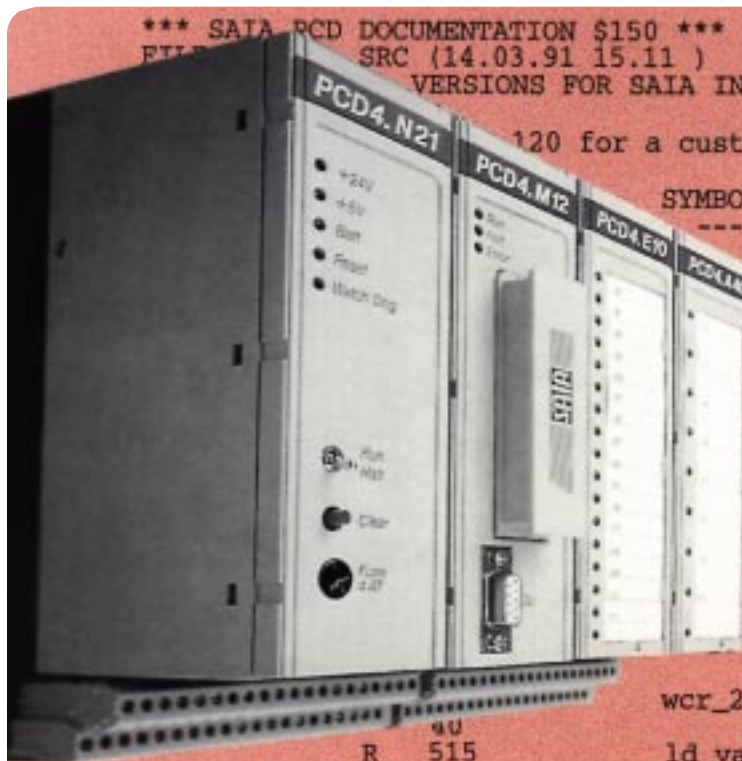


SAIA® PCD Process Control Devices

Manuale delle istruzioni



```

*** SAIA PCD DOCUMENTATION $150 ***                               PAGE 2
FILE SRC (14.03.91 15.11 )                                         PRODUCED: 14.03.91 15.12
VERSIONS FOR SAIA INTERNAL USE ONLY >>>

120 for a customer in Lyon

SYMBOL      COMMENT
-----
;init syst 1 and syst 2

; aux. register for indexing
; reset

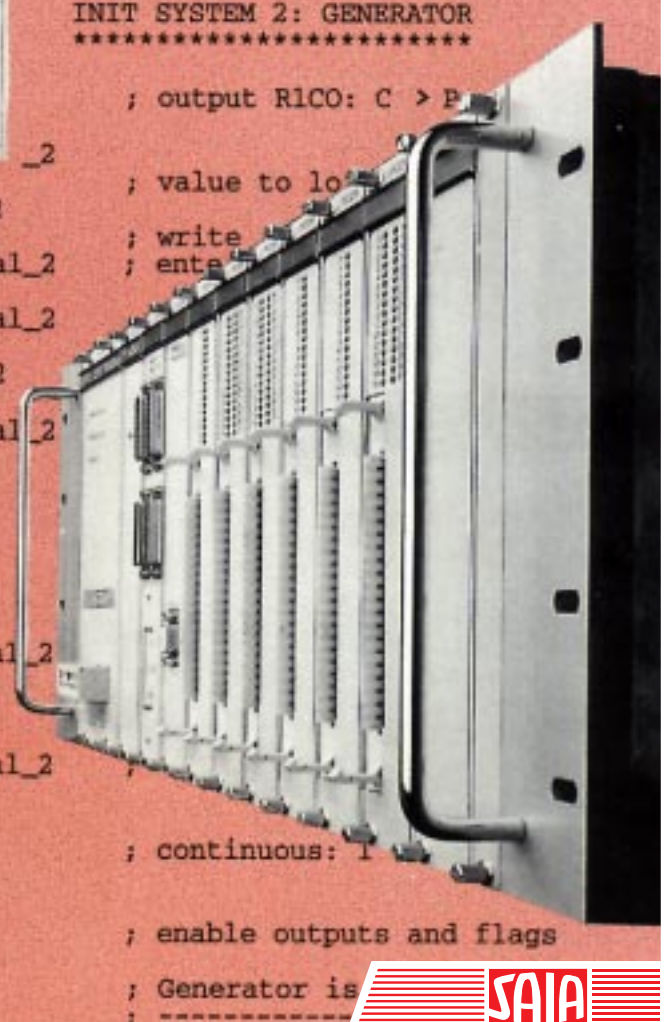
INIT SYSTEM 2: GENERATOR
*****

; output R1CO: C > P

; value to lo
wcr_2
; write
ld_val_2 ; ente

22      LDL      R      515      ld_val_2
23      2
24      CFB      812      wcr_2
25      36
26      R      515      ld_val_2
27      CFB      810      wt_2
28      211
29      CFB      810      wf_2
30      165
31      LDL      R      515      ld_val_2
32      49
33      CFB      811      wd_2
34      64
35      R      515      ld_val_2
36      CFB      810      wf_2
37      188
; continuous: 1
38      CFB      810      wf_2
39      106
; enable outputs and flags
; Generator is
; -----

```



SAIA® Process Control Devices

Manuale delle istruzioni

Edizione 26/733 I7 02.99

SAIA-Burgess Electronics SA 1994-99. Tutti i diritti riservati

Soggetto a modifiche tecniche

Indice

| | Pagina |
|--|--------|
| 1. Introduzione | |
| 1.1 Codice Supporto (MC) - MEDIUM Code | 1-3 |
| 1.2 Costanti | 1-5 |
| 1.3 I Codici Condizione [cc] | 1-6 |
| 1.4 Tipi e Valori delle Risorse | 1-7 |
| | |
| 2. Istruzioni BIT | |
| STH Inizio su Stato Alto (Start High) | 2-3 |
| STL Inizio su Stato Basso (Start Low) | 2-4 |
| ANH AND su Stato Alto (AND High) | 2-5 |
| ANL AND su Stato Basso (AND Low) | 2-6 |
| ORH OR su Stato Alto (OR High) | 2-7 |
| ORL OR su Stato Basso (Low) | 2-9 |
| XOR OR Esclusivo (Exclusive OR) | 2-10 |
| ACC Operazioni sull'Accumulatore (ACCU operations) | 2-11 |
| DYN Rilevazione Dinamica Fronti (Dynamic edge detection) | 2-12 |
| OUT Imposta Elemento da Accumulatore (Set Element from Accumulator) | 2-13 |
| SET Imposta Elemento (Set Element) | 2-14 |
| RES Reset Elemento (Reset Element) | 2-15 |
| COM Complemento dell'Elemento (Complement Element) | 2-16 |
| SETD Set Elemento con Ritardo (Set Element Delayed) | 2-17 |
| RESD Reset Elemento con Ritardo (Reset Element Delayed) | 2-18 |

| | | Pagina |
|---------------------------|--|--------|
| 3. Istruzioni WORD | | |
| LD | Carica (Valore a 32 bit) (Load - 32 bit value) | 3-3 |
| LDL | Carica Parola Meno Significativa (primi 16 bit) (Load Low Word - lower 16 bits) | 3-4 |
| LDH | Carica Parola Più Significativa (ultimi 16 bit) (Load High Word - higher 16 bits) | 3-5 |
| DSP | Carica Registro Display (Load Display Register) | 3-6 |
| INC | Incrementa un Registro o un Contatore (Increment Register or Counter) | 3-7 |
| DEC | Decrementa un Registro o un Contatore (Decrement Register or Counter) | 3-8 |
| SEI | Imposta il Registro Indice (Set Index Register) | 3-9 |
| INI | Incrementa il Registro indice (Increment Index Register) | 3-10 |
| DEI | Decrementa il Registro indice (Decrement Index Register) | 3-11 |
| STI | Memorizza il Registro Indice (Store Index Register) | 3-12 |
| RSI | Ripristina il Registro Indice (Restore Index Register) | 3-13 |
| MOV | Spostamento Dati (Move Data) | 3-14 |
| COPY | Copia Dati (Copy Data) | 3-15 |
| GET | Lettura Dati (Get Data) | 3-16 |
| PUT | Scrittura Dati (Put Data) | 3-19 |
| TFR | Trasferimento Dati (Transfer Data) | 3-21 |
| TFRI | Trasferimento Dati Indiretto (Transfer Data Indirect) | 3-23 |
| BITI | Lettura Bit (Bit In) | 3-25 |
| BITIR | Lettura Bit Reverse (Bit In Reversed) | 3-26 |
| BITO | Uscita Bit (Bit Out) | 3-27 |
| BITOR | Uscita Bit Reverse (Bit Out Reversed) | 3-28 |
| DIGI | Lettura Digit (Digit In) | 3-29 |
| DIGIR | Lettura Digit Reverse (Digit In Reversed) | 3-30 |
| DIGO | Uscita Digit (Digit Out) | 3-31 |
| DIGOR | Uscita Digit Reverse (Digit Out Reversed) | 3-32 |
| AND | AND Tra Registri (AND Registers) | 3-33 |
| OR | OR tra Registri (OR Registers) | 3-34 |
| EXOR | OR Esclusivo tra Registri (Exclusive OR Registers) | 3-35 |
| NOT | Complemento Registro (Complement Register) | 3-36 |
| SHIU | Shift Avanti di Registri (Shift Registers Up) | 3-37 |
| SHID | Shift Indietro di Registri (Shift Registers Down) | 3-38 |
| ROTU | Rotazione Avanti di Registri (Rotate Registers Up) | 3-39 |
| ROTD | Rotazione Indietro di Registri (Rotate Registers Down) | 3-40 |
| SHIL | Shift a Sinistra di un Registro (Shift Register Left) | 3-41 |
| SHIR | Shift a Destra di un Registro (Shift Register Right) | 3-42 |
| ROTL | Rotazione a Sinistra di un Registro (Rotate Register Left) | 3-43 |
| ROTR | Rotazione a Destra di un Registro (Rotate Register Right) | 3-44 |

4. Istruzioni Aritmetiche in INTERO

| | | |
|-----|--|-----|
| ADD | Somma di Registri (ADD Registers) | 4-3 |
| SUB | Sottrazione di Registri (SUBtract Registers) | 4-4 |
| MUL | Moltiplicazione tra Registri (MULtipliy Registers) | 4-5 |
| DIV | Divisione tra Registri (DIVide Registers) | 4-6 |
| SQR | Radice Quadrata (SQure Root) | 4-7 |
| CMP | Confronto tra Registri (CoMPare Registers) | 4-8 |

5. Istruzioni Aritmetiche in VIRGOLA MOBILE

| | | |
|-------|---|------|
| IFP | Conversione da Intero a Virgola Mobile (Integer to Floating Point) | 5-3 |
| FPI | Conversione da Virgola Mobile a Intero (Floating Point to Integer) | 5-4 |
| FADD | Somma di Numeri in Virgola Mobile (Floating point ADDition) | 5-5 |
| FSUB | Sottrazione tra Numeri in Virgola Mobile (Floating point SUBtraction) | 5-6 |
| FMUL | Moltiplicazione tra Numeri in Virgola Mobile (Floating point MULtiplication) | 5-7 |
| FDIV | Divisione tra Numeri in Virgola Mobile (Floating point DIVision) | 5-8 |
| FSQR | Radice Quadrata di Numeri in Virgola Mobile (Floating point SQure Root) | 5-9 |
| FCMP | Confronto tra Numeri in Virgola Mobile (Floating point CoMPare) | 5-10 |
| FSIN | Funzione Seno (SINE function) | 5-11 |
| FCOS | Funzione Coseno (COSine function) | 5-12 |
| FATAN | Arcotangente (Arc TANgent) | 5-13 |
| FEXP | Funzione Esponenziale (EXPonential function) | 5-14 |
| FLN | Funzione Logaritmo in Virgola Mobile (Floating Point Logarithm function) | 5-15 |
| FABS | Valore Assoluto in Virgola Mobile (Floating point ABSolute value) | 5-16 |

6. Istruzioni BLOCTEC

| | | |
|------|--|------|
| COB | Blocco ad Organizzazione Ciclica (Cyclic Organisation Block) | 6-3 |
| ECOB | Fine Blocco ad Organizzazione Ciclica (End of Cyclic Organisation Block) | 6-4 |
| XOB | Blocco ad Organizzazione Esclusiva (Exception Organisation Block) | 6-5 |
| EXOB | Fine Blocco ad Organizzazione Esclusiva (End of Exception Organisation Block) | 6-9 |
| PB | Blocco di Programma (Program Block) | 6-10 |
| EPB | Fine del Blocco di Programma (End of Program Block) | 6-11 |
| CPB | Chiamata Blocco di Programma (Call Program Block) | 6-12 |
| CPBI | Chiamata Indiretta Blocco di Programma (Call Program Block Indirect) | 6-13 |
| FB | Blocco Funzione (Function Block) | 6-14 |
| EFB | Fine Blocco Funzione (End of Function Block) | 6-15 |
| CFB | Chiamata Blocco Funzione (Call Function Block) | 6-16 |
| NCOB | Prossimo COB (Next Cyclic Organisation Block) | 6-17 |
| SCOB | Stop COB (Stop Cyclic Organisation Block) | 6-18 |
| CCOB | Continua COB (Continue Cyclic Organisation Block) | 6-19 |
| RCOB | Ripartenza COB (Restart Cyclic Organisation Block) | 6-20 |

7. Istruzioni GRAFTEC

| | | |
|-----|---|------|
| SB | Blocco Sequenziale (Sequential Block) | 7-3 |
| ESB | Fine del Blocco Sequenziale (End Sequential Block) | 7-4 |
| CSB | Chiamata Blocco Sequenziale (Call Sequential Block) | 7-5 |
| RSB | Ripartenza Blocco Sequenziale (Restart Sequential Block) | 7-6 |
| IST | Step Iniziale (Initial SStep) | 7-7 |
| ST | SStep | 7-8 |
| EST | Fine Step (End of Step) | 7-9 |
| TR | Transizione (TRansition) | 7-10 |
| ETR | Fine Transizione (End of TRansition) | 7-11 |

8. Istruzioni di COMUNICAZIONE SERIALE

| | |
|--|------|
| Modalità C (Mode C) | 8-3 |
| Modalità D (Mode D) | 8-4 |
| Modalità MM4 (Mode MM4) | 8-5 |
| Modalità S-Bus (Mode S-Bus) | 8-6 |
| PROFIBUS | 8-7 |
| SASI Assegna l'Interfaccia Seriale (Assign Serial Interface) | 8-8 |
| Testi SASI (SASI Texts) | 8-9 |
| Modalità OFF (Mode OFF) | 8-9 |
| <uart_def> | 8-10 |
| <mode_def> | 8-11 |
| <diag_def> | 8-12 |
| <rx_buf> | 8-16 |
| <tx_buf> | 8-16 |
| Esempi di Testi SASI | 8-17 |
| Utilizzo di Simboli nei Testi | 8-18 |
| \$SASI, \$ENDSASI | 8-19 |
| SASII Assegna Interfaccia Seriale Indiretta (Assign Serial Interface Indirect) | 8-20 |
| SRXD Ricezione Carattere su Linea Seriale (Modalità C) (Serial Receive Character - Mode C) | 8-21 |
| STXD Trasmissione Carattere su Linea Seriale (Modalità C) (Serial Transmit Character - Mode C) | 8-22 |
| STXT Trasmissione Testo su Linea Seriale (Modalità C) (Serial Transmit Text - Mode C) | 8-23 |
| Testi | 8-24 |
| Testi e Variabili | 8-25 |
| Formati di Output (Output formats) | 8-27 |
| Utilizzo di SIMBOLI nei Testi | 8-31 |
| SRXM Ricezione Seriale Dati (Serial Receive Media) | 8-32 |
| SRXMI Ricezione Seriale Dati Indiretta (Serial Receive Media Indirect) | 8-37 |
| STXM Trasmissione Seriale Dati (Serial Transmit Media) | 8-39 |
| STXMI Trasmissione Seriale Dati Indiretta (Serial Transmit Media Indirect) | 8-44 |
| SICL Segnali di Controllo del Canale Seriale in Ingresso (Serial Input Control Line) | 8-46 |
| SOCL Segnali di Controllo del Canale Seriale in Uscita (Serial Output Control Line) | 8-47 |
| SCON Connessione Seriale (Serial Connect) | 8-49 |
| SCONI Connessione Seriale Indiretta (Serial Connect Indirect) | 8-51 |

9. Istruzioni LAN2

| | | |
|------|--|-----|
| LRXD | Ricezione Dati via LAN 2 (Receive Data via LAN 2) | 9-3 |
| LTXD | Trasmissione Dati via LAN 2 (Transmit Data via LAN 2) | 9-4 |
| LRXS | Ricezione Stato via LAN 2 (Receive Status via LAN 2) | 9-5 |
| LTXS | Trasmissione Stato via LAN 2 (Transmit Status via LAN 2) | 9-6 |

10. Istruzioni di CONTROLLO

| | | |
|--------|--|------|
| JR | Salto Relativo (Jump Relative) | 10-3 |
| JPD | Salto Diretto (Jump Direct) | 10-4 |
| JPI | Salto Indiretto (Jump Indirect) | 10-5 |
| HALT | Arresta la CPU (Halts the cpu) | 10-6 |
| LOCK | Attiva un Semaforo (Lock semaphore) | 10-7 |
| UNLOCK | Disattiva un Semaforo (Unlock semaphore) | 10-8 |

11. Istruzioni di DEFINIZIONE

| | | |
|--------|---|------|
| DEFVM | Definizione Memoria Volatile (Flag) (Define Volatile Memory - Flags) | 11-3 |
| DEFTC | Definizione Temporizzatori/Contatori (Define Timer/Counter) | 11-4 |
| DEFTB | Definizione Base Tempi (Define Time Base) | 11-5 |
| DEFTR | Definizione Risoluzione Temporizzatori (Define Timer Resolution) | 11-6 |
| DEFWPR | Definizione Area Protetta da Scrittura (Run) (Define Write Protected Area - Run) | 11-7 |
| DEFWPH | Definizione Area Protetta da Scrittura (Halt) (Define Write Protected Area - Halt) | 11-8 |

12. Istruzioni SPECIALI

| | | |
|--------|---|-------|
| NOP | Nessuna Operazione (No Operation) | 12-3 |
| RTIME | Lettura Ora (Read TIME) | 12-4 |
| WTIME | Scrittura Ora (Write TIME) | 12-5 |
| PID | Algoritmo di Controllo PID (PID Control Algorithm) | 12-6 |
| TEST | TEST hardware | 12-10 |
| DIAG | Lettura Diagnostica (Read XOB Diagnostic) | 12-13 |
| SYSRD | Lettura Dati Sistema (System Read) | 12-14 |
| SYSWR | Scrittura Dati Sistema (System Write) | 12-17 |
| SYSCMP | Comparazione Dati Sistema (System Compare) | 12-20 |
| ALGI | Ingresso Analogico (Analogue Input) | 12-21 |
| ALGO | Uscita Analogica (Analogue Output) | 12-22 |
| STHS | Inizio Lento su Stato Alto (Start High Slow) | 12-23 |
| OUTS | Imposta Elemento da Accumulatore - Lento (Set Element form Accumulator Slow) | 12-24 |

13. Tabella Storica**14. Indice Analitico**

Elenco delle Istruzioni in Ordine Alfabetico

| Istr. | Ind | Par. | Pag. | Istr. | Ind | Par. | Pag. | Istr. | Ind | Par. | Pag. |
|--------|-----|------|-------|-------|-----|------|-------|--------|-----|------|-------|
| ACC | | | 2-11 | FPI | X | = | 5-4 | SHIL | X | = | 3-41 |
| ADD | | = | 4-3 | FSIN | X | = | 5-11 | SHIR | X | = | 3-42 |
| ALGI | X | = | 12-21 | FSQR | | = | 5-9 | SHIU | | = | 3-37 |
| ALGO | X | = | 12-22 | FSUB | | = | 5-6 | SICL | | = | 8-46 |
| AND | X | = | 3-33 | GET | X | = | 3-16 | SOCL | | = | 8-47 |
| ANH | X | = | 2-5 | HALT | | | 10-6 | SQR | | = | 4-7 |
| ANL | X | = | 2-6 | IFP | X | = | 5-3 | SRXD | X | = | 8-21 |
| BITI | X | = | 3-25 | INC | X | = | 3-7 | SRXM | | = | 8-32 |
| BITIR | X | = | 3-26 | INI | | = | 3-10 | SRXMI | | = | 8-37 |
| BITO | X | = | 3-27 | IST | | | 7-7 | ST | | | 7-8 |
| BITOR | X | = | 3-28 | JPD | | | 10-4 | STH | X | = | 2-3 |
| CCOB | | | 6-19 | JPI | | | 10-5 | STHS | X | = | 12-23 |
| CFB | | | 6-16 | JR | | | 10-3 | STI | | = | 3-12 |
| CMP | X | = | 4-8 | LD | X | | 3-3 | STL | X | = | 2-4 |
| COB | | | 6-3 | LDH | X | = | 3-5 | STXD | X | = | 8-22 |
| COM | X | = | 2-16 | LDL | X | = | 3-4 | STXM | | = | 8-39 |
| COPY | X | = | 3-15 | LOCK | | | 10-7 | STXMI | | = | 8-44 |
| CPB | | | 6-12 | LRXD | X | = | 9-3 | STXT | X | = | 8-23 |
| CPBI | | | 6-13 | LRXS | X | = | 9-5 | SUB | | = | 4-4 |
| CSB | | | 7-5 | LTXD | X | = | 9-4 | SYSCMP | | = | 12-20 |
| DEC | X | = | 3-8 | LTXS | X | = | 9-6 | SYSRD | | = | 12-14 |
| DEFTB | | | 11-5 | MOV | X | = | 3-14 | SYSWR | | = | 12-17 |
| DEFTC | | | 11-4 | MUL | | = | 4-5 | TEST | | | 12-10 |
| DEFTR | | | 11-6 | NCOB | | | 6-17 | TR | | | 7-10 |
| DEFVM | | | 11-3 | NOP | | | 12-3 | TFR | X | = | 3-21 |
| DEFWPH | | | 11-8 | NOT | X | = | 3-36 | TFRI | X | = | 3-23 |
| DEFWPR | | | 11-7 | OR | X | = | 3-34 | UNLOCK | | | 10-8 |
| DEI | | = | 3-11 | ORH | X | = | 2-7 | WTIME | | = | 12-5 |
| DIAG | | | 12-13 | ORL | X | = | 2-9 | XOB | | | 6-5 |
| DIGI | X | = | 3-29 | OUT | X | = | 2-13 | XOR | X | = | 2-10 |
| DIGIR | X | = | 3-30 | OUTS | X | = | 12-24 | | | | |
| DIGO | X | = | 3-31 | PB | | | 6-10 | | | | |
| DIGOR | X | = | 3-32 | PID | | | 12-6 | | | | |
| DIV | | = | 4-6 | PUT | X | | 3-19 | | | | |
| DSP | | | 3-6 | RCOB | | | 6-20 | | | | |
| DYN | X | = | 2-12 | RES | X | = | 2-15 | | | | |
| ECOB | | | 6-4 | RESD | X | = | 2-18 | | | | |
| EFB | | | 6-15 | ROTD | | = | 3-40 | | | | |
| EPB | | | 6-11 | ROTL | X | = | 3-43 | | | | |
| ESB | | | 7-4 | ROTR | X | = | 3-44 | | | | |
| EST | | | 7-9 | ROTU | | = | 3-39 | | | | |
| ETR | | | 7-11 | RSB | | | 7-6 | | | | |
| EXOB | | | 6-9 | RSI | | = | 3-13 | | | | |
| EXOR | X | = | 3-35 | RTIME | | = | 12-4 | | | | |
| FABS | X | = | 5-16 | SASI | | = | 8-8 | | | | |
| FADD | | | 5-5 | SASII | | = | 8-20 | | | | |
| FATAN | | | 5-13 | SB | | | 7-3 | | | | |
| FB | | = | 6-14 | SCOB | | | 6-18 | | | | |
| FCMP | X | | 5-10 | SCON | | = | 8-49 | | | | |
| FCOS | | = | 5-12 | SCONI | | = | 8-51 | | | | |
| FDIV | | = | 5-8 | SEI | | = | 3-9 | | | | |
| FEXP | | = | 5-14 | SET | X | = | 2-14 | | | | |
| FNL | | = | 5-15 | SETD | X | | 2-17 | | | | |
| FMUL | | = | 5-7 | SHID | | = | 3-38 | | | | |

Legenda:

Colonna "Ind": il simbolo 'X' indica che l'istruzione può essere indicizzata.

Colonna "Par": il simbolo '=' indica che l'istruzione accetta come operando un parametro FB

1. Introduzione

Il presente Manuale delle Istruzioni descrive in dettaglio ciascuna istruzione della famiglia SAIA PCD. Le istruzioni sono raggruppate in base al tipo per facilitarne l'apprendimento.

Lo scopo è quello di fare da supporto al “Manuale Utente” che fornisce la descrizione dettagliata delle “Utilities PCD” e dei metodi di programmazione strutturata utilizzati nei SAIA PCD

Importante

Le istruzioni descritte in questo Manuale sono valide per:

- PCD1 versione V001 (e successive)
- PCD2 versione V004 (e successive)
- PCD4.Mxx0 versione V005 (e successive)
- PCD4.Mxx5 versione V00C (e successive)
- PCD6.M5.. versione V004 (e successive)
- PCD6.M1/M2 versione V009 (e successive)
- Utilities PCD versione V2.0 (e successive)

Se si dispone di una precedente versione di PCD, alcune istruzioni potrebbero presentare delle differenze o non esistere del tutto.

Guida alla consultazione del manuale

Per ogni istruzione del set SAIA PCD vengono fornite una o più pagine.

| | |
|---------------------|---|
| Prima riga | Per ogni istruzione vengono riportate nella prima riga in alto il codice mnemonico, il nome dell'istruzione stessa e (tra parentesi), il nome originale inglese. |
| Descrizione | Descrive cosa fa l'istruzione e i suoi operandi. |
| Utilizzo | Descrive come viene utilizzata l'istruzione e fornisce il tipo e i possibili valori di ciascun operando. Una "[X]" dopo il codice mnemonico significa che è possibile utilizzare l'indirizzamento indicizzato, aggiungendo al codice mnemonico il suffisso opzionale " X" (es. STHX, INCX). Per l'indirizzamento indicizzato, l'operando (o gli operandi) indicizzato(i) viene contrassegnato con una "(i)" |
| Esempio | Riporta un tipico esempio di utilizzo dell'istruzione. |
| Flag | Indica quali Flag di stato sono influenzati (ACCU, N, P, Z, E) |
| Vedere anche | Riporta un elenco di altre istruzioni o argomenti che possono risultare utili nel contesto dell'istruzione trattata. |
| Uso pratico | Riporta un esempio pratico che illustra l'utilizzo dell'istruzione in un contesto appropriato. |

Convenzioni tipografiche:

| | |
|-----------------------|---|
| [] | Le parentesi quadre vengono utilizzate per racchiudere input o dati opzionali. Ad esempio [;commenti] significa che ";commenti" è opzionale e che quindi non è necessario includere dei commenti. |
| [X] | Una "[X]" dopo il codice mnemonico significa che si può utilizzare l'indirizzamento indicizzato, aggiungendo il carattere opzionale 'X' al codice mnemonico stesso (es: STHX, INCX). |
| (i) | Quando è ammesso l'indirizzamento indicizzato, (vedere [X]); l'operando indicizzato viene contrassegnato con una "(i)". |
| | Una serie di puntini (".") in un esempio, indica che la continuazione dell'esempio stesso può essere effettuata dal lettore. |
| LABEL: (etichetta) | Negli esempi, tutte le label sono riportate con il rispettivo nome seguito da un ':'; I due punti sono necessari se si utilizza un editor diverso da SEDIT. Nel caso in cui si utilizzi SEDIT non devono essere messi i due punti dopo il nome della label. |
| <> | I simboli di 'maggiore' e 'minore' racchiudono dei testi o delle espressioni che non devono essere digitati "alla lettera", ma sostituiti da un qualsiasi testo o espressione validi. |

1.1 Codice Supporto (MC) - MEDIUM Code

I codici di controllo del supporto informativo [mc] sono utilizzati per selezionare il tipo di elemento.

| Codice | Tipo | Campo |
|---------------|----------------|--------------|
| I | Ingresso | 0..8191 |
| O | Uscita | 0..8191 |
| F | Flag | 0..8191 |
| R | Registro | 0..4095 |
| T | Temporizzatore | 0..450 |
| C | Contatore | 0..1599 |
| K | Costante | 0..16383 |
| X | teXt (Testo) | 0..7999 |
| DB | Data Block | 0..7999 |

Ingressi e Uscite

Gli Ingressi e le Uscite sotto forma di moduli di interfaccia possono essere innestati nel PCD, il campo di indirizzamento può essere assegnato, in base alle esigenze, attraverso la posizione del modulo nel PCD4 oppure attraverso la configurazione dei DIP switch nel PCD6. Lo stato degli ingressi può essere solo interrogato e non modificato. Le Uscite possono essere attivate o disattivate e possono anche essere interrogate per quanto riguarda lo stato dei segnali.

Flag

I Flag sono celle di memoria di bit che possono essere trattate come le Uscite, possono cioè essere attivate, disattivate ed interrogate, per quanto riguarda lo stato. Compatibilmente con la loro struttura, i flag possono essere utilizzati per memorizzare qualsiasi informazione.

Temporizzatori e Contatori

I Temporizzatori e i Contatori sono dei registri programmabili che possono contenere un valore di 31 bit (0 - 2.147.483.647 in decimale); il numero di temporizzatori utilizzati dipende dall'istruzione DEFTC (il valore di default è di 32 Temporizzatori - indirizzi da 0 a 31 - e di 1568 Contatori - indirizzi da 32 a 1599). L'unica differenza tra un Temporizzatore e un Contatore consiste nel fatto che il Temporizzatore viene decrementato in funzione della base tempi (definita con l'istruzione DEFTB) il cui valore di default è 1/10 sec. I Temporizzatori e i Contatori possono contenere solo valori positivi o nulli.

Quando un Temporizzatore o un Contatore contiene un valore diverso da zero, il suo stato è alto (H), quando invece contiene il valore zero, il suo stato è basso (L).

Registri

Un registro è una cella di memorizzazione di 32 bit, che può contenere qualsiasi tipo di informazione in formato binario, decimale, esadecimale o in virgola mobile. Sui registri, si possono effettuare operazioni aritmetiche, trasferimenti di informazioni (da o verso: ingressi, uscite, flag, temporizzatori, contatori o altri registri).

I registri possono contenere sia valori positivi che valori negativi.

Costanti

Possono essere utilizzati diversi tipi di costanti (vedere dettagli nella pagina seguente).

Testo (X)

I testi sono stringhe ASCII che possono essere memorizzate nel PCD per essere emesse su una linea seriale.

Data Block (Blocco Dati)

Un "Data Block" è un'area di memoria che viene utilizzata per la memorizzazione di dati a 32 bit, i quali possono essere trasferiti a Registri, Temporizzatori, Contatori e viceversa.

1.2 Costanti

Nelle istruzioni PCD vengono utilizzati diversi tipi di costanti; il campo dei valori dipende dall'istruzione specifica.

Le costanti K NON sono ammesse per le istruzioni LOAD (caricamento) (LD, LDH e LDL).

Binarie

Il valore termina con una 'Q', esempio: 1001Q, 11111111Q. Massimo 12 bit.

Decimali

Tipo di default per le costanti; non esiste alcun formato speciale per definire le costanti decimali.

Esadecimale

Il valore esadecimale termina con una 'H', esempio ABCDH, 1234H, DEADH.

ASCII

Il carattere ASCII deve essere racchiuso tra apici, esempio 'A', 'z'.

Virgola mobile

Includono un '.' e/o un esponente 'E', esempio 1.2, 12E-1.

Il campo dei valori ammessi (FFP) è compreso tra 2.710505E-20 e -9.223371E+18..

Costanti K

Utilizzate quando si richiede un codice di controllo supporto (mc); questo tipo di costanti non viene mai utilizzato per le istruzioni LOAD (LD, LDL, LDH).

Per indicare che si tratta di una costante, si utilizza il tipo mc 'K'.

Possono essere assegnati valori binari, decimali, esadecimale o ASCII.

Esempio: K 10, K 11Q, K FFH, K'A',

| Tipo di costante | Valore minimo | Valore massimo |
|------------------|---|---------------------------------------|
| Decimale | - 2.147.483.648 | 2.147.483.647 |
| Esadecimale | 0 H | FFFFFFFF H |
| Binario | 0 Q | 111111111111 Q |
| ASCII | 'a', 'B', '%', ecc. | |
| Virgola Mobile | - 9.22337177 E+18 - 2.71050535 E -20 | +9.22337177 E+18 +5.42101070 E -20 |

1.3 I Codici Condizione [cc]

Flag di Stato sulle Operazioni Aritmetiche

I flag di stato aritmetici sono influenzati principalmente dalle istruzioni che operano su interi o su valori in virgola mobile.

Il flag di Errore può essere impostato a livello Alto da qualsiasi istruzione eseguita su dati non validi.

| | | |
|----------|----------|--|
| P | Positivo | Alto se il risultato di una istruzione aritmetica è positivo |
| N | Negativo | Alto se il risultato di una istruzione aritmetica è negativo. (Il Flag P è sempre l'inverso del Flag N) |
| Z | Zero | Alto se il risultato di una istruzione è zero. |
| E | Errore | Alto se una istruzione non può essere eseguita (impostato a livello Alto in caso di overflow, underflow o conversione) |

Accumulatore

Dal momento che il processore "legge" una riga di istruzione dopo l'altra, ne consegue che può interrogare solo un elemento dopo l'altro, per quanto riguarda lo stato del segnale degli elementi stessi (H o L).

Per poter elaborare una sezione di "linkage"¹⁾ completa fino ad una istruzione di azione, il risultato intermedio del "linkage" stesso deve essere memorizzato nell'Accumulatore (ACCU). Al termine del "linkage" ACCU contiene il risultato finale (0 o 1). In base a questo risultato, l'elemento corrispondente (esempio un'uscita) verrà attivata (ACCU=1) oppure non attivata (ACCU=0).

L'ACCUMULATORE viene impostato a livello Alto/Basso principalmente dalle istruzioni che operano sui Bit. Può essere impostato ad uno stato specifico oppure allo stato corrispondente ad un flag aritmetico, utilizzando l'istruzione ACC.

Codici di Condizione [cc]

I codici di condizione [cc] sono utilizzati per selezionare la condizione dei flag di stato prima di eseguire una istruzione.

Se la condizione è falsa, l'istruzione non viene eseguita.

Nota: Certe istruzioni dipendono dall'Accumulatore e vengono eseguite solo se l'Accumulatore stesso è Alto (1).

| | |
|------------|---|
| spazio | Nessun codice condizione |
| H | Se l'Accumulatore = H (1) |
| L | Se l'Accumulatore = L (0) |
| P | Se il flag Positivo = H (Flag negativo = L) |
| N | Se il flag Negativo = L |
| Z | Se il flag Zero = H |
| E | Se il flag Errore = H |
| (C) | Complemento, utilizzato solo con l'istruzione ACC |

1.4 Tipi e Valori delle Risorse

| Tipo | Descrizione | | Campo | Note | |
|------------|-----------------------------------|--|----------|-----------------------|-------------|
| I | Ingresso | condividono gli stessi indirizzi | 0..8191 | | Per sistema |
| O | Uscita | | 0..8191 | | |
| F | Flag | | 0..8191 | Volatile/Non volatile | |
| T | Temporizzatore | condividono gli stessi indirizzi | 0..450 | Volatile | |
| C | Contatore | | 0..1599 | Non Volatile | |
| R | Registro | | 0..4095 | Non Volatile | |
| K | Costante K | | 0..16383 | | |
| COB | Blocco a Organizzazione Ciclica | | 0..15 | | Per CPU |
| XOB | Blocco a Organizzazione Esclusiva | | 0..31 | | |
| PB | Blocco di Programma | | 0..299 | | |
| FB | Blocco Funzione | | 0..999 | | |
| SB | Blocco Sequenziale | | 0..31 | | |
| IST | Step Iniziale | | 0..1999 | | |
| ST | Step | | 0..1999 | | |
| TR | Transizione | | 0..1999 | | |
| X | Testo | condividono gli stessi indirizzi | 0..7999 | 4000..7999 | |
| DB | Data Block | | 0..7999 | in memoria estesa | |
| - | Semaforo | | 0..99 | | Per sistema |

2. Istruzioni BIT

Le istruzioni sui Bit, agiscono sull'Accumulatore, sugli Ingressi, sulle Uscite, sui Flag e sullo stato dei Contatori o dei Timer

| | |
|-------------|---|
| STH | Start high (Inizio su stato alto) |
| STL | Start low (Inizio su stato basso) |
| ANH | AND high (AND su stato alto) |
| ANL | AND low (AND su stato basso) |
| ORH | OR high (OR su stato alto) |
| ORL | OR low (OR su stato basso) |
| XOR | Exclusive OR (OR esclusivo) |
| ACC | Accumulator operations (Operazioni sull'Accumulatore) |
| DYN | Dynamic (edge detection) (Dinamizzazione - Rilevazione Fronti) |
| OUT | Set element from ACCU (Imposta elemento da ACCU) |
| SET | Set element (Set elemento) |
| RES | Reset element (Reset elemento) |
| COM | Complement element (Complmenta elemento) |
| SETD | Set element delayed (Set elemento con ritardo) |
| RESD | Reset element delayed (Reset elemento con ritardo) |

STH **INIZIO SU STATO ALTO (Start High)**

Descrizione L'ACCU è posto nello stato logico dell'elemento indirizzato.
 Questo è l'inizio di una nuova linea di combinazione.
 Con l'istruzione di start vengono cancellati i risultati della combinazione precedente; contemporaneamente viene letto lo stato "H" del segnale corrispondente all'elemento indirizzato I, O, F, T, C e il risultato viene memorizzato in ACCU.

| | |
|-----------------|---|
| Utilizzo | STH[X] elemento (i) ; I 0-8191, O 0-8191, F 0-8191 ; T 0-450, C 0-1599 |
|-----------------|---|

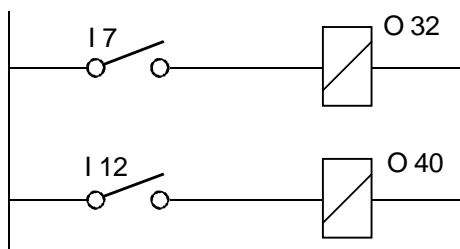
Esempio STH I 7 ; ACCU = stato di I 7

Flag ACCU impostato in base allo stato dell'elemento indirizzato

Nota Se un timer o un contatore contiene 0, il suo stato è definito basso (L), in caso contrario il suo stato è alto (H).

