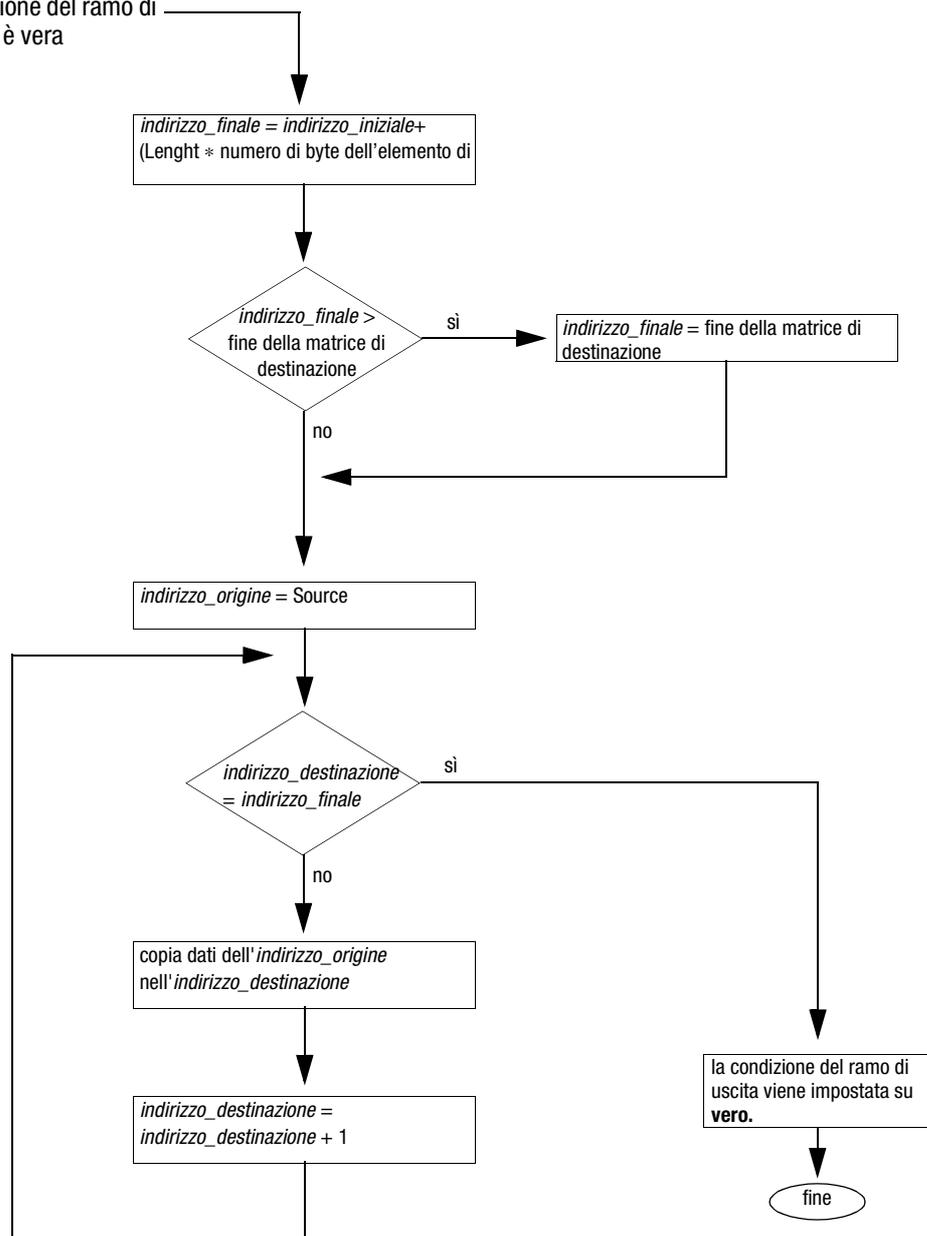
prescausione

La condizione del ramo di uscita viene impostata su falso.

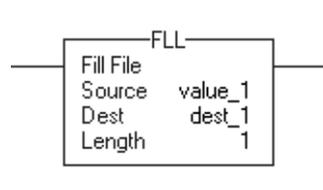
la condizione del ramo di ingresso è falsa

La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è vera

**Indicatori di stato aritmetico:** non influenzati**Condizioni di errore:** nessuna

**Esempio di FLL:**



Se è abilitata, l'istruzione FLL copia i dati di *value\_1* in *dest\_1*.

Tipo dati Source ( <i>value_1</i> ):	Valore Source ( <i>value_1</i> ):	Tipo dati Destination ( <i>dest_1</i> ):	Valore Destination ( <i>dest_1</i> ) dopo FLL:
SINT	16#80 (-128)	DINT	16#FFFF FF80 (-128)
DINT	16#1234 5678	SINT	16#78
SINT	16#01	REAL	1,0
REAL	2,0	INT	16#0002
SINT	16#01	TIMER	16#0101 0101 16#0101 0101 16#0101 0101
INT	16#0001	TIMER	16#0001 0001 16#0001 0001 16#0001 0001
DINT	16#0000 0001	TIMER	16#0000 0001 16#0000 0001 16#0000 0001

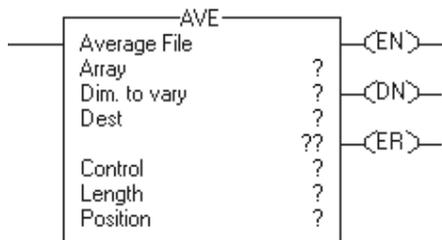
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>FLL(<i>source</i>,<i>destination</i>,<i>length</i>);</code>
testo ASCII	<code>FLL <i>source destination length</i></code>

**Istruzioni correlate:** FAL, COP, MOV

**Media file (AVE)**

L'istruzione AVE è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Array	SINT INT DINT <b>REAL</b>	tag matrice	calcola la media dei valori di questa matrice specifica il primo elemento del gruppo di elementi su cui calcolare la media <b>non</b> usare CONTROL.POS nell'indice
Dimension to vary	DINT	immediato (0, 1, 2)	quale dimensione usare a seconda del numero delle dimensioni, l'ordine è array[dim_0,dim_1,dim_2] array[dim_0,dim_1] array[dim_0]
Destination	SINT INT DINT <b>REAL</b>	tag	risultato dell'operazione
Control	CONTROL	tag	struttura di controllo per l'operazione
Lenght	DINT	immediato	numero di elementi della matrice su cui calcolare la media
Position	DINT	immediato	elemento corrente della matrice il valore iniziale generalmente è 0

**Struttura CONTROL:**

Mnemonic:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione AVE è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando l'istruzione ha operato sull'ultimo elemento di Array (.POS = .LEN).
.ER	BOOL	Il bit di errore viene impostato se l'istruzione genera un overflow. L'istruzione interrompe la sua esecuzione fino a quando il programma non azzeri il bit .ER. La posizione dell'elemento che ha provocato l'overflow viene memorizzata nel valore .POS.
.LEN	DINT	La lunghezza specifica il numero di elementi della matrice su cui opera l'istruzione.
.POS	DINT	Contiene la posizione dell'elemento corrente a cui l'istruzione accede.

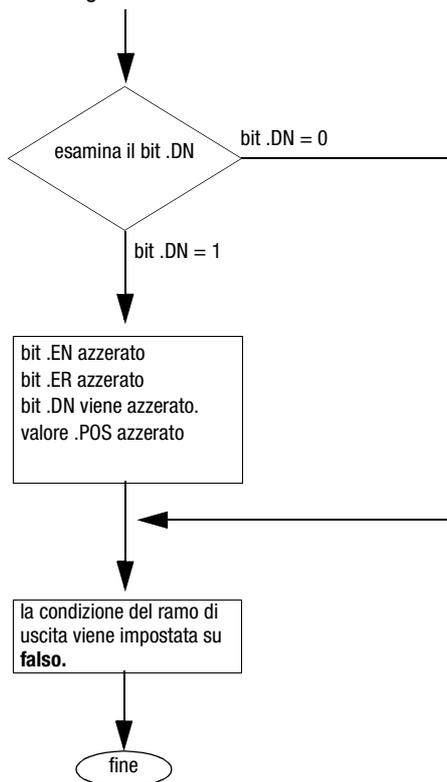
**Descrizione:** L'istruzione AVE calcola la media di un gruppo di valori.

**Importante:** assicurarsi che il valore Lenght non porti l'istruzione a superare la dimensione da variare specificata. In questo caso, la destinazione sarà errata. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN viene azzerato. Il bit .DN viene azzerato. Il bit .ER viene azzerato. La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è falsa

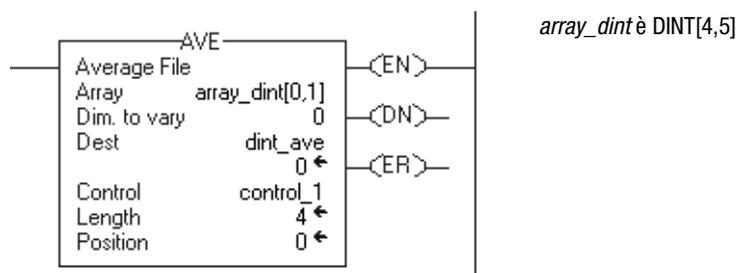


la condizione del ramo di ingresso è vera	L'istruzione AVE calcola la media sommando tutti gli elementi specificati nella matrice e dividendo per il numero di elementi. Internamente, l'istruzione utilizza un'istruzione FAL per calcolare la media: Espressione = calcolo della media Modalità = ALL Per ulteriori informazioni sul funzionamento dell'istruzione FAL, vedere pagina 7-8.
---	--

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:**

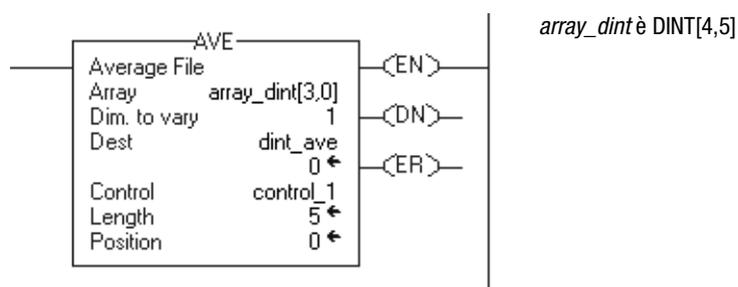
Si verifica un errore grave se:	Tipo errore:	Codice errore:
.POS < 0 oppure .LEN < 0	4	21
la dimensione da variare non esiste per la matrice specificata	4	20

**Esempio di AVE:****esempio 1**

		dimensione 1				
		Indici				
		0	1	2	3	4
dimensione 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

$$AVE = \frac{19 + 14 + 9 + 4}{4} = \frac{46}{4} = 11.5$$

$$dint\_ave = 12$$

**esempio 2**

		dimensione 1				
		Indici				
		0	1	2	3	4
dimensione 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

$$AVE = \frac{5 + 4 + 3 + 2 + 1}{5} = \frac{15}{5} = 3$$

$$dint\_ave = 3$$

**Altri formati:**

**Formato:**

**Sintassi:**

---

testo neutro	<i>AVE(array,dim_to_vary,destination,control,lenght,position);</i>
--------------	--

---

testo ASCII	<i>AVE array dim_to_vary destination control lenght position</i>
-------------	--

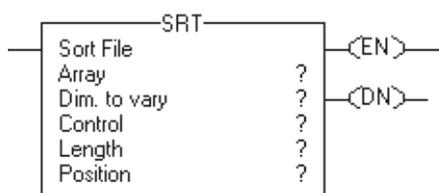
---

**Istruzioni correlate:** SRT, STD

## Ordina file (SRT)

L'istruzione SRT è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Array	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b>	tag matrice	matrice da ordinare specifica il primo elemento del gruppo di elementi da ordinare <b>non</b> usare CONTROL.POS nell'indice
Dimension to vary	DINT	immediato (0, 1, 2)	quale dimensione usare a seconda del numero delle dimensioni, l'ordine è array[dim_0,dim_1,dim_2] array[dim_0,dim_1] array[dim_0]
Control	CONTROL	tag	struttura di controllo per l'operazione
Length	DINT	immediato	numero di elementi della matrice da ordinare
Position	DINT	immediato	elemento corrente della matrice il valore iniziale generalmente è 0

### Struttura CONTROL:

Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione SRT è abilitata.
.DN	BOOL	Il bit di fine è impostato quando gli elementi specificati sono stati ordinati.
.ER	BOOL	Il bit di errore è impostato quando .LEN < 0 o quando .POS < 0. Una qualsiasi di queste condizioni genera anche un errore grave.
.LEN	DINT	La lunghezza specifica il numero di elementi della matrice su cui opera l'istruzione.
.POS	DINT	Contiene la posizione dell'elemento corrente a cui l'istruzione accede.

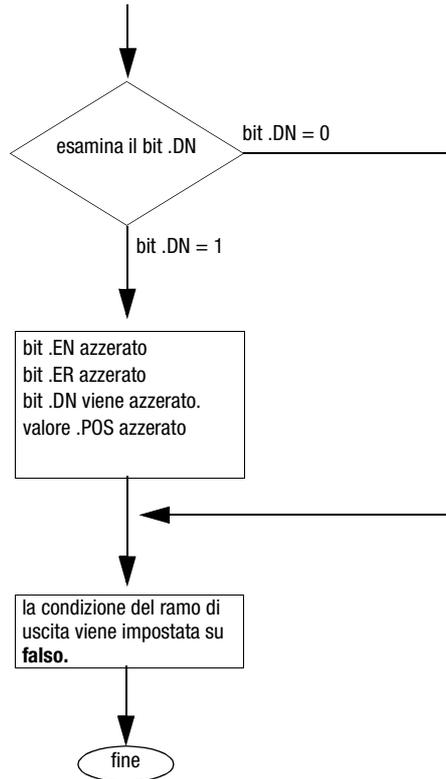
**Descrizione:** L'istruzione SRT ordina in modo crescente un gruppo di valori di una dimensione (Dim da variare) della Matrice.

**Importante:** assicurarsi che il valore Length non porti l'istruzione a superare la dimensione da variare specificata. Se ciò accadesse, si potrebbero verificare dei risultati inaspettati. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN viene azzerato. Il bit .DN viene azzerato. Il bit .ER viene azzerato. La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è falsa

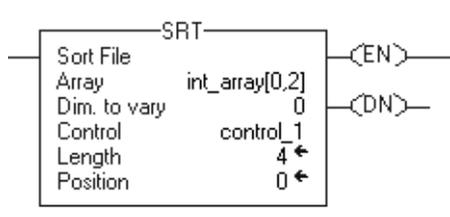


la condizione del ramo di ingresso è vera L'istruzione SRT ordina in modo crescente gli elementi specificati della matrice.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:**

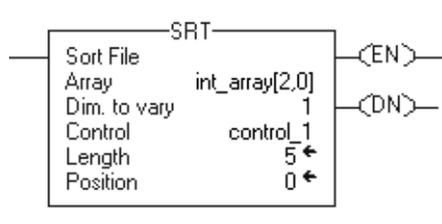
Si verifica un errore grave se:	Tipo errore:	Codice errore:
.POS < 0 oppure .LEN < 0	4	21
la dimensione da variare non esiste per la matrice specificata	4	20
L'istruzione tenta di avere accesso ai dati posti all'esterno dei limiti della matrice	4	20

**Esempio di SRT:****esempio 1***int\_array* è DINT[4,5]**Prima**

		dimensione 1				
		indici				
		0	1	2	3	4
dimensione 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

**Dopo**

		dimensione 1				
		indici				
		0	1	2	3	4
dimensione 0	0	20	19	3	17	16
	1	15	14	8	12	11
	2	10	9	13	7	6
	3	5	4	18	2	1

**esempio 2***int\_array* è DINT[4,5]**Prima**

		dimensione 1				
		indici				
		0	1	2	3	4
dimensione 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

**Dopo**

		dimensione 1				
		indici				
		0	1	2	3	4
dimensione 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	6	7	8	9	10
	3	5	4	3	2	1

**Altri formati:****Formato:****Sintassi:**


---

 testo neutro     *SRT(array,dim\_to\_vary,control,length,position);*


---

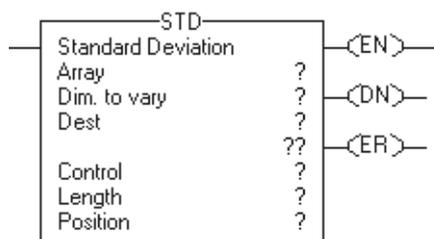
 testo ASCII     *SRT array dim\_to\_vary control length position*


---

**Istruzioni correlate:** AVE, STD

**Deviazione standard file (STD)** L'istruzione STD è un'istruzione di uscita.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
Array	SINT INT <b>DINT</b> <b>REAL</b>	tag matrice	calcola la deviazione standard dei valori di una matrice specifica il primo elemento del gruppo di elementi da utilizzare per il calcolo della deviazione standard <b>non</b> usare CONTROL.POS nell'indice

Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag.A-6.

Dimension to vary	DINT	immediato (0, 1, 2)	quale dimensione usare a seconda del numero delle dimensioni, l'ordine è array[dim_0,dim_1,dim_2] array[dim_0,dim_1] array[dim_0]
Destination	REAL	tag	risultato dell'operazione
Control	CONTROL	tag	struttura di controllo per l'operazione
Length	DINT	immediato	numero di elementi della matrice da utilizzare per il calcolo della deviazione standard
Position	DINT	immediato	elemento corrente della matrice il valore iniziale generalmente è 0

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione STD è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando il calcolo è stato completato.
.ER	BOOL	Il bit di errore viene impostato se l'istruzione genera un overflow. L'istruzione interrompe la sua esecuzione fino a quando il programma non azzeri il bit .ER. La posizione dell'elemento che ha provocato l'overflow viene memorizzata nel valore .POS.
.LEN	DINT	La lunghezza specifica il numero di elementi della matrice su cui opera l'istruzione.
.POS	DINT	Contiene la posizione dell'elemento corrente a cui l'istruzione accede.

**Descrizione:** L'istruzione STD calcola la deviazione standard di un insieme di valori di una dimensione della matrice e memorizza il risultato nella destinazione.

**Importante:** assicurarsi che il valore Length non porti l'istruzione a superare la dimensione da variare specificata. In questo caso la destinazione sarà errata. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

La deviazione standard viene calcolata in base alla seguente formula:

$$\text{Deviazione standard} = \sqrt{\frac{\sum_{i=1}^N [\langle X_{(start+i)} - AVE \rangle^2]}{(N-1)}}$$

Dove:

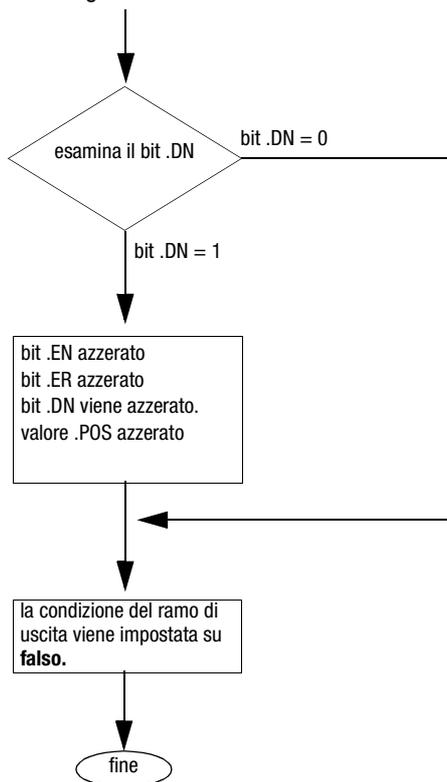
- start = indice della dimensione da variare dell'operando della matrice
- $x_i$  = elemento variabile della matrice
- N = numero di elementi specificati della matrice
- AVE =

$$\frac{\left( \sum_{i=1}^N x_{(start+i)} \right)}{N}$$

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN viene azzerato. Il bit .DN viene azzerato. Il bit .ER viene azzerato. La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è falsa



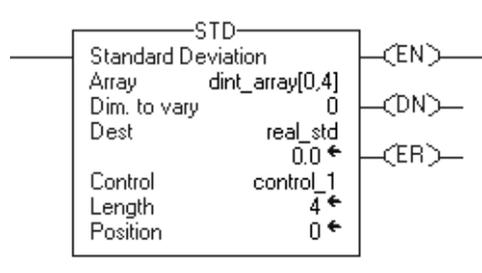
la condizione del ramo di ingresso è vera

L'istruzione STD calcola la deviazione standard degli elementi specificati. Internamente, l'istruzione utilizza un'istruzione FAL per calcolare la media:  
Espressione = calcolo della deviazione standard  
Modalità = ALL  
Per ulteriori informazioni sul funzionamento dell'istruzione FAL, vedere pagina 7-8.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
.POS < 0 oppure .LEN < 0	4	21
la dimensione da variare non esiste per la matrice specificata	4	20

**Esempio di STD:****esempio 1***dint\_array* è DINT[4,5]

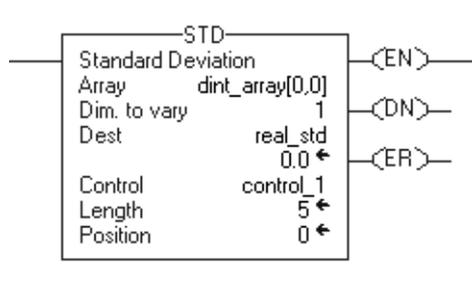
dimensione 0

indici	dimensione 1				
	0	1	2	3	4
0	20	19	18	17	16
1	15	14	13	12	11
2	10	9	8	7	6
3	5	4	3	2	1

$$AVE = \frac{16 + 11 + 6 + 1}{4} = \frac{34}{4} = 8.5$$

$$STD = \sqrt{\frac{\langle 16 - 8.5 \rangle^2 + \langle 11 - 8.5 \rangle^2 + \langle 6 - 8.5 \rangle^2 + \langle 1 - 8.5 \rangle^2}{\langle 4 - 1 \rangle}} = 6.454972$$

$$real\_std = 6,454972$$

**esempio 2***dint\_array* è DINT[4,5]

dimensione 0

indici	dimensione 1				
	0	1	2	3	4
0	20	19	18	17	16
1	15	14	13	12	11
2	10	9	8	7	6
3	5	4	3	2	1

$$AVE = \frac{20 + 19 + 18 + 17 + 16}{5} = \frac{90}{5} = 18$$

$$STD = \sqrt{\frac{\langle 20 - 18 \rangle^2 + \langle 19 - 18 \rangle^2 + \langle 18 - 18 \rangle^2 + \langle 17 - 18 \rangle^2 + \langle 16 - 18 \rangle^2}{\langle 5 - 1 \rangle}} = 1.581139$$

$$real\_std = 1,581139$$

**Altri formati:****Formato:****Sintassi:**


---

testo neutro     *STD(array, dim\_to\_vary, destination, control, lenght, position);*


---

testo ASCII     *STD array dim\_to\_vary destination control lenght position*


---

**Istruzioni correlate:** AVE, SRT

**Note:**

## Istruzioni di Matrice (File)/Scorrimento (BSL, BSR, FFL, FFU, LFL, LFU)

### Introduzione

Utilizzare le istruzioni di matrice (file)/scorrimento per modificare la posizione dei dati all'interno delle matrici.

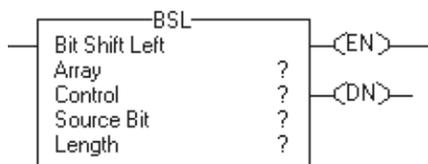
Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
Caricare, spostare e scaricare bit da una matrice di bit, un bit alla volta.	BSL	8-2
	BSR	8-5
Caricare e scaricare valori nello stesso ordine.	FFL	8-8
	FFU	8-14
Caricare e scaricare valori nell'ordine inverso.	LFL	8-20
	LFU	8-26

È possibile mischiare vari tipi di dati, ma si potrebbero verificare errori di precisione e di arrotondamento.

I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente ed utilizzando meno memoria se tutti gli operandi di un'istruzione utilizzano il medesimo tipo di dati ottimale, generalmente DINT o REAL.

**Sposta bit a sinistra (BSL)**

L'istruzione BSL è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Array	DINT	tag matrice	matrice da modificare specifica il primo elemento del gruppo di elementi <b>non</b> usare CONTROL.POS nell'indice
Control	CONTROL	tag	struttura di controllo per l'operazione
Source bit	BOOL	tag	bit da spostare
Length	DINT	immediato	numero di bit della matrice da spostare

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione BSL è abilitata.
.DN	BOOL	Il bit di fine è impostato per indicare che i bit sono stati spostati di una posizione a sinistra.
.UL	BOOL	Il bit di scaricamento è l'uscita dell'istruzione. Il bit .UL memorizza lo stato del bit che è stato spostato fuori dell'intervallo dei bit.
.ER	BOOL	Il bit di errore viene impostato quando .LEN < 0.
.LEN	DINT	La lunghezza specifica il numero di bit della matrice da spostare.

**Descrizione:** L'istruzione BSL sposta i bit specificati della matrice di una  
posizione verso sinistra. Se è abilitata, l'istruzione scarica il primo dei bit  
specificati al bit .UL, fa scorrere i restanti bit di una posizione a sinistra e  
carica il Source bit nel bit 0 della matrice.

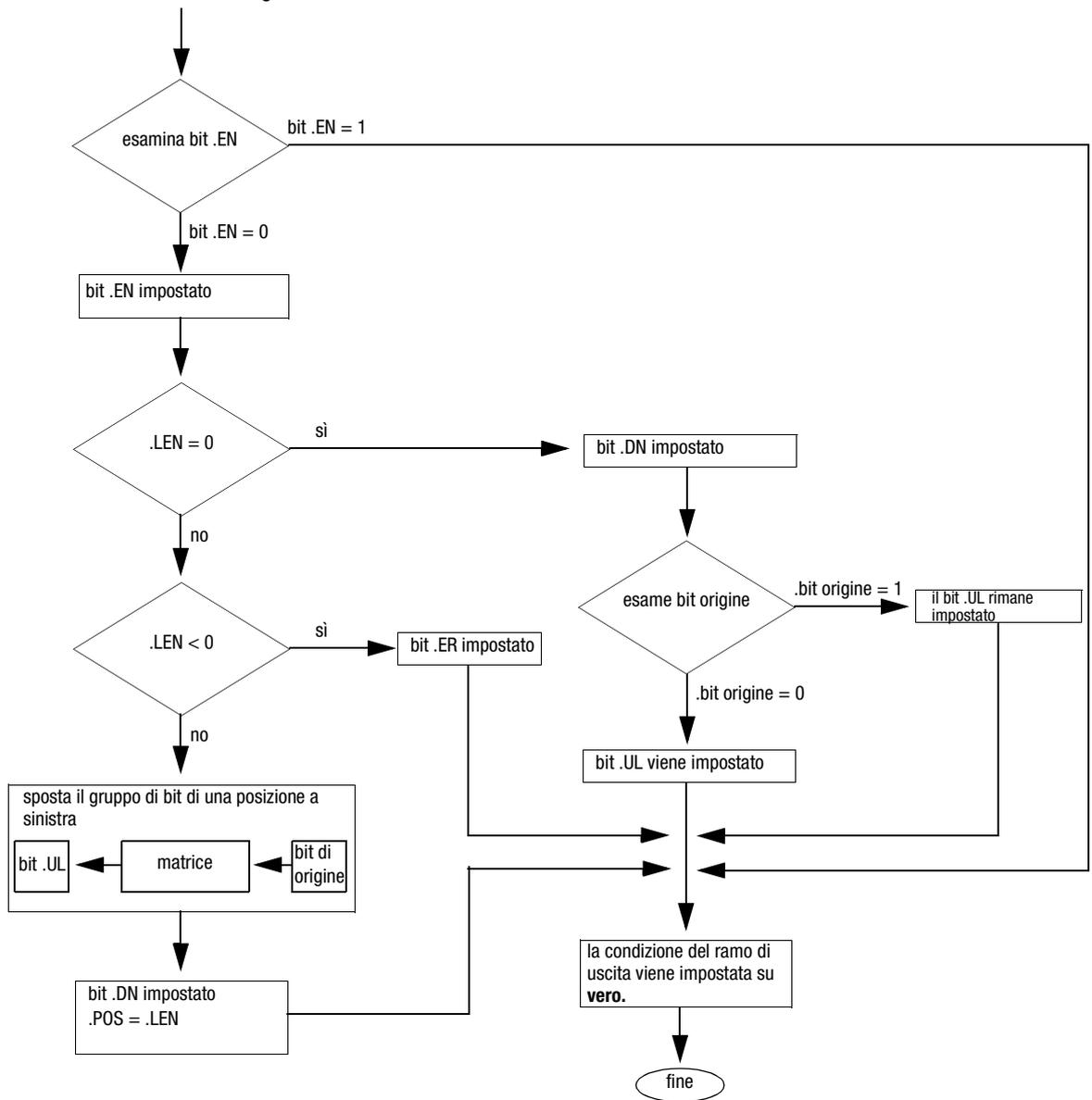
L'istruzione BSL opera su una memoria contigua. Per ulteriori informazioni,  
vedere *Visualizzazione di una matrice come blocco di memoria* a  
pagina B-4.

**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN viene azzerato. Il bit .DN viene azzerato. Il bit .ER viene azzerato. Il valore .POS viene azzerato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit .EN viene azzerato. Il bit .DN viene azzerato. Il bit .ER viene azzerato. Il valore .POS viene azzerato. La condizione del ramo di uscita viene impostata su falso.

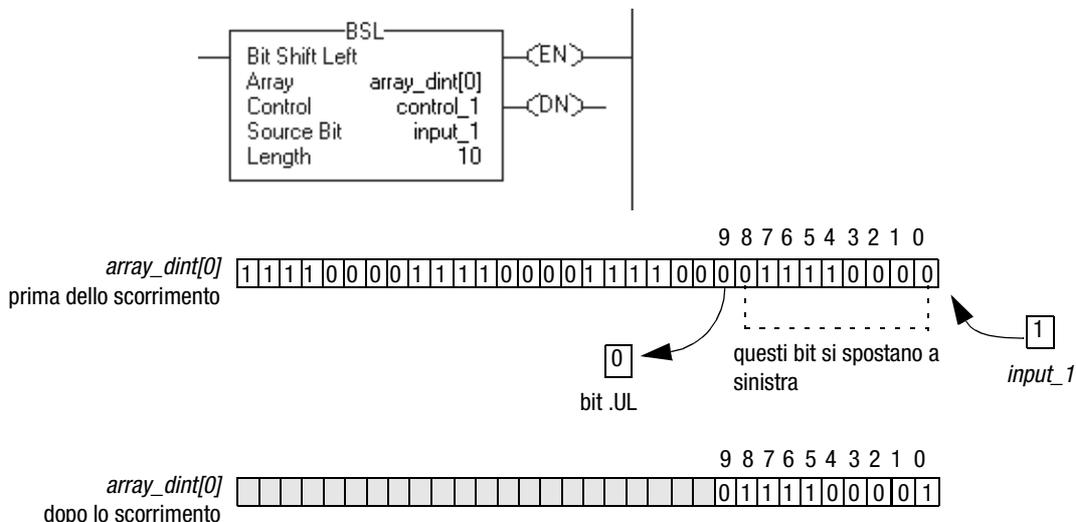
**Condizione:****Azione:**

la condizione del ramo di ingresso è vera

**Indicatori di stato aritmetico:** non influenzati**Condizioni di errore:** nessuna

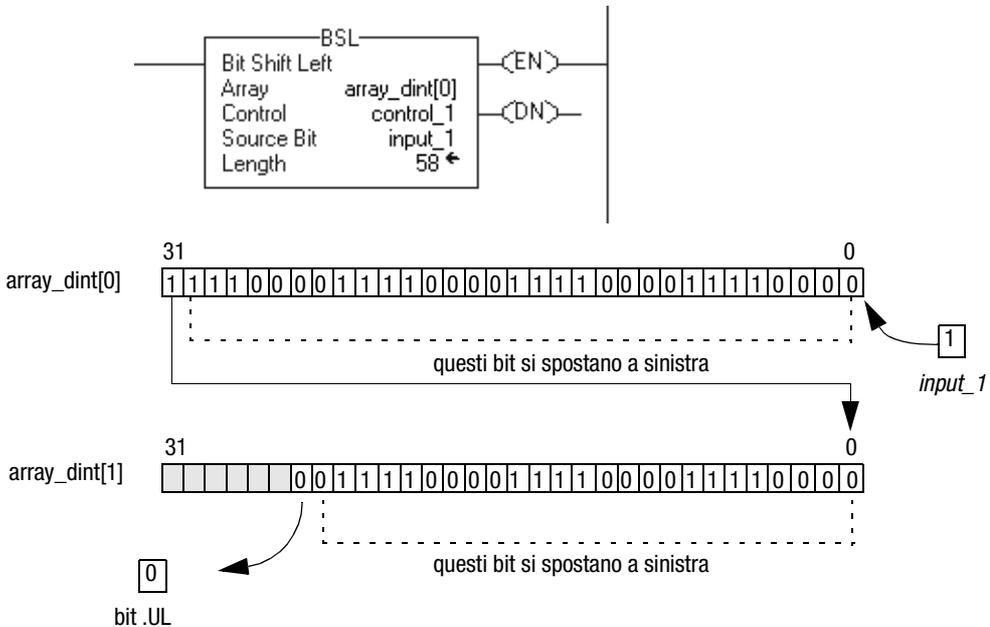
**Esempio di BSL:**

**esempio 1**



Se è abilitata, l'istruzione BSL inizia dal bit 0 di *array\_dint[0]*. L'istruzione scarica *array\_dint[0].9* nel bit .UL, sposta i rimanenti bit e carica *input\_1* in *array\_dint[0].0*. I valori dei rimanenti bit (10-31) non sono validi.

**esempio 2**



Se è abilitata, l'istruzione BSL inizia dal bit 0 di *array\_dint[0]*. L'istruzione scarica *array\_dint[1].25* nel bit .UL, sposta i rimanenti bit e carica *input\_1* in *array\_dint[0].0*. I valori dei rimanenti bit (31-26 di *array\_dint[1]*) non sono validi. Si noti come *array\_dint[0].31* si sposti tra le parole fino a *array\_dint[1].0*.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>BSL(array, control, source_bit, length);</code>
testo ASCII	<code>BSL matrice controllo source_bit length</code>

**Istruzioni correlate:** BSR

**Sposta bit a destra (BSL)**

L'istruzione BSR è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Array	DINT	tag matrice	matrice da modificare specifica l'elemento da cui iniziare lo scorrimento <b>non</b> usare CONTROL.POS nell'indice
Control	CONTROL	tag	struttura di controllo per l'operazione
Source bit	BOOL	tag	bit da spostare
Length	DINT	immediato	numero di bit della matrice da spostare

**Struttura CONTROL:**

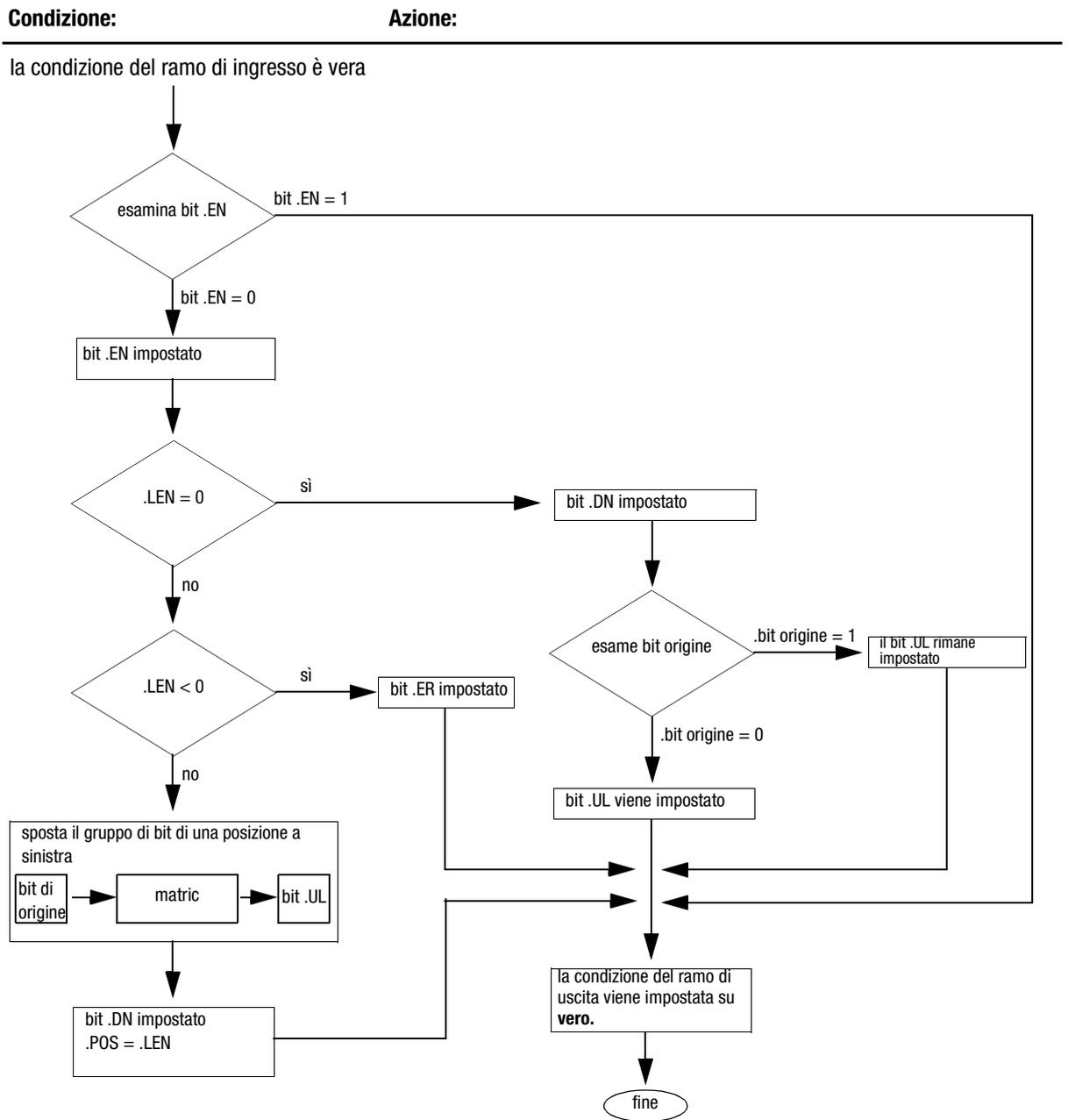
Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione BSR è abilitata.
.DN	BOOL	Il bit di fine è impostato per indicare che i bit sono stati spostati di una posizione a destra.
.UL	BOOL	Il bit di scaricamento è l'uscita dell'istruzione. Il bit .UL memorizza lo stato del bit che è stato spostato fuori dell'intervallo dei bit.
.ER	BOOL	Il bit di errore viene impostato quando .LEN < 0.
.LEN	DINT	La lunghezza specifica il numero di bit della matrice da spostare.

**Descrizione:** L'istruzione BSR sposta i bit specificati della matrice di una  
posizione verso destra. Se è abilitata, l'istruzione scarica il valore del bit 0  
della matrice nel bit .UL, sposta i restanti bit di una posizione a destra e  
carica il bit di origine nel primo bit degli bit specificati.

L'istruzione BSR opera su una memoria contigua. Per ulteriori  
informazioni, vedere *Visualizzazione di una matrice come blocco di  
memoria* a pagina B-4.

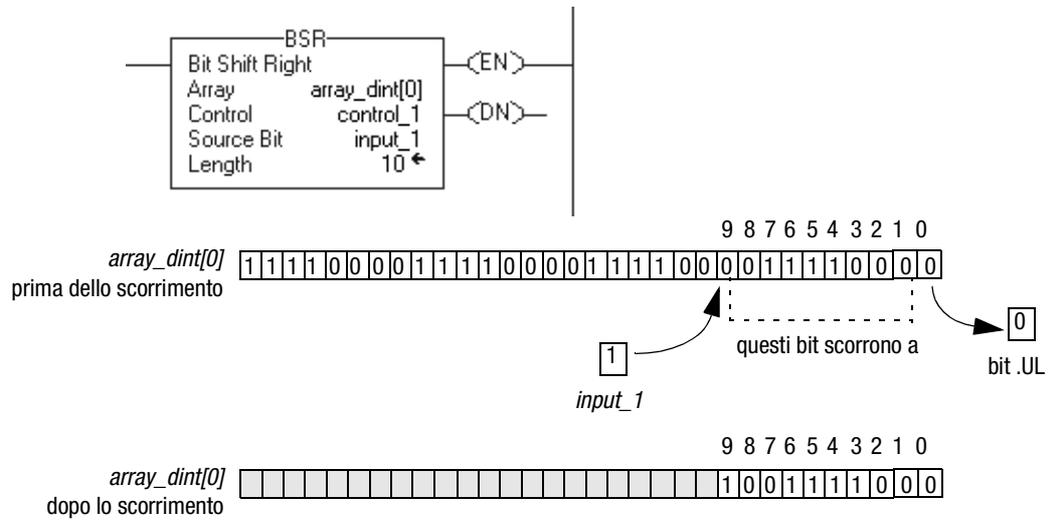
**Esecuzione:**

Condizione:	Azione:
prescansione	Il bit .EN viene azzerato. Il bit .DN viene azzerato. Il bit .ER viene azzerato. Il valore .POS viene azzerato. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit .EN viene azzerato. Il bit .DN viene azzerato. Il bit .ER viene azzerato. Il valore .POS viene azzerato. La condizione del ramo di uscita viene impostata su falso.

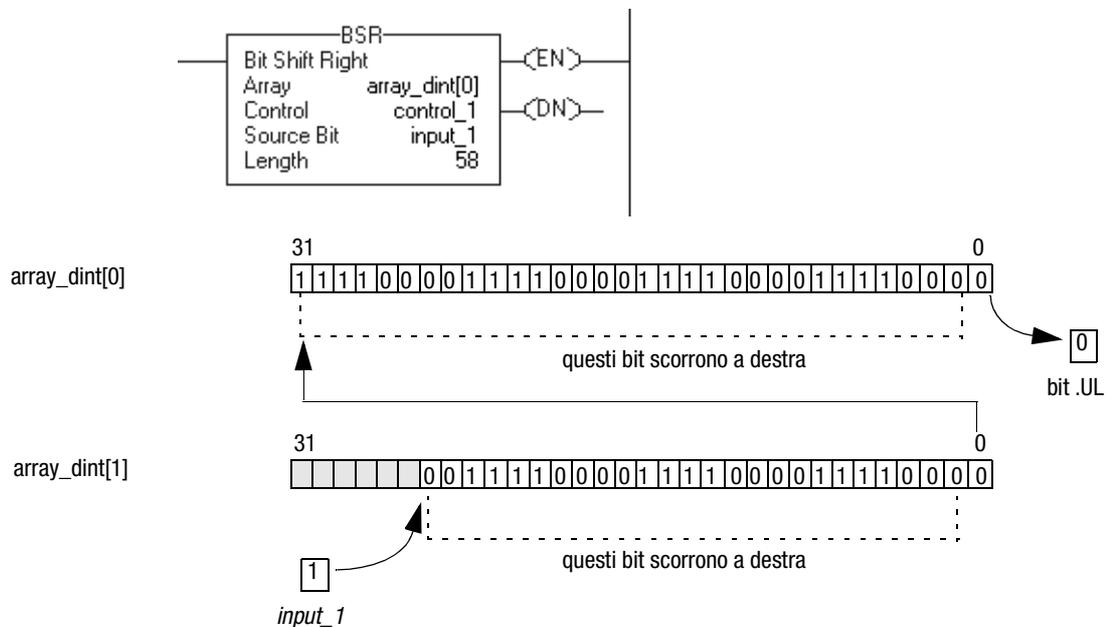


**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di BSR:****esempio 1**

Se è abilitata, l'istruzione BSR inizia dal bit 9 di *array\_dint[0]*. L'istruzione scarica *array\_dint[0].0* nel bit .UL, sposta i rimanenti bit a destra e carica *input\_1* in *array\_dint[0].9*. I valori dei rimanenti bit (10-31) non sono validi.

**esempio 2**

Se è abilitata, l'istruzione BSR inizia dal bit 25 di *array\_dint[1]*. L'istruzione scarica *array\_dint[0].0* nel bit .UL, sposta i rimanenti bit a destra e carica *input\_1* in *array\_dint[1].25*. I valori dei rimanenti bit (31-26 di *array\_dint*) non sono validi. Si noti come *array\_dint[1].0* scorra attraverso le parole in *array\_dint[0].31*.

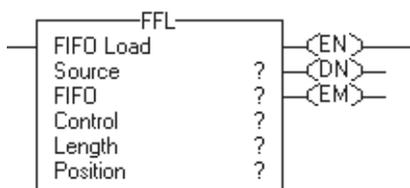
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>BSR(array, control, source_bit, length);</code>
testo ASCII	<code>BSR matrice controllo source_bit length</code>

**Istruzioni correlate:** BSL

**Carica FIFO (FFL)**

L'istruzione FFL è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Origine	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	immediato tag	dati da memorizzare nel FIFO
FIFO	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	tag matrice	FIFO da modificare specifica il primo elemento del FIFO <b>non</b> usare CONTROL.POS nell'indice
Control	CONTROL	tag	struttura di controllo per l'operazione in genere, utilizzare lo stesso CONTROL del FFU associato
Length	DINT	immediato	numero massimo di elementi che il FIFO può contenere contemporaneamente
Position	DINT	immediato	posizione successiva nel FIFO in cui l'istruzione carica i dati il valore iniziale generalmente è 0

Il valore Source viene convertito nel tipo di dati della tag matrice. Un intero più piccolo viene convertito in un intero più grande mediante estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Se come tipo di dati per l'operando Source o FIFO si utilizza una struttura definita dall'utente, utilizzare la stessa struttura per entrambi gli operandi.

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione FFL è abilitata.
.DN	BOOL	Il bit di fine è impostato per indicare che il FIFO è pieno (.POS = .LEN). Il bit .DN impedisce il caricamento del FIFO finché .POS < .LEN.
.EM	BOOL	Il bit di vuoto indica che il FIFO è vuoto. Se .LEN ≤ 0 oppure .POS < 0, vengono impostati sia il bit .EM che il bit .DN.
.LEN	DINT	La lunghezza specifica il numero massimo di elementi che il FIFO può contenere contemporaneamente.
.POS	DINT	La posizione identifica la posizione del FIFO in cui l'istruzione caricherà il valore successivo.

**Descrizione:** L'istruzione FFL copia il valore di Origine nel FIFO. Utilizzare l'istruzione FFL con l'istruzione FFU per memorizzare ed estrarre dati nell'ordine primo ad entrare/primo ad uscire. Se sono usate in coppia, le istruzioni FFL e FFU formano un registro a scorrimento asincrono.

In genere Source e FIFO sono dello stesso tipo di dati.