Vedere anche STHS, STL.

Uso pratico



; Nel PCD un programma deve, come minimo, contenere un COB

```

COB            0      ; Inizio COB
                 0
STH        I 7      ; Se l'ingresso 7 è H
OUT            O 32    ; Allora imposta (set) l'uscita 32
                         ; Altrimenti azzera (Reset) l'uscita 32
STH        I 12     ; Se l'ingresso 12 è H
OUT            O 40    ; Allora imposta (set) l'uscita 40
                         ; Altrimenti azzera (Reset) l'uscita 40
ECOB                   ; FINE COB
    
```

STL INIZIO SU STATO BASSO (Start Low)

Descrizione L'ACCU è posto nello stato logico opposto a quello dell'elemento indirizzato. Questo è l'inizio di una nuova linea di combinazione. Con l'istruzione di start vengono cancellati i risultati della combinazione precedente; contemporaneamente viene letto lo stato "L" del segnale corrispondente all'elemento indirizzato I, O, F, T, C e il risultato viene memorizzato in ACCU

Utilizzo **STL[X] elemento (i) ; I 0-8191, O 0-8191, F 0-8191 ; T 0-450, C 0-1599**

Esempio STL I 9 ; ACCU = stato di I 9 invertito

Flag ACCU è impostato al valore opposto rispetto allo stato dell'elemento indirizzato.

Vedere anche STH.

Uso pratico



COB 0 ; Inizio COB
0

STH I 8 ; Se l'ingresso 8 diventa lato
DYN F 10

LD T 15 ; Allora imposta il temporizzatore a 2 sec
20

STL T 15 ; Se il tempo è scaduto
OUT O 33 ; Allora imposta (Set) l'uscita 33
; Altrimenti azzera (Reset) l'uscita 33

STH T 15 ; Se il tempo non è scaduto
OUT O 34 ; Allora imposta (Set) l'uscita 34
; Altrimenti azzera (Reset) l'uscita 34

ECOB ; FINE COB

ANH AND SU STATO ALTO (AND High)

Descrizione Viene eseguito l'AND tra il valore dell'ACCU e lo stato logico dell'elemento indirizzato; ACCU viene impostato con il valore del risultato.

Utilizzo

**ANH[X] elemento (i) ; I 0-8191, O 0-8191, F 0-8191
; T 0-450, C 0-1599**

Esempio

ANH I 3 ; esegue l'AND tra ACCU e lo stato dell'Ingresso 3
 ANHX I 128 ; esegue l'AND tra ACCU e l'Ingresso (128 + Reg. Indice)

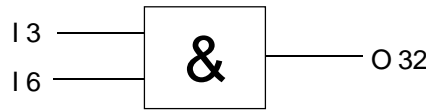
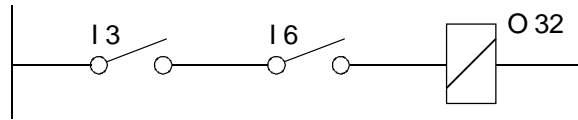
Flag

ACCU impostato in base al risultato.

Vedere anche

ANL.

Uso pratico



COB 0 ; Inizio COB
0

STH I 3 ; Se l'ingresso 3 è H
ANH I 6 ; E (AND) l'ingresso 6 è H
 OUT O 32 ; Allora imposta (Set) l'uscita 32
 ; Altrimenti azzera (Reset) l'uscita 32

ECOB ; FINE COB

ANL AND SU STATO BASSO (AND Low)

Descrizione Viene eseguito l'AND tra il valore dell'ACCU e il valore opposto a quello dello stato logico dell'elemento indirizzato; ACCU viene impostato con il valore del risultato.

Utilizzo

| | | |
|---------------|---------------------|---|
| ANL[X] | elemento (i) | ; I 0-8191, O 0-8191, F 0-8191 ; T 0-450, C 0-1599 |
|---------------|---------------------|---|

Esempio

ANL I 4 ; esegue l'AND tra ACCU e il valore opposto ripetuto allo
; stato dell'Ingresso 4.
ANHX I 128 ; esegue l'AND tra ACCU e il valore opposto a quello
; dell'ingresso (128 + Reg. Indice)

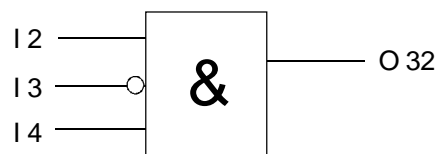
Flag

ACCU impostato in base al risultato.

Vedere anche

ANH.

Uso pratico



COB 0 ; Inizio COB
0

STH I 2 ; Se l'ingresso 2 è H
ANL I 3 ; E (AND) l'ingresso 3 è L
ANH I 4 ; E (AND) l'ingresso 4 è H
OUT O 32 ; Allora imposta (Set) l'uscita 32
; Altrimenti azzera (Reset) l'uscita 32

ECOB ; FINE COB

ORH OR SU STATO ALTO (OR High)

Descrizione Viene eseguito l'OR tra il valore dell'ACCU e lo stato logico dell'elemento indirizzato; ACCU viene impostato con il valore del risultato.
 Le istruzioni di OR vengono utilizzate per realizzare collegamenti in parallelo di elementi.
 Il collegamento globale inizia con una istruzione di start (STH oppure STL); ogni collegamento parziale aggiunto in parallelo inizia con una istruzione ORH.
 Se viene soddisfatta la condizione di uno dei collegamenti parziali aggiuntivi (ACCU=1), lo stato logico dei successivi collegamenti parziali, non esercita più alcuna influenza sul risultato del collegamento globale..

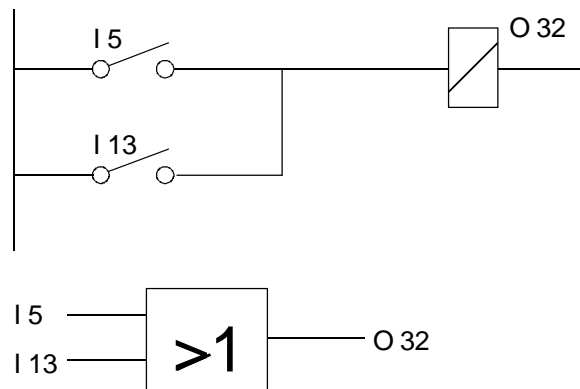
| | | | |
|-----------------|---------------|---------------------|---|
| Utilizzo | ORH[X] | elemento (i) | ; I 0-8191, O 0-8191, F 0-8191 ; T 0-450, C 0-1599 |
|-----------------|---------------|---------------------|---|

Esempio STH I 5 ; Se I 5 è H
 ORH I 13 ; oppure (OR) I 13 è H
 ; Allora ACCU = 1, altrimenti ACCU = 0

Flag ACCU impostato in base al risultato.

Vedere anche ORL.

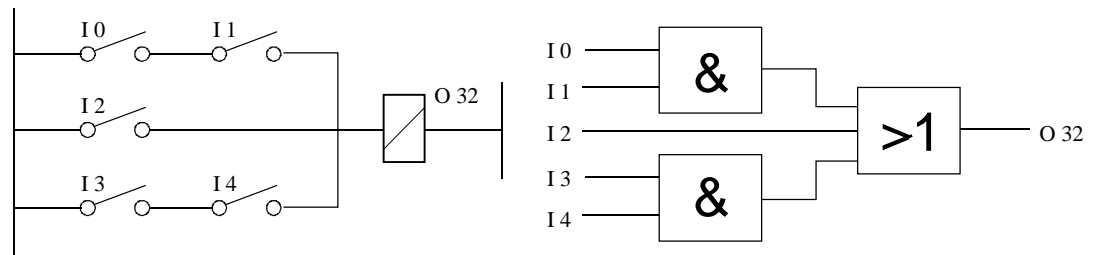
Uso pratico



COB 0 ; Inizio COB
 0

STH I 5 ; Se l'ingresso 5 è H
ORH I 13 ; oppure (OR) l'ingresso 13 è H
 OUT O 32 ; Allora imposta (Set) l'uscita 32
 ; Altrimenti azzera (Reset) l'uscita 32

ECOB ; FINE COB

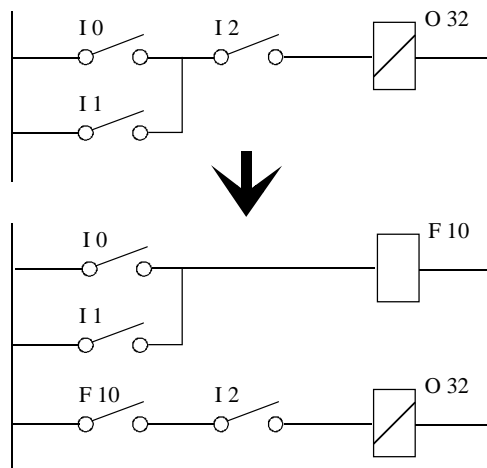


COB 0 ; Inizio COB
 0

STH I 0 ; Se l'ingresso 0 è H
 ANH I 1 ; e (AND) l'ingresso 1 è H
ORH I 2 ; oppure (OR) l'ingresso 2 è H
ORH I 3 ; oppure (OR) l'ingresso 3 è H
 ANH I 4 ; e (AND) l'ingresso 4 è H
 OUT O 32 ; Allora imposta (Set) l'uscita 32
 ; Altrimenti azzera (Reste) l'uscita 32

ECOB ; FINE COB

Dall'esempio sopra riportato si può vedere che l'istruzione OR è "prioritaria" rispetto all'istruzione AND.



COB 0 ; Inizio COB
 0

STH I 0 ; Se l'ingresso 0 è H
ORH I 1 ; oppure (OR) l'ingresso 1 è H
 OUT F 10 ; Allora imposta (Set) flag 10
 ; Altrimenti azzera (Reset) flag 10
 STH F 10 ; Se Flag 10 è H
 ANH I 2 ; e (AND) l'ingresso 2 è H
 OUT O 32 ; Allora imposta (Set) l'uscita 32
 ; Altrimenti azzera (Reset) l'uscita 32

ECOB ; FINE COB

ORL OR SU STATO BASSO (OR Low)

Descrizione Viene eseguito l'OR tra il valore dell'ACCU e lo stato logico opposto a quello dell'elemento indirizzato; ACCU viene impostato con il valore del risultato. Le istruzioni di OR vengono utilizzate per realizzare collegamenti in parallelo di elementi..

| | |
|-----------------|--|
| Utilizzo | ORL[X] elemento (i) ; I 0-8191, O 0-8191, F 0-8191 ; T 0-450, C 0-1599 |
|-----------------|--|

Esempio STH I 3 ; Se I 3 è H
 ORL I 7 ; oppure (OR) I 7 è L
 ; Allora ACCU = 1, altrimenti ACCU = 0

Flag ACCU impostato in base al risultato.

Vedere anche ORH.

XOR OR ESCLUSIVO (Exclusive OR)

Descrizione Viene eseguito l'OR Esclusivo tra il valore dell'ACCU e lo stato logico dell'elemento indirizzato; ACCU viene impostato con il valore del risultato. Con l'istruzione XOR, gli stati di due elementi possono essere messi a confronto. Se sono uguali il contenuto dell'ACCU è 0; se invece sono diversi il contenuto dell'ACCU è 1.

Utilizzo

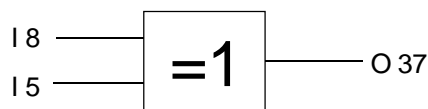
**XOR[X] elemento (i) ; I 0-8191, O 0-8191, F 0-8191
; T 0-450, C 0-1599**

Esempio XOR I 5; ACCU = ACCU XOR I 5

Flag ACCU impostato in base al risultato.

Vedere anche OR, AND.

Uso pratico



COB 0 ; Inizio COB
0

STH I 8 ; Se l'ingresso 8 è H
XOR I 5 ; or esclusivo (XOR) l'ingresso 5 è H
OUT O 37 ; Allora imposta (Set) l'uscita 37
; Altrimenti azzera (Reset) l'uscita 37

ECOB ; FINE COB

| I 8 | I 5 | O 37 |
|-----|-----|------|
| L | L | L |
| L | H | H |
| H | L | H |
| H | H | L |

ACC OPERAZIONI SULL'ACCUMULATORE (Accumulator Operations)

Descrizione Modifica lo stato dell' Accumulatore in base al codice (code):

| | | |
|----------|-------------|--|
| C | Complemento | Accu è complementato |
| H | High (Alto) | Accu è impostato al valore Alto (1) |
| L | Low (Basso) | Accu è impostato al valore Basso (0) |
| P | Positivo | Accu è impostato al valore di stato del Flag Positivo (P) |
| N | Negativo | Accu è impostato al valore di stato del Flag Negativo (N) |
| Z | Zero | Accu è impostato al valore di stato del Flag di Zero (Z) |
| E | Errore | Accu è impostato al valore di stato del Flag di Errore (E) |

L'operando non può essere fornito come parametro di un Blocco Funzione

Utilizzo

| | | |
|------------|---------------|---------------------------------|
| ACC | codice | ; codice = C H L P N Z E |
|------------|---------------|---------------------------------|

Esempio

ACC H ; Imposta ACCU a 1
 ACC E ; Imposta ACCU al valore di stato del flag E

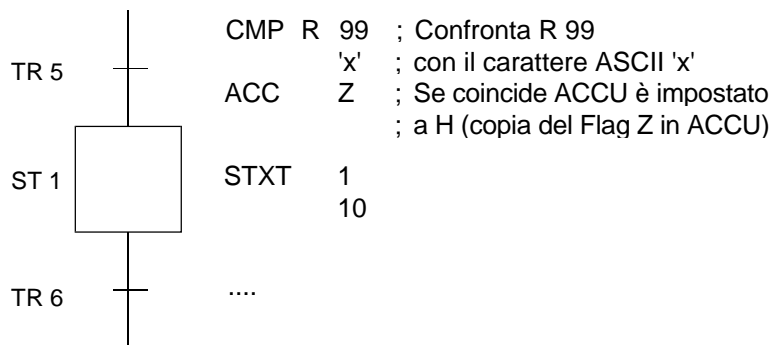
Flag

ACCU viene modificato

Vedere anche

OUT, Codici di Condizione.

Uso pratico



DYN RILEVAZIONE DINAMICA FRONTI) (Dynamic Edge Detection)

Descrizione Utilizzata per la rilevazione dei fronti di salita e di discesa dei segnali. Il risultato nell'ACCU è Alto (H), solo quando ACCU passa da basso (L) ad alto (H) a seguito di esecuzioni consecutive dell'istruzione DYN (fronte di salita del segnale). Il Flag fornito come operando, memorizza lo stato precedente dell'ACCU. Se l'ACCU è basso (L), rimane basso (L) e anche il Flag viene impostato al valore basso (L). Non è necessario che il Flag sia basso quando viene eseguita per la prima volta l'istruzione DYN. Per la rilevazione dei fronti di salita, si può utilizzare l'istruzione STH per interrogare l'elemento; per la rilevazione del fronte di discesa, si può utilizzare invece l'istruzione STL

Utilizzo

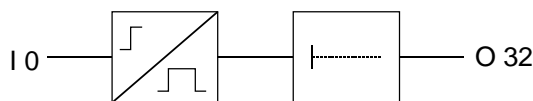
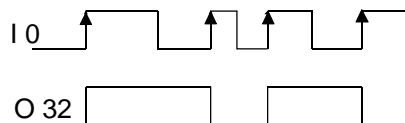
DYN[X] flag (i) ; F 0-8191

Esempio DYN F 100 ; Il Flag 100 memorizza lo stato dell'ACCU in modo dinamico

Flag ACCU impostato in base al risultato.

Vedere anche STH, STL, ANH, ANL, ORH, ORL.

Uso pratico



; Soluzione con istruzione DYN

```

COB      0      ; Inizio COB
          0
STH      I 0    ; Se l'ingresso 0 diventa H
DYN      F 500 ; (Rilevazione Fronte)
COM      O 32  ; Allora complementa l'uscita 32
          ; Altrimenti non fare nulla
          ; FINE COB
          ECOB
    
```

; Soluzione senza istruzione DYN

```

COB      0      ; Inizio COB
          0
STH      I 0    ; Se l'ingresso 0 è H
ANL      F 500 ; e (AND) flag 500 è L
SET      F 500 ; Allora imposta (Set) flag 500 = H
COM      O 32  ; complementa l'uscita 32
          ; Altrimenti non fare nulla
          ; Se l'ingresso 0 è L
STL      I 0
RES      F 500 ; Allora azzerà (Reset) flag 500 (stato = L)
          ; Altrimenti non fare nulla
          ; FINE COB
          ECOB
    
```

OUT **IMPOSTA ELEMENTO DA ACCUMULATORE** **(Set Element from Accumulator)**

Descrizione Imposta un'Uscita o un Flag in base allo stato dell'ACCU. Se l'ACCU è Alto (H), anche l'Uscita o il Flag vengono impostati al valore alto. Se l'ACCU è basso (L), anche l'Uscita o il Flag vengono impostate al valore basso.

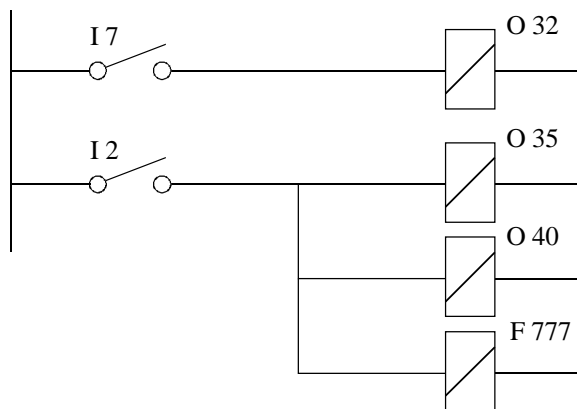
Utilizzo **OUT[X] elemento (i) ; O 0-8191, F 0-8191**

Esempio OUT O 32 ; Imposta l'uscita 32 in base allo stato dell'ACCU

Flag Invariati.

Vedere anche OUTS.

Uso pratico



```

COB      0      ; Inizio COB
          0

STH      I 7      ; Se l'ingresso 7 è H
OUT      O 32      ; Allora imposta (Set) l'uscita 32
                  ; Altrimenti azzera (Reset) l'uscita 32

STH      I 2      ; Se l'ingresso 2 è H
OUT      O 35      ; Allora imposta (Set) l'uscita 35,
                          altrimenti azzera (Reset) l'uscita 35
OUT      O 40      ;      imposta (Set) l'uscita 40,
                          altrimenti azzera (Reset) l'uscita 40
OUT      F 777      ;      imposta (Set) flag 777
                          altrimenti azzera (Reset) flag 777

ECOBS                    ; FINE COB
    
```

SET IMPOSTA ELEMENTO (Set Element)

Descrizione L'Uscita (o il Flag), viene impostata al valore Alto (H) solo se l'ACCU è Alto. In caso contrario non succede niente.

Un'uscita (o un Flag) impostata con l'istruzione SET, rimane alta (H) finchè non viene azzerata da una istruzione RES.

Le istruzioni SET e RES, sono generalmente utilizzate in programmi sequenziali (GRAFTEC).

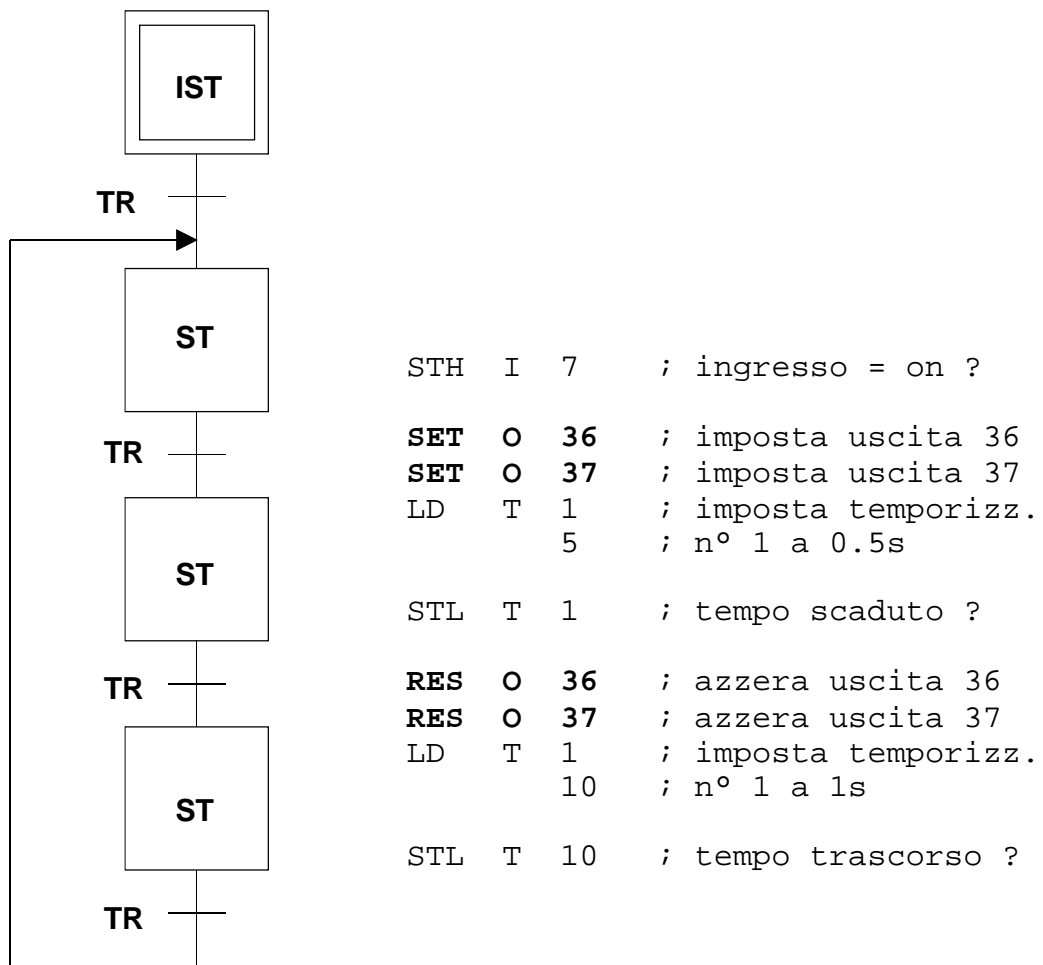
Utilizzo SET[X] elemento (i) ; O 0-8191, F 0-8191

Esempio SET O 32 ; Se ACCU è 1 allora O 32 = H

Flag Invariati.

Vedere anche RES, SETD, RESD.

Uso pratico Le Uscite 36 e 37 devono commutare ripetutamente dopo l'attivazione dell'ingresso 7



RES RESET ELEMENTO (Reset Element)

Descrizione L'uscita (o il Flag) viene impostata al valore basso (L) solo se l'ACCU è Alto (H).
Se l'ACCU è basso (L), non succede niente.

Utilizzo **RES[X] elemento (i) ; O 0-8191, F 0-8191**

Esempio RES O 13 ; Se ACCU è 1 allora O 13 = L

Flag Invariati.

Vedere anche SET, SETD, RESD.

Uso pratico vedere istruzione SET

COM **COMPLEMENTO DELL'ELEMENTO** **(Complement Element)**

Descrizione Lo stato dell'Uscita (o del Flag) è complementato (invertito) solo se l'ACCU è Alto (H). Se l'ACCU è basso (L), non succede niente

Nota:

Questa istruzione viene utilizzata principalmente per l'attivazione del "WATCHDOG". L'istruzione "COM O 255" deve essere inserita in un programma ciclico ed è necessario accertarsi che essa venga eseguita solo quando l'ACCU è allo stato alto (H) (a tale scopo è possibile inserire l'istruzione "ACC H" prima dell'istruzione "COM" oppure quest'ultima può essere posta immediatamente dopo l'istruzione COB).

Utilizzo

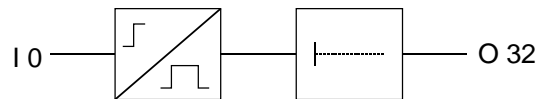
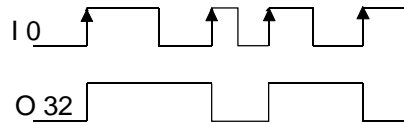
COM[X] elemento (i) ; O 0-8191, F 0-8191

Esempio COM O 32 ; Se ACCU è 1, viene invertito lo stato dell'Uscita 32

Flag Invariati.

Vedere anche OUT, SET, RES, DYN.

Uso pratico



COB 0 ; Inizio COB
 0

STH I 0 ; Se l'ingresso 0 diventa H
DYN F 500 ; (Rilevazione Fronte)
COM **O 32** ; Allora complementa l'uscita 32
 ; Altrimenti non fare nulla

ECOB ; FINE COB

RESD RESET ELEMENTO CON RITARDO (Reset Element Delayed)

Descrizione L'uscita (o il flag) viene impostata al livello Basso (L) dopo il ritardo definito nel secondo operando dell'istruzione **solo se l'ACCU è Alto (H)**. Il ritardo è espresso in unità di base tempi, secondo quanto impostato con l'istruzione DEFTB.

Le istruzioni SETD e RESD sono dedicate ai programmi sequenziali (GRAFTEC) e il loro uso permette di ottenere strutture più semplici dato che non è necessario attendere lo scadere del ritardo definito.

(Se le suddette istruzioni vengono usate in programmi BLOCTEC; è necessario utilizzarle sempre in combinazione con un'istruzione DYN. In caso contrario, viene attivato un altro temporizzatore ad ogni ciclo di programma, determinando così l'impostazione del flag di ERRORE dopo il 16° ciclo, quando tutti i temporizzatori disponibili sono stati utilizzati.)

Gli operandi non possono essere forniti come parametri di Blocchi Funzione.

Utilizzo

| | | |
|----------------|---------------------------------|--|
| RESD[X] | elemento (i) ritardo | ; O 0-8191, F 0-8191 ; Ritardo in unità di base tempi (es. 100ms) |
|----------------|---------------------------------|--|

Esempio

RESD O 32 ; Se ACCU è 1 allora l'uscita 32 = L
 100 ; dopo 100 x 100ms = 10 secondi

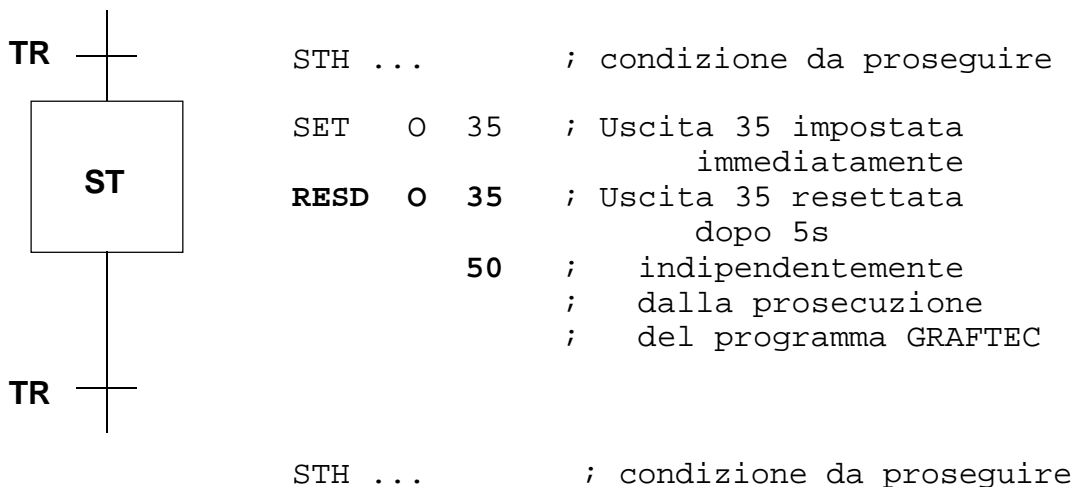
Flag

Se si tenta di eseguire per più di 16 volte un'azione ritardata, viene impostato il flag di **Errore (E)**.

Vedere anche

SETD, DEFTB.

Uso pratico



3. Istruzioni WORD

Queste istruzioni operano tutte sui Registri.

I Registri contengono valori binari, decimali, BCD o in virgola mobile. Per i valori in virgola mobile occorre utilizzare le istruzioni specifiche “Floating Point”.

| | | |
|----------------------------------|--------------|--|
| Caricamento Dati | LD | Load (32-bit value) (Carica valore a 32 bit) |
| | LDL | Load low word (lower 16 bits) (Carica parola meno significativa - primi 16 bit) |
| | LDH | Load high word (upper 16 bits) (Carica parola più significativa - primi 16 bit) |
| | DSP | Load Display Register (Carica registro display) |
| Aritmetiche primarie | INC | Increment Register (Incrementa registro) |
| | DEC | Decrement Register (Decrementa registro) |
| Registro Indice | SEI | Set Index register (Imposta registro indice) |
| | INI | Increment Index register (Incrementa registro indice) |
| | DEI | Decrement Index register (Decrementa registro indice) |
| | STI | Store Index register (Memorizza registro indice) |
| | RSI | Restore Index register (Ripristina registro indice) |
| Spostamento Dati | MOV | Move data (Spostamento dati) |
| | COPY | Copy data (Copia dati) } Utili specialmente |
| | GET | Get data (Acquisizione dati) } per indirizzamenti |
| | PUT | Put data (Scrittura dati) } indicizzati |
| | TFR | Transfer data (Trasferimento dati) |
| | TFRI | Transfer data indirect (Trasferimento dati con indirizzamento indiretto) |
| Ingresso/Uscita Binari | BITI | Bit in (Lettura bit) |
| | BITIR | Bit in reversed (Lettura bit reverse) |
| | BITO | Bit out (Uscita bit) |
| | BITOR | Bit out reversed (Uscita bit reverse) |
| Ingresso/Uscita Digit BCD | DIGI | Digit in (Lettura digit) |
| | DIGIR | Digit in reversed (Lettura digit reverse) |
| | DIGO | Digit out (Uscita digit) |
| | DIGOR | Digit out reversed (Uscita digit reverse) |
| Logiche | AND | AND Registers (AND tra registri) |
| | OR | OR Registers (OR tra registri) |
| | EXOR | Exclusive-OR Registers (OR esclusivo tra registri) |
| | NOT | Complement Register (Complemento di registri) |
| Rotazione e Shift | SHIU | Shift Registers up (Shift avanti di registri) |
| | SHID | Shift Registers down (Shift indietro di registri) |
| | ROTU | Rotate Registers up (Rotazione avanti di registri) |
| | ROTD | Rotate Registers down (Rotazione indietro di registri) |
| | SHIL | Shift Register left (Shift a sinistra di registri) |
| | SHIR | Shift Register right (Shift a destra di registri) |
| | ROTL | Rotate Register left (Rotazione a sinistra di registri) |
| | ROTR | Rotate Register right (Rotazione a destra di registri) |

LD CARICA (VALORE A 32 BIT) (Load 32-Bit Value)

Descrizione Il Registro, il Temporizzatore o il Contatore indirizzati vengono caricati con il valore indicato (32 bit).

Per Temporizzatori e Contatori:

- l'operazione viene effettuata solo se l'ACCU è Alto (H).
- non è possibile caricare valori negativi o in virgola mobile. (solo valori decimali, Esadecimali, ASCII o binari).
- Se l'operazione di caricamento riguarda un temporizzatore, quest'ultimo parte immediatamente.
- Lo stato di un timer o di un contatore è H quando il contenuto relativo è diverso da zero, in caso contrario lo stato è L.

Per i Registri:

- l'operazione è indipendente dallo stato dell'ACCU.
- il valore può essere decimale, esadecimale, ASCII o in virgola mobile.

I valori binari devono avere il suffisso 'Q' o 'Y'.

I valori Esadecimali devono avere il suffisso 'H'.

I valori in virgola mobile devono contenere il punto decimale '.' oppure un esponente "E6".

I valori ASCII devono essere racchiusi tra apici 'a', 'A'.

Dal momento che il valore è a 32 bit, per l'intera istruzione vengono utilizzate tre righe di programma.

Non è possibile fornire gli operandi come parametri di un Blocco Funzione.

Utilizzo

| | | |
|--------------|---------------------|---|
| LD[X] | elemento (i) | ; R 0-4095, T 0-450, C 0-1599. |
| | valore | ; Decimale: da -2.147.483.648 a +2.147.483.647 |
| | | ; Hex: da 0H a FFFFFFFFH |
| | | ; Binario: da 0Y a 111...111Y (32 bit) |
| | | ; Virgola mobile: da ± 5.42101E-20 a ± 9.22337E+18 |
| | | ; ASCII: 'A'-'Z', '0'-'9', '!', '?' ecc. |

Esempio LD R 0 ; Carica R 0 con il valore 321 in virgola mobile
3.21E1 ; (Temporizzatori & Contatori devono avere valori positivi)

Flag Invariati.

Vedere anche LDH, LDL (caricamento a 16 bit), Costanti.

Nota L'istruzione LD richiede tre righe di programma
LD T/C: eseguita solo quando ACCU = H (1).
LD R: sempre eseguita

LDL **CARICA PAROLA MENO SIGNIFICATIVA (PRIMI 16 BIT) (Load Low Word - Lower 16 Bit)**

Descrizione Carica i primi 16 bit (0-15) di un Registro, di un Temporizzatore o di un Contatore con un valore a 16 bit (0-65535); i 16 bit più significativi vengono sempre impostati al valore 0..

Per Temporizzatori e Contatori:

l'istruzione LDL viene eseguita **solo se ACCU = 1**.

Per i Registri:

l'istruzione LDL viene sempre eseguita.

LDL (così come LDH, che carica i 16 bit più significativi) consente di passare costanti a 16 bit come parametri di un Blocco Funzione, oppure di caricarne direttamente il valore.

LDH carica i 16 bit più significativi.

Utilizzando le due istruzioni si può caricare un valore a 32 bit. In questo caso, per caricare tutti i 32 bit occorre eseguire per prima l'istruzione LDL, in quanto questa pone a zero (0) i 16 bit più significativi. I valori possono essere caricati in decimale, esadecimale, binario o ASCII, NON in virgola mobile.

Utilizzo

| | | |
|---------------|---------------------|--------------------------------------|
| LDL[X] | elemento (i) | ; R 0-4095, T 0-450, C 0-1599 |
| | valore | ; Decimale: 0-65535 |
| | | ; Esadecimale: 0H-FFFFH |
| | | ; Binario: 16 bit |

Esempio LDL R 100 ; Carica il registro 100 con il valore esadecimale FFFF,
 FFFFH ; che corrisponde al valore decimale 65535.
 ; R100 = 0000FFFFH

Flag Invariati.

Vedere anche LDH, LD, Costanti.

Nota Quando vengono utilizzati unicamente valori < 65535, l'istruzione LDL può essere di norma utilizzata per caricare Contatori, Temporizzatori o Registri.

LDH CARICA PAROLA PIU' SIGNIFICATIVA (ULTIMI 16 BIT) (Load High Word - Upper 16 Bits)

Descrizione Carica i 16 bit più significativi (16-31) di un registro, i 16 bit meno significativi non vengono influenzati. LDH (così come LDL, che carica i 16 bit meno significativi) consente di passare costanti a 16 bit (0-65535) come parametri di un Blocco Funzione, oppure di caricarne direttamente il valore. Utilizzando queste istruzioni si può caricare un valore a 32 bit. In questo caso, per caricare tutti i 32 bit occorre eseguire per prima l'istruzione LDL, in quanto questa pone a zero (0) i 16 bit più significativi. I valori possono essere caricati in decimale, esadecimale, ASCII o binario, NON in virgola mobile.
LDH non può essere utilizzata per caricare Temporizzatori o Contatori in quanto non è possibile caricare separatamente i 16 bit più significativi.

Utilizzo

| | | |
|---------------|---------------------|--------------------------------|
| LDH[X] | elemento (i) | ; R 0-4095 |
| | valore | ; Decimale: 0-65535 |
| | | ; Esadecimale: 0H-FFFFH |
| | | ; Binario: 16 bit |

Esempio LDH R 100 ; Carica i bit 31-16 del Registro 100
 FFFFH ; con il valore esadecimale FFFF.
 ; R 100 = FFFFxxxx Esadecimale

Flag Invariati.

Vedere anche LDL, LD, Costanti

Uso pratico Per caricare un Registro in un Blocco Funzione con una costante a 32 bit, non si può utilizzare direttamente l'istruzione LD; al suo posto occorre utilizzare LDL e LDH. (I 16 bit più significativi di una costante possono essere separati dai 16 bit meno significativi utilizzando l'istruzione Assembler '&' (AND) e '/' (DIVIDE)). Ad un Blocco Funzione deve essere passata come parametro una costante (12345678) caricata in un Registro. Si ricorda che LDL deve essere eseguita PRIMA di LDH.

```

COB            0            ; Inizio COB
                 0

CFB            0            ; Chiamata Blocco Funzione 0
12345678 & 0FFFFH ; Parametro 1 (primi 16 bit)
12345678 / 0FFFFH ; Parametro 2 (ultimi 16 bit)

ECOB                        ; FINE COB

FB             0            ; Inizio FB
LDL          R 0            ; Carica i primi 16 bit del Registro 0
                 = 1           ; con il 1° parametro (primi 16 bit)
LDH          R 0            ; Carica gli ultimi 16 bit del Registro 0
                 = 2           ; con il 2° parametro (ultimi 16 bit)

EFB
    
```

**DSP CARICA REGISTRO DISPLAY
(Load Display Register)**

Descrizione Nel registro Display è possibile caricare lo stato logico di un Ingresso, di una Uscita, di un Flag oppure il contenuto di un Registro, di un Timer, di un Contatore oppure una costante. Tale valore viene visualizzato in forma decimale sull'Unità di Programmazione PCD8.P1.
Può essere utile per visualizzare un codice di errore o uno stato.

L'operando non può essere fornito come parametro di un Blocco Funzione.

| | |
|-----------------|--|
| Utilizzo | DSP valore ; I 0-8191, O 0-8191, F 0-8191, T 0-450, ; C 0-1599, R 0-4095, K 0-16383 |
|-----------------|--|

Esempio DSP R 0 ; Registro Display = contenuto di R 0
 DSP K 1234 ; Registro Display = costante 1234

Flag Invariati.

Vedere anche Manuale Hardware PCD1/2

Uso pratico (vedere istruzione INC)

Nota: Sul PCD2 l'istruzione DSP viene utilizzata per i moduli display PCD2.F510 e PCD2.F530.

INC INCREMENTA UN REGISTRO O UN CONTATORE (Increment Register or Counter)

Descrizione Il Registro o il Contatore viene incrementato di uno.

I Contatori vengono incrementati **solo se ACCU = 1**.

I Registri vengono incrementati **indipendentemente dallo stato dell'ACCU**.

Utilizzo

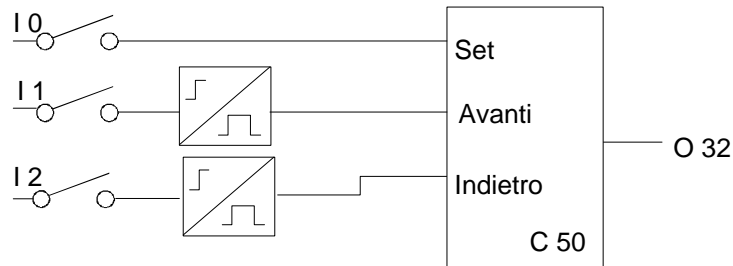
INC[X] elemento (i) ; R 0-4095, C 0-1599

Esempio INC R 100 ; R 100 = R 100 + 1

Flag I flag **Zero (Z)** e **Segno (P o N)** vengono impostati in base al risultato.
Il flag di **Errore (E)** viene impostato in caso di overflow.

Vedere anche DEC, ADD

Uso pratico Contatore Avanti/Indietro con preselezione e visualizzazione del valore del contatore



```

COB            0            ; Inizio COB
                 0

STH            I 0            ; Se l'ingresso 0 è H
LD             C 50           ; Allora carica il contatore 50 con 5
                 5
                 ; Altrimenti non fare nulla
STH            I 1            ; Se l'ingresso 1 diventa H
DYN            F 1            ; (Rilevazione Fronte)
INC            C 50           ; Allora incrementa di 1 il contatore 50
                 ; Altrimenti non fare nulla
STH            I 2            ; Se l'ingresso 2 diventa H
DYN            F 2            ; (Rilevazione Fronte)
DEC            C 50           ; Allora decrementa di 1 il contatore 50
                 ; Altrimenti non fare nulla
STH            C 50           ; Se il contenuto del contatore 50 è >> 0
OUT            O 32           ; Allora imposta (Set) l'uscita 32
                 ; Altrimenti azzera (Reset) l'uscita 32
DSP            C 50           ; Visualizza il contatore 50
ECOB
    
```

DEC DECREMENTA UN REGISTRO O UN CONTATORE (Decrement Register or Counter)

Descrizione Il Registro o il Contatore viene decrementato di uno.

I Contatori vengono decrementati **solo se ACCU = 1**.

I Registri vengono decrementati **indipendentemente dallo stato dell'ACCU**

Utilizzo

DEC[X] elemento (i) ; R 0-4095, C 0-1599

Example

DEC R 100 ; R 100 = R 100 - 1

Esempio

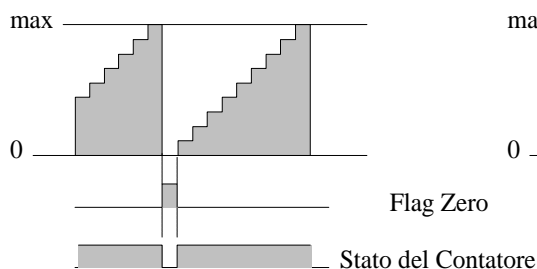
I flag **Zero (Z)** e **Segno (P o N)** vengono impostati in base al risultato.

Il flag di **Errore (E)** viene impostato in caso di underflow.

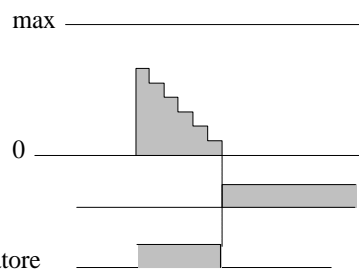
Vedere anche

INC, SUB

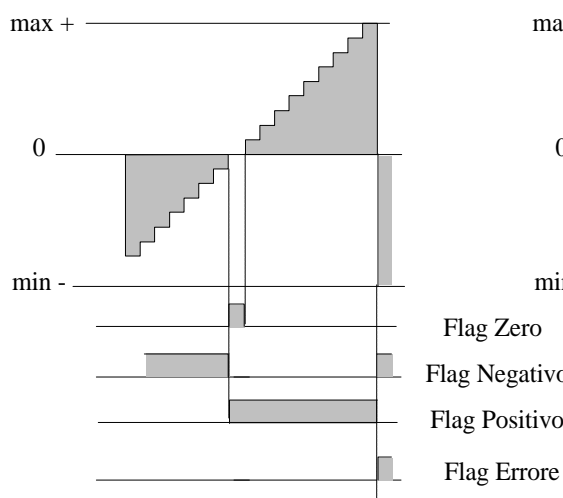
INC Contatore



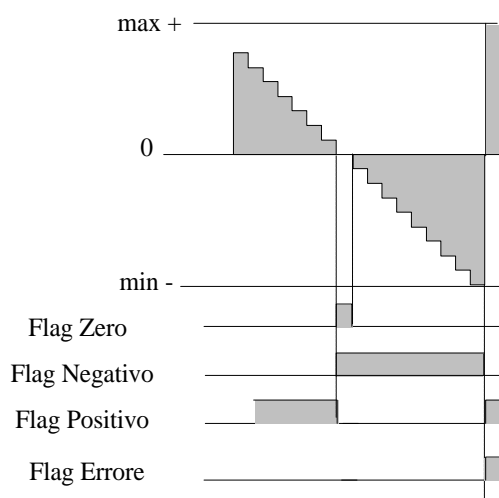
DEC Contatore



INC Registro



DEC Registro



SEI IMPOSTA IL REGISTRO INDICE (Set Index Register)

Indirizzamento Indicizzato

É spesso necessario trattare nello stesso modo una serie di ingressi, di uscite, di flag,..... (ad esempio azzerare tutti i Registri o i flag contenenti dei valori). In casi come questo, con l'aiuto dell'indirizzamento indicizzato si può ridurre drasticamente la lunghezza dei programmi.

Ciascun COB o XOB ha il proprio **Registro Indice**.

Tale registro è utilizzato per effettuare indirizzamenti indicizzati, dove il contenuto del Registro Indice viene sommato al valore dell'operando per fornire l'indirizzo effettivo. Le istruzioni indicizzate terminano sempre con una 'X', ad esempio STHX, BITIX. Il Registro Indice può essere caricato o salvato, oltre che incrementato o decrementato fino al limite predefinito.

Descrizione

Il Registro Indice corrente viene caricato con la costante fornita (K 0-8191) oppure con il contenuto del Registro indicato.

NOTA: Il Registro Indice può avere un valore compreso tra 0..8191 (13 bit).

Se viene introdotto un valore > 8191, l'indice viene impostato a 8191 e viene chiamato l'XOB 12 .

Se viene introdotto un valore < 0, l'indice viene impostato a 0 e viene chiamato l'XOB 12.

Utilizzo

| | | |
|------------|---------------|-----------------------------|
| SEI | valore | ; K 0-8191, R 0-4095 |
|------------|---------------|-----------------------------|

Esempio

SEI K 32 ; Carica il Registro Indice con il valore 32
 SEI R 32 ; Carica il Registro Indice con il valore del Registro 32

Flag

Invariati.

Vedere anche

INI, DEI, STI, RSI

Uso pratico

Lo stato di un ingresso il cui indirizzo è fornito da un preselettore BCD deve essere riportato sull'uscita 32

| | | |
|------------|--------------|---|
| COB | 0 | |
| | 0 | |
| DIGI | 2 | ; Legge 2 digits |
| I 24 | | ; dall'ingresso 24 (fino a 31) |
| R 500 | | ; e li memorizza in R 500 |
| SEI | R 500 | ; Carica Indice con il contenuto di R 500 |
| STHX | I 0 | ; Se l'ingresso (0 + Indice) è H |
| OUT | O 32 | ; Allora imposta (Set) l'uscita 32 |
| | | ; Altrimenti azzerà (Reset) l'uscita 32 |
| ECOB | | |

INI INCREMENTA IL REGISTRO INDICE (Increment Index Register)

Descrizione Il Registro Indice corrente viene confrontato con il valore dell'operando (costante K fornita o contenuto di un Registro).
 Se il Registro Indice è inferiore a tale valore, viene incrementato e l'ACCU viene impostato a 1.
 Se il Registro Indice è uguale o maggiore rispetto al valore dell'operando, NON viene incrementato e l'ACCU viene impostato a 0.
 Se il valore dell'operando è >8191 o <0 viene chiamato l'XOB 12.

Utilizzo

| | | |
|------------|---------------|------------------------------|
| INI | valore | ; K 0-8191, R 0..4095 |
|------------|---------------|------------------------------|

Esempio

INI K 100 ; Incrementa il Registro Indice se è inferiore di 100.
 INI R 333 ; Incrementa il Registro indice
 se è inferiore al contenuto di R 333

Flag

ACCU = 1 se il Registro Indice è inferiore al valore dell'operando.
ACCU = 0 se il Registro Indice è maggiore o uguale al valore dell'operando.

Vedere anche

DEI, SEI

Uso pratico

All'accensione, i registri da 1500 a 1999 devono essere azzerati (valore 0)

```

XOB      16      ; XOB eseguito all'accensione
. . . . .
SEI      K 0      ; Imposta il Registro Indice a 0
                    ; Ripetizione
Reset:   LDX      R 1500 ; Carica il Registro (1500 + Reg. Ind.)
                    0      ; con 0
INI     K 499   ; Incrementa di 1 il Registro Indice
JR       H Reset   ; Finché il Registro Indice > 499
(ACC     H)
. . . . .
EXOB
  
```

DEI DECREMENTA IL REGISTRO INDICE (Decrement Index Register)

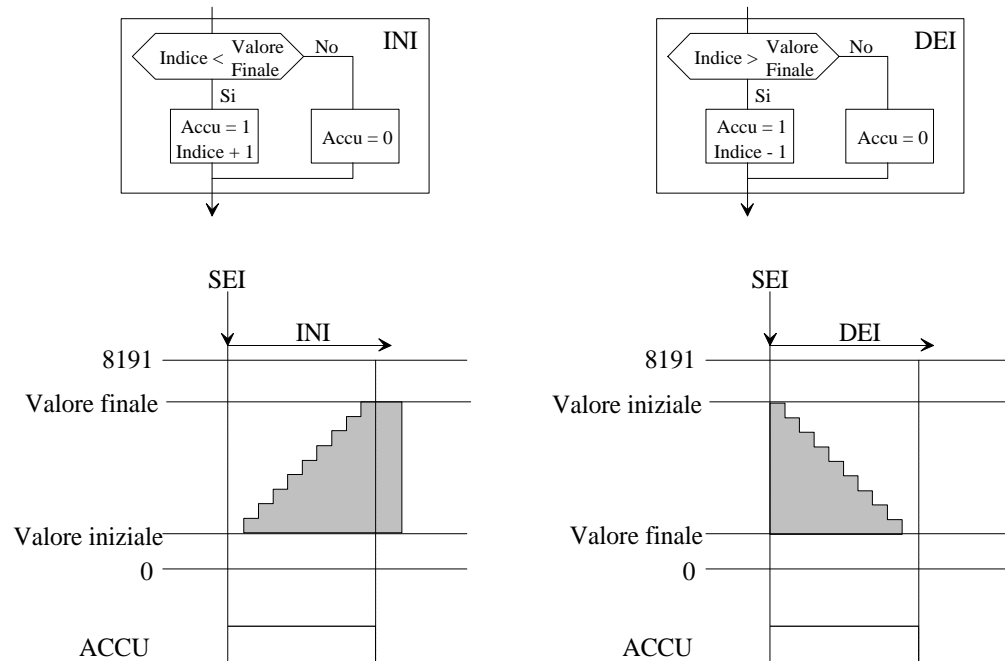
Descrizione Il Registro Indice corrente viene confrontato con il valore dell'operando (costante K fornita o contenuto di un Registro).
 Se il Registro Indice è superiore a tale valore, viene decrementato e l'ACCU viene impostato a 1.
 Se il Registro Indice è uguale o inferiore rispetto al valore dell'operando (costante o contenuto del Registro), NON viene incrementato e l'ACCU viene impostato a 0.
 Se il valore dell'operando è >8191 o <0 viene chiamato l'XOB 12.

| | |
|-----------------|--|
| Utilizzo | DEI valore ; K 0-8191, R 0..4095, |
|-----------------|--|

Esempio DEI K 100 ; Decrementa il Registro Indice se è maggiore
 ; della costante 100.
 DEI R 444 ; Decrementa il Registro Indice se è maggiore
 ; del contenuto di R 444.

Flag **ACCU = 1** se il Registro Indice è maggiore del valore dell'operando.
ACCU = 0 se il Registro Indice è inferiore o uguale al valore dell'operando.

Vedere anche INI, SEI



Valore finale = valore dell'operando dell'istruzione INI / DEI

STI **MEMORIZZA IL REGISTRO INDICE** **(Store Index Register)**

Descrizione Il valore del Registro Indice corrente viene memorizzato nel Registro specificato. Può essere ricaricato nel Registro Indice utilizzando l'istruzione RSI. Il Registro Indice non viene modificato

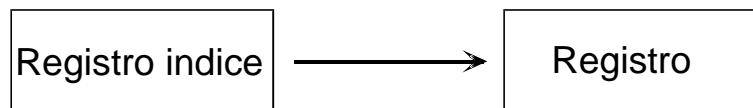
Utilizzo

| | | |
|------------|------------|-------------------|
| STI | reg | ; R 0-4095 |
|------------|------------|-------------------|

Esempio STI R 100 ; Memorizza il contenuto del Registro Indice nel Registro 100

Flag Invariati.

Vedere anche RSI



**RSI RIPRISTINA IL REGISTRO INDICE
(Restore Index Register)**

Descrizione Carica il Registro Indice con il contenuto del Registro specificato.
 Il valore del Registro sarà tipicamente il valore di un Registro Indice precedentemente salvato con l'istruzione STI.
 Il valore massimo che il Registro può contenere è 8191, vengono caricati solo i 13 bit meno significativi.
 Se si introduce un parametro < 0 o > 8191 viene chiamato l'XOB 12.

Utilizzo

| |
|---|
| RSI reg ; R 0-4095 |
|---|

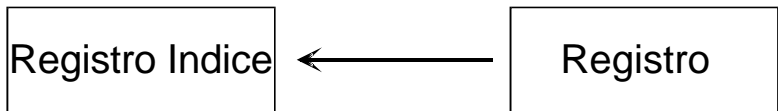
Esempio RSI R 100 ; Carica il contenuto del Registro Indice con il contenuto del ; Registro 100.

E' identico a : SEI R 100

Flag Invariati.

Vedere anche STI, SEI

Usò pratico



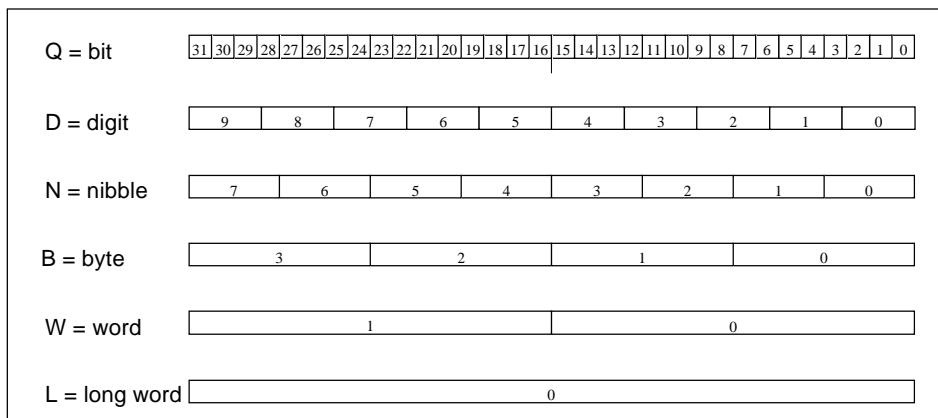
MOV SPOSTAMENTO DATI (Move Data)

Descrizione Sposta i Dati da un Temporizzatore, Contatore o Registro in un Registro.
É una istruzione composta da 4 righe:

- Il 1° e il 3° operando rappresentano la sorgente e la destinazione.
- il 2° e il 4° operando rappresentano il tipo di dati e la posizione.

| | |
|---------------------------|------|
| Q = Bit (sposta un 1 bit) | 0-31 |
| D = Digit (4 Bit BCD) | 0-9 |
| N = Nibble (4 Bit Binari) | 0-7 |
| B = Byte (8 Bit) | 0-3 |
| W = Word (16 Bit) | 0/1 |
| L = Long word (32 Bit) | 0 |

I tipi di dati (Q, D, ecc) del 2° e del 4° operando devono essere uguali, mentre le posizioni di sorgente e destinazione possono essere diverse.



Utilizzo

| | | |
|---------------|-----------------------|--------------------------------------|
| MOV[X] | sorgente (i) | ; R 0-4095, T 0-450, C 0-1599 |
| | tipo posizione | ; Q D N B W L vedere figura |
| | dest (i) | ; R 0-4095 |
| | tipo posizione | ; Q D N B W L vedere figura |

Esempio

Sposta il nibble più significativo (N7) del Registro 100 nel nibble meno significativo (N0) del Registro 101.

```

; Prima: R100: 1111 1010 1010 1010 1010 1010 1010 1010
          R101: 0001 0001 0001 0001 0001 0001 0001 0001

          MOV   R 100
              N 7
              R 101
              N 0
; Dopo:  R100: 1111 1010 1010 1010 1010 1010 1010 1010
          R101: 0001 0001 0001 0001 0001 0001 0001 1111

```

Flag

Invariati.

Vedere anche

COPY, GET, PUT, LD, LDH, LDL

COPY COPIA DATI (Copy Data)

Descrizione Le istruzioni COPY, GET e PUT hanno delle caratteristiche in comune
 Sono utilizzate per trasferimenti indicizzati tra Registri, Temporizzatori e Contatori.
 In ogni caso viene copiato un intero Registro, Temporizzatore o Contatore.

Nel caso di **COPYX**, il contenuto del primo operando viene copiato nel secondo; ENTRAMBI gli operandi sono indicizzati.

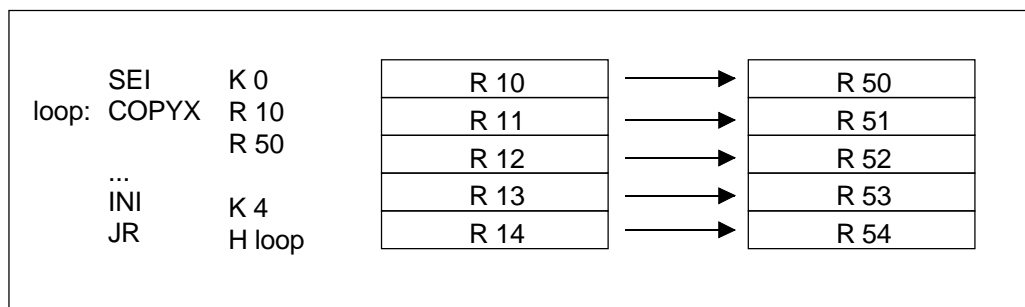
| | |
|-----------------|--|
| Utilizzo | COPY[X] sorgente (i) ; R 0-4095, T0-450, C0-1599 destinazione (i) ; R 0-4095, T0-450, C0-1599 |
|-----------------|--|

Esempio COPYX R 10 ; Trasferisce il contenuto di R (10 + Indice)
 R 50 ; in R (50 + Indice)

Flag I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore copiato.

Vedere anche GET, PUT, MOV

Uso pratico



GET LETTURA DATI (Get Data)

Descrizione Le istruzioni COPY, GET e PUT hanno delle caratteristiche in comune Sono utilizzate per trasferimenti indicizzati tra Registri, Temporizzatori e Contatori. In ogni caso viene copiato un intero Registro Temporizzatore o Contatore.

Nel caso di **GETX**, il contenuto del primo operando viene copiato nel secondo; solo il primo operando è indicizzato.

Inoltre, l'istruzione GET consente il trasferimento da Testi/Blocchi di Dati in Registri/Temporizzatori/Contatori.

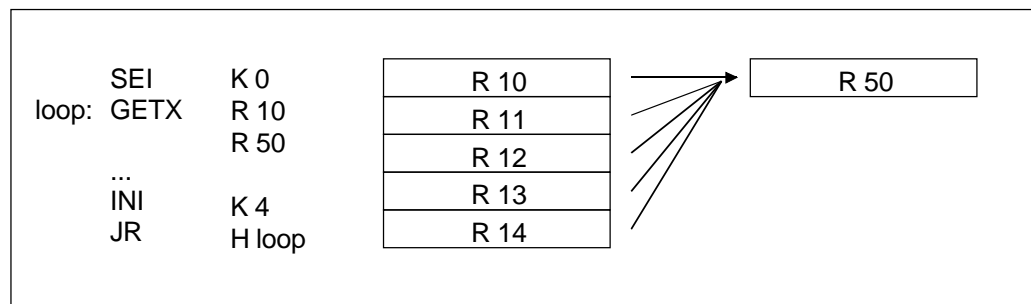
Utilizzo **GET[X]** sorgente (i) ; R 0-4095, T 0-450, C 0-1599, X 0-7999, DB 0-7999
destinazione ; R 0-4095, T 0-450, C 0-1599

Esempio GETX R 10 ; Trasferisce il contenuto di R (10 + Indice)
R 50 ; in R 50

Flag I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore copiato.

Vedere anche PUT, COPY, MOV

Uso pratico



Trasferimento tra Testo/Blocco Dati e R/T/C

L'istruzione GET[X] può trasferire il Testo specificato in R/C/T, finchè non viene riconosciuto il carattere di fine Testo (00 - terminatore NUL). Se il Testo non termina esattamente alla fine di un R/T/C, la parte rimanente del medesimo rimane inalterata.

Analogamente, GET[X] può trasferire il contenuto di un Blocco Dati in una serie di R/T/C fino al termine del Blocco Dati stesso. Un BLOCCO DATI è un'area di memoria utente in cui si può memorizzare "run time" un elevato numero di Registri, Temporizzatori o Contatori. I Blocchi Dati possono essere utilizzati per memorizzare valori che sono specifici per un determinato processo e quindi liberare dei R/C/T che potranno venire utilizzati da altri processi.

Se l'istruzione tenta di leggere da un Testo o da un Blocco Dati inesistente, viene chiamato l'XOB 13 (e viene attivato il Flag di Errore). Se il numero del Testo o del Blocco Dati indicizzato è maggiore di 7999, viene chiamato l'XOB 12 (superamento della capacità - overflow - del Registro Indice)

Formati del BLOCCO DATI e del TESTO

Tutti i Blocchi di Dati e i Testi devono essere definiti nel modo seguente all'interno del programma sorgente:

```
DB      numero  '[' [lunghezza] ']' [valore 1 [,valore 2]...[,valore n] ]

TEXT   numero  '[' [lunghezza] ']' "riga testo 1"  [;commento]
                                     ["riga testo 2"  [;commento]]
                                     ["riga testo n"  [;commento]]
```

dove numero e lunghezza possono essere in formato:

```

                                     decimale
                                     simbolico
e valore 1, ... in formato          decimale
                                     esadecimale
                                     binario
                                     virgola mobile
                                     ASCII
                                     simbolico
```

I Testi e i Blocchi di Dati sono numerati (0...3999) ma, dal momento che condividono la stessa area di memoria, non possono utilizzare lo stesso numero. Ad esempio, se viene definito il Blocco Dati 10, non è possibile utilizzare il Testo 10. Un Testo può essere definito, di norma, con una lunghezza massima di 3069 caratteri.

Un Blocco Dati può contenere invece fino a 383 elementi dove, per elemento, si intende un Registro oppure un Temporizzatore o un Contatore.

Nota: Non è consentito utilizzare <254> o <255> come primo byte di un Testo normale. Se il PCD contiene una memoria EPROM, i Blocchi Dati e i Testi possono solo essere letti (istruzione GET).

Esempi:

```
DB 100  [10]                ; Blocco Dati numero 100 di 10 elementi
                                     ; tutti i valori inizializzati a zero.

DB 101  [] 1.0,konst,3.0,4   ; Blocco Dati numero 101 di 4 elementi
                                     ; 2 dei quali in virgola mobile e 1 è un simbolo

DB 102  [4] 1,2,3           ; Blocco Dati numero 102 di 4 elementi 1, 2, 3, 0

DB num  [data_len]         ; Blocco Dati con simboli predefiniti

TEXT 100 "NORMAL TEXT"
TEXT 103 [5] "123"         ; Testo 103 cioè "123 " (con due spazi)
```

ACQUISIZIONE (GET) di un Blocco Dati o di un Testo

Esempio 1:

Blocco Dati come dichiarato nel programma sorgente:

```
DB 100 [5] 0h,1h,2h,0a5a5a5a5h,720h
```

Istruzioni:

```
GET DB 100 ; Trasferisce il Blocco Dati 100
R 1000 ; nei Registri da 1000 in poi.
```

Risultato:

| Registro | Valore in Esad. |
|----------|-----------------|
| 1000 | 00000000 |
| 1001 | 00000001 |
| 1002 | 00000002 |
| 1003 | a5a5a5a5 |
| 1004 | 00000720 |

Esempio 2:

Testo come dichiarato nel programma sorgente:

```
TEXT 100 "THIS IS A TEXT 123"
```

Istruzioni:

```
GET X 100 ; Trasferisce il Testo 100
R 1000 ; nei Registri da 1000 in poi.
```

Risultato:

| Registro | Valore in ASCII | Value in Esad. |
|----------|-----------------|----------------|
| 1000 | THIS | 54484953 |
| 1001 | IS | 20495320 |
| 1002 | A TE | 41205445 |
| 1003 | XT 1 | 58542031 |
| 1004 | 23 | 32332020 |

PUT SCRITTURA DATI (Put Data)

Descrizione Le istruzioni COPY, GET e PUT hanno delle caratteristiche in comune. Sono utilizzate per trasferimenti indicizzati tra Registri, Temporizzatori e Contatori. In ogni caso viene copiato un intero Registro Temporizzatore o Contatore.

Nel caso di **PUTX**, il contenuto del primo operando viene copiato nel secondo; solo l'operando destinazione (secondo) è indicizzato.

Inoltre, l'istruzione PUT consente il trasferimento da Registri/Temporizzatori/Contatori in Testi/Blocchi di Dati.

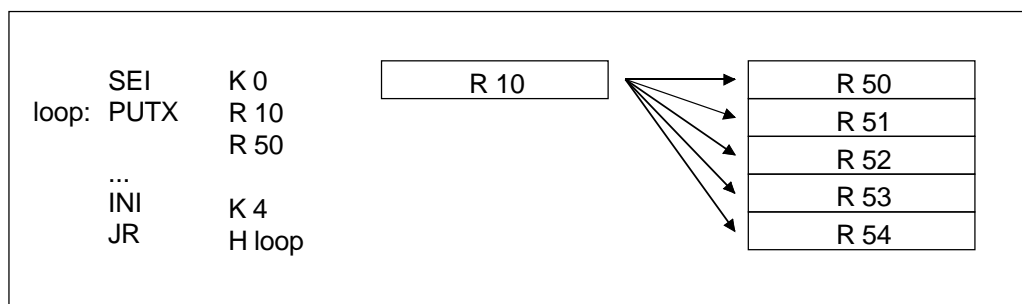
| | |
|-----------------|--|
| Utilizzo | PUT[X] sorgente ; R 0-4095, T 0-450, C 0-1599 destinazione (i) ; R 0-4095, T 0-450, C 0-1599, X 0-7999, DB 0-7999 |
|-----------------|--|

Esempio PUTX R 10 ; Trasferisce il contenuto del Registro 10
 R 50 ; nel Registro (50 + Indice)

Flag I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore copiato.

Vedere anche GET, COPY, MOV

Uso pratico



Trasferimento tra R/T/C e Testo/Blocco Dati

L'istruzione PUT[X] copia il contenuto del gruppo di R/T/C (a partire dal R/T/C specificato) nel testo indicato, fino a fine Testo. Se nel R/T/C è presente un carattere NUL (00H), l'istruzione PUT trasforma tale valore in uno spazio. Analogamente, PUT[X] copia il contenuto del gruppo di R/T/C (a partire dal R/T/C specificato) nel Blocco Dati specificato fino al riempimento del Blocco Dati stesso. Un BLOCCO DATI è un'area di memoria utente in cui si può memorizzare "run time" un elevato numero di Registri, Temporizzatori o Contatori. I Blocchi Dati possono essere utilizzati per memorizzare valori che sono specifici per un determinato processo e quindi liberare dei R/C/T che possono venire utilizzati da altri processi. Per la dichiarazione dei formati Testo/Blocco Dati vedere l'istruzione GET.

Se l'istruzione tenta di copiare da un R/T/C inesistente, viene chiamato l'XOB 13 (e viene attivato il Flag di Errore). Se il numero del Testo o del Blocco Dati indicizzato è maggiore di 7999, viene chiamato l'XOB 12 (superamento della capacità - overflow - del Registro Indice).

SCRITTURA (PUT) in un Blocco Dati/Testo

NOTA:

L'istruzione PUT non può estendere la lunghezza del testo o del blocco dati. Inoltre, l'istruzione PUT non può trasferire valori in un testo o in un blocco dati se vengono utilizzate EPROM o RAM con il ponticello in posizione "WP".

Esempio 1:

Blocco Dati come dichiarato nel programma sorgente:

```
DB 100 [5] ; I valori iniziali sono uguali a zero
```

Contenuto dei Registri:

| Registro | Valore in Dec |
|----------|---------------|
| 1000 | 00000001 |
| 1001 | 00000002 |
| 1002 | 00000003 |
| 1003 | 01234567 |
| 1004 | 00000720 |

Istruzioni:

```
PUT R 1000 ;Trasferisce il Registro 1000 e i successivi
DB 100 ; nl Blocco Dati 100.
```

Risultato visualizzato con il Debugger in formato Decimale:

```
DB 100 (0): 1 2 3 1234567 720
```

Esempio 2:

Testo come dichiarato nel programma sorgente:

```
TEXT 100 [17] ; Testo composto da 17 spazi
```

Contenuto dei Registri:

| Registro | Valore in ASCII | Valore in Esad. |
|----------|-----------------|-----------------|
| 1000 | THIS | 54484953 |
| 1001 | IS | 20495320 |
| 1002 | A TE | 41205445 |
| 1003 | XT 1 | 58542031 |
| 1004 | 23 | 32332020 |

Istruzioni:

```
PUT R 1000 ; Trasferisce il Registro 1000 e i successivi
X 100 ; nel Testo 100
```

Risultato visualizzato con il Debugger:

```
TEXT 100: "THIS IS A TEXT 12"
(il testo si interrompe dopo 17 caratteri)
```

TFR TRASFERIMENTO DATI (Transfer Data)

Descrizione Questa istruzione permette il trasferimento indicizzato di singoli dati da un Blocco Dati a Registri, Temporizzatori o Contatori e viceversa.

Utilizzo **TRASFERIMENTO BLOCCO DATI (o Testo) → R, T, C**
 Copia di un singolo valore (32 bit) da un Blocco Dati in un Registro, Temporizzatore o Contatore:

| | | |
|---------------|-------------------------|--|
| TFR[X] | sorgente | ; DB (o X) 0..3999, 4000..7999 |
| | posizione | ; R 0..4095, K 0..382, 0..16383 |
| | destinazione (i) | ; R 0..4095, T C 0..1599 |

Il primo operando è il Blocco Dati (o Testo) contenente il valore da trasferire.
 Il secondo operando è la posizione del valore all'interno del Blocco Dati (o Testo); tale posizione può essere fornita sia direttamente per mezzo di una costante sia indirettamente per mezzo di un Registro.

Il terzo operando è il Registro, Temporizzatore o Contatore di destinazione

TRASFERIMENTO R, T, C → BLOCCO DATI (o Testo)
 Copia di un Registro, Temporizzatore o Contatore in un Blocco Dati (o Testo)

| | | |
|---------------|---------------------|--|
| TFR[X] | sorgente (i) | ; R0..4095, T C 0..1599 |
| | destinazione | ; DB (o X) 0..3999, 4000..7999 |
| | posizione | ; R 0..4095, K 0..382, 0..16383 |

Il primo operando è il Registro, Timer o Contatore contenente il valore da trasferire (sorgente).

Il secondo operando è il Blocco Dati (o Testo) di destinazione.

Il terzo operando è la posizione, all'interno del Blocco Dati (o Testo) in cui deve essere trasferito il valore; tale posizione può essere fornita sia direttamente per mezzo di una costante sia indirettamente per mezzo di un Registro.

Nota:

La lunghezza del Blocco Dati dipende dal tipo di memoria del PCD:

| Memoria | Indirizzo | Lunghezza massima di 1 DB |
|----------|---------------|---------------------------|
| Standard | DB 0..3999 | 383 valori |
| Estesa | DB 4000..7999 | 16383 valori |

Esempio

TFR DB 4010 ; Copia dal Blocco Dati 4010
 K 13 ; il valore della posizione 13
 R 26 ; nel Register 26

TFR R 120 ; Copia il Registro 120
 DB 4025 ; nel Blocco Dati 4025
 K 6 ; alla posizione 6

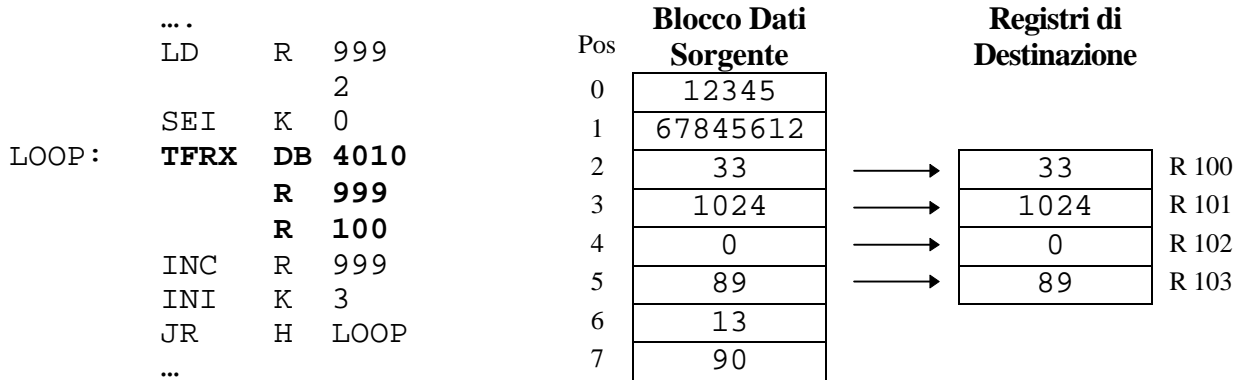
Flag I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore copiato.

Vedere anche PUT, GET

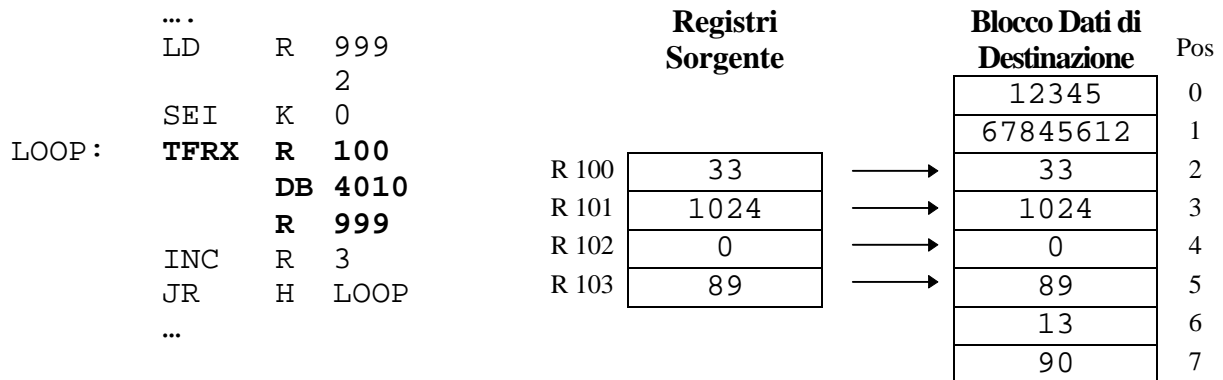
Nota Per motivi di organizzazione della memoria, l'accesso ai DB 4000..7999 è notevolmente più veloce rispetto ai DB 0..3999. Pertanto è consigliabile l'impiego

di questa istruzione principalmente per i DB 4000 ..7999.

Uso pratico I 4 valori contenuti nelle posizioni 2..5 del DB 4010 vengono copiati nei Registri 100..103.



Esempio 1: I Registri 100..103 vengono copiati nelle posizioni 2..5 del DB 4010:



TFRI TRASFERIMENTO DATI INDIRETTO (Transfer Data Indirect)

Descrizione Questa istruzione permette il trasferimento, mediante indirizzamento indiretto, di singoli valori da un Blocco Dati a Testi, Registri, Temporizzatori o Contatori e viceversa.
Questa istruzione non può operare in modo parametrizzato.

Utilizzo **TRASFERIMENTO BLOCCO DATI (o Testo) → R, T, C**
 Copia di un singolo valore (32 bit) da un Blocco Dati in un Registro, Temporizzatore o Contatore:

| | | |
|-------------|---------------------|--------------------------------|
| TFRI | sorgente | ; DB (o X) reg1 |
| | posizione | ; R 0..4095, K 0..16383 |
| | destinazione | ; R o T C reg2 |

Il primo operando specifica che la sorgente è un Blocco Dati o un Testo mentre la variabile reg1 indica il numero di un Registro contenente l'indirizzo del Blocco Dati (DB) o del Testo in oggetto.