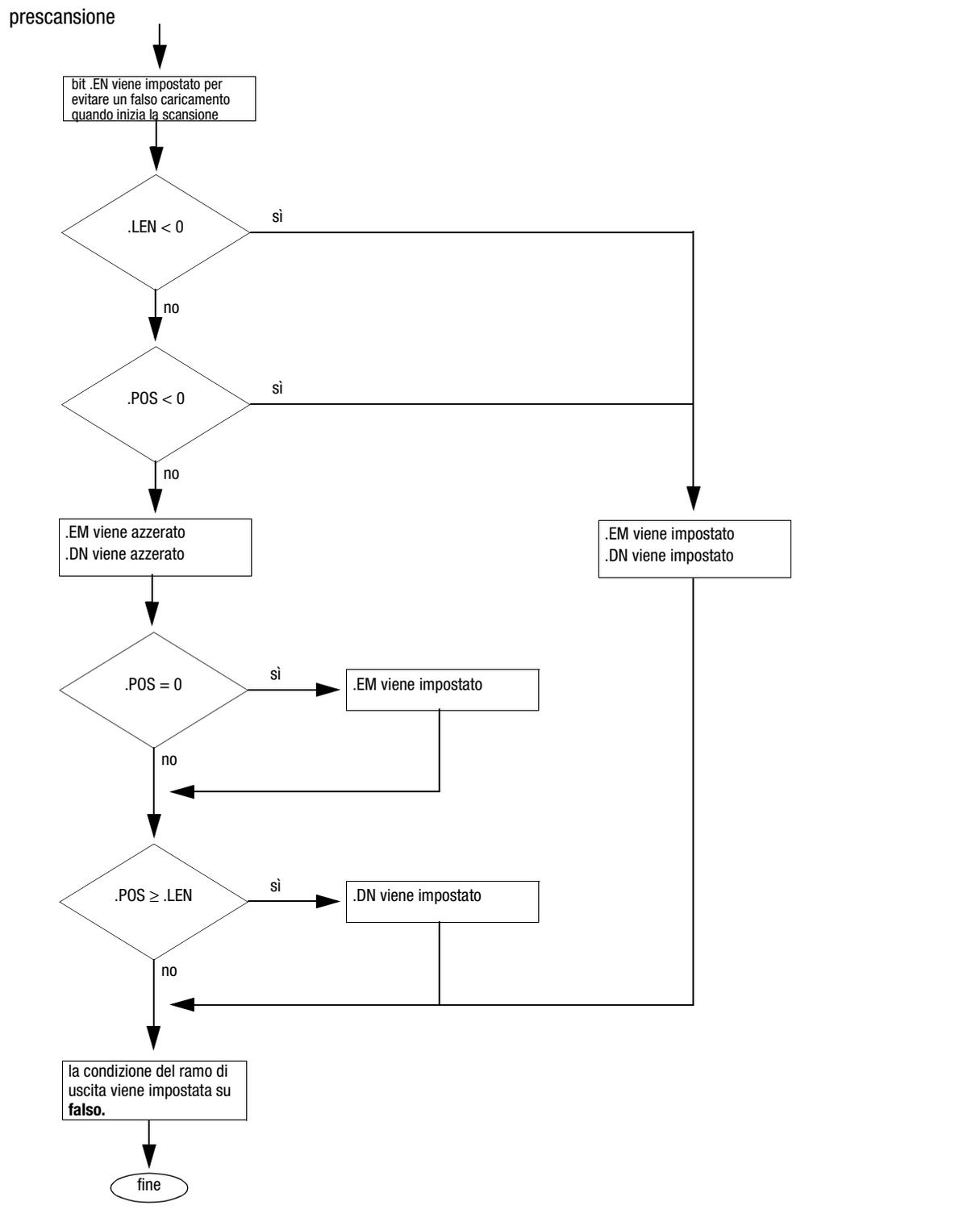
Se è abilitata, l'istruzione FFL carica il valore dell'origine nella posizione del FIFO identificata dal valore .POS. L'istruzione carica un valore ogni volta che l'istruzione è abilitata fino a quando il FIFO non è pieno.

L'istruzione FFL opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

**Esecuzione:**

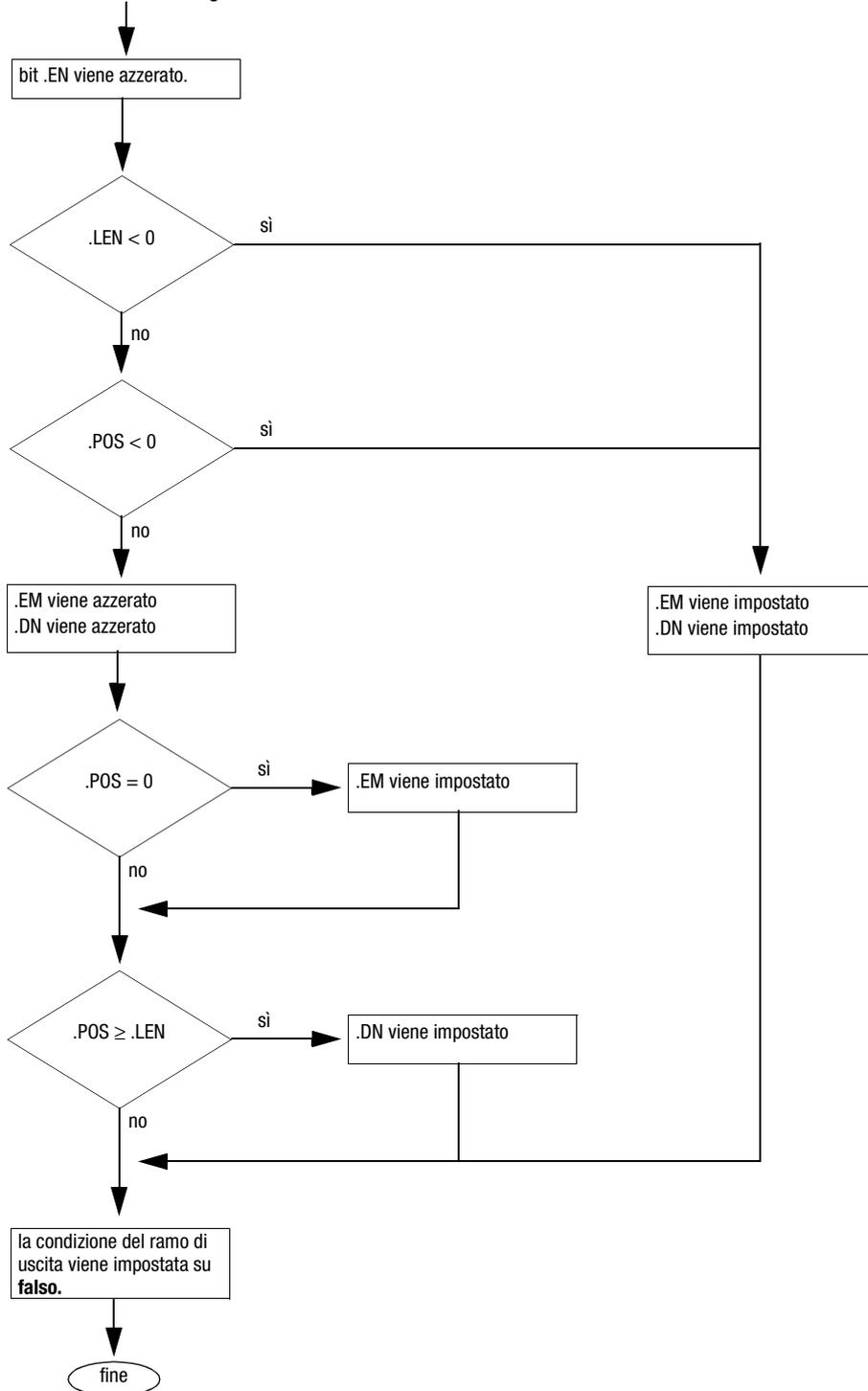
**Condizione:**

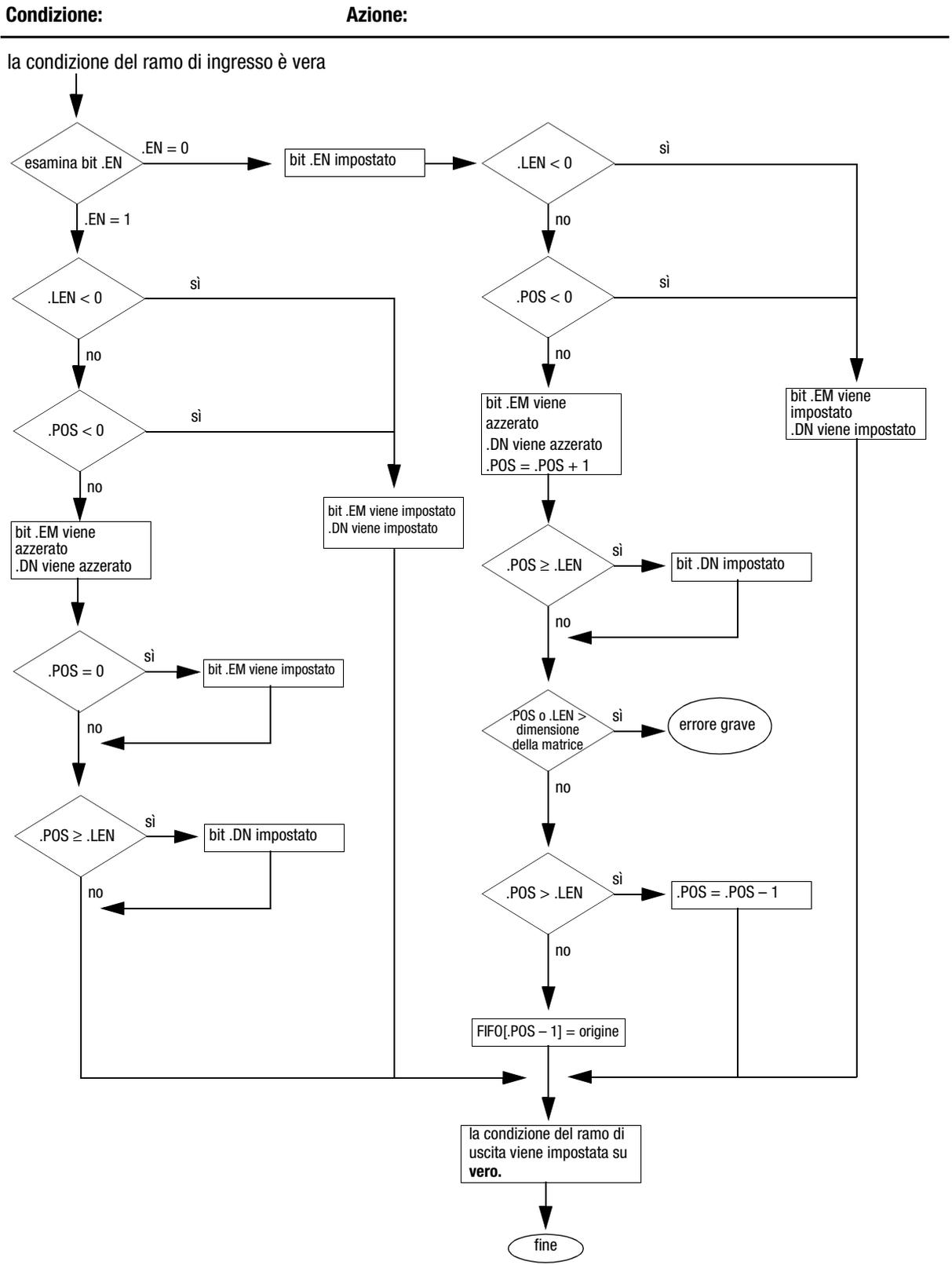
**Azione:**



**Condizione:****Azione:**

la condizione del ramo di ingresso è falsa

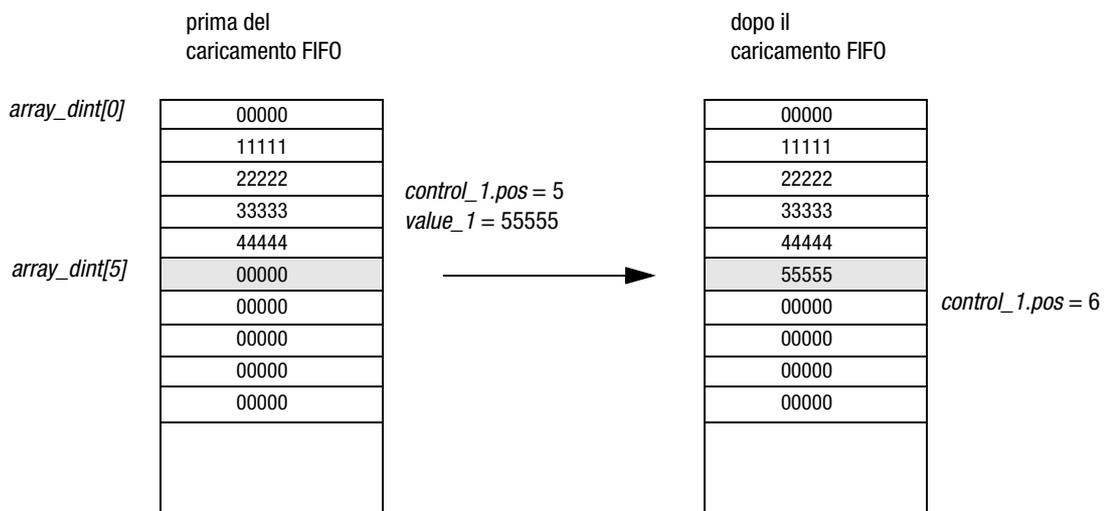
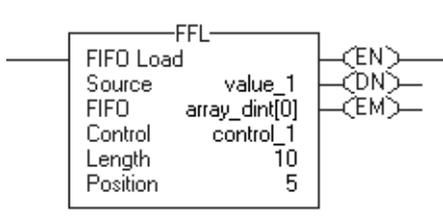




**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
(elemento iniziale + .POS) > dimensione matrice FIFO	4	20

**Esempio di FFL:**

Se è abilitata, l'istruzione FFL carica *value\_1* nella posizione successiva del FIFO, che in questo esempio è *array\_dint[5]*.

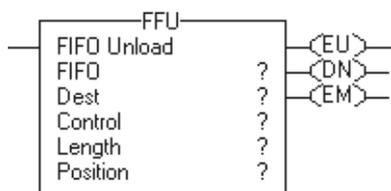
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>FFL(source, FIFO, control, length, position);</code>
testo ASCII	<code>FFL source FIFO control length position</code>

**Istruzioni correlate:** FFU, LFL, LFU

**Scarica FIFO (FFU)**

L'istruzione FFU è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
FIFO	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	tag matrice	FIFO da modificare specifica il primo elemento del FIFO <b>non</b> usare CONTROL.POS nell'indice
Destination	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	tag	valore che esce dal FIFO  Il valore Destination viene convertito nel tipo di dati della tag Destination. Un intero più piccolo viene convertito in un intero più grande mediante estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.
Control	CONTROL	tag	struttura di controllo per l'operazione in genere, utilizzare lo stesso CONTROL del FFL associato
Length	DINT	immediato	numero massimo di elementi che il FIFO può contenere contemporaneamente
Position	DINT	immediato	posizione successiva nel FIFO da cui l'istruzione scarica i dati il valore iniziale generalmente è 0

Se come tipo di dati per l'operando FIFO o Destination si utilizza una struttura definita dall'utente, utilizzare la stessa struttura per entrambi gli operandi.

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.EU	BOOL	Il bit di abilitazione scaricamento indica che l'istruzione FFU è abilitata. Il bit EU viene impostato per prevenire un falso scaricamento quando comincia la scansione del programma.
.DN	BOOL	Il bit di fine è impostato per indicare che il FIFO è pieno (.POS = .LEN).
.EM	BOOL	Il bit di vuoto indica che il FIFO è vuoto. Se .LEN ≤ 0 oppure .POS < 0, vengono impostati sia il bit .EM che il bit .DN.
.LEN	DINT	La lunghezza specifica il numero massimo di elementi del FIFO.
.POS	DINT	La posizione identifica il punto fino al quale sono stati caricati dei dati nel FIFO.

**Descrizione:** L'istruzione FFU scarica il valore dalla posizione 0 (prima posizione) del FIFO e lo memorizza nella destinazione. I restanti dati del FIFO scorrono di una posizione. Utilizzare l'istruzione FFU con l'istruzione FFL per memorizzare ed estrarre dati nell'ordine primo ad entrare/primo ad uscire.

Se il FIFO è un tipo di dati DINT, la destinazione deve essere un tipo di dati DINT; se il FIFO è un tipo di dati REAL, la destinazione deve essere un tipo di dati REAL.

Se è abilitata, l'istruzione FFU scarica i dati dal primo elemento del FIFO e li inserisce nella destinazione. L'istruzione scarica un valore ogni volta che l'istruzione è abilitata fino a quando il FIFO non è vuoto. Se il FIFO è vuoto, l'FFU invia uno 0 alla destinazione.

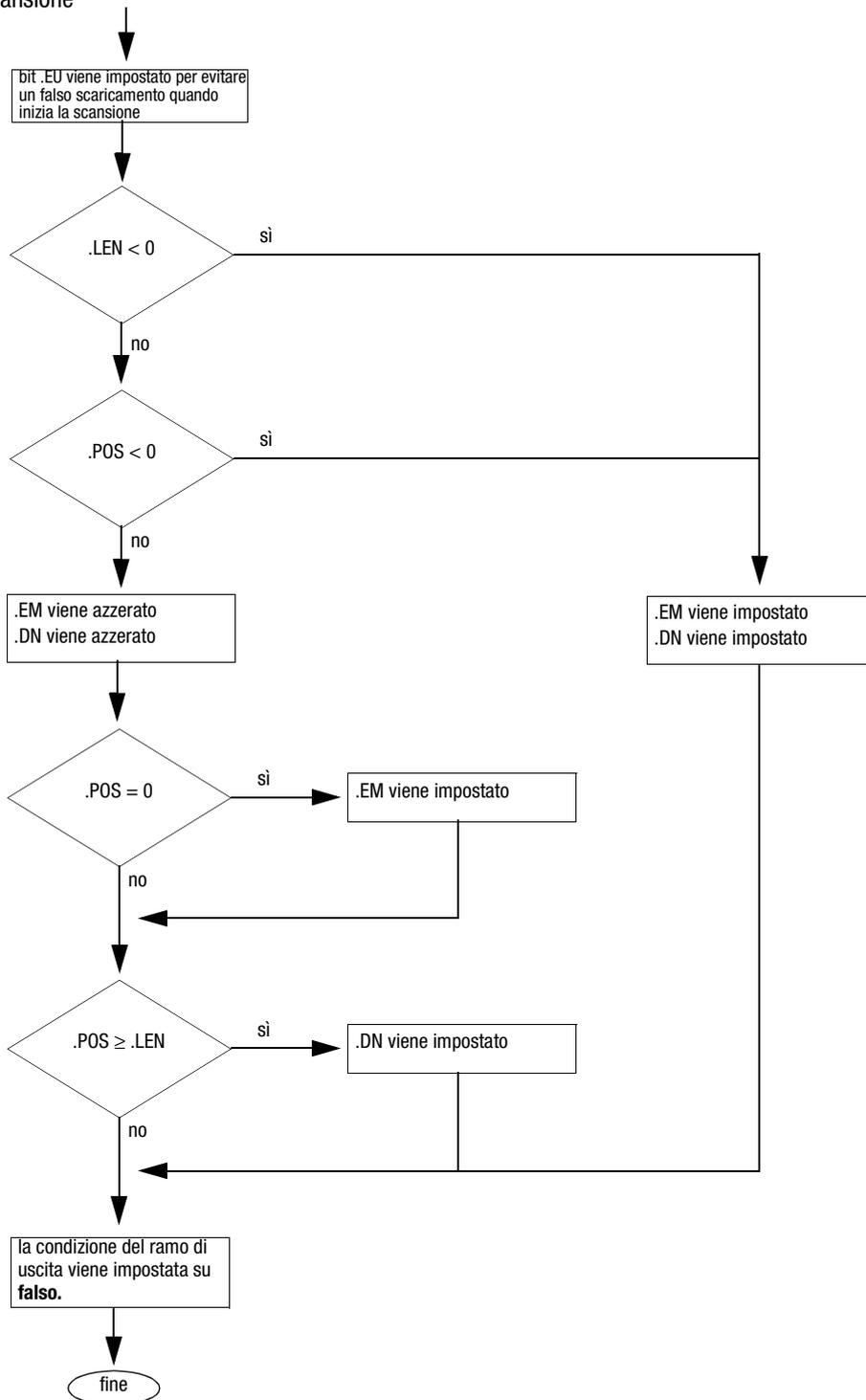
L'istruzione FFU opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

**Esecuzione:**

**Condizione:**

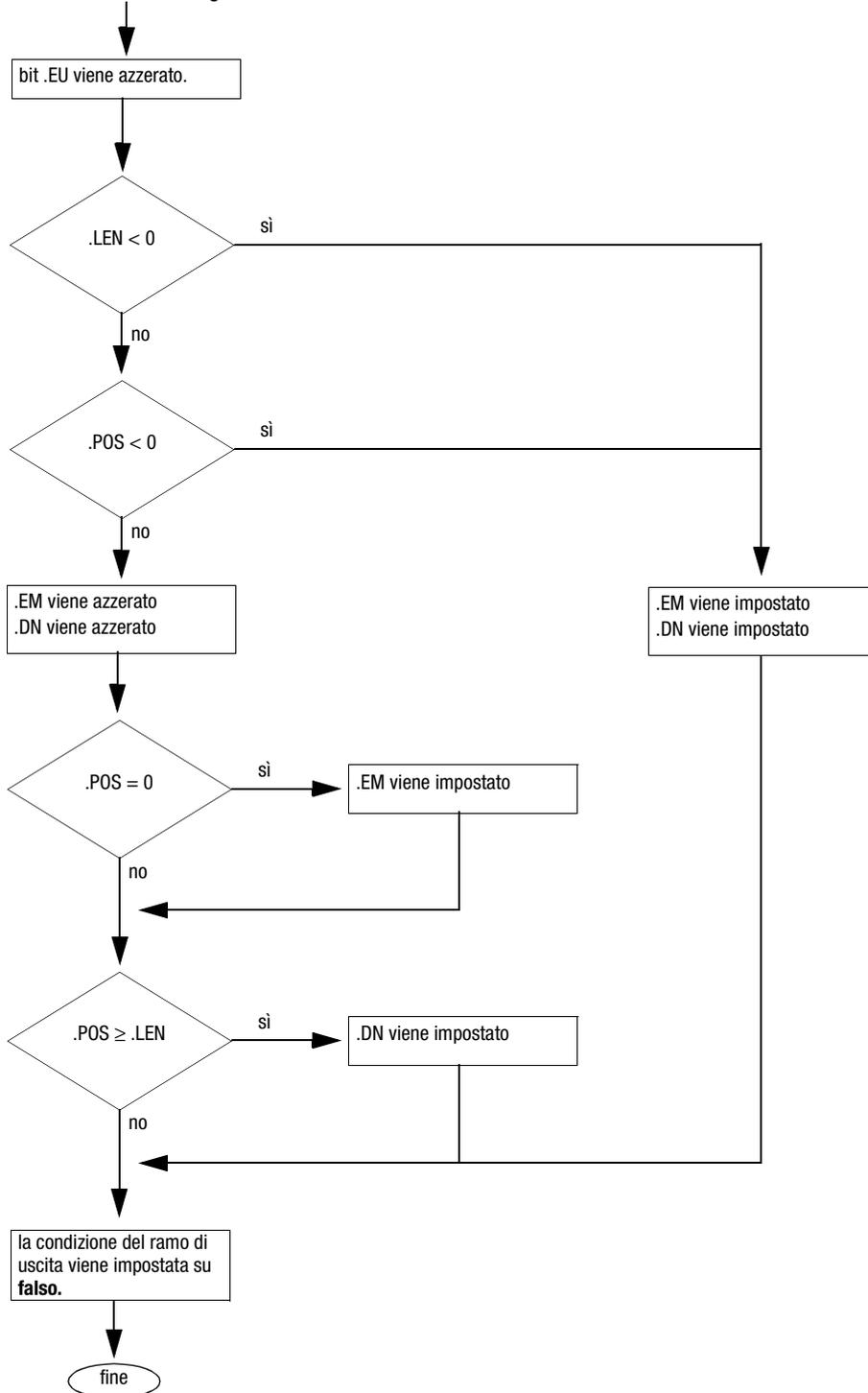
**Azione:**

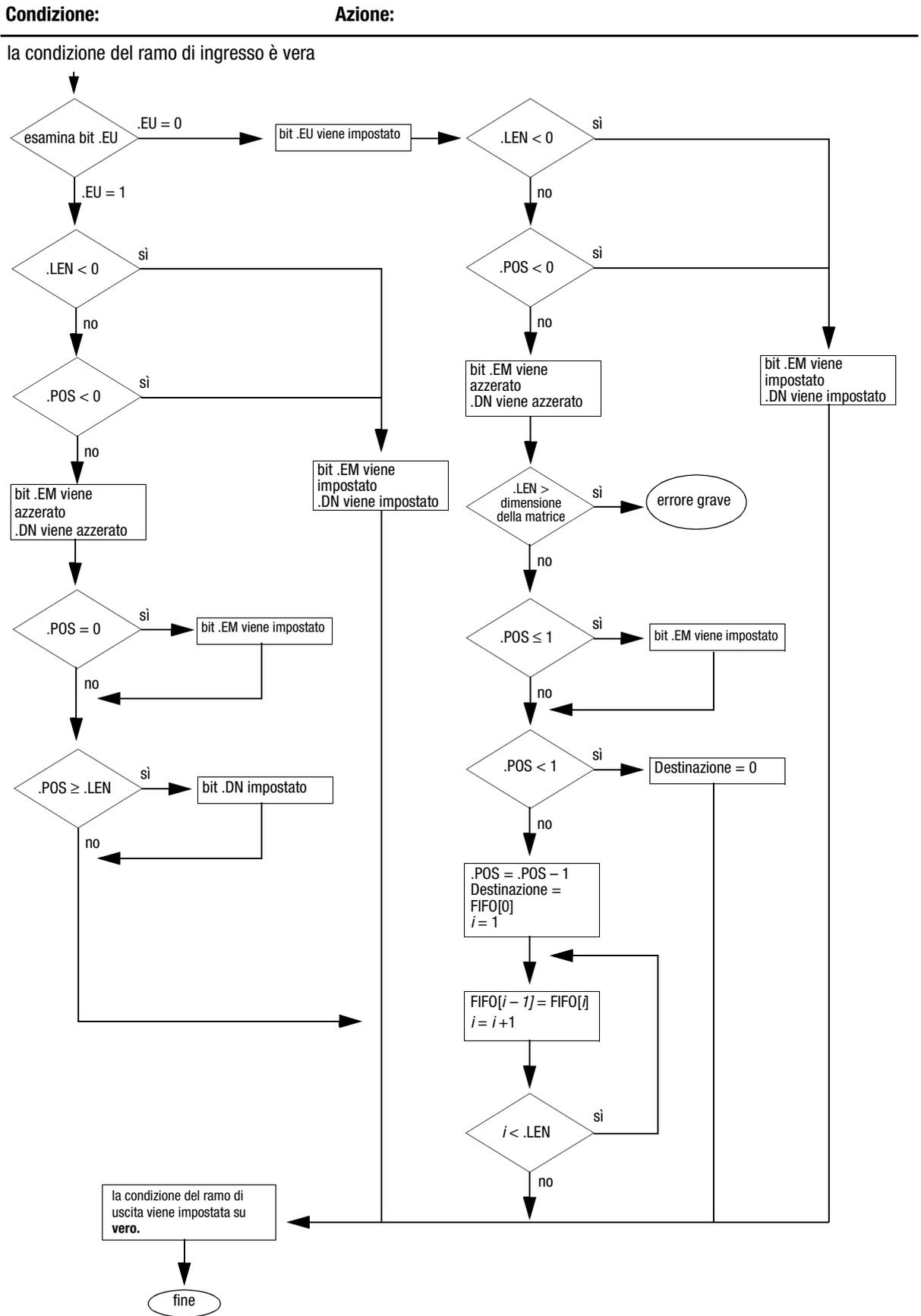
prescansione



**Condizione:****Azione:**

la condizione del ramo di ingresso è falsa

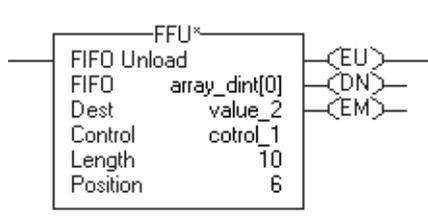




**Indicatori di stato aritmetico:** non influenzati

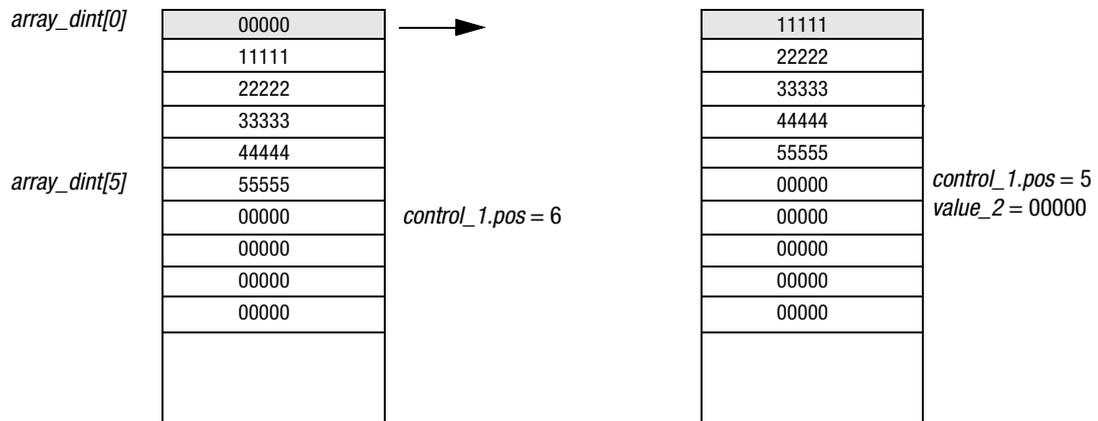
**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
Lunghezza > dimensione matrice FIFO	4	20

**Esempio di FFU:**

prima dello scaricamento FIFO

dopo lo scaricamento FIFO



Se è abilitata, l'istruzione FFU scarica *array\_dint[0]* in *value\_2* e fa scorrere i rimanenti elementi lungo *array\_dint*.

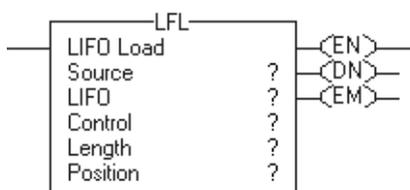
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>FFU(FIFO,destination,control,length,position);</code>
testo ASCII	<code>FFU FIFO destination control length position</code>

**Istruzioni correlate:** FFL, LFL, LFU

**Carica LIFO (FFL)**

L'istruzione LFL è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	immediato tag	dati da memorizzare nel LIFO
LIFO	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	tag matrice	LIFO da modificare specifica il primo elemento del LIFO <b>non</b> usare CONTROL.POS nell'indice
Control	CONTROL	tag	struttura di controllo per l'operazione in genere, utilizzare lo stesso CONTROL del LFU associato
Length	DINT	immediato	numero massimo di elementi che il LIFO può contenere contemporaneamente
Position	DINT	immediato	posizione successiva nel LIFO in cui l'istruzione carica i dati il valore iniziale generalmente è 0

Il valore Source viene convertito nel tipo di dati della tag matrice. Un intero più piccolo viene convertito in un intero più grande mediante estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.

Se per l'operando Source o LIFO si utilizza una struttura definita dall'utente come tipo di dati, utilizzare la stessa struttura per entrambi gli operandi.

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione LFL è abilitata.
.DN	BOOL	Il bit di fine è impostato per indicare che il LIFO è pieno (.POS = .LEN). Il bit .DN impedisce il caricamento del LIFO finché .POS < .LEN.
.EM	BOOL	Il bit vuoto indica che il LIFO è vuoto. Se .LEN ≤ 0 oppure .POS < 0, vengono impostati sia il bit .EM che il bit .DN.
.LEN	DINT	La lunghezza specifica il numero massimo di elementi che il LIFO può contenere contemporaneamente.
.POS	DINT	La posizione identifica la posizione del LIFO in cui l'istruzione caricherà il valore successivo.

**Descrizione:** L'istruzione LFL copia il valore di Origine nel LIFO. Utilizzare l'istruzione LFL con l'istruzione LFU per memorizzare ed estrarre dati nell'ordine ultimo ad entrare/primo ad uscire. Se sono usate in coppia, le istruzioni LFL e LFU formano un registro a scorrimento asincrono.

In genere Source e LIFO sono dello stesso tipo di dati.

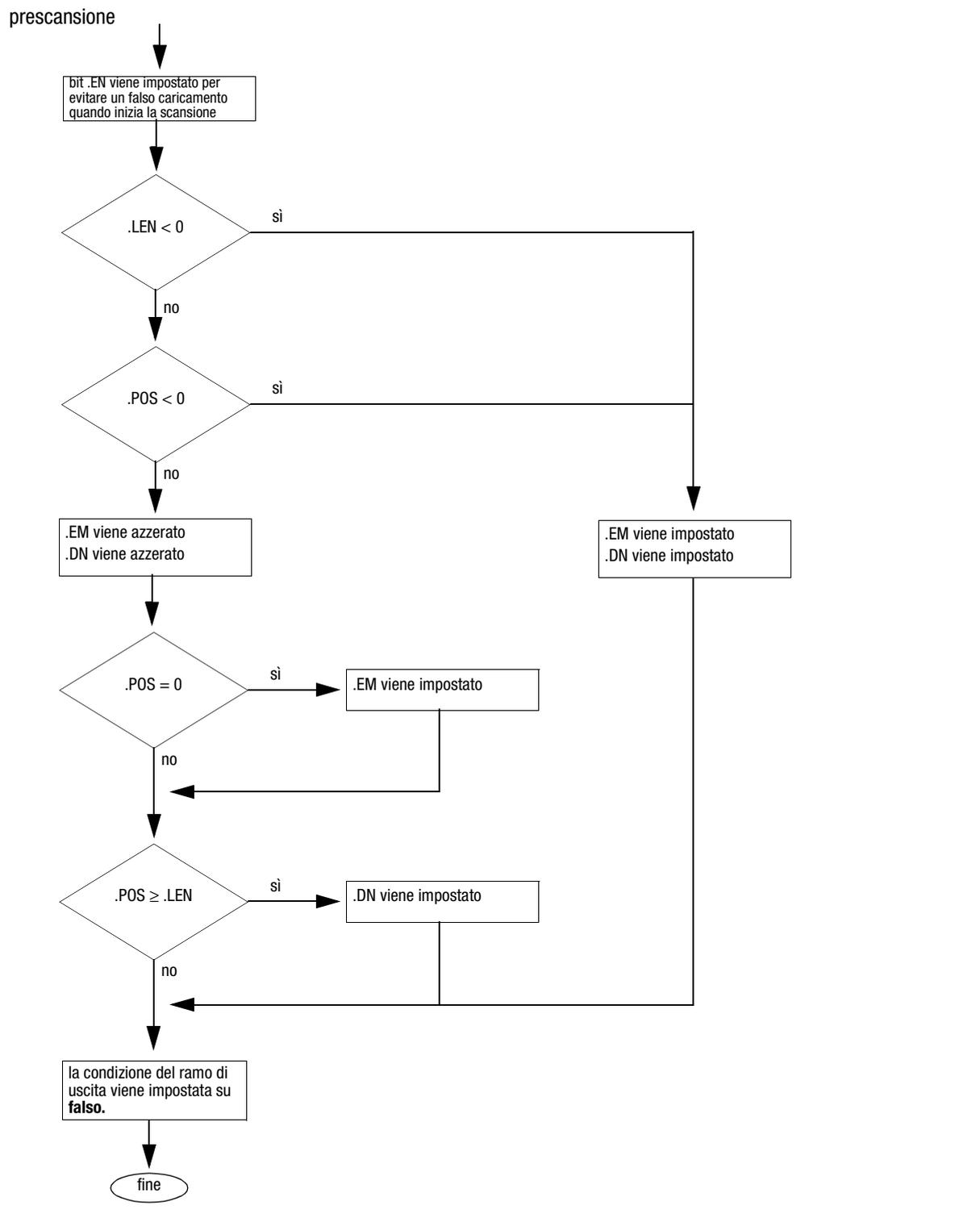
Se è abilitata, l'istruzione LFL carica il valore dell'origine nella posizione del LIFO identificata dal valore .POS. L'istruzione carica un valore ogni volta che l'istruzione è abilitata fino a quando il LIFO non è vuoto.

L'istruzione LFL opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

**Esecuzione:**

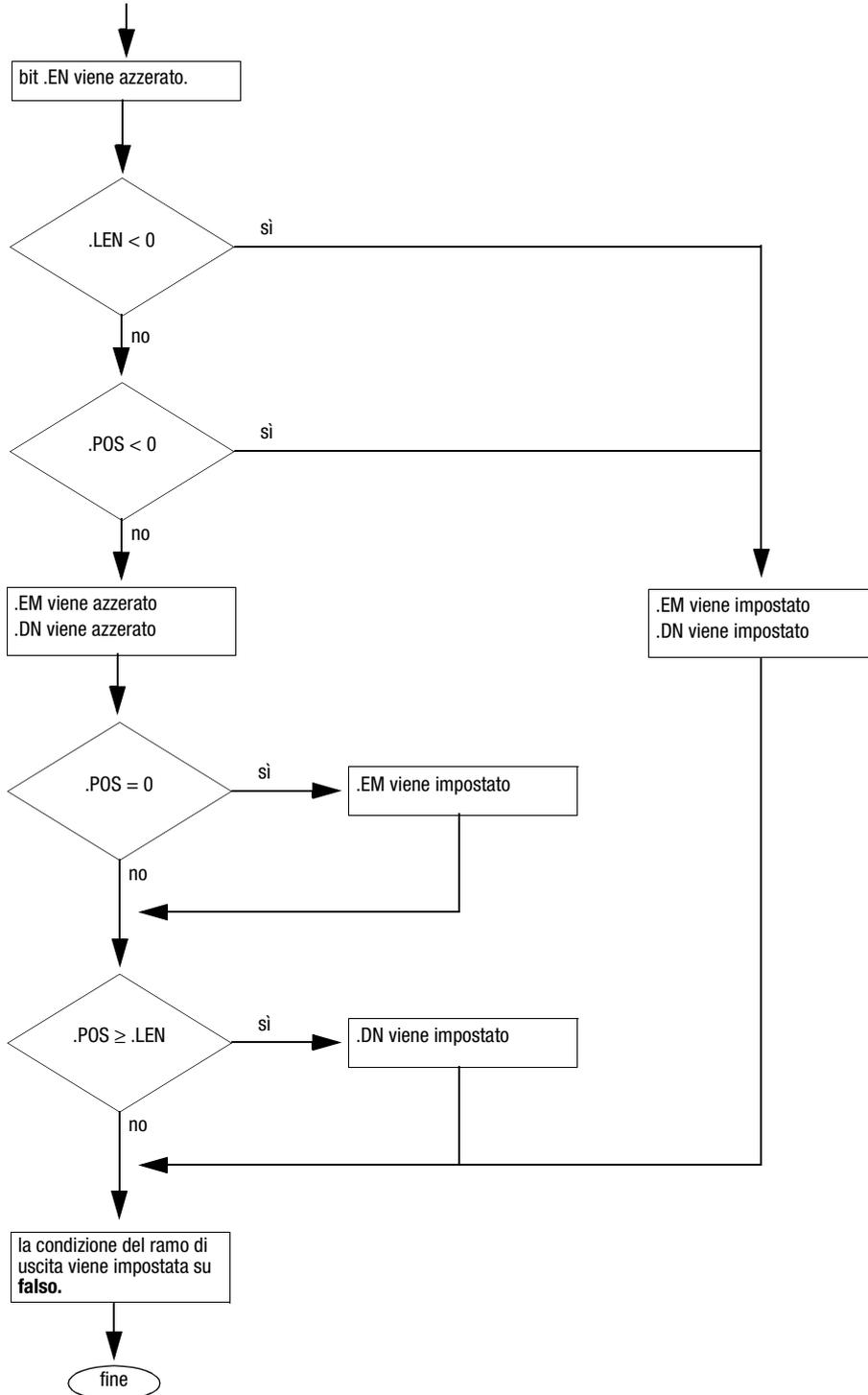
**Condizione:**

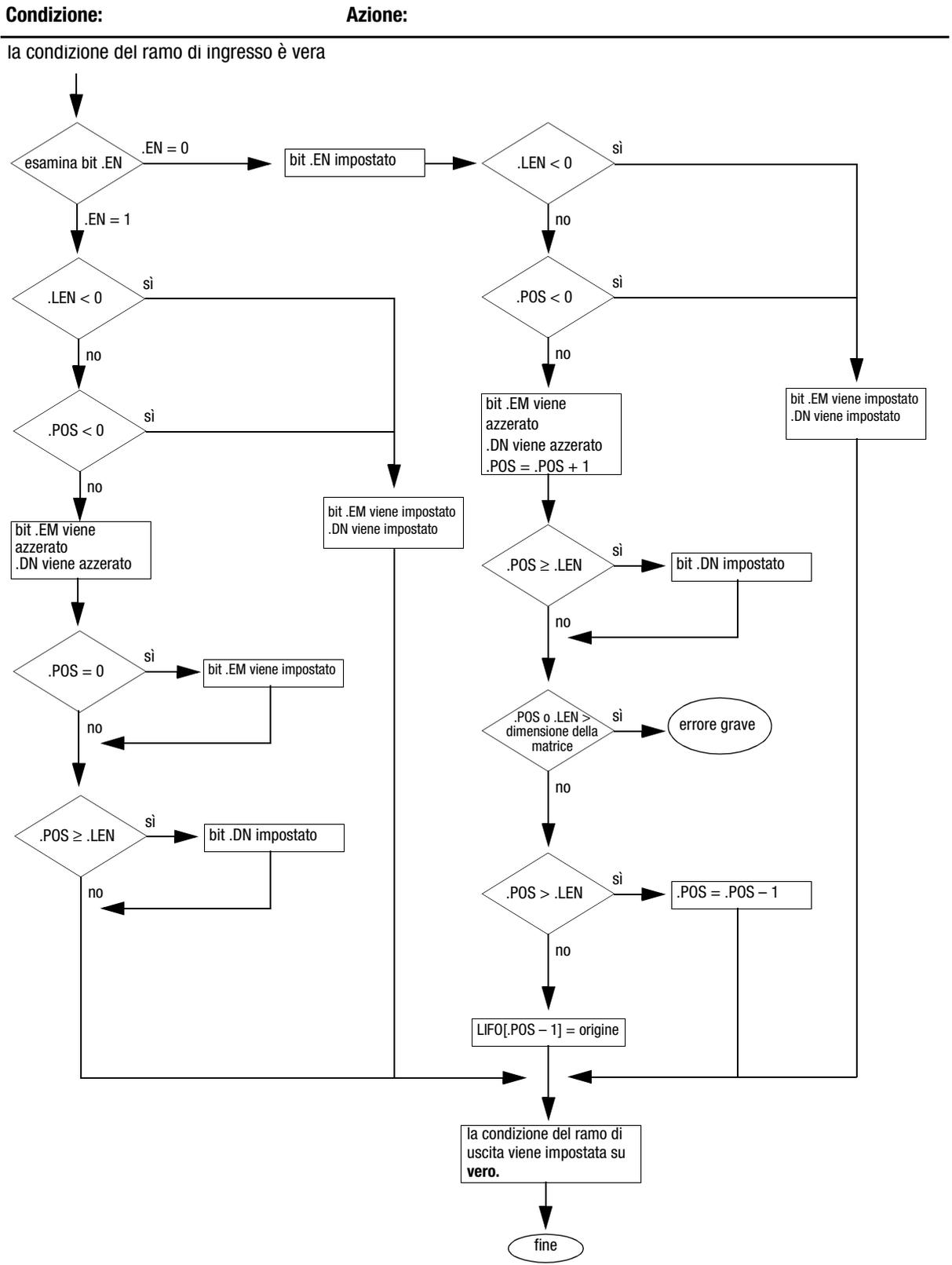
**Azione:**



**Condizione:****Azione:**

la condizione del ramo di ingresso è falsa

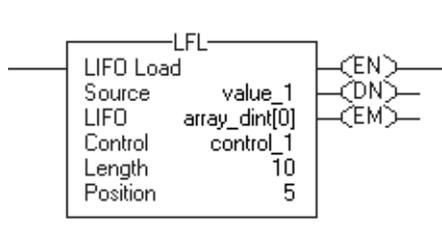




**Indicatori di stato aritmetico:** non influenzati

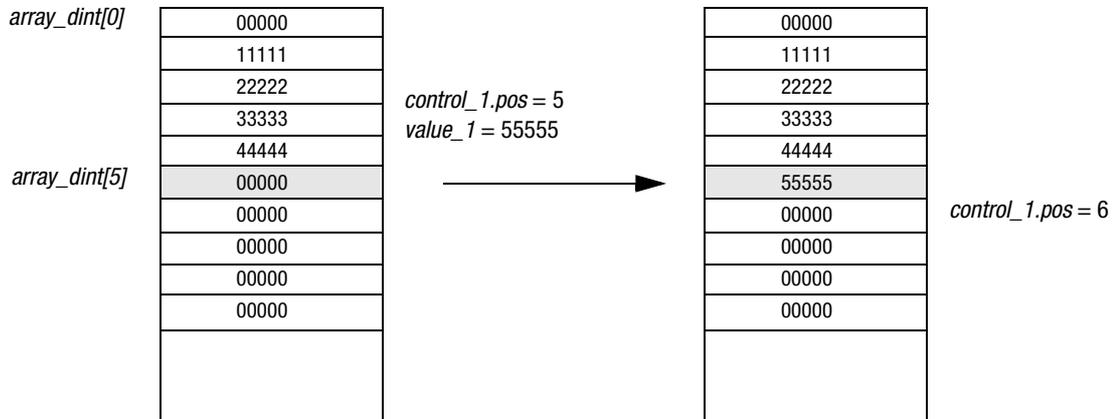
**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
(elemento iniziale + .POS) > dimensione matrice LIFO	4	20

**Esempio di LFL:**

prima del caricamento LIFO

dopo il caricamento LIFO



Se è abilitata, l'istruzione LFL carica *value\_1* nella posizione successiva del LIFO che, in questo esempio, è *array\_dint[5]*.

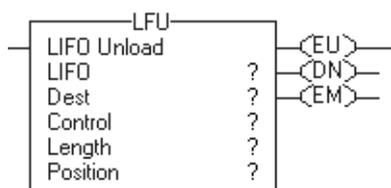
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>LFL(source, LIFO, control, length, position);</code>
testo ASCII	<code>LFL source LIFO control length position</code>

**Istruzioni correlate:** LFU, FFL, FFU

**Scarica LIFO (FFU)**

L'istruzione LFU è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
LIFO	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	tag matrice	LIFO da modificare specifica il primo elemento del LIFO <b>non</b> usare CONTROL.POS nell'indice
Destination	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b> <b>Struttura</b>	tag	valore che esce dal LIFO  Il valore Destination viene convertito nel tipo di dati della tag Destination. Un intero più piccolo viene convertito in un intero più grande mediante estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.
Control	CONTROL	tag	struttura di controllo per l'operazione in genere, utilizzare lo stesso CONTROL dell'LFL associato
Length	DINT	immediato	numero massimo di elementi che il LIFO può contenere contemporaneamente
Position	DINT	immediato	posizione successiva nel LIFO in cui l'istruzione scarica i dati il valore iniziale generalmente è 0

Se come tipo di dati per l'operando LIFO o Destination si utilizza una struttura definita dall'utente, utilizzare la stessa struttura per entrambi gli operandi.

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.EU	BOOL	Il bit di abilitazione scaricamento indica che l'istruzione LFU è abilitata. Il bit EU viene impostato per prevenire un falso scaricamento quando comincia la scansione del programma.
.DN	BOOL	Il bit di fine è impostato per indicare che il LIFO è pieno (.POS = .LEN).
.EM	BOOL	Il bit vuoto indica che il LIFO è vuoto. Se .LEN ≤ 0 oppure .POS < 0, vengono impostati sia il bit .EM che il bit .DN.
.LEN	DINT	La lunghezza specifica il numero massimo di elementi che il LIFO può contenere contemporaneamente.
.POS	DINT	La posizione identifica il punto fino al quale sono stati caricati dei dati nel LIFO.

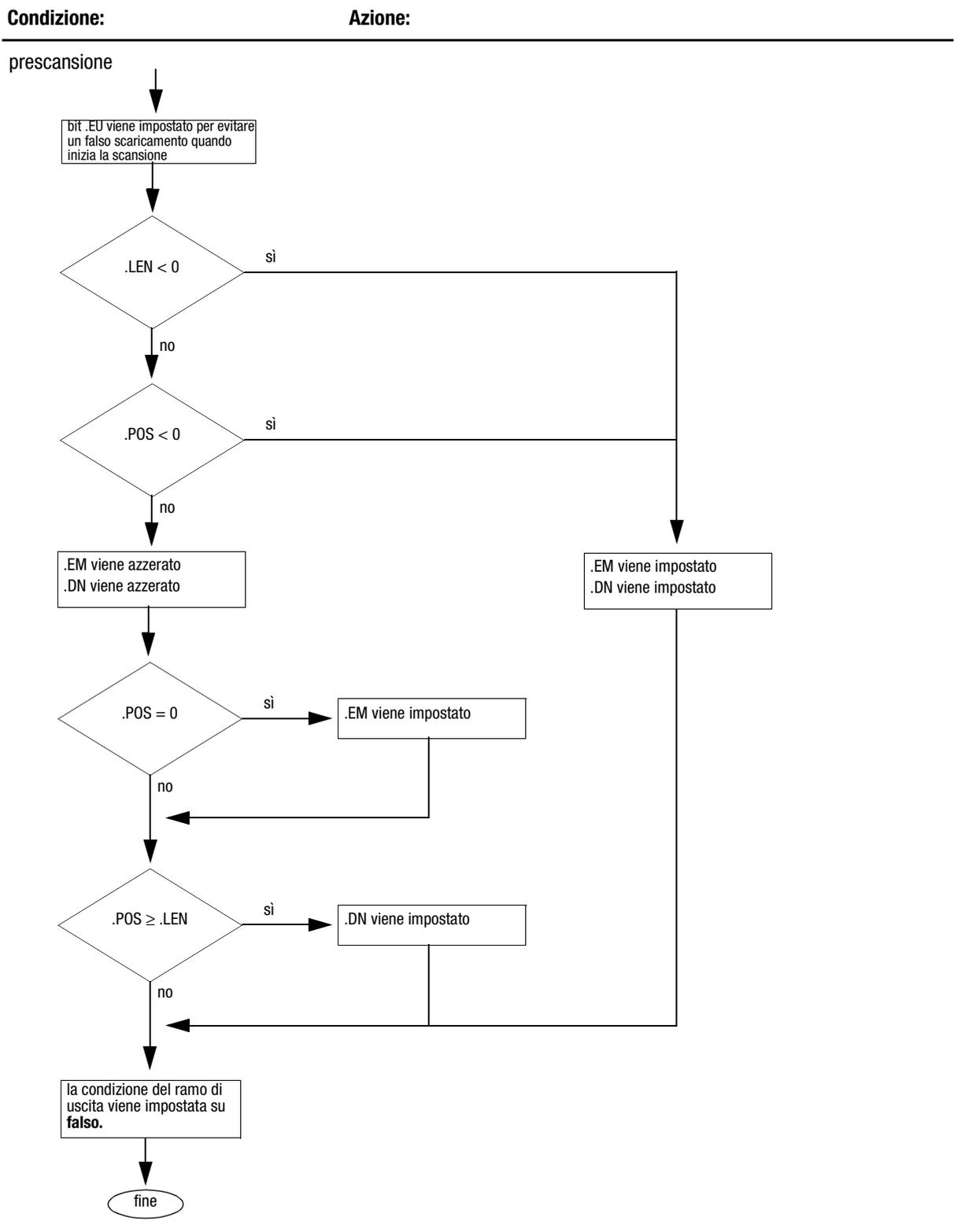
**Descrizione:** L'istruzione LFU scarica il valore di .POS del LIFO e memorizza uno 0 in quella posizione. Utilizzare l'istruzione LFU con l'istruzione LFL per memorizzare ed estrarre dati nell'ordine ultimo ad entrare/primo ad uscire.

Se il LIFO è un tipo di dati DINT, la destinazione deve essere un tipo di dati DINT; se il LIFO è un tipo di dati REAL, la destinazione deve essere un tipo di dati REAL.

Se è abilitata, l'istruzione LFU scarica il valore di .POS del LIFO e lo inserisce nella destinazione. L'istruzione scarica un valore e lo sostituisce con uno 0 ogni volta che l'istruzione è abilitata, fino a quando il LIFO non è vuoto. Se il LIFO è vuoto, l'LFU invia uno 0 alla destinazione.

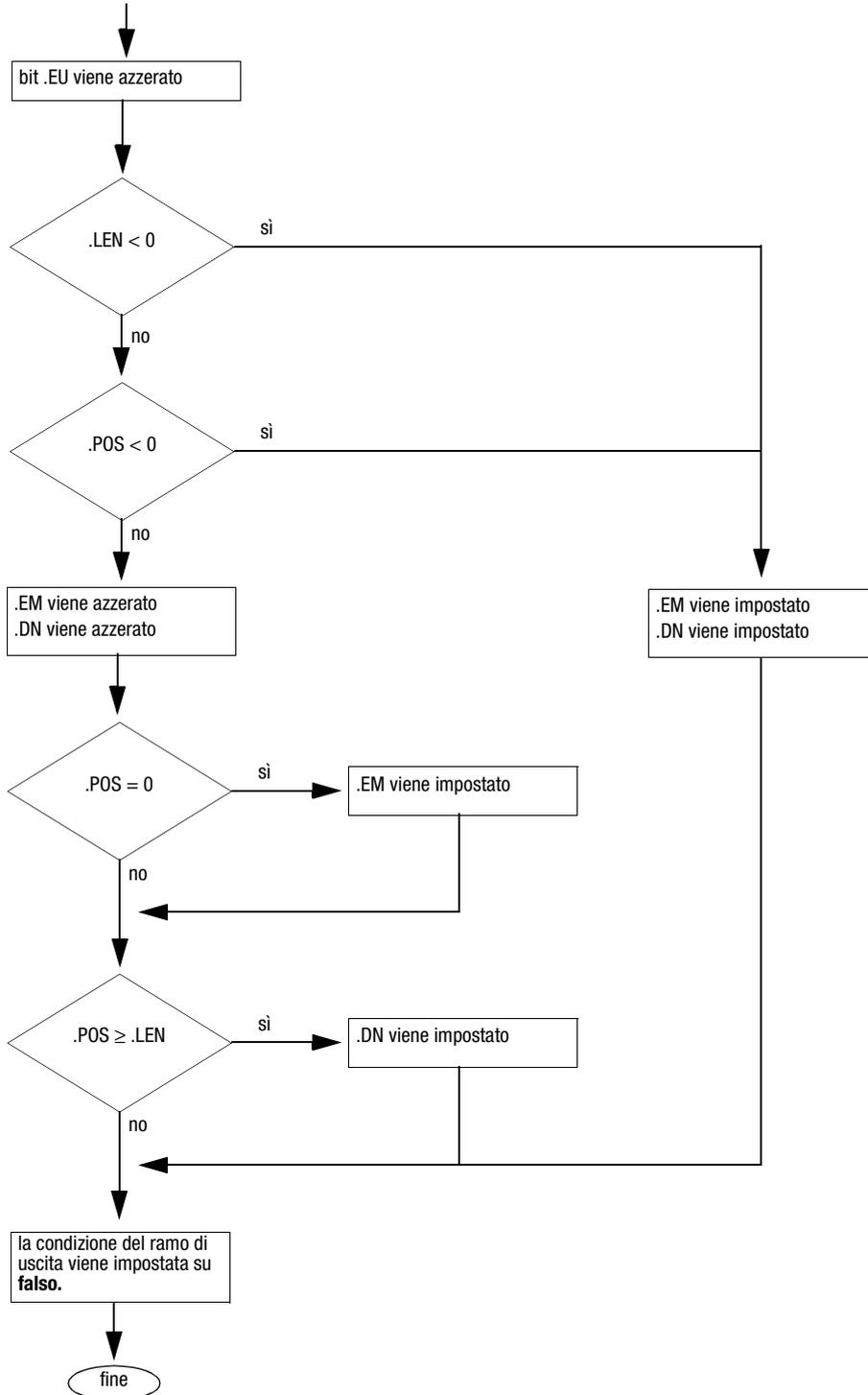
L'istruzione LFU opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

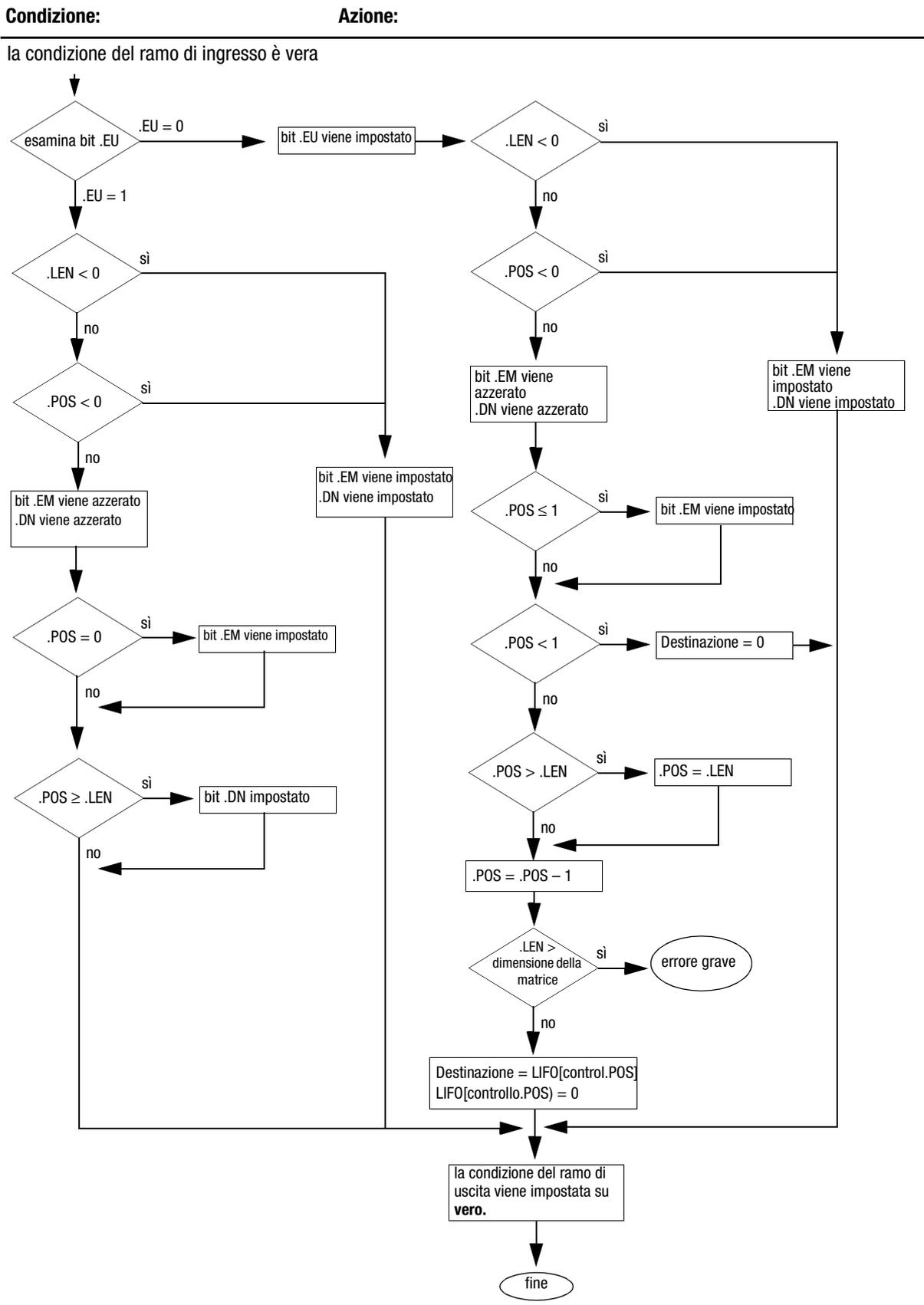
**Esecuzione:**



**Condizione:****Azione:**

la condizione del ramo di ingresso è falsa

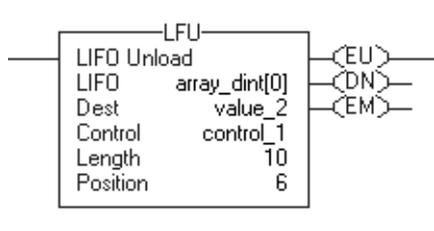




**Indicatori di stato aritmetico:** non influenzati

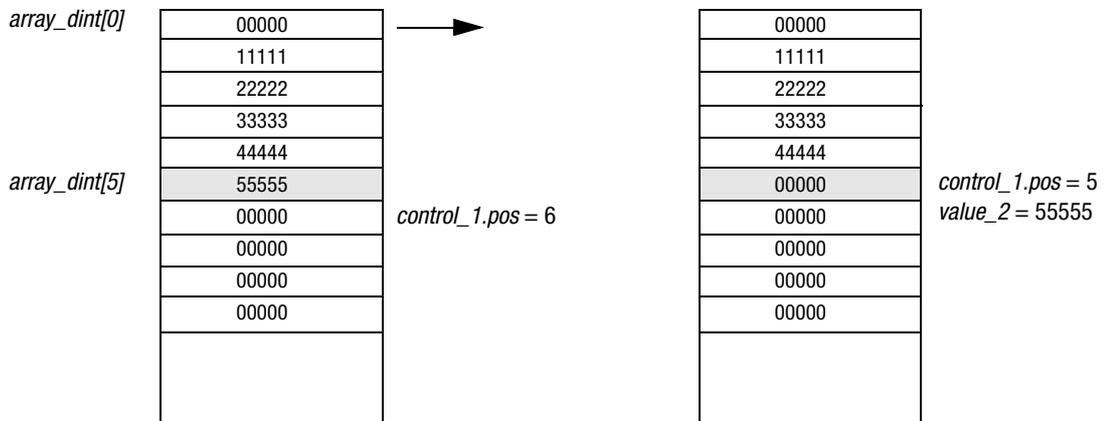
**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
Lunghezza > dimensione matrice LIFO	4	20

**Esempio di LFU:**

prima dello scaricamento LIFO

dopo lo scaricamento LIFO

Se è abilitata, l'istruzione LFU scarica *array\_dint[5]* in *value\_2*.**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>LFU(LIFO,destination,control,length,position);</code>
testo ASCII	<code>LFU LIFO destination control length position</code>

**Istruzioni correlate:** LFU, FFL, FFU

**Nota:**

## Istruzioni sequenziatore (SQI, SQO, SQL)

### Introduzione

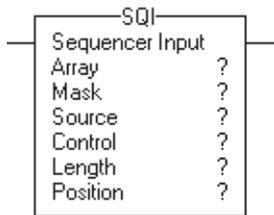
Le istruzioni sequenziatore controllano operazioni costanti e ripetibili.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
Sapere quando una fase è stata completata.	SQI	9-2
Impostare le condizioni di uscita per la fase successiva.	SQO	9-6
Caricare le condizioni di riferimento nelle matrici sequenziatore	SQL	9-11

I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente e richiede meno memoria se tutti gli operandi dell'istruzione utilizzano il medesimo tipo di dati ottimali, in genere DINT o REAL.

**Sequenziatore di ingresso (SQI)** L'istruzione SQI è un'istruzione di ingresso.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
Array	DINT	tag matrice	matrice sequenziatore specifica il primo elemento della matrice sequenziatore <b>non</b> usare CONTROL.POS nell'indice
Mask	SINT INT <b>DINT</b>	tag immediato	quali bit bloccare o passare  Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.
Source	SINT INT DINT	tag	dati di ingresso della matrice sequenziatore  Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.
Control	CONTROL	tag	struttura di controllo per l'operazione generalmente usa lo stesso CONTROL delle istruzioni SQO ed SQL
Length	DINT	immediato	numero di elementi della matrice (tabella sequenziatore) da confrontare
Position	DINT	immediato	posizione corrente nella matrice il valore iniziale generalmente è 0

**Struttura CONTROL:**

Mnemonico:	Tipo di dati:	Descrizione:
.ER	BOOL	Il bit di errore viene impostato quando $.LEN \leq 0$ , $.POS < 0$ o $.POS > .LEN$ .
.LEN	DINT	La lunghezza specifica il numero di passi della matrice sequenziatore.
.POS	DINT	La posizione identifica l'elemento che l'istruzione sta confrontando.

**Descrizione:** L'istruzione SQI rileva quando un passo è completo in una coppia in sequenza di istruzioni SQO/SQI. Se è abilitata, l'istruzione SQI verifica l'uguaglianza di un elemento Source, tramite una maschera, con un elemento della matrice.

In genere si usa la stessa struttura CONTROL delle istruzioni SQO ed SQL.

L'istruzione SQI opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

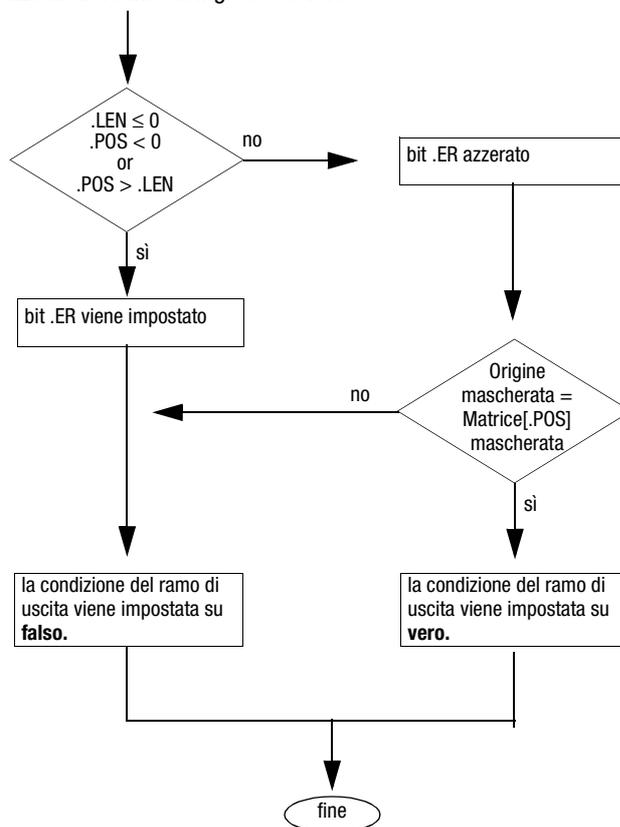
## Immissione di un valore di maschera immediato

Quando si immette una maschera, il software di programmazione imposta automaticamente valori decimali. Se si desidera immettere una maschera utilizzando un altro formato, fare precedere il prefisso corretto al valore.

Prefisso:	Descrizione:
16#	Esadecimale per esempio; 16#0F0F
8#	Ottale per esempio; 8#16
2#	Binario per esempio; 2#00110011

### Esecuzione:

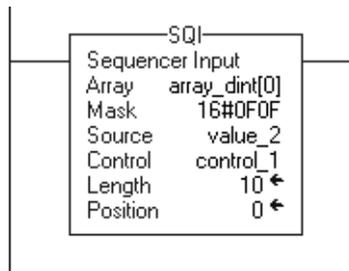
Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di SQI:**



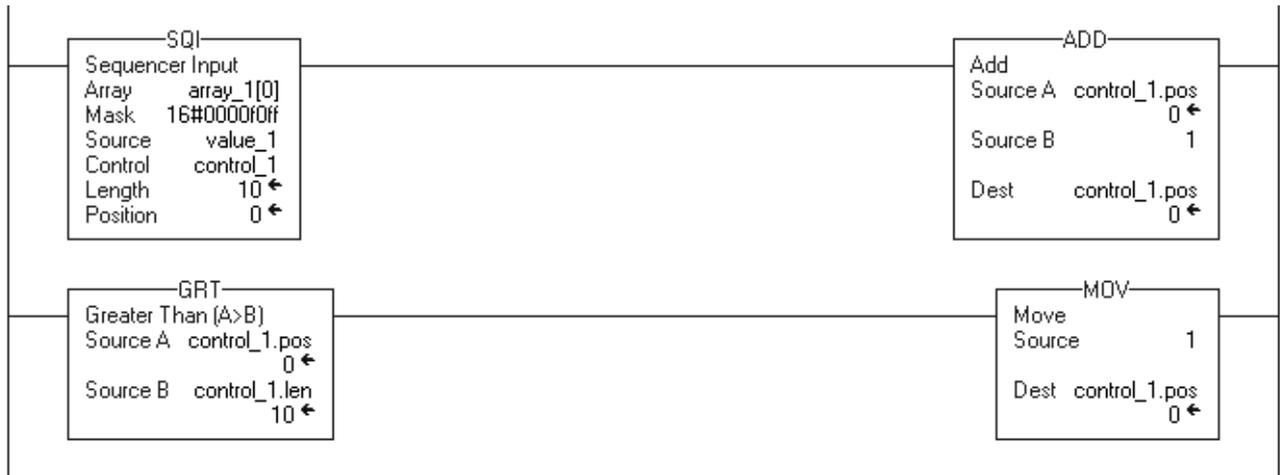
Se è abilitata, l'istruzione SQI fa passare *value\_2* attraverso la maschera per stabilire se il risultato è uguale all'elemento corrente della *array\_dint*. Il confronto mascherato è vero per cui la condizione del ramo di uscita diventa vera.

Operando SQI:	Valori di esempio (DINT visualizzati in binario):
Source	xxxxxxxx xxxxxxxx xxxx0101 xxxx1010
Mask	00000000 00000000 00001111 00001111
Array	xxxxxxxx xxxxxxxx xxxx0101 xxxx1010

Uno 0 nella maschera significa che il bit non viene confrontato (indicato dalle xxxx in questo esempio).

## Uso di SQI senza SQO

Se si utilizza l'istruzione SQI senza una istruzione SQO, è necessario incrementare dall'esterno la matrice sequenziatore.



L'istruzione SQI confronta il valore di origine.

L'istruzione ADD fa incrementare la matrice sequenziatore

L'istruzione GRT determina se nella matrice sequenziatore c'è un altro valore da controllare.

L'istruzione MOV ripristina il valore della posizione dopo avere completato una volta tutte le fasi della matrice sequenziatore.

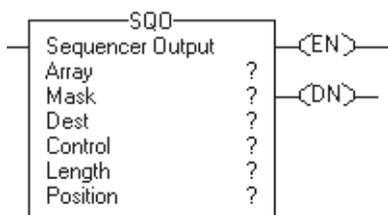
### Altri formati:

Formato:	Sintassi:
testo neutro	<code>SQI(array,mask,source,control,length,position);</code>
testo ASCII	<code>SQI array mask source control length position</code>

**Istruzioni correlate:** SQO, SQL

**Sequenziatore di uscita (SQO)**

L'istruzione SQO è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Array	DINT	tag matrice	matrice sequenziatore specifica il primo elemento della matrice sequenziatore <b>non</b> usare CONTROL.POS nell'indice
Mask	SINT INT <b>DINT</b>	tag immediato	quali bit bloccare o passare
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.			
Destination	DINT	tag	dati in uscita dalla matrice sequenziatore
Control	CONTROL	tag	struttura di controllo per l'operazione utilizzare lo stesso CONTROL delle istruzioni SQI ed SQL
Length	DINT	immediato	numero di elementi della matrice (tabella sequenziatore) per l'uscita
Position	DINT	immediato	posizione corrente nella matrice il valore iniziale generalmente è 0

**Struttura CONTROL:****Mnemonico: Tipo di dati: Descrizione:**

.EN	BOOL	Il bit di abilitazione indica che l'istruzione SQO è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando tutti gli elementi specificati sono stati spostati nella destinazione.
.ER	BOOL	Il bit di errore viene impostato quando $.LEN \leq 0$ , $.POS < 0$ o $.POS > .LEN$ .
.LEN	DINT	La lunghezza specifica il numero di passi della matrice sequenziatore.
.POS	DINT	La posizione identifica l'elemento che il controllore sta manipolando.

**Descrizione:** L'istruzione SQO imposta le condizioni di uscita per il passo successivo di una coppia in sequenza di istruzioni SQO/SQI. Se è abilitata, l'istruzione SQO incrementa la posizione, sposta i dati della posizione filtrati dalla maschera e memorizza il risultato nella destinazione. Se  $.POS > .LEN$ , l'istruzione ritorna all'inizio della matrice sequenziatore e continua con  $.POS = 1$ .

In genere utilizzare la stessa struttura CONTROL delle istruzioni SQI ed SQL.

L'istruzione SQO opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

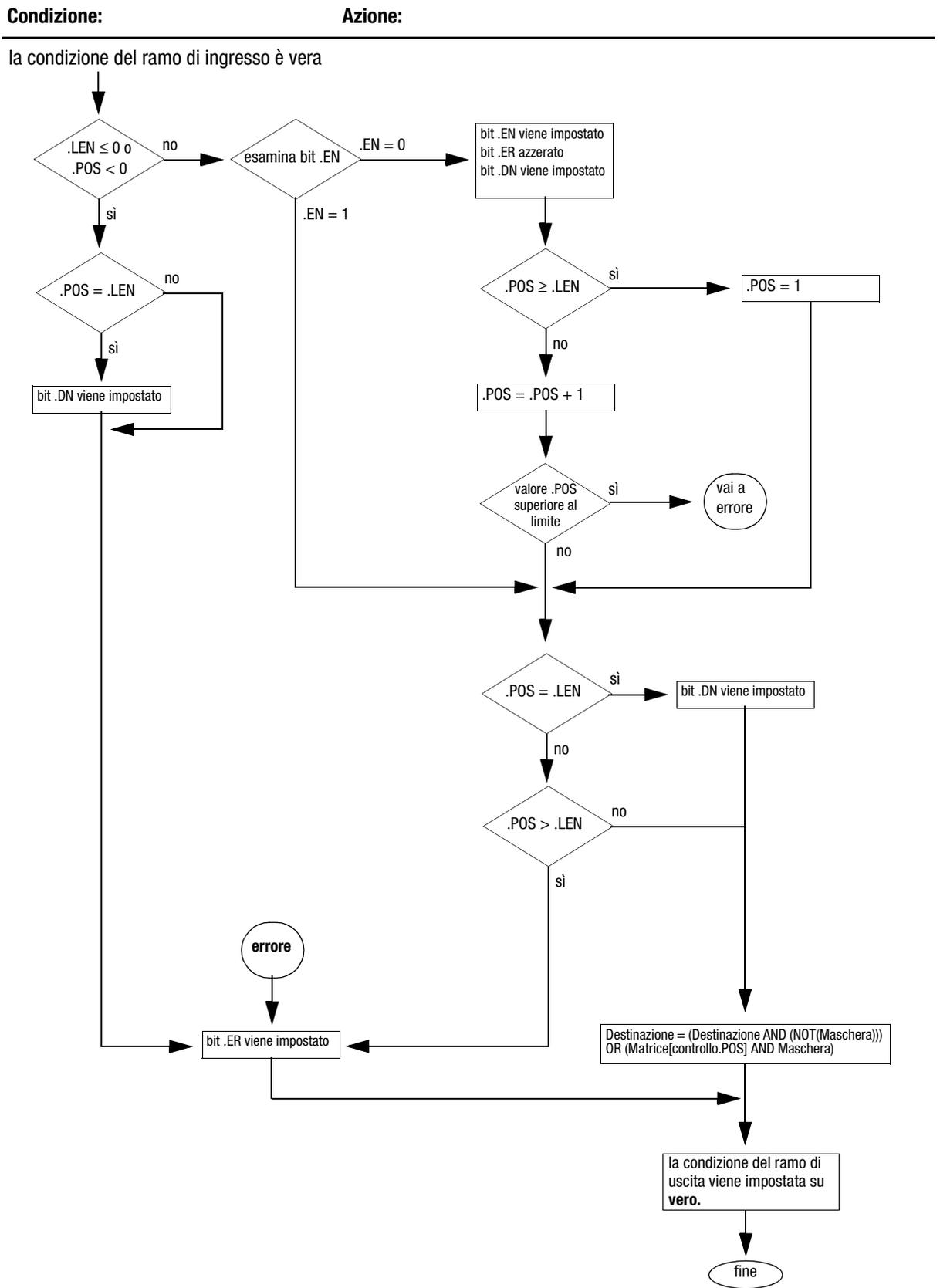
### Immissione di un valore di maschera immediato

Quando si immette una maschera, il software di programmazione imposta automaticamente valori decimali. Se si desidera immettere una maschera utilizzando un altro formato, fare precedere il prefisso corretto al valore.

Prefisso:	Descrizione:
16#	Esadecimale per esempio; 16#0F0F
8#	Ottale per esempio; 8#16
2#	Binario per esempio; 2#00110011

### Esecuzione:

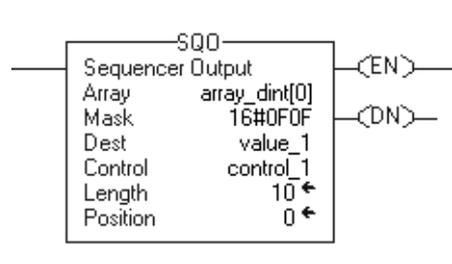
Condizione:	Azione:
prescansione	Il bit EN viene impostato per prevenire un falso caricamento quando comincia la scansione del programma. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit .EN viene azzerato. La condizione del ramo di uscita viene impostata su falso.



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

### Esempio di SQO:



Se è abilitata, l'istruzione SQO incrementa la posizione, passa i dati di quella posizione in *array\_dint* attraverso la maschera e memorizza il risultato in *value\_1*.

#### Operando SQO: Valori di esempio (con INT visualizzati in binario):

Array	xxxxxxxx xxxxxxxx xxxx0101 xxxx1010
Mask	00000000 00000000 00001111 00001111
Destination	xxxxxxxx xxxxxxxx xxxx0101 xxxx1010

Uno 0 nella maschera lascia il bit immutato (indicato dalle xxxx in questo esempio).

### Uso di SQI con SQO

Se si utilizza una coppia di istruzioni SQI ed SQO, assicurarsi che entrambe le istruzioni utilizzino gli stessi valori Control, Length e Position.



### Ripristino della posizione dell'istruzione SQO

Ogni volta che il controllore passa dalla modalità Program alla modalità Run, l'istruzione SQO azzerà (inizializza) il valore .POS. Per riportare .POS al valore di inizializzazione (.POS = 0), utilizzare un'istruzione RES per azzerare il valore della posizione. In questo esempio si utilizza lo stato del bit di prima scansione per azzerare il valore .POS.



#### Altri formati:

##### Formato:

##### Sintassi:

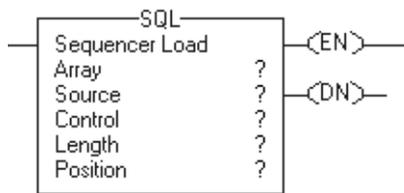
Formato:	Sintassi:
testo neutro	<i>SQO(array,mask,destination,control,length,position);</i>
testo ASCII	<i>SQO array mask destination control length position</i>

**Istruzioni correlate:** SQI, SQL

## Carica sequenziatore (SQL)

L'istruzione SQL è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Array	DINT	tag matrice	matrice sequenziatore specifica il primo elemento della matrice sequenziatore <b>non</b> usare CONTROL.POS nell'indice
Source	SINT INT <b>DINT</b>	tag immediato	dati di ingresso da caricare nella matrice sequenziatore
Una tag SINT o INT viene convertita in un valore DINT mediante l'estensione del segno. Fare riferimento a "Conversione di dati" a pag. A-6.			
Control	CONTROL	tag	struttura di controllo per l'operazione utilizzare lo stesso CONTROL delle istruzioni SQI ed SQO
Length	DINT	immediato	numero di elementi della matrice (tabella sequenziatore) da caricare
Position	DINT	immediato	posizione corrente nella matrice il valore iniziale generalmente è 0

### Struttura CONTROL:

#### Mnemonico: Tipo di dati: Descrizione:

.EN	BOOL	Il bit di abilitazione indica che l'istruzione SQL è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando tutti gli elementi specificati sono stati caricati nella matrice.
.ER	BOOL	Il bit di errore viene impostato quando $.LEN \leq 0$ , $.POS < 0$ o $.POS > .LEN$ .
.LEN	DINT	La lunghezza specifica il numero di passi della matrice sequenziatore.
.POS	DINT	La posizione identifica l'elemento che il controllore sta manipolando.

**Descrizione:** L'istruzione SQL carica le condizioni di riferimento in una matrice sequenziatore. Se è abilitata, l'istruzione SQL avanza fino alla posizione successiva della matrice sequenziatore e carica il valore Source in quella posizione. Se il bit .DN è impostato oppure se  $.POS \geq .LEN$ , l'istruzione imposta  $.POS=1$ .

In genere è necessario utilizzare la stessa struttura CONTROL delle istruzioni SQI ed SQO.

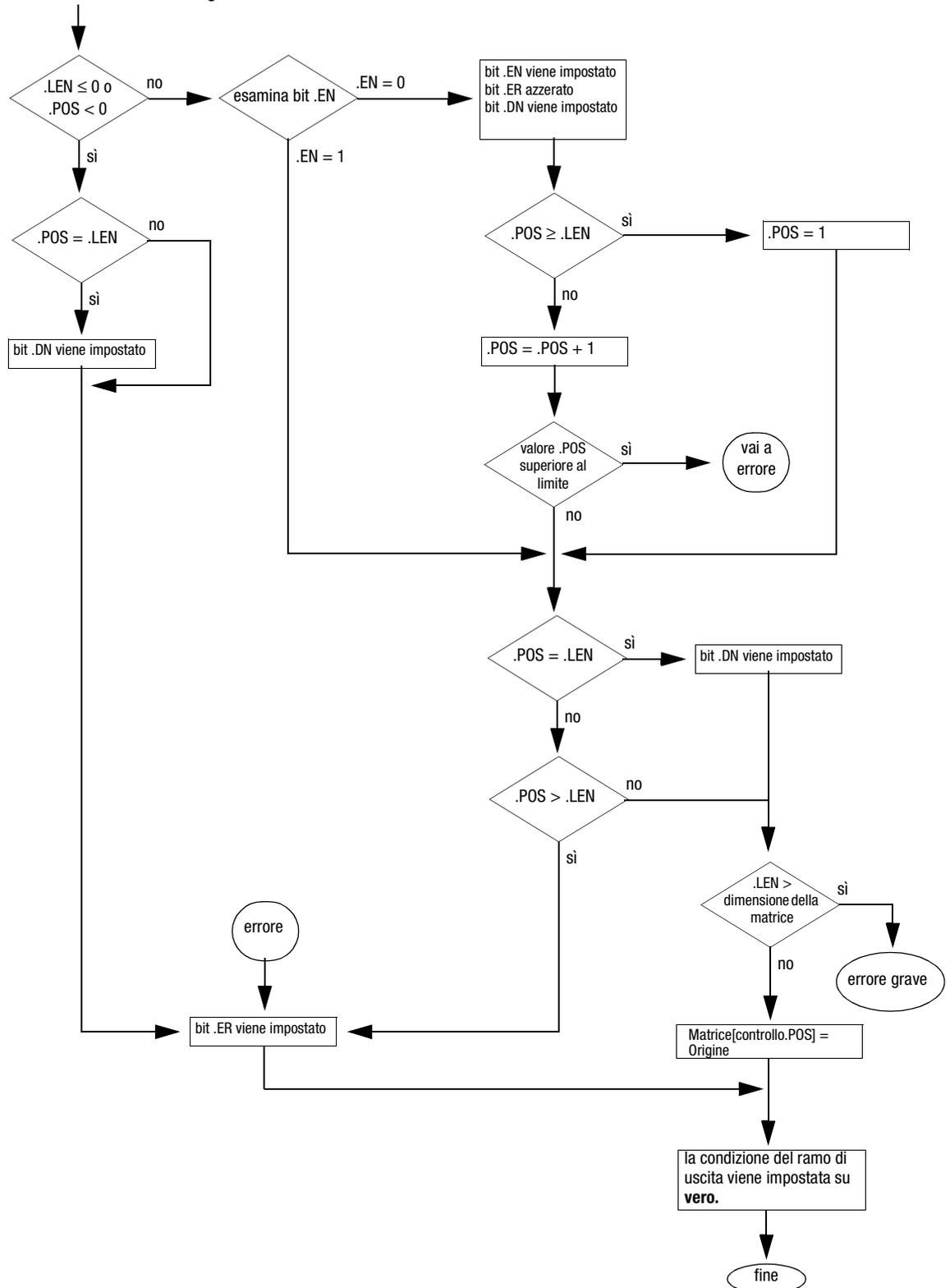
L'istruzione SQL opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

**Esecuzione:**

<b>Condizione:</b>	<b>Azione:</b>
prescansione	Il bit EN viene impostato per prevenire un falso caricamento quando comincia la scansione del programma. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Il bit .EN viene azzerato. La condizione del ramo di uscita viene impostata su falso.

**Condizione:****Azione:**

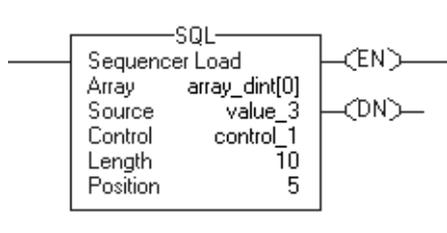
la condizione del ramo di ingresso è vera

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

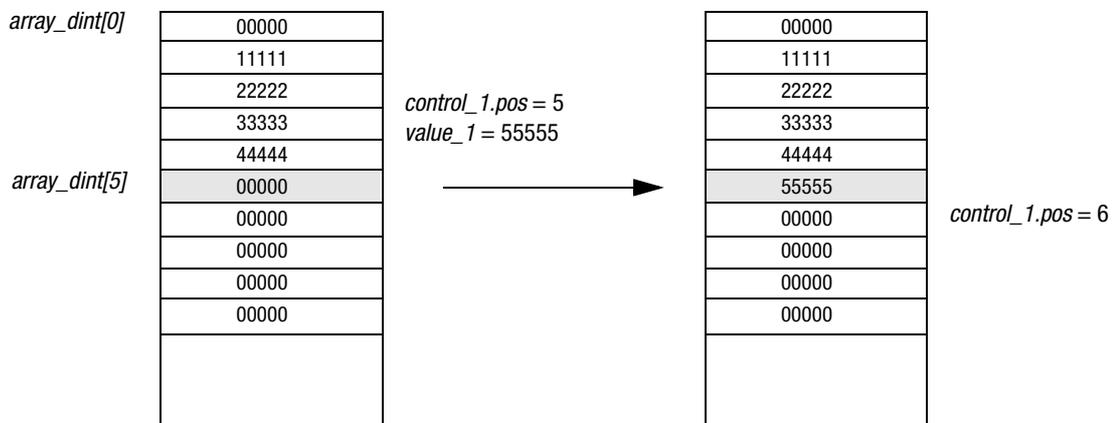
Si verifica un errore grave se:	Tipo errore:	Codice errore:
Lunghezza > dimensione della matrice	4	20

**Esempio di SQL:**



prima del caricamento

dopo il caricamento



Se è abilitata, l'istruzione SQL carica il *value\_3* nella posizione successiva della matrice sequenziatore, che è la *array\_dint[5]* in questo esempio.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>SQL(array,source,control,length,position);</code>
testo ASCII	<code>SQL array source control length position</code>

**Istruzioni correlate:** SQI, SQO

---

## **Istruzioni di controllo programma (JMP, LBL, JSR, RET, SBR, TND, MCR, UID, UIE, AFI, NOP)**

### **Introduzione**

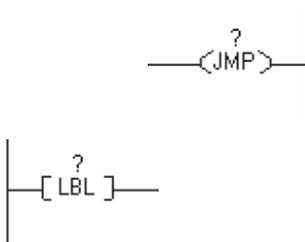
Utilizzare le istruzioni di controllo programma per modificare l'esecuzione della logica.

<b>Se si desidera:</b>	<b>Utilizzare questa istruzione:</b>	<b>Vedere pagina:</b>
Saltare un segmento di logica che non è necessario eseguire sempre.	JMP LBL	10-2
Saltare ad un'altra routine, trasferire dati alla routine, eseguire la routine ed inviare i risultati.	JSR SBR RET	10-4
Impostare una fine temporanea che interrompa l'esecuzione della routine.	TND	10-10
Disabilitare tutti i rami di una sezione di logica.	MCR	10-11
Disabilitare i task utente.	UID	10-13
Abilitare i task utente.	UIE	10-14
Disabilitare un ramo.	AFI	10-15
Inserire un marcatore di posizione nella logica.	NOP	10-16

**Salta all'etichetta (JMP)  
Etichetta (LBL)**

L'istruzione JMP è un'istruzione di uscita.  
L'istruzione LBL è un'istruzione di ingresso.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
<b>Istruzione JMP</b>			
Label name		nome etichetta	inserire il nome dell'istruzione LBL associata
<b>Istruzione LBL</b>			
Label name		nome etichetta	il programma salta all'istruzione LBL con il nome di riferimento

**Descrizione:** Le istruzioni JMP ed LBL consentono di saltare porzioni di logica ladder. Se è abilitata, l'istruzione JMP va all'istruzione LBL di riferimento ed il controllore continua l'esecuzione da quel punto. Se è disabilitata, l'istruzione JMP non ha alcun effetto sull'esecuzione della logica ladder.

L'istruzione JMP può spostare l'esecuzione ladder in avanti o indietro. Il salto ad una LBL fa diminuire il tempo di scansione del programma poiché si evita un segmento di logica. Il salto indietro fa ripetere al controllore iterazioni di logica.

Fare attenzione a non saltare indietro troppe volte. Il timer watchdog potrebbe scadere perché il controllore non raggiunge mai la fine della logica, e quindi causare l'errore del controllore.

**ATTENZIONE:** la logica saltata non viene scandita. Inserire la logica critica al di fuori dell'area saltata.



L'istruzione LBL rappresenta la destinazione dell'istruzione JMP con lo stesso nome di etichetta. **Assicurarsi che l'istruzione LBL sia la prima istruzione sul suo ramo.**

Esiste solo un'etichetta con quel nome in una routine. Il nome può:

- contenere un massimo di 40 caratteri
- contenere lettere, numeri e sottolineature ( \_ )

**Esecuzione:**

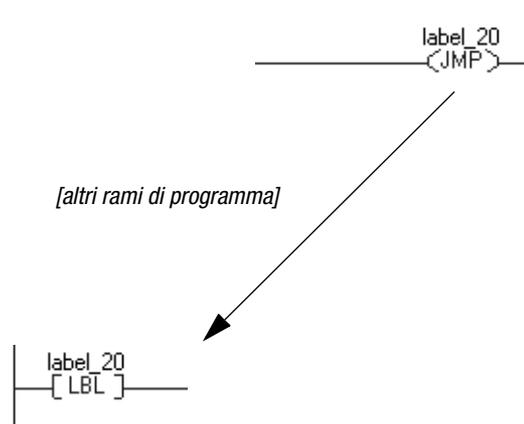
Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su vera. L'esecuzione salta al ramo che contiene l'istruzione LBL con il nome di etichetta di riferimento.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
l'etichetta non esiste	4	42

**Esempio di JMP/LBL:**



Se l'istruzione JMP è abilitata, l'esecuzione salta i rami di logica seguenti fino a raggiungere il ramo contenente l'istruzione LBL con *label\_20*.

**Altri formati:**

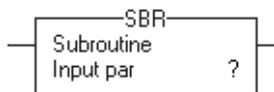
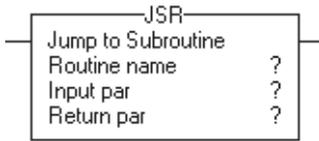
Formato:	Sintassi:
testo neutro	JMP ( <i>label_name</i> ); LBL ( <i>label_name</i> );
testo ASCII	JMP <i>label_name</i> LBL <i>label_name</i>

**Istruzioni correlate:** JSR, SBR, RET, FOR, BRK

**Salta alla subroutine (JSR)  
Subroutine (SBR)  
Ritorno (RET)**

L'istruzione JSR è un'istruzione di uscita.  
L'istruzione SBR è un'istruzione di ingresso.  
L'istruzione RET è un'istruzione di uscita.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
<b>Istruzione JSR</b>			
Routine name	ROUTINE	nome	routine da eseguire
Input par	SINT INT DINT REAL struttura	immediato tag tag matrice	parametri da inviare alla routine
Return par	SINT INT DINT REAL struttura	tag tag matrice	parametri (0-n) da ricevere dalla routine
<b>Istruzione SBR</b>			
Input par	SINT INT DINT REAL struttura	tag tag matrice	parametri (0-n) ricevuti dalla JSR
<b>Istruzione RET</b>			
Return par	SINT INT DINT REAL struttura	immediato tag tag matrice	parametri (0-n) da inviare alla JSR

**ATTENZIONE:** il parametro di ingresso ed il corrispondente parametro di ritorno devono essere dello stesso tipo di dati, altrimenti possono ottenersi dati imprevisti o un funzionamento pericoloso.



**Descrizione:** Le istruzioni JSR, SBR e RET dirigono l'esecuzione della logica ad una subroutine separata all'interno del programma, scandiscono questa subroutine una sola volta e quindi ritornano al punto di partenza.

**TI** È inoltre possibile utilizzare l'istruzione JSR per testare una routine di errore o per spegnere il controllore. Per ulteriori informazioni, fare riferimento al *Logix5000 Controllers User Manual*, publication 1756-6.5.12.

Se abilitata, l'istruzione JSR esegue la subroutine specificata e, se necessario, fornisce i parametri della subroutine. Se non vi sono parametri di ingresso, il controllo passa dall'istruzione JSR al primo ramo della subroutine.

Se abilitata, l'istruzione JSR fornisce i parametri di ingresso, se presenti, ed esegue il primo ramo della subroutine. L'istruzione SBR riceve i parametri di ingresso e copia i relativi valori nelle tag specificate. Il numero ed il tipo dei parametri di ingresso dell'istruzione JSR devono corrispondere con quelli dell'istruzione SBR. Se l'istruzione JSR ha un numero inferiore di parametri di ingresso rispetto all'istruzione SBR associata, il controllore segnala un errore grave. L'istruzione JSR può avere un numero maggiore di parametri di ingresso rispetto all'istruzione SBR associata senza che ciò provochi un errore.

Utilizzare l'istruzione SBR nella subroutine solo se si desidera trasferire i parametri alla subroutine. Se si utilizza un'istruzione SBR, essa deve essere la prima istruzione sul primo ramo della routine. L'istruzione SBR opzionale identifica le tag che memorizzano i parametri in ingresso. È possibile inserire un'istruzione SBR senza parametri di ingresso per aiutare ad identificare la routine come una subroutine.

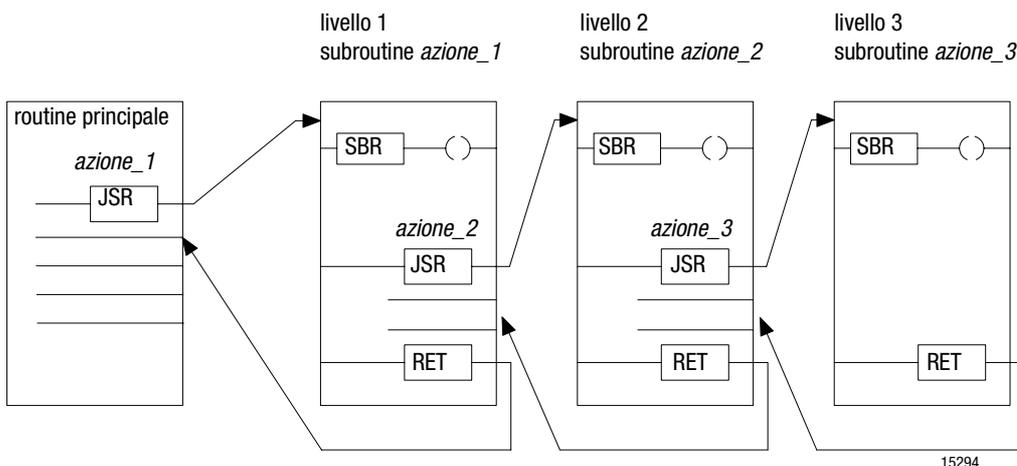
Non è possibile inserire un'istruzione SBR in una routine principale.

Usare l'istruzione RET solo se si ritornano i parametri all'istruzione JSR. L'istruzione RET conclude la subroutine e, se necessario, ritorna i parametri all'istruzione JSR. Una subroutine può avere più di un'istruzione RET.

Se abilitata, l'istruzione RET passa i propri parametri, se presenti, e riprende l'esecuzione dal ramo successivo dell'istruzione JSR associata. Il numero ed il tipo dei parametri di ritorno dell'istruzione RET devono corrispondere con quelli dell'istruzione JSR. Se il numero dei parametri di ritorno dell'istruzione RET è inferiore rispetto a quelli dell'istruzione JSR, il controllore segnala un errore grave. L'istruzione RET può avere un numero maggiore di parametri di ritorno rispetto all'istruzione SBR associata senza che ciò provochi un errore.