Il secondo operando indica la posizione del valore all'interno del Blocco Dati (o del Testo); questa posizione può essere fornita sia direttamente, per mezzo di una costante, sia indirettamente, per mezzo di un Registro.

Il terzo operando specifica il tipo di destinazione (R o T|C). La variabile reg2 indica il numero di un Registro contenente il numero di destinazione del dato.

TRASFERIMENTO R, T, C → BLOCCO DATI (o Testo)
 Copia di un Registro, Temporizzatore o Contatore in un Blocco Dati (o Testo)

| | | |
|-------------|---------------------|--------------------------------|
| TFRI | sorgente | ; R o T C reg1 |
| | destinazione | ; DB (o X) reg2 |
| | posizione | ; R 0..4095, K 0..16383 |

Il primo operando specifica il tipo di sorgente (R o T|C). La variabile reg1 indica il numero di un Registro contenente il numero di destinazione del dato.

Il secondo operando specifica il tipo di destinazione (Blocco Dati o Testo). La variabile reg2 indica il numero di un Registro contenente l'indirizzo della destinazione.

Il terzo operando è la posizione, all'interno del Blocco Dati (o Testo) in cui deve essere trasferito il valore; tale posizione può essere fornita sia direttamente per mezzo di una costante sia indirettamente per mezzo di un Registro.

Nota:

La lunghezza del Blocco Dati dipende dal tipo di memoria del PCD:

| Memoria | Indirizzo | Lunghezza massima di 1 DB |
|----------------|------------------|----------------------------------|
| Standard | DB 0..3999 | 383 valori |
| Estesa | DB 4000..7999 | 16383 valori |

Esempi

Trasferisce l'elemento posto nella posizione 10 del Blocco Dati 4000 nel Registro 4095.

```
LD      R  100      ; Inizializzazione indirizzo DB
          4000
```

```
LD      R  101      ; Inizializzazione indirizzo Registro
          4095
```

```
TFRI    DB 100      ; Trasferimento DB
          K  10
          R  101
```

Trasferisce il contenuto del Contatore 1000 nella posizione 50 del Blocco Dati 4000

```
LD      R  100      ; Inizializzazione indirizzo DB
          4000
```

```
LD      R  101      ; Inizializzazione posizione
          50
```

```
LD      R  102      ; Inizializzazione indirizzo Contatore
          4095
```

```
TFRI    C 102      ; Trasferimento Contatore
          DB 100
          R  101
```

Flag

I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore copiato.

Vedere anche

TFR, PUT, GET

Nota

Per motivi di organizzazione della memoria, l'accesso ai DB 4000..7999 è notevolmente più veloce rispetto ai DB 0..3999. Pertanto è consigliabile l'impiego di questa istruzione principalmente per i DB 4000 ..7999.

BITI LETTURA BIT (Bit In)

Descrizione Trasferisce una sequenza binaria di bit da Ingressi, Uscite, Flag, Temporizzatori o Contatori in un Registro.
 Il 1° operando è il numero di bit da trasferire (1-32).
 Il 2° operando è la sorgente (I, O, F, T o C).
 Il 3° operando è il Registro di destinazione.
 Se la sorgente è costituita da Ingressi, Uscite o Flag, l'indirizzo sorgente fornito è l'elemento più basso della sequenza.
 L'elemento PIÙ BASSO diventa il bit MENO SIGNIFICATIVO del Registro di destinazione.
 Ciò è il contrario di quanto avviene nel SAIA PCA..

| | |
|-----------------|--|
| Utilizzo | BITI[X] bit ; Numero di bit da leggere 1-32 sorgente ; Indirizzo sorgente dati I, O, F, T, C dest (i) ; Numero Registro destinazione R 0-4095 |
|-----------------|--|

Esempio BITI 16 ; Legge 16 bit
 I 32 ; dagli Ingressi 32-47 e
 R 0 ; li memorizza nel Registro 0, bit 0-15

Flag I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore letto.

Vedere anche BITIR, DIGI, DIGIR

Uso pratico Quando l'ingresso 8 diventa Alto, un valore binario di 8 bit viene letto dagli Ingressi 0 - 7 e memorizzato nel Registro 500

```

COB            0        ; Inizio COB
                  0
                  . . . . .
STH            I 8        ; Se l'ingresso 8 diventa H (Alto)
DYN            F 100
JR             L NEXT
BITI            8        ; Allora leggi 8 bit binari
                  I 0        ;            dall'ingresso 0 (fino a 7)
                  R 500      ;            e memorizzali in R 500
NEXT:           . . . .
                  ECOB
    
```

BITIR LETTURA BIT REVERSE (Bit In Reversed)

Descrizione Trasferisce una sequenza binaria di bit da Ingressi, Uscite, Flag, Temporizzatori o Contatori in un Registro.
 Il 1° operando è il numero di bit da trasferire (1-32).
 Il 2° operando è la sorgente (I, O, F, T o C).
 Il 3° operando è il Registro di destinazione.
 Se la sorgente è costituita da Ingressi, Uscite o Flag, l'indirizzo sorgente fornito è l'elemento più basso della sequenza.
 L'elemento PIU' BASSO diventa il bit PIU' SIGNIFICATIVO del Registro di destinazione.
 Ciò è analogo a quanto avviene nel SAIA PCA.

Utilizzo

| | | |
|-----------------|-----------------|--|
| BITIR[X] | bit | ; Numero di bit da leggere 1-32 |
| | sorgente | ; Indirizzo sorgente dati I, O, F, T, C |
| | dest (i) | ; Numero Registro destinazione R 0-4095 |

Esempio

BITIR 16 ; Legge 16 bit
 I 32 ; dagli Ingressi 32-47 e
 R 0 ; li memorizza nel Registro 0, bit 15-0

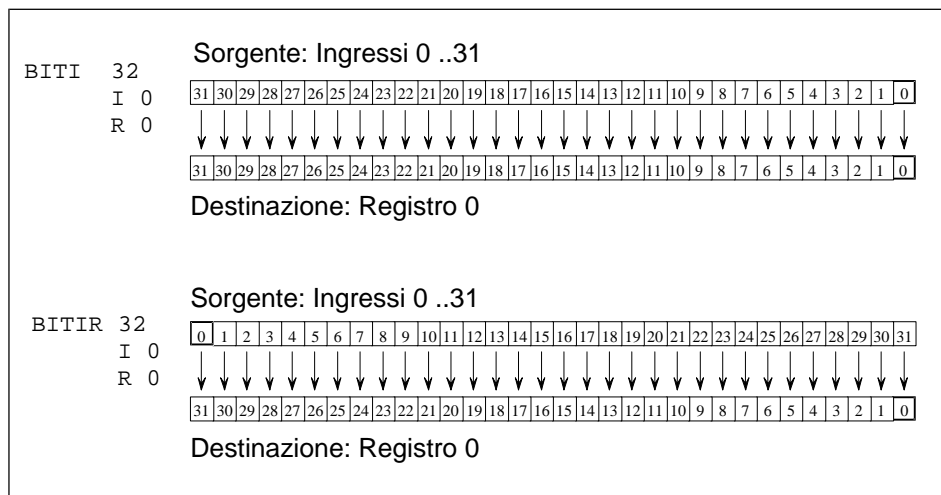
Flag

I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore letto.

Vedere anche

BITI, DIGI, DIGIR

Uso pratico



BITO USCITA BIT (Bit Out)

Descrizione Trasferisce una sequenza binaria di bit da un Registro ai corrispondenti Flag, Uscite, o bit di un Temporizzatore o Contatore.
 Il 1° operando è il numero di bit da trasferire (1-32).
 Il 2° operando è il numero del Registro sorgente.
 Il 3° operando rappresenta le Uscite, i Flag, il Temporizzatore o il Contatore di destinazione.
 Se la destinazione è rappresentata da Uscite o Flag, l'indirizzo fornito è quello dell'elemento più basso della sequenza.
 Il bit MENO SIGNIFICATIVO del Registro viene trasferito nell'elemento PIU' BASSO.
 Ciò è il contrario di quanto avviene nel SAIA PCA

| | | | |
|-----------------|----------------|------------------------------|---|
| Utilizzo | BITO[X] | bit sorgente (i) dest | ; Numero di bit da trasferire 1-32 ; Registro sorgente R 0-4095 ; Indirizzo destinazione dati O, F, T, C |
|-----------------|----------------|------------------------------|---|

Esempio BITO 8 ; Trasferisce 8 bit
 R 10 ; dal Registro 10, bit 0-7
 O 48 ; e li emette sulle Uscite 48-55

Flag Invariati.

Vedere anche BITOR, DIGO, DIGOR

Uso pratico Copiare lo stato degli ingressi 0 → 15 sulle uscite 32 → 47

```

COB      0      ; Inizio COB
          0

BITI     16     ; Leggi 16 bit
          I 0    ; dall'ingresso 0 (fino al 15)
          R 0    ; e memorizzali in R 0

BITO     16     ; Scrivi 16 bit
          R 0    ; da R 0
          O 32   ; alle uscite 32 - 47

          ECOB
    
```

BITOR USCITA BIT REVERSESE (Bit Out Reversed)

Descrizione Trasferisce una sequenza binaria di bit da un Registro ai corrispondenti Flag, Uscite, o bit di un Temporizzatore o Contatore.
 Il 1° operando è il numero di bit da trasferire (1-32).
 Il 2° operando è il numero del Registro sorgente.
 Il 3° operando rappresenta le Uscite, i Flag, il Temporizzatore o il Contatore di destinazione.
 Se la destinazione è rappresentata da Uscite o Flag, l'indirizzo fornito è quello dell'elemento più basso della sequenza.
 Il bit MENO SIGNIFICATIVO del Registro viene trasferito nell'elemento PIU' ALTO.
 Ciò è analogo a quanto avviene nel SAIA PCA.

Utilizzo

| | | |
|-----------------|---------------------|---|
| BITOR[X] | bit | ; Numero di bit da trasferire 1-32 |
| | sorgente (i) | ; Registro sorgente R 0-4095 |
| | dest | ; Indirizzo destinazione dati O, F, T, C |

Esempio

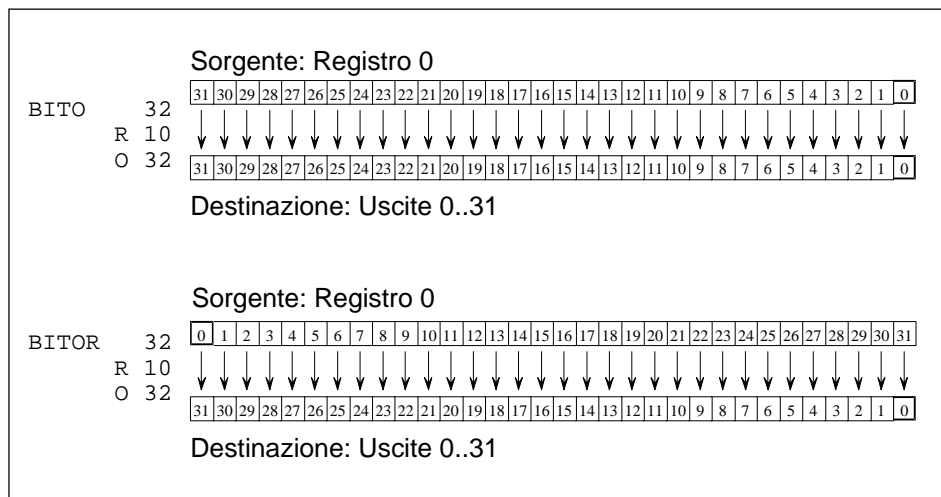
```

BITOR      8      ; Trasferisce 8 bit
           R 10    ; dal Registro 10, bit 0-7
           O 48    ; e li emette sulle Uscite 55-48
    
```

Flag Invariati.

Vedere anche BITO, DIGO, DIGOR

Uso pratico



DIGI LETTURA DIGIT (Digit In)

Descrizione Trasferisce digit Binari Codificati in Decimale (BCD) da Ingressi, Uscite o Flag in un Registro.

Un digit BCD è costituito da 4 bit (es. 4 Ingressi), che rappresentano una cifra decimale (0-9).

Il 1° operando è il numero di digit da trasferire (1-10).

Il 2° operando è l'Ingresso, l'Uscita o il Flag di base.

Il 3° operando è il Registro di destinazione.

L'Ingresso, l'Uscita o il Flag all'indirizzo più basso diventa il bit meno significativo del digit meno significativo del Registro di destinazione.

Ciò è il contrario di quanto avviene nel SAIA PCA..

Utilizzo

DIGI[X] **digit** ; Numero di digit BCD 1-10
 sorgente ; Elemento sorgente I 0-8191, O 0-8191, F 0-8191
 dest (i) ; Registro destinazione R 0-4095

Esempio

DIGI 2 ; Legge 2 digit BCD
 I 32 ; dagli Ingressi 39-36 e 35-32
 R 100 ; nel Registro 100

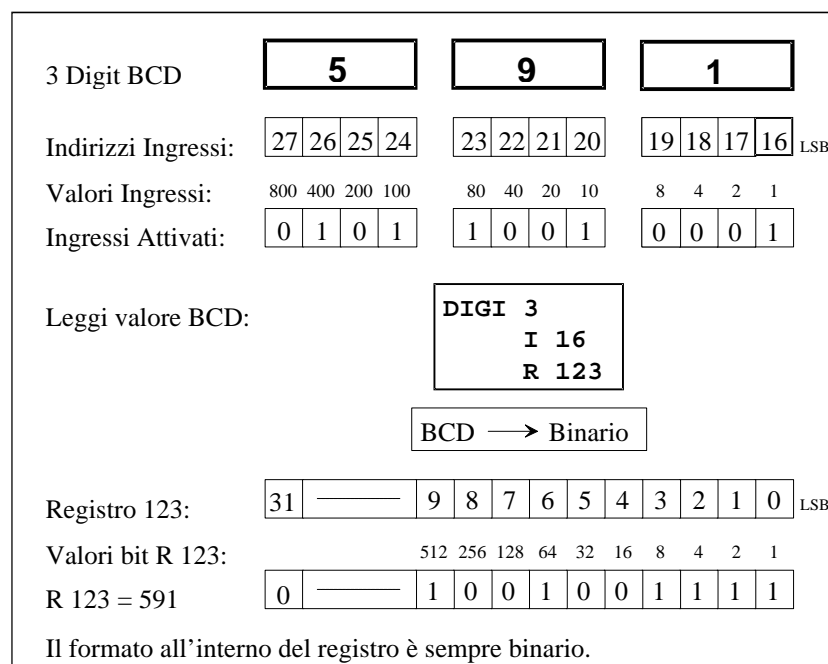
Flag

I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore letto.

Vedere anche

DIGIR, DIGO, DIGOR, BITI, BITIR

Uso pratico



DIGIR LETTURA DIGIT REVERSE (Digit In Reversed)

Descrizione Trasferisce digit Binari Codificati in Decimale (BCD) da Ingressi, Uscite o Flag in un Registro.
 Un digit BCD è costituito da 4 bit (es. 4 Ingressi), che rappresentano una cifra decimale (0-9).
 Il 1° operando è il numero di digit da trasferire (1-10).
 Il 2° operando è l'Ingresso, l'Uscita o il Flag di base.
 Il 3° operando è il Registro di destinazione.
 L'Ingresso, l'Uscita o il Flag all'indirizzo più basso diventa il bit più significativo del digit più significativo del Registro di destinazione.
 Ciò è analogo a quanto avviene nel SAIA PCA

Utilizzo

| | | |
|-----------------|--------------------------------|---|
| DIGIR[X] | digit sorgente dest (i) | ; Numero di digit BCD 1-10 ; Elemento sorgente I 0-8191, O 0-8191, F 0-8191 ; Registro destinazione R 0-4095 |
|-----------------|--------------------------------|---|

Esempio

DIGIR 2 ; Legge 2 digit BCD
 I 32 ; dagli Ingressi 32-35 e 36-39
 R 100 ; nel Registro 100

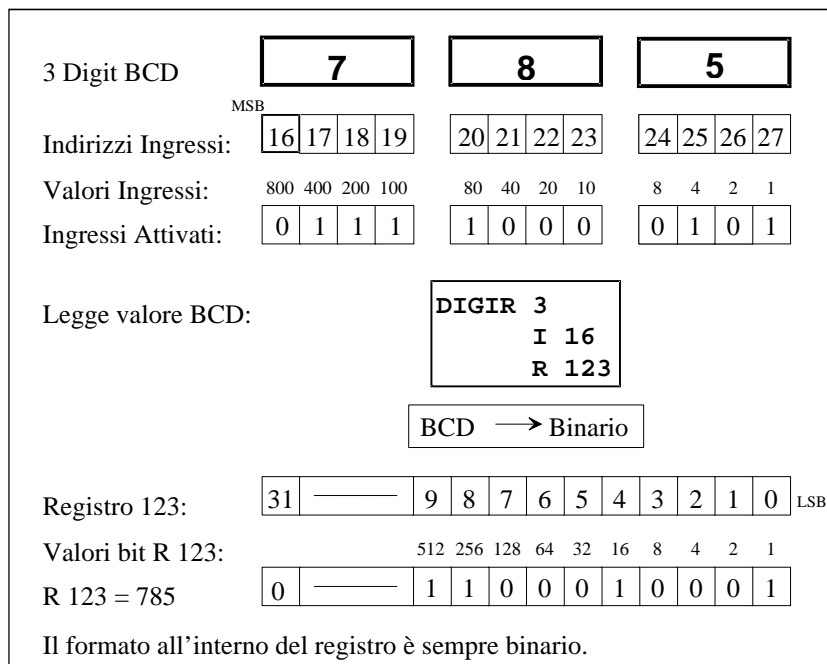
Flag

I flag **Zero (Z)** e **Segno (P o N)** sono impostati in base al valore letto.

Vedere anche

DIGI, DIGO, DIGOR, BITI, BITIR

Uso pratico



DIGO USCITA DIGIT (Digit Out)

Descrizione Trasferisce digit BCD da un Registro ad una sequenza di Uscite o di Flag.
 Un digit BCD è costituito da 4 bit binari.
 Il 1° operando è il numero di digit da trasferire.
 Il 2° operando è il Registro sorgente.
 Il 3° operando è l'indirizzo dell'Uscita o del Flag di base.
 L'Uscita o il Flag all'indirizzo più basso diventa il bit meno significativo del digit BCD meno significativo.
 Ciò è il contrario di quanto avviene nel SAIA PCA.

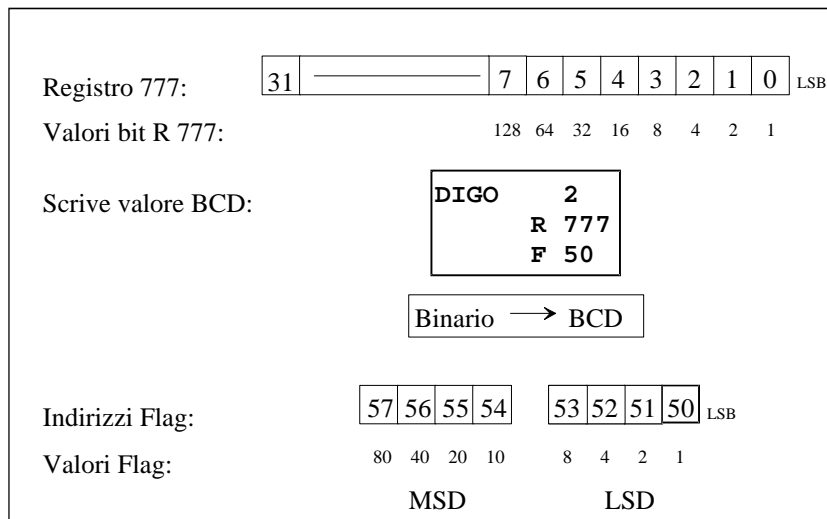
| | |
|-----------------|--|
| Utilizzo | DIGO[X] digit ; Numero digit BCD 1-10 sorgente (i) ; Registro sorgente R 0-4095 dest ; Elemento destinazione O 0-8191, F 0-8191 |
|-----------------|--|

Esempio DIGO 2 ; Emette 2 digit BCD
 R 123 ; dal Registro 123
 O 40 ; sulle Uscite 47-44 e 43-40

Flag Il flag **Errore (E)** viene impostato in caso di digit BCD errato (> 9).

Vedere anche DIGOR, DIGI, DIGIR, BITO, BITOR

Uso pratico



DIGOR USCITA DIGIT REVERSE (Digit Out Reversed)

Descrizione Trasferisce digit BCD da un Registro ad una sequenza di Uscite o di Flag.
 Un digit BCD è costituito da 4 bit binari.
 Il 1° operando è il numero di digit da trasferire.
 Il 2° operando è il Registro sorgente.
 Il 3° operando è l'indirizzo dell'Uscita o del Flag di base.
 L'Uscita o il Flag all'indirizzo più basso diventa il bit più significativo del digit BCD più significativo.
 Ciò è analogo a quanto avviene nel SAIA PCA..

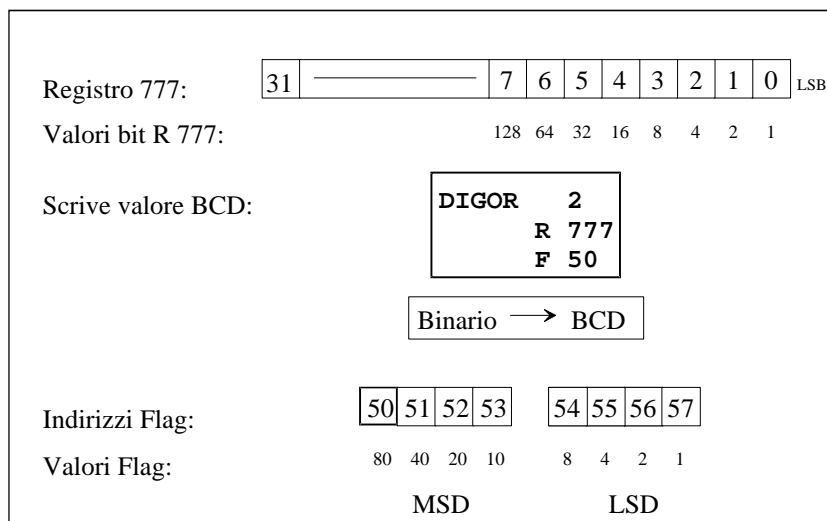
| | | | |
|-----------------|-----------------|---------------------|---|
| Utilizzo | DIGOR[X] | digit | ; Numero digit BCD 1-10 |
| | | sorgente (i) | ; Registro sorgente R 0-4095 |
| | | dest | ; Elemento destinazione O 0-8191, F 0-8191 |

Esempio DIGOR 2 ; Emette 2 digit BCD
 R 123 ; dal Registro 123
 O 40 ; sulle Uscite 40-43 e 44-47

Flag Il flag **Errore (E)** viene impostato in caso di digit BCD errato (> 9).

Vedere anche DIGO, DIGI, DIGIR, BITO, BITOR

Uso pratico



AND AND TRA REGISTRI (AND Registers)

Descrizione Viene eseguito l'AND logico tra il contenuto del 1° Registro e il contenuto del 2° Registro e il risultato viene posto nel 3° Registro..

Utilizzo

AND [X] valore1 (i) ; R 0-4095
valore2 ; R 0-4095
risultato (i) ; R 0-4095

Esempio

AND R 11 ; esegue l'AND tra R 11 e
 R 12 ; R 12 e pone il risultato
 R 13 ; in R 13.

I bit di R 13 sono a 1 quando i corrispondenti bit in R 11 E (AND) R 12 sono entrambi a 1

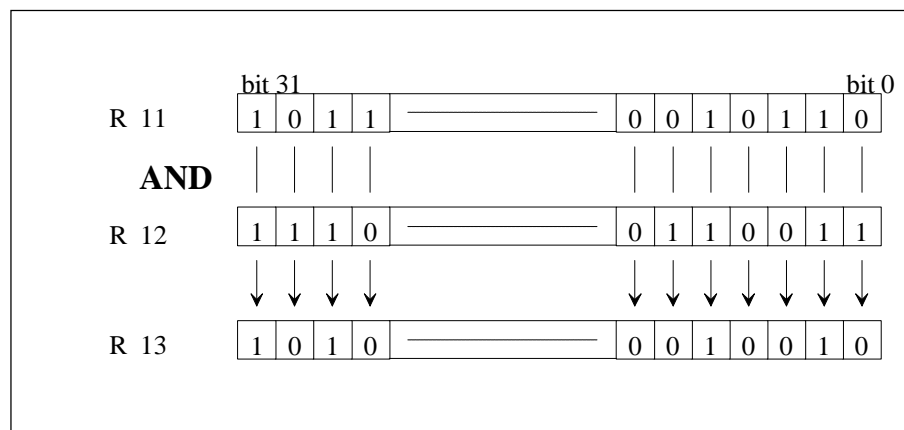
Flag

I flag **Zero (Z)** e **Segno (N o P)** sono impostati in base al risultato.
 Il flag **Errore (E)** è sempre impostato al livello basso (L).

Vedere anche

OR, NOT, EXOR

Uso pratico



OR OR TRA REGISTRI (OR Registers)

Descrizione Viene eseguito l'OR logico tra il contenuto del 1° Registro e il contenuto del 2° Registro e il risultato viene posto nel 3° Registro.

Utilizzo

| | | |
|--------------|----------------------|-------------------|
| OR[X] | valore1 (i) | ; R 0-4095 |
| | valore2 | ; R 0-4095 |
| | risultato (i) | ; R 0-4095 |

Esempio

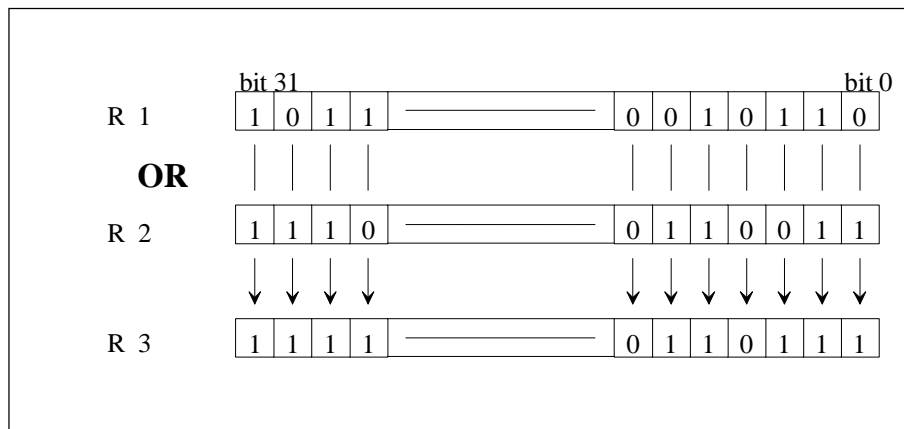
OR R 1 ; esegue l'OR tra il Registro 1
 R 2 ; e il Registro 2
 R 3 ; e pone il risultato nel Registro 3

Flags

I flag **Zero (Z)** e **Segno (N o P)** sono impostati in base al risultato.
 Il flag **Errore (E)** è sempre impostato al livello basso (L).

Vedere anche EXOR

Practice



EXOR OR ESCLUSIVO TRA REGISTRI (Exclusive-OR Registers)

Descrizione Viene eseguito l'OR Esclusivo tra il contenuto del 1° Registro e il contenuto del 2° Registro e il risultato viene posto nel 3° Registro.

Utilizzo

| | | |
|----------------|----------------------|-------------------|
| EXOR[X] | valore1 (i) | ; R 0-4095 |
| | valore2 | ; R 0-4095 |
| | risultato (i) | ; R 0-4095 |

Esempio

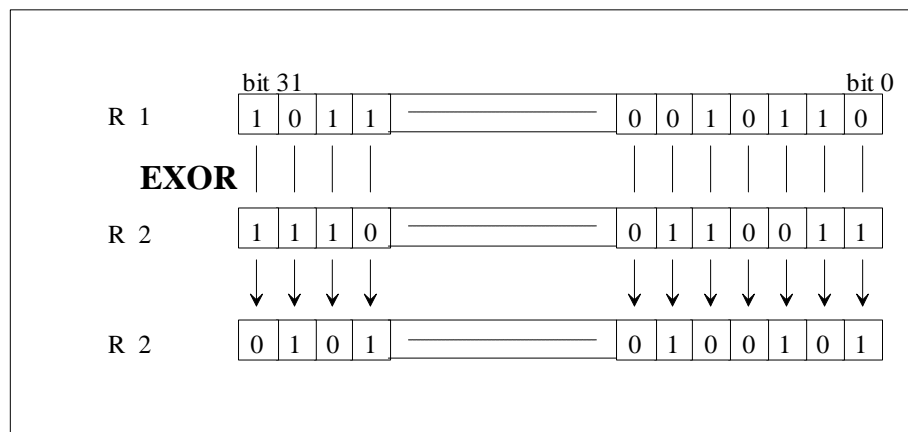
```
EXOR R 1 ; esegue l'OR Esclusivo tra il Registro 1
      R 2 ; e il Registro 2 e
      R 2 ; pone il risultato nel Registro 2
```

Flag

I flag **Zero (Z)** e **Segno (N o P)** sono impostati in base al risultato. Il flag **Errore (E)** è sempre impostato al livello basso (L).

Vedere anche OR

Uso pratico



NOT COMPLEMENTO REGISTRO (Complemento Register)

Descrizione Il contenuto del 1° Registro viene invertito (complemento a 1) e memorizzato nel 2° Registro.

Utilizzo

| | | |
|---------------|----------------------|-------------------|
| NOT[X] | valore (i) | ; R 0-4095 |
| | risultato (i) | ; R 0-4095 |

Esempio

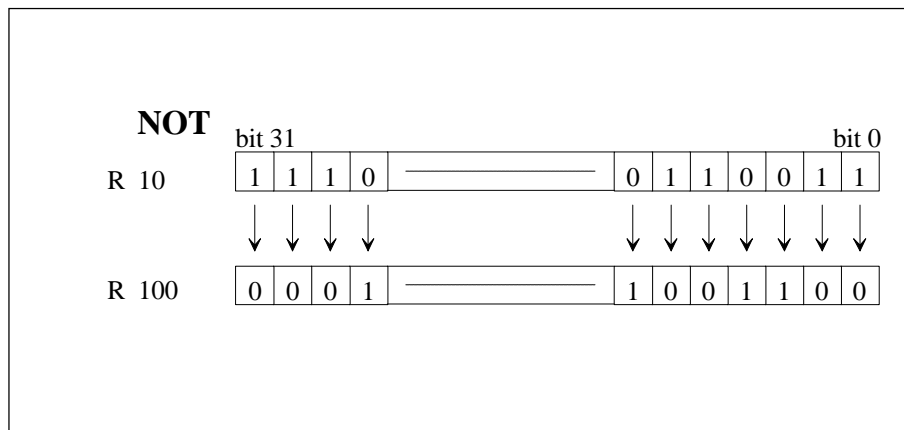
NOT R 10 ; Inverte il contenuto del Registro 10
 R 100 ; e pone il risultato nel Registro 100

Flag

I flag **Zero (Z)** e **Segno (N o P)** sono impostati in base al risultato.
 Il flag **Errore (E)** è sempre impostato al livello basso (L).

Vedere anche

Uso pratico



SHIU **SHIFT AVANTI DI REGISTRI** (Shift Registers Up)

Descrizione Sposta il contenuto di un blocco dei Registri di una posizione in avanti. Il 1° e il 2° operando rappresentano l'inizio e la fine del blocco di Registri da spostare. Dopo lo shift, il Registro più basso contiene zero e il più alto ricopre il Registro successivo. Si può specificare per primo il Registro più alto o quello più basso.

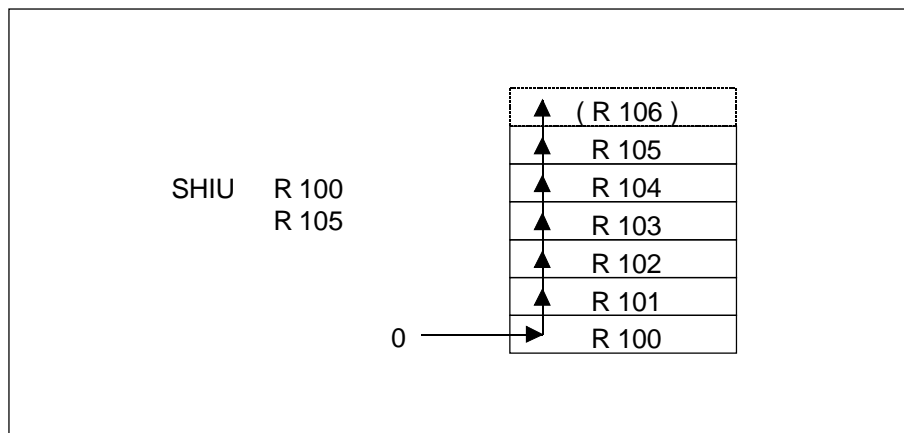
| | | | |
|-----------------|-------------|-----------------------|-------------------|
| Utilizzo | SHIU | inizio (start) | ; R 0-4094 |
| | | fine (end) | ; R 0-4094 |

Esempio SHIU R 100 ; Sposta avanti di un indirizzo da R 100 a R 105
R 105
; R 100 = 0, R 101 = R 100 ... R 106 = R 105

Flag Invariati.

Vedere anche SHID, ROTU, ROTD

Uso pratico



NOTA:

Questa istruzione utilizza un ulteriore Registro oltre a quelli specificati: viene utilizzato infatti anche il Registro successivo alla fine del blocco.

SHID SHIFT INDIETRO DI REGISTRI (Shift Registers Down)

Descrizione Sposta il contenuto di un blocco dei Registri di una posizione indietro.
 Il 1° e il 2° operando rappresentano l'inizio e la fine del blocco di Registri da spostare.
 Dopo lo shift, il Registro più alto contiene zero e il più basso ricopre quello sottostante.
 Si può specificare per primo il Registro più alto o quello più basso

Utilizzo

| | | |
|-------------|-----------------------|-------------------|
| SHID | inizio (start) | ; R 0-4094 |
| | fine (end) | ; R 0-4094 |

Esempio

```
SHID    R 100 ; Sposta indietro di un indirizzo da R 100 a R 105
        R 105
        ; R 99 = R 100 ... R 104 = R 105, R 105 = 0
```

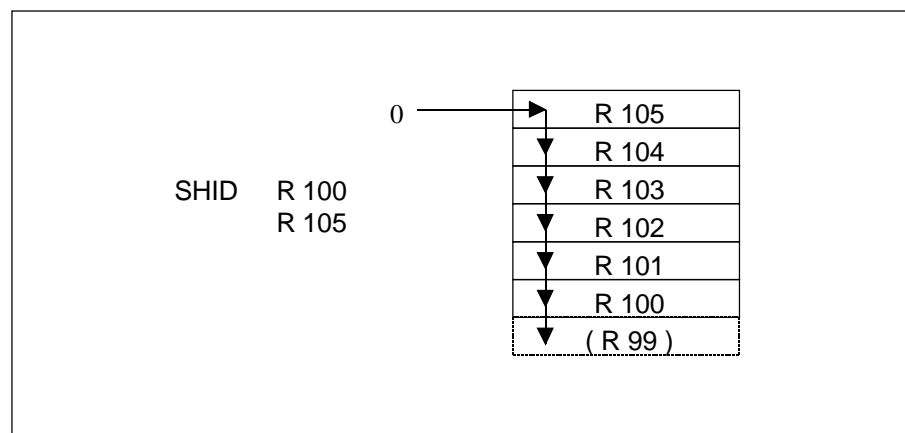
Flag

Invariati.

Vedere anche

SHIU, ROTU, ROTD

Uso pratico



NOTA:

Tale istruzione utilizza un ulteriore Registro oltre a quelli specificati: viene infatti utilizzato anche il Registro precedente l'inizio del blocco.

**ROTU ROTAZIONE AVANTI DI REGISTRI
(Rotate Registers Up)**

Descrizione Ruota il contenuto di un blocco di Registri di una posizione avanti.
 Il primo e il secondo operando indicano rispettivamente l'inizio e la fine del blocco di Registri da ruotare.
 Dopo la rotazione il Registro più basso contiene il valore del più alto.
 Si può specificare per primo il Registro più alto o quello più basso.

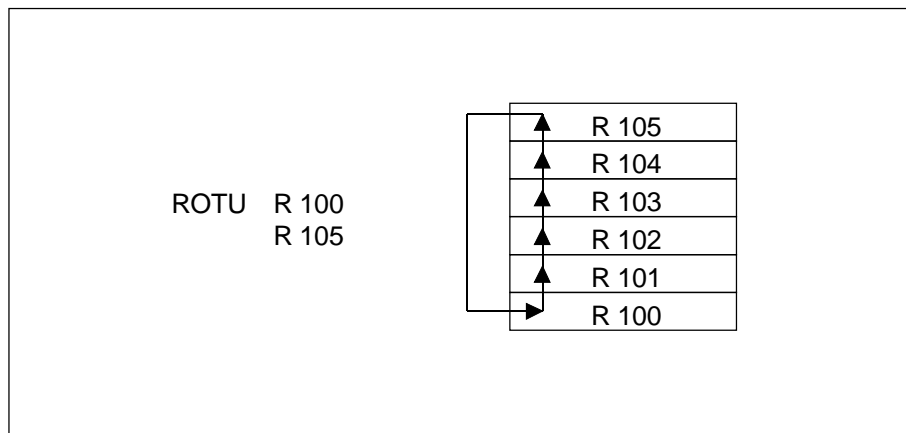
| | |
|-----------------|--|
| Utilizzo | ROTU inizio (start) ; R 0-4094 fine (end) ; R 0-4095 |
|-----------------|--|

Esempio ROTU R 100 ; Ruota avanti di un indirizzo da R 100 a R 105.
 R 105
 ; R 100 = R 105, R 101 = R100 ... R 105 = R 104

Flag Invariati.