Se è disabilitata, l'istruzione RET non ha alcun effetto sull'esecuzione della logica ladder. Il controllore continua ad eseguire la subroutine corrente.

Il numero di routine annidate che è possibile avere o di parametri che è possibile passare o ritornare dipende solo dalla memoria del controllore.



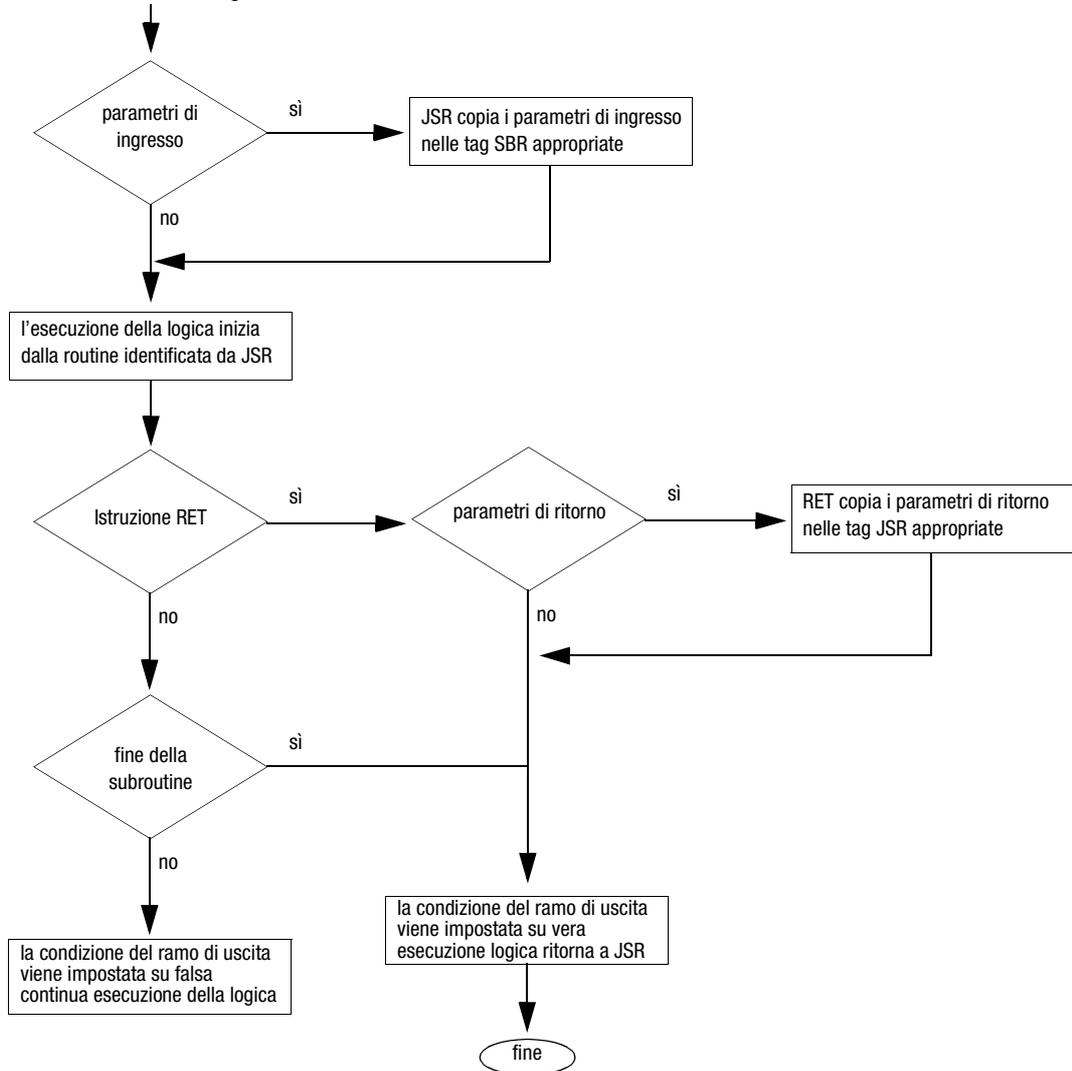
Le istruzioni JSR, SBR e RET passano i valori dei parametri verso/da una routine. Ciò significa che le istruzioni utilizzano un tempo d'esecuzione più lungo e maggiore memoria per copiare i valori. È possibile ridurre il tempo d'esecuzione se, invece di passare i valori, si accede ai dati dell'ambito del programma e dell'ambito del controllore direttamente dall'interno della routine.

È possibile passare singoli elementi di matrice, matrice intere, singoli elementi di struttura e strutture intere. Matrici e strutture vengono copiate nello stesso modo con cui un'istruzione COP copia i valori. Si consiglia di passare parametri di matrice o di struttura a parametri di matrice o di struttura rispettivamente dello stesso tipo.

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso. Il controllore esegue tutte le subroutine a prescindere dalla condizione del ramo, ma ignora le istruzioni RET. Alla subroutine vengono passati tutti i parametri di ingresso. Tutti i parametri di ritorno vengono anch'essi passati, ma l'istruzione RET non esce dalla subroutine. Questo garantisce che tutti i rami della subroutine vengano prescanditi. Le subroutine ricorsive, vengono prescandite solo la prima volta. Una subroutine richiamata più volte (non ricorsiva), viene prescandita ogni volta.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è vera



**Indicatori di stato aritmetico:** non influenzati

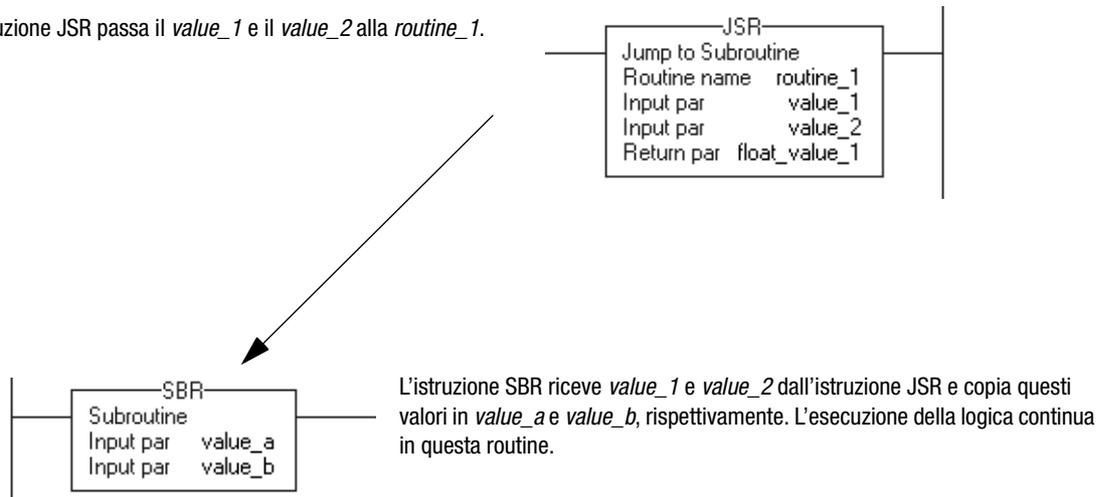
**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
l'istruzione JSR ha meno parametri di ingresso dell'istruzione SBR	4	31
l'istruzione JSR salta ad una routine di errore	4 o fornito dall'utente	0 o fornito dall'utente
l'istruzione RET ha meno parametri di ritorno dell'istruzione JSR	4	31

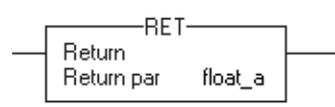
**Esempio di JSR/SBR/RET:**

**esempio 1**

Se è abilitata, l'istruzione JSR passa il *value\_1* e il *value\_2* alla *routine\_1*.

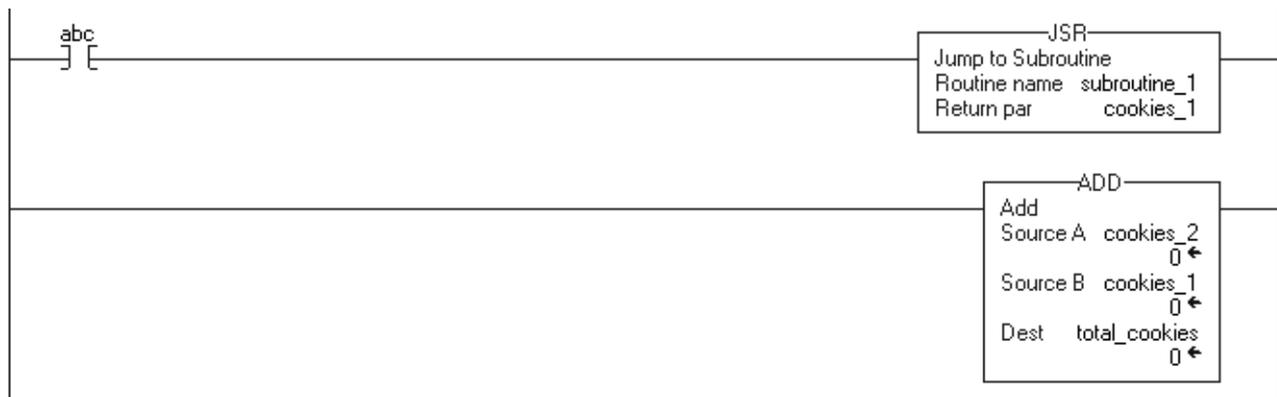


Se è abilitata, l'istruzione RET invia *float\_a* all'istruzione JSR. L'istruzione JSR riceve *float\_a* e copia il valore in *float\_value\_1*. L'esecuzione della logica continua con l'istruzione successiva all'istruzione JSR.



## esempio 2

routine principale

routine *subroutine\_1*

Se *abc* diventa vero, l'istruzione JSR viene abilitata e il controllo passa a *subroutine\_1*. Se *def* è abilitato, l'istruzione RET ritorna il *value\_1* al parametro *cookies\_1* di JSR ed il resto della subroutine non viene scandito. Se *ghi* è abilitato, l'istruzione RET ritorna il *value\_2* al parametro *cookies\_1* di JSR ed il resto della subroutine non viene scandito. Se né *def* né *ghi* sono abilitati, l'istruzione RET ritorna il *value\_3* al parametro *cookies\_1* di JSR. L'istruzione ADD, poi, somma il valore di *cookies\_1* con quello di *cookies\_2* e memorizza il risultato in *total\_cookies*.

**Altri formati:**

Formato:	Sintassi:
testo neutro	JSR( <i>routine_name</i> , <i>input_1</i> ,... <i>input_n</i> , <i>return_1</i> ,... <i>return_n</i> ); SBR( <i>routine_name</i> , <i>input_1</i> ,... <i>input_n</i> ); RET( <i>return_1</i> ,... <i>return_n</i> );
testo ASCII	JSR <i>routine_name</i> <i>input_1</i> ... <i>input_n</i> <i>return_1</i> .. <i>return_n</i> SBR <i>routine_name</i> <i>input_1</i> ... <i>input_n</i> RET <i>return_1</i> ... <i>return_n</i>

**Istruzioni correlate:** JMP, LBL, FOR, BRK

## Fine temporanea (TND)

L'istruzione TND è un'istruzione di uscita.

—(TND)—

**Operandi:** nessuno

**Descrizione:** L'istruzione TND agisce come una delimitazione.

Se è abilitata, l'istruzione TND permette al controllore di eseguire la logica solo fino a questa istruzione.

Se è abilitata, l'istruzione TND agisce come fine della routine. Quando il controllore esegue la scansione di un'istruzione TND, il controllore si sposta alla fine della routine corrente. Se l'istruzione TND si trova in una subroutine, il controllo ritorna alla routine chiamante. Se l'istruzione TND si trova invece in una routine principale, il controllo ritorna al programma successivo del task corrente.

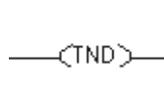
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su vero. La routine corrente termina.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuno

**Esempio di TND:** È possibile utilizzare l'istruzione TND per il debugging o la ricerca guasti eseguendo la logica solo fino ad un certo punto. Via via che viene eseguito il debugging di una nuova sezione, spostare l'istruzione TND nella restante logica.



Se l'istruzione TND è abilitata, il controllore interrompe la scansione della routine corrente.

### Altri formati:

Formato:	Sintassi:
testo neutro	TND ( ) ;
testo ASCII	TND

**Istruzioni correlate:** AFI, MCR, NOP

## Relè di controllo principale (MCR)

L'istruzione MCR è un'istruzione di uscita.

—(MCR)—

**Operandi:** nessuno

**Descrizione:** L'istruzione MCR, usata a coppie, crea una zona di programma in grado di disabilitare tutti i rami all'interno dell'istruzione MCR.

Se la zona MCR è abilitata, i rami della zona MCR vengono scanditi normalmente per verificare le condizioni di vero o falso. Se è disabilitata, il controllore scandisce comunque i rami della zona MCR, ma il tempo di scansione è ridotto poiché tutte le uscite non ritentive della zona sono disabilitate. La condizione del ramo di ingresso è falsa per tutte le istruzioni all'interno della zona MCR disabilitata.

Quando si programma una zona MCR, notare quanto segue:

- È necessario terminare la zona con un'istruzione MCR incondizionata.
- Non è possibile annidare una zona MCR all'interno di un'altra.
- Non saltare in una zona MCR. Se la zona è falsa, il saltare in questa la rende attiva a partire dal punto in cui si è saltati fino alla fine della zona.
- Se una zona MCR continua fino alla fine della routine, non è necessario programmare un'istruzione MCR fino alla fine della zona.

**Importante:** l'istruzione MCR non sostituisce il relè di controllo principale cablato per l'arresto di emergenza. Per garantire l'arresto di emergenza degli I/O, bisogna in ogni caso installare un relè di controllo principale cablato.

---

**ATTENZIONE:** non sovrapporre o annidare zone MCR. Ogni zona MCR deve essere separata e completa. In caso contrario potrebbe verificarsi un funzionamento anomalo della macchina con possibili danni alle apparecchiature o alle persone.



Inserire le operazioni critiche al di fuori della zona MCR. Se si avvia un'istruzione, ad esempio un timer, in una zona MCR, l'esecuzione dell'istruzione si interrompe quando la zona è disabilitata ed il timer viene azzerato.

---

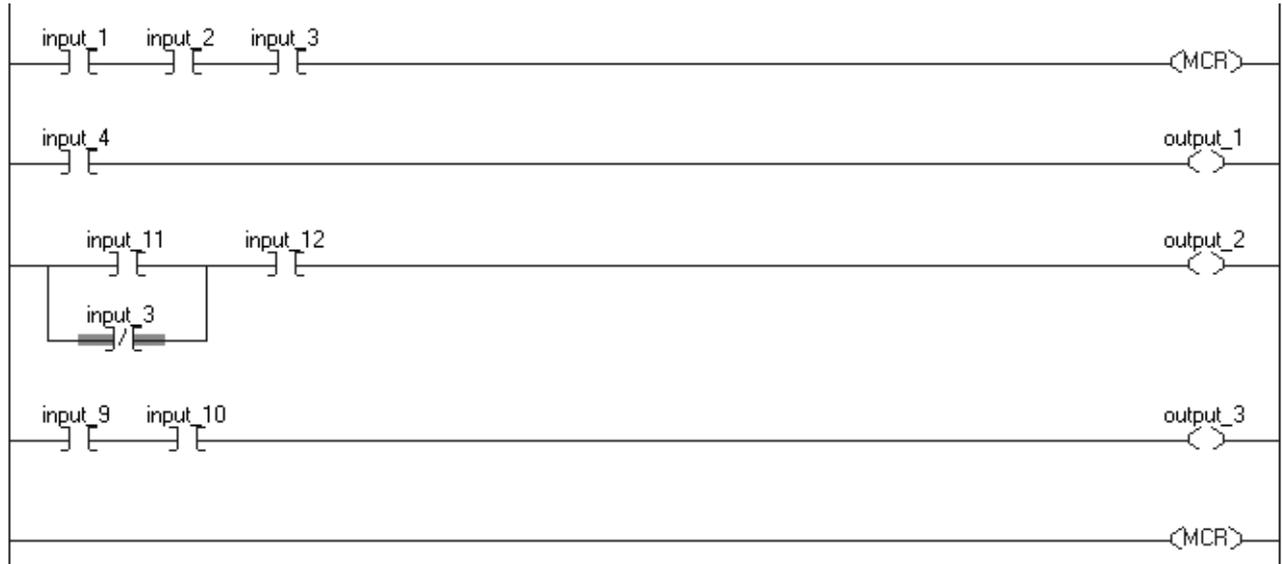
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falsa.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falsa. Le istruzioni della zona vengono scandite, ma la condizione del ramo di ingresso è falsa e le uscite non ritentive sono disabilitate.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su vero. Le istruzioni della zona vengono scandite normalmente.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuno

**Esempio di MCR:**



Quando la prima istruzione MCR è abilitata (*input\_1*, *input\_2* e *input\_3* impostati), il controllore esegue i rami nella zona MCR (tra le due istruzioni MCR) e, a seconda delle condizioni di ingresso, imposta o azzerava le uscite.

Quando la prima istruzione MCR è disabilitata (*input\_1*, *input\_2* e *input\_3* non sono tutti impostati), il controllore esegue i rami nella zona MCR (tra le due istruzioni MCR) e la condizione del ramo di ingresso diventa falsa per tutti i rami della zona MCR, a prescindere dalle condizioni di ingresso.

**Altri formati:**

Formato:	Sintassi:
testo neutro	MCR ( ) ;
testo ASCII	MCR

**Istruzioni correlate:** AFI, NOP, TND

**Disabilita interrupt utente (UID)** L'istruzione UID è un'istruzione di uscita.

—<UID>—

**Operandi:** nessuno

**Descrizione:** L'istruzione UID disabilita temporaneamente la commutazione tra i task utente.

Quando l'istruzione UID è abilitata, il task corrente continua ad essere eseguito senza interruzione da parte dei task con priorità più elevata, fino a quando non viene eseguita un'istruzione UIE oppure viene raggiunta la fine di un programma. L'istruzione UID non disabilita l'esecuzione di una routine di errore o un task di errore.

Se è abilitata, l'istruzione UID fa incrementare un contatore interno. Finché il contatore non raggiunge lo zero, il task correntemente in esecuzione non può essere interrotto. È possibile annidare le istruzioni UID fino a 65.535 livelli.

**Esecuzione:**

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Viene impedita l'interruzione da parte dei task con maggiore priorità. Il contatore interno dell'UID incrementa. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuno

**Esempio di UID:**

—<UID>—

Se è abilitata l'istruzione UID disabilita la commutazione tra i task utente.

**Altri formati:**

Formato:	Sintassi:
testo neutro	UID ( ) ;
testo ASCII	UID

**Istruzioni correlate:** UIE

**Abilita interrupt utente (UIE)** L'istruzione UIE è un'istruzione di uscita.

—{UIE}—

**Operandi:** nessuno

**Descrizione:** L'istruzione UIE riabilita la commutazione tra i task utente.

Se l'istruzione UIE è abilitata ed il contatore interno è maggiore di zero, il contatore viene decrementato. Quando il contatore raggiunge lo zero, il task correntemente in esecuzione può essere nuovamente interrotto. A questo punto possono essere eseguiti quei task a priorità più elevata a cui era stato impedito di interrompere il task corrente.

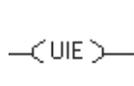
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il contatore interno dell'UID decresce. Se il conteggio è pari a 0, i task a maggiore priorità possono interrompere il task corrente. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuno

**Esempio di UIE:**



Se è abilitata l'istruzione UIE riabilita la commutazione tra i task utente.

**Altri formati:**

Formato:	Sintassi:
testo neutro	UIE ( ) ;
testo ASCII	UIE

**Istruzioni correlate:** UID

**Istruzione sempre falso (AFI)** L'istruzione AFI è un'istruzione di ingresso.

—[ AFI ]—

**Operandi:** nessuno

**Descrizione:** L'istruzione AFI imposta la sua condizione del ramo di uscita su falso.

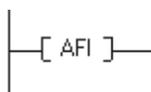
**Esecuzione:**

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su falso.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuno

**Esempio di AFI:** Utilizzare l'istruzione AFI per disabilitare temporaneamente un ramo mentre si esegue il debugging di un programma.



Se è abilitata, l'istruzione AFI disabilita tutte le istruzioni di questo ramo.

**Altri formati:**

Formato:	Sintassi:
testo neutro	AFI ( ) ;
testo ASCII	AFI

**Istruzioni correlate:** MCR, NOP, TND

**Nessuna operazione (NOP)** L'istruzione NOP è un'istruzione di ingresso e di uscita

-[NOP]-

**Operandi:** nessuno

**Descrizione:** L'istruzione NOP funziona come un marcatore di posizione.

È possibile inserire l'istruzione NOP in una posizione qualsiasi del ramo. Se è abilitata, l'istruzione NOP non esegue alcuna operazione. Se è disabilitata, l'istruzione NOP non esegue alcuna operazione.

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuno

**Esempio di NOP:** Questa istruzione è utile per localizzare diramazioni incondizionate inserendo un'istruzione NOP sulla diramazione.



L'istruzione NOP aggira l'istruzione XIC per abilitare l'uscita.

**Altri formati:**

Formato:	Sintassi:
testo neutro	NOP ( ) ;
testo ASCII	NOP

**Istruzioni correlate:** AFI, MCR, TND

## **Istruzioni For/Break (FOR, BRK, RET)**

### **Introduzione**

Utilizzare l'istruzione FOR per richiamare ripetutamente una subroutine. Utilizzare l'istruzione BRK per interrompere l'esecuzione di una subroutine.

<b>Se si desidera:</b>	<b>Utilizzare questa istruzione:</b>	<b>Vedere pagina:</b>
Eeguire ripetutamente una routine.	FOR	11-2
Terminare l'esecuzione ripetuta di una routine.	BRK	11-5
Ritornare all'istruzione FOR.	RET	11-6

**For (FOR)**

L'istruzione FOR è un'istruzione di uscita.

**Operandi:**

FOR	
For	
Routine name	?
Index	?
	??
Initial value	?
Terminal value	?
Step size	?

Operando:	Tipo:	Formato:	Descrizione:
Routine name	ROUTINE	nome routine	routine da eseguire
Index	DINT	tag	calcola quante volte è stata eseguita la routine
Initial value	SINT INT DINT	immediato tag	valore iniziale dell'indice
Terminal value	SINT INT DINT	immediato tag	valore a cui interrompere l'esecuzione della routine
Step size	SINT INT DINT	immediato tag	valore da sommare all'indice ogni volta che l'istruzione FOR esegue la routine

**Descrizione:** L'istruzione FOR esegue una subroutine ripetutamente.

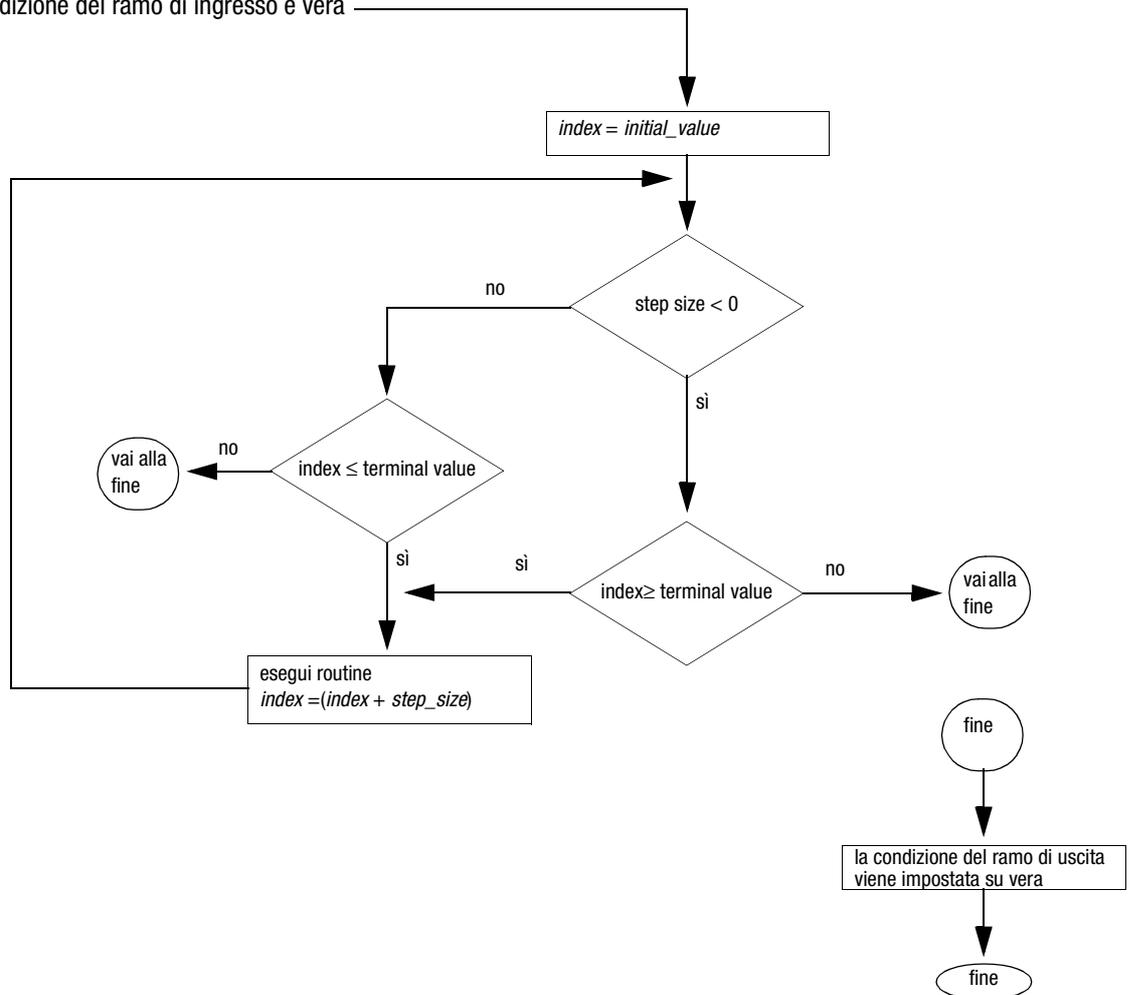
Se è abilitata, l'istruzione FOR esegue ripetutamente la routine fino a quando il valore Index non supera il Terminal value. Questa istruzione non passa parametri alla routine.

Ogni volta che l'istruzione FOR esegue la routine essa aggiunge il valore Step size all'Index.

Fare attenzione a non creare troppe iterazioni in un singola scansione. Un numero eccessivo di ripetizioni può provocare il timeout del watchdog del controllore e quindi un errore grave.

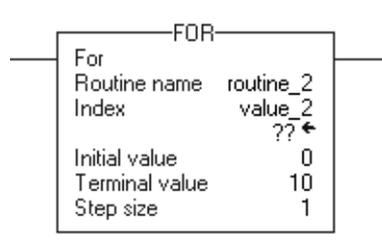
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso. Il controllore esegue la subroutine una volta. Se vi sono delle istruzioni FOR ricorsive che richiamano la stessa subroutine, questa viene prescandita solo la prima volta. Se vi sono più istruzioni FOR (non ricorsive) che richiamano la stessa subroutine, questa viene prescandita ogni volta.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di FOR:**

Se è abilitata, l'istruzione FOR esegue ripetutamente la *routine\_2* e, ogni volta, fa incrementare *value\_2* di 1. Quando *value\_2* è > 10 oppure se è abilitata un'istruzione BRK, l'istruzione FOR non esegue più la *routine\_2*.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>FOR(<i>routine_name</i>,<i>index</i>,<i>initial_value</i>,<i>terminal_value</i>,<i>step_size</i>);</code>
testo ASCII	<code>FOR <i>routine_name index initial_value terminal_value step_size</i></code>

**Istruzioni correlate:** BRK, JMP, LBL, JSR, SBR, RET

**Break (BRK)**

L'istruzione BRK è un'istruzione di uscita.

**Operandi:** nessuna

**Descrizione:** L'istruzione BRK interrompe l'esecuzione di una routine richiamata da un'istruzione FOR.

Se è abilitata, l'istruzione BRK esce dalla routine e riporta il controllore all'istruzione seguente alla FOR.

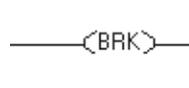
Se vi sono delle istruzioni FOR annidate, l'istruzione BRK riporta il controllo all'istruzione FOR più interna.

**Esecuzione:**

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su vero. L'esecuzione ritorna all'istruzione successiva all'istruzione FOR chiamante.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:** nessuna

**Esempio di BRK:**

Se è abilitata, l'istruzione BRK interrompe l'esecuzione della routine corrente e ritorna all'istruzione successiva all'istruzione FOR chiamante.

**Altri formati:**

Formato:	Sintassi:
testo neutro	BRK ( ) ;
testo ASCII	BRK

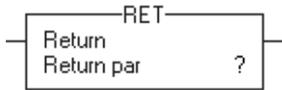
**Istruzioni correlate:** FOR, JMP, LBL, JSR, SBR, RET

## Ritorno (RET)

L'istruzione RET è un'istruzione di uscita.

**Operandi:** nessuno

**Descrizione:** L'istruzione RET riporta all'istruzione FOR chiamante. L'istruzione FOR non utilizza parametri. L'istruzione FOR ignora qualsiasi parametro inserito in un'istruzione RET.



Se è abilitata, l'istruzione RET riporta all'istruzione FOR. L'istruzione FOR fa incrementare il valore Index del valore Step size ed esegue nuovamente la subroutine. Se il valore Index supera il Terminal value, l'istruzione FOR viene completata e l'esecuzione continua con l'istruzione seguente l'istruzione FOR.

Per terminare l'esecuzione di una subroutine è possibile anche utilizzare un'istruzione TND.

### Esecuzione:

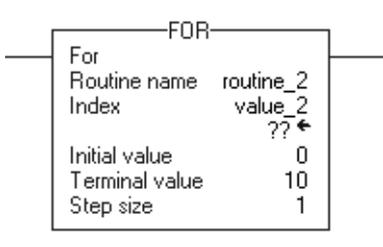
Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

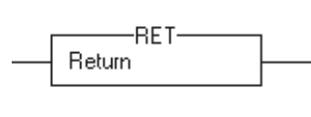
**Condizioni di errore:** nessuna

### Esempio di RET:

routine chiamante



subroutine



Se è abilitata, l'istruzione RET riporta all'istruzione FOR chiamante. L'istruzione FOR o esegue nuovamente la subroutine e fa incrementare il valore Index del valore Step size oppure, se il valore Index supera il Terminal value, l'istruzione viene completata e l'esecuzione continua con l'istruzione seguente all'istruzione FOR.

Se è abilitata, l'istruzione FOR esegue ripetutamente la routine\_2 e, ogni volta, fa incrementare value\_2 di 1. Quando value\_2 è > 10 oppure se è abilitata un'istruzione BRK, l'istruzione FOR non esegue più la routine\_2.

### Altri formati:

Formato:	Sintassi:
testo neutro	RET ( ) ;
testo ASCII	RET

**Istruzioni correlate:** BRK, JMP, LBL, TND

## **Istruzioni speciali (FBC, DDT, DTR, PID)**

### **Introduzione**

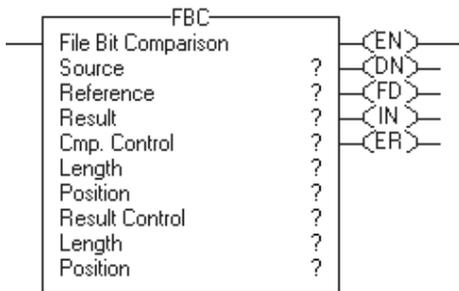
Le istruzioni speciali eseguono operazioni specifiche per applicazioni.

<b>Se si desidera:</b>	<b>Utilizzare questa istruzione:</b>	<b>Vedere pagina:</b>
Confrontare dei dati con un riferimento conosciuto e registrare le eventuali differenze.	FBC	12-2
Confrontare dei dati con un riferimento conosciuto, registrare le eventuali differenze ed aggiornare il riferimento in modo che corrisponda ai dati di origine.	DDT	12-9
Passare i dati di origine attraverso una maschera e confrontare i risultati con i dati di riferimento, quindi, scrivere i dati di origine nel riferimento per il successivo confronto.	DTR	12-16
Controllare un anello PID.	PID	12-19

## Confronto bit di file (FBC)

L'istruzione FBC è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	DINT	tag matrice	matrice da confrontare con il riferimento <b>non</b> usare CONTROL.POS nell'indice
Reference	DINT	tag matrice	matrice da confrontare con l'origine <b>non</b> usare CONTROL.POS nell'indice
Result	DINT	tag matrice	matrice dove memorizzare il risultato <b>non</b> usare CONTROL.POS negli indici
Cmp control	CONTROL	struttura	struttura di controllo per il confronto
Length	DINT	immediato	numero di bit da confrontare
Position	DINT	immediato	posizione corrente nell'origine il valore iniziale generalmente è 0
Result control	CONTROL	struttura	struttura di controllo per i risultati
Length	DINT	immediato	numero di posizioni di memorizzazione nel risultato
Position	DINT	immediato	posizione corrente nel risultato il valore iniziale generalmente è 0

**ATTENZIONE:** utilizzare tag diverse per la struttura di controllo del confronto e la struttura di controllo del risultato. Se si utilizza la stessa tag si potrebbe verificare un funzionamento anomalo della macchina con possibili danni alle apparecchiature o alle persone.



**Struttura CONTROL di confronto:**

<b>Mnemonico:</b>	<b>Tipo di dati:</b>	<b>Descrizione:</b>
.EN	BOOL	Il bit di abilitazione indica che l'istruzione FBC è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando l'istruzione FBC confronta l'ultimo bit delle matrici Source e Reference.
.FD	BOOL	Il bit di trovato viene impostato ogni volta che l'istruzione FBC rileva una mancata corrispondenza (modalità "uno alla volta") o dopo avere registrato tutte le mancate corrispondenze (modalità "tutti").
.IN	BOOL	Il bit di inibizione indica la modalità di ricerca dell'FBC. 0 = Modalità Tutti 1 = Modalità Uno alla volta
.ER	BOOL	Il bit di errore viene impostato se il confronto .POS < 0, il confronto .LEN < 0, il risultato .POS < 0 o il risultato .LEN < 0. L'istruzione interrompe l'esecuzione fino a quando il programma non azzeri il bit .ER.
.LEN	DINT	Il valore Length indica il numero di bit da confrontare.
.POS	DINT	Il valore di posizione indica il bit corrente.

**Struttura CONTROL risultato:**

<b>Mnemonico:</b>	<b>Tipo di dati:</b>	<b>Descrizione:</b>
.DN	BOOL	Il bit di fine viene impostato quando la matrice Result è piena.
.LEN	DINT	Il valore Length indica il numero di posizioni di memorizzazione nella matrice Result.
.POS	DINT	Il valore di posizione identifica la posizione corrente nella matrice Result.

**Descrizione:** L'istruzione FBC confronta i bit della matrice Source con i bit della matrice Reference.

Se è abilitata, l'istruzione FBC confronta i bit della matrice Source con i bit della matrice Reference e registra nella matrice Result il numero di bit di ogni mancata corrispondenza.

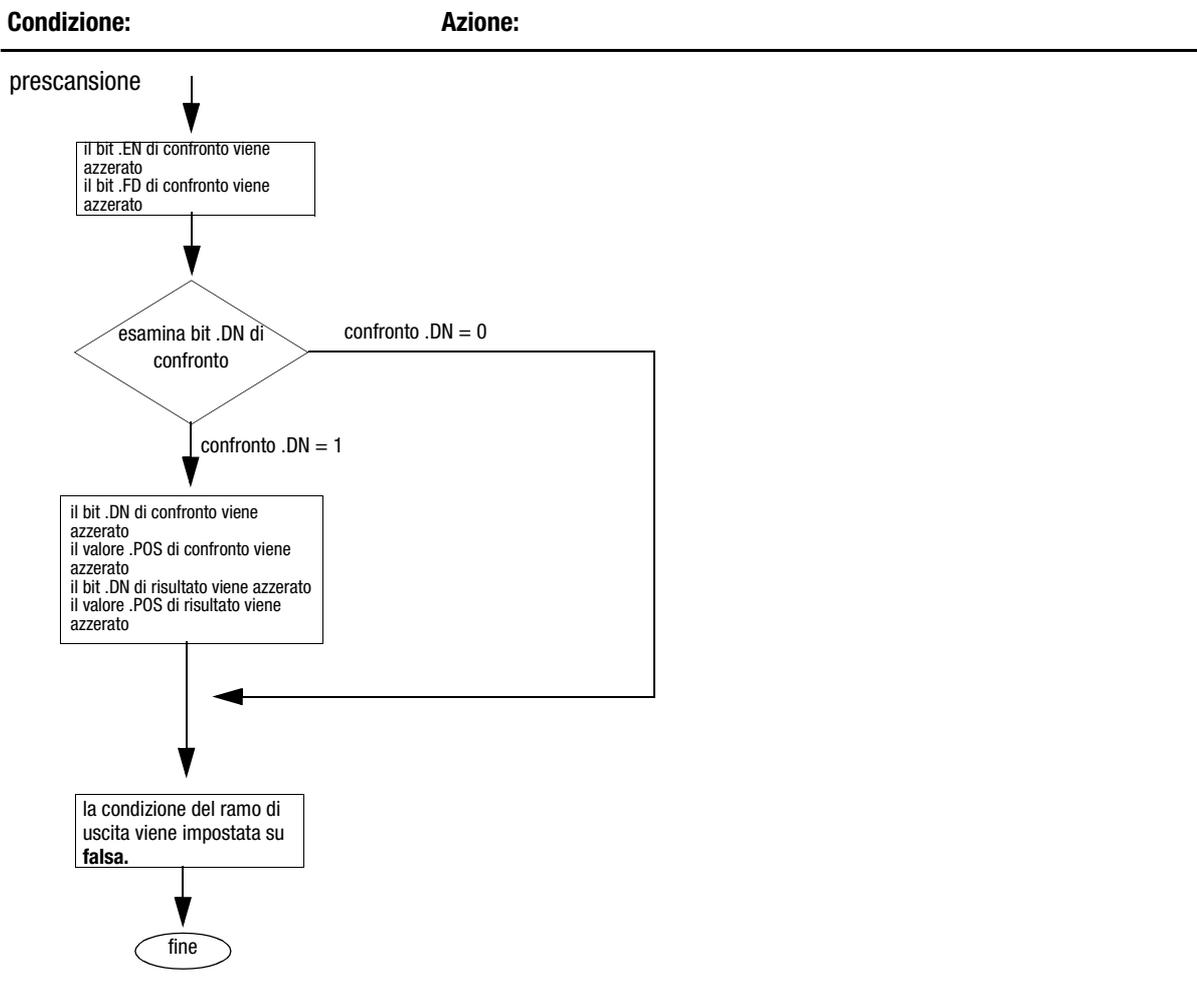
L'istruzione FBC opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

La differenza tra l'istruzione DDT e FBC consiste nel fatto che ogni volta che un'istruzione DDT trova una mancata corrispondenza, l'istruzione cambia il bit di riferimento facendolo corrispondere con il bit di origine. L'istruzione FBC non cambia il bit di riferimento.

### Scelta della modalità di ricerca

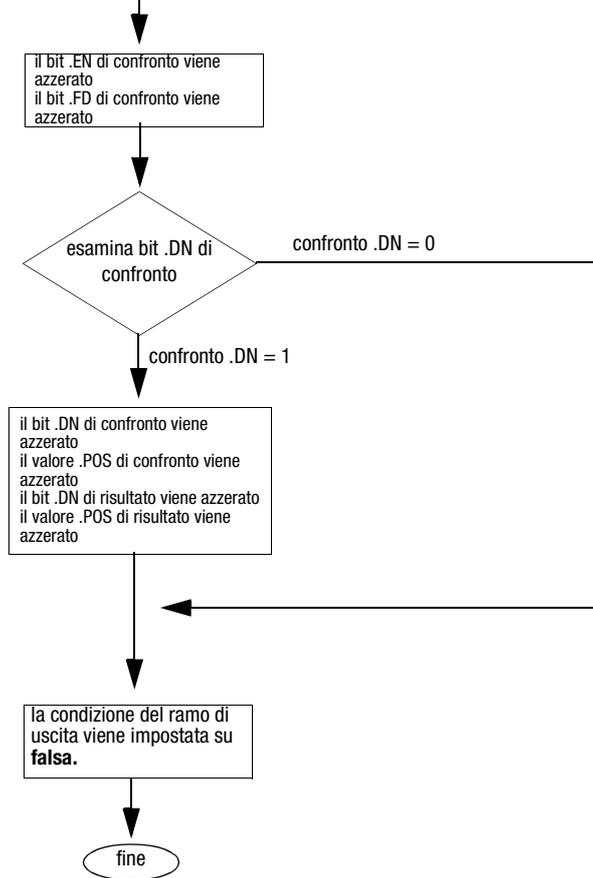
Se si vuole rilevare:	Selezionare questa modalità:
Una mancata corrispondenza per volta	Impostare il bit .IN nella struttura CONTROL del confronto. Ogni volta che la condizione del ramo di ingresso passa da falsa a vera, l'istruzione FBC cerca la successiva mancata corrispondenza tra le matrici Source e Reference. Quando trova una mancata corrispondenza, l'istruzione imposta il bit .FD, registra la posizione della mancata corrispondenza ed interrompe l'esecuzione.
Tutte le mancate corrispondenze	Azzerare il bit .IN nella struttura CONTROL del confronto. Ogni volta che la condizione del ramo di ingresso passa da falsa a vera, l'istruzione FSC cerca tutte le mancate corrispondenze tra le matrici Source e Reference.

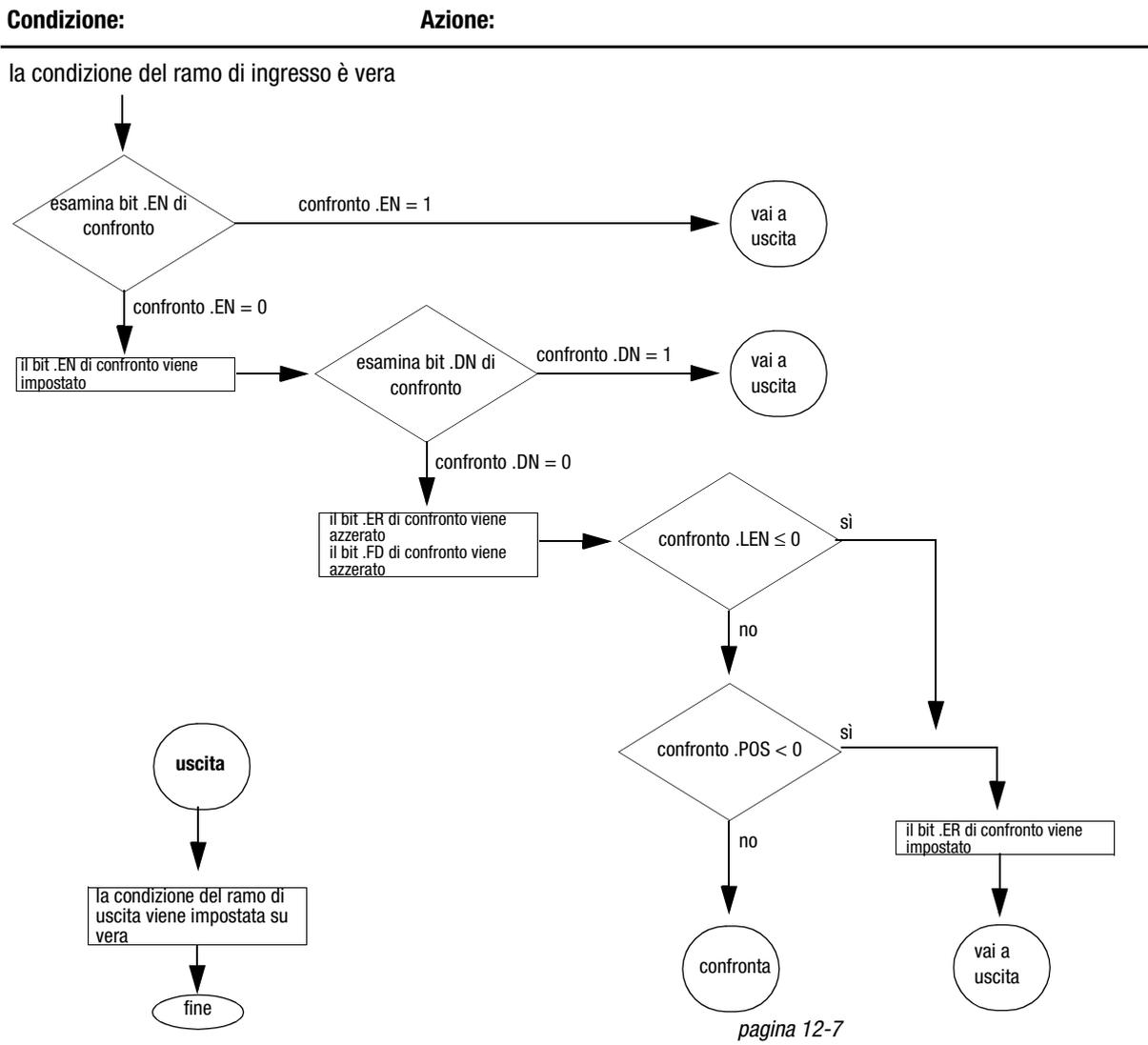
### Esecuzione:



**Condizione:****Azione:**

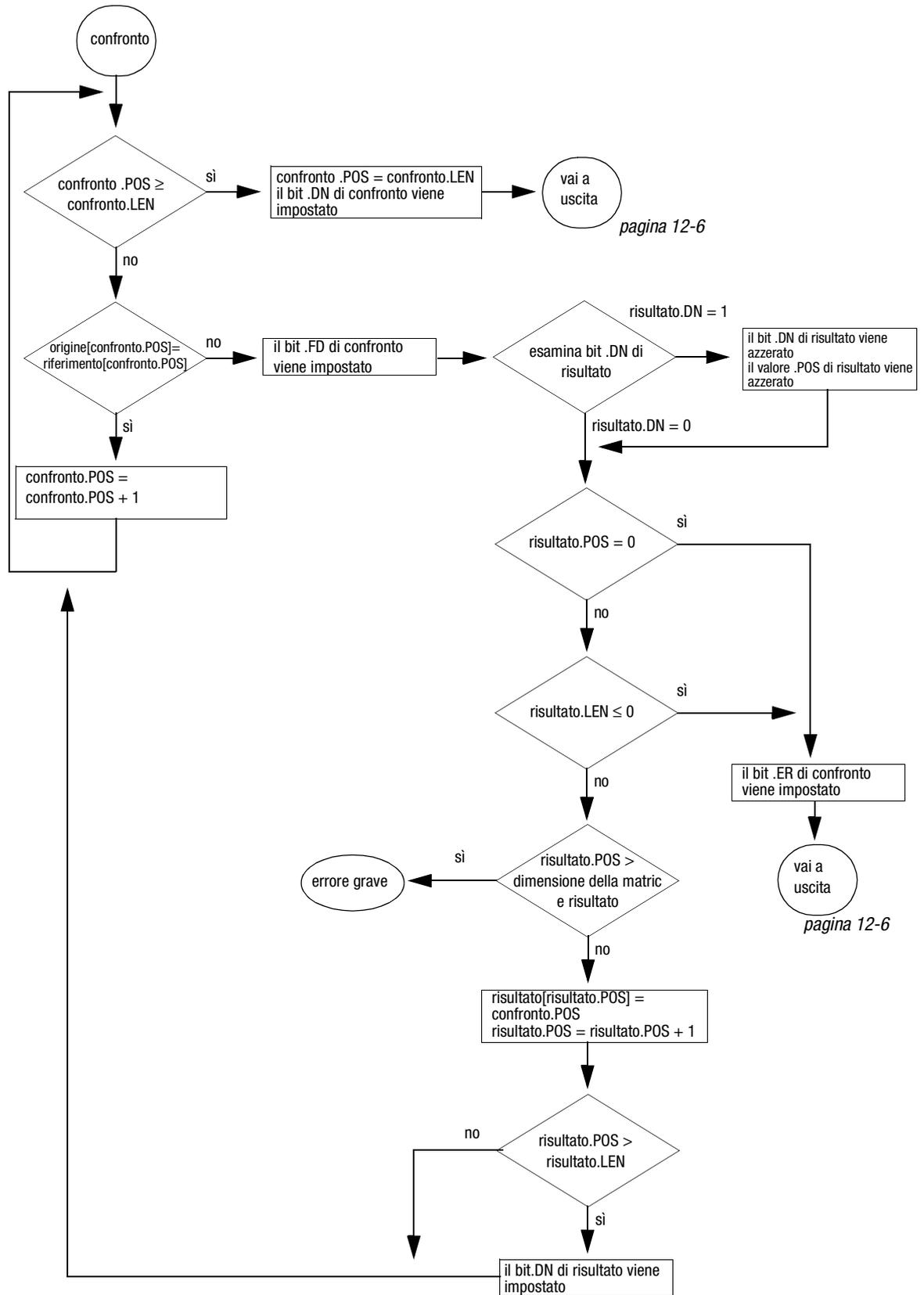
la condizione del ramo di ingresso è falsa





Condizione:

Azione:

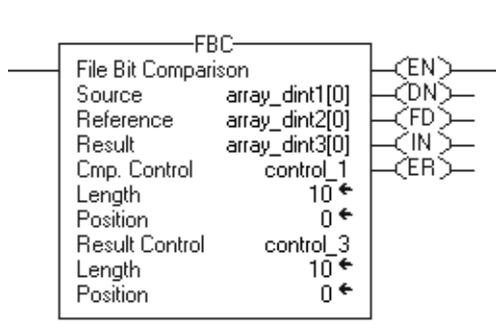


**Indicatori di stato aritmetico:** non influenzati

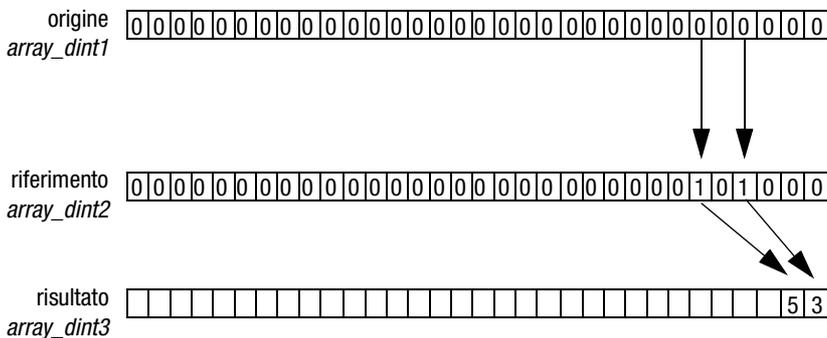
**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
Result.POS > dimensione della matrice Result	4	20

**Esempio di FBC:**



Se è abilitata, l'istruzione FBC confronta *array\_dint1* di origine con la *array\_dint2* di riferimento e memorizza le posizioni delle mancate corrispondenze nella *array\_dint3* di risultato.



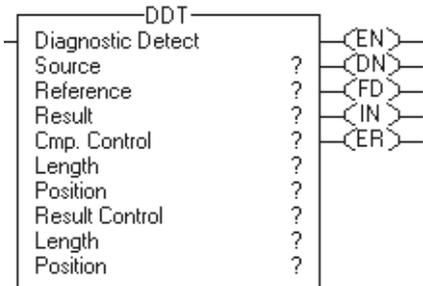
**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>FBC(source,reference,result, cmp_control,length,position, result_control,length,position);</code>
testo ASCII	<code>FBC source reference result cmp_control length position result_control length position</code>

**Istruzioni correlate:** DDT, DTR

**Rileva diagnostica (DDT)**

L'istruzione DDT è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source	DINT	tag matrice	matrice da confrontare con il riferimento <b>non</b> usare CONTROL.POS nell'indice
Reference	DINT	tag matrice	matrice da confrontare con l'origine <b>non</b> usare CONTROL.POS nell'indice
Result	DINT	tag matrice	matrice dove memorizzare i risultati <b>non</b> usare CONTROL.POS nell'indice
Cmp Control	CONTROL	struttura	struttura di controllo per il confronto
Length	DINT	immediato	numero di bit da confrontare
Position	DINT	immediato	posizione corrente nell'origine il valore iniziale generalmente 0
Result control	CONTROL	struttura	struttura di controllo per i risultati
Length	DINT	immediato	numero di posizioni di memorizzazione nel risultato
Position	DINT	immediato	posizione corrente nel risultato il valore iniziale generalmente 0

**ATTENZIONE**

**ATTENZIONE:** utilizzare tag diverse per la struttura di controllo del confronto e la struttura di controllo del risultato. Se si utilizza la stessa tag si potrebbe verificare un funzionamento anomalo della macchina con possibili danni alle apparecchiature o alle persone.

**Struttura CONTROL di confronto:**

Mnemonico:	Tipo di dati:	Descrizione:
.EN	BOOL	Il bit di abilitazione indica che l'istruzione DDT è abilitata.
.DN	BOOL	Il bit di fine viene impostato quando l'istruzione DDT confronta l'ultimo bit delle matrici di origine e di riferimento.
.FD	BOOL	Il bit di trovato viene impostato ogni volta che l'istruzione DDT rileva una mancata corrispondenza (modalità "uno alla volta") o dopo avere registrato tutte le mancate corrispondenze (modalità "tutti").
.IN	BOOL	Il bit di inibizione indica la modalità di ricerca dell'DDT. 0 = Modalità Tutti 1 = Modalità Uno alla volta
.ER	BOOL	Il bit di errore viene impostato se il confronto .POS < 0, il confronto .LEN < 0, il risultato .POS < 0 o il risultato .LEN < 0. L'istruzione interrompe l'esecuzione fino a quando il programma non azzeri il bit .ER.
.LEN	DINT	Il valore lunghezza indica il numero di bit da confrontare.
.POS	DINT	Il valore posizione indica il bit corrente.

**Struttura CONTROL risultato:**

Mnemonico:	Tipo di dati:	Descrizione:
.DN	BOOL	Il bit di fine viene impostato quando la matrice Result è piena.
.LEN	DINT	Il valore Length indica il numero di posizioni di memorizzazione nella matrice Result.
.POS	DINT	Il valore di posizione identifica la posizione corrente nella matrice Result.

**Descrizione:** L'istruzione DDT confronta i bit della matrice Source con i bit della matrice Reference per determinare eventuali cambiamenti di stato.

Se è abilitata, l'istruzione DDT confronta i bit della matrice Source con i bit della matrice Reference, registra nella matrice Result il numero di bit di ogni mancata corrispondenza e cambia il valore del bit di Reference in modo che corrisponda con il valore del relativo bit di Source.

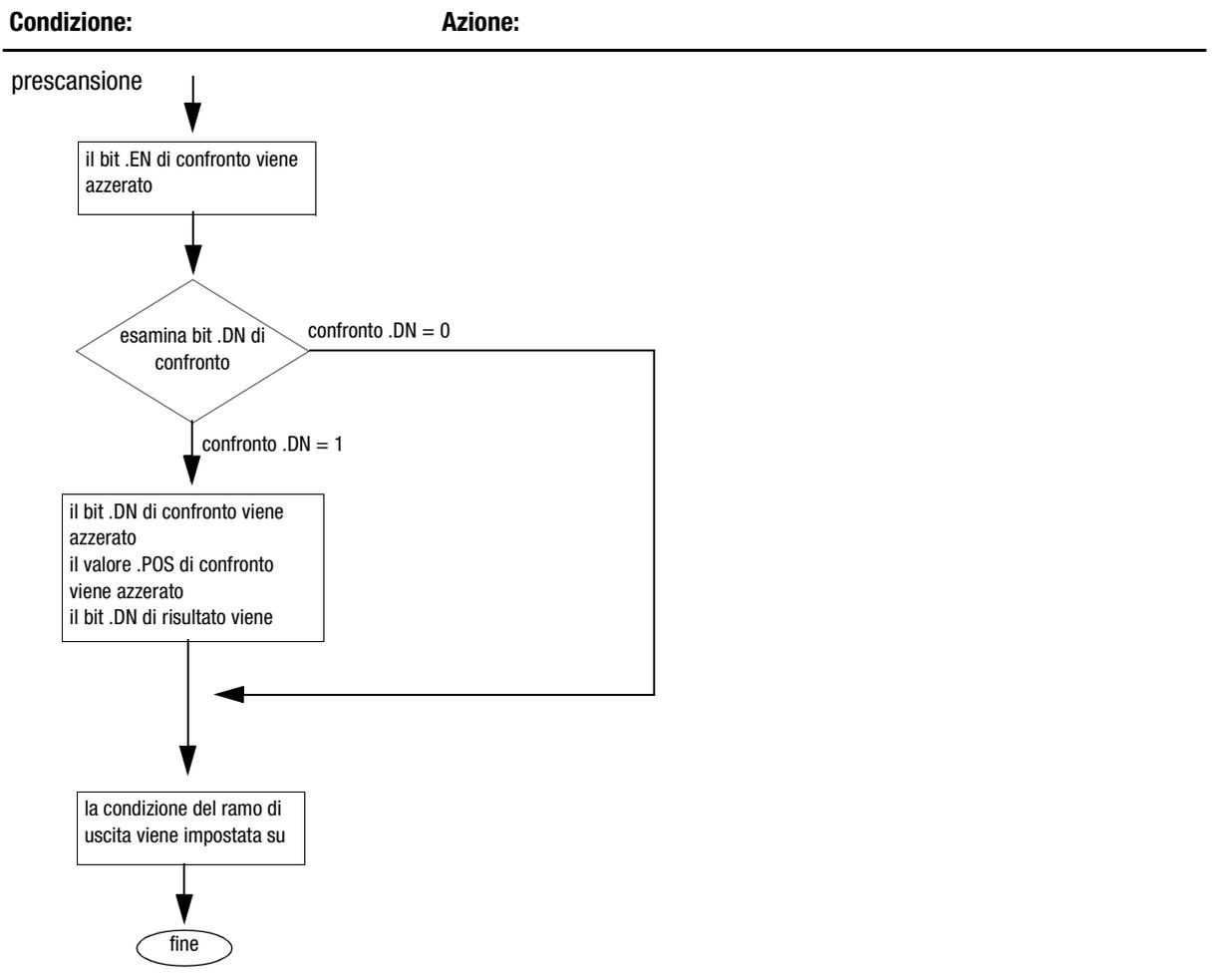
L'istruzione DDT opera su una memoria contigua. Per ulteriori informazioni, vedere *Visualizzazione di una matrice come blocco di memoria* a pagina B-4.

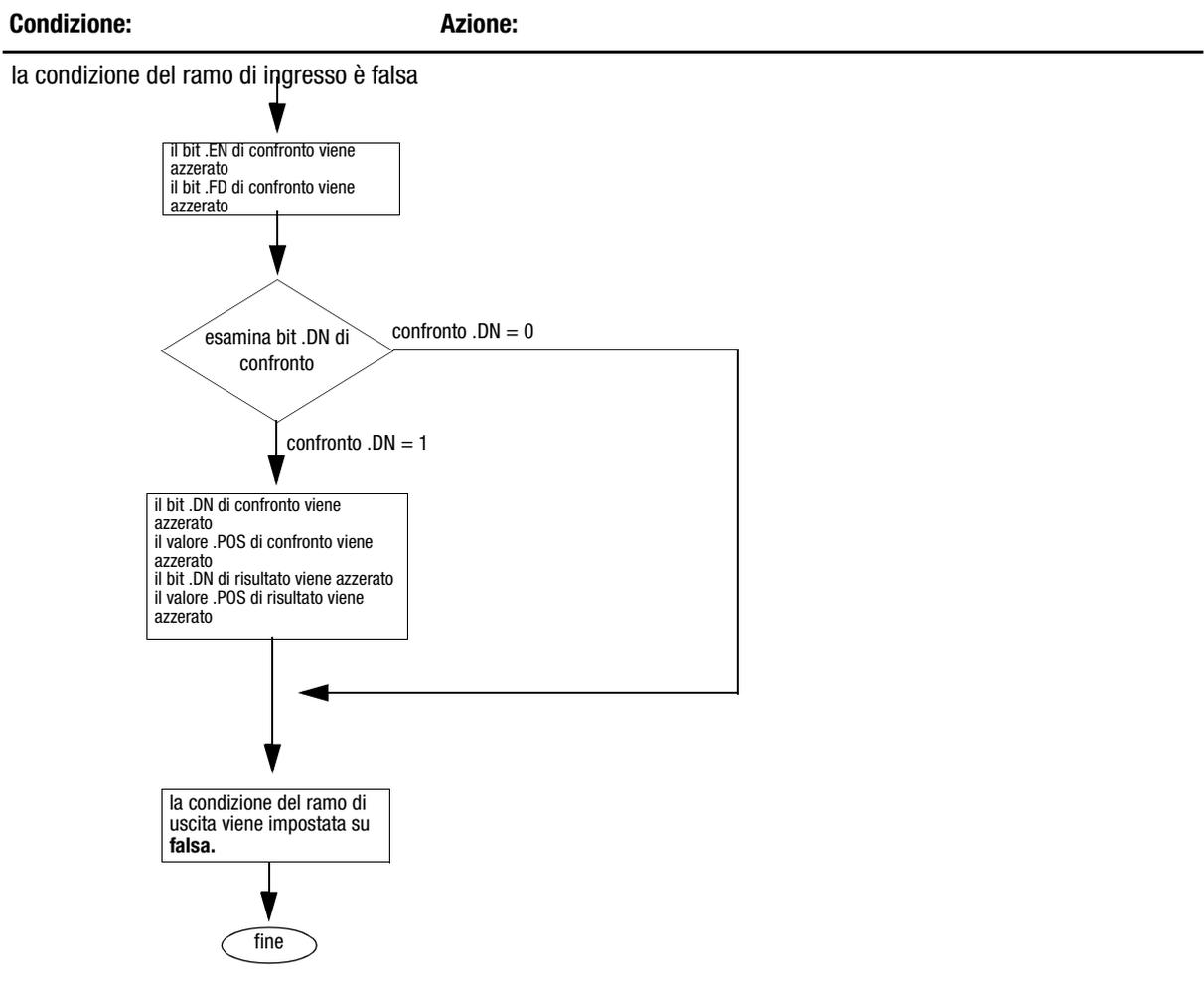
La differenza tra l'istruzione DDT e FBC consiste nel fatto che ogni volta che un'istruzione DDT trova una mancata corrispondenza, l'istruzione cambia il bit di riferimento in modo che corrisponda con il bit di origine. L'istruzione FBC non cambia il bit di riferimento.

## Scelta della modalità di ricerca

Se si vuole rilevare:	Selezionare questa modalità:
Una mancata corrispondenza per volta	Impostare il bit .IN nella struttura CONTROL del confronto. Ogni volta che la condizione del ramo di ingresso passa da falsa a vera, l'istruzione DDT cerca la successiva mancata corrispondenza tra le matrici Source e Reference. Quando trova una mancata corrispondenza, l'istruzione imposta il bit .FD, registra la posizione della mancata corrispondenza ed interrompe l'esecuzione.
Tutte le mancate corrispondenze	Azzerare il bit .IN nella struttura CONTROL del confronto. Ogni volta che la condizione del ramo di ingresso passa da falsa a vera, l'istruzione DDT cerca tutte le mancate corrispondenze tra le matrici Source e Reference.

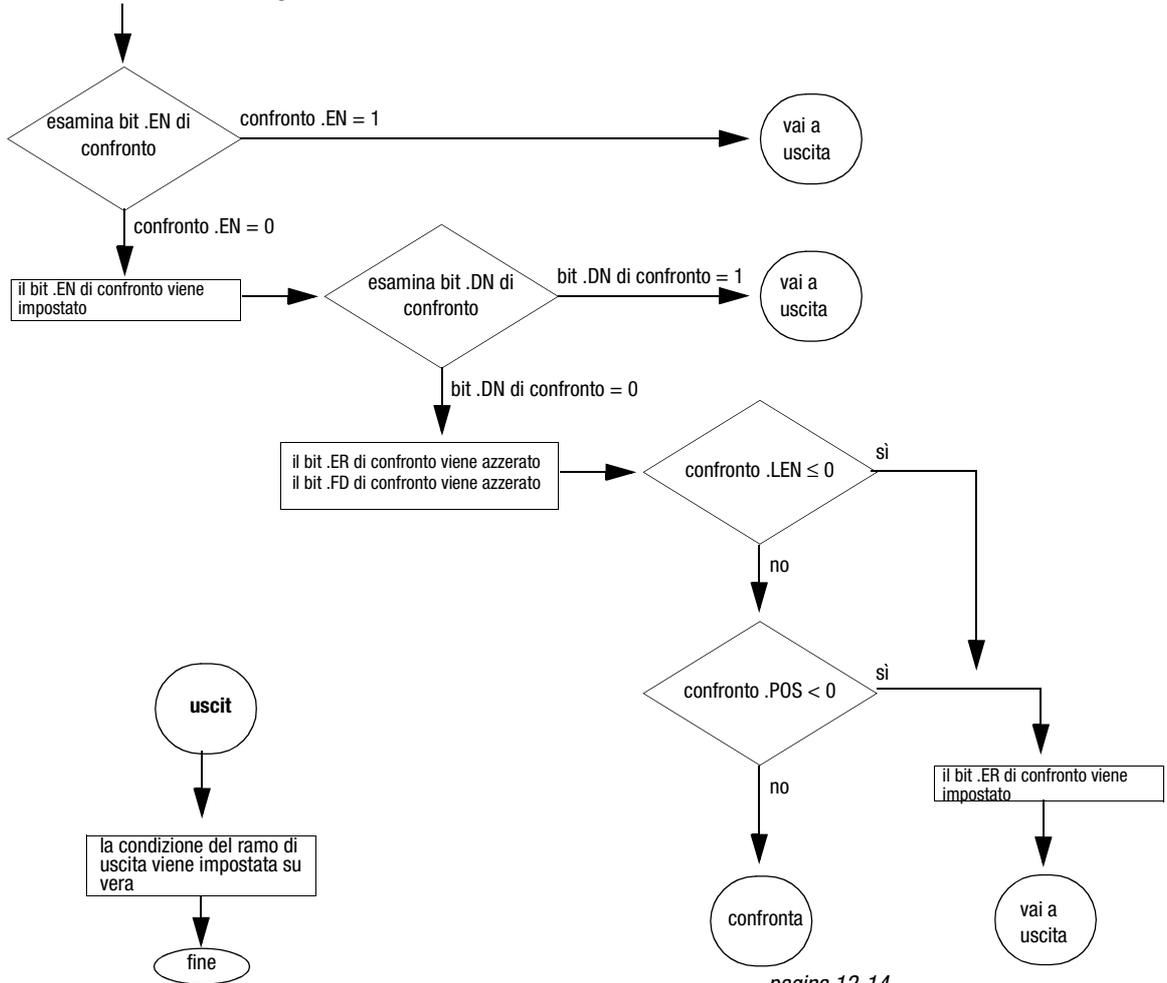
### Esecuzione:



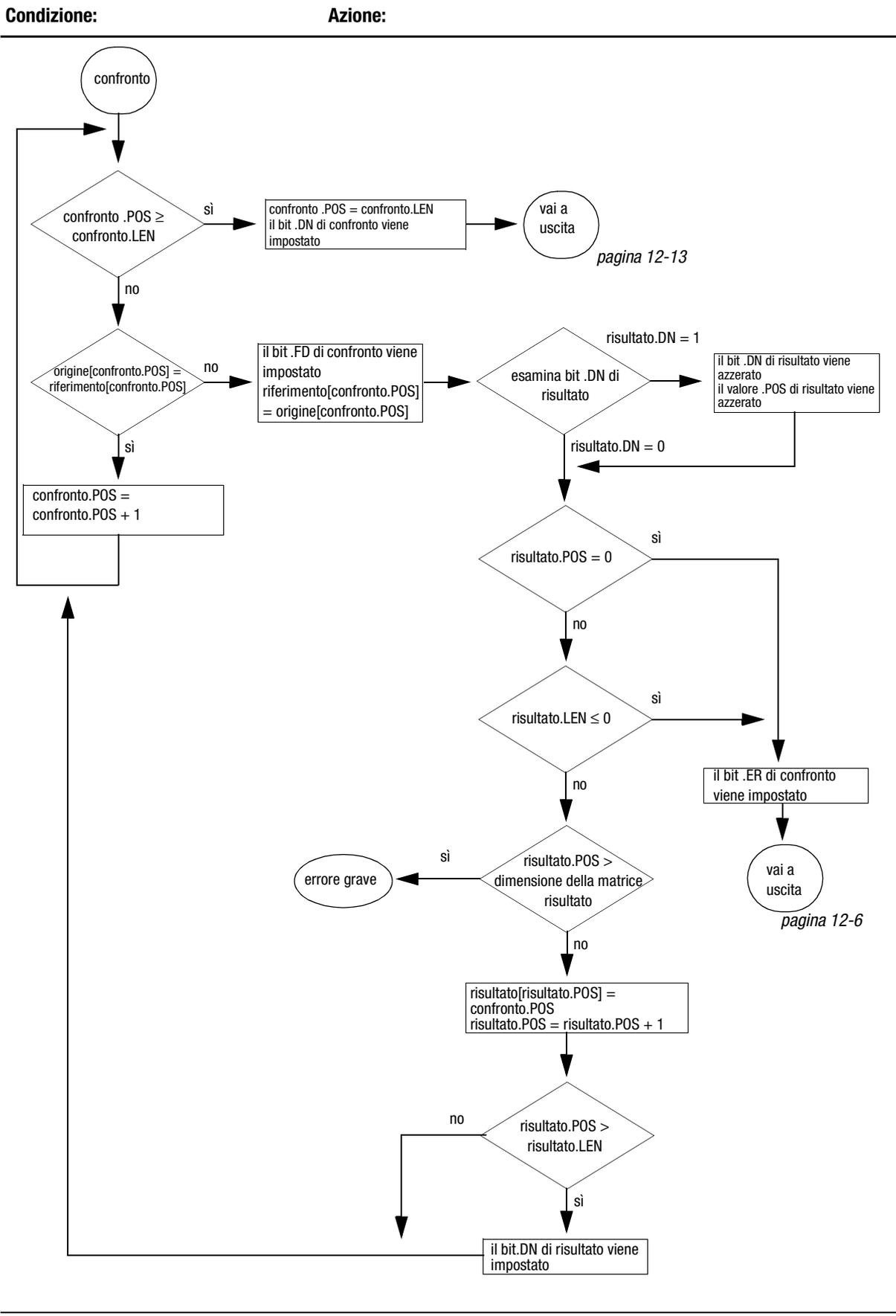


**Condizione:****Azione:**

la condizione del ramo di ingresso è vera



pagina 12-14

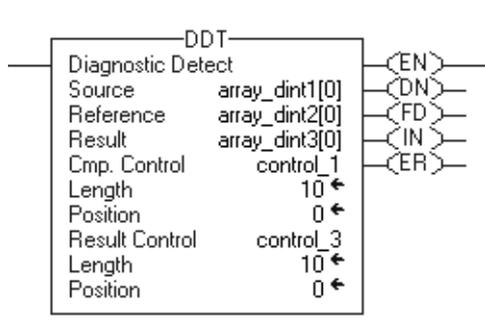


**Indicatori di stato aritmetico:** non influenzati

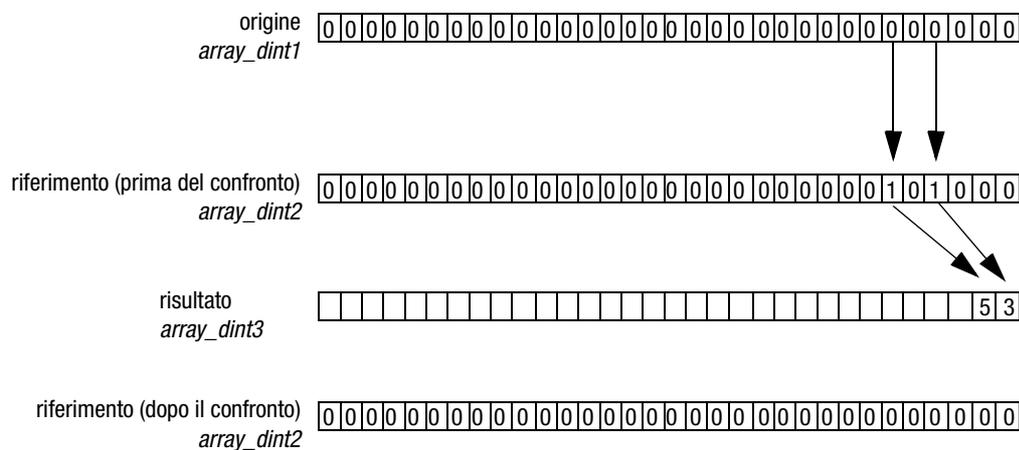
**Condizioni di errore:**

Si verifica un errore grave se:	Tipo errore:	Codice errore:
Result.POS > dimensione della matrice Result	4	20

**Esempio di DDT:**



Se è abilitata, l'istruzione DDT confronta *array\_dint1* di origine con *array\_dint2* di riferimento e memorizza le posizioni delle mancate corrispondenze nella *array\_dint3* di risultato. Il controllore cambia anche i bit non corrispondenti di *array\_dint2* di riferimento in modo che corrispondano a quelli di *array\_dint1* di origine.



**Altri formati:**

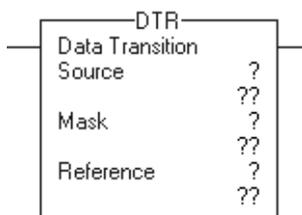
Formato:	Sintassi:
testo neutro	DDT( <i>source</i> , <i>reference</i> , <i>result</i> , <i>cmp_control</i> , <i>length</i> , <i>position</i> , <i>result_control</i> , <i>length</i> , <i>position</i> );
testo ASCII	DDT <i>source</i> <i>reference</i> <i>result</i> <i>cmp_control</i> <i>length</i> <i>position</i> <i>result_control</i> <i>length</i> <i>position</i>

**Istruzioni correlate:** FBC, DTR

## Transizione dati (DTR)

L'istruzione DTR è un'istruzione di ingresso.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	DINT	immediato tag	matrice da confrontare con il riferimento
Mask	DINT	immediato tag	quali bit bloccare o passare
Reference	DINT	tag	matrice da confrontare con l'origine

**Descrizione:** L'istruzione DTR passa il valore Source attraverso una maschera e confronta il risultato con il valore Reference. L'istruzione DTR, inoltre, scrive il valore Source mascherato nel valore Reference per il successivo confronto. Il valore Source rimane invariato.

Un "1" nella maschera significa che il bit dati viene passato. Uno "0" nella maschera significa che il bit dati viene bloccato.

Se il valore Source con maschera è diverso dal valore Reference, la condizione del ramo di uscita diventa vera per una scansione. Se il valore Source con maschera e il valore Reference sono gli stessi, la condizione del ramo di uscita è falsa.

**ATTENZIONE:** è pericoloso programmare online con questa istruzione. Se il valore Reference è diverso dal valore Source, la condizione del ramo di uscita diventa vera. Fare attenzione nell'inserire questa istruzione quando il processore si trova in modalità di Esecuzione (Run) o di Esecuzione remota (Remote Run).



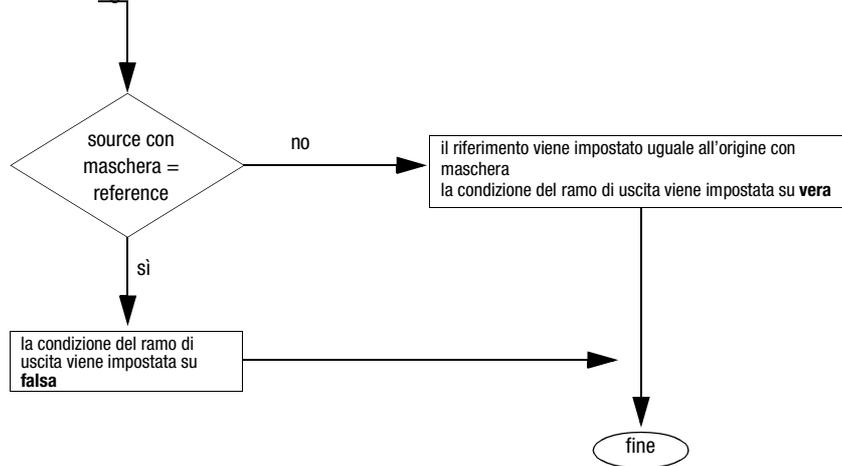
### Immissione di un valore di maschera immediato

Quando si immette una maschera, il software di programmazione imposta automaticamente valori decimali. Se si desidera immettere una maschera utilizzando un altro formato, fare precedere il prefisso corretto al valore.

Prefisso:	Descrizione:
16#	Esadecimale per esempio; 16#0F0F
8#	Ottale per esempio; 8#16
2#	Binario per esempio; 2#00110011

**Esecuzione:**

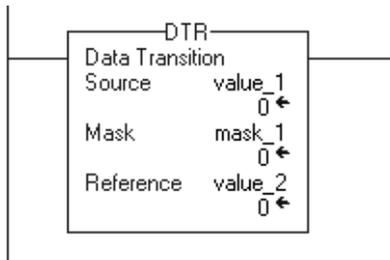
<b>Condizione:</b>	<b>Azione:</b>
prescausione	Reference = Source AND Mask. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	Reference = Source AND Mask. La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	



**Indicatori di stato aritmetico:** non influenzati

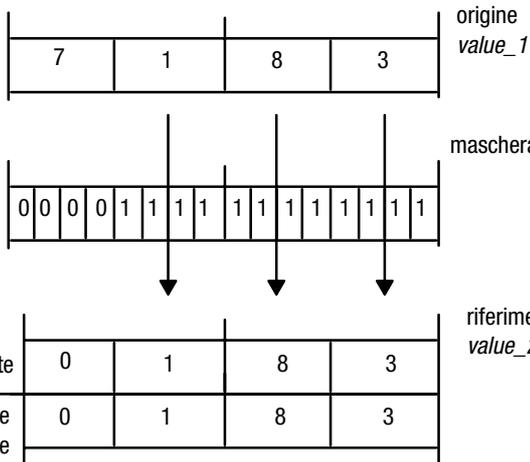
**Condizioni di errore:** nessuna

**Esempio di DTR:**



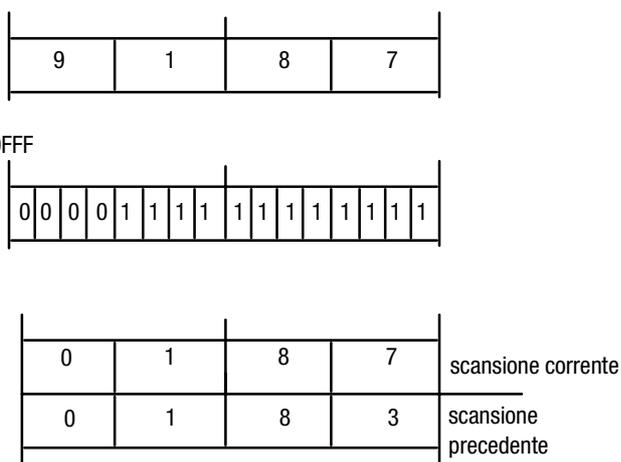
Se è abilitata, l'istruzione DTR maschera il *value\_1*. Se c'è differenza tra i due valori, la condizione del ramo di uscita viene impostata a vera.

**esempio 1**



Il ramo rimane falso fino a quando il valore di ingresso non cambia.

**esempio 2**



Il ramo rimane vero per una scansione se viene rilevato un cambiamento.

13385

Uno 0 nella maschera lascia il bit immutato.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>DTR ( source , mask , reference ) ;</code>
testo ASCII	<code>DTR source mask reference</code>

**Istruzioni correlate:** FBC, DDT

## Proporzionale, integrale e derivativo (PID)

L'istruzione PID è un'istruzione di uscita.

### Operandi:

PID	
Proportional Integral Derivative	
PID	?
Process variable	?
PV Data Type	?
Tieback	?
Control variable	?
CV Data Type	?
PID Master Loop	?
Inhold bit	?
Inhold Value	?
Setpoint	??
Process Variable	??
Output %	??

Operando:	Tipo:	Formato:	Descrizione:
PID	PID	struttura	struttura PID
Process variable	SINT INT DINT REAL	tag	valore che si vuole controllare
Tieback	SINT INT DINT REAL	immediato tag	<i>(opzionale)</i> uscita di una stazione manuale/automatica hardware che aggira l'uscita del controllore immettere 0 se non si desidera utilizzare questo parametro
Control variable	SINT INT DINT REAL	tag	valore che viene inviato al dispositivo di controllo finale (valvola, smorzatore, ecc.) se si utilizza la banda morta, la variabile di controllo deve essere REAL altrimenti essa verrà forzata a 0 se l'errore è all'interno della banda morta
PID master loop	PID	struttura	opzionale tag PID per il PID master se si esegue un controllo a cascata e questo PID è un anello slave, inserire il nome del PID master immettere 0 se non si desidera utilizzare questo parametro
Inhold bit	BOOL	tag	opzionale stato corrente del bit inhold di un canale di uscita analogico 1756 per il supporto del riavviamento graduale immettere 0 se non si desidera utilizzare questo parametro
Inhold value	SINT INT DINT REAL	tag	opzionale valore di ritorno dati da un canale di uscita analogico 1756 per il supporto del riavviamento graduale immettere 0 se non si desidera utilizzare questo parametro
Setpoint			solo visualizzazione valore corrente del setpoint
Process variable			solo visualizzazione valore corrente della variabile di processo in scala
Output %			solo visualizzazione valore in percentuale dell'uscita corrente

**Struttura PID:** Specificare una sola struttura PID per ciascuna istruzione PID.

<b>Mnemonico:</b>	<b>Tipo di dati:</b>	<b>Descrizione:</b>	
.CTL	DINT	Il membro .CTL permette di accedere ai membri di stato (bit) in una parola a 32 bit. L'istruzione PID imposta i bit 07 -15.	
		<b>Questo bit:</b>	<b>È questo membro:</b>
		31	.EN
		30	.CT
		29	.CL
		28	.PVT
		27	.DOE
		26	.SWM
		25	.CA
		24	.MO
		23	.PE
		22	.NDF
		21	.NOBC
		20	.NOZC
		<b>Questo bit:</b>	<b>E' questo membro impostato dall'istruzione PID:</b>
		15	.INI
		14	.SPOR
		13	.OLL
		12	.OLH
		11	.EWD
		10	.DVNA
		09	.DVPA
		08	.PVLA
		07	.PVHA
.SP	REAL	setpoint	
.KP	REAL	indipendente	guadagno proporzionale (adimensionale)
		dipendente	guadagno del controllore (adimensionale)
.KI	REAL	indipendente	guadagno integrale (1/sec.)
		dipendente	tempo integrale (minuti per ripetizione)
.KD	REAL	indipendente	guadagno derivativo (secondi)
		dipendente	tempo derivativo (minuti)
.BIAS	REAL	% della compensazione anticipata o del bias	
.MAXS	REAL	fondo scala superiore in unità ingegneristiche	
.MINS	REAL	fondo scala inferiore in unità ingegneristiche	
.DB	REAL	banda morta in unità ingegneristica	

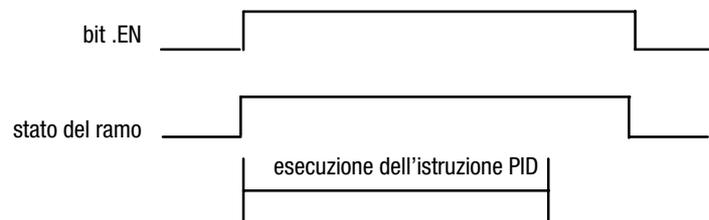
<b>Mnemonico:</b>	<b>Tipo di dati:</b>	<b>Descrizione:</b>
.SO	REAL	% di uscita impostata
.MAXO	REAL	limite uscita massimo (% dell'uscita)
.MINO	REAL	limite uscita minimo (% dell'uscita)
.UPD	REAL	tempo di aggiornamento dell'anello (secondi)
.PV	REAL	valore PV in scala
.ERR	REAL	valore errore in scala
.OUT	REAL	% uscita
.PVH	REAL	limite superiore allarme della variabile di processo
.PVL	REAL	limite inferiore allarme della variabile di processo
.DVP	REAL	limite allarme della deviazione positiva
.DVN	REAL	limite allarme della deviazione negativa
.PVDB	REAL	banda morta dell'allarme della variabile di processo
.DVDB	REAL	banda morta dell'allarme della deviazione
.MAXI	REAL	valore PV massimo (ingresso non in scala)
.MINI	REAL	valore PV minimo (ingresso non in scala)
.TIE	REAL	valore di tieback per il controllo manuale
.MAXCV	REAL	valore CV massimo (equivalente al 100 %)
.MINCV	REAL	valore CV minimo (equivalente allo 0 %)
.MINTIE	REAL	valore tieback minimo (equivalente al 100 %)
.MAXTIE	REAL	valore tieback massimo (equivalente allo 0 %)

<b>Mnemonico:</b>	<b>Tipo di dati:</b>	<b>Descrizione:</b>																																				
.DATA	REAL[17]	Il membro .DATA memorizza:																																				
		<table border="1"> <thead> <tr> <th><b>Elemento:</b></th> <th><b>Descrizione:</b></th> </tr> </thead> <tbody> <tr> <td>.DATA[0]</td> <td>accumulazione integrale</td> </tr> <tr> <td>.DATA[1]</td> <td>valore temporaneo attenuazione derivativa</td> </tr> <tr> <td>.DATA[2]</td> <td>valore .PV precedente</td> </tr> <tr> <td>.DATA[3]</td> <td>valore .ERR precedente</td> </tr> <tr> <td>.DATA[4]</td> <td>valore .SP precedente valido</td> </tr> <tr> <td>.DATA[5]</td> <td>costante di conversione in scala della percentuale</td> </tr> <tr> <td>.DATA[6]</td> <td>costante di conversione in scala della .PV</td> </tr> <tr> <td>.DATA[7]</td> <td>costante di conversione in scala della derivativa</td> </tr> <tr> <td>.DATA[8]</td> <td>valore .KP precedente</td> </tr> <tr> <td>.DATA[9]</td> <td>valore .KI precedente</td> </tr> <tr> <td>.DATA[10]</td> <td>valore .KD precedente</td> </tr> <tr> <td>.DATA[11]</td> <td>guadagno dipendente .KP</td> </tr> <tr> <td>.DATA[12]</td> <td>guadagno dipendente .KI</td> </tr> <tr> <td>.DATA[13]</td> <td>guadagno dipendente .KD</td> </tr> <tr> <td>.DATA[14]</td> <td>valore .CV precedente</td> </tr> <tr> <td>.DATA[15]</td> <td>costante di riconversione dalla scala della .CV</td> </tr> <tr> <td>.DATA[16]</td> <td>costante di riconversione dalla scala del tieback</td> </tr> </tbody> </table>	<b>Elemento:</b>	<b>Descrizione:</b>	.DATA[0]	accumulazione integrale	.DATA[1]	valore temporaneo attenuazione derivativa	.DATA[2]	valore .PV precedente	.DATA[3]	valore .ERR precedente	.DATA[4]	valore .SP precedente valido	.DATA[5]	costante di conversione in scala della percentuale	.DATA[6]	costante di conversione in scala della .PV	.DATA[7]	costante di conversione in scala della derivativa	.DATA[8]	valore .KP precedente	.DATA[9]	valore .KI precedente	.DATA[10]	valore .KD precedente	.DATA[11]	guadagno dipendente .KP	.DATA[12]	guadagno dipendente .KI	.DATA[13]	guadagno dipendente .KD	.DATA[14]	valore .CV precedente	.DATA[15]	costante di riconversione dalla scala della .CV	.DATA[16]	costante di riconversione dalla scala del tieback
<b>Elemento:</b>	<b>Descrizione:</b>																																					
.DATA[0]	accumulazione integrale																																					
.DATA[1]	valore temporaneo attenuazione derivativa																																					
.DATA[2]	valore .PV precedente																																					
.DATA[3]	valore .ERR precedente																																					
.DATA[4]	valore .SP precedente valido																																					
.DATA[5]	costante di conversione in scala della percentuale																																					
.DATA[6]	costante di conversione in scala della .PV																																					
.DATA[7]	costante di conversione in scala della derivativa																																					
.DATA[8]	valore .KP precedente																																					
.DATA[9]	valore .KI precedente																																					
.DATA[10]	valore .KD precedente																																					
.DATA[11]	guadagno dipendente .KP																																					
.DATA[12]	guadagno dipendente .KI																																					
.DATA[13]	guadagno dipendente .KD																																					
.DATA[14]	valore .CV precedente																																					
.DATA[15]	costante di riconversione dalla scala della .CV																																					
.DATA[16]	costante di riconversione dalla scala del tieback																																					
.EN	BOOL	abilitato																																				
.CT	BOOL	tipo in cascata (0=slave; 1=master)																																				
.CL	BOOL	anello in cascata (0=no; 1=si)																																				
.PVT	BOOL	tracciabilità variabile di processo (0=no; 1=si)																																				
.DOE	BOOL	derivata di (0=PV; 1=errore)																																				
.SWM	BOOL	modalità manuale software (0=no-auto; 1=si- manuale sw)																																				
.CA	BOOL	azione di controllo (0 indica E=SP-PV; 1 indica E=PV-SP)																																				
.MO	BOOL	modalità stazione (0=automatica; 1=manuale)																																				
.PE	BOOL	equazione PID (0=indipendente; 1=dipendente)																																				
.NDF	BOOL	attenuazione derivativa (0=no; 1=si)																																				
.NOBC	BOOL	calcolo bias (0=no; 1=si)																																				
.NOZC	BOOL	banda morta con attraversamento dello zero (0=no; 1=si per banda morta)																																				
.INI	BOOL	PID inizializzato (0=no; 1=si)																																				
.SPOR	BOOL	setpoint fuori gamma (0=no; 1=si)																																				
.OLL	BOOL	CV al di sotto del limite minimo di uscita (0=no; 1=si)																																				
.OLH	BOOL	CV al di sopra del limite massimo di uscita (0=no; 1=si)																																				
.EWD	BOOL	errore all'interno della banda morta (0=no; 1=si)																																				
.DVNA	BOOL	deviazione con allarme basso (0=no; 1=si)																																				

Mnemonico:	Tipo di dati:	Descrizione:
.DVPA	BOOL	deviazione con allarme alto (0=no; 1=si)
.PVLA	BOOL	PV con allarme basso (0=no; 1=si)
.PVHA	BOOL	PV con allarme alto (0=no; 1=si)

**Descrizione:** L'istruzione PID controlla una variabile di processo (PV) come ad esempio flusso, pressione, temperatura o livello. In genere, l'istruzione PID riceve la variabile di processo (PV) da un modulo di ingresso analogico e modula un'uscita di variabile di controllo (CV) su un modulo di uscita analogico al fine di mantenere la variabile di processo sul valore di riferimento desiderato (setpoint).

Il bit .EN indica lo stato dell'esecuzione. Il bit .EN viene impostato quando la condizione del ramo di ingresso passa da falsa a vera. Quando la condizione del ramo di ingresso diventa falsa, il bit .EN viene azzerato. L'istruzione PID non utilizza un bit .DN e viene eseguita ogni scansione fino a quando la condizione del ramo di ingresso rimane vera.



41027

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** non influenzati

**Condizioni di errore:**

Si verifica un errore minore se:	Tipo errore:	Codice errore:
.UPD ≤ 0	4	35
setpoint fuori gamma	4	36

**Important:** nel processore PLC-5 questi sarebbero stati degli errori gravi.

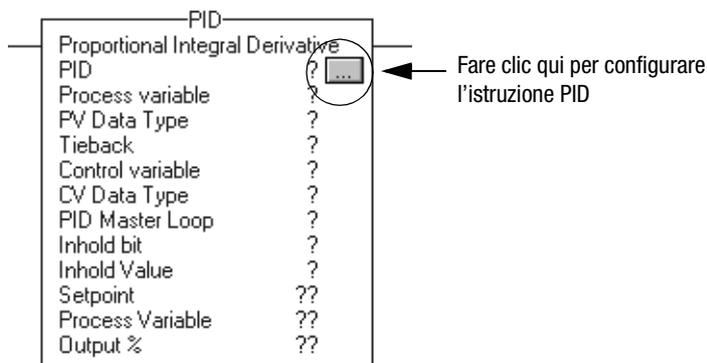
**Altri formati:**

**Formato:**            **Sintassi:**

testo neutro	<code>PID(pv,pv_type,tieback,cv,cv_type,master,inhold_bit,inhold_value);</code>
testo ASCII	<code>PID pv pv_type tieback cv cv_type master inhold_bit inhold_value</code>

**Configurazione di un'istruzione PID**

Dopo avere immesso l'istruzione PID e specificato la struttura PID, utilizzare le schede di configurazione per specificare in che modo l'istruzione deve funzionare.



**Calcolo parametri**

Selezionare la scheda Calcolo parametri. Le modifiche hanno effetto dal momento in cui si fa clic su un altro campo, su OK, su Applica o quando si preme Invio.

In questo campo:	Specificare:
Setpoint (SP)	Inserire un valore di setpoint (.SP).
Set output %	Inserire una percentuale di uscita (.SO). In modalità Manuale software, questo valore viene utilizzato per l'uscita. In modalità Auto, questo valore visualizza la % dell'uscita
Output bias	Inserire una percentuale del bias di uscita (.BIAS).
Proportional gain ( $K_p$ )	Inserire il guadagno proporzionale (.KP). Per i guadagni indipendenti, è il guadagno proporzionale (adimensionale). Per i guadagni dipendenti, è il guadagno del controllore (adimensionale).
Integral gain ( $K_i$ )	Inserire il guadagno integrale (.KI). Per i guadagni indipendenti, è il guadagno integrale (1/sec). Per i guadagni dipendenti, è il tempo di azzeramento (minuti per ripetizione).
Derivative time ( $K_d$ )	Inserire il guadagno derivativo (.KD). Per i guadagni indipendenti, è il guadagno derivativo (secondi). Per i guadagni dipendenti, è il tempo derivativo (minuti).
Manual mode	Selezionare Manuale (.MO) o Manuale software (.SWM). Se si selezionano entrambe le modalità, la modalità Manuale sarà quella prevalente.

## Configurazione

Selezionare la scheda Configurazione. Fare clic su OK o Applica affinché i cambiamenti abbiano effetto.

In questo campo:	Specificare:
PID equation	Selezionare guadagni indipendenti o guadagni dipendenti (.PE). Scegliere Indipendente se si desidera che i tre guadagni (P, I e D) operino indipendentemente. Scegliere Dipendente se si desidera un guadagno generale del controllore che influenzi tutti e tre i termini (P, I e D).
Control action	Selezionare E=PV-SP oppure E=SP-PV per l'azione di controllo (.CA).
Derivative of	Selezionare PV o errore (.DOE). Utilizzare la derivata della PV per eliminare i picchi transitori derivanti dalle modifiche del setpoint. Utilizzare la derivata dell'errore per una rapida risposta alle modifiche del setpoint se l'algoritmo è in grado di tollerare sovralongazioni.
Loop update time	Immettere il tempo di aggiornamento (.UPD) dell'istruzione (maggiore o uguale a 0,01 secondi).
CV high limit	Inserire un limite superiore per la variabile di controllo (.MAXO).
CV low limit	Inserire un limite inferiore per la variabile di controllo (.MINO).
Deadband value	Inserire un valore per la banda morta (.DB).
No derivative smoothing	Abilitare o disabilitare questa voce (.NDF).
No bias calculation	Abilitare o disabilitare questa voce (.NOBC).
No zero crossing in deadband	Abilitare o disabilitare questa voce (.NOZC).
PV tracking	Abilitare o disabilitare questa voce (.PVT).
Cascade loop	Abilitare o disabilitare questa voce (.CL).
Cascade type	Se l'anello in cascata è abilitato, selezionare slave o master (.CT).