Vedere anche ROTD, SHIU, SHID

Uso pratico



**ROTD ROTAZIONE INDIETRO DI REGISTRI
(Rotate Registers Down)**

Descrizione Ruota il contenuto di un blocco di Registri di una posizione indietro.
 Il primo e il secondo operando indicano rispettivamente l'inizio e la fine del blocco di Registri da ruotare.
 Dopo la rotazione il Registro più Alto contiene il valore del più basso.
 Si può specificare per primo il Registro più alto o quello più basso

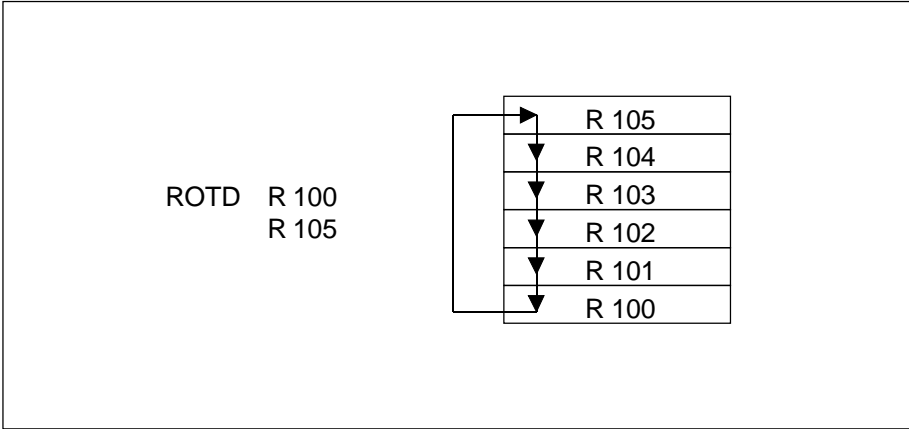
| | |
|-----------------|--|
| Utilizzo | ROTD inizio (start) ; R 0-4095 |
| | fine (end) ; R 0-4095 |

Esempio ROTD R 100 ; Ruota indietro di un indirizzo da R 100 a R 105.
 R 105
 ; R 100 = R 101 ... R 104 = R 105, R 105 = R 100.

Flag Invariati.

Vedere anche ROTU, SHIU, SHID

Uso pratico



SHIL SHIFT A SINISTRA DI UN REGISTRO (Shift Register Left)

Descrizione Il contenuto del Registro indirizzato viene fatto scorrere a sinistra di tanti bit quanti sono specificati nel secondo operando.
 Il contenuto dell'ACCU (1 o 0) viene inserito nella posizione del bit 0 (il bit meno significativo) e viene fatto scorrere a sinistra n volte.
 Alla fine dell'operazione, l'ACCU viene impostato con il valore dell'ultimo bit uscito dal Registro a seguito dello shift

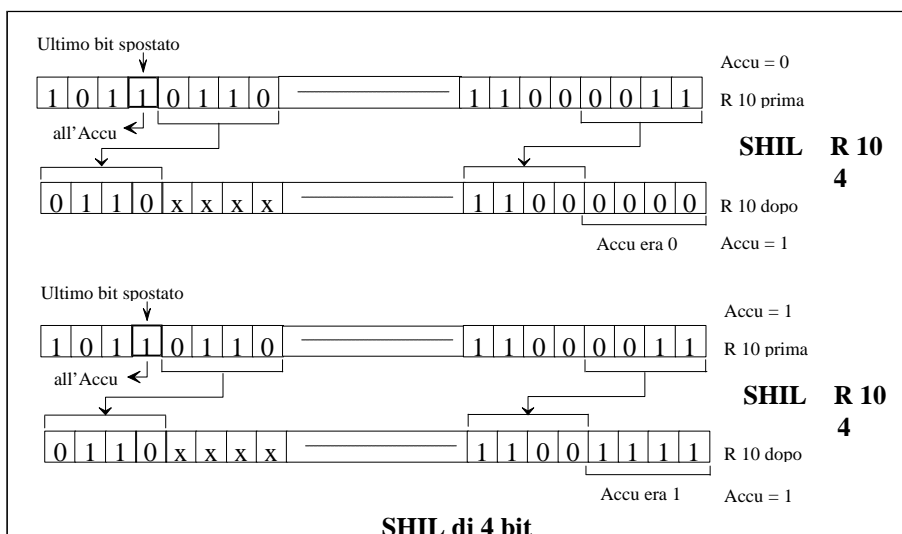
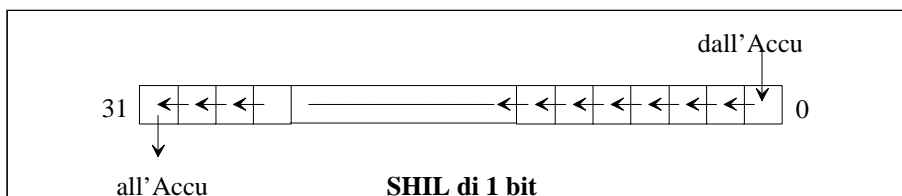
| | |
|-----------------|---|
| Utilizzo | SHIL[X] reg (i) ; R 0-4095 n bit ; Nuemro di bit 1-32 |
|-----------------|---|

Esempio SHIL R 10 ; Il Registro 10 viene fatto scorrere a sinistra
 4 ; di 4 bit
 ; (moltiplicato per 16 se ACCU era a 0)

Flag L'ACCU viene impostato con il valore dell'ultimo bit uscito dal Registro in seguito allo scorrimento (shift).

Vedere anche SHIR, ROTL, ROTR

Usò pratico



SHIR SHIFT A DESTRA DI UN REGISTRO (Shift Register Right)

Descrizione Il contenuto del Registro indirizzato viene fatto scorrere a destra di tanti bit quanti sono specificati nel secondo operando.
Il contenuto dell'ACCU (1 o 0) viene inserito nella posizione del bit 31 (il più significativo) e viene fatto scorrere a destra n volte.
Alla fine dell'operazione, l'ACCU viene impostato con il valore dell'ultimo bit uscito dal Registro a seguito dello shift.

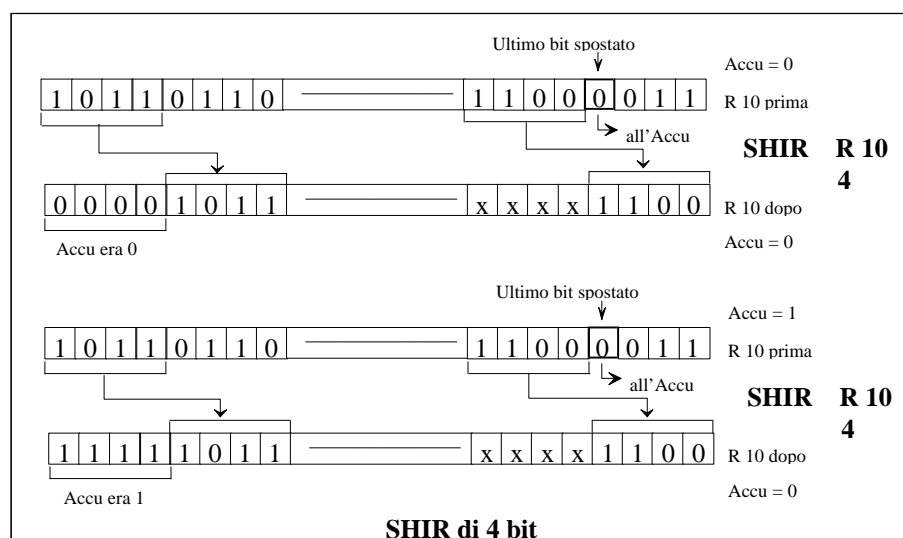
Utilizzo **SHIR[X] reg (i) ; R 0-4095**
n bit ; Numero di bit 1-32

Esempio SHIR R 10 ; Il Registro 10 viene spostato a destra
16 ; di 16 bit
; (diviso per 65536 se ACCU era a 0)

Flag L'ACCU viene impostato con il valore dell'ultimo bit uscito dal Registro in seguito allo scorrimento (shift).

Vedere anche SHIL, ROTL, ROTR

Usò pratico



ROTL ROTAZIONE A SINISTRA DI UN REGISTRO (Rotate Register Left)

Descrizione Il contenuto del Registro indirizzato viene ruotato a sinistra di tanti bit quanti sono specificati nel 2° operando.
 Il bit più significativo (bit 31) viene copiato nel bit meno significativo (bit 0).
 L'ACCU viene impostato con il valore dell'ultimo bit ruotato.

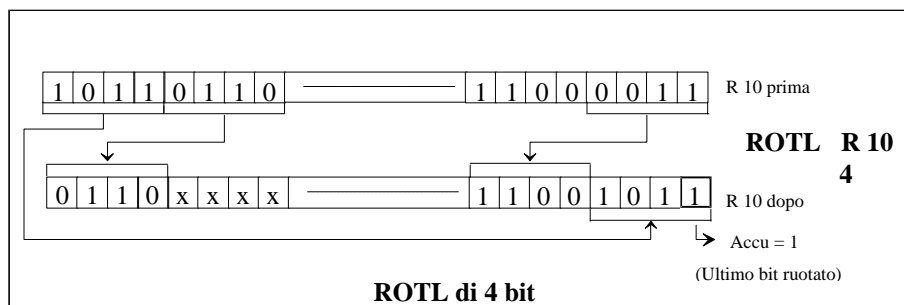
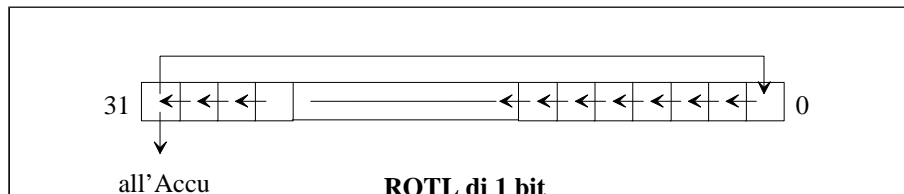
| | |
|-----------------|--|
| Utilizzo | ROTL[X] reg (i) ; R 0-4095 n bit ; Numero di bit 1-32 |
|-----------------|--|

Esempio `ROTL R 10 ;` Il Registro 10 viene ruotato a sinistra di 4 bit
4

Flag L'ACCU viene impostato con il valore dell'ultimo bit ruotato.

Vedere anche ROTR, SHIL, SHIR

Uso pratico



ROTR ROTAZIONE A DESTRA DI UN REGISTRO (Rotate Register Right)

Descrizione Il contenuto del Registro indirizzato viene ruotato a destra di tanti bit quanti sono specificati nel 2° operando.
 Il bit meno significativo (bit 0) viene copiato nel bit più significativo (bit 31).
 L'ACCU viene impostato con il valore dell'ultimo bit ruotato..

Utilizzo

ROTR[X] reg (i) ; R 0-4095
n bit ; Numero di bit 1-32

Esempio

ROTR R 10 ; Il Registro 10 viene ruotato a destra di 4 bit
 4

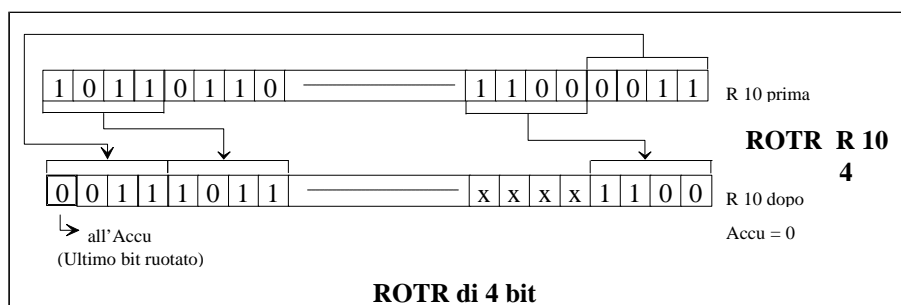
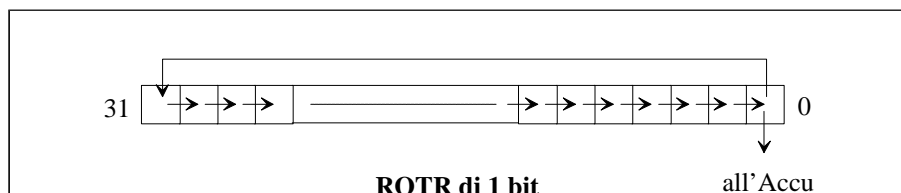
Flag

L'ACCU viene impostato con il valore dell'ultimo bit ruotato.

Vedere anche

ROTR, SHIL, SHIR

Uso pratico



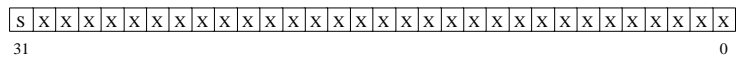
4. Istruzioni Aritmetiche in INTERO

Le istruzioni aritmetiche su interi operano solo a livello registri.

| | |
|-----|--|
| ADD | Add Registers (Somma di Registri) |
| SUB | Subtract Registers (Sottrazione tra Registri) |
| MUL | Multiply Registers (Moltiplicazione tra Registri) |
| DIV | Divide Registers (Divisione tra Registri) |
| SQR | Square Root (Radice Quadrata) |
| CMP | Compare Registers (Confronto tra Registri) |

Per i valori in virgola mobile, devono essere usate le specifiche istruzioni in virgola mobile.

Gli interi si basano sul seguente formato a 32 bit:



dove X: valore del bit (0 o 1)
S: segno

I bit da 0 a 30 rappresentano il valore intero in formato binario.

Il bit 31 è il bit di segno (0 = valore positivo, 1 = valore negativo)

Il campo di valori consentiti per questo formato è il seguente:

| | | |
|-----------------------|------------------|------------------|
| Decimale | da 2.147.483.647 | a -2.147.483.648 |
| Binario (esadecimale) | da 7FFF'FFFF | a 8000'0000 |

Formato INTERO

ADD SOMMA DI REGISTRI (ADD Registers)

Descrizione Somma di interi.
Somma il contenuto del primo Registro (o costante) al contenuto del secondo Registro (o costante), e memorizza il risultato nel terzo Registro.

| | |
|-----------------|---|
| Utilizzo | ADD valore1 ; R 0-4095, K 0-16383 |
| | valore2 ; R 0-4095, K 0-16383 |
| | risultato ; R 0-4095 |

Esempio ADD R 20 ; Somma 123 al Registro 20
 K 123
 R 20

Flag I flag **Z**ero (Z) e **S**egno (P o N) vengono impostati in base al risultato.
Il flag **E**rrore (E) viene impostato in caso di superamento della capacità del registro (overflow).

Vedere anche FADD (floating point add - somma in virgola mobile)

Uso pratico Leggere due numeri, sommarli e porre il risultato in un altro registro.
I due numeri provengono dai preselettori BCD (2 digit) rispettivamente collegati agli ingressi 16-23 e 24-31

```

COB            0      ; Inizio COB
                  0

DIGI           2      ; Legge 2 digit
                  I 16     ;    dall'ingresso 16 (fino a 23)
                  R 100    ;    e li memorizza in R 100

DIGI           2      ; Legge 2 digit
                  I 24     ;    dall'ingresso 24 (fino a 31)
                  R 200    ;    e li memorizza in R 200

ADD          R 100    ; R 0 = R 100 + R 200
                  R 200
                  R 0

                  ECOB
    
```

SUB SOTTRAZIONE TRA REGISTRI (SUBstract Registers)

Descrizione Sottrazione tra interi.
 Sottrae il contenuto del secondo Registro (o costante K) dal contenuto del primo Registro (o costante K) e memorizza il risultato nel terzo Registro

Utilizzo

| | | |
|------------|------------------|------------------------------|
| SUB | valore1 | ; R 0-4095, K 0-16383 |
| | valore2 | ; R 0-4095, K 0-16383 |
| | risultato | ; R 0-4095 |

Esempio SUB R 1 ; Registro 3 = Registro 1 meno Registro 2
 R 2
 R 3

Flag I flag **Zero (Z)** e **Segno (P o N)** vengono impostati in base al risultato.
 Il flag **Errore (E)** viene impostato in caso di superamento del valore minimo del registro (underflow).

Vedere anche ADD, FSUB

Uso pratico Leggere due numeri, sottrarli e porre il risultato in un altro registro.
 I due numeri provengono dai preselettori BCD (2 digit) rispettivamente collegati agli ingressi 16-23 e 24-31

 COB 0 ; Inizio COB
 0

 DIGI 2 ; Legge 2 digit
 I 16 ; dall'ingresso 16 (fino a 23)
 R 10 ; e li memorizza in R 10

 DIGI 2 ; Legge 2 digit
 I 24 ; dall'ingresso 24 (fino a 31)
 R 11 ; e li memorizza in R 11

SUB R 10 ; R 12 = R 10 - R 11
 R 11
 R 12

 ECOB

MUL MULTIPLICAZIONE TRA REGISTRI (MULTIply Registers)

Descrizione Moltiplicazione tra interi.
Esegue la moltiplicazione tra il contenuto del primo Registro (o costante K) e il contenuto del secondo Registro (o costante K) e memorizza il risultato nel terzo Registro

| | | | |
|-----------------|------------|------------------|------------------------------|
| Utilizzo | MUL | valore1 | ; R 0-4095, K 0-16383 |
| | | valore2 | ; R 0-4095, K 0-16383 |
| | | risultato | ; R 0-4095 |

Esempio MUL R 0 ; Moltiplica per 10 il Registro 0
 K 10
 R 0

Flags I flag **Zero (Z)** e **Segno (P o N)** vengono impostati in base al risultato.
Il flag **Errore (E)** viene impostato in caso di superamento della capacità del registro (overflow).

Vedere anche DIV, FMUL

Uso pratico Leggere due numeri, moltiplicarli e porre il risultato in un altro registro.

I due numeri provengono dai preselettori BCD (2 digit) rispettivamente collegati agli ingressi 16-23 e 24-31

COB 0 ; Inizio COB
0

DIGI 2 ; Legge 2 digit
I 16 ; dall'ingresso (fino a 23)
R 50 ; e li memorizza in R 50

DIGI 2 ; Legge 2 digit
I 24 ; dall'ingresso 24 (fino a 31)
R 55 ; e li memorizza in R 55

MUL R 50 ; R 4000 = R 50 * R 55
 R 55
 R 4000

ECOB

DIV DIVISIONE TRA REGISTRI (DIVide Registers)

Descrizione Divisione tra interi.
 Eseguce la divisione tra il contenuto del primo Registro (o costante K) e il contenuto del secondo Registro (o costante K) e memorizza il risultato nel terzo Registro. Il resto viene posto nel quarto registro.

| | | | |
|-----------------|------------|------------------|------------------------------|
| Utilizzo | DIV | valore 1 | ; R 0-4095, K 0-16383 |
| | | valore 2 | ; R 0-4095, K 0-16383 |
| | | risultato | ; R 0-4095 |
| | | resto | ; R 0-4095 |

Esempio DIV R 20 ; R 20 viene
 K 1000 ; diviso per 1000
 R 21 ; il risultato viene posto nel Registro 21
 R 1 ; e il resto nel Registro 1

Flag I flag **Zero (Z)** e **Segno (P o N)** vengono impostati in base al risultato.
 Il flag **Errore (E)** viene impostato in caso di divisione per zero.

Vedere anche FDIV

Uso pratico Leggere due numeri, dividerli e porre il risultato in un altro registro.
 I due numeri provengono dai preselettori BCD (2 digit) rispettivamente collegati agli ingressi 16-23 e 24-31

```

COB            0        ; Inizio COB
                 0

DIGI           2        ; Legge 2 digit
                 I 16    ; dall'ingresso 16 (fino a 23)
                 R 1    ; e li memorizza in R 1

DIGI           2        ; Legge 2 digit
                 I 24    ; dall'ingresso 24 (fino a 31)
                 R 2    ; e li memorizza in R 2

DIV           R 1    ; R 100 = R 1 / R 2
                 R 2
                 R 100 ; risultato
                 R 101 ; resto

CPB            E 99    ; Se viene impostato il Flag Errore,
                          ; chiama il Blocco di Programma 99

ECOB

PB             99

SET            O 47    ; Allarme in caso di divisione per zero (uscita 47)

EPB
    
```

SQR RADICE QUADRATA (SQuare Root)

Descrizione Radice quadrata di un intero.
 Esegue la radice quadrata del numero contenuto nel primo Registro e memorizza il risultato nel secondo Registro.
 Se il primo Registro contiene un valore negativo, viene attivato il flag di errore e l'operazione non viene eseguita

| | | | |
|-----------------|------------|-------------------------|---|
| Utilizzo | SQR | valore risultato | ; R 0-4095, K 0-16383 ; R 0-4095 |
|-----------------|------------|-------------------------|---|

Esempio SQR R 0 ; La radice quadrata del Registro 0
 R 100 ; viene posta nel Registro 100

Flags I flag **Zero (Z)** e **Segno (P o N)** vengono impostati in base al risultato.
 Il flag **Errore (E)** viene impostato se si tenta di eseguire la radice quadrata di un valore negativo

Vedere anche FSQR

Uso pratico Estrarre la radice quadrata di un numero letto sui preselettori BCD (4 digit), collegati agli ingressi 16-31.

COB 0 ; Inizio COB
 0

DIGI 4 ; Legge 4 digit
 I 16 ; dall'ingresso 16 (fino a 31)
 R 100 ; e li memorizza in R 100

SQR R 100 ; $R 101 = \sqrt{R 100}$
 R 101

ECOB

CMP CONFRONTO TRA REGISTRI (Compare Registers)

Descrizione Confronta il contenuto del primo Registro (o costante) con il contenuto del secondo Registro (o costante). L'operazione consiste nel sottrarre il secondo valore dal primo valore; I flag di stato vengono impostati in base al risultato. Il contenuto dei Registri rimane inalterato.

| | |
|-----------------|--|
| Utilizzo | CMP[X] valore 1 (i) ; R 0-4095, K 0-16383 valore 2 ; R 0-4095, K 0-16383 |
|-----------------|--|

Esempio CMP R 0 ; Confronta il Registro 0 con il Registro 1
 R 1 ; e imposta il flag di stato in base al risultato.

Flag I flag Zero (Z), Positivo (P) e Negativo (N) vengono impostati come indicato nella seguente tabella:

| | Z | P | N |
|---------------------|----------|----------|----------|
| Valore 1 = Valore 2 | Alto | Alto | Basso |
| Valore 1 > Valore 2 | Basso | Alto | Basso |
| Valore 1 < Valore 2 | Basso | Basso | Alto |

Vedere anche AND, OR, EXOR, FCMP

Uso pratico Leggere due numeri; se il primo numero è maggiore, uguale o inferiore al secondo numero devono essere attivate rispettivamente le uscite 32, 33 o 34. I due numeri provengono dai preselettori BCD (2 digit) rispettivamente collegati agli ingressi 16-23 e 24-31

```

COB            0    ; Inizio COB
                  0
DIGI            2    ; Legge 2 digit
                  I 16 ; dall'ingresso 16 (fino a 23)
                  R 1    ; e li memorizza in R 1
DIGI            2    ; Legge 2 digit
                  I 24 ; dall'ingresso 24 (fino a 31)
                  R 2    ; e li memorizza in R 2
CMP            R 1    ; Confronta R 1 con R 2
                  R 2
ACC            Z    ; Se R1 = R2
OUT            O 33 ; Allora imposta Uscita 33 e flag 0
OUT            F 0            Altrimenti azzera Uscita 33 e flag 0
ACC            N    ; Se R 1 < R2
OUT            O 34 ; Allora imposta uscita 34, altrimenti azzera uscita 34
ACC            P    ; Se R1 > R2
ANL            F 0    ;            (e non uguale)
OUT            O 32 ; Allora imposta uscita 32, altrimenti azzera uscita 32
ECOB
    
```

5. Istruzioni Aritmetiche in VIRGOLA MOBILE

I valori in virgola mobile possono essere memorizzati solo nei Registri.

Possono essere caricati nei Registri utilizzando l'istruzione LD.

Per specificare che si tratta di un numero in virgola mobile, il numero stesso deve includere un punto decimale '.' o un esponente 'E'. Ad esempio 1.21E3, -4.656E-2.

Il campo dei valori per i numeri in virgola mobile è:

+ 5.42101E-20 ... + 9.22337E+18 (precisione fino a 5 cifre significative)
 - 2.71056E-20 ... - 9.22337E+18

| | |
|--------------|--|
| IFP | I nteger to f loating p oint (Conversione da intero a virgola mobile) |
| FPI | F loating p oint to i nteger (Conversione da virgola mobile a intero) |
| FADD | F loating p oint a dd (Somma di numeri in virgola mobile) |
| FSUB | F loating p oint s ubtract (Sottrazione di numeri in virgola mobile) |
| FMUL | F loating p oint m ultiply (Moltiplicazione tra numeri in v.m.) |
| FDIV | F loating p oint d ivide (Divisione tra numeri in v. m.) |
| FSQR | S quare r oot (Radice Quadrata) |
| FCMP | F loating p oint c ompare (Confronto tra numeri in v.m.) |
| FSIN | S ine function (Funzione Seno) |
| FCOS | C osine function (Funzione Coseno) |
| FATAN | A rc t angent (Arcotangente) |
| FEXP | E xponential function (Funzione esponenziale) |
| FLN | L ogarithm function (Funzione logaritmo) |
| FABS | A bsolute value (Valore assoluto) |

NOTA:

I valori in virgola mobile vengono memorizzati nei registri in uno speciale formato binario; utilizzando tale valore come se fosse un intero si ottengono risultati errati. Se si mischiano valori interi e valori in virgola mobile in operazioni aritmetiche, si ottengono risultati non attendibili. In tal caso occorre prima convertire gli interi in virgola mobile utilizzando l'istruzione IFP. I valori in virgola mobile possono invece essere convertiti in valori interi utilizzando l'istruzione FPI..

I numeri in virgola mobile si basano sul seguente formato a 32 bit:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | m | s | e | e | e | e | e | e | | | | | | | | | | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |

dove m: mantissa, su 24 bit
 s: segno del numero
 e: esponente in notazione eccesso a 64, su 7 bit

Il bit di segno è uguale a 0 per indicare un valore positivo, 1 per indicare un valore negativo.

La mantissa è considerata come una frazione binaria in virgola fissa e, eccetto che per lo zero, è sempre normalizzata (ovvero ha un bit a 1 nella posizione più significativa).

L'esponente è la potenza di due che occorre applicare alla mantissa per ottenere il valore aritmetico vero e proprio del numero. E' gestito in notazione eccesso a 64, il che significa che i valori del complemento a due vengono arrotondati per eccesso a 64.

Formato valori in Virgola Mobile

IFP CONVERSIONE DA INTERO A VIRGOLA MOBILE (Integer to Floating Point)

Descrizione Converte il valore intero contenuto nel Registro specificato, nel corrispondente formato in virgola mobile. Il secondo operando indica la potenza di dieci a cui occorre elevare l'intero; questo controlla la posizione della virgola decimale.

Ad esempio, se la potenza di dieci è +3, il contenuto del Registro viene moltiplicato per 1000 (10^3), e il risultato viene memorizzato nel Registro in formato virgola mobile. Se il Registro conteneva 12, il risultato diventa 12000,00.

Se non è possibile effettuare la conversione (numero troppo grande o troppo piccolo), viene impostato il Flag di Errore e la conversione non viene eseguita.

Utilizzo

| | | | |
|---------------|----------------|------------|---|
| IFP[X] | reg | (i) | ; R 0-4095 |
| | potenza | | ; Potenza di dieci, da -20 a +18 |

Esempio

IFP R 0 ; R 0 = Valore in Virgola Mobile di $R 0 * 10^3$
 3

Flag

Il flag **Errore (E)** viene impostato se la conversione non è possibile.

Vedere anche

FPI

Uso pratico

| R 500 Prima | Istruzione | Conversione | R 500 Dopo |
|--------------------|--------------------|--------------------|-------------------|
| 123 | IFP R 500 0 | $R 500 * 10^0$ | 1.23E+2 |
| | IFP R 500 -2 | $R 500 * 10^{-2}$ | 1.23E+0 |
| | IFP R 500 3 | $R 500 * 10^3$ | 1.23E+5 |

FPI **CONVERSIONE DA VIRGOLA MOBILE A INTERO (Floating Point to Integer)**

Descrizione Converte il valore in virgola mobile contenuto nel Registro specificato, in un formato intero. Il secondo operando indica la potenza di dieci da utilizzare per la conversione. Il risultato è l'intero che si ottiene elevando il contenuto del Registro alla potenza definita nel secondo operando.

Ad esempio, se il Registro contiene 1234,56 e la potenza di dieci è -2, l'intero risultante è 12.

Se non è possibile effettuare la conversione, viene impostato il flag di Errore e non viene eseguita alcuna operazione.

Utilizzo

| | | | |
|---------------|----------------|------------|---|
| FPI[X] | reg | (i) | ; R 0-4095 |
| | potenza | | ; Potenza di dieci, da -20 a +18 |

Esempio

FPI R 0 ; Se R 0 contiene 1234.56, tale valore viene convertito
 0 ; nel valore intero 1234 (la potenza di dieci è zero)

Flag

Il flag **Errore (E)** viene impostato se la conversione non è possibile.

Vedere anche

IFP

Uso pratico

| R 500 Prima | Istruzione | Conversione | R 500 Dopo |
|--------------------|--------------------|--------------------|-------------------|
| 123.456 | FPI R 500 0 | $R\ 500 * 10^0$ | 123 |
| | FPI R 500 -2 | $R\ 500 * 10^{-2}$ | 1 |
| | FPI R 500 3 | $R\ 500 * 10^3$ | 123456 |

FADD SOMMA DI NUMERI IN VIRGOLA MOBILE (Floating Point Add)

Descrizione Somma il contenuto del primo Registro al contenuto del secondo Registro, e memorizza il risultato nel terzo Registro.
I Registri devono contenere valori in formato virgola mobile valido.

| | |
|-----------------|------------------------------------|
| Utilizzo | FADD reg1 ; R 0-4095 |
| | reg2 ; R 0-4095 |
| | risultato ; R 0-4095 |

Esempio FADD R 100 ; R 500 = R 100 + R 101
 R 101
 R 500

Flags I flag **Zero** (Z) e **Segno** (P o N) vengono impostati in base al risultato.
Il flag **Errore** (E) viene impostato in caso di superamento della capacità del Registro (overflow).

Vedere anche ADD

**FSUB SOTTRAZIONE TRA NUMERI IN VIRGOLA MOBILE
(Floating Point Subtract)**

Descrizione Sottrae il contenuto del secondo Registro al contenuto del primo Registro, e memorizza il risultato nel terzo Registro.
Entrambi i Registri devono contenere valori in formato virgola mobile valido.

| | |
|-----------------|---|
| Utilizzo | FSUB reg1 ; R 0-4095 reg2 ; R 0-4095 risultato ; R 0-4095 |
|-----------------|---|

Esempio FSUB R 0 ; R 0 = R 0 - R 1
 R 1
 R 0

Flags I flag **Zero** (Z) e **Segno** (P o N) vengono impostati in base al risultato.
 Il flag **Errore** (E) viene impostato in caso di superamento della capacità del Registro (overflow).

Vedere anche SUB

**FMUL MOLTIPLICAZIONE TRA NUMERI IN VIRGOLA
MOBILE (Floating Point Multiply)**

Descrizione Moltiplica il contenuto del primo Registro per il contenuto del secondo Registro, e memorizza il risultato nel terzo Registro.
Entrambi i Registri devono contenere valori in formato virgola mobile valido.

| | | | |
|-----------------|-------------|------------------|-------------------|
| Utilizzo | FMUL | reg1 | ; R 0-4095 |
| | | reg2 | ; R 0-4095 |
| | | risultato | ; R 0-4095 |

Esempio FMUL R 20 ; R 0 = R 20 * R 30
 R 30
 R 0

Flag I flag **Zero (Z)** e **Segno (P o N)** vengono impostati in base al risultato.
 Il flag **Errore (E)** viene impostato in caso di superamento della capacità del Registro (overflow), ovvero se il risultato è maggiore di 4.611686 E+18

Vedere anche MUL

FDIV DIVISIONE TRA NUMERI IN VIRGOLA MOBILE (Floating Point Divide)

Descrizione Divide il contenuto del primo Registro per il contenuto del secondo Registro, e memorizza il risultato nel terzo Registro.
In caso di divisione per zero, viene impostato il Flag di Errore, e l'operazione non è eseguita.
NOTA: Dal momento che le operazioni aritmetiche in virgola mobile forniscono risultati precisi, non esiste il resto.

| | | | |
|-----------------|-------------|------------------|-------------------|
| Utilizzo | FDIV | reg | ; R 0-4095 |
| | | divisore | ; R 0-4095 |
| | | risultato | ; R 0-4095 |

Esempio FDIV R 1 ; R 3 = R 1 / R 2
 R 2
 R 3

Flag I flag **Zero** (Z) e **Segno** (P o N) vengono impostati in base al risultato.
 Il flag **Errore** (E) viene impostato in caso di divisione per zero.

Vedere anche DIV

FSQR RADICE QUADRATA DI NUMERI IN VIRGOLA MOBILE (Floating Point Square Root)

Descrizione Memorizza la radice quadrata del contenuto del primo Registro nel secondo Registro.

Se il primo Registro contiene un valore negativo, viene impostato il Flag di Errore e viene assunto come risultato il valore della radice quadrata del valore assoluto (positivo).

Utilizzo

| | | |
|-------------|------------------|-------------------|
| FSQR | reg | ; R 0-4095 |
| | risultato | ; R 0-4095 |

Esempio FSQR R 0 ; R 0 = Radice quadrata di R 0
 R 0

Flag Il flag **Zero (Z)** viene impostato in base al risultato.
 Il risultato è sempre positivo, P è sempre Alto (H), N è sempre Basso (L).
 Se il valore è negativo, viene impostato il flag **Errore (E)**.

Vedere anche SQR

FCMP CONFRONTO TRA NUMERI IN VIRGOLA MOBILE (Floating Point Compare)

Descrizione Confronta il contenuto del primo Registro con il contenuto del secondo Registro e imposta i flag di Stato in base al risultato.
Nessuno dei due Registri viene alterato.
Entrambi i Registri devono contenere i valori in formato virgola mobile valido

Utilizzo

| | | | |
|---------|------|-----|------------|
| FCMP[X] | reg1 | (i) | ; R 0-4095 |
| | reg2 | | ; R 0-4095 |

Esempio

FCMP R 0 ; Confronta R 0 con R 1, e imposta i flag
 R 1 ; di Stato in base al risultato

Flag

I flag di Stato vengono impostati come segue:

| | Z | P | N |
|---------------------|----------|----------|----------|
| Valore 1 = Valore 2 | Alto | Alto | Basso |
| Valore 1 > Valore 2 | Basso | Alto | Basso |
| Valore 1 < Valore 2 | Basso | Basso | Alto |

Il flag **Errore** (E) viene impostato al livello basso (L)

Vedere anche

CMP

Nota

NON eseguire MAI il confronto tra valori in virgola mobile per verificarne l'uguaglianza, utilizzare > o < per evitare errori di precisione!

FSIN FUNZIONE SENO (Sine Function)

Descrizione Il seno del contenuto del primo Registro viene memorizzato nel secondo Registro. Il primo Registro deve contenere un valore in virgola mobile espresso in RADIANTI e compreso nel range $\pm 10^6$.

| | |
|-----------------|--|
| Utilizzo | FSIN[X] reg (i) ; R 0-4095 risultato (i) ; R 0-4095 |
|-----------------|--|

Esempio FSIN R 0 ; R 100 = Seno di R 0
R 100

Flag I flag **Zero (Z)** e **Stato (N o P)** vengono impostati in base al risultato.

Vedere anche FCOS, FATAN

FCOS FUNZIONE COSENO (Cosine Function)

Descrizione Il coseno del contenuto del primo Registro viene memorizzato nel secondo Registro.
 Il primo Registro deve contenere un valore in virgola mobile espresso in RADIANTI e compreso nel range $\pm 10^6$.

| | |
|-----------------|---|
| Utilizzo | FCOS[X] reg (i) ; R 0-4095 |
| | risultato (i) ; R 0-4095 |

Esempio FCOS R 100 ; R 20 = Coseno di R 100
 R 20

Flag I flag **Zero** (Z) e **Stato** (N o P) vengono impostati in base al risultato.

Vedere anche FSIN, FATAN

FATAN ARCOTANGENTE (Arc Tangent)

Descrizione L'arcotangente del contenuto del primo Registro viene memorizzato nel secondo Registro.
 Il primo Registro deve contenere un valore in virgola mobile valido, espresso in RADIANTI.
 Il risultato, contenuto nel secondo Registro può andare da $-\pi/2$ to $+\pi/2$.

Utilizzo

| | | | | |
|-----------------|------------------|------------|----------|-----------------|
| FATAN[X] | reg | (i) | ; | R 0-4095 |
| | risultato | (i) | ; | R 0-4095 |

Esempio FATAN R 1 ; R 0 = Arcotangente di R 1
 R 0

Flag I flag **Zero (Z)** e **Stato (N o P)** vengono impostati in base al risultato.

Vedere anche FSIN, FCOS

FEXP FUNZIONE ESPONENZIALE (Exponential Function)

Descrizione Il valore 'e' elevato alla potenza del contenuto del primo Registro, viene memorizzato nel secondo Registro.

Il Registro deve contenere un valore in formato virgola mobile valido.

Utilizzo

| |
|--|
| FEXP[X] reg (i) ; R 0-4095 |
| risultato (i) ; R 0-4095 |

Esempio FEXP R 0 ; R 1 = e^{R 0}
 R 1

Flag Il flag **Zero** (Z) viene impostato in base al risultato, il flag **Negativo** (N) è sempre impostato al livello Basso (P = 1).
Il flag **Errore** (E) viene impostato in caso di superamento della capacità del Registro (overflow).

Vedere anche FPI, IFP

FLN FUNZIONE LOGARITMO IN VIRGOLA MOBILE (Floating Point Logarithm Function)

Descrizione Il logaritmo naturale del contenuto del primo Registro, viene memorizzato nel secondo Registro.
 Il primo Registro deve contenere un valore in formato virgola mobile valido.
 Se si esegue il logaritmo naturale di un valore negativo, viene attivato il flag di Errore e viene assunto il logaritmo del valore assoluto (positivo).

Utilizzo

FLN[X] **reg** **(i)** **; R 0-4095**
 risultato **(i)** **; R 0-4095**

Esempio FLN R 1 ; R 2 = ln R 1
 R 2

Flag I flag **Zero** (Z) e **Segno** (P o N) vengono impostati in base al risultato.
 Il flag **Errore** (E) viene impostato in caso si esegua l'operazione "ln" su un valore zero o negativo

Vedere anche FEXP

**FABS VALORE ASSOLUTO IN VIRGOLA MOBILE
(Floating Point Absolute Value)**

Descrizione Il valore assoluto (convertito in positivo nel caso di numero negativo) del primo Registro, viene memorizzato nel secondo Registro.
Il primo Registro deve contenere un valore in formato virgola mobile valido.

| | |
|-----------------|--|
| Utilizzo | FABS[X] reg (i) ; R 0-4095 |
| | risultato (i) ; R 0-4095 |

Esempio FABS R 1 ; R 2 = valore assoluto di R 1
 R 2 ; Se R 1 contiene -7,5 allora R 2 = 7,5

Flag Il flag **Zero (Z)** viene impostato in base al risultato

6. Istruzioni BLOCTEC

Il Bloctec è un metodo di programmazione strutturata che suddivide un programma in blocchi di istruzioni separati.