## Avvisi

Selezionare la scheda Avvisi. Fare clic su OK o Applica affinché i cambiamenti abbiano effetto.

In questo campo:	Specificare:
PV high	Inserire un valore di allarme alto di PV (.PVH).
PV low	Inserire un valore di allarme basso di PV (.PVL).
PV deadband	Inserire un valore di banda morta dell'allarme di PV (.PVBD).
positive deviation	Inserire un valore di deviazione positiva (.DVP).
negative deviation	Inserire un valore di deviazione negativa (.DVN).
deviation deadband	Inserire un valore di banda morta dell'allarme della deviazione (.DVDB).

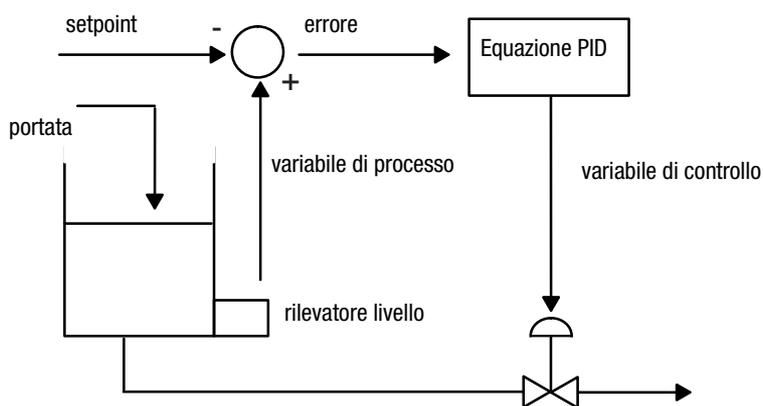
## Conversione in scala

Selezionare la scheda Conversione in scala. Fare clic su OK o Applica affinché i cambiamenti abbiano effetto.

In questo campo:	Specificare:
PV unscaled maximum	Inserire il valore massimo di PV (.MAXI) uguale al valore massimo non in scala ricevuto dal canale di ingresso analogico per il valore PV.
PV unscaled minimum	Inserire il valore minimo di PV (.MINI) uguale al valore minimo non in scala ricevuto dal canale di ingresso analogico per il valore PV.
PV engineering units maximum	Inserire il valore massimo di unità ingegneristiche corrispondente a .MAXI (.MAXS)
PV engineering units minimum	Inserire il valore minimo di unità ingegneristiche corrispondente a .MINI (.MINS)
CV maximum	Inserire il valore massimo di CV corrispondente al 100 % (.MAXCV).
CV minimum	Inserire il valore minimo di CV corrispondente allo 0 % (.MINCV).
Tieback maximum	Inserire il valore massimo di tieback (.MAXTIE) uguale al valore massimo non in scala ricevuto dal canale di ingresso analogico per il valore di tieback.
Tieback minimum	Inserire il valore minimo di tieback (.MINTIE) uguale al valore minimo non in scala ricevuto dal canale di ingresso analogico per il valore di tieback.
PID Initialized	Se si modificano le costanti di conversione in scala durante la modalità Run, disattivare questa opzione per inizializzare di nuovo i valori di riconversione interni (.INI).

## Uso delle istruzioni PID

Il controllo ad anello chiuso PID mantiene una variabile di processo al valore di riferimento (setpoint) desiderato. La seguente figura mostra un esempio relativo alla portata/livello di fluidi.



14271

In questo esempio, il livello del serbatoio viene confrontato con un valore di riferimento o setpoint. Se il livello è superiore al setpoint, l'equazione PID fa aumentare la variabile di controllo e provoca l'apertura della valvola di scarico del serbatoio facendo quindi diminuire il livello.

L'equazione PID utilizzata nell'istruzione PID è un'equazione posizionale con la possibilità di utilizzo di guadagni indipendenti o dipendenti. Quando si utilizzano i guadagni indipendenti, i guadagni proporzionali, integrali e derivativi influenzano solo i relativi termini specifici proporzionale, integrale o derivativo. Quando si utilizzano i guadagni dipendenti, il guadagno proporzionale è sostituito da un guadagno del controllore che influenza tutti e tre i termini. Per eseguire lo stesso tipo di controllo è possibile utilizzare indifferentemente una delle due forme di equazione. L'utente può scegliere l'equazione che preferisce.

<b>Opzioni di guadagno:</b>	<b>Derivata di:</b>	<b>Equazione:</b>
Guadagni dipendenti (standard ISA)	errore (E)	$CV = K_C \left[ E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dE}{dt} \right] + BIAS$
	variabile di processo (PV)	$E = SP - PV$ $CV = K_C \left[ E + \frac{1}{T_i} \int_0^t E dt - T_d \frac{dPV}{dt} \right] + BIAS$ $E = PV - SP$ $CV = K_C \left[ E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dPV}{dt} \right] + BIAS$
Guadagni indipendenti	errore (E)	$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} + BIAS$
	variabile di processo (PV)	$E = SP - PV$ $CV = K_p E + K_i \int_0^t E dt - K_d \frac{dPV}{dt} + BIAS$ $E = PV - SP$ $CV = K_p E + K_i \int_0^t E dt + K_d \frac{dPV}{dt} + BIAS$

Dove:

<b>Variabile:</b>	<b>Descrizione:</b>
K <sub>p</sub>	guadagno proporzionale (adimensionale) K <sub>p</sub> = K <sub>c</sub> adimensionale
K <sub>i</sub>	guadagno integrale (secondi <sup>-1</sup> ) La relazione tra K <sub>i</sub> (guadagno integrale) e T <sub>i</sub> (tempo integrale), è: $K_i = \frac{K_C}{60T_i}$
K <sub>d</sub>	guadagno derivativo (secondi) La relazione tra K <sub>d</sub> (guadagno derivativo) e T <sub>d</sub> (tempo derivativo), è: K <sub>d</sub> = K <sub>c</sub> (T <sub>d</sub> ) 60
K <sub>c</sub>	guadagno del controllore (adimensionale)
T <sub>i</sub>	tempo integrale (minuti/ripetizione)
T <sub>d</sub>	tempo derivativo (minuti)
SP	setpoint
PV	variabile di processo
E	errore [(SP-PV) o (PV-SP)]
BIAS	compensazione anticipata o bias
CV	variabile di controllo
dt	tempo di aggiornamento anello

Se non si desidera utilizzare un particolare termine dell'equazione PID, impostare i suoi guadagni a zero. Ad esempio, se non si desidera nessuna azione derivativa, impostare K<sub>d</sub> o T<sub>d</sub> a zero.

### **Antiaccumulo dell'integrale e trasferimento graduale da manuale ad automatico**

L'istruzione PID evita automaticamente l'accumulo dell'integrale evitando che questi accumuli ogniqualvolta che l'uscita della variabile di controllo raggiunge il suo valore massimo o minimo, in base all'impostazione di .MAXO e .MINO. L'integrale accumulato rimane bloccato fino a quando l'uscita della variabile di controllo non scende al di sotto del limite massimo o sale al di sopra del limite minimo. A questo punto la normale accumulazione dell'integrale riprende automaticamente.

L'istruzione PID supporta due modalità manuali di controllo:

<b>Modalità manuale di controllo:</b>	<b>Descrizione:</b>
Manuale software (.SWM)	conosciuta anche come modalità di impostazione uscita consente all'utente di impostare la % dell'uscita dal software Il valore di uscita impostato (.SO) viene utilizzato come uscita dell'anello. Questo valore in genere viene immesso da un operatore tramite un dispositivo di interfaccia operatore.
Manuale (.MO)	considera il valore di tieback come un ingresso e regola le sue variabili interne in modo da generare lo stesso valore all'uscita L'ingresso tieback per l'istruzione PID è in scala dallo 0 al 100 % con riferimento ai valori di .MINTIE e .MAXTIE ed è utilizzato come uscita dell'anello. L'ingresso tieback in genere deriva dall'uscita di una stazione hardware manuale/automatica che aggira l'uscita dal controllore. <b>Nota:</b> se si impostano entrambi i bit di modalità, prevale la modalità Manuale.

L'istruzione PID consente inoltre trasferimenti graduali automatici dalla modalità manuale software alla modalità Auto o dalla modalità manuale alla modalità Auto. L'istruzione PID calcola nuovamente il valore accumulato dell'integrale richiesto per fare in modo che l'uscita della variabile di controllo inseguo o il valore impostato dell'uscita (.SO) nella modalità manuale software o l'ingresso tieback nella modalità manuale. In questo modo, quando l'anello passa alla modalità Auto, l'uscita della variabile di controllo inizia dal valore di uscita impostato o di tieback senza che si verifichino "picchi" nel valore dell'uscita.

L'istruzione PID è inoltre in grado di fornire automaticamente un trasferimento graduale da Manuale ad Auto anche senza l'uso del controllo integrale (cioè,  $K_i = 0$ ). In questo caso l'istruzione modifica il termine – BIAS per far sì che l'uscite della variabile di controllo inseguo o i valori di uscita impostata o di tieback. Quando viene ripristinato il controllo automatico, il termine .BIAS mantiene il suo ultimo valore. È possibile disabilitare il calcolo del .BIAS impostando il bit .NOBC della struttura dati PID. Considerare che se si imposta il bit .NOBC su vero, l'istruzione PID non fornisce più un trasferimento graduale da Manuale ad Auto quando non si usa il controllo integrale.

### **Temporizzazione dell'istruzione PID**

L'istruzione PID ed il campionamento della variabile di processo devono essere aggiornati con frequenza periodica. Questo tempo di aggiornamento è in stretta relazione con il processo fisico che si controlla. Nel caso di anelli molto lenti, ad esempio anelli di temperatura, un tempo di aggiornamento di una volta al secondo o più, è in genere sufficiente ad ottenere un buon controllo. Anelli più veloci, come gli anelli di pressione o di flusso, possono richiedere un tempo di aggiornamento di una volta ogni 250 millisecondi.

Solo in casi molto rari, come nel caso del controllo della tensione di un svolgitoro di bobine, è necessario un aggiornamento dell'anello ogni 10 millisecondi o più veloce.

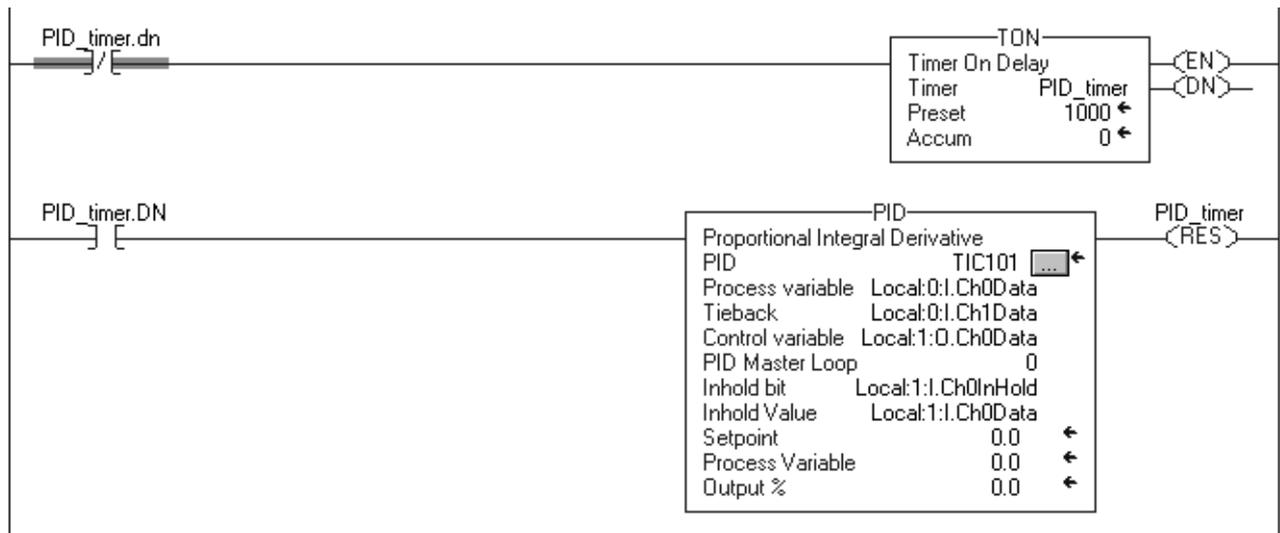
Poiché l'istruzione PID utilizza una base tempo per i suoi calcoli, è necessario sincronizzare l'esecuzione di questa istruzione con il campionamento della variabile di processo (PV).

Il modo più semplice per eseguire l'istruzione PID è quello di inserire l'istruzione PID in un task periodico. Configurare il tempo di aggiornamento dell'anello (.UPD) uguale alla frequenza del task periodico ed accertarsi che l'istruzione PID venga eseguita ad ogni scansione del task periodico. Ad esempio, utilizzare un ramo ladder incondizionato.



Quando si utilizza un task periodico, accertarsi che l'ingresso analogico utilizzato per la variabile di processo venga aggiornato nel processore con una frequenza significativamente più veloce di quella del task periodico. Idealmente, la variabile di processo dovrebbe essere inviata al processore con una velocità almeno 5 – 10 volte superiore a quella del task periodico. Ciò consente di ridurre la differenza di tempo tra i campioni effettivi della variabile di processo e l'esecuzione dell'anello PID. Per esempio, se l'anello PID si trova in un task periodico di 250 millisecondi, utilizzare un tempo di aggiornamento dell'anello di 250 millisecondi (.UPD = .25) e configurare il modulo di ingresso analogico in modo che produca dati almeno ogni 25 – 50 msec circa.

Un altro metodo, ma meno preciso, di eseguire un'istruzione PID è quello di inserire l'istruzione in un task continuo ed utilizzare il bit di fine di un timer per attivare l'esecuzione dell'istruzione PID.

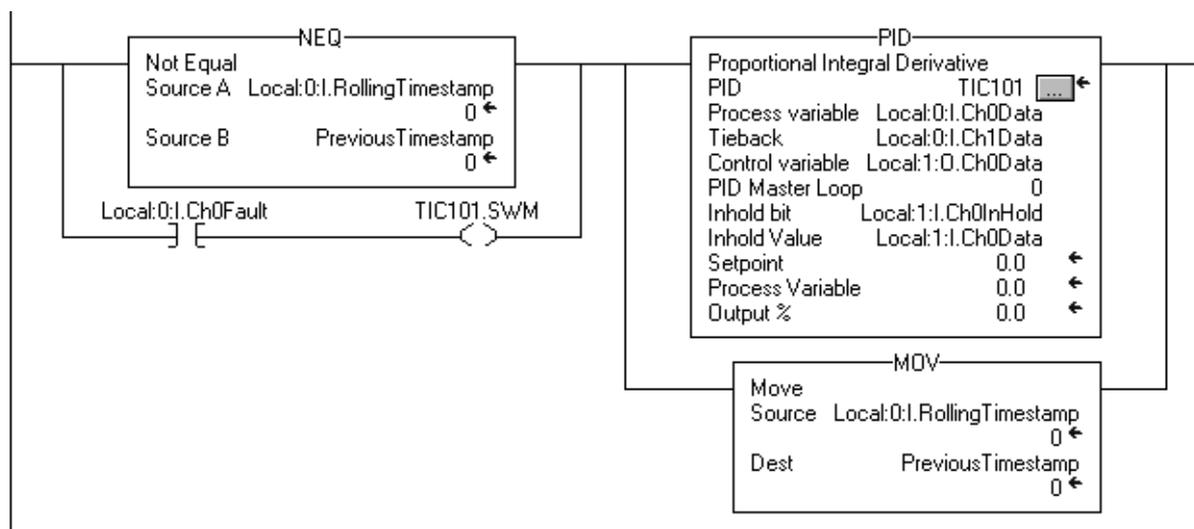


Con questo metodo, il tempo di aggiornamento dell'anello dell'istruzione PID deve essere posto uguale al valore preimpostato del timer. Come nel caso dell'utilizzo di un task periodico, è necessario impostare il modulo di ingresso analogico in modo che produca la variabile di processo ad una velocità molto più rapida del tempo di aggiornamento dell'anello. Il metodo mediante timer deve essere utilizzato solo per anelli con tempi di aggiornamento molto più lunghi rispetto al caso peggiore del tempo di esecuzione del task continuo.

Il modo più preciso per l'esecuzione di un'istruzione PID è quello di usare la funzione di campionamento in tempo reale (RTS) dei moduli di ingresso analogico 1756. Il modulo di ingresso analogico campiona i suoi ingressi ad una velocità di campionamento in tempo reale impostata dall'utente al momento della configurazione del modulo. Alla scadenza del periodo di campionamento in tempo reale, il modulo aggiorna i suoi ingressi ed registratore eronologico ciclico (rappresentato dal membro .RollingTimestamp della struttura dati dell'ingresso analogico) prodotto dal modulo. Il marcatore di tempo conta da 0 a 32.767 millisecondi. Controllate il registratore eronologico. Quando cambia, significa che è stato ricevuto un nuovo campione della variabile di processo. Ogni volta che il contatore cambia, eseguite l'istruzione PID una volta. Dato che il campione della variabile di processo dipende dal modulo di ingresso analogico, il tempo del campione di ingresso è molto preciso ed il tempo di aggiornamento dell'anello utilizzato dall'istruzione PID deve essere impostato uguale al tempo dell'RTS del modulo di ingresso analogico.

Per assicurarsi di non perdere alcun campione della variabile di processo, eseguire la logica ad una velocità maggiore del tempo dell'RTS. Per esempio, se il tempo RTS è di 250 msec., si potrebbe inserire la logica PID in un task periodico che viene eseguito ogni 100 msec. per fare in modo che non si perda alcun campione. È possibile, inoltre, inserire la logica PID in un task continuo, purché ci si assicuri che la logica venga aggiornata più frequentemente di una volta ogni 250 millisecondi.

Di seguito viene fornito un esempio del metodo di esecuzione mediante RTS. L'esecuzione dell'istruzione PID dipende dalla ricezione di nuovi dati di ingresso analogici. Se il modulo di ingresso analogico va in errore o viene rimosso, il controllore interrompe la ricezione del tempo dal marcatore e l'anello PID si interrompe. È necessario monitorare il bit di stato dell'ingresso analogico della variabile di processo e in caso di stato negativo, forzare l'anello nella modalità Manuale software ed eseguire l'anello ad ogni scansione. Ciò consente sempre all'operatore di cambiare manualmente l'uscita dell'anello PID.



### Riavviamento graduale

L'istruzione PID può interagire con i moduli di uscita analogici 1756 per supportare un riavviamento graduale quando il controllore passa dalla modalità Program alla modalità Run oppure quando il controllore viene acceso.

Se un modulo di uscita analogico 1756 perde la comunicazione con il controllore o rileva che il controllore è in modalità Programmazione, il modulo d'uscita analogico imposta le sue uscite alla condizione di errore specificata al momento della configurazione del modulo. Quindi, quando il controllore ritorna in modalità Run oppure ristabilisce la comunicazione con il modulo d'uscita analogico, utilizzando i parametri del Bit Inhold e del Valore Inhold sull'istruzione PID, è possibile far sì che l'istruzione PID riporti automaticamente la propria uscita della variabile di controllo uguale all'uscita analogica.

Per impostare un riavviamento graduale:

**Procedere in questo modo:****Dettagli:**

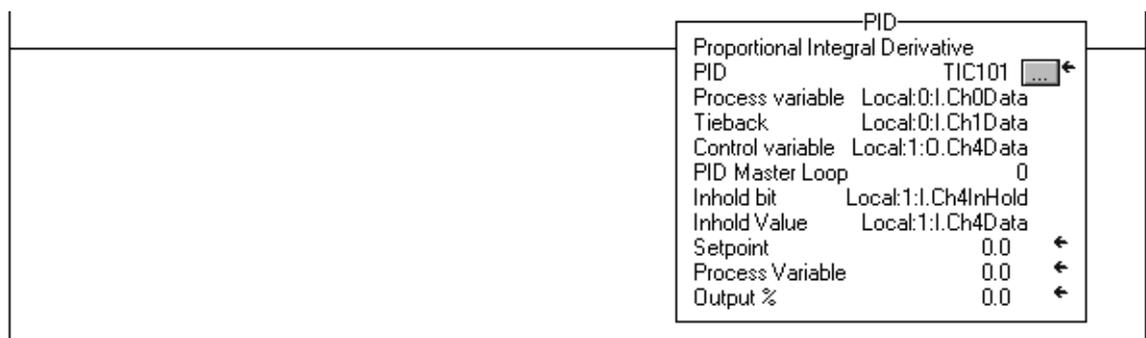
Configurare il canale del modulo di uscita analogico 1756 che riceve la variabile di controllo dall'istruzione PID

Selezionare la casella di controllo "hold for initialization" nella pagina Proprietà dello specifico canale del modulo. In tal modo si dice al modulo di uscita analogico che quando il controllore ritorna alla modalità Run, oppure quando ristabilisce la comunicazione con il modulo, esso deve mantenere l'uscita analogica sul suo valore corrente fino a quando il valore inviato dal controllore non corrisponde (con una differenza massima dello 0,1 %) con il valore corrente utilizzato dal canale di uscita. L'uscita del controllore passerà al valore di uscita correntemente mantenuto utilizzando il termine .BIAS. Questo passaggio è simile al trasferimento graduale ad Auto.

Inserire nell'istruzione PID la tag Inhold bit e la tag Inhold Value

Il modulo di uscita analogico 1756 ritorna due valori per ciascun canale nella sua struttura dati di ingresso. Il bit di stato InHold (.Ch2InHold, per esempio), se è vero, indica che il canale di uscita analogico sta mantenendo il suo valore. Il valore Data readback (.Ch2Data, per esempio) mostra il valore corrente dell'uscita in unità ingegneristiche. Inserire la tag del bit di stato InHold come parametro Inhold bit dell'istruzione PID. Inserire la tag del valore Data readback come parametro di Inhold Value. Se Inhold bit diventa vero, l'istruzione PID sposta Inhold Value nell'uscita della variabile di controllo e viene inizializzata di nuovo per supportare un riavviamento graduale con quel valore. Quando il modulo di uscita analogico riceve questo valore dal controllore, esso disabilita il bit di stato InHold, consentendo all'istruzione PID di iniziare il controllo normalmente.

La seguente istruzione PID utilizza il bit Inhold e Inhold value:

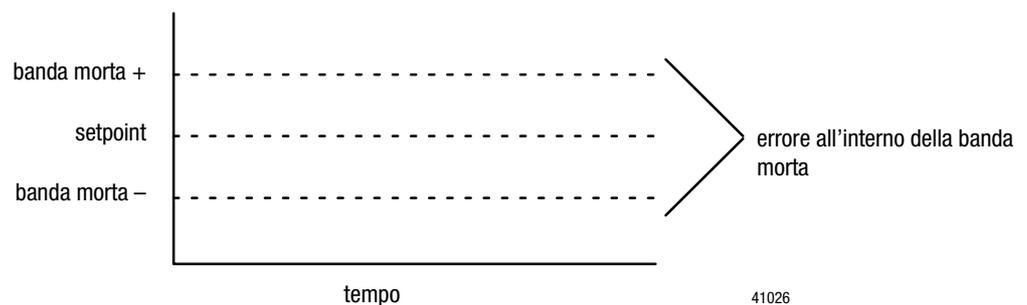


## Attenuazione derivativa

Il calcolo della derivata è migliorato da un filtro di attenuazione derivativa. Questo filtro digitale passa basso di primo livello contribuisce a ridurre i grandi picchi della derivata provocati dai disturbi nella variabile di processo. Questa attenuazione diventa più sensibile con valori del guadagno derivativo più grandi. Se il processo richiede valori di guadagno derivativo molto grandi, è possibile disabilitare la attenuazione della derivata ( $K_d > 10$ , per esempio). Per disabilitare l'attenuazione derivativa, selezionare l'opzione "No derivative smoothing" della scheda Configurazione oppure impostare il bit .NDF della struttura PID.

## Impostazione della banda morta

La banda morta regolabile permette di selezionare una gamma di errore, al di sopra e al di sotto del setpoint, in cui l'uscita non cambia fino a quando l'errore rimane entro i limiti della gamma stessa. Questa banda morta consente di controllare quanto la variabile di processo ed il setpoint possono discostarsi senza dover cambiare l'uscita. Inoltre, la banda morta contribuisce a ridurre l'usura ed il logorio del dispositivo di controllo finale.



L'attraversamento dello zero è il controllo della banda morta che permette all'istruzione di utilizzare l'errore per scopi di calcolo dal momento in cui la variabile di processo attraversa la banda morta fino all'attraversamento del setpoint. Una volta che la variabile di processo attraversa il setpoint (l'errore attraversa lo zero e cambia di segno) e fino a quando la variabile di processo rimane nella banda morta, l'uscita non cambia.

La banda morta si estende al di sopra e al di sotto del setpoint in base ad un valore specificato dall'utente. Per inibire la banda morta, inserire zero. La banda morta ha le stesse unità in scala del setpoint. È possibile utilizzare la banda morta senza l'attraversamento dello zero selezionando nella scheda Configurazione l'opzione "no zero crossing for deadband" oppure impostando il bit .NOZC nella struttura PID.

Se si utilizza la banda morta, la variabile di controllo deve essere REAL altrimenti essa verrà forzata a 0 se l'errore è all'interno della banda morta

## Uso del limite dell'uscita

È possibile impostare un limite di uscita (percentuale dell'uscita) sul controllo dell'uscita. Quando l'istruzione rileva che l'uscita ha raggiunto un limite, essa imposta un bit di allarme impedendo all'uscita di superare il limite inferiore o superiore.

## Compensazione anticipata o bias dell'uscita

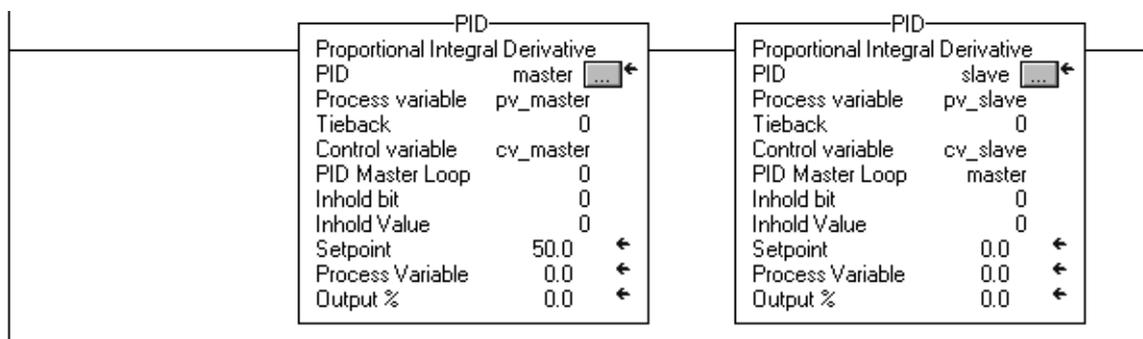
È possibile anticipare un disturbo dal sistema inserendo il valore .BIAS nel valore di compensazione anticipata/bias dell'istruzione PID.

Il valore della compensazione anticipata rappresenta un disturbo immesso nell'istruzione PID prima che il disturbo possa cambiare la variabile di processo. La compensazione anticipata viene spesso utilizzata per controllare i processi con ritardo di propagazione. Per esempio, un valore della compensazione anticipata che rappresenta "l'acqua fredda versata in una miscela calda" potrebbe fare aumentare il valore dell'uscita molto più velocemente piuttosto che aspettare che la variabile di processo cambi in conseguenza della nuova condizione creatasi.

Un valore di bias in genere viene utilizzato quando non si usa l'integrale del controllo. In questo caso, il valore bias può essere regolato in modo da mantenere l'uscita entro la gamma richiesta e tenere la variabile di processo vicino al setpoint.

## Anelli in cascata

L'istruzione PID pone in cascata due anelli assegnando l'uscita in percentuale dell'anello master al setpoint dell'anello slave. L'anello slave converte automaticamente l'uscita dell'anello master nelle unità ingegneristiche corrette per il setpoint dell'anello slave, in base ai valori .MAXS e .MINS dell'anello slave.



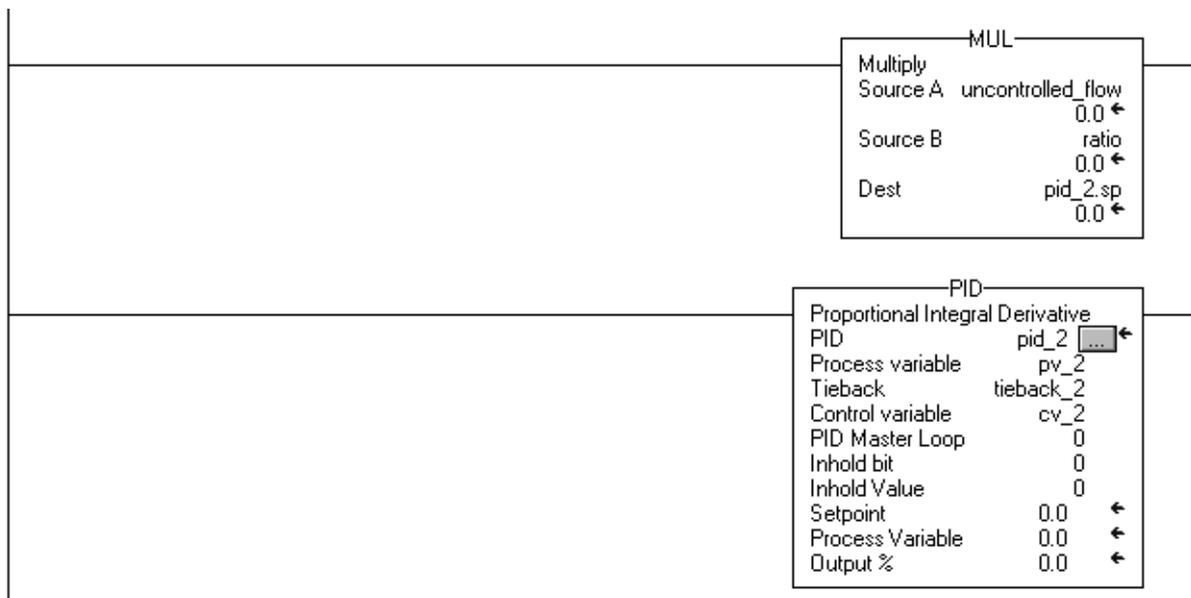
### Controllo di un rapporto

È possibile stabilire un rapporto tra due valori utilizzando un'istruzione MUL con questi parametri:

- valore non controllato
- valore controllato (il risultante setpoint che deve essere utilizzato dall'istruzione PID)
- rapporto tra questi due valori

Nell'istruzione MUL, inserire:

Per questo parametro MUL:	Inserire questo valore:
destination	valore controllato
source A	valore non controllato
source B	rapporto



## Istruzioni trigonometriche (SIN, COS, TAN, ASN, ACS, ATN)

### Introduzione

Le istruzioni trigonometriche eseguono il calcolo delle funzioni trigonometriche.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
Calcolare il seno di un valore.	SIN	13-2
Calcolare il coseno di un valore.	COS	13-4
Calcolare la tangente di un valore.	TAN	13-6
Calcolare l'arcoseno di un valore.	ASN	13-8
Calcolare l'arcocoseno di un valore.	ACS	13-10
Calcolare l'arcotangente di un valore.	ATN	13-12

È possibile usare tipi di dati diversi, ma si potrebbe perdere in precisione ed in arrotondamento ed inoltre l'istruzione potrebbe impiegare più tempo per essere eseguita. Controllare il bit S:V per verificare se il risultato è stato troncato.

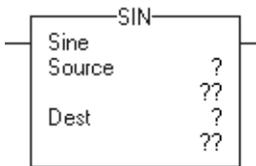
I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente ed utilizzando meno memoria se tutti gli operandi di un'istruzione utilizzano il medesimo tipo di dati ottimale, generalmente DINT o REAL.

Un'istruzione trigonometrica viene eseguita ogni volta che l'istruzione viene scandita e per tutto il tempo che la condizione del ramo di entrata rimane vera. Se si desidera che l'istruzione venga calcolata solo una volta, utilizzare un'istruzione ONS per attivare l'istruzione trigonometrica.

## Seno (SIN)

L'istruzione SIN è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola il seno di questo valore
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione SIN calcola il seno del valore Source (in radianti) e memorizza il risultato in Destination.

Il valore Source deve essere maggiore o uguale a  $-205887,4$  ed inferiore o uguale a  $205887,4$ . Il valore risultante di Destination è sempre maggiore o uguale a  $-1$  e minore o uguale a  $1$ .

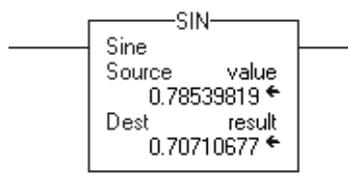
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola il seno di Source ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di SIN:



Se è abilitata, l'istruzione SIN calcola il seno di *value* ed inserisce il risultato in *result*.

**Altri formati:**

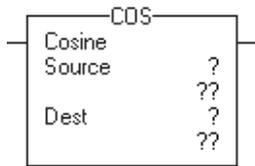
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>SIN(source,destination);</i>
testo ASCII	<i>SIN source destination</i>

**Istruzioni correlate:** CMP, CPT, COS, TAN, ASN, ACS, ATN, DEG, RAD

## Coseno (COS)

L'istruzione COS è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola il coseno di questo valore
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione COS calcola il coseno del valore Source (in radianti) e memorizza il risultato in Destination.

Il valore Source deve essere maggiore o uguale a  $-205887,4$  ed inferiore o uguale a  $205887,4$ . Il valore risultante di Destination è sempre maggiore o uguale a  $-1$  e minore o uguale a  $1$ .

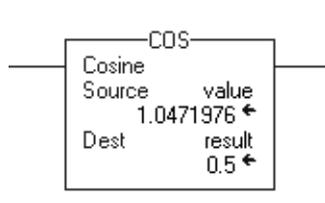
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola il coseno di Source ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di COS:



Se è abilitata, l'istruzione COS calcola il coseno di *value* ed inserisce il risultato in *result*.

**Altri formati:**

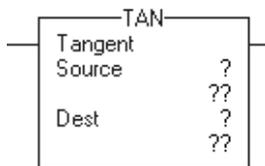
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>COS(source, destination);</i>
testo ASCII	<i>COS source destination</i>

**Istruzioni correlate:** CMP, CPT, SIN, TAN, ASN, ACS, ATN, DEG, RAD

## Tangente (TAN)

L'istruzione TAN è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola la tangente di questo valore
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione TAN calcola la tangente del valore Source (in radianti) e memorizza il risultato in Destination.

Il valore Source deve essere maggiore o uguale a  $-102943,7$  e minore o uguale a  $102943,7$ .

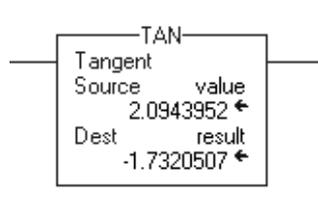
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola la tangente di Source e memorizza il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di TAN:



Se è abilitata, l'istruzione TAN calcola la tangente di *value* ed inserisce il risultato in *result*.

**Altri formati:**

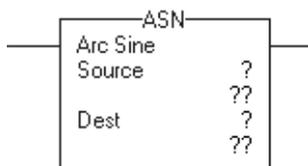
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	$TAN(source, destination);$
testo ASCII	$TAN source destination$

**Istruzioni correlate:** CMP, CPT, COS, SIN, ASN, ACS, ATN, DEG, RAD

## Arcoseno (ASN)

L'istruzione ASN è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola l'arcoseno di questo valore
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione ASN calcola l'arcoseno del valore Source e memorizza il risultato in Destination (in radianti).

Il valore Source deve essere maggiore o uguale a  $-1$  ed inferiore o uguale a  $1$ . Il valore risultante di Destination è sempre maggiore o uguale a  $-\pi/2$  e minore o uguale a  $\pi/2$  (dove  $\pi = 3.141593$ ).

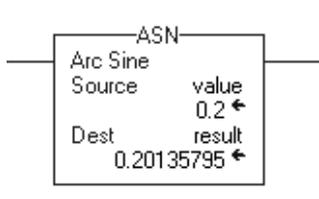
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola l'arcoseno di Source ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di ASN:



Se è abilitata, l'istruzione ASN calcola l'arcoseno di *value* ed inserisce il risultato in *result*.

**Altri formati:**

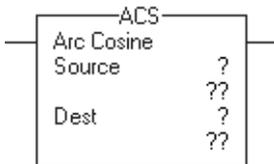
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>ASN(source,destination);</i>
testo ASCII	<i>ASN source destination</i>

**Istruzioni correlate:** CMP, CPT, ACS, ATN, SIN, COS, TAN, DEG, RAD

## Arcocoseno (ACS)

L'istruzione ACS è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola l'arcocoseno di questo valore
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione ACS calcola l'arcocoseno del valore Source e memorizza il risultato in Destination (in radianti).

Il valore Source deve essere maggiore o uguale a -1 ed inferiore o uguale a 1. Il valore risultante di Destination è sempre maggiore o uguale a 0 e minore o uguale a  $\pi$  (dove  $\pi = 3.141593$ ).

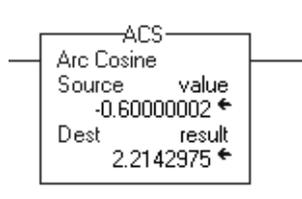
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola l'arcocoseno di Source ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di ACS:



Se è abilitata, l'istruzione ACS calcola l'arcocoseno di *value* ed inserisce il risultato in *result*.

**Altri formati:**

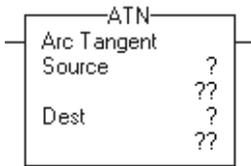
<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>ACS(source,destination);</i>
testo ASCII	<i>ACS source destination</i>

**Istruzioni correlate:** CMP, CPT, ASN, ATN, SIN, COS, TAN, DEG, RAD

## Arcotangente (ATN)

L'istruzione ATN è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola l'arcotangente di questo valore
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione ATN calcola l'arcotangente del valore Source e memorizza il risultato in Destination (in radianti).

Il valore risultante di Destination è sempre maggiore o uguale a  $-\pi/2$  ed inferiore o uguale a  $\pi/2$  (dove  $\pi = 3.141593$ ).

### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola l'arcotangente di Source ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di ATN:



Se è abilitata, l'istruzione ATN calcola l'arcotangente di *value* ed inserisce il risultato in *result*.

**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	<i>ATN(source,destination);</i>
testo ASCII	<i>ATN source destination</i>

**Istruzioni correlate:** CMP, CPT, ACS, ASN, SIN, COS, TAN, DEG, RAD

**Nota:**

## Istruzioni di matematica avanzata (LN, LOG, XPY)

### Introduzione

Le istruzioni di matematica avanzata comprendono queste istruzioni.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
Calcolare il logaritmo naturale di un valore.	LN	14-2
Calcolare il logaritmo in base 10 di un valore.	LOG	14-4
Elevare un valore alla potenza di un altro valore.	XPY	14-6

È possibile usare tipi di dati diversi, ma si potrebbe perdere in precisione ed arrotondamento ed inoltre l'istruzione potrebbe impiegare più tempo per essere eseguita. Controllare il bit S:V per verificare se il risultato è stato troncato.

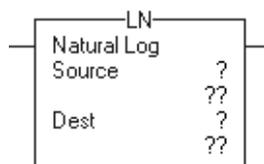
I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente e richiede meno memoria se tutti gli operandi dell'istruzione utilizzano il medesimo tipo di dati ottimali, in genere DINT o REAL.

Un'istruzione di matematica avanzata viene eseguita ogni volta che l'istruzione viene scandita e per tutto il tempo che la condizione del ramo di entrata rimane vera. Se si desidera che l'istruzione venga calcolata solo una volta, utilizzare un'istruzione ONS per attivare l'istruzione matematica.

## Logaritmo naturale (LN)

L'istruzione LN è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola il logaritmo naturale di questo valore.
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione LN calcola il logaritmo naturale di Source e memorizza il risultato in Destination.

Il valore Source deve essere maggiore di zero, altrimenti viene impostato S:V. Il valore Destination è maggiore o uguale a -87,33655 e minore o uguale a 88,72284.

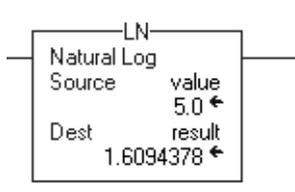
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola il logaritmo naturale di Source e memorizza il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di LN:



Se è abilitata, l'istruzione LN calcola il logaritmo naturale del *value* ed inserisce il risultato in *result*.

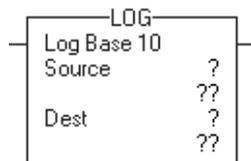
**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	LN( <i>source,destination</i> );
testo ASCII	LN <i>source destination</i>

**Istruzioni correlate:** CMP, CPT, LOG, XPY

**Logaritmo in base 10 (LOG)**

L'istruzione LOG è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT <b>REAL</b>	immediato tag	calcola il logaritmo di questo valore.
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione LOG calcola il logaritmo in base 10 di Source e memorizza il risultato in Destination.

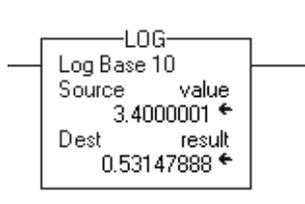
Il valore Source deve essere maggiore di zero, altrimenti viene impostato S:V. La risultante Destination è maggiore o uguale a  $-37,92978$  e minore o uguale a  $38,53184$ .

**Esecuzione:**

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore calcola il logaritmo di Source e memorizza il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

**Esempio di LOG:**

Se è abilitata, l'istruzione LOG calcola il logaritmo di *value* ed inserisce il risultato in *result*.

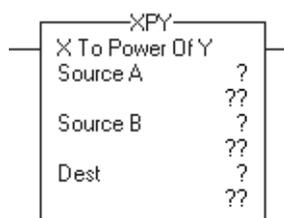
**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	LOG( <i>source,destination</i> );
testo ASCII	LOG <i>source destination</i>

**Istruzioni correlate:** CMP, CPT, LN, XPY

**X alla potenza di Y (XPY)**

L'istruzione XPY è un'istruzione di uscita.

**Operandi:**

Operando:	Tipo:	Formato:	Descrizione:
Source A	SINT INT DINT <b>REAL</b>	immediato tag	valore base
Source B	SINT INT DINT <b>REAL</b>	immediato tag	esponente
Destination	SINT INT DINT <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione XPY eleva Source A (X) alla potenza di Source B (Y) e memorizza il risultato in Destination. Se Source A è negativa, Source B deve essere un valore intero, altrimenti si verifica un errore minore.

L'istruzione XPY utilizza questo algoritmo:  $Destination = X^{**}Y$

Il controllore calcola  $x^0=1$  e  $0^x=0$ .

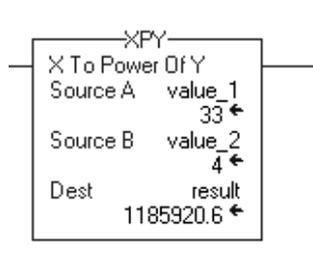
**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore eleva Source A alla potenza di Source B ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:**

Si verifica un errore minore se:	Tipo errore:	Codice errore:
Source A è negativa e Source B non è un valore intero	4	4

**Esempio di XPY:**

Se è abilitata, l'istruzione XPY eleva *value\_1* alla potenza di *value\_2* ed inserisce il risultato in *result*.

**Altri formati:**

Formato:	Sintassi:
testo neutro	<code>XPY(<i>source_A</i>,<i>source_B</i>,<i>destination</i>);</code>
testo ASCII	<code>XPY <i>source_A</i> <i>source_B</i> <i>destination</i></code>

**Istruzioni correlate:** CMP, CPT, LN, LOG

**Nota:**

## Istruzioni di conversione matematica (DEG, RAD, TOD, FRD, TRN)

### Introduzione

Le istruzioni di conversione matematica convertono valori.

Se si desidera:	Utilizzare questa istruzione:	Vedere pagina:
Convertire radianti in gradi.	DEG	15-2
Convertire gradi in radianti.	RAD	15-3
Convertire un valore intero in un valore BCD.	TOD	15-4
Convertire un BCD in un valore intero.	FRD	15-6
Eliminare la parte frazionaria di un valore	TRN	15-8

È possibile usare tipi di dati diversi, ma si potrebbe perdere in precisione ed in arrotondamento ed inoltre l'istruzione potrebbe impiegare più tempo per essere eseguita. Controllare il bit S:V per verificare se il risultato è stato troncato.

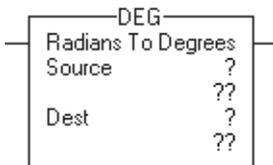
I tipi di dati in **neretto** indicano i tipi di dati ottimali. Un'istruzione viene eseguita più velocemente e richiede meno memoria se tutti gli operandi dell'istruzione utilizzano il medesimo tipo di dati ottimali, in genere DINT o REAL.

Un'istruzione di conversione matematica viene eseguita ogni volta che l'istruzione viene scandita e per tutto il tempo che la condizione del ramo di entrata rimane vera. Se si desidera che l'istruzione venga calcolata solo una volta, utilizzare un'istruzione ONS per attivare l'istruzione di conversione.

## Gradi (DEG)

L'istruzione DEG è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT REAL	immediato tag	valore da convertire in gradi
Destination	SINT INT DINT REAL	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione DEG converte il valore Source (in radianti) in gradi e memorizza il risultato in Destination.

L'istruzione RAD utilizza questo algoritmo:  
 $Source * 180 / \pi$  (dove  $\pi = 3,141593$ )

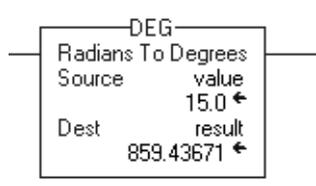
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore converte Source in gradi ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di DEG:



Se è abilitata, l'istruzione DEG converte *value* in gradi ed inserisce il risultato in *result*.

### Altri formati:

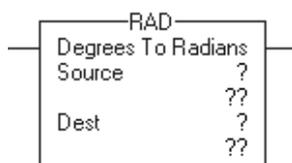
Formato:	Sintassi:
testo neutro	<code>DEG(<i>source</i>,<i>destination</i>);</code>
testo ASCII	<code>DEG <i>source destination</i></code>

**Istruzioni correlate:** CMP, CPT, RAD, SIN, COS, TAN, ASN, ACS, ATN

## Radiani (RAD)

L'istruzione RAD è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT DINT REAL	immediato tag	valore da convertire in radianti
Destination	SINT INT DINT REAL	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione RAD converte il valore Source (in gradi) in radianti e memorizza il risultato in Destination.

L'istruzione RAD utilizza questo algoritmo:  
Source\* $\pi$ /180 (dove  $\pi = 3,141593$ )

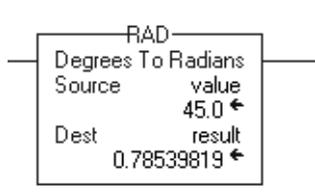
### Esecuzione:

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore converte Source in radianti ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di RAD:



Se è abilitata, l'istruzione RAD converte *value* in radianti ed inserisce il risultato in *result*.

### Altri formati:

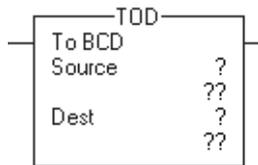
Formato:	Sintassi:
testo neutro	RAD( <i>source</i> , <i>destination</i> );
testo ASCII	RAD <i>source destination</i>

**Istruzioni correlate:** CMP, CPT, DEG, SIN, COS, TAN, ASN, ACS, ATN

### Conversione in BCD (TOD)

L'istruzione TOD è un'istruzione di uscita.

**Operandi:**



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b>	immediato tag	valore da convertire in BCD $0 \leq \text{Source} \leq 99,999,999$
Destination	SINT INT <b>DINT</b>	tag	tag dove memorizzare il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione TOD converte un valore intero ( $0 \leq \text{Source} \leq 99,999,999$ ) in un valore BCD e memorizza il risultato in Destination.

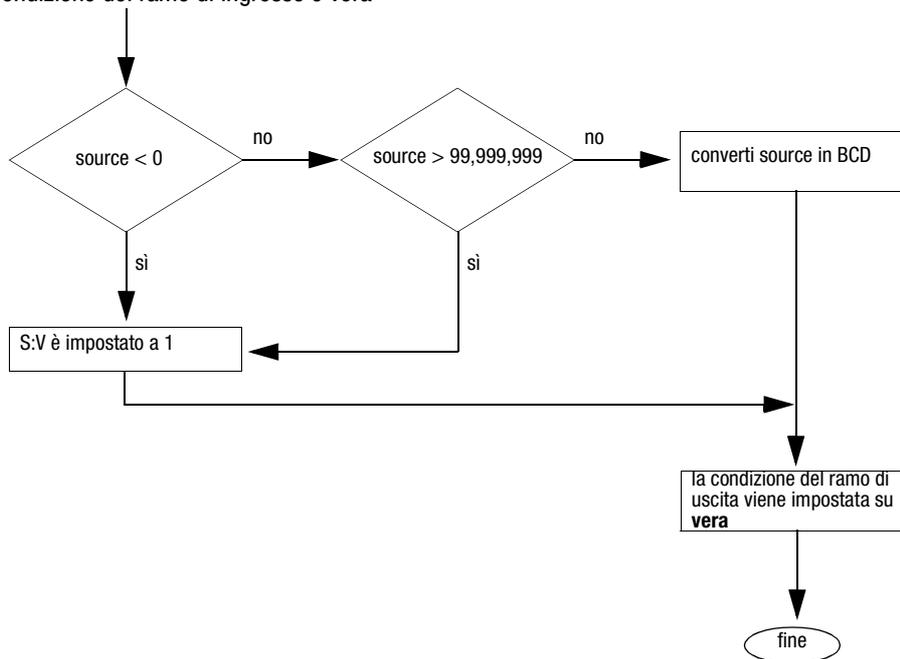
La sigla BCD è l'acronimo di Binary Coded Decimal (decimale codificato in binario), il sistema che esprime le singole cifre decimali (0-9) sotto forma di notazione binaria a 4 bit.

Se si inserisce un valore Source negativo, l'istruzione genera un errore minore ed azzerla Destination.

**Esecuzione:**

Condizione:	Azione:
prescansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.

la condizione del ramo di ingresso è vera



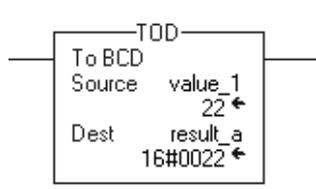
Condizione:	Azione:
la condizione del ramo di ingresso è vera	Il controllore converte Source in BCD ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:**

Si verifica un errore minore se:	Tipo errore:	Codice errore:
source < 0	4	4

**Esempio di TOD:**



Se è abilitata, l'istruzione TOD converte *value\_1* in un valore BCD ed inserisce il risultato in *result\_a*.

**Altri formati:**

Formato:	Sintassi:
testo neutro	TOD( <i>source</i> , <i>destination</i> );
testo ASCII	TOD <i>source destination</i>

**Istruzioni correlate:** CMP, CPT, FRD

## Conversione in interi (FRD)

L'istruzione FRD è un'istruzione di uscita.

### Operandi:



Operando:	Tipo:	Formato:	Descrizione:
Source	SINT INT <b>DINT</b>	immediato tag	valore da convertire in decimali
Destination	SINT INT <b>DINT</b>	tag	tag dove memorizzare il risultato

Una tag SINT o INT viene convertita in un valore DINT mediante riempimento di zeri. Fare riferimento a "Conversione di dati" a pag. A-6.

**Descrizione:** L'istruzione FRD converte un valore BCD (Source) in un valore decimale e memorizza il risultato in Destination.

### Esecuzione:

Condizione:	Azione:
prescaansione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore converte il valore Source in decimali ed inserisce il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di FRD:



Se è abilitata, l'istruzione FRD converte il *value\_a* in un valore decimale ed inserisce il risultato in *result\_1*.

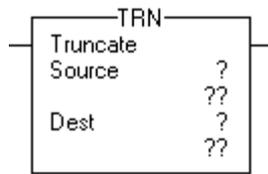
**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	FRD( <i>source,destination</i> );
testo ASCII	FRD <i>source destination</i>

**Istruzioni correlate:** CMP, CPT, TOD

## Tronca (TRN)

L'istruzione TRN è un'istruzione di uscita.



### Operandi:

Operando:	Tipo:	Formato:	Descrizione:
Source	<b>REAL</b>	immediato tag	valore da troncare
Destination	<b>SINT</b> <b>INT</b> <b>DINT</b> <b>REAL</b>	tag	tag dove memorizzare il risultato

**Descrizione:** L'istruzione TRN elimina (tronca) la parte frazionaria di un valore (Source) e memorizza il risultato in Destination. Il troncamento non arrotonda il valore in quanto la parte non frazionaria rimane la stessa a prescindere dal valore della parte frazionaria.

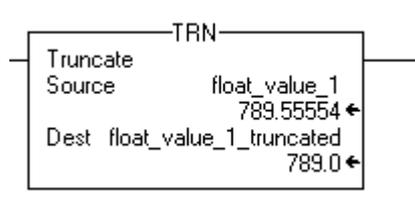
### Esecuzione:

Condizione:	Azione:
prescazione	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è falsa	La condizione del ramo di uscita viene impostata su falso.
la condizione del ramo di ingresso è vera	Il controllore elimina la parte frazionaria di Source e memorizza il risultato in Destination. La condizione del ramo di uscita viene impostata su vero.

**Indicatori di stato aritmetico:** gli indicatori di stato aritmetici sono influenzati.

**Condizioni di errore:** nessuna

### Esempio di TRN:



Se è abilitata, l'istruzione TRN elimina la parte frazionaria di *float\_value\_1*, lasciando immutata la parte non frazionaria, ed inserisce il risultato in *float\_value\_1\_truncated*.

**Altri formati:**

<b>Formato:</b>	<b>Sintassi:</b>
testo neutro	TRN( <i>source,destination</i> );
testo ASCII	TRN <i>source destination</i>

**Istruzioni correlate:** ABS, CMP, CPT, FRD, TOD

**Nota:**

## Attributi comuni

### Introduzione

Questa appendice descrive gli attributi comuni delle istruzioni del Logix5550.

Per informazioni su:	Vedere pagina:
Parole chiave di stato aritmetico	A-1
Altre parole chiave	A-4
Tipi di dati	A-5
Valori immediati	A-6
Conversione di dati	A-6

### Parole chiave di stato aritmetico

Utilizzare le parole chiave di stato aritmetico per controllare lo stato degli indicatori di stato aritmetico.

Parola chiave:	Indicatore di stato:	Descrizione:
S:V	overflow	Un overflow viene impostato se non è possibile memorizzare il valore nella destinazione. Il valore è più grande del valore massimo consentito per la destinazione oppure il valore è inferiore al valore minimo consentito per la destinazione.  <b>Importante:</b> ogni volta che viene impostato S:V, viene generato un errore minore (tipo 4, codice 4)
S:Z	zero	Viene impostato a zero se il valore di destinazione dell'istruzione è 0.
S:N	segno (il risultato è negativo)	Il segno viene impostato se il valore di destinazione dell'istruzione è negativo. Vedere gli esempi seguenti.
S:C	riporto	Il riporto viene impostato nel caso in cui un'operazione aritmetica provochi un "riporto" o un "prestito" che tenta di utilizzare bit che non fanno parte del tipo di dati della destinazione. Per esempio: la somma $3 + 9$ provoca un riporto di 1 la sottrazione $25 - 18$ richiede un prestito di 10 Vedere gli esempi seguenti.

Le parole chiave di stato aritmetico possono essere indifferentemente maiuscole o minuscole.

Data la rapidità di cambiamento degli indicatori di stato aritmetici, il software non riesce ad attivare la relativa tag. Per tale motivo non è possibile determinare visualmente lo stato di questi indicatori.

Gli schemi seguenti mostrano dove i tipi di dati interi memorizzano gli indicatori di stato S:N e S:C.

**Se il tipo di dati è SINT**

valore massimo = +127  
 valore minimo = -128

S:C	7	6	5	4	3	2	1	0
	S:N							

**Se il tipo di dati è INT**

valore massimo = +32.767  
 valore minimo = -32.768

S:C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	S:N															

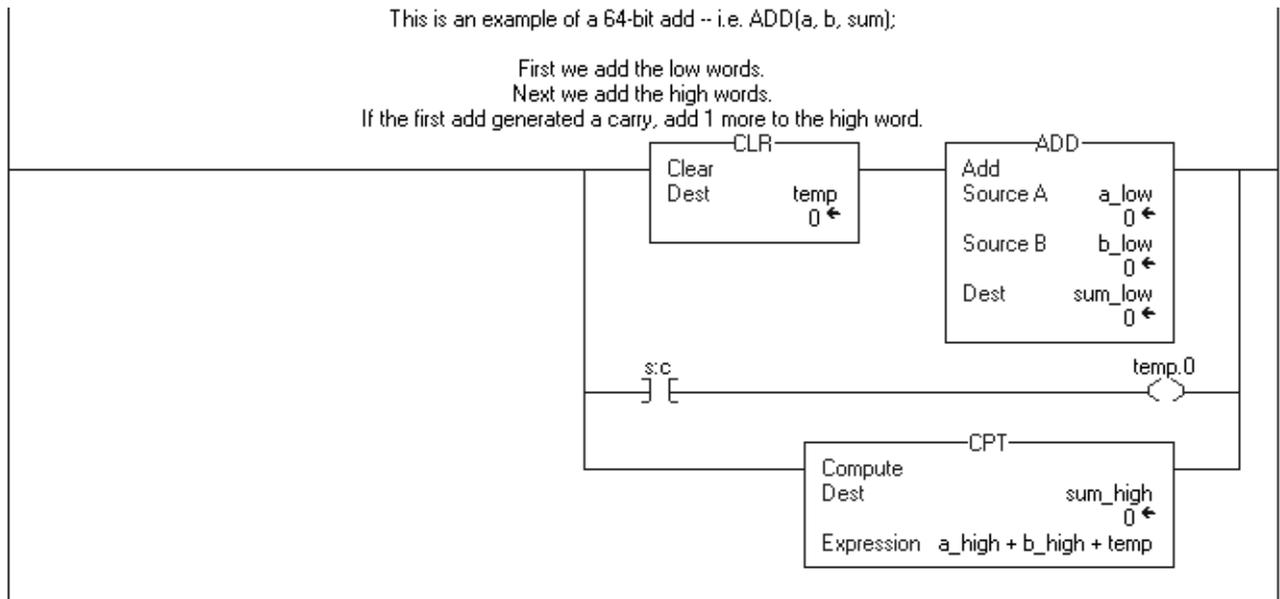
**Se il tipo di dati è DINT**

valore massimo = +2.147.483.647  
 valore minimo = -2.147.483.648

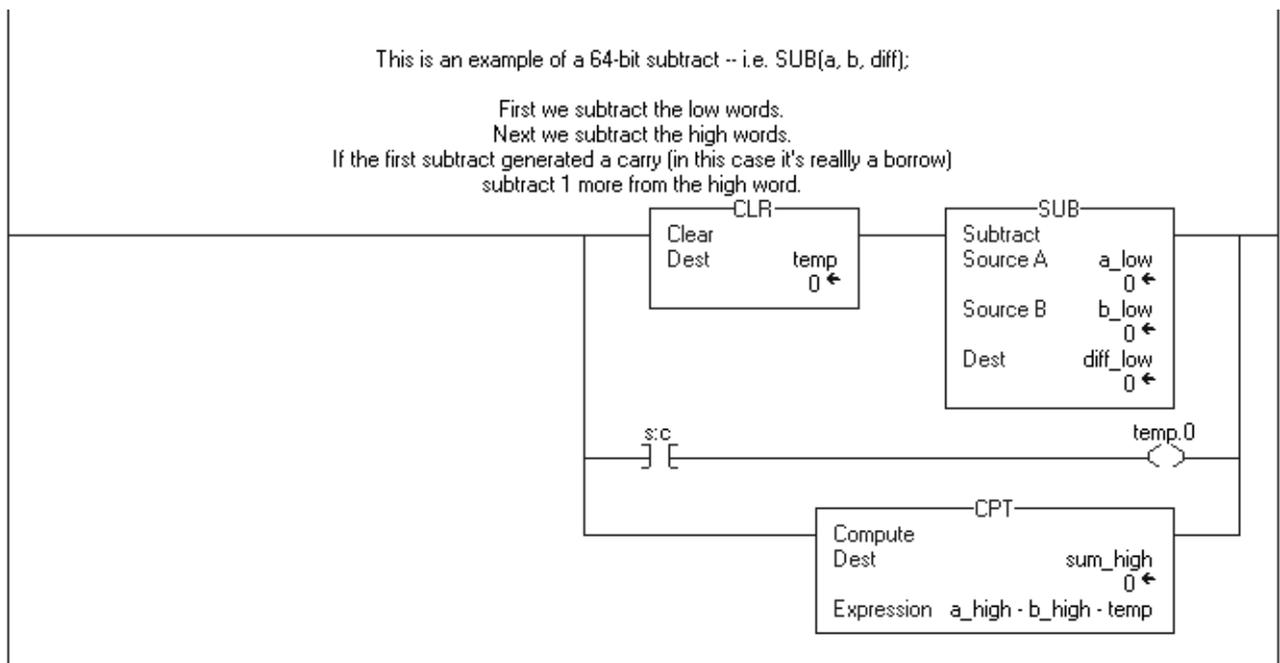
S:C	31	30	29	28	27	26	25	24	•	•	•	•	•	7	6	5	4	3	2	1	0	
	S:N																					

Il seguente esempio di programmazione mostra come utilizzare il bit di riporto.

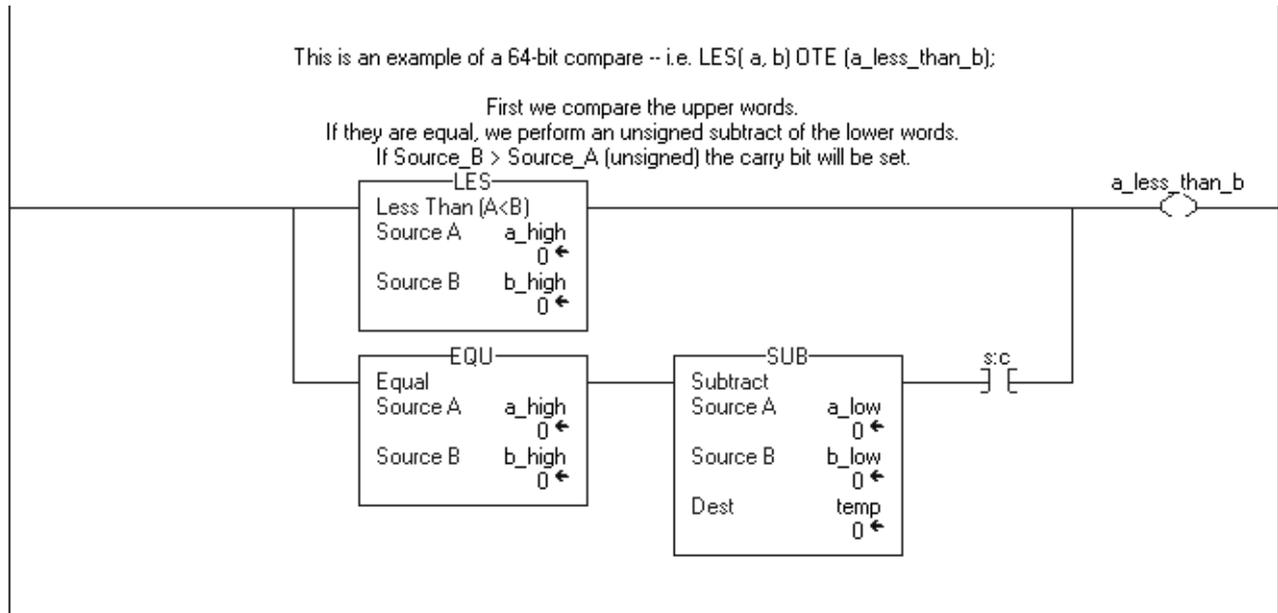
**esempio 1**



**esempio 2**



**esempio 3**



**Altre parole chiave**

Oltre alle parole chiave di stato aritmetico, il controllore supporta queste parole chiave.

Parola chiave:	Accesso:	Descrizione:
S:FS	lettura	Il bit di prima scansione viene impostato se questa è la prima scansione normale delle routine nel programma corrente.
S:MINOR	lettura scrittura	Il bit di errore minore viene impostato se è stato generato almeno un errore minore. Il controllore imposta questo bit quando si verifica un errore minore dovuto all'esecuzione del programma. Il controllore non imposta questo bit per gli errori minori che non sono relativi all'esecuzione del programma, ad esempio per batteria scarica.
THIS	na	Il valore THIS è valido solamente con le istruzioni GSV e SSV che fanno riferimento a un TASK, ad un PROGRAM o ad una ROUTINE. Utilizzare THIS per specificare il TASK, il PROGRAM o la ROUTINE corrente.

Le parole chiave possono essere scritte indifferentemente in maiuscolo o in minuscolo.

A causa della rapida velocità di cambiamento di questi indicatori, queste parole chiave non vengono riportate nel software di programmazione nel loro stato effettivo.

**Non è possibile** definire tag alias per una parola chiave.

**Tipi di dati**

I tipi di dati del controllore sono conformi ai tipi di dati definiti dalle IEC 1131-3. I tipi di dati di base predefiniti sono:

<b>Tipo di dati:</b>	<b>Descrizione:</b>	<b>Gamma</b>
BOOL	booleano di 1 bit	0 = azzerato 1 = impostato
SINT	intero di 1 byte	da -128 a 127
INT	intero di 2 byte	da -32.768 a 32.767
DINT	intero di 4 byte	da -2.147.483.648 a 2.147.483.647
REAL	numero a virgola mobile di 4 byte	da $-3.402823E^{38}$ a $-1.1754944E^{-38}$ (valori negativi) e 0 e da $1.1754944E^{-38}$ a $3.402823E^{38}$ (valori positivi)

I dati REAL memorizzano anche  $\pm$ infinito e  $\pm$ NAN, ma la visualizzazione mediante software differisce a seconda del formato di visualizzazione.

<b>Formato di visualizzazione:</b>	<b>Equivalente:</b>
Reale	+infinito 1.\$ -infinito -1.\$ +NAN 1.#QNAN -NAN -1.#QNAN
Esponenziale	+infinito 1.#INF000e+000 -infinito -1.#INF000e+000 +NAN 1.#QNAN00e+000 -NAN -1.#QNAN00e+000

Le strutture predefinite sono:

<b>Tipo di dati:</b>	<b>Descrizione:</b>
AXIS <sup>(1)</sup>	struttura di controllo di un asse
CONTROL	struttura di controllo per istruzioni di matrice (file)
COUNTER	struttura di controllo per istruzioni di contatore
MESSAGE <sup>(1)</sup>	struttura di controllo per istruzione MSG
MOTION_GROUP <sup>(1)</sup>	struttura di controllo per un gruppo di movimento
MOTION_INSTRUCTION	struttura di controllo per istruzioni di movimento
PID	struttura di controllo per istruzione PID
TIMER	struttura di controllo per istruzioni di timer

<sup>(1)</sup> Queste strutture non supportano matrici, non possono essere annidate in strutture definite dall'utente e non possono essere passate ad altre routine mediante un'istruzione JSR.

## Valori immediati

Ogni volta che si inserisce un valore immediato (costante) in formato decimale (ad esempio -2, 3), il controllore memorizza il valore utilizzando 32 bit. Se si inserisce un valore in un formato diverso dal decimale, ad esempio binario o esadecimale, e non si specificano tutti i 32 bit, il controllore inserisce uno zero nei bit che non vengono specificati (riempimento con zeri).

### Esempio: Riempimento con zeri dei valori immediati

Se si inserisce:	Il controllore memorizza:
-1	16#ffff ffff (-1)
16#ffff (-1)	16#0000 ffff (65535)
8#1234 (668)	16#0000 029c (668)
2#1010 (10)	16#0000 000a (10)

## Conversione di dati

Le istruzioni del Logix5000 vengono eseguite più velocemente e richiedono meno memoria se tutti gli operandi dell'istruzione utilizzano:

- lo stesso tipo di dati
- un tipo di dati ottimale:
  - Nella sezione “Operandi” di ciascuna istruzione contenuta in questo manuale, il tipo di dati in **neretto** indica il tipo di dati ottimale.
  - I tipi di dati DINT e REAL generalmente rappresentano i tipi di dati ottimali.

Se si mischiano tipi di dati diversi e si utilizzano tag di tipo non ottimale, il controllore converte i dati in base a queste regole:

- uno *qualsiasi* degli operandi è un valore REAL?

Se:	Gli operandi di ingresso (ad esempio, source, tag di un'espressione, limit) vengono convertiti in:
Sì	REAL
No	DINT

- Dopo l'esecuzione di un'istruzione, se è necessario, il risultato (un valore DINT o REAL) viene convertito nel tipo di dati della destinazione.

**TI** Attualmente non è possibile specificare una tag BOOL di un'istruzione che opera su tipi di dati interi o REAL.

Poiché la conversione dei dati implica un maggiore utilizzo di tempo e memoria, è possibile incrementare l'efficienza dei programmi nel modo seguente:

- utilizzando lo stesso tipo di dati nell'istruzione
- riducendo al minimo l'uso dei tipi di dati SINT o INT

In breve, nelle istruzioni, si consiglia di utilizzare tutte tag DINT o tag REAL assieme ai valori immediati.

Le seguenti sezioni spiegano in che modo vengono convertiti i dati quando si utilizzano tag SINT o INT oppure quando si mischiano tipi di dati diversi.

### Da SINT o INT in DINT

Per quelle istruzioni che convertono i valori SINT o INT in valori DINT, la sezione "Operandi" indica il metodo di conversione.

Questo metodo di conversione:	Converte i dati in questo modo:
Estensione segno	il valore del bit più a sinistra (il segno del valore) viene inserito in ciascuna posizione bit alla sinistra dei bit esistenti fino a raggiungere 32 bit.
Riempimento con zeri	vengono inseriti degli zeri alla sinistra dei bit esistenti fino a raggiungere 32 bit

Il seguente esempio mostra il risultato della conversione di un valore mediante estensione del segno e riempimento con zeri.

Questo valore	2#1111_1111_1111_1111	(-1)
viene convertito in questo valore mediante estensione del segno	2#1111_1111_1111_1111_1111_1111_1111_1111	(-1)
viene convertito in questo valore mediante riempimento con zeri	2#0000_0000_0000_0000_1111_1111_1111_1111	(65535)

Poiché i valori immediati sono sempre riempiti con zeri, la conversione di un valore SINT o INT *potrebbe* produrre risultati imprevisti. Nell'esempio seguente, il confronto è falso in quanto Source A, un valore INT, viene convertito mediante estensione del segno, mentre Source B, un valore immediato, viene riempito con zeri.



Se in un'istruzione che converte dati mediante estensione segno si utilizzano una tag SINT o INT ed un valore immediato, per la gestione dei valori immediati si consiglia di seguire uno dei seguenti metodi:

- Specificare un qualsiasi valore immediato in decimale
- Se si inserisce un valore non decimale, specificare tutti i 32 bit del valore immediato. Per fare questo, inserire il valore del bit più a sinistra in ciascuna posizione bit alla sua sinistra fino a raggiungere 32 bit.
- Creare una tag per ciascun operando ed utilizzare lo stesso tipo di dati in tutta l'istruzione. Per assegnare un valore costante, procedere in uno dei seguenti modi:
  - Inserire il valore costante in una delle tag
  - Aggiungere un'istruzione MOV che sposti il valore in una delle tag.
- Utilizzare un'istruzione MEQ per controllare solo i bit richiesti

I seguenti esempi mostrano due modi di combinare un valore immediato con una tag INT. Entrambi gli esempi controllano i bit di un modulo I/O 1771 per determinare se tutti i bit sono attivi. Poichè la parola dati di ingresso di un modulo I/O 1771 è una tag INT, risulta più semplice utilizzare un valore costante a 16 bit.

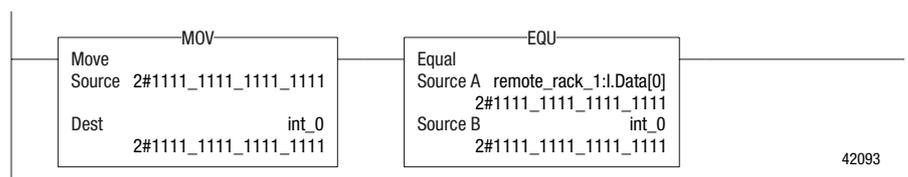
### Esempio 1: Combinazione di una tag INT con un valore immediato

Dato che *remote\_rack\_1:I.Data[0]* è una tag INT, il valore con cui raffrontarlo viene inserito come tag INT.



### Esempio 2: Combinazione di una tag INT con un valore immediato

Dato che *remote\_rack\_1:I.Data[0]* è una tag INT, il valore con cui raffrontarlo viene prima spostato in *int\_0*, anch'essa una tag INT. L'istruzione EQU, quindi, confronta le due tag.



## Da intero a REAL

Il controllore memorizza i valori REAL in numeri a virgola mobile singoli IEEE. Viene utilizzato un bit per il segno, 23 bit per il valore della base ed otto bit per l'esponente (32 bit in totale). Se nella stessa istruzione si utilizzano una tag di interi (SINT, INT o DINT) ed una tag REAL come ingressi, il controllore, prima che l'istruzione venga eseguita, converte gli interi in valori REAL.

- Un valore SINT o INT viene sempre convertito nello stesso valore REAL.
- Un valore DINT può non essere convertito nello stesso valore REAL:
  - un valore REAL utilizza un massimo di 24 bit per il valore base (23 bit memorizzati più un bit “nascosto”).
  - un valore DINT utilizza un massimo di 32 bit per il valore (1 per il segno e 31 per il valore).
  - se il valore DINT richiede più di 24 bit significativi, questo potrebbe *non* essere convertito nello stesso valore REAL. Se viene convertito, il controllore arrotonda per approssimazione al valore REAL più vicino utilizzando 24 bit significativi.

## Da DINT in SINT o INT

Per convertire un valore DINT in un valore SINT o INT, il controllore tronca la parte superiore del valore DINT e, se necessario, imposta l'indicatore di stato di overflow. Il seguente esempio mostra il risultato di una conversione da DINT in SINT o INT.

---

### Esempio: Conversione di un valore DINT in un valore INT e SINT

Questo valore DINT:	Viene convertito in questo valore più piccolo:
16#0001_0081 (65.665)	INT: 16#0081 (129)
	SINT: 16#81 (-127)

---

## Da REAL in intero

Per convertire un valore REAL in un valore intero, il controllore arrotonda la parte frazionaria e tronca la parte superiore della parte non frazionaria. Se si perdono dei dati il controllore imposta l'indicatore di stato di overflow. I numeri si arrotondano come segue:

- I numeri diversi da  $x.5$  si arrotondano per eccesso o per difetto.
- $X.5$  si arrotonda al numero pari più vicino.

Il seguente esempio mostra il risultato di una conversione da valori REAL in valori DINT.

---

### Esempio: Conversione di valori REAL in valori DINT

Questo valore REAL:	Viene convertito in questo valore DINT:
-2.5	-2
-1.6	-2
-1.5	-2
-1.4	-1
1.4	1
1.5	2
1.6	2
2.5	2

---

**Importante:** Gli indicatori di stato aritmetico sono impostati in base al valore che viene memorizzato. Le istruzioni che normalmente non influenzano le parole chiave di stato aritmetico potrebbero farlo se la conversione di tipo si verifica a causa dell'utilizzo di tipi di dati diversi nei parametri dell'istruzione. Il processo di conversione del tipo di dati imposta le parole chiave di stato aritmetico.

## Concetti sulle matrici

### Visualizzazione di una matrice come insieme di elementi

Le matrici consentono di raggruppare un insieme di dati (dello stesso tipo) sotto lo stesso nome e di utilizzare degli indici per identificare singoli elementi. L'elemento di una matrice può essere un tipo di dati di base o una struttura.

Specificare un elemento di una matrice mediante il suo indice (o indici). Inserire il nome della tag della matrice seguito dall'indice in parentesi quadra. L'indice deve specificare un valore per ciascuna dimensione della matrice. Le dimensioni partono da zero.

<b>Per questa matrice:</b>	<b>Specificare:</b>
----------------------------	---------------------

una dimensione	<i>array_name[subscript_0]</i>
----------------	--------------------------------

due dimensioni	<i>array_name[subscript_0, subscript_1]</i>
----------------	---

tre dimensioni	<i>array_name[subscript_0, subscript_1, subscript_2]</i>
----------------	--

Una matrice può avere fino a tre dimensioni. Il numero totale di elementi di una matrice è il risultato della grandezza di ciascuna dimensione.

**Questa matrice: Memorizza i dati così**

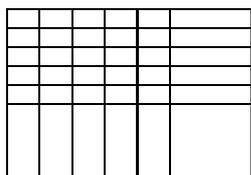
**Per esempio:**

una dimensione



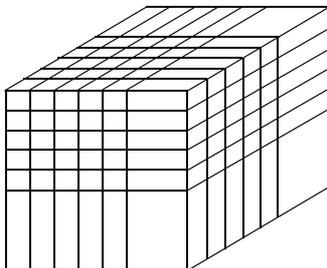
Nome tag	Tipo	Dimensione 0	Dimensione 1	Dimensione 2
<i>one_d_array</i>	DINT[7]	7	--	--
numero totale di elementi = 7				
gamma indice valida DINT[x] dove x=0-6				

due dimensioni



Nome tag	Tipo	Dimensione 0	Dimensione 1	Dimensione 2
<i>two_d_array</i>	DINT[4,5]	4	5	--
numero totale di elementi = 4 * 5 = 20				
gamma indice valida DINT[x,y] dove x=0-3; y=0-4				

tre dimensioni



Nome tag	Tipo	Dimensione 0	Dimensione 1	Dimensione 2
<i>three_d_array</i>	DINT[2,3,4]	2	3	4
numero totale di elementi = 2 * 3 * 4 = 24				
gamma indice valida DINT[x,y,z] dove x=0-1; y=0-2, z=0-3				

### Indicizzazione mediante matrici

Per cambiare dinamicamente l'elemento della matrice a cui fa riferimento la logica, utilizzare una tag o un'espressione come indice per indicare l'elemento. Si tratta di un processo simile all'indirizzamento indiretto dei PLC-5. Per specificare un indice di matrice è possibile utilizzare questi operatori nell'espressione:

Operatore:	Descrizione:
+	somma
-	sottrai/nega
*	moltiplica
/	dividi
ABS	valore assoluto
AND	AND
FRD	da BCD a intero
MOD	Modulo
NOT	complementa
OR	OR
SQR	radice quadrata
TOD	da intero a BCD
TRUN	tronca
XOR	OR esclusivo

Per esempio:

Definizioni:	Esempio:	Descrizione:
<i>my_list</i> definito come DINT[10]	<i>my_list</i> [5]	In questo elenco si fa riferimento all'elemento 5 della matrice. Il riferimento è statico poiché il valore dell'indice rimane costante.
<i>my_list</i> definito come DINT[10] <i>position</i> definito come DINT	MOV il valore 5 in <i>position</i> <i>my_list</i> [ <i>position</i> ]	In questo elenco si fa riferimento all'elemento 5 della matrice. Il riferimento è dinamico poiché la logica può modificare l'indice cambiando il valore di <i>position</i> .
<i>my_list</i> definito come DINT[10] <i>position</i> definito come DINT <i>offset</i> definito come DINT	MOV il valore 2 in <i>position</i> MOV il valore 5 in <i>offset</i> <i>my_list</i> [ <i>position</i> + <i>offset</i> ]	In questo esempio si fa riferimento all'elemento 7 (2+5) della matrice. Il riferimento è dinamico poiché la logica può modificare l'indice cambiando il valore di <i>position</i> o di <i>offset</i> .

Assicurarsi che qualsiasi indice inserito ricada entro i limiti della matrice specificata. Istruzioni che visualizzano le matrici come un insieme di elementi generano un errore grave (tipo 4, codice 20) qualora un indice sia superiore alle corrispondenti dimensioni.

## Specificazione di bit all'interno di matrici

È possibile indirizzare bit all'interno di elementi di matrici. Per esempio:

Definizioni:	Esempio:	Descrizione:
<i>array1</i> definita come DINT[5]	<i>array1</i> [1].2	In questo esempio si fa riferimento al bit 2 dell'elemento 1 della matrice.
<i>array2</i> definita come INT[17,36] la 1° dimensione contiene 17 elementi la 2° dimensione contiene 36 elementi	<i>array2</i> [3,4].15	In questo esempio si fa riferimento al bit 15 dell'elemento <i>array2</i> [3,4].
<i>array3</i> definita come SINT[2,4,6] la 1° dimensione contiene 2 elementi la 2° dimensione contiene 4 elementi la 3° dimensione contiene 6 elementi	<i>array3</i> [1,3,2].4	In questo esempio si fa riferimento al bit 4 dell'elemento <i>array3</i> [1,3,2].
<i>MyArray</i> definita come SINT[100] <i>MyIndex</i> definito come SINT	<i>MyArray</i> [( <i>MyIndex</i> AND NOT 7) / 8].[ <i>MyIndex</i> AND 7]	In questo esempio si fa riferimento ad un bit all'interno di una matrice SINT.
<i>MyArray</i> definita come INT[100] <i>MyIndex</i> definita come INT	<i>MyArray</i> [( <i>MyIndex</i> AND NOT 15) / 16].[ <i>MyIndex</i> AND 15]	In questo esempio si fa riferimento ad un bit all'interno di una matrice INT.
<i>MyArray</i> definita come DINT[100] <i>MyIndex</i> definito come DINT	<i>MyArray</i> [( <i>MyIndex</i> AND NOT 31) / 32].[ <i>MyIndex</i> AND 31]	In questo esempio si fa riferimento ad un bit all'interno di una matrice DINT.

Per specificare i bit è anche possibile utilizzare gli operatori riportati nella tabella a pagina B-3.

## Visualizzazione di una matrice come blocco di memoria

I dati di una matrice vengono archiviati in memoria in modo continuativo. Le istruzioni di un file (matrice) generalmente richiedono un indirizzo di avviamento all'interno di una matrice ed una lunghezza che determina quali e quanti elementi l'istruzione legge o scrive.

**Important:** Se un'istruzione tenta di leggere i dati oltre la fine di una matrice, l'istruzione legge questi dati e li elabora come se fossero dati validi (senza che si verifichi alcun errore). Se un'istruzione tenta di scrivere dati al di fuori di una matrice, si verifica un errore grave (tipo 4, codice 20).

Queste istruzioni manipolano i dati della matrice come un blocco contiguo di memoria (le restanti istruzioni manipolano i dati di una matrice come singoli elementi):

BSL	FBC	LFL	SQO
BSR	FFL	LFU	
COP	FFU	SQI	
DDT	FLL	SQL	

### Come il controllore memorizza i dati della matrice

La seguente tabella mostra l'ordine sequenziale degli elementi degli esempi a pagina B-2.

<b>Elementi di matrice ad una dimensione (ordine crescente):</b>	<b>Elementi di matrice a due dimensioni (ordine crescente):</b>	<b>Elementi di matrice a tre dimensioni (ordine crescente):</b>
one_d_array[0]	two_d_array[0,0]	three_d_array[0,0,0]
one_d_array[1]	two_d_array[0,1]	three_d_array[0,0,1]
one_d_array[2]	two_d_array[0,2]	three_d_array[0,0,2]
one_d_array[3]	two_d_array[0,3]	three_d_array[0,0,3]
one_d_array[4]	two_d_array[0,4]	three_d_array[0,1,0]
one_d_array[5]	two_d_array[1,0]	three_d_array[0,1,1]
one_d_array[6]	two_d_array[1,1]	three_d_array[0,1,2]
	two_d_array[1,2]	three_d_array[0,1,3]
	two_d_array[1,3]	three_d_array[0,2,0]
	two_d_array[1,4]	three_d_array[0,2,1]
	two_d_array[2,0]	three_d_array[0,2,2]
	two_d_array[2,1]	three_d_array[0,2,3]
	two_d_array[2,2]	three_d_array[1,0,0]
	two_d_array[2,3]	three_d_array[1,0,2]
	two_d_array[2,4]	three_d_array[1,0,3]
	two_d_array[3,0]	three_d_array[1,1,0]
	two_d_array[3,1]	three_d_array[1,1,1]
	two_d_array[3,2]	three_d_array[1,1,2]
	two_d_array[3,3]	three_d_array[1,1,3]
	two_d_array[3,4]	three_d_array[1,2,0]
		three_d_array[1,2,1]
		three_d_array[1,2,2]
		three_d_array[1,2,3]
Per una matrice ad una sola dimensione, <i>tag_name[subscript_0]</i> , <i>subscript_0</i> aumenta fino al suo valore massimo.	Per una matrice a due dimensioni, <i>tag_name[subscript_0,subscript_1]</i> , <i>subscript_0</i> è fisso a 0 mentre <i>subscript_1</i> aumenta da 0 fino al suo valore massimo. <i>Subscript_0</i> aumenta quindi di 1 (se la dimensione 0 è maggiore di 1) e rimane fisso mentre <i>subscript_1</i> aumenta ancora nella propria gamma. E così finché entrambi gli indici non raggiungono i loro valori massimi.	Nel caso di una matrice a tre dimensioni, <i>tag_name[subscript_0, subscript_1, subscript_2]</i> , <i>subscript_0</i> è fisso a 0 mentre <i>subscript_1</i> e <i>subscript_2</i> aumentano come in una matrice a due dimensioni. <i>Subscript_0</i> aumenta quindi di 1 (se la dimensione 0 è maggiore di 1) e rimane fisso finché <i>subscript_1</i> e <i>subscript_2</i> raggiungono i loro valori massimi. E così finché tutti e tre gli indici non raggiungono i loro valori massimi.

### Variazione di una dimensione

Le istruzioni AVE, SRT ed STD hanno un operando Dimension da variare. L'istruzione utilizza questo operando per calcolare un offset che l'istruzione usa per determinare quali elementi della matrice leggere o scrivere.

Matrice:	Dimensione da variare:	Offset:
una dimensione	0	1
due dimensioni	0	dimension_1
	1	1
tre dimensioni	0	$(dimension_1) * (dimension_2)$
	1	dimension_2
	2	1

### Allocazione di memoria per matrici

La quantità di memoria che una matrice utilizza dipende dal tipo di dati usati per creare la matrice. L'allocazione minima nel controllore è di quattro byte pari a quella del 32 BOOL, 4 SINT, 2 INT o 1 DINT.

I seguenti esempio mostrano l'allocazione di memoria per le varie matrici:

I seguenti esempio mostrano l'allocazione di memoria per le varie matrici:

bit\_values come BOOL[32] Questo è un esempio di matrice a 32 elementi bit di tipo di dati BOOL (1 bit per elemento).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
allocazione dati 1	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
allocazione dati 1 <i>continua</i>	[31]	[30]	[29]	[28]	[27]	[26]	[25]	[24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]

In questo esempio si utilizzano 64 bit di memoria.

small\_values come SINT[8] Questo è un esempio di matrice ad 8 elementi di tipo di dati SINT (1 byte per elemento).

Bit:	31	24	23	16	15	8	7	0
allocazione dati 1	small_values[3]	small_values[2]	small_values[1]	small_values[0]				
allocazione dati 2	small_values[7]	small_values[6]	small_values[5]	small_values[4]				

In questo esempio si utilizzano 64 bit di memoria.

**small\_values come SINT[3]** Questo è un esempio di matrice a 3 elementi di tipo di dati SINT (1 byte per elemento). Dato che l’allocazione di dati minima è di 4 byte, l’ultimo byte è zero.

Bit:	31	24	23	16	15	8	7	0
allocazione dati 1	0		small_values[2]		small_values[1]		small_values[0]	

In questo esempio si utilizzano 32 bit di memoria.

**valori come INT[4]** Questo è un esempio di matrice a 4 elementi di tipo di dati INT (2 byte per elemento).

Bit:	31	16	15	0
allocazione dati 1	values[1]		values[0]	
allocazione dati 2	values[3]		values[2]	

In questo esempio si utilizzano 64 bit di memoria.

**big\_values come DINT[2]** Questo è un esempio di matrice a 2 elementi di tipo di dati DINT (4 byte per elemento).

Bit:	31	0
allocazione dati 1	big_values[0]	
allocazione dati 2	big_values[1]	

In questo esempio si utilizzano 64 bit di memoria.

**timer\_list come TIMER[2]** Questo è un esempio di matrice a 2 elementi dove ogni elemento è una struttura TIMER (12 byte per struttura).

Bit:	31	0
allocazione dati 1	timer_list[0] bit di stato	
allocazione dati 2	timer_list[0].pre	
allocazione dati 3	timer_list[0].acc	
allocazione dati 4	timer_list[1] bit di stato	
allocazione dati 5	timer_list[1].pre	
allocazione dati 6	timer_list[1].acc	

In questo esempio si utilizzano 192 bit di memoria.

small\_values come SINT[2,2,2] Questo è un esempio di matrice a tre dimensioni con 8 elementi di tipo di dati SINT.

Bit:	31	24	23	16	15	8	7	0
allocazione dati 1	small_values[0,1,1]	small_values[0,1,0]	small_values[0,0,1]	small_values[0,0,0]				
allocazione dati 2	small_values[1,1,1]	small_values[1,1,0]	small_values[1,0,1]	small_values[1,0,0]				

In questo esempio si utilizzano 64 bit di memoria.

big\_values come DINT[2,2,2] Questo è un esempio di matrice a tre dimensioni con 8 elementi di tipo di dati DINT.

Bit:	31	0
allocazione dati 1	big_values[0,0,0]	
allocazione dati 2	big_values[0,0,1]	
allocazione dati 3	big_values[0,1,0]	
allocazione dati 4	big_values[0,1,1]	
allocazione dati 5	big_values[1,0,0]	
allocazione dati 6	big_values[1,0,1]	
allocazione dati 7	big_values[1,1,0]	
allocazione dati 8	big_values[1,1,1]	

In questo esempio si utilizzano 256 bit di memoria.

Nella programmazione offline è possibile modificare le dimensioni di una matrice senza perdere i dati delle tag. Non è possibile modificare le dimensioni di una matrice durante la programmazione online.

## Tempo d'esecuzione

### Introduzione

Utilizzare questa appendice per calcolare il tempo necessario per l'esecuzione di un'istruzione. I tempi calcolati si riferiscono al controllore Logix5550 (Num. di cat. 1756-L1) e al software RSLogix 5000, versione 2.25.

**Important:** si consiglia di utilizzare queste informazioni per scegliere l'adeguata opzione di programmazione. I tempi di esecuzione effettivi possono variare in base alla configurazione del progetto e alla versione del software RSLogix 5000 utilizzato.