Un **Blocco ad Organizzazione Ciclica (COB)** è un blocco di istruzioni principale che richiama tipicamente dei **Blocchi di Programma (PB)**, i quali a loro volta richiamano dei **Blocchi Funzione (FB)**.

Nel programma ci deve essere almeno un COB (COB 0).

I COB possono richiamare PB o FB (con parametri opzionali).

I PB e FB possono essi stessi richiamare qualsiasi altro PB o FB fino a 7 livelli di nidificazione.

Gli operandi di queste istruzioni non possono essere forniti come parametri di un Blocco Funzione.

Per ulteriori informazioni sui metodi di programmazione strutturata con l'uso di BLOCTEC, fare riferimento alla sezione "Programmazione Strutturata" del Manuale Utente

COB Cyclic Organisation Block (Blocco ad Organizzazione Ciclica)

ECOB End of Cyclic Org'n Block (Fine Blocco ad Organizzazione Ciclica)

XOB Exception Organisation Block (Blocco ad Organizzazione Esclusiva)

EXOB End of Exception Org'n Block (Fine XOB)

PB Program Block (Blocco di Programma)

EPB End of Program Block (Fine Blocco di Programma)

CPB Call Program Block (Chiamata Blocco di Programma)

CPBI Call Program Block Indirect (Chiamata Blocco di Programma Indiretta)

FB Function Block (Blocco Funzione)

EFB End of Function Block (Fine Blocco Funzione)

CFB Call Function Block (Chiamata Blocco Funzione)

NCOB Next Cyclic Org'n Block (Prossimo COB)

SCOB Stop Cyclic Org'n Block (Stop COB)

CCOB Continue Cyclic Org'n Block (Continua COB)

RCOB Restart Cyclic Org'n Block (Ripartenza COB)



Le seguenti istruzioni BLOCTEC, se non utilizzate correttamente, possono provocare gravi anomalie:

RCOB, NCOB, SCOB, CCOB e Tempo di Supervisione COB.

Se le suddette istruzioni vengono usate all'interno di un programma GRAFCET si possono verificare seri problemi. Se non utilizzate correttamente, queste istruzioni possono infatti rallentare notevolmente l'esecuzione del programma utente, nel migliore dei casi, se non addirittura causare la totale messa fuori sincronismo del GRAFTEC e quindi un BLOCCO dell'apparecchiatura (Crash).

Evitare perciò l'uso di queste istruzioni all'interno di una struttura GRAFTEC.

COB BLOCCO AD ORGANIZZAZIONE CICLICA (Cyclic Organisation Block)

Descrizione Inizia il Blocco ad Organizzazione Ciclica (COB) specificato.
 Il secondo operando definisce il tempo di supervisione del COB in incrementi di 10 millisecondi.
 Se il tempo di supervisione scade prima che sia terminata l'esecuzione del COB (cioè prima che sia stata raggiunta l'istruzione ECOB), viene eseguito, se presente, il Blocco "Esclusivo" XOB 11; nel caso in cui quest'ultimo non sia presente, viene avviato il COB successivo.
 Se il tempo di supervisione è 0, l'XOB 11 non viene mai eseguito, il COB successivo parte solo quando il COB corrente è terminato (cioè quando si incontra l'istruzione ECOB).
 Se sono stati programmati diversi COB, questi vengono eseguiti uno dopo l'altro in ordine numerico.

L'ACCU è sempre impostato Alto (1) all'inizio di ogni COB.

Nota: l'istruzione COB richiede 3 righe di programma..

| | |
|-----------------|--|
| Utilizzo | COB numero ; Numero COB 0-15 tempo ; Tempo di supervisione in incrementi ; di 10ms 0-100000 |
|-----------------|--|

Esempio COB 0 ; Inizio COB 0
 0 ; Tempo di Supervisione = 0
 ; Corpo del COB 0

 ECOB ; Fine COB 0

Flag All'inizio del COB, l'ACCU viene impostato al livello alto (H).

Vedere anche ECOB, NCOB, RCOB, SCOB, XOB, Manuale Utente

**ECOB FINE BLOCCO AD ORGANIZZAZIONE CICLICA
(End of Organisation Block)**

Descrizione Termina il COB corrente. Inizia l'esecuzione del COB successivo (se presente).
Il corpo di un COB deve sempre terminare con una istruzione ECOB.

Utilizzo

| |
|---|
| ECOB ; Non sono richiesti operandi |
|---|

Esempio

```
COB      0      ; Inizio COB 0
          0
          ...   ;   corpo del COB
          ...
          ...
ECOB     ; Fine COB
```

Flag

Invariati.

Vedere anche

COB, Manuale Utente

XOB BLOCCO AD ORGANIZZAZIONE ESCLUSIVA (Exception Organisation Block)

Descrizione Contrassegna l'inizio di un Blocco ad Organizzazione Esclusiva (XOB).

Utilizzo

| | | |
|------------|---------------|---------------------------|
| XOB | numero | ; Numero XOB, 0-30 |
|------------|---------------|---------------------------|

Esempio

XOB 16 ; XOB di Partenza a freddo
 ... ; corpo dell'XOB
 EXOB ; Fine XOB

Flag

All'inizio dell'XOB, l'ACCU viene impostato al livello alto (H).

Vedere anche

EXOB, Manuale Utente

Un Blocco ad Organizzazione Esclusiva (XOB) viene richiamato quando si verifica un errore o un altro evento significativo. L'XOB può contenere le istruzioni di programma che gestiscono tali eventi. Se non è presente alcun XOB associato all'evento, non viene intrapresa alcuna azione (l'evento viene ignorato) e si accende la lampada di segnalazione ERROR.

Al termine dell'XOB, il programma ritorna alla locazione dalla quale era stata richiamata la routine esclusiva.

Ciascun XOB svolge una specifica funzione:

| XOB | Descrizione | Priorità |
|------------|---|-----------------|
| 0 | Caduta di alimentazione | 4 |
| 8 | Codice istruzione non valido | 4 |
| 7 | Sovraccarico sistema | 3 |
| 11 | Superamento tempo di supervisione COB | 3 |
| 14 | XOB ciclico | 3 |
| 15 | XOB ciclico | 3 |
| 17 | XOB per richiesta interrupt S-Bus | 3 |
| 18 | XOB per richiesta interrupt S-Bus | 3 |
| 19 | XOB per richiesta interrupt S-Bus | 3 |
| 20 | Ingresso di interrupt INB1 | 3 |
| 25 | Ingresso di interrupt INB2 | 3 |
| 1 | Caduta di tensione sul rack di estensione | 2 |
| 2 | Batteria scarica | 2 |
| 4 | Errore di parità sul bus princip. (solo PCD6) | 1 |
| 5 | Nessuna risposta dal modulo di I/O | 1 |
| 9 | Troppi task attivi (GRAFTEC) | 1 |
| 10 | Superamento limiti di nidificazione PB/FB | 1 |
| 12 | Superamento capacità registro indice | 1 |
| 13 | Impostazione flag di errore | 1 |
| 16 | Eseguito all'accensione del PCD | 1 |
| 30 | Connessione RIO master ↔ slave | 1 |

XOB con livello di priorità 4:

Il livello 4 rappresenta la più alta priorità, solo gli XOB 0 (Caduta di tensione) e 8 (codice istruzione non valido) possono interrompere l'esecuzione di un altro XOB.

XOB 0 Caduta di tensione

Tra la chiamata dell'XOB 0 e la definitiva interruzione di alimentazione alla CPU vi può essere un ritardo massimo di 10 ms, per lasciare all'utente il tempo di eseguire il salvataggio di alcuni dati importanti. Se l'XOB 0 è stato programmato, nella tabella storica (History table) viene scritto il messaggio "XOB 0 START EXEC" all'avvio dell'XOB ed il messaggio "XOB 0 EXECUTED" al termine dello stesso. Questo per indicare all'utente che l'XOB è stato eseguito prima della caduta di tensione. Se l'XOB non viene programmato, alla rilevazione della caduta di tensione viene eseguita immediatamente una ripartenza a freddo mentre se l'XOB è programmato, la ripartenza a freddo viene eseguita al termine dell'elaborazione dell'XOB stesso, se vi è ancora energia sufficiente.

XOB 8 Codice istruzione non valido

L'XOB 8 viene chiamato quando il firmware rileva un'istruzione non valida all'interno del programma utente.

XOB con livello di priorità 3:

Se si verifica un interrupt di livello 2 o 3 durante l'esecuzione di un XOB con priorità inferiore, tale evento viene trattato successivamente all'XOB in corso di elaborazione. E' stata aumentata la priorità degli XOB 20/25/11 in modo da assicurarne l'elaborazione direttamente dopo il completamento dell'XOB corrente, nel caso in cui uno dei suddetti XOB venga generato durante l'esecuzione di un XOB con priorità inferiore od uguale.

XOB 7 Sovraccarico sistema

Il meccanismo di gestione della coda di XOB con livello di priorità 3 in attesa di esecuzione è sovraccarico.

XOB 11 Superamento tempo di supervisione COB

Se nella seconda riga di un'istruzione COB viene indicato un periodo di controllo (espresso in 1/100 secondo) e se il tempo di elaborazione del COB in oggetto supera tale periodo, viene richiamato l'XOB 11. Il tempo di elaborazione di un COB è l'intervallo di tempo che intercorre tra le istruzioni COB ed ECOB.

XOB 14 XOB ciclico

XOB 15

Gli XOB 14 e 15 vengono richiamati periodicamente, con una frequenza che può variare tra 10 ms e 1,000s. Tale frequenza può essere definita utilizzando l'istruzione SYSWR.

XOB 17 XOB per richiesta interrupt S-Bus

XOB 18

XOB 19

Questi tre XOB possono essere utilizzati come routine di interrupt. La loro esecuzione può essere richiesta attraverso la rete S-Bus. E' anche possibile richiederne l'esecuzione utilizzando l'istruzione SYSWR.

XOB 20 Ingressi di interrupt INB1, INB2**XOB 25**

L'XOB 20 (o 25) viene richiamato quando sull'ingresso di interrupt INB1 (o INB2) di un PCD1/2 è stato rilevato un fronte di salita del segnale (vedere il Manuale Hardware PCD1/2 per maggiori dettagli).

XOB con livello di priorità 2:**XOB 1 Caduta di tensione sul rack di estensione**

Il dispositivo di monitoraggio della tensione dell'alimentatore di un rack di estensione (PCD 2 o PCD6) ha rilevato un'eccessiva caduta della tensione di alimentazione. In questi casi, tutte le Uscite del rack di estensione interessato vengono portate al livello basso entro 2ms e viene richiamato l'XOB 1. Se le varie Uscite del rack di estensione "in avaria" continuano ad essere gestite (impostate, azzerate o interrogate) dal programma utente di una qualsiasi CPU, viene richiamato anche l'XOB 4 e/o l'XOB 5 (solo PCD4). L'XOB 1 verrà richiamato una sola volta entro 250 ms dalla rilevazione della condizione di anomalia.

XOB 2 Batteria scarica o in avaria

La batteria è scarica, in avaria oppure è assente.

I dati contenuti in Flag non volatili e Registri, il programma utente presente nella memoria RAM tamponata e le impostazioni dell'orologio in tempo reale potrebbero andare persi. In caso di malfunzionamento della batteria, l'XOB 2 viene richiamato esclusivamente dalla CPU 0 ogni 250 ms.

XOB con livello di priorità 1:

Qualsiasi XOB di livello 1 richiamato durante l'esecuzione di un altro XOB non verrà mai gestito.

XOB 4 Errore di parità

L'XOB 4 può essere richiamato esclusivamente utilizzando unità PCD equipaggiate con rack di estensione (solo PCD6). Il circuito di controllo del bus degli indirizzi ha rilevato un errore di parità. Questo tipo di errore può essere causato da un guasto sul cavo di estensione, da un rack di estensione difettoso o da un modulo di estensione del bus oppure, molto semplicemente, derivare dal fatto che il rack di estensione indirizzato non è presente.

XOB 5 Nessuna risposta dal modulo di I/O (Errore di I/O assente)

I moduli di Ingresso e Uscita del PCD rispondono con un segnale agli indirizzamenti da parte della CPU. Se questo segnale non viene ricevuto, viene richiamato l'XOB 5. In genere, questo accade quando il modulo indirizzato non è presente ma può anche essere determinato da una codifica errata degli indirizzi da parte del modulo. In un modulo PCD4 equipaggiato con solo 8 elementi, l'XOB 5 non viene richiamato se viene indirizzato uno degli elementi assenti, dal momento che l'indirizzo viene comunque codificato e viene inviato il segnale di risposta.

Sulle unità PCD1 e 2 questo meccanismo non è implementato.

XOB 9 Troppi task Graftec attivi

Sono stati attivati simultaneamente più di 32 rami GRAFTEC all'interno di un Blocco Sequenziale(SB).

XOB 10 Più di 7 livelli di chiamata PB/FB nidificati

I PB e i FB possono essere nidificati fino ad un massimo di 7 livelli. Una ulteriore chiamata (chiamata all'ottavo livello) determina l'esecuzione dell'XOB 10.

La chiamata all'ottavo livello non viene eseguita.

XOB 12 Superamento della capacità del registro indice

Se un programma contiene un elemento indirizzato oltre i limiti del relativo campo di indirizzi (da 0 a 8191), viene richiamato l'XOB 12.

XOB 13 Impostazione flag di errore

L'XOB 13 viene richiamato ogni volta che il Flag di Errore risulta attivo (impostato), indipendentemente dalla causa che ne ha determinato l'impostazione (calcolo, trasferimento di dati o errore di comunicazione).

XOB 16 Partenza a freddo

L'XOB 16 rappresenta l'XOB di avvio del sistema (XOB di Partenza a Freddo) e viene eseguito all'accensione del PCD oppure quando si determina un riavvio a freddo del sistema. L'XOB 16 può inizializzare qualsiasi elemento prima di avviare l'esecuzione del programma.

XOB 30 Connessione RIO master ↔ slave (in preparazione)

Al termine di ciascun messaggio inviato dal master ad una qualsiasi stazione slave, viene eseguito un test della connessione. Se la stazione slave interessata non risponde positivamente a tale test, la CPU master richiama l'XOB 30. Questo avviene principalmente quando, durante il funzionamento on-line, una stazione viene rimossa dalla rete o disabilitata.

**EXOB FINE BLOCCO AD ORGANIZZAZIONE ESCLUSIVA
(End of Exception Organisation Block)**

Descrizione Fine dell'XOB corrente.
Quando incontra l'istruzione EXOB, l'XOB restituisce il controllo alla locazione di memoria da cui era stato chiamato.

Utilizzo

| |
|---|
| EXOB ; Non sono richiesti operandi |
|---|

Esempio

```
XOB    16    ; Inizio XOB 16
      ...    ;    Corpo dell'XOB 16
      ...
      ...
EXOB           ; Fine XOB 16
```

Flag Invariati.

Vedere anche XOB, Manuale Utente

PB BLOCCO DI PROGRAMMA (Program Block)

Descrizione Contrassegna l'inizio di un Blocco di Programma (PB), cioè di una subroutine senza parametri.

Utilizzo

| |
|---|
| PB numero ; Numero PB, 0-299 |
|---|

Esempio PB 26 ; Inizio PB 26
 ... ; Corpo del PB 26
 ...
 ...
 EPB ; Fine PB 26

Flag All'inizio del PB, l'**ACCU** viene impostato al livello alto (H)

Vedere anche EPB, CPB, FB, Manuale Utente

**EPB FINE DEL BLOCCO DI PROGRAMMA
 (End of Program Block)**

Descrizione Termina il Blocco di Programma corrente (PB).
 Il controllo viene ceduto all'istruzione successiva a quella che aveva richiamato il
 Blocco di Programma (CPB).

Utilizzo

| |
|--|
| EPB ; Non sono richiesti operandi |
|--|

Esempio PB 0 ; Inizio PB 0
 ... ; Corpo del PB 0
 ...
 ...
 EPB ; Fine PB 0

Flag Viene ripristinato lo stato che l'**ACCU** aveva prima della chiamata del PB

Vedere anche PB, CPB, Manuale Utente

CPB CHIAMATA BLOCCO DI PROGRAMMA (Call Program Block)

Descrizione Chiama in modo condizionato o incondizionato un Blocco di Programma. Se la condizione non è soddisfatta, il PB non viene chiamato.

| Condizione | Il Blocco di Programma viene chiamato: |
|------------|--|
| spazio | Sempre (nessun codice condizione) |
| H | Se Accumulatore = H (1) |
| L | Se Accumulatore = L (0) |
| P | Se Flag Positivo = H (Flag Negativo = L) |
| N | Se Flag Negativo = H |
| Z | Se Flag Zero = H |
| E | Se Flag Errore = H |

Utilizzo

| | | |
|------------|-------------|---------------------------------------|
| CPB | [cc] numero | ; Numero PB 0-299 |
| | | ; cc = codice condizione: H L P N Z E |

Esempio CPB 10 ; chiamata incondizionata del PB 10

Flag All'inizio del PB, l'ACCU viene impostato al livello alto (H). Nel programma in cui era stata effettuata la chiamata al PB, l'ACCU viene ripristinato con lo stato che aveva prima della chiamata al PB

Vedere anche PB, EPB, CFB, Manuale Utente

Uso pratico Struttura IF .. THEN .. ELSE

```

COB      0
          0
...
STH      I 15 ; SE (IF) l'ingresso 15 è Alto (H)
CPB      H 20 ; ALLORA (THEN) chiama il PB 20
CPB      L 25 ; ALTRIMENTI (ELSE) chiama il PB 25
...
ECOB
PB      20
.....
EPB
PB      25
.....
EPB
    
```

CPBI CHIAMATA INDIRETTA BLOCCO DI PROGRAMMA (Call Program Block Indirect)

Descrizione Chiama in modo condizionato o incondizionato un Blocco di Programma il cui numero è contenuto nel Registro specificato.
 Poiché l'istruzione utilizza un codice condizione, non è necessario specificare il tipo di dato 'R'.
 Se il Registro specificato contiene un numero di PB errato (>299) o se il PB è inesistente, viene attivato il flag Errore e viene chiamato l'XOB 13 (se presente).
 Se la condizione non è soddisfatta, il PB non viene chiamato.

| Condizione | Il Blocco di Programma viene chiamato: |
|-------------------|---|
| spazio | Sempre (nessun codice condizione) |
| H | Se Accumulatore = H (1) |
| L | Se Accumulatore = L (0) |
| P | Se Flag Positivo = H (Flag Negativo = L) |
| N | Se Flag Negativo = H |
| Z | Se Flag Zero = H |
| E | Se Flag Errore = H |

| | |
|-----------------|--|
| Utilizzo | CPBI [cc] reg ; reg = Numero del registro contenente il ; numero del PB da chiamare ; cc = codice condizione: H L P N Z E |
|-----------------|--|

Esempio CPBI L 10 ; Se l'ACCU è Basso (0), allora viene chiamatao il PB
; il cui numero è contenuto in R 10

Flag Il flag Errore viene attivato qualora il Registro specificato contenga un numero di PB errato o il PB sia inesistente.
 All'inizio del PB, l'ACCU viene impostato al livello alto (H).
 Nel programma in cui era stata effettuata la chiamata al PB, l'ACCU viene ripristinato con lo stato che aveva prima della chiamata al PB

Vedere anche PB, EPB, CFB, Manuale Utente

FB BLOCCO FUNZIONE (Function Block)

Descrizione Inizia un Blocco Funzione (FB). Un FB è una sobroutine con parametri opzionali.
 Si può definire un elenco di parametri opzionali, tali parametri vengono forniti all'atto della chiamata al FB.

Utilizzo

| |
|--|
| FB numero ; Numero FB, 0-999 |
|--|

Esempio

```

FB      0      ; Inizio FB 0
...
STH    = 1      ; Riferimento parametri FB
...
EFB                                ; Fine FB 0
    
```

Flag All'inizio dell'FB, l'**ACCU** viene impostato al livello alto (H)

Vedere anche EFB, CFB, Manuale Utente

Uso pratico Calcolo della formula: $Z = X * (X+Y)$

```

FB      25      ; Blocco Funzione X * (X+Y)
ADD      = 1      ; Z = X + Y
          = 2
          = 3
MUL      = 3      ; Z = Z * X
          = 1
          = 3
EFB
COB      7
          0
...
STH      I 1      ; Se l'ingresso 1 diventa Alto (H)
DYN      F 1
CFB      H 25     ; Allora R107 = R100 * (R100+330)
          R 100   ; Parametro 1 (X)
          K 330   ; Parametro 2 (Y)
          R 107   ; Parametro 3 (Z)
...
STH      I 2      ; Se l'ingresso 2 diventa Alto (H)
DYN      F 2
CFB      H 25     ; Allora R107 = R200 * (R200+R201)
          R 200   ; Parametro 1 (X)
          R 201   ; Parametro 2 (Y)
          R 107   ; Parametro 3 (Z)
...
ECOB
    
```

**EFB FINE BLOCCO FUNZIONE
(End of Function Block)**

Descrizione Termina il Blocco Funzione (FB). Restituisce il controllo all'istruzione successiva a quella che aveva chiamato il Blocco Funzione (CFB).

Utilizzo

| |
|--|
| EFB ; Non sono richiesti operandi |
|--|

Esempio
 FB 0 ; Inizio FB 0
 ... ; Corpo di FB 0
 ...
 ...
 EFB ; Fine FB 0

Flag L'ACCU viene ripristinato con lo stato che aveva prima della chiamata dell'FB.

Vedere anche FB, CFB, Manuale Utente

CFB CHIAMATA BLOCCO FUNZIONE (Call Function Block)

Descrizione Richiama in modo condizionato o incondizionato un Blocco Funzione. Se la condizione non è soddisfatta, l'FB non viene chiamato. L'istruzione CFB può essere seguita da una lista di parametri opzionali. Tali parametri vengono utilizzati dalle istruzioni incluse nel Blocco Funzioni. Il riferimento ai parametri viene fatto utilizzando "=n" come operando, dove "n" è il numero del parametro da utilizzare (1-128). Il valore di questo parametro è sostituito dall'operando.

| Tipo | Descrizione | Campo di Valori |
|------|--------------------------|--|
| I | Input (Ingresso) | 0..8191 |
| O | Output (Uscita) | 0..8191 |
| F | Flag | 0..8191 |
| C | Contatore | 0..1599 |
| T | Timer | 0..450 |
| R | Registro | 0..4095 |
| K | Costante K | 0..16383 |
| X | TeXt (Testo) | 0..7999 |
| DB | Data Block (Blocco dati) | 0..7999 |
| S | Semaforo | 0..99 |
| W | Word (Parola) | 0..65535 Usato per qualsiasi costante in LDL, LDH |
| M | Tipo dati per MOV | Q 0..31 D 0..9 N 0..7 B 0..3 W 0..1 L 0 |

Utilizzo

| | | |
|------------|--------------------|--|
| CFB | [cc] numero | ; Numero FB, 0-999 |
| | [param 1] | ; cc = codice condizione: H L P N Z E |
| | [param 2] | ; elenco parametri opzionali |
| | [param n] | |

Esempio CFB H 10 ; Chiama il FB 10 se l'ACCU è alto (H)
I 32 ; Parametro 1
R 10 ; Parametro 2

Flag All'inizio dell'FB, l'ACCU viene impostato al livello alto (H).

Vedere anche FB, CPB, Manuale Utente

NCOB PROSSIMO COB (Next Cyclic Organisation Block)

Descrizione Forza il programma ad eseguire in modo condizionato o incondizionato il COB successivo.
 Se il codice condizione non viene soddisfatto, l'istruzione NCOB viene ignorata. Con l'utilizzo di istruzioni NCOB si possono programmare cicli di attesa senza interferire con l'esecuzione di alcun altro COB.
 Per ciascun ciclo di attesa, occorre inserire una istruzione NCOB. Questo consente l'esecuzione "parallela" dei COB.

Programmi in BLOCTEC o GRAFTEC ben strutturati NON dovrebbero contenere cicli di attesa, e quindi non è necessario ricorrere all'istruzione NCOB. In genere, per controllare l'esecuzione del programma utilizzare lo stato dell'ACCU.
Processi sequenziali possono essere facilmente realizzati in GRAFTEC


| Condizione | |
|------------|--|
| spazio | Sempre (nessun codice condizione) |
| H | Se Accumulatore = H (1) |
| L | Se Accumulatore = L (0) |
| P | Se Flag Positivo = H (Flag Negativo = L) |
| N | Se Flag Negativo = H |
| Z | Se Flag Zero = H |
| E | Se Flag Errore = H |

Utilizzo NCOB [cc] ; cc = codice condizione: H|L|P|N|Z|E

Esempio
 STH I 15 ; Attendi finché I 15 = L
 NCOB L
 JR L -2

Flag Invariati.

Vedere anche RCOB, SCOB, CCOB, Manuale Utente



Le seguenti istruzioni BLOCTEC, se non utilizzate correttamente, possono provocare gravi anomalie:
 RCOB, NCOB, SCOB, CCOB e Tempo di Supervisione COB.

Se le suddette istruzioni vengono usate all'interno di un programma GRAFCET si possono verificare seri problemi. Se non utilizzate correttamente, queste istruzioni possono infatti rallentare notevolmente l'esecuzione del programma utente, nel migliore dei casi, se non addirittura causare la totale messa fuori sincronismo del GRAFTEC e quindi un BLOCCO dell'apparecchiatura (Crash). Evitare perciò l'uso di queste istruzioni all'interno di una struttura GRAFTEC.

SCOB STOP COB (Stop Cyclic Organisation Block)

Descrizione Arresta l'esecuzione di un determinato COB in modo condizionato o incondizionato.
 L'esecuzione prosegue con il COB successivo.
 Il COB non viene più rieseguito finchè non viene eseguita la corretta istruzione CCOB da parte di un altro COB.
 Un COB può autoarrestare la propria esecuzione, ma deve essere fatto ripartire da un altro COB che contenga una istruzione CCOB.
 Se la condizione non è soddisfatta, l'istruzione SCOB è ignorata.

**Un programma ben strutturato non necessita di questa istruzione.
 Essa deve essere usata nell'applicazione utente con la massima attenzione.**


| Condizione | |
|------------|--|
| spazio | Sempre (nessun codice condizione) |
| H | Se Accumulatore = H (1) |
| L | Se Accumulatore = L (0) |
| P | Se Flag Positivo = H (Flag Negativo = L) |
| N | Se Flag Negativo = H |
| Z | Se Flag Zero = H |
| E | Se Flag Errore = H |

Utilizzo SCOB [cc] cob ; Numero COB 0-15
 ; cc = codice condizione: H|L|P|N|Z|E

Esempio SCOB L 10 ; Arresta il COB 10 se l'ACCU è basso (0)

Flag Invariati.

Vedere anche CCOB, NCOB, RCOB, Manuale Utente



Le seguenti istruzioni BLOC TEC, se non utilizzate correttamente, possono provocare gravi anomalie:

RCOB, NCOB, SCOB, CCOB e Tempo di Supervisione COB.

Se le suddette istruzioni vengono usate all'interno di un programma GRAFCET si possono verificare seri problemi. Se non utilizzate correttamente, queste istruzioni possono infatti rallentare notevolmente l'esecuzione del programma utente, nel migliore dei casi, se non addirittura causare la totale messa fuori sincronismo del GRAFTEC e quindi un BLOCCO dell'apparecchiatura (Crash). Evitare perciò l'uso di queste istruzioni all'interno di una struttura GRAFTEC.

CCOB CONTINUA COB (Continue Cyclic Organisation Block)

Descrizione Consente a un COB, precedentemente sospeso con una istruzione SCOB, di riprendere l'esecuzione in modo condizionato o incondizionato. Se la condizione non è soddisfatta, il COB non viene ripristinato. L'istruzione CCOB non determina l'esecuzione immediata del COB, ma ne abilita l'esecuzione alla prossima chiamata.

Un programma ben strutturato non necessita di questa istruzione. Essa deve essere usata nell'applicazione utente con la massima attenzione.

| Condizione | |
|------------|--|
| spazio | Sempre (nessun codice condizione) |
| H | Se Accumulatore = H (1) |
| L | Se Accumulatore = L (0) |
| P | Se Flag Positivo = H (Flag Negativo = L) |
| N | Se Flag Negativo = H |
| Z | Se Flag Zero = H |
| E | Se Flag Errore = H |


Utilizzo CCOB [cc] numero ; Numero COB 0-15
; cc = codice condizione: H|L|P|N|Z|

Esempio CCOB L 10 ; Il COB 10 viene ripristinato se l'ACCU è basso (0)

CCOB 0 ; Il COB 0 viene ripristinato in modo incondizionato

Flag Invariati.

Verdere anche NCOB, RCOB, SCOB, Manuale Utente



Le seguenti istruzioni BLOCTEC, se non utilizzate correttamente, possono provocare gravi anomalie:

RCOB, NCOB, SCOB, CCOB e Tempo di Supervisione COB.

Se le suddette istruzioni vengono usate all'interno di un programma GRAFCET si possono verificare seri problemi. Se non utilizzate correttamente, queste istruzioni possono infatti rallentare notevolmente l'esecuzione del programma utente, nel migliore dei casi, se non addirittura causare la totale messa fuori sincronismo del GRAFTEC e quindi un BLOCCO dell'apparecchiatura (Crash). Evitare perciò l'uso di queste istruzioni all'interno di una struttura GRAFTEC.

RCOB RIPARTENZA COB (Restart Cyclic Organisation Block)

Descrizione Fa ripartire qualsiasi COB, in modo condizionato o incondizionato, a partire da una determinata riga di programma. Questa istruzione può essere utilizzata all'interno di qualsiasi COB o XOB. Se la condizione non viene soddisfatta, l'istruzione RCOB viene ignorata. Il primo operando rappresenta il numero del COB da far ripartire. Il secondo operando rappresenta la riga di programma da cui far partire l'esecuzione. La riga di programma è espressa sotto forma di offset rispetto all'inizio del COB; non è quindi il numero di riga assoluto del programma.

**Un programma ben strutturato non necessita di questa istruzione.
Essa deve essere usata nell'applicazione utente con la massima attenzione.**

| Condizione | |
|------------|--|
| spazio | Sempre (nessun codice condizione) |
| H | Se Accumulatore = H (1) |
| L | Se Accumulatore = L (0) |
| P | Se Flag Positivo = H (Flag Negativo = L) |
| N | Se Flag Negativo = H |
| Z | Se Flag Zero = H |
| E | Se Flag Errore = H |

Utilizzo **RCOB** [cc] cob ; Numero COB 0-15
 riga ; N° di righe dall'inizio del COB (0-65535)
 ; cc = codice condizione: H|L|P|N|Z|E

Esempio RCOB 0 ; Ripartenza COB 0
 10 ; L'esecuzione inizia dalla riga 10 del COB 0

Flag Invariati.

Vedere anche NCOB, SCOB, CCOB, Manuale Utente



Le seguenti istruzioni BLOC TEC, se non utilizzate correttamente, possono provocare gravi anomalie:

RCOB, NCOB, SCOB, CCOB e Tempo di Supervisione COB.

Se le suddette istruzioni vengono usate all'interno di un programma GRAFCET si possono verificare seri problemi. Se non utilizzate correttamente, queste istruzioni possono infatti rallentare notevolmente l'esecuzione del programma utente, nel migliore dei casi, se non addirittura causare la totale messa fuori sincronismo del GRAFTEC e quindi un BLOCCO dell'apparecchiatura (Crash).

Evitare perciò l'uso di queste istruzioni all'interno di una struttura GRAFTEC.

7. Istruzioni GRAFTEC

Il GRAFTEC SAIA® è un metodo di programmazione autodocumentato per processi passo-passo.

Un programma **GRAFTEC** è costituito da una sequenza alternata di S**T**ep (ST) e TRansizioni (TR). Tale sequenza di ST e TR costituisce il corpo di un Blocco Sequenziale, che è chiamato da un Blocco ad Organizzazione Ciclica (COB).

Gli **STEP** contengono le azioni da eseguire e sono rappresentati da istruzioni quali SET, RES, STXT, ecc.

Le **TRANSIZIONI** contengono le combinazioni logiche condizionate e sono rappresentate da istruzioni quali STH, ANL CMP, ecc. Una TRansizione deve sempre essere seguita da un S**T**ep. Lo S**T**ep viene eseguito solo se la TRansizione precedente è soddisfatta (ACCU=H).

Per facilitare la programmazione e il test dei programmi GRAFTEC, si raccomanda di utilizzare l'**EDITOR GRAFTEC SAIA (SGRAF)**. Questo editor gestisce in modo automatico la struttura del programma creato graficamente sullo schermo. Con tale editor non è necessario utilizzare le istruzioni a basso livello del GRAFTEC SAIA® elencate nel seguito.

Per ulteriori informazioni sulla programmazione GRAFTEC fare riferimento al capitolo "Programmazione Strutturata" del Manuale Utente.

Gli operandi di queste istruzioni non possono essere forniti come parametri di un Blocco Funzione.

| | |
|------------|--|
| SB | Sequential B lock (Blocco Sequenziale) |
| ESB | End of Sequential B lock (Fine del Blocco Sequenziale) |
| CSB | Call Sequential B lock (Chiamata Blocco Sequenziale) |
| RSB | Restart Sequential B lock (Ripartenza Blocco Sequenziale) |
| IST | Initial S tep (Step Iniziale) |
| ST | S tep (Step) |
| EST | End of S tep (Fine Step) |
| TR | T ransition (Transizione) |
| ETR | End of T ransition (Fine Transizione) |

SB BLOCCO SEQUENZIALE (Sequential Block)

Descrizione Inizia un Blocco Sequenziale (SB).
 Un Blocco Sequenziale contiene un programma GRAFTEC indipendente.
 Un SB contiene solo istruzioni GRAFTEC come IST, ST, TR, EST, ETR e ESB

| | |
|-----------------|---|
| Utilizzo | SB numero ; Numero SB 0-31 |
|-----------------|---|

Esempio SB 10 ; Inizio SB 10
 ... ; Corpo di SB 10, contenente ST e TR.
 ESB ; Fine SB 10

Flag Invariati.

Vedere anche ESB, CSB, RSB, IST, ST, TR, Manuale Utente

CSB CHIAMATA BLOCCO SEQUENZIALE (Call Sequential Block)

Descrizione Chiama (in modo condizionato o incondizionato) un Blocco Sequenziale.
Se la condizione non viene soddisfatta, l'SB non viene chiamato.

Un blocco sequenziale non può essere chiamato da un altro blocco sequenziale.

| Condizione | Il Blocco Sequenziale viene chiamato: |
|-------------------|--|
| spazio | Sempre (nessun codice condizione) |
| H | Se l'Accumulatore = H |
| L | Se l'Accumulatore = L |
| P | Se il flag Positivo = H (flag Negativo = L) |
| N | Se il flag Negativo = H |
| Z | Se il flag Zero = H |
| E | Se il flag Errore = H |

Utilizzp

| | | |
|------------|--------------------|---|
| CSB | [cc] numero | ; Numero SB 0-31 ; cc = codice condizione: H L P N Z E |
|------------|--------------------|---|

Esempio CSB L 10 ; Chiama l'SB 10 se l'ACCU è basso (L)

Flag Alla partenza di un SB, l'ACCU è Alto (H).
Nel programma in cui era stata effettuata la chiamata all'SB, l'ACCU viene ripristinato con lo stato che aveva prima della chiamata all'SB stesso

Vedere anche SB, CPB, Manuale Utente

RSB RIPARTENZA BLOCCO SEQUENZIALE (Restart Sequential Block)

Descrizione Fa ripartire in modo condizionato o incondizionato un Blocco Sequenziale (SB).
Il primo operando rappresenta il numero dell'SB da far ripartire.
Il secondo operando rappresenta il numero di STEP da cui l'SB deve ripartire.

Se più rami devono essere fatti ripartire contemporaneamente (programmi paralleli), l'istruzione RSB dovrà contenere tante righe aggiuntive quanti sono gli step da riavviare

| Condizione | : |
|------------|---|
| spazio | Sempre (nessun codice condizione) |
| H | Se l'Accumulatore = H |
| L | Se l'Accumulatore = L |
| P | Se il flag Positivo = H (flag Negativo = L) |
| N | Se il flag Negativo = H |
| Z | Se il flag Zero = H |
| E | Se il flag Errore = H |

Utilizzo

| | | |
|------------|-------------|---------------------------------------|
| RSB | [cc] numero | ; Numero SB 0-31 |
| | | ; cc = codice condizione: H L P N Z E |
| | step | ; Numero STep, 0-1999 |
| | [step] | ; [Numero STep, 0-1999] |
| | [...] | ; [...] |
| | [step] | ; [Numero STep, 0-1999] |

Esempio RSB 12 ; L'SB 12 riparte dallo Step 1.
1

Flag Prima della ripartenza dell'SB, l'ACCU viene impostato al valore Alto (H).

Vedere anche SB, CSB, ST, Manuale Utente

Uso pratico



L'istruzione RSB, se non utilizzata correttamente, può provocare seri problemi. Non vi è infatti alcuna verifica dei parametri specificati per l'istruzione in oggetto che, se erroneamente definiti, possono, nel migliore dei casi, causare la totale messa fuori sincronismo del programma se non addirittura un BLOCCO dell'apparecchiatura (Crash).