#### Per calcolare il tempo di esecuzione di un'istruzione:

1. Dalla Table C.2 a pagina C-3, trovare la base di tempo dell'istruzione.
2. A questo valore sommare il tempo per l'espressione, gli indici e la conversione dei dati:

Se un operando:	Aggiungere tempo per:	Utilizzando questa tabella:
contiene un'espressione	ciascuna operazione dell'espressione. Per ogni operazione, utilizzare il tempo della corrispondente istruzione.	Table C.2 a pagina C-3
è un elemento di una matrice ed uno o più degli indici è una tag (ad esempio, <i>tag_c[tag_d]</i> )	il calcolo dell'indice della matrice	Table C.5 a pagina C-15
viene convertito mediante estensione del segno	la conversione dei dati. Fare riferimento a "Conversione di dati" a pag. A-6.	Table C.1 a pagina C-3
contiene una tag SINT o INT in un indice della matrice	la conversione del valore SINT o INT in DINT	

**Esempi:**

<b>Istruzione:</b>	<b>Tipo di dati:</b>	<b>Tempo d'esecuzione:</b>
<hr/>		
CMP ( <i>tag_a * tag_b / 100</i> )		4,3 μs di overhead
* (MUL)	REAL	17,7 μs per operazione MUL
/ (DIV)	REAL	18,3 μs per l'operazione DIV
		40,3 μs totali
<hr/>		
ADD		10,7 μs per l'istruzione ADD
Source A	DINT	8,4 μs per la conversione da DINT in REAL
Source B	REAL	
Destination	REAL	
		19,1 μs totali
<hr/>		
ADD		10,7 μs per l'istruzione ADD
Source A	DINT[0,DINT]	25,5 μs per l'indice della matrice 8,4 μs per la conversione da DINT in REAL della matrice
Source B	REAL	
Destination	REAL	
		44,6 μs totali
<hr/>		
MUL		17,7 μs per l'istruzione MUL
Source A	REAL	
Source B	DINT	8,4 μs per la conversione da DINT in REAL
Destination	DINT	12,3 μs per la conversione da REAL in DINT
		38,4 μs totali
<hr/>		

**Tabelle di riferimento****Table C.1 Tempi della conversione mediante estensione del segno**

Da un valore:	In un valore:	Tempo ( $\mu$ s):
SINT	INT	8,3
	DINT	0,98
	REAL	2,1
INT	SINT	8,3
	DINT	1,1
	REAL	2,3
DINT	SINT	7,3
	INT	7,4
	REAL	8,4
REAL	SINT	14,7
	INT	14,8
	DINT	12,3

**Table C.2 Tempo d'esecuzione delle istruzioni (Continua)**

Istruzione:	Tipo di dati ottimale:	Tempo vero ( $\mu$ s):	Tempo falso ( $\mu$ s):	Note:
ABS	DINT	0,59	0,17	
	REAL	0,89	0,19	
ACS	REAL	268,2	0,12	
ADD	DINT	0,56	0,10	
	REAL	10,7	0,11	
AFI	na	0,05		
AND	DINT	0,56	0,11	
ASN	REAL	258,9	0,11	
ATN	REAL	220,9	0,11	
AVE	REAL	$43,5 + (\text{Length} * 14,51)$	23,5	
BRK	na	0		
BSL	DINT	$7,8 + ((\text{Length}/32) * 0,77)$	4,3	Length/32 viene arrotondato al numero intero.
BSR	DINT	$8,7 + ((\text{Length}/32) * 0,77)$	4,3	Length/32 viene arrotondato al numero intero.
BTD	DINT	3,3	0,16	
CLR	DINT	0,35	0,10	
	REAL	0,40	0,11	

Table C.2 Tempo d'esecuzione delle istruzioni (Continua)

Istruzione:	Tipo di dati ottimale:	Tempo vero ( $\mu$ s):	Tempo falso ( $\mu$ s):	Note:
CMP	Fare riferimento alle operazioni all'interno dell'espressione.	$4,3 + expression$	0,11	
COP	SINT	$5,8 + (Length * 0,09)$		
	INT	$5,6 + (Length * 0,17)$		
	DINT, REAL	$6 + (Length * 0,35)$	0,15	
COS	REAL	218,7	0,08	
CPT	Fare riferimento alle operazioni all'interno dell'espressione.	$4,7 + expression$	0,11	
CTD	COUNTER	0,43	0,43	
CTU	COUNTER	0,43	0,43	
DDT	DINT		9,1	Modalità All (Continua)
	0 mancate corrispondenze	$13 + (Cmp. Length * 1,38)$		
	1 mancata corrispondenza	$27,6 + (Cmp. Length * 1,38)$		
	2 mancate corrispondenze	$41,2 + (Cmp. Length * 1,38)$		
DEG	REAL	18,5	0,11	
DIV	DINT	10,4	0,10	
	REAL	18,3	0,11	
DTR	DINT	2,8	2,8	
EQU	DINT	0,37	0,10	
	REAL	0,37	0,11	
FAL	Fare riferimento alle operazioni all'interno dell'espressione.	$10,2 + (n * (3,99 + expression))$	7,3	$n$ è il numero di elementi manipolati in una scansione.
FBC	DINT		9,2	Modalità All (Continua)
	0 mancate corrispondenze	$13,1 + (Cmp. Length * 1,38)$		
	1 mancata corrispondenza	$26,4 + (Cmp. Length * 1,38)$		
	2 mancate corrispondenze	$38,6 + (Cmp. Length * 1,38)$		
FFL	SINT	9,7		
	INT	10,7		
	DINT, REAL	10,1	7,5	

**Table C.2 Tempo d'esecuzione delle istruzioni (Continua)**

<b>Istruzione:</b>	<b>Tipo di dati ottimale:</b>	<b>Tempo vero (<math>\mu</math>s):</b>	<b>Tempo falso (<math>\mu</math>s):</b>	<b>Note:</b>
FFU	SINT	$10,4 + (\text{Length} * 0,56)$		
	INT	$11,3 + (\text{Length} * 0,98)$		
	DINT, REAL	$11,7 + (\text{Length} * 0,56)$	7,7	
FLL	SINT	$5,8 + (\text{Length} * 0,09)$		
	INT	$5,6 + (\text{Length} * 0,17)$		
	DINT, REAL	$6 + (\text{Length} * 0,35)$	0,16	
FOR	DINT	$16 + ((\text{Terminal value}/\text{Step size}) * 6,92)$	3	
FRD	DINT	9,1	0,10	
FSC	Fare riferimento alle operazioni all'interno dell'espressione.	$10,7 + (n * (3,89 + \text{expression}))$	7,8	<i>n</i> è il numero di elementi manipolati in una scansione.
GEQ	DINT	0,37	0,11	
	REAL	0,58	0,11	
GRT	DINT	0,37	0,11	
	REAL	0,59	0,11	
GSV	na	See Table C.3 on page C-8	0,16	
JMP	na	1,6	0,10	
JSR	na	10,62		Nessun parametro
JSR/SBR	SINT	$21 + (\text{numero di parametri} * 3,49)$		Il tempo si riferisce alla coppia JSR/SBR.
	INT	$21 + (\text{numero di parametri} * 3,86)$		
	DINT	$21 + (\text{numero di parametri} * 3,27)$		
	REAL	$21 + (\text{numero di parametri} * 3,33)$		
JSR/RET	SINT	$20,4 + (\text{numero di parametri} * 3,49)$		Il tempo si riferisce alla coppia JSR/RET.
	INT	$20,4 + (\text{numero di parametri} * 3,86)$		
	DINT	$20,4 + (\text{numero di parametri} * 3,27)$		
	REAL	$20,4 + (\text{numero di parametri} * 3,33)$		
LBL	na	0,16		
LEQ	DINT	0,37	0,11	
	REAL	0,58	0,11	
LES	DINT	0,37	0,10	
	REAL	0,58	0,11	
LFL	SINT	9,7		
	INT	10,7		
	DINT, REAL	10,1	7,5	

Table C.2 Tempo d'esecuzione delle istruzioni (Continua)

Istruzione:	Tipo di dati ottimale:	Tempo vero ( $\mu$ s):	Tempo falso ( $\mu$ s):	Note:
LFU	SINT	11,5		
	INT	12,8		
	DINT, REAL	13,7	7,6	
LIM	DINT	0,79	0,11	
	REAL	3,5	0,11	
LN	REAL	189,8	0,10	
LOG	REAL	190,3	0,11	
MCR	na	0,05	0,05	
MEQ	DINT	0,58	0,10	
MOD	DINT	20,1	0,22	
	REAL	63,3	0,20	
MOV	DINT	0,51	0,11	
	REAL	0,88	0,10	
MSG	MESSAGE	93,7	7	
MUL	DINT	9,5	0,11	
	REAL	17,7	0,11	
MVM	DINT	2,1	0,16	
NEG	DINT	0,55	0,11	
	REAL	0,93	0,11	
NEQ	DINT	0,37	0,10	
	REAL	0,37	0,11	
NOP	na	0,05	0,06	
NOT	DINT	0,50	0,11	
ONS	BOOL	2,9	2,7	
OR	DINT	0,55	0,11	
OSF	BOOL	3,4	4	
OSR	BOOL	3,7	3,5	
OTE	BOOL	0,19	0,19	
OTL	BOOL	0,19	0,05	
OTU	BOOL	0,19	0,05	

**Table C.2 Tempo d'esecuzione delle istruzioni (Continua)**

<b>Istruzione:</b>	<b>Tipo di dati ottimale:</b>	<b>Tempo vero (<math>\mu</math>s):</b>	<b>Tempo falso (<math>\mu</math>s):</b>	<b>Note:</b>
PID	PID		18,4	
	Indipendente	349,3		
	ISA-dipendente	416,9		
	Modalità manuale	304,6		
	Modalità di impostazione uscita	304,6		
	Modalità indipendente-slave	371		
RAD	REAL	18,1	0,10	
RES	CONTROL, COUNTER o TIMER	0,32	0,21	
RET (in anello FOR)	na	4,55		
RTO	TIMER	0,42	0,31	
SIN	REAL	224,8	0,09	
SQI	DINT	3,6	0,16	
SQL	DINT	6,1	3,9	
SQO	DINT	6,2	3,8	
SQR	DINT	9,9	0,10	
	REAL	37,1	0,11	
SRT	DINT	30,1 + x	4,6	Il tempo "x" varia in base alla lunghezza ed alla casualità dei numeri.
	REAL	30,8 + x	4,3	
SSV	na	See Table C.4 on page C-13	0,15	
STD	SINT	104,5 + (Length * 51,18)	22	
	INT	111,9 + (Length * 51,79)	25,3	
	DINT	112 + (Length * 50,71)	25,3	
	REAL	113,5 + (Length * 55,08)	25,4	
SUB	DINT	0,55	0,10	
	REAL	10,6	0,11	
TAN	REAL	284,7	0,08	
TND	na	0,00	0,11	
TOD	DINT	14,8	0,10	
TOF	TIMER	0,32	0,42	
TON	TIMER	0,43	0,34	
TRN	DINT	13	0,21	
	REAL	21	0,21	

Table C.2 Tempo d'esecuzione delle istruzioni (Continua)

Istruzione:	Tipo di dati ottimale:	Tempo vero ( $\mu$ s):	Tempo falso ( $\mu$ s):	Note:
UID	na	39,1	2,6	
UIE	na	38,5	2,6	
XIC	BOOL	0,11	0,05	
XIO	BOOL	0,11	0,05	
XOR	DINT	0,56	0,11	
XPY	REAL	274,4	0,10	Il tempo vero può variare tra 200 e 400 $\mu$ s, a seconda dei valori degli operandi.

Table C.3 Tempi di esecuzione dell'istruzione GSV (Continua)

Oggetto:	Attributo:	Tempo vero ( $\mu$ s):
AXIS	AccelerationFeedforwardGain	90,3
AXIS	ActualPosition	160
AXIS	ActualVelocity	165
AXIS	AverageVelocity	250,2
AXIS	AverageVelocityTimebase	25,6
AXIS	AxisConfigurationState	20,6
AXIS	AxisState	18,7
AXIS	AxisType	20
AXIS	C2CConnectionInstance	22,3
AXIS	C2CMapTableInstance	22,6
AXIS	CommandPosition	159
AXIS	CommandVelocity	164
AXIS	ConversionConstant	22,7
AXIS	DampingFactor	20,3
AXIS	DriveFaultAction	21
AXIS	EffectiveInertia	22,4
AXIS	EncoderLossFaultAction	19,6
AXIS	EncoderNoiseFaultAction	20,7
AXIS	FrictionCompensation	83,7
AXIS	GroupInstance	22,5
AXIS	HomeMode	20,3
AXIS	HomePosition	159,9

**Table C.3 Tempi di esecuzione dell'istruzione GSV (Continua)**

<b>Oggetto:</b>	<b>Attributo:</b>	<b>Tempo vero (<math>\mu</math>s):</b>
AXIS	HomeReturnSpeed	34,2
AXIS	HomeSequenceType	19,6
AXIS	HomeSpeed	33,9
AXIS	Instance	18,6
AXIS	MapTableInstance	22,6
AXIS	MaximumAcceleration	62,4
AXIS	MaximumDeceleration	62,7
AXIS	MaximumNegativeTravel	160,5
AXIS	MaximumPositiveTravel	160,2
AXIS	MaximumSpeed	61,2
AXIS	ModuleChannel	21
AXIS	MotionConfigurationBits	21,3
AXIS	MotionFaultBits	22,1
AXIS	MotionStatusBits	22,4
AXIS	MotorEncoderTestIncrement	159,6
AXIS	OutputFilterBandwidth	22,5
AXIS	OutputLimit	82,3
AXIS	OutputOffset	83,2
AXIS	OutputScaling	200,8
AXIS	PositionError	110,9
AXIS	PositionErrorFaultAction	21,3
AXIS	PositionErrorTolerance	198,6
AXIS	PositionIntegralGain	191,4
AXIS	PositionIntegratorError	112,4
AXIS	PositionLockTolerance	159,8
AXIS	PositionProportionalGain	119,2
AXIS	PositionServoBandwidth	21,8
AXIS	PositionUnwind	21,9
AXIS	ProgrammedStopMode	20,3
AXIS	RegistrationPosition	159,7
AXIS	ServoConfigurationBits	22,9
AXIS	ServoConfigurationUpdateB	21,9
AXIS	ServoEventBits	21,6
AXIS	ServoFaultBits	21,7

**Table C.3 Tempi di esecuzione dell'istruzione GSV (Continua)**

<b>Oggetto:</b>	<b>Attributo:</b>	<b>Tempo vero (<math>\mu</math>s):</b>
AXIS	ServoOutputLevel	108,2
AXIS	ServoStatusBits	22,3
AXIS	ServoStatusUpdateBits	21,4
AXIS	SoftOvertravelFaultAction	21
AXIS	StartActualPosition	160
AXIS	StartCommandPosition	158,7
AXIS	StrobeActualPosition	160
AXIS	StrobeCommandPosition	159
AXIS	TestDirectionForward	20,3
AXIS	TestStatus	19,8
AXIS	TuneAcceleration	186,8
AXIS	TuneAccelerationTime	91,8
AXIS	TuneDeceleration	187
AXIS	TuneDecelerationTime	92,5
AXIS	TuneRiseTime	93,1
AXIS	TuneSpeedScaling	200,5
AXIS	TuneStatus	20,2
AXIS	TuneVelocityBandwidth	20,9
AXIS	TuningConfigurationBits	22,1
AXIS	TuningSpeed	165
AXIS	TuningTravelLimit	159,9
AXIS	VelocityCommand	111,4
AXIS	VelocityError	111,4
AXIS	VelocityFeedback	112,7
AXIS	VelocityFeedforwardGain	90,6
AXIS	VelocityIntegralGain	191,3
AXIS	VelocityIntegratorError	111,6
AXIS	VelocityProportionalGain	119,3
AXIS	WatchPosition	159,7
CONTROLLER	TimeSlice	16,9
CONTROLLERDEVICE	DeviceName	55,2
CONTROLLERDEVICE	ProductCode	15,4
CONTROLLERDEVICE	ProductRev	15,4
CONTROLLERDEVICE	SerialNumber	16,3

**Table C.3 Tempi di esecuzione dell'istruzione GSV (Continua)**

<b>Oggetto:</b>	<b>Attributo:</b>	<b>Tempo vero (<math>\mu</math>s):</b>
CONTROLLERDEVICE	Status	15,4
CONTROLLERDEVICE	Tipo	15,4
CONTROLLERDEVICE	Produttore	15,4
CST	CurrentStatus	14,4
CST	CurrentValue	28,2
DF1	ACTTimeout	16,5
DF1	DiagnosticCounters	67,2
DF1	DuplicationDetect	14,9
DF1	EmbeddedResponseEnable	14,9
DF1	ENQTransmitLimit	14,9
DF1	EOTSuppression	14,9
DF1	ErrorDetection	14,9
DF1	MasterMessageTransmit	14,9
DF1	NAKReceiveLimit	14,9
DF1	NormalPollGroupSize	15,7
DF1	PollingMode	14,9
DF1	ReplyMessageWait	16,5
DF1	StationAddress	15,7
DF1	SlavePollTimeout	16,5
DF1	TransmitRetries	14,9
FAULTLOG	MajorEvents	16,7
FAULTLOG	MinorEvents	16,7
FAULTLOG	MajorFaultBits	17,5
FAULTLOG	MinorFaultBits	17,5
MESSAGE	ConnectionPath	53,1
MESSAGE	ConnectionRate	17,4
MESSAGE	MessageType	15,7
MESSAGE	Port	15,7
MESSAGE	TimeoutMultiplier	15,7
MESSAGE	UnconnectedTimeout	17,4
MODULE	EntryStatus	16,7
MODULE	FaultCode	16,7
MODULE	FaultInfo	17,6
MODULE	ForceStatus	144,5

**Table C.3 Tempi di esecuzione dell'istruzione GSV (Continua)**

<b>Oggetto:</b>	<b>Attributo:</b>	<b>Tempo vero (μs):</b>
MODULE	Instance	17,8
MODULE	Mode	16,7
MODULE	LEDStatus	17,2
MOTIONGROUP	Instance	24,2
PROGRAM	DisableFlag	16,4
PROGRAM	Instance	17,8
PROGRAM	LastScanTime	17,6
PROGRAM	MajorFaultRecord	59
PROGRAM	MaxScanTime	17,6
PROGRAM	MinorFaultRecord	59
PROGRAM	SFCRestart	16,7
ROUTINE	Instance	16,8
SERIALPORT	BaudRate	16,7
SERIALPORT	DataBits	15
SERIALPORT	Parity	15
SERIALPORT	RTSOffDelay	15,8
SERIALPORT	RTSSendDelay	15,8
SERIALPORT	StopBits	15
TASK	Instance	17,7
TASK	LastScanTime	17,5
TASK	MaxInterval	21,6
TASK	MaxScanTime	17,4
TASK	MinInterval	21,6
TASK	Priority	16,6
TASK	Rate	17,4
TASK	StartTime	21,6
TASK	Watchdog	17,4
WALLCLOCKTIME	CSTOffset	21,2
WALLCLOCKTIME	CurrentValue	37,6
WALLCLOCKTIME	DateTime	59,8

**Table C.4 Tempi di esecuzione dell'istruzione SSV (Continua)**

<b>Oggetto:</b>	<b>Attributo:</b>	<b>Tempo vero (<math>\mu</math>s):</b>
AXIS	AccelerationFeedforwardGain	105,4
AXIS	AverageVelocityTimebase	168,1
AXIS	AxisType	39,2
AXIS	ConversionConstant	126,9
AXIS	DampingFactor	129,8
AXIS	DriveFaultAction	40,3
AXIS	EncoderLossFaultAction	40,2
AXIS	EncoderNoiseFaultAction	40
AXIS	FrictionCompensation	140,1
AXIS	HomeMode	38,2
AXIS	HomePosition	223,1
AXIS	HomeReturnSpeed	511,7
AXIS	HomeSequenceType	37,6
AXIS	HomeSpeed	511,3
AXIS	MaximumAcceleration	511,5
AXIS	MaximumDeceleration	511,8
AXIS	MaximumNegativeTravel	224,7
AXIS	MaximumPositiveTravel	224,7
AXIS	MaximumSpeed	511,7
AXIS	MotionConfigurationBits	38,8
AXIS	MotorEncoderTestIncrement	207
AXIS	OutputFilterBandwidth	45,7
AXIS	OutputLimit	139,4
AXIS	OutputOffset	140,2
AXIS	OutputScaling	383,1
AXIS	PositionErrorFaultAction	39,2
AXIS	PositionErrorTolerance	202,7
AXIS	PositionIntegralGain	218
AXIS	PositionLockTolerance	258,1
AXIS	PositionProportionalGain	153,1
AXIS	PositionServoBandwidth	44,1
AXIS	PositionUnwind	128
AXIS	ProgrammedStopMode	38,6
AXIS	ServoConfigurationBits	42,2
AXIS	SoftOvertravelFaultAction	40,3
AXIS	TuningConfigurationBits	38,9

**Table C.4 Tempi di esecuzione dell'istruzione SSV (Continua)**

<b>Oggetto:</b>	<b>Attributo:</b>	<b>Tempo vero (<math>\mu</math>s):</b>
AXIS	TuningSpeed	580,6
AXIS	TuningTravelLimit	207,2
AXIS	VelocityFeedforwardGain	106,1
AXIS	VelocityIntegralGain	218
AXIS	VelocityProportionalGain	152,1
CONTROLLER	TimeSlice	35,9
DF1	PendingACKTimeout	109,4
DF1	PendingDuplicateDetection	108,3
DF1	PendingEmbeddedResponseEnable	108,7
DF1	PendingENQTransmitLimit	108,3
DF1	PendingEOTSuppression	108,2
DF1	PendingErrorDetection	108,9
DF1	PendingNormalPollGroupSize	108,9
DF1	PendingMasterMessageTransmit	108,7
DF1	PendingNAKReceiveLimit	108,3
DF1	PendingPollingMode	108,7
DF1	PendingReplyMessageWait	109,4
DF1	PendingStationAddress	109,1
DF1	PendingSlavePollTimeout	109,4
DF1	PendingTransmitRetries	108,3
FAULTLOG	MajorEvents	17
FAULTLOG	MinorEvents	17
FAULTLOG	MajorFaultBits	17,4
FAULTLOG	MinorFaultBits	17,4
MESSAGE	ConnectionPath	36,9
MESSAGE	ConnectionRate	32,3
MESSAGE	MessageType	69,9
MESSAGE	Port	31,6
MESSAGE	TimeoutMultiplier	31,7
MESSAGE	UnconnectedTimeout	32,3
MODULE	Mode	28,4
PROGRAM	DisableFlag	37,3
PROGRAM	LastScanTime	17,5
PROGRAM	MajorFaultRecord	48,8
PROGRAM	MaxScanTime	17,5
PROGRAM	MinorFaultRecord	48,8
PROGRAM	SFCRestart	33

**Table C.4 Tempi di esecuzione dell'istruzione SSV (Continua)**

<b>Oggetto:</b>	<b>Attributo:</b>	<b>Tempo vero (<math>\mu</math>s):</b>
SERIALPORT	PendingBaudRate	108,5
SERIALPORT	PendingDataBits	106,7
SERIALPORT	PendingParity	106,6
SERIALPORT	PendingRTSOffDelay	106,8
SERIALPORT	PendingRTSSendDelay	107,4
SERIALPORT	PendingStopBits	106,7
TASK	LastScanTime	17,7
TASK	MaxInterval	24,6
TASK	MaxScanTime	17,7
TASK	MinInterval	24,6
TASK	StartTime	24,6
TASK	Watchdog	33,8
WALLCLOCKTIME	CSTOffset	2149,4
WALLCLOCKTIME	CurrentValue	70,5
WALLCLOCKTIME	DateTime	120

**Table C.5 Tempi degli indici di matrice**

<b>Per una tag nell'indice di una:</b>	<b>Aggiungere (<math>\mu</math>s):</b>
matrice ad una dimensione	1,5
matrice a due dimensioni	25,5
matrice a tre dimensioni	30,3

**Nota:**

## Utilizzo della memoria

### Introduzione

Utilizzare questa appendice per calcolare la memoria utilizzata per la conversione dei dati, le istruzioni e gli indici di matrice. I valori si riferiscono al controllore Logix5550 (Num. di cat. 1756-L1) e al software RSLogix 5000, versione 2.25.

**TIP**

Quando si scarica un progetto, il controllore ottimizza l'uso della memoria riducendo il suo utilizzo previsto di circa il 10 % rispetto ai valori riportati in questa appendice.

<b>Per calcolare la memoria per:</b>	<b>Vedere pagina:</b>
conversione dati	D-2
istruzioni ladder	D-3
indici di matrice	D-7

## Conversione dei dati

Se si utilizza un tipo di dati non ottimale o si mischiano tipi di dati diversi, è necessario aggiungere memoria per la conversione dei dati. Refer to “Conversione di dati” on page A-6.

### Requisiti di memoria per l'estensione del segno

Utilizzare la seguente tabella per determinare la memoria necessaria per convertire i dati mediante estensione del segno.

Per convertire un:	Da:	A:	Aggiungere (byte):
operando Source	SINT	DINT	48
		REAL	96
	INT	DINT	60
		REAL	108
	DINT	REAL	52
	operando Destination	DINT	SINT
INT			52
REAL			76
REAL		SINT	48
		INT	48
		DINT	48

### Requisiti di memoria per il riempimento con zeri

Utilizzare la seguente tabella per determinare la memoria necessaria per convertire i dati mediante riempimento di zeri.

Da un valore:	Da:	A:	Aggiungere (byte):
operando Source	SINT	DINT	0
	INT	DINT	8
operando Destination	DINT	SINT	52
		INT	52

**Esempi:**

<b>Istruzione:</b>	<b>Tipo di dati:</b>	<b>Memoria (byte):</b>
ADD	REAL	44 byte per l'istruzione ADD
Source A	DINT	52 byte per la conversione da DINT in REAL mediante estensione del segno
Source B	REAL	
Destination	REAL	
		96 byte totali
MUL	REAL	44 byte per l'istruzione MUL
Source A	REAL	
Source B	DINT	52 byte per la conversione da DINT in REAL mediante estensione del segno
Destination	DINT	48 byte per la conversione da REAL in DINT mediante estensione del segno
		144 byte totali

**Istruzioni**

La seguente tabella mostra l'utilizzo di memoria delle istruzioni ladder: Per le istruzioni che utilizzano un'espressione (CMP, CPT, FAL, FSC), è necessario aggiungere memoria per ogni operazione all'interno dell'espressione. Per ogni operazione, utilizzare il valore della corrispondente istruzione.

**Esempio:**

<b>Istruzione:</b>	<b>Tipo di dati</b>	<b>Memoria:</b>
CMP ( $tag\_a * tag\_b / 100$ )		76 byte di overhead
* (MUL)	REAL	44 byte per l'operazione MUL
/ (DIV)	REAL	44 byte per l'operazione DIV
		164 byte totali

:

<b>Categoria:</b>	<b>Mnemonico:</b>	<b>Titolo:</b>	<b>Memoria per DINT (byte):</b>	<b>Memoria per REAL (byte):</b>
Ramo e diramazione		Ramo	16	
		Inizio ramo	4	
		Ramo successivo	4	
		Fine ramo	4	

<b>Categoria:</b>	<b>Mnemonico:</b>	<b>Titolo:</b>	<b>Memoria per DINT (byte):</b>	<b>Memoria per REAL (byte):</b>
Bit	XIC	Esamina se chiuso	4	
	XIO	Esamina se aperto	4	
	OTE	Eccitazione uscita	4	
	OTL	Aggancio uscita	4	
	OTU	Sgancio uscita	4	
	ONS	One Shot	36	
	OSR	One Shot sul fronte di salita	44	
	OSF	One shot sul fronte di discesa	44	
Timer e contatore	TON	Timer ritardato all'eccitazione	8	
	TOF	Timer ritardato alla diseccitazione	8	
	RTO	Timer ritentivo all'eccitazione	8	
	CTU	Contatore incrementale	8	
	CTD	Contatore decrementale	8	
	RES	Ripristina	8	
Ingresso e Uscita	MSG	Messaggio	36	
	GSV	Ottieni valore di sistema	76	76
	SSV	Imposta valore di sistema	76	76
Confronta	CMP	Confronta	76 + espressione	76 + espressione
	EQU	Uguale	20	20
	GEQ	Maggiore di o uguale a	20	36
	GRT	Maggiore di	20	36
	LEQ	Minore di o uguale a	20	36
	LES	Minore di	20	36
	LIM	Limite	52	44
	MEQ	Maschera uguale a	32	
	NEQ	Diverso da	20	20

<b>Categoria:</b>	<b>Mnemonico:</b>	<b>Titolo:</b>	<b>Memoria per DINT (byte):</b>	<b>Memoria per REAL (byte):</b>
Calcola	CPT	Calcola	76 + espressione	96 + espressione
	ADD	Somma	28	44
	SUB	Sottrai	28	44
	MUL	Moltiplica	44	44
	DIV	Dividi	44	44
	MOD	Modulo	44	44
	SQR	Radice quadrata	40	40
	NEG	Nega	28	56
	ABS	valore assoluto	32	56
Sposta/Logica	MOV	Sposta	24	48
	MVM	Spostamento con maschera	80	
	BTD	Distribuisci campo bit	52	
	CLR	Azzera	20	20
	AND	AND di bit	28	
	OR	OR di bit	28	
	XOR	OR esclusivo di bit	28	
	NOT	NOT di bit	28	
Matrice (File)/ Varie	FAL	Aritmetica e logica di file	92 + espressione	116 + espressione
	FSC	Ricerca e confronto di file	148 + espressione	152 + espressione
	COP	Copia	64	64
	FLL	Riempi	60	60
	AVE	Media	164	116
	SRT	Ordina	108	108
	STD	Deviazione standard	104	104
Scorrimento file	BSL	Scorrimento bit a sinistra	52	
	BSR	Scorrimento bit a destra	52	
	FFL	Carica FIFO	64	64
	FFU	Scarica FIFO	64	64
	LFL	Carica LIFO	64	64
	LFU	Scarica LIFO	64	64
Sequenziatore	SQI	Sequenziatore di ingresso	48	
	SQL	Carica sequenziatore	48	
	SQO	Sequenziatore di uscita	44	

<b>Categoria:</b>	<b>Mnemonico:</b>	<b>Titolo:</b>	<b>Memoria per DINT (byte):</b>	<b>Memoria per REAL (byte):</b>
Controllo programma	JMP	Salta ad etichetta	24	
	LBL	Etichetta	12	
	JSR	Salta a subroutine	56	
	SBR	Subroutine	40	
	RET	Ritorno	40	
	AFI	Sempre falso	4	
	MCR	Relè controllo principale	4	
	NOP	Nessuna operazione	4	
	TND	Fine temporanea	12	
	UID	Interrupt utente disabilitato	28	
	UIE	Interrupt utente abilitato	28	
For/Break	FOR	For	64	
	BRK	Interruzione	36	
Speciale	FBC	Confronto bit di file	72	
	DDT	Rilevamento diagnostica	72	
	DTR	Transizioni dati	40	
	PID	Proporzionale, Integrale, Derivativo	228	92
Trigonometria	SIN	Seno	140	48
	COS	Coseno	140	48
	TAN	Tangente	140	48
	ASN	Arcoseno	140	48
	ACS	Arcocoseno	140	48
	ATN	Arcotangente	140	48
Matematica avanzata	LN	Logaritmo naturale	140	48
	LOG	Logaritmo in base 10	140	48
	XPY	X elevato Y	144	52
Conversione matematica	DEG	Da radianti in gradi	144	52
	RAD	Da gradi in radianti	144	52
	TOD	Da interi in BCD	40	
	FRD	Da BCD in interi	40	
	TRN	tronca	40	92

**Indici di matrice**

L'utilizzo della memoria di un'istruzione aumenta per ogni parametro che fa riferimento ad una matrice. Considerate quanto segue:

- Quando una matrice utilizza dei valori immediati per l'indice, non viene aggiunta ulteriore memoria.

Per esempio:

<b>Esempio:</b>	<b>Memoria aggiuntiva utilizzata:</b>
myTag[0]	nessuna
myTag[0,0,0]	nessuna

- Quando una matrice utilizza una tag (tipo DINT) come indice, viene utilizzata ulteriore memoria a seconda del numero di dimensioni della matrice.

Per esempio:

<b>Esempio:</b>	<b>Memoria aggiuntiva utilizzata:</b>
myTag[offset]	84 byte
myTag[0,offset,0]	152 byte

- Il controllore utilizza un indice a 32 bit per tutte le matrici. Se, per fare riferimento ad un elemento di una matrice, si utilizza una tag diversa da DINT, aggiungere la quantità di memoria per la conversione dei dati (vedere pagina D-2) e quella per l'uso della tag come indice.

Per esempio:

TagA	DINT
TagB	DINT
TagC	INT
TableA	DINT[10]
TableB	DINT[2,4,6]

<b>Esempio:</b>		<b>Memoria aggiuntiva utilizzata:</b>
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	nessuna ulteriore aggiunta
Destinazione	TagB	24 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	nessuna ulteriore aggiunta
Destinazione	TableA[0]	24 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	84 byte per indice di tag DINT (TabellaA)
Destinazione	TableA[TagB]	108 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	60 byte per la conversione da INT in DINT (TagC)
Destinazione	TableA[TagC]	84 byte per indice di tag DINT (TabellaA)
		168 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	nessuna ulteriore aggiunta
Destinazione	TableB[0,0,0]	24 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	152 byte per indice di tag DINT (TabellaB)
Destinazione	TableB[TagB,0,0]	176 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	60 byte per la conversione da INT in DINT (TagC)
Destinazione	TableB[TagC,0,0]	152 byte per indice di tag DINT (TabellaB)
		236 byte totali

- Se nei suddetti esempi si cambiano i tipi di dati utilizzati per l'origine e la destinazione, l'utilizzo della memoria aumenterà.

Per esempio:

TagA	DINT
TagC	INT
TagD	SINT
TableC	INT[10]
TableD	SINT[2,4,6]

<b>Esempio:</b>		<b>Memoria aggiuntiva utilizzata:</b>
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	52 byte per la conversione da DINT in INT (TagA in TabellaC)
Destinazione	TableC[TagC]	60 byte per la conversione da INT in DINT (TagC)
		84 byte per indice di tag DINT (TabellaC)
		220 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagD	92 byte per la conversione da SINT in INT (TagD in TabellaC)
Destinazione	TableC[TagC]	60 byte per la conversione da INT in DINT (TagC)
		84 byte per indice di tag DINT (TabellaC)
		260 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagA	52 byte per la conversione da DINT in SINT (TagA in TabellaD)
Destinazione	TableD[TagC,0,0]	60 byte per la conversione da INT in DINT (TagC)
		152 byte per indice di tag DINT (TabellaD)
		288 byte totali
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TagC	104 byte per la conversione da INT in SINT (TagC in TabellaC)
Destinazione	TableD[TagC,0,0]	60 byte per la conversione da INT in DINT (TagC)
		152 byte per indice di tag DINT (TabellaD)
		340 byte totali

- Se nell'indice viene utilizzata un'espressione, viene utilizzata memoria adizionale dipendente dagli operatori.

Per esempio:

```

TagA      DINT
TagB      DINT
TagF      DINT
TableA    DINT[10]
    
```

<b>Esempio:</b>	<b>Memoria aggiuntiva utilizzata:</b>
Istruzione MOV	24 byte per l'istruzione MOV
Origine TagA	84 byte per indice di tag DINT (TabellaA)
Destinazione TableA[TagB + TagF]	28 byte per l'istruzione ADD
	136 byte totali

- Se in un'istruzione vengono utilizzate più matrici, aggiungere l'utilizzo di memoria appropriato per ciascun riferimento di matrice.

Per esempio:

```

TagA      DINT
TagB      DINT
TableA    DINT[10]
    
```

<b>Esempio:</b>	<b>Memoria aggiuntiva utilizzata:</b>
Istruzione MOV	24 byte per l'istruzione MOV
Origine TableA[TagA]	84 byte per indice di tag DINT (TabellaA)
Destinazione TableA[TagB]	84 byte per il secondo indice di tag DINT (TabellaA)
	192 byte totali

- La situazione peggiore si verifica quando si hanno più riferimenti a matrici non DINT, con più dimensioni e con indici non DINT.

Per esempio:

TagC	INT
TagD	SINT
TagE	INT
TableD	SINT[2,4,6]
TableE	INT[2,4,6]

**Esempio:****Memoria aggiuntiva utilizzata:**

Istruzione MOV		24 byte per l'istruzione MOV
Origine	TableD[TagC, TagD, TagE]	60 byte per la conversione da INT in DINT (TagC)
Destinazione	TableE[TagC, TagD, TagE]	48 byte per la conversione da SINT in DINT (TagD)
		60 byte per la conversione da INT in DINT (TagE)
		152 byte per indice di tag DINT (TabellaD)
		60 byte per la conversione da INT in DINT (TagC)
		48 byte per la conversione da SINT in DINT (TagD)
		60 byte per la conversione da INT in DINT (TagE)
		152 byte per indice di tag DINT (TabellaE)
		92 byte per la conversione da SINT in INT (da TabellaD in TabellaE)
		756 byte totali

- È possibile ottimizzare questa istruzione rendendo entrambe le matrici di tipo DINT con indici con tag DINT. Per esempio:

TagA	DINT
TagB	DINT
TagF	DINT
TableB	DINT[2,4,6]
TableF	DINT[2,4,6]

**Esempio:****Memoria aggiuntiva utilizzata:**

Istruzione MOV		24 byte per l'istruzione MOV
Origine	TableB[TagA, TagB, TagF]	152 byte per indice di tag DINT (TabellaB)
Destinazione	TableF[TagA, TagB, TagF]	152 byte per indice di tag DINT (TabellaE)
		328 byte totali

- È possibile ottimizzare ulteriormente questa istruzione cambiando tutti gli indici della matrice in valori immediati. Per esempio:

TagA	DINT
TagB	DINT
TagF	DINT
TableB	DINT[2,4,6]
TableF	DINT[2,4,6]

<b>Esempio:</b>		<b>Memoria aggiuntiva utilizzata:</b>
Istruzione MOV		24 byte per l'istruzione MOV
Origine	TableB[0,1,2]	nessuna ulteriore aggiunta
Destinazione	TableF[3,4,5]	24 byte totali

---

## A

- abilita interrupt utente 10-14
- aggancia uscita 1-7
- allocazione di memoria
  - matrice B-6
- arccoseno 13-10
- arcoseno 13-8
- arcotangente 13-12
- aritmetica e logica di file 7-6
- attributi
  - conversione di tipi di dati A-6
  - parole chiave A-4
  - parole chiave di stato
    - aritmetico A-1
  - tipi di dati A-5
- attributi comuni
  - conversione di tipi di dati A-6
  - parole chiave A-4
  - parole chiave di stato
    - aritmetico A-1
  - tipi di dati A-5
- avvisi 12-25
- azzera 6-8

## B

- banda morta 12-34
- bit all'interno di matrici B-4
- break 11-5

## C

- calcola 5-2
- calcolo parametri 12-24
- carica FIFO 8-8
- carica LIFO 8-20
- carica sequenziatore 9-11
- codici di errore
  - istruzione MSG 3-7
- combinazione di tipi di dati A-6
- compensazione anticipata 12-35
- concetti di matrice
  - allocazione di memoria B-6
  - indicizzazione B-3
  - specificazione di bit B-4
  - utilizzo della memoria D-7

- configurazione 3-13
  - istruzione MSG 3-13
  - istruzione PID 12-24
- confronto 4-2
- confronto bit di file 12-2
- conteggio decrementale 2-14
- conteggio incrementale 2-11
- conversione da BCD 15-6
- conversione di tipi di dati A-6
- conversione in BCD 15-4
- conversione in scala 12-26
- conversioni dei dati D-1
- copia 7-27
- coseno 13-4

## D

- da gradi a radianti 15-3
- da radianti a gradi 15-2
- deviazione standard 7-41
- di bit
  - AND 6-9
  - NOT 6-15
  - OR 6-11
  - XOR 6-13
- disabilita interrupt utente 10-13
- distribuisci campo bit 6-5
- diverso da 4-22
- dividi 5-12

## E

- eccita uscita 1-6
- elevamento a potenza 14-6
- esamina se aperto 1-4
- esamina se chiuso 1-2
- espressioni
  - formato 4-4, 5-4, 7-15, 7-25
  - operatori validi 4-4, 5-4, 7-15, 7-25
  - ordine delle operazioni 4-5, 5-5, 7-16, 7-26
- etichetta 10-2

## F

- fine temporanea 10-10

- I
- imposta valore di sistema 3-27
- indicizzazione B-3
- istruzione ABS 5-19
- istruzione ACS 13-10
- istruzione ADD 5-6
- istruzione AFI 10-15
- istruzione AND 6-9
- istruzione ASN 13-8
- istruzione ATN 13-12
- istruzione AVE 7-34
- istruzione BRK 11-5
- istruzione BSL 8-2
- istruzione BSR 8-5
- istruzione BTD 6-5
- istruzione CLR 6-8
- istruzione CMP 4-2
- istruzione COP 7-27
- istruzione COS 13-4
- istruzione CPT 5-2
- istruzione CTD 2-14
- istruzione CTU 2-11
- istruzione DDT
  - modalità di ricerca 12-11
  - operandi 12-9
- istruzione DEG 15-2
- istruzione di calcolo
  - ADD 5-6
  - SUB 5-8
- istruzione DIV 5-12
- istruzione DTR 12-16
- istruzione EQU 4-6
- istruzione FAL
  - modalità operativa 7-1
  - operandi 7-6
- istruzione FBC
  - modalità di ricerca 12-4
  - operandi 12-2
- istruzione FFL 8-8
- istruzione FFU 8-14
- istruzione FLL 7-31
- istruzione FOR 11-2
- istruzione FRD 15-6
- istruzione FSC
  - modalità operativa 7-1
  - operandi 7-17
- istruzione GEQ 4-8
- istruzione GRT 4-10
- istruzione GSV
  - oggetti 3-29
  - operandi 3-27
- istruzione JMP 10-2
- istruzione JSR 10-4
- istruzione LBL 10-2
- istruzione LEQ 4-12
- istruzione LES 4-14
- istruzione LFL 8-20
- istruzione LFU 8-26
- istruzione LIM 4-16
- istruzione LN 14-2
- istruzione LOG 14-4
- istruzione MCR 10-11
- istruzione MEQ 4-19
- istruzione MOD 5-14
- istruzione MOV 6-2
- istruzione MSG 3-13
  - codici di errore 3-7
  - connessione con cache 3-26
  - metodo di comunicazione 3-25
  - operandi 3-2
  - struttura 3-2
- istruzione MUL 5-10
- istruzione MVM 6-3
- istruzione NEG 5-18
- istruzione NEQ 4-22
- istruzione NOP 10-16
- istruzione NOT 6-15
- istruzione ONS 1-9
- istruzione OR 6-11
- istruzione OSF 1-14
- istruzione OSR 1-11
- istruzione OTE 1-6
- istruzione OTL 1-7
- istruzione OTU 1-8
- istruzione PID
  - banda morta 12-34
  - calcolo parametri 12-24
  - compensazione anticipata 12-35
  - configurazione 12-24
  - conversione in scala 12-26
  - operandi 12-19
  - polarizzazione uscita 12-35
  - struttura 12-20
- istruzione RAD 15-3
- istruzione RES 2-18
- istruzione RET 10-4, 11-6
- istruzione RTO 2-8
- istruzione SBR 10-4

- 
- istruzione SIN 13-2
  - istruzione SQI 9-2
  - istruzione SQL 9-11
  - istruzione SQO 9-6
  - istruzione SQR 5-16
  - istruzione SRT 7-38
  - istruzione SSV
    - oggetti 3-29
    - operandi 3-27
  - istruzione SUB 5-8
  - istruzione TAN 13-6
  - istruzione TND 10-10
  - istruzione TOD 15-4
  - istruzione TOF 2-5
  - istruzione TON 2-2
  - istruzione TRN 15-8
  - istruzione UID 10-13
  - istruzione UIE 10-14
  - istruzione XIC 1-2
  - istruzione XIO 1-4
  - istruzione XOR 6-13
  - istruzione XPY 14-6
  - istruzioni
    - attributi comuni A-1
    - bit 1-1
    - confronto 4-1
    - contatore 2-1
    - controllo programma 10-1
    - conversione 15-1
    - conversione matematica 15-1
    - di calcolo 5-1
    - di matrici
    - for/break 11-1
    - input/output 3-1
    - logica 6-1
    - matematica avanzata 14-1
    - scorrimento 8-1
    - sequenziatore 9-1
    - speciali 12-1
    - spostamento 6-1
    - tempo d'esecuzione C-1
    - timer 2-1
    - trigonometriche 13-1
    - utilizzo della memoria D-3
  - istruzioni di bit
    - introduzione 1-1
    - ONS 1-9
    - OSF 1-14
    - OSR 1-11
  - OTE 1-6
  - OTL 1-7
  - OTU 1-8
  - XIC 1-2
  - XIO 1-4
  - istruzioni di calcolo
    - ABS 5-19
    - CPT 5-2
    - DIV 5-12
    - formato dell'espressione 5-4, 7-15
    - introduzione 5-1
    - MOD 5-14
    - MUL 5-10
    - NEG 5-18
    - operatori validi 5-4, 7-15
    - ordine delle operazioni 5-5, 7-16
    - SQR 5-16
  - istruzioni di confronto
    - CMP 4-2
    - EQU 4-6
    - formato dell'espressione 4-4, 7-25
    - GEQ 4-8
    - GRT 4-10
    - introduzione 4-1
    - LEQ 4-12
    - LES 4-14
    - LIM 4-16
    - MEQ 4-19
    - NEQ 4-22
    - operatori validi 4-4, 7-25
    - ordine delle operazioni 4-5, 7-26
  - istruzioni di contatore
    - CTD 2-14
    - CTU 2-11
    - introduzione 2-1
    - RES 2-18
  - istruzioni di controllo programma
    - AFI 10-15
    - introduzione 10-1
    - JMP 10-2
    - JSR 10-4
    - LBL 10-2
    - MCR 10-11
    - NOP 10-16
    - RET 10-4
    - SBR 10-4
    - TND 10-10
    - UID 10-13
    - UIE 10-14

- istruzioni di conversione
  - DEG 15-2
  - FRD 15-6
  - introduzione 15-1
  - RAD 15-3
  - TOD 15-4
  - TRN 15-8
- istruzioni di conversione matematica
  - DEG 15-2
  - FRD 15-6
  - introduzione 15-1
  - RAD 15-3
  - TOD 15-4
  - TRN 15-8
- istruzioni di file. Vedere istruzioni di matrici
- istruzioni di input/output
  - GSV 3-27
  - introduzione 3-1
  - MSG 3-2
  - SSV 3-27
- istruzioni di logica
  - AND 6-9
  - introduzione 6-1
  - NOT 6-15
  - OR 6-11
  - XOR 6-13
- istruzioni di matematica avanzata
  - introduzione 14-1
  - LN 14-2
  - LOG 14-4
  - XPY 14-6
- istruzioni di matrice
  - AVE 7-34
  - BSL 8-2
  - BSR 8-5
  - COP 7-27
  - DDT 12-9
  - FAL 7-6
  - FBC 12-2
  - FFL 8-8
  - FFU 8-14
  - FLL 7-31
  - FSC 7-17
  - LFL 8-20
  - LFU 8-26
  - modalità operativa 7-1
  - RES 2-18
  - scorrimento 8-1
  - sequenziatore 9-1
  - SQI 9-2
  - SQL 9-11
  - SQO 9-6
  - SRT 7-38
  - STD 7-41
- istruzioni di matrici
  - concetti sulle matrici B-1
  - file/varie 7-1
- istruzioni di scorrimento
  - BSL 8-2
  - BSR 8-5
  - FFL 8-8
  - FFU 8-14
  - introduzione 8-1
  - LFL 8-20
  - LFU 8-26
- istruzioni di spostamento
  - BTD 6-5
  - CLR 6-8
  - introduzione 6-1
  - MOV 6-2
  - MVM 6-3
- istruzioni di timer
  - introduzione 2-1
  - RES 2-18
  - RTO 2-8
  - TOF 2-5
  - TON 2-2
- istruzioni PID
  - avvisi 12-25
- istruzioni sequenziatore
  - introduzione 9-1
  - SQI 9-2
  - SQL 9-11
  - SQO 9-6
- istruzioni speciali
  - DDT 12-9
  - DTR 12-16
  - FBC 12-2
  - introduzione 12-1
  - PID 12-19
- istruzioni STD 7-41
- istruzioni trigonometriche
  - ACS 13-10
  - ASN 13-8
  - ATN 13-12
  - COS 13-4
  - introduzione 13-1
  - SIN 13-2
  - TAN 13-6

**L**

limite 4-16  
 logaritmo in base 10 14-4  
 logaritmo naturale 14-2

**M**

maggiore di 4-10  
 maggiore di o uguale a 4-8  
 maschera uguale a 4-19  
 maschere 4-19, 6-3, 12-16  
 media 7-34  
 messaggio 3-2  
 minore di 4-14  
 minore di o uguale a 4-12  
 modalità All (Tutti) 7-2  
 modalità di ricerca 12-4, 12-11  
 modalità incrementale 7-4  
 modalità numerica 7-3  
 modalità operativa 7-1  
 modulo 5-14  
 moltiplica 5-10

**N**

nega 5-18  
 nessuna operazione 10-16

**O**

oggetti  
 AXIS 3-30  
 CONTROLLER 3-37  
 CONTROLLERDEVICE 3-37  
 CST 3-39  
 DF1 3-40  
 FAULTLOG 3-43  
 istruzione GSV/SSV 3-29  
 MESSAGE 3-43  
 MODULE 3-45  
 MOTIONGROUP 3-46  
 PROGRAM 3-47  
 ROUTINE 3-47  
 SERIALPORT 3-48  
 WALLCLOCKTIME 3-50  
 oggetto  
 TASK 3-49

oggetto AXIS 3-30  
 oggetto CONTROLLER 3-37  
 oggetto CONTROLLERDEVICE  
 3-37  
 oggetto CST 3-39  
 oggetto DF1 3-40  
 oggetto FAULTLOG 3-43  
 oggetto MESSAGE 3-43  
 oggetto MODULE 3-45  
 oggetto MOTIONGROUP 3-46  
 oggetto PROGRAM 3-47  
 oggetto ROUTINE 3-47  
 oggetto SERIALPORT 3-48  
 oggetto TASK 3-49  
 oggetto WALLCLOCKTIME 3-50  
 one shot 1-9  
 one shot sul fronte di discesa 1-14  
 one shot sul fronte di salita 1-11  
 operatori 4-4, 5-4, 7-15, 7-25  
 operators 4-4, 7-25  
 OR esclusivo 6-13  
 ordina 7-38  
 ordine delle operazioni 4-5, 5-5, 7-16,  
 7-26  
 ottieni valore di sistema 3-27

**P**

parole chiave  
 altre A-4  
 stato aritmetico A-1  
 parole chiave di stato aritmetico A-1  
 polarizzazione uscita 12-35  
 proporzionale, integrale e derivativo  
 12-19

**R**

radice quadrata 5-16  
 relè di controllo principale 10-11  
 ricerca e confronto file 7-17  
 riempi file 7-31  
 riempimento con zeri A-6  
 rileva diagnostica 12-9  
 ripristino 2-18  
 ritorno 10-4

- 
- S**
- salta 10-2
  - salta alla subroutine 10-4
  - scarica FIFO 8-14
  - scarica LIFO 8-26
  - sempre falso 10-15
  - seno 13-2
  - sequenziatore di ingresso 9-2
  - sequenziatore di uscita 9-6
  - sgancia uscita 1-8
  - somma 5-6
  - sottrai 5-8
  - sposta bit a destra 8-5
  - sposta bit a sinistra 8-2
  - sposta con maschera 6-3
  - struttura CONTROL 7-6, 7-17, 7-34, 7-38, 7-41, 8-2, 8-5, 8-8, 8-14, 8-20, 8-26, 9-2, 9-6, 9-11, 12-3, 12-10
  - struttura COUNTER 2-11, 2-14
  - struttura MESSAGE 3-2
  - struttura TIMER 2-2, 2-5, 2-8
  - strutture
    - CONTROL 7-6, 7-17, 7-34, 7-38, 7-41, 8-2, 8-5, 8-8, 8-14, 8-20, 8-26, 9-2, 9-6, 9-11, 12-3, 12-10
    - COUNTER 2-11, 2-14
  - istruzione RES 2-18
  - MSG 3-2
  - PID 12-20
  - TIMER 2-2, 2-5, 2-8
  - subroutine 10-4
- T**
- tangente 13-6
  - tempo
    - di esecuzione C-1
    - tempo d'esecuzione C-1
  - timer ritardato all'eccitazione 2-2
  - timer ritardato alla diseccitazione 2-5
  - timer zitartrato
    - all'eccitazione ritentivo 2-8
  - tipi di dati A-5
  - transizione dati 12-16
  - troncamento di un valore 15-8
- U**
- uguale a 4-6
  - utilizzo della memoria C-1, D-1
- V**
- valore assoluto 5-19
- X**
- X alla potenza di Y 14-6

**Notes:**

---

**Reach us now at [www.rockwellautomation.com](http://www.rockwellautomation.com)**

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

**Americas Headquarters**, 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444

**European Headquarters SA/NV**, avenue Herrmann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40

**Asia Pacific Headquarters**, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846



**Rockwell  
Automation**