IST STEP INIZIALE (Initial Step)

Descrizione Lo Step Iniziale definisce il primo Step da eseguire quando viene chiamato un Blocco Sequenziale (SB).
 Ogni SB deve avere almeno uno Step Iniziale.
 A tutti gli effetti, lo Step iniziale è uguale a tutti gli altri Step (vedere istruzione ST).
 IST è seguito da un elenco di transizioni in ingresso (I) e in uscita (O).

| | |
|-----------------|---|
| Utilizzo | IST numero ; Numero Step Iniziale 0-1999 elenco ; Elenco delle transizioni in ingresso e in uscita ; (lunghezza variabile) |
|-----------------|---|

Esempio

```

IST            1            ; Step Iniziale 1.
                 I 900       ;            Ingresso dalla Transizione 900.
                 O 1           ;            Uscita alla Transizione 1
...                            ;            Corpo di ST 1
EST                            ; Fine ST 1
    
```

Flag Alla partenza dello Step Iniziale, l'**ACCU** viene impostato al livello Alto (H)

Vedere anche EST, SB, ST

ST PASSO (Step)

Descrizione Definisce l'inizio di uno Step (ST).
 L'istruzione ST deve essere seguita da un elenco di Transizioni in Ingresso (I) e in Uscita (O).
 Uno Step dovrebbe contenere tipicamente solo istruzioni relative ad azioni quali SET, RES, OUT, LD, MOV, FADD ecc.
 Non deve MAI contenere cicli di attesa.
 Gli Step possono chiamare Blocchi Programma (PB) e Blocchi Funzione (FB) a patto che questi non contengano cicli di attesa.
 Nel GRAFTEC SAIA, non appena uno Step è stato eseguito, il puntatore del programma passa alla Transizione successiva.
 Gli SStep possono comparire solo all'interno di Blocchi Sequenziali.

Utilizzo

| | | |
|-----------|---------------|---|
| ST | numero | ; Numero Step, 0-1999 |
| | elenco | ; Elenco delle transizioni in ingresso e in uscita |
| | | ; (lunghezza variabile) |

Esempio

```

ST      10 ; Step 10
        I 9  ;      Ingresso dalla Transizione 9
        O 10 ;      Uscita alla Transizione 10
...
EST     ; Fine Step
  
```

Flag

Alla partenza dello Step, l'ACCU viene impostato al livello Alto (H)

Vedere anche

EST, IST, TR, SB, Manuale Utente

TR TRANSIZIONE (Transition)

Descrizione Definisce l'inizio di una Transizione (TR).
 L'istruzione TR deve essere seguita da un elenco di SStep in Ingresso (I) e in Uscita (O).
 Una Transizione dovrebbe tipicamente contenere istruzioni logiche combinate in modo tale per cui il risultato finale indichi se lo step seguente deve essere eseguito o meno.
 Se il risultato finale della Transizione è Falso (ACCU = L (0)), lo Step successivo non viene eseguito e l'esecuzione continua con la restante parte del COB.
 Alla prossima esecuzione del programma, verrà rielaborata per intero la Transizione in questione.
 Lo step successivo viene eseguito solo se il risultato finale della Transizione è vero (ACCU = H (1)).

In presenza di ramificazioni OR, l'ordine con cui vengono gestite le TR parallele dipende dall'ordine delle Transizioni di uscita definite dallo Step precedente.
 Le istruzioni TR possono comparire solo all'interno di Blocchi Sequenziali (SB).

Utilizzo

| | | |
|-----------|---------------|--|
| TR | numero | ; Numero Transizione 0-1999 |
| | elenco | ; Elenco degli Step in ingresso e in uscita |
| | | ; (lunghezza variabile) |

Esempio

```
TR          10 ; Transizione numero 10.
            I 900 ;      Ingresso dallo Step 900
            O 1  ;      Uscita allo Step 1
            O 2  ;      Uscita allo Step 2
            ... ;      Corpo di TR 10
ETR          ; Fine Transizione 10
```

Flag

Alla partenza dell TR, l'ACCU viene impostato al livello Alto (H).

See Also

ETR, SB, ST, Manuale Utente

8. Istruzioni di Comunicazione SERIALE

Queste istruzioni operano solo nei moduli di CPU che contengono porte seriali. Prima di effettuare qualsiasi comunicazione, è necessario eseguire l'istruzione SASI per ciascuna porta seriale (fino a 4). Questa istruzione configura la modalità operativa della porta e la velocità (baud rate). Ogni porta seriale può operare in modalità diversa e a diverse velocità. Ogni porta seriale dispone inoltre di propri buffer di ricezione e trasmissione.

| Modi | Funzione | Istruzioni Relative |
|-----------------|---|------------------------------|
| C | Trasmissione/Ricezione di caratteri ASCII | STXD, SRXD |
| | Trasmissione di testi interi | STXT |
| MD/SD | Trasmissione/Ricezione dati | STXM, SRXM |
| MM4 | Trasmissione/Ricezione di registri su rete LAC | STXM, SRXM |
| SBUS | Trasmissione/Ricezione dati su SAIA BUS | STXM(I), SRXM(I) |
| PROFIBUS | Trasmissione/Ricezione di dati su rete PROFIBUS | STXM(I), SRXM(I) SCON(I), |
| OFF | Disattiva assegnazione interfaccia seriale | - |

I segnali di controllo delle porte seriali (CTS, RTS, DSR, DTR e DCD) possono essere letti o impostati (SICL, SOCL). L'istruzione SCON permette di aprire o chiudere un canale PROFIBUS virtuale. Dopo aver eseguito l'istruzione SCON è anche possibile effettuare la comunicazione attraverso LAN 1.

SASI Assign serial interface (Assegna l'interfaccia seriale)

SASII Assign serial interface indirect
(Assegna l'interfaccia seriale con indirizzamento indiretto)

SRXD Serial receive character (Ricezione carattere su linea seriale)

STXD Serial transmit character (Trasmissione carattere su linea seriale)

STXT Serial transmit text (Trasmissione testo su linea seriale)

SRXM Serial receive media (Ricezione seriale dati)

STXM Serial transmit media (Trasmissione seriale dati)

SRXMI Serial receive media
(Ricezione seriale dati con indirizzamento indiretto)

STXMI Serial transmit media
(Trasmissione seriale dati con indirizzamento indiretto)

SICL Serial input control line (Lettura segnali di controllo)

SOCL Serial output control line (Comando segnali di controllo)

SCON Opens Serial communication channel to LAN 1 and PROFIBUS
(Apertura canale di comunicazione seriale su LAN 1 e PROFIBUS)

SCONI Opens Serial communication channel to LAN 1 and PROFIBUS indirect
(Apertura indiretta canale di comunicazione seriale su LAN 1 e PROFIBUS)

Tabella caratteri ASCII:
vedere l'ultima pagina del presente capitolo

MODALITA' C (Mode C)

Modalità Carattere o Testo:

- Vengono emessi caratteri singoli da un Registro oppure viene emesso un testo.
- I caratteri singoli possono essere ricevuti e trasferiti in un Registro.
- Utilizzata spesso per comunicare con terminali o stampanti

MC0

Modalità C senza "handshake" automatico:

L'utente deve farsi carico del controllo dei segnali di interfaccia utilizzando le istruzioni SICL e SOCL.

MC1

Modalità C con "handshake" RTS e CTS:

Il segnale di controllo RTS, viene comandato automaticamente dal PCD in funzione dello spazio rimanente nel buffer di ricezione.

Il segnale CTS influenza la trasmissione del PCD.

| | | |
|-----|-------|---|
| RTS | Basso | Il buffer di ricezione contiene più di 450 caratteri |
| | Alto | Il buffer di ricezione contiene meno di 300 caratteri |
| CTS | Basso | La trasmissione si arresta |
| | Alto | La trasmissione viene ripristinata |

MC2

Modalità C con protocollo Xon/Xoff:

Questa modalità è simile a quella basata su RTS/CTS ed è utilizzata quando non sono presenti i segnali di controllo (ad esempio nell'interfaccia current loop).

Per poter controllare la trasmissione del corrispondente, vengono trasmessi due caratteri speciali con funzione Xon (CTRL/Q) e Xoff (CTRL/S).

| | | |
|-----------------|--------|---|
| Il Ricevente | invia | quando |
| | Xoff | Il buffer di ricezione contiene più di 450 caratteri |
| | Xon | Il buffer di ricezione contiene meno di 300 caratteri |
| Il Trasmittente | riceve | quindi |
| | Xoff | La Trasmissione si arresta |
| | Xon | La Trasmissione viene ripristinata |

MC3

Modalità C con Eco:

Questa modalità viene utilizzata quando si comunica con un terminale; tutti i caratteri ricevuti vengono ritrasmessi al terminale

MC4

Modalità C per interfaccia RS485:

La modalità MC4 è una modalità di basso livello che imposta i driver/receiver RS485 in modo drive solo durante la trasmissione di informazioni (caratteri/testo) mentre in tutti gli altri casi (default) l'interfaccia viene impostata per la ricezione.

MODALITA' D (Mode D)

Fa uso di telegrammi secondo quanto previsto dai protocolli ISO 1745, IBM BSC e DIN 66019.

Tra due PCD o tra un PCD e un altro sistema intelligente (PC IBM ecc) collegati direttamente o attraverso la rete LAN1 SAIA, si possono scambiare dati specifici relativi ai PCD[®] SAIA.

I dati possono essere rappresentati dagli stati di Ingressi, Uscite o Flag; oppure dal contenuto di Registri, Temporizzatori o Contatori.

| <modo> | Descrizione |
|--------|-------------------|
| MD0 | Modalità D master |
| SD0 | Modalità D slave |

Le due modalità sono funzionalmente equivalenti; la sola differenza è rappresentata dal fatto che, quando si verifica un conflitto nella comunicazione full-duplex, la stazione master ha sempre priorità sulla stazione slave nel ripetere la richiesta.

Quando è in comunicazione con un PC, il PCD deve essere impostato come Slave (SD0).

Per la descrizione completa del protocollo, consultare le specifiche funzionali del protocollo P8 SAIA

SAIA LAN 1

La modalità D può essere utilizzata con la rete SAIA LAN1: in tal caso è possibile ottenere il collegamento tra due stazioni tramite l'istruzione SCON.

LAN1 scrive automaticamente lo stato del collegamento in un Registro, e l'indirizzo del Registro viene fornito nell'istruzione SASI.

Per ulteriori informazioni sulla rete SAIA LAN1, consultare il manuale relativo

MODALITA' MM4 (Mode MM4)

La modalità MM4 consente di collegare i PCD alla rete COMPEX LAC/LAC2. La LAC/LAC2 è una rete locale industriale che consente di collegare con facilità macchine intelligenti di tipo eterogeneo. Il PCD viene collegato alla rete attraverso un dispositivo di comunicazione che fornisce i servizi di trasmissione richiesti.

La modalità MM4 consiste nello scambio di Registri a 32 bit (4 caratteri ASCII); con un telegramma si possono trasferire 64 Registri.

Il controllo CRC-16 assicura la rilevazione degli errori.

Questa modalità può inoltre sostenere un collegamento punto a punto tra 2 PCD.

Per ulteriori informazioni consultare la "Descrizione del Protocollo LAC MM4".

MODALITA' SBUS (Mode SBUS)

La sigla S-Bus identifica un efficiente protocollo di comunicazione per la generazione di controllori SAIA® PCD. Questa soluzione può essere utilizzata sia per comunicazioni punto-punto che all'interno di una rete locale master/slave. Per le comunicazioni punto-punto è possibile usare una qualsiasi delle interfacce seriali del PCD. A livello fisico, una rete S-Bus sfrutta lo standard RS 485, attraverso un cavo schermato a due fili "twistati".

L'S-Bus rappresenta una semplice ed economica soluzione per la connessione in rete di fino a 255 sistemi PCD, suddivisi in un massimo di 8 segmenti, ciascuno dei quali contenente fino a 32 stazioni.

L'S-Bus presenta le seguenti principali caratteristiche:

- E' semplice da gestire (installazione, messa in servizio e programmazione da parte dell'utente)
- E' economico, dato che il protocollo S-Bus è già integrato in tutti i processori PCD. Questo significa che non è necessario adottare alcun processore addizionale dedicato alle comunicazioni.
- Permette un trasferimento dati sicuro, grazie al controllo CRC-16 per la rilevazione degli errori.
- Consente elevate velocità di trasmissione, grazie all'efficiente protocollo binario che permette di raggiungere velocità fino a 38,4 kbps.
- Supporta l'accesso ai dati e la diagnostica remoti per mezzo di un modem connesso a linee affittate o a commutazione.
- Permette l'uso dei Driver disponibili per sistemi di controllo e supervisione quali Wizcon, InTouch, FactoryLink, Fix D-Macs e Genesis.
- Adottando il Livello Applicazione 2 (assistenza/messa in servizio) l'unità di programmazione può accedere a tutte le stazioni slave presenti sulla rete. Questo significa che una qualsiasi stazione slave in rete può essere controllata per mezzo dell'unità di programmazione operando da un punto centrale (ad esempio, via debugger).
- Permette la realizzazione di reti Multi-master adottando il Gateway S-Bus.

| <modo> | Descrizione |
|--------|--|
| SM2 | Master S-Bus, modalità dati |
| SM1 | Master S-Bus, con controllo bit di parità |
| SM0 | Master S-Bus, nessuna parità, con carattere di interruzione |
| SS2 | Slave S-Bus, modalità dati |
| SS1 | Slave S-Bus, con controllo bit di parità |
| SS0 | Slave S-Bus, nessuna parità, con carattere di interruzione |
| GS2 | Slave Gateway S-Bus, modalità dati |
| GS1 | Slave Gateway S-Bus, con controllo bit di parità |
| GS0 | Slave Gateway S-Bus, nessuna parità, con carattere di interruzione |
| GM | Master Gateway S-Bus |
| OFF | Disattiva inizializzazione linea seriale |

Per maggiori informazioni, consultare il "Manuale SAIA S-BUS" (rif 26/739).

PROFIBUS

PROFIBUS rappresenta il più diffuso bus di campo per reti aperte, utilizzabile in un'ampia gamma di applicazioni.

PROFIBUS permette a dispositivi realizzati da costruttori diversi di comunicare tra loro senza necessitare di interfacce dedicate.

PROFIBUS è definito da uno standard Europeo (EN 50170).

PROFIBUS è indipendente dal tipo e dalla marca dei dispositivi. Molti rivenditori qualificati offrono sul mercato dispositivi che supportano questo standard.

PROFIBUS consiste in pratica in un assortimento di prodotti compatibili. Esistono tre principali varianti dello standard PROFIBUS dedicate a diverse aree di applicazione:

- PROFIBUS-DP Decentral Periphery
- PROFIBUS-FMS Fieldbus Message Specification
- PROFIBUS-PA Process Automation

I SAIA[®] PCD operano con i profili PROFIBUS-FMS e PROFIBUS DP

PROFIBUS-FMS rappresenta la soluzione universale per la comunicazione tra controllori e dispositivi intelligenti installati sul campo e per lo scambio di informazioni tra controllori.

Per poter realizzare una rete PROFIBUS utilizzando i SAIA[®] PCD è necessario adottare un processore dedicato ovvero un modulo PCD4.M445 o PCD7.F700.

La definizione e la configurazione (parametri bus, elenco delle relazioni per le comunicazioni e dizionario degli oggetti) di una rete PROFIBUS possono essere molto complicate e dipendono dalla dimensione del progetto. Per facilitare queste operazioni, SAIA ha sviluppato un programma di configurazione PROFIBUS operante in ambiente Windows. Questo configuratore crea un file contenente i testi per la definizione di tutti i canali PROFIBUS di una stazione. Questo file di definizione viene poi utilizzato in combinazione con l'istruzione SASI dedicata al canale PROFIBUS.

Per maggiori informazioni, consultare il "Manuale SAIA[®] PROFIBUS" (rif 26/742).

SASI ASSEGNA L'INTERFACCIA SERIALE (Assign Serial Interface)

Descrizione Inizializza un canale seriale.
 Il primo operando rappresenta il numero della porta seriale.
 Il secondo operando rappresenta il numero di un Testo che contiene la definizione della modalità di funzionamento della linea (vedere le pagine seguenti).
 Tale inizializzazione deve essere ripetuta per ciascuna porta seriale che si intende utilizzare. In genere, l'istruzione(i) SASI viene inserita nell'XOB 16. Le quattro linee seriali possono operare con modalità e velocità diverse. Per l'assegnazione di un canale PROFIBUS, consultare il manuale "SAIA[®] PROFIBUS" (rif 26/742).

| | |
|-----------------|--|
| Utilizzo | SASI canale ; Numero porta (canale) seriale, 0-3, [PROFIBUS: 10-99] nr. testo ; Numero testo di definizione 0-7999 |
|-----------------|--|

Esempio SASI 0 ; Inizializza la porta seriale 0
 100 ; utilizzando le definizioni contenute nel Testo 100

Flag Il flag **Errore** (E) viene impostato se il testo di definizione è assente o non è valido.

Vedere anche Testi SASI

Nota Per reti PROFIBUS, il numero canale (porta seriale - 10..99) identifica il canale virtuale (chiamato anche CREF). Tutte le informazioni necessarie per l'inizializzazione di tale canale sono contenute in un Testo generato automaticamente dal configuratore PROFIBUS.

Uso pratico Inizializzare la prima porta seriale (numero 0) per la modalità TESTO e con una velocità di 4800 baud, 7 bit di dati, parità pari e un bit di stop
 L'istruzione SASI viene collocata nell'XOB 16.

 XOB 16 ; Blocco ad Organizzazione Esclusiva eseguito all'accensione

SASI 0 ; Assegna la porta Seriale 0
 10 ; con i parametri contenuti nel Testo 10

 EXOB

```
TEXT 10 "UART:4800,7,E,1;"
        "MODE:MC0;"
        "DIAG:F1000,R4000;"
```

I Flag 1000 .. 1007 vengono utilizzati come flag diagnostici e il Registro 4000 è utilizzato come registro diagnostico.

Il TESTO (Text 10) può essere scritto anche in una sola riga:

```
TEXT 10 "UART:4800,7,E,1;MODE:MC0;DIAG:F1000,R4000;"
```

TESTI SASI (SASI Texts)

Per eseguire l'istruzione SASI (Assegnazione dell'Interfaccia Seriale) è necessario utilizzare un testo di definizione.

FORMATO:

| | |
|------------------|--|
| TEXT xxxx | "<uart_def>;" "<mode_def>;" "<diag_def>;" ["<rx_buf>;"] ["<tx_buf>;"] |
|------------------|--|

dove xxxx è un qualsiasi numero corrispondente ad un testo valido (0..3999)

Tale testo può anche essere scritto in una sola riga.

I diversi parametri sono:

- <uart_def> Definisce la velocità, la lunghezza dei dati, la parità,
- <mode_def> Definisce la Modalità Operativa (C, D, ...).
- <diag_def> Indirizzo dei Flag diagnostici e del Registro Diagnostico.

Gli ultimi due parametri sono opzionali e vengono utilizzati solo in modalità C:

- <rx_buf> Lunghezza del Buffer di ricezione (default = 1).
- <tx_buf> Lunghezza del Buffer di trasmissione.

MODALITA' OFF (Mode OFF)

Questa modalità differisce rispetto alla modalità testi SASI standard ed è principalmente utilizzata quando è necessario riassegnare un'interfaccia già assegnata.

Per evitare contese sull'interfaccia, è stato implementato un meccanismo a semaforo.

Quando viene assegnata una interfaccia seriale, viene attivato un semaforo, in modo tale, che se si tenta di riassegnare la stessa interfaccia seriale, si accenda il led di errore e venga interrotta l'istruzione. Una interfaccia seriale può essere riassegnata dopo che sia stata eseguita la DISASSEGNAZIONE cioè dopo aver eseguito l'istruzione SASI con il seguente testo.text:

TEXT xxxx "MODE:OFF"

Analogamente, se una CPU tenta di disassegnare una interfaccia seriale già disassegnata, viene richiamata la routine di gestione del Flag di Errore.

<uart_def>

Definisce la velocità (baud rate), la lunghezza del carattere, la parità, il numero di bit di stop, il valore del timeout. Formato:

"UART:<baud-rate>,<char_len>,<parity>,<stop_bit>[,<timeout>];

| <baud_rate> | <char_len> | <parity> | <stop_bit> | <time_out> | o per default |
|-------------|------------|-------------|------------|--------------|---------------|
| 110 | 7 | E (pari) | 1 | 10..15000 ms | 15 s |
| 150 | 8 | O (dispari) | 2 | | 9 s |
| 300 | | L (bassa) | | | 5 s |
| 600 | | H (alta) | | | 3 s |
| 1.200 | | N (nessuna) | | | 2 s |
| 2.400 | | | | | 1 s |
| 4.800 | | | | | 0,5 s |
| 9.600 | | | | | 0,25 s |
| 19.200 | | | | | 0,2 s |
| 38.400 | | | | | 0,1 s |

<time_out>:

Per la Modalità C, il time_out è irrilevante.

I valori di default dei timeout sono espressi in secondi e sono definiti in funzione della velocità. Nelle altre modalità il timeout è il tempo scaduto il quale il messaggio viene ripetuto, nel caso in cui il corrispondente non fornisca alcuna risposta positiva. Se, dopo due tentativi di ritrasmissione, il corrispondente non risponde, viene dichiarato "non connesso".

Modalità S-BUS:

In modalità SBUS vengono usati sempre 11 bit (10 bit per la modalità dati) per trasmettere un carattere: non deve essere fornita alcuna definizione per <char_len>, <parity> e <stop_bit>.

"UART:<Baudrate>[,<Timeout>][,<TS-Delay>][,<TN-Delay>][,<Break-Length>];"

<baud_rate>: 110 .. 38.400
 <time_out>: 10 .. 15000 ms
 <TS delay>: 10 .. 15000 ms
 <Break-Length>: 4 .. 15 caratteri

TimeOut, TS-Delay, TN-Delay e Break-Length sono opzionali e generalmente devono essere definiti solo per applicazioni speciali.

Per maggiori informazioni, consultare il "Manuale SAIA® S-BUS" (ref. 26/739).

Esempio di testo <uart_def> :

"UART:9600,7,E,1;"

9600 baud, 7 bit di dati, Parità pari, 1 bit di stop timeout di default

Nota: Tutti i caratteri devono essere digitati in Maiuscolo.

Quando il testo è compreso fra le direttive \$SASI e \$ENDSASI, l'assemblatore verifica la sintassi del testo e tutti i caratteri vengono convertiti in maiuscolo.

<mode_def>

Definisce la modalità operativa della linea seriale.

Formato: **"MODE: >[,<mode_opt>];"**

<mode_opt> rappresenta una serie di parametri opzionali che dipendono dalla modalità selezionata.

| <mode> | <mode_opt> | Descrizione |
|--------|------------|---|
| MC0 | - | Modalità C senza "handshake" automatico |
| MC1 | - | Modalità C con "handshake" RTS e CTS |
| MC2 | - | Modalità C con protocollo Xon/Xoff |
| MC3 | - | Modalità C con eco |
| MC4 | - | Modalità C per interfaccia RS485 |
| MD0 | - | Modalità D Master |
| | R xxxx (1) | via LAN1; Registro = stato SCON |
| SD0 | - | Modalità D Slave |
| | R xxxx (1) | via LAN1; Registro = stato SCON |
| SM2 | R xxxx (2) | Mod. SBUS Master (client) Modo Dati |
| SM1 | R xxxx (2) | Mod. SBUS Master (client) Modo Parità stazione remota |
| SM0 | R xxxx (2) | Modo Break |
| SS2 | - | Mod. SBUS Slave (server) Modo Dati |
| SS1 | - | Mod. SBUS Slave (server) Modo Parità |
| SS0 | - | Modo Break |
| GM | - | Modalità SBUS Gateway Master |
| GS2 | - | Mod. SBUS Gateway Slave Modo Dati |
| GS1 | - | Mod. SBUS Gateway Slave Modo Parità |
| GS0 | - | Modo Break |
| MM4 | (3) | Modalità MM4 |
| OFF | - | Disassegnazione della linea seriale. |

(1) Modalità D con LAN 1

Quando si utilizza SAIA LAN1 l'interfaccia PCA2.T9x usa un Registro per comunicare al PCD lo stato del collegamento. Per ulteriori informazioni, vedere l'istruzione SCON.

(2) Client SBUS

L'indirizzo della stazione partner remota viene inserito in un Registro.

(3) MM4 <mode_opt> è costituito dai seguenti parametri:

<BCS_opt>,<trpartner>,<trinfo>,<repartner>,<reinfo>,<rechar>

| <mode_opt> | Valore | Descrizione |
|-------------|--------|---|
| <BCS_opt> | 0 or 1 | Block Check Sum (0: nessun BCS, 1: CRC-16) |
| <trpartner> | R xxxx | Numero della stazione partner in trasmissione |
| <trinfo> | R xxxx | Informazione sull'ACK remoto |
| <repartner> | R xxxx | Numero della stazione partner in Ricezione |
| <reinfo> | R xxxx | Ricezione informazioni |
| <rechar> | R xxxx | Numero di caratteri ricevuti |

<diag_def>

Definisce i supporti diagnostici per le comunicazioni.

Formato:

"DIAG:<dia_elem>,<dia_reg>,"

| | Tipo | Descrizione |
|------------|------------------|---|
| <dia_elem> | O xxxx F xxxx | Indirizzo Base di 8 flag consecutivi (o Uscite) |
| <dia_reg> | R xxxx | Indirizzo di un registro diagnostico |

dove xxxx è un indirizzo valido

Gli 8 flag forniscono informazioni circa lo stato della linea seriale. In caso di errore durante l'esecuzione di un'istruzione di comunicazione seriale, si possono ottenere ulteriori informazioni esaminando il contenuto del registro diagnostico.

FLAG DIAGNOSTICI

L'indirizzo dell'Uscita o del Flag che segue la definizione DIAG dei testi SASI, rappresenta l'indirizzo di base di 8 Uscite o Flag consecutivi, usati nel modo seguente:

| Indirizzo | Nome | Descrizione |
|-----------|-------------|------------------------------|
| xxxx | RBSY | Ricevente occupato |
| xxxx+1 | RFUL | Buffer di ricezione pieno |
| xxxx+2 | RDIA | Diagnostica ricezione |
| xxxx+3 | TBSY | Trasmittente occupato |
| xxxx+4 | TFUL | Buffer di trasmissione pieno |
| xxxx+5 | TDIA | Diagnostica trasmissione |
| xxxx+6 | XBSY | Testo in trasmissione |
| xxxx+7 | NEXE | Non eseguito |

RBSY

Receiver Busy (Ricevente occupato)

| Modalità | |
|------------------------|--|
| C | RBSY è Alto (H) quando nel buffer di ricezione è disponibile almeno un carattere. Quando tutti i caratteri in attesa nel buffer di ricezione sono stati letti con l'istruzione SRDX, RBSY viene azzerato |
| D MM4 | RBSY è Alto quando il ricevente è occupato. |
| SBUS | RBSY è Alto (H) quando una stazione slave riceve un telegramma. Il flag viene azzerato non appena viene trasmesso il telegramma di risposta. |
| PROFIBUS | Non utilizzato |

RFUL

Receive Buffer Full (Buffer di ricezione pieno)

| Modalità | |
|------------------------|--|
| C | RFUL è Alto (H) quando il numero di caratteri ricevuti nel buffer di ricezione del PCD è uguale o maggiore rispetto al valore di rx_buf (lunghezza del Buffer di Ricezione). RFUL è Basso (L) quando il numero di caratteri rimanenti nel buffer di ricezione è inferiore al valore di rx_buf. Il buffer di ricezione interno del PCD ha sempre spazio per 512 caratteri. |
| D MM4 | RFUL è Alto (H) quando è stato ricevuto un frame di dati corretto. |
| SBUS | RFUL è Alto (H) quando la stazione master ha modificato dei dati nella stazione slave. |
| PROFIBUS | RFUL Alto (H) indica che è stato ricevuto un telegramma. |

RDIA

Receiver Diagnostic (Diagnostica ricezione)

| Modalità | |
|--|---|
| C D SBUS MM4 PROFIBUS | RDIA è Alto (H) se il PCD rileva un errore durante la ricezione di un carattere; ulteriori informazioni per quanto riguarda l'errore, si possono ottenere esaminando il contenuto del registro diagnostico della comunicazione. RDIA verrà azzerato quando tutti i bit di diagnostica ricezione (0...15) del registro diagnostico verranno azzerati. |

TBSY

Transmitter Busy (Trasmittente occupato)

| Modalità | |
|--|--|
| C | TBSY è Alto (H) quando il PCD trasmette caratteri sulla linea seriale. TBSY è basso (L) quando sono stati trasmessi tutti i caratteri del buffer di trasmissione. |
| D SBUS MM4 PROFIBUS | TBSY è impostato ad un valore Alto (H) quando il PCD è in fase di trasferimento dati. TBSY è impostato ad un valore Basso (L) quando è stata fornita risposta al telegramma oppure quando è stato raggiunto il numero di tentativi |

TFUL

Transmit Buffer Full (Buffer di trasmissione pieno)

| Modalità | |
|-----------------|---|
| C | TFUL è Alto (H) quando il numero di caratteri rimanenti nel buffer di trasmissione del PCD è maggiore o uguale al valore dichiarato in tx_buf (lunghezza del Buffer di trasmissione). TFUL viene azzerato quando il numero di caratteri rimasti nel buffer di trasmissione è inferiore al valore di TBUF |
| D | Non utilizzato |
| SBUS | |
| PROFIBUS | |
| MM4 | TFUL è Alto (H) quando è stata ricevuta una risposta positiva. |

TDIA

Transmitter Diagnostic (Diagnostica trasmissione)

| Modalità | |
|--|--|
| C D SBUS MM4 PROFIBUS | TDIA è Alto (H) quando il PCD rileva un errore durante la Trasmissione di un carattere; ulteriori informazioni per quanto riguarda l'errore si possono ottenere esaminando il contenuto del registro diagnostico della comunicazione. TDIA verrà azzerato quando tutti i bit di diagnostica trasmissione (16...31) del registro diagnostico verranno azzerati. |

XBSY

Text busy (Testo in trasmissione)

| Modalità | |
|-----------------|---|
| C | XBSY è Alto (H) quando il PCD trasmette un testo (STXT); dopo aver trasmesso tutto il testo, XBSY viene azzerato. Nota: XBSY viene azzerato <u>all'inizio</u> della trasmissione dell'ultimo carattere. |
| D | XBSY è Alto (H) quando è aperto un collegamento mediante la rete LAN1. |
| SBUS | XBSY è Basso (L) quando l'utente ha il permesso di eseguire un'istruzione SASI OFF. |
| MM4 | XBSY è Alto (H) quando si rileva un'attività sulla rete LAC (istruzione STXM) |
| PROFIBUS | Trasmissione incrociata / canale aperto |

NEXE

Not executed (Non eseguito)

| Modalità | |
|---|---|
| C D SBUS MM4 | Se il PCD non è in grado di eseguire l'operazione richiesta, NEXE viene posto al valore Alto (H); ulteriori informazioni per quanto riguarda l'errore si possono ottenere esaminando il registro diagnostico della comunicazione. |
| PROFIBUS | Impostato al livello Alto (H) quando, dopo 3 tentativi, non è stato possibile eseguire un'istruzione (STXM o RSXM). Il flag viene azzerato all'esecuzione dell'istruzione successiva. |

REGISTRO DIAGNOSTICO

Con la definizione DIAG del testo SASI deve essere fornito anche l'indirizzo di un Registro Diagnostico. Normalmente, tutti i 32 bit di questo Registro Diagnostico sono Bassi (0). Se un bit è alto (H) si può individuare, il suo significato nella tabella che segue (vedere la modalità interessata):

| Bit | Descrizione | Causa | MODALITÀ | | | | |
|-----|-------------------------------------|--|----------|---|------|-----|-----------|
| | | | C | D | SBUS | MM4 | PROFI BUS |
| 0 | Errore di "overrun" | Non dovrebbe mai verificarsi, segnalarlo alla SAIA | ● | ● | ● | ● | |
| 1 | Errore di parità | Ricevuto un carattere con parità errata | ● | ● | | ● | |
| 2 | Errore di "Framing" | Solitamente causato da "baud rate" errato | ● | ● | ● | ● | |
| 3 | Break | Linea dati interrotta | ● | ● | ● | ● | |
| 4 | Errore BCC / CRC 16 | Blocco di Controllo Errato (o CRC-16) | | ● | ● | ● | |
| 5 | Stato S-BUS PGU | S-BUS PGU con Modem linea pubblica | | | ● | | |
| 6 | Fine trasmissione | Transmiss. conclusa / SASI OFF ammessa | | ● | | | |
| 7 | Errore di Overflow | Superamento capacità buffer di ricezione | ● | ● | | ● | |
| 8 | Errore di lunghezza | Lunghezza telegramma non valida | | | ● | | |
| 9 | Errore di formato | Formato telegramma non valido | | ● | | ● | ● |
| 10 | Errore di telegramma | Telegramma non valido | | | ● | ● | |
| 11 | Errore di stato | Il PCD si trova in uno stato non valido. | | | ● | | |
| 12 | Errore di campo | L'indirizzo dell'elemento è errato | | ● | ● | | ● |
| 13 | Valore errato | Valore del dato non valido | | | | | |
| 14 | Errore supp. mancante | Indirizzo supporto non definito/non valido | | | ● | | |
| 15 | Errore di programma | Lettura da buffer di ricezione vuoto | ● | | | | |
| | | LAN 1 non assegnata o numero di stazione non valido | | ● | ● | | |
| 16 | Conteggio tentativi di trasmissione | Indica il numero di tentativi di trasmissione (in binario) | | | | | |
| 17 | | | | | | | |
| 18 | Interr. trasmissione | Trasmissione sospesa (CTS = L o XOFF) | ● | | | | |
| 19 | | | | | | | |
| 20 | Risposta NAK | È stato ricevuto un NAK | | ● | ● | ● | ● |
| 21 | Nessuna risposta | Nessuna risposta ricevuta dopo il timeout | | ● | ● | ● | |
| 22 | NAK Multiplo | NAK ricevuto dopo più tentativi | | ● | ● | ● | |
| 23 | Buffer TX pieno | Manca spazio nel buffer di trasmissione | ● | | | | |
| | Ritardo TS | Nessun CTS dopo il Ritardo TS | | | | | |
| 24 | Errore di Enquiry | Nessuna risposta a ENQ (dopo + tentativi) | | ● | | | |
| 25 | Errore di formato | Testo di definizione non valido | ● | | | | |
| | | Comando non valido | | ● | | | ● |
| 26 | Errore con il Partner | Esiste un problema lato corrispondente | | | | ● | |
| 27 | Errore di Rete | Si è verificato un problema in rete | | | | ● | |
| 28 | Errore di campo | L'indirizzo dell'elemento è errato | | ● | ● | ● | ● |
| 29 | | | | | | | |
| 30 | Errore di ricezione | È avvenuto un errore | | ● | | | ● |
| 31 | Errore di programma | Tentativo di trasmissione non autorizzato | | ● | ● | ● | ● |

<rx_buf> e <tx_buf> sono utilizzati solo in modalità C

<rx_buf>

Definisce i limiti del buffer di ricezione.

Formato:

"RBUF:<rbuf_len>";

| | Valore | Descrizione |
|------------|---------|-------------------------------|
| <rbuf_len> | 1.. 511 | Lunghezza buffer di ricezione |

Il Buffer di Ricezione ha sempre lo spazio per 512 caratteri di 8 bit.
 Per la Modalità C, la definizione RBUF (1-511) indica quando deve essere attivato lo stato di buffer pieno (RFUL).
 Per le altre modalità RBUF non è utilizzato.

<tx_buf>

Definisce i limiti del buffer di trasmissione.

Formato:

"TBUF:<tbuf_len>";

| | Valore | Descrizione |
|------------|---------|----------------------------------|
| <tbuf_len> | 1.. 511 | Lunghezza buffer di trasmissione |

Simile al Buffer di Ricezione.
 Per la Modalità C, la definizione TBUF (1-511) indica quando deve essere attivato lo stato di buffer pieno (TFUL).
 Per le altre modalità RBUF non è utilizzato.

Esempi di Testi SASI

Modo MC0 a 9600 Baud, 7 bit di dati, parità pari, 1 bit di stop, utilizzo di F1000-F1007 come flag diagnostici e di R4000 come registro diagnostico:
 TEXT 10 "UART:9600,7,E,1;MODE:MC0;DIAG:F1000,R4000;"

Modo MC2 a 4800 Baud, 8 bit di dati, nessuna parità, 1 bit di stop, utilizzo di F0-F7 come flag diagnostici e di R4000 come registro diagnostico, buffer di ricezione lungo 25 caratteri:
 TEXT 20 "UART:4800,8,N,1;MODE:MC2;DIAG:F0,R100;"
 "RBUF:25;"

Modo SD0 (Slave) a 9600 Baud, 7 bit di dati, parità pari, 1 bit di stop, utilizzo di F1000-F1007 come flag diagnostici e di R4000 come registro diagnostico:
 TEXT 30 "UART:9600,7,E,1;MODE:SD0;DIAG:F1000,R4000;"

Modo MD0 (Master) a 9600 Baud, 7 bit di dati, parità pari, 1 bit di stop, utilizzo di F1000-F1007 come flag diagnostici e di R4000 come registro diagnostico; utilizzo della LAN1 SAIA e timeout di 3 secondi:
 TEXT 40 "UART:9600,7,E,1,3000;"
 "MODE:MD0,R1;"
 "DIAG:F1000,R4000;"

Il Registro R 1 è utilizzato per memorizzare lo stato della connessione LAN1.

Modo MM4 a 9600 Baud, 8 bit di dati, nessuna parità, 1 bit di stop, timeout a 300 ms, BCS assente, impiego dei Registri 100..104 per il numero del partner remoto, ...
 Utilizzo di F1000-F1007 come flag diagnostici e di R1000 come registro diagnostico:
 TEXT 50 "UART:9600,8,N,1,300;"
 "MODE:MM4,0,R100,R101,R102,R103,R104;"
 "DIAG:F1000,R1000;"

Modo SBUS Modo Parità (Master) a 9600 Baud, utilizzo del Registro 555 contenente il numero del partner, utilizzo di F8000-F8007 come flag diagnostici e di R4005 come registro diagnostico:
 TEXT 60 "UART:9600;MODE:SM1,R555;DIAG:F8000,R4005;"

Modo SBUS Modo Parità (Slave) a 9600 Baud, utilizzo di F8000-F8007 come flag diagnostici e di R4005 come registro diagnostico:
 TEXT 60 "UART:9600;MODE:SS1;DIAG:F8000,R4005;"

Modo SBUS Modo Dati (Slave) a 9600 Baud, utilizzo di F8000-F8007 come flag diagnostici e di R4005 come registro diagnostico:
 TEXT 60 "UART:9600;MODE:SS2,R55;DIAG:F8000,R4005;"

Modo PROFIBUS SASI 10 ; canale 10
 T_As_10 ; Testo SASI

Il testo SASI "T_As_10" viene generato dal configuratore PROFIBUS.

Utilizzo di SIMBOLI nei testi

Nei testi SASI si possono utilizzare dei simboli.

Il valore e opzionalmente il tipo del simbolo vengono inseriti all'interno del testo. Il simbolo viene scritto al di fuori del segmento di testo ASCII tra virgolette, e deve essere separato dagli altri simboli per mezzo di una virgola. Dopo il simbolo, si possono specificare (come opzioni) la larghezza del campo e il tipo di prefisso.

Formato:

| simbolo [. [[-] [0] larghezza] [t T]] | |
|---|---|
| simbolo | Nome del simbolo. Può essere in effetti una qualsiasi espressione che include un simbolo ad esempio: MotorOn + 100, ... Non sono ammessi simboli con valori in virgola mobile. |
| . | Il punto immediatamente dopo il simbolo indica che esiste un campo larghezza e/o un prefisso. |
| larghezza | É il campo larghezza: numero di cifre o di spazi richiesti dal numero. Se la larghezza inizia con 0, vengono inseriti gli zeri non significativi. |
| t T | Prefisso opzionale tipo 't' o 'T'. Se 't', il valore viene preceduto dal tipo del simbolo in minuscolo (o, f, r, ...); se 'T', il tipo del simbolo sarà in maiuscolo (O, F, R,...). |

Esempi:

```
BAUD      EQU      9600
D_FLAGS  EQU      F 500
D_REG     EQU      R 4095

          XOB      16
          SASI     1
          3999
```

```
TEXT 3999      "UART: ", BAUD, ", 7, E, 1; MODE: MC0; "
               "DIAG: ", D_FLAGS.T, ", ", D_REG.T, "; "
```

```
EXOB
```

Il testo risultante sarà:

```
"UART: 9600, 7, E, 1; MODE: MC0; "
"DIAG: F500, R4095; "
```


\$\$SASI, \$ENDSASI

Si possono utilizzare queste direttive assembler per delimitare i testi utilizzati dall'istruzione SASI. Tutti i testi racchiusi tra queste direttive vengono controllati dall'assembler che ne rileva gli eventuali errori.

Formato:

```

$SASI
<definizione testo SASI>
...
$ENDSASI
    
```

Se non si utilizzano \$\$SASI .. \$ENDSASI, si può introdurre un testo non valido che potrebbe causare una inizializzazione errata della porta seriale.

Esempio:

```

XOB      16
...
SASI     0      ; Inizializzazione porta seriale 0
          100   ; utilizzando il Testo 100
...
EXOB
    
```

```

$SASI    ; Il testo 100 viene controllato come Testo SASI dall'Assemblatore
TEXT 100 "UART:9600,7,E,1;MODE:MC0;DIAG:F1000,R4000;"
$ENDSASI
    
```

SASII ASSEGNA INTERFACCIA SERIALE INDIRECTA (Assign Serial Interface Indirect)

Descrizione Inizializza un canale seriale o un canale PROFIBUS tramite un indirizzamento indiretto- Questa istruzione opera analogamente all'istruzione SASI. La differenza è rappresentata dal fatto che l'istruzione in oggetto opera in modo indiretto, ovvero il numero del canale ed il testo di definizione possono essere contenuti all'interno di Registri.

| | | | |
|-----------------|--------------|-----------------------------|---|
| Utilizzo | SASII | canale | ; Numero porta (canale) seriale o Registro |
| | | testo di definizione | ; Registro |

Canale

Numero della porta (canale) seriale da inizializzare

Questo parametro può essere fornito in modo diretto o indiretto:

| | |
|-----------|---|
| 0..3 | Numero porta seriale |
| 10..99 | Numero canale PROFIBUS |
| R 0..4095 | Registro contenente il numero del canale (0..3, 10..99) |

Testo di definizione

Questo parametro indica un numero di Registro (R 0..4095)

Questo registro contiene l'indirizzo di un testo in cui sono definiti i parametri dell'interfaccia. Indirizzi validi per tale testo sono:

| | |
|------------|------------------------|
| 0..3999 | nella memoria standard |
| 4000..7999 | nella memoria estesa |

Esempio

| | | |
|-------|---|--|
| SASII | 0 | ; Inizializza porta seriale 0 |
| R 1 | | ; utilizzando il testo di definizione il cui |
| | | ; indirizzo è contenuto in R 1 |

Flag Il flag **Errore** (E) viene impostato se il testo di definizione specificato è mancante o non valido

Vedere anche Testi SASI

SRXD RICEZIONE CARATTERE SU LINEA SERIALE (modalità C) (Serial Receive Character - Mode C)

Descrizione Carica il prossimo carattere ASCII presente nel buffer della porta specificata dal 1° operando, nel Registro specificato dal 2° operando.

L'istruzione SRXD deve essere eseguita solo se RBSY indica che c'è un carattere pronto (RBSY = H), altrimenti viene impostato il Flag di Errore.

Dopo l'esecuzione di SRXD, gli 8 bit meno significativi del registro contengono il carattere, gli altri bit del Registro vengono forzati a 0.

Nel Buffer di Ricezione possono entrare fino a 512 caratteri. Ad ogni esecuzione di SRXD, viene letto il carattere successivo.

Se il Buffer di Ricezione va in "overrun" (più di 512 caratteri), verrà segnalato un Errore di ricezione (con impostazione del flag RDIA e del corrispondente bit di stato nel Registro Diagnostico delle linee seriali).

Utilizzo

| | | | |
|----------------|---------------|------------|---|
| SRXD[X] | canale | (i) | ; Numero porta (canale) seriale, 0-3 |
| | reg | | ; Registro in cui ricevere i caratteri, R 0-4095 |

Esempio

SRXD 3 ; Legge un carattere dalla porta 3
 R 100 ; e lo memorizza nel Registro 100

Flag

Il flag **Errore (E)** viene impostato se l'istruzione SRXD viene eseguita con il buffer di ricezione vuoto o se la porta seriale non è stata inizializzata correttamente o non esiste.

Vedere anche

STXD, SRXM, Istruzioni di Comunicazione, Flag Diagnostici

Uso pratico

Applicazione tipica in un programma strutturato Bloctec:

```

....
STH        F RBSY                    ; Se c'è un carattere in attesa
CFB        H READ_CHAR             ; Legge tale carattere
. . . . .

FB                READ_CHAR         ; FB di lettura carattere
[ STH        H RDIA ]                ; Se c'è un Errore di Ricezione
[ CFB        H RCV_ERROR ]         ; Allora gestisce l'errore
SRXD        0                        ; Legge il carattere dalla porta 0
                  R 999               ; e lo memorizza in R 999
. . . . .
EFC
```

Nota: Nelle applicazioni più semplici, la gestione dell'errore (istruzioni tra parentesi quadre) può essere omessa.

STXD TRASMISSIONE CARATTERE SU LINEA SERIALE (Modalità C) (Serial Transmit Character - Mode C)

Descrizione Il carattere presente nei bit meno significativi del Registro specificato dal 2° operando, viene caricato nel buffer di trasmissione della porta seriale specificata dal 1° operando. Il carattere viene quindi trasmesso automaticamente.

Il Buffer di Trasmissione può contenere fino a 512 caratteri. Se il buffer è vuoto (tutti i caratteri sono stati trasmessi) viene impostato a livello Basso (L) il flag di stato TBSY. Fin tanto che vi sono caratteri in attesa di trasmissione TBSY rimane al livello Alto (H).

Se il flag di stato TDIA è Alto dopo l'esecuzione di una STXD, significa che esiste un problema e che, quindi, occorre esaminare il Registro Diagnostico.

Utilizzo

| | | | |
|----------------|---------------|------------|---|
| STXD[X] | canale | (i) | ; Numero porta (canale) seriale, 0-3 |
| | reg | | ; Registro contenente il carattere, R 0-4095 |

Esempio

STXD 1 ; Trasmette il carattere contenuto nel
R 100 ; Registro 100 (bit 7-0) sulla porta seriale 1

Flag

Il flag **Errore** (E) viene attivato se la porta seriale non è stata inizializzata correttamente o non esiste.

Vedere anche

SRXD, STXT, Istruzioni di Comunicazione, Flag Diagnostici

Uso pratico

Applicazione tipica in un programma strutturato Bloctec:

```

...
STL        F  TFUL                ; Se c'è spazio nel buffer TX
CFB        H  SEND_CHAR        ; Allora trasmette un carattere
. . . .

FB                SEND_CHAR        ; FB di trasmissione carattere
STXD            0                ; Invia sulla porta seriale 0
                  R  900           ; il carattere memorizzato in R 900
[STH        H  TDIA]               ; Se c'è un Errore di Trasmissione
[CFB        H  SND_ERROR]        ; Allora gestisce l'errore
. . . .
EFB

```

Nota: Nelle applicazioni più semplici, la gestione dell'errore (istruzioni tra parentesi quadre) può essere omessa.

Testi

I Testi possono essere definiti in qualsiasi punto del programma Sorgente Assembler, ma vengono inseriti in una specifica area di memoria del PCD allocata per il testo. I Testi possono essere scritti immediatamente dopo le rispettive istruzioni di riferimento; in alternativa, tutti i testi possono essere scritti in un modulo sorgente Assembler separato dal resto del programma.

Devono essere osservate le seguenti regole:

- Un testo viene definito con:
TEXT n "Testo vero e proprio"
 dove 'n' è il numero del testo (0..7999)
- Il testo può essere costituito da diverse righe, ognuna delle quali deve essere racchiusa tra virgolette: " ". I Testi possono essere di qualunque lunghezza.
- I caratteri di controllo si possono inserire racchiusi tra parentesi angolari: <nnn>. Ad esempio: <LF>, <CR>, <FF>, <ESC>, ...
- I caratteri di controllo con codice ASCII decimale (1.. 31), o i caratteri speciali (codice 127..255) possono essere inseriti come valori decimali racchiusi tra parentesi angolari :<nnn>. Ad esempio: CR = <13>, LF = <10>, ESC = <27>, BELL = <7>, ...
- I caratteri ASCII standard (codice 32..126) possono essere introdotti direttamente da tastiera.

Nella memoria del PCD, tutti i testi terminano con un carattere NUL (codice ASCII 0) che viene aggiunto automaticamente alla fine del testo dall'Assembler. Un testo non può quindi contenere il carattere NUL.

Esempi:

I due testi che seguono generano lo stesso risultato:

```
TEXT 10  "The quick brown fox jumps over the lazy dog"
TEXT 11  "The quick brown fox"
          "jumps over the lazy dog"
```

Se, dopo il testo, occorre trasmettere un "Line Feed" e un "Carriage Return":

```
TEXT 12  "The quick brown fox jumps over the lazy dog<CR><LF>"
```

Se si dispone di una stampante EPSON, si può stampare una parte del testo in grassetto, inviando alla stampante stessa speciali caratteri di controllo. Supponendo di voler stampare il seguente testo:

Per il controllore programmabile **SAIA PCD4** non vi sono limiti

```
TEXT 13  " Per il controllore programmabile"
          "<ESC>E SAIA PCD4 <ESC>F"
          " non vi sono limiti <LF><CR>"
```

Nota: Inviando i caratteri ESC E la stampante stampa in grassetto mentre inviando i caratteri ESC F essa ritorna alla modalità di stampa normale

Testi e Variabili

I testi possono anche contenere variabili come: il valore dell'orologio, lo stato di un Ingresso, il contenuto di un Registro....

Due caratteri assumono un significato speciale nel PCD: \$ and @

\$ = INDIRIZZAMENTO DIRETTO

Viene fornito il numero dell'elemento assoluto.

| | | |
|---------|---|---------------------------|
| \$H | Ora (Ore,Minuti,Secondi): hh:mm:ss | |
| \$HH | Ora (solo Ore): hh | |
| \$HM | Ora (solo Minuti): mm | |
| \$HS | Ora (solo Secondi): ss | |
| \$D | Data (Anno, Mese, Giorno): aa-mm-gg | |
| \$d | Data (Giorno, Mese, Anno): gg.mm.aa | |
| \$DD | Data (solo Giorno): gg | |
| \$DM | Data (solo Mese): mm | |
| \$DY | Data (solo Anno): aa | |
| \$W | Settimana (N° Sett., Giorno Sett.): ss-gg | |
| \$WN | Settimana (solo N° Settimana): ss | |
| \$WD | Settimana (solo N° Giorno): gg | |
| \$nnnn | Stato logico di un singolo Ingresso (0, 1) | nnnn : |
| \$onnnn | Stato logico di una singola Uscita (0, 1) | numero elemento |
| \$fnnnn | Stato logico di un singolo Flag (0, 1) | (deve essere di 4 cifre) |
| \$Innnn | Stato logico di 8 Ingressi (nnnn a nnnn+7) | nnnn: numero |
| \$Onnnn | Stato logico di 8 Uscite (nnnn a nnnn+7) | primo elemento |
| \$Fnnnn | Stato logico di 8 Flag (nnnn a nnnn+7) | (deve essere di 4 cifre) |
| \$Cnnnn | Contenuto del contatore | nnnn: |
| \$Rnnnn | Contenuto del registro | numero elemento |
| \$Tnnnn | Contenuto del Temporizzatore | (deve essere di 4 cifre) |
| \$Lnnnn | incLude un altro testo (max 3 livelli) | nnnn:numero testo |
| | | (deve essere di 4 cifre) |
| \$xnn | Il carattere 'x' viene ripetuto 'nn' volte Il carattere non deve essere un tipo dati: (H D W i o f I O F C R L x) | nn deve essere di 2 cifre |
| \$Annnn | Contenuto del registro di uscita come carattere ASCII | nnnn: numero Registro |
| | | (deve essere di 4 cifre) |

Esempio di \$Annnn:

```
"$A0999"  dove R 999 = 00000000 hex      'NUL'
"$A0999"  dove R 999 = 00000061 hex      'a'
"$A0999"  dove R 999 = 00006162 hex      'ab'
"$A0999"  dove R 999 = 00616263 hex      'abc'
"$A0999"  dove R 999 = 61626364 hex      'abcd'
```

Gli zeri che precedono il valore non vengono inviati in uscita. Uno zero ASCII viene inviato in uscita solo se il valore del byte meno significativo è uguale a 0.

@ = INDIRIZZAMENTO INDIRETTO

Viene fornito il numero dell'elemento contenuto in un Registro

| | | |
|--------|--|---|
| @innnn | Stato logico di un singolo Ingresso (0, 1) | nnnn : numero Registro (deve essere di 4 cifre) |
| @onnnn | Stato logico di una singola Uscita (0, 1) | |
| @fnnnn | Stato logico di un singolo Flag (0, 1) | |
| @Innnn | Stato logico di 8 Ingressi (da Ind. a Ind.+7) | |
| @Onnnn | Stato logico di 8 Uscite (da Ind. a Ind.+7) | |
| @Fnnnn | Stato logico di 8 Flag (da Ind. a Ind.+7) | |
| @Cnnnn | Contenuto del contatore | |
| @Rnnnn | Contenuto del registro | |
| @Lnnnn | incLude un altro testo (max 3 livelli) | |
| @xnnnn | Il carattere 'x' viene ripetuto per il numero di volte definito dal contenuto del Registro:(H D W i o f I O F C R L x) | |

NOTA: Per emettere un carattere '\$' occorre utilizzare la notazione "\$\$" mentre per emettere un carattere '@' usare la notazione "@@".

Esempio 1:

```
TEXT 10 "Data: $D Ora: $H <CR><LF>"
"$-32 <CR><LF>"
"Ingressi 0..7: $I0000 <CR><LF>"
"Registro 100: $R0100 <CR><LF>"
"$+20 <CR><LF>"
```

Supponendo che questo testo venga stampato il 16 Agosto 91 alle 09:00 del mattino; che gli Ingressi 0 e 1 siano alti, e che il contenuto del Registro 100 sia 12345, verrebbe stampato il testo seguente:

```
Data: 90-08-16  Ora: 09:00:00
-----
Ingressi 0..7: 11000000
Registro 100:  12345
+++++
```

Esempio 2:

Utilizzo pratico di "\$A...."

2 registri, uno per la posizione X e uno per la Y, dovrebbero determinare la posizione del cursore sullo schermo: posizione X da Registro R 1 (1..80)
posizione Y da Registro R 2 (1..25)

La sequenza "escape" per il posizionamento del cursore è:

```
<27><17><valore di X><valore di Y>
```

E' possibile programmare: "...<27><17>\$A0001\$A0002..."

Ad esempio, per specificare una posizione fissa con X = 40 e Y = 12, si può scrivere l'intera sequenza di 4 caratteri in un solo registro ed inviarla in uscita utilizzando \$A....

Si noti che tutti i valori devono essere specificati in formato esadecimale:

ESC = 1B hex; 17 = 11 hex; 40 = 28 hex (valore di X); 12 = 0C hex (valore di Y)

```
Caricamento Registro R 1000:          LD  R 1000
                                         1B11280CH
```

Emissione Testo per posizionamento cursore: "... \$A1000 ..."

FORMATI DI OUTPUT (Output Formats)

Nel testo si può anche specificare il formato con cui trasmettere i dati contenuti in un Registro o in un Contatore. Possono essere specificate anche la larghezza del campo e il numero di decimali. Le definizioni di formato possono essere introdotte con il testo "\$%xxxx", dove 'xxxx' rappresenta il formato richiesto (vedere oltre). Se viene inviata in output una tale definizione, tutti i valori dei Registri o dei contatori che seguono, vengono emessi secondo questo formato, finchè non si incontra una nuova definizione di formato.

Nelle definizioni di formato che seguono, 'd' significa 'decimale', e = carattere spazio. Attualmente non vengono supportati altri formati (h=Esadecimale, b=Binario, o=Ottale). Se il valore è troppo grande per essere inserito nel campo, viene adottata la formattazione di default (cioè nessuna formattazione).

Definizione del formato di output:

Supponiamo che i Registri 10, 11 e 12 contengano rispettivamente i seguenti valori costanti: 123456, -7890 e 5.

NESSUNA FORMATTAZIONE (DEFAULT):

La larghezza del campo dipende dalla dimensione del numero.

```
TEXT 0    "REGISTRO 10: $R0010 <10><13>"
          "REGISTRO 11: $R0011 <10><13>"
          "REGISTRO 12: $R0012"
```

Output:

```
REGISTRO 10: 123456
REGISTRO 11: -7890
REGISTRO 12: 5
```

CAMPO A LARGHEZZA FISSA:

Utilizzare la definizione di formato "\$%xxd", dove 'xx' (1 - 99) rappresenta la larghezza del campo.

"\$%xxd": Il valore viene giustificato a destra **con spazi iniziali**.

```
TEXT 1    "$%08dREGISTRO 10: $R0010 <10><13>"
          "REGISTRO 11: $R0011 <10><13>"
          "REGISTRO 12: $R0012"
```

Output:

```
REGISTRO 10:  123456
REGISTRO 11:  -7890
REGISTRO 12:    5
```

"\$%xxD": Il valore viene giustificato a destra **con zeri iniziali**.

```
TEXT 1    "$%08DREGISTRO 10: $R0010 <10><13>"
          "REGISTRO 11: $R0011 <10><13>"
          "REGISTRO 12: $R0012"
```

Output:

```
REGISTRO 10: 00123456
REGISTRO 11: -0007890
REGISTRO 12: 00000005
```

CAMPO A LARGHEZZA FISSA e NUMERO FISSO DI CIFRE DECIMALI:

Il valore viene giustificato a destra, ma il numero di cifre decimali viene sempre visualizzato e normalizzato a destra con degli zeri.

Utilizzare la definizione di formato "\$%xx.yd", dove 'xx' è la larghezza totale del campo e 'y' è il numero di cifre a destra del punto decimale.

```
TEXT 2    "$%07.3dREGISTRO 10: $R0010 <10><13>"  
          "REGISTRO 11: $R0011"<10><13>"  
          "REGISTRO 12: $R0012"
```

Output:

```
REGISTRO 10: 123.456  
REGISTRO 11: -7.890  
REGISTRO 12: 0.005
```

SOLO NUMERO FISSO DI CIFRE DECIMALI:

Il numero di cifre decimali è fisso ma la larghezza del campo dipende dalla dimensione del numero.

Occorre utilizzare la definizione di formato "\$%00.yd", dove 'y' è il numero di cifre decimali normalizzate a destra con l'aggiunta di zeri se necessario.

```
TEXT 2    "$%00.5dREGISTRO 10: $R0010 <10><13>"  
          "REGISTRO 11: $R0011"<10><13>"  
          "REGISTRO 12: $R0012"
```

Output:

```
REGISTRO 10: 1.23456  
REGISTRO 11: -0.07890  
REGISTRO 12: 0.00005
```

RIMOZIONE FORMATTAZIONE:

"\$%00d" imposta il formato standard (nessuna formattazione).

Salvataggio / Ripristino delle definizioni di formato:

Le definizioni di formato possono essere salvate utilizzando "\$sn", dove 'n' è un 'numero d'ordine per il salvataggio'.

Si possono salvare fino a 10 definizioni di formato (0-9).

I formati salvati possono essere richiamati utilizzando "\$n", dove 'n' è il numero d'ordine del formato da ripristinare.

I formati possono essere salvati nell'ambito del processo di inzializzazione, ovvero all'interno dell'XOB 16, ossia l'XOB di partenza. Per salvare un formato, il Testo contenente il formato in oggetto deve essere emesso sulla linea seriale con l'istruzione STXT. Se si cerca di ripristinare un formato non precedentemente salvato, viene utilizzato il formato di default (nessuna formattazione).

Esempio:

```

XOB          16
. . . .
TEXT 991 "$%05.1d$s1"      ; Definizione formato 1 (nnn.n)
TEXT 992 "$%04.2d$s2"      ; Definizione formato 1(n.nn)
TEXT 993 "$%08.3d$s3"      ; Definizione formato 1(nnnn.nnn)

DEF:         SEI          K 0          ; Attivazione definizioni di formato
            STH          XBSY
            JR           H DEF
            STXTX        0
            991
            INI          K 2
            JR           H DEF
            . . . .
            EXOB

            COB          0
            0
            . . . .
            STXT         1
            10
TEXT 10 "Pompa Litri      Prezzo/L      Totale <10><13>"
      " 1  $1$R0010      $2$R0011      $3$R0012 <10><13>"
      " 2  $1$R0013      $2$R0014      $3$R0015 <10><13>"
            . . . .
            ECOB
    
```

Il risultato è identico a quello che si otterrebbe con il testo formattato come:

```

"1 $%05.1d$R0010 $%04.2d$R0011 $%8.3d$R0012"
"2 $%05.1d$R0013 $%04.2d$R0014 $%8.3d$R0015"
    
```

Risultati:

| Pompa | Litri | Prezzo/l | Totale |
|-------|-------|----------|---------|
| 1 | 13.8 | 0.86 | 11.868 |
| 2 | 158.2 | 0.95 | 150.290 |

Inclusione di altri testi:

La sequenza "\$Lnnnn" 'incLude' un altro testo che viene trattato a tutti gli effetti come se facesse parte del testo originale. Se tale testo incluso contiene una nuova definizione di formato, viene usato quel formato fino alla fine del testo. Ritornando al testo originale, viene ripristinata la definizione di formato di quest'ultimo.

Esempio:

```

COB          0
              0
          ....
          STXT      1      ; Invia Testo (Text) 10
              10
TEXT 10 "$L0100 Velocità motore troppo alta<10><13>"
          ....
          STXT      1      ; Invia Testo (Text) 11
              11
TEXT 11 "$L0100 Pressione olio troppo bassa<10><13>"
          ...
          ECOB
TEXT 100 "ALLARME Motore Diesel:"

```

Risultato:

```

ALLARME Motore Diesel: Velocità motore troppo alta
ALLARME Motore Diesel: Pressione olio troppo bassa

```

Utilizzo di SIMBOLI nei Testi

Con le utility PCD (V. 1.3 e successive) si possono utilizzare dei SIMBOLI all'interno dei Testi. Il valore e opzionalmente il tipo del simbolo vengono inseriti all'interno del Testo. Il simbolo viene scritto al di fuori del segmento di testo ASCII tra virgolette, e deve essere separato dagli altri simboli per mezzo di una virgola. Dopo il simbolo, si possono specificare (come opzioni) la larghezza del campo e il tipo di prefisso.

Formato:

| simbolo [. [[-] [0] larghezza] [t T]] | |
|--|---|
| simbolo | Nome del simbolo. Può essere in effetti una qualsiasi espressione che include un simbolo ad esempio: MotorOn + 100, ... Non sono ammessi simboli con valori in virgola mobile. |
| . | Il punto immediatamente dopo il simbolo indica che esiste un campo larghezza e/o un prefisso. |
| larghezza | È il campo larghezza: numero di cifre o di spazi richiesti dal numero. Se la larghezza inizia con 0, vengono inseriti gli zeri non significativi. |
| t T | Prefisso opzionale tipo 't' o 'T'. Se 't', il valore viene preceduto dal tipo del simbolo in minuscolo (o, f, r, ...); se 'T', il tipo del simbolo sarà in maiuscolo (O, F, R,...). |

Esempi:

```

Flag      EQU      F 123
Output    EQU      O 32
Reg       EQU      R 999
TEXT 0    "$",Flag.04T      ; Il Testo 0 è "$F0123"
TEXT 1    "",Flag          ; Il Testo 1 è "123" (" è necessario)
TEXT 2    "DIAG:",Output.T," ",Reg.T
                        ; Il Testo 2 è "DIAG:O32,R999"
TEXT 3    "55:",Flag.T,"-",Flag+7," :",Output.T,"-",Output+7
                        ; Il Testo 3 è "55:F123-130:O32-39"
TEXT 4    "FLAG Number: *",Flag.-8,"*"
                        ; Il Testo 4 "Numero FLAG: *123  *"
    
```

I simboli possono essere usati anche nei Testi SASI

```

D_FLAGS  EQU      F 500
D_REG    EQU      R 4095

                XOB      16
                SASI     1
                3999
TEXT 3999      "UART:9600,7,E,1;MODE:MC0;"
                "DIAG:",D_FLAG.T," ",D_REG.T,";"
    
```

Questo crea il testo "DIAG:F500,R4095;"

Nota: I simboli (per i testi) e i testi devono essere definiti nello stesso file.

SRXM RICEZIONE SERIALE DATI (Modalità D) (Serial Receive Media - Mode D)

Descrizione Legge gli elementi dal PCD remoto e li copia negli elementi di destinazione nel PCD locale. I trasferimenti possibili sono: da I|O|F a O|F, da R|T|C a R|T|C.
 Il 1° operando è il numero di porta seriale.
 Il 2° operando è il numero di elementi da trasferire.
 Il 3° operando è l'indirizzo più basso degli elementi sorgente nel PCD Remoto.
 Il 4° operando è l'indirizzo più basso degli elementi di destinazione nel PCD Locale.
 Dopo aver eseguito l'istruzione SRXM, il flag TBSY viene posto al valore Alto (H) quindi ritorna al valore basso al termine dell'operazione.

Utilizzo

| | | |
|----------------|-------------------------|--|
| SRXM[X] | canale | ; Numero porta (canale) seriale 0-3 |
| | conteggio | ; Numero di elementi da ricevere 1-16 |
| | sorgente (i) | ; Elemento sorgente I O F, R T C |
| | destinazione (i) | ; Elemento destinazione O F, R T C |

Esempio

```
SRXM      0      ; Legge il contenuto di R 100-115
          16     ; in R 0-15 attraverso la porta seriale 0
R 100
R 0
```

Flag

Il flag **Errore** (E) viene impostato se la porta seriale non è stata inizializzata correttamente o se si esegue una istruzione SRXM quando c'è già una comunicazione in corso.

Vedere anche

STXM, Istruzioni di Comunicazione, Flag Diagnostici

Uso pratico

Gli ingressi 0 .. 15 del PCD Remoto devono essere copiati nelle uscite 32..47 del PCD locale. (I due PCD sono collegati per mezzo di una linea seriale).

PCD Locale:

```
XOB      16
SASI     1
          0

TEXT 0   "UART:9600,7,E,1;MODE:MD0;"
          "DIAG:F1000,R1000;"

EXOB

COB      0
          0

STH      F 1003 ; Se non già occupato (TBSY)
JR       H next

SRXM   1      ; Allora ricevi su porta seriale 1
          16     ; 16 elementi
I 0      ; da I 0 (a 15) del PCD Remoto
O 32     ; su O 32 (a 47) del PCD Locale

next :   ECOB
```

PCD Remoto:

(è sufficiente assegnare la linea seriale)

```
XOB      16
SASI     1
          0

TEXT 0   "UART:9600,7,E,1;MODE:SD0;"
          "DIAG:F1000,R1000;"

EXOB

Dopo l'esecuzione dell'istruzione SASI (in modo
RUN o TRACE), la CPU dal PCD remoto può
essere fermata.
```

SRXM RICEZIONE SERIALE DATI (Modalità MM4) (Serial Receive Media - Mode MM4)

Descrizione Copia il buffer di ricezione (frame ricevuti) in registri consecutivi del PCD. Quando è stato ricevuto un telegramma senza segnalazione di errore RFUL è impostato a 1 e SRXM azzerà il flag.

Il primo operando è il numero del canale seriale.
 Il secondo operando è sempre uguale a 0.
 Il terzo operando è un Registro o un Contatore che conterrà il numero di caratteri ricevuti (dopo l'esecuzione dell'istruzione).
 Il quarto operando è l'indirizzo del primo Registro in cui verranno copiati i caratteri ricevuti.

Ogni singolo carattere ricevuto richiede 8 bit all'interno di un Registro; un Registro può contenere un massimo di 4 caratteri.
 I caratteri sono posizionati nei Registri come segue

| | | | | | |
|--------|----------|----------|----------|----------|--------------------|
| reg 1: | 11111111 | 22222222 | 33333333 | 44444444 | Caratteri da 1 a 4 |
| reg 2: | 55555555 | 66666666 | 77777777 | 88888888 | Caratteri da 5 a 8 |
| | ... | | | | |

Se il numero dei caratteri ricevuti non è un multiplo di 4, il resto del Registro è impostato a 0.

L'indirizzo del partner che ha inviato il telegramma è contenuto nel Registro <repartner> definito nel testo SASI.

| | | | |
|-----------------|-------------|---------------|--|
| Utilizzo | SRXM | canale | ; Numero porta (canale) seriale, 0-3 |
| | | 0 | ; Non utilizzato |
| | | reg 1 | ; Numero caratteri letti R 0-4095 |
| | | reg 2 | ; Indirizzo registro di destinazione R 0-4095 |

dove:

- reg 1 E' il Registro che contiene il numero di caratteri letti (dopo l'esecuzione). Si può utilizzare anche un Contatore.
- reg 2 Indirizzo del primo registro in cui verranno copiate le informazioni (un registro contiene fino a 4 caratteri).

Esempio SRXM 1 ; Trasferisce il telegramma ricevuto sul canale 1
 0 ; nel Registro 20 e successivi
 C 100 ; C100: è il numero di caratteri ricevuti
 R 20

Flag Il flag **Errore (E)** viene impostato se il canale seriale non è stato inizializzato correttamente o se si esegue SRXM quando non è stato ricevuto alcun telegramma

Vedere anche

Per ulteriori informazioni, consultare "Descrizione del Protocollo LAC MM4"

SRXM RICEZIONE SERIALE DATI (Modalità SBUS) (Serial Receive Media - Mode SBUS)

Per maggiori informazioni, consultare il "Manuale S-Bus"

Descrizione Legge gli elementi da una stazione PCD Slave e li copia negli elementi di destinazione del PCD Master. L'indirizzo della stazione Slave viene passato tramite un registro, come definito nel testo dell'istruzione SASI (vedi pag. 8-11). I trasferimenti possibili sono: da I|O|F a O|F, da R|T|C a R|T|C.
 Il 1° operando è il numero di porta seriale (canale).
 Il 2° operando è il numero di elementi da trasferire.
 Il 3° operando è l'indirizzo più basso dell'elemento sorgente nel PCD Slave.
 Il 4° operando è l'indirizzo più basso dell'elemento di destinazione nel PCD Master.
 Dopo aver eseguito l'istruzione SRXM, il flag TBSY viene posto al valore Alto (H) quindi ritorna al valore basso al termine dell'operazione.

L'istruzione SRXM può essere utilizzata solo sul PCD Master

Utilizzo

| | | |
|----------------|-------------------------|--|
| SRXM[X] | canale | ; Numero porta (canale) seriale 0-3 |
| | conteggio | ; Numero di elementi da ricevere |
| | sorgente (i) | ; Indirizzo base dell'elemento sorgente |
| | destinazione (i) | ; Indirizzo base dell'elemento destinazione |

Dove:

| | | | |
|--------------|---------|--|--|
| conteggio | 1..32 | numero di R/T/C da leggere | |
| | 1..128 | numero di I/O/F da leggere | |
| sorgente | 0 | Codice funzione speciale | |
| | R nnnn | Usato per trasferimento di Blocchi di Dati | |
| | I/O/F | 0..8191 | |
| | R | 0..4095 | Indirizzo base degli elementi nel PCD Slave |
| | T/C | 0..1599 | |
| destinazione | DB | 0..7999 | |
| | K | 0..6000 | Codice funzione speciale |
| | I/O/F | 0..8191 | Indirizzo base degli elementi nel PCD Master |
| | R | 0..4095 | |
| T/C | 0..1599 | | |
| DB | 0..7999 | | |

Esempio

```
LD      R 100 ; Registro come definito nel Testo SASI text per
        10   ; il numero della stazione sorgente
SRXM   0    ; Legge attraverso la porta seriale 0
        20   ; 20 elementi
        R 100 ; da R100-119 della stazione 10
        R 0   ; in R0-R19
```

Flag

Il flag **Errorre** (E) viene attivato se la porta seriale non è stata inizializzata correttamente o se si esegue una istruzione SRXM quando c'è già una comunicazione in corso.

Vedere anche

STXM, Istruzioni di Comunicazione, Flag Diagnostici

La seguente tabella illustra quali elementi possono essere copiati dalla stazione sorgente agli appropriati elementi della stazione di destinazione.

| | | PCD Master (destinazione) | | | | | |
|-------------------------|----|---------------------------|---|---|---|---|----|
| | | O | F | R | C | T | DB |
| PCD Slave (sorgente) | I | • | • | | | | |
| | O | • | • | | | | |
| | F | • | • | | | | |
| | R | | | • | • | • | • |
| | C | | | • | • | • | • |
| | T | | | • | • | • | • |
| | K | | | • | | | |
| | DB | | | • | • | • | |

Funzioni speciali

| Codice | Descrizione funzione | Esempi di risultati |
|---------|---|--|
| K 0 ..7 | Lettura Stato della CPU: 0..6: Numero CPU del PCD Slave 7: stato CPU locale | R, C, H, S, D |
| K 1000 | Lettura Orologio | formato uguale all'istruzione RTIME |
| K 2000 | Lettura Registro Display | |
| K 3000 | Lettura Dimensione Data Block | |
| K 5000 | Lettura Tipo Unità in ASCII | " D1", " D2", " D4", ... |
| K 5010 | in decimale | 1, 2, 4, ... |
| K 5100 | Lettura Tipo Modulo in ASCII | " M1_", " M11", " M12", " M14", .. |
| K 5110 | in decimale | 10, 11, 12, 14, 24, ... |
| K 5200 | Lettura Vers. Firmware in ASCII | " \$4C", " 004", " X41", ... |
| K 5210 | in dec. | 5, ... o -1 dec. per ogni '\$', 'X', 'β' |
| K 5300 | Lettura Numero CPU in ASCII | " 0", " 1" |
| K 5310 | in decimale | 0 .. 6 |
| K 6000 | Lettura Numero stazione S-Bus in BROADCAST Operante esclusivamente con comunicazioni punto-punto. | |

Trasferimento di Blocchi di Dati (Data Block)

Quando si opera con i Blocchi di Dati, il formato dell'istruzione SRXM varia leggermente da quello convenzionale. Per indirizzare un elemento di un Blocco di Dati, è necessario specificare sempre il numero del Blocco di Dati in oggetto e la posizione dell'elemento all'interno del Blocco di Dati stesso.

**SRXM Canale
 Conteggio + Posizione
 Sorgente
 Destinazione**

Dove Conteggio + Posizione : Numero Registro.

Questo registro contiene il "Conteggio" ovvero il numero di elementi da trasferire (campo 1...32) e la "Posizione" all'interno del Blocco di Dati in cui inserire o da cui prelevare i dati. Il "Conteggio" è dato dalla parola (word) meno significativa del registro mentre la "Posizione" dalla parola (word) più significativa.

SRXM RICEZIONE SERIALE DATI (PROFIBUS) (Serial Receive Media - PROFIBUS)

Per maggiori informazioni, consultare il "Manuale PROFIBUS"

Descrizione Legge i dati (oggetti) dalla stazione remota specificata e li copia sul PCD Locale.
 Il 1° operando è il numero della porta (canale) seriale.
 Il 2° operando è il sottoindice degli oggetti sorgente e di destinazione.
 Il 3° operando è l'indice dell'oggetto sorgente (stazione remota).
 Il 4° operando è l'indice dell'oggetto di destinazione (stazione locale).

Utilizzo

| | | |
|-------------|---------------------|---|
| SRXM | canale | ; Numero porta (canale) seriale 10-99 |
| | conteggio | ; Sottoindice 0-255 |
| | sorgente | ; Indice oggetto sorgente K 0-16383 |
| | destinazione | ; Indice oggetto di destinazione K 100-499 |

Dove:

| | | |
|--------------|------------|--|
| canale | 10..99 | per PCD4.M445 |
| | 10..19 | per PCD2 con PCD7.F700 |
| conteggio | 0..255 | Sottoindice (0 = nessun sottoindice) |
| sorgente | | Indice oggetto sorgente (stazione remota) |
| | K 0..16383 | |
| destinazione | | Indice oggetto di destinazione (stazione locale) |
| | K 100..499 | PCD4.M445 |
| | K 100..199 | PCD2 con PCD7.F700 |

Esempio **SRXM** **13** ; Legge attraverso il canale 13
 0 ; senza sottoindice
 K 22 : copia l'oggetto 22 dalla stazione remota
 K 150 ; nell'oggetto 150 della stazione locale

Flag Il flag **Errore** (E) viene attivato se il canale non è stato inizializzato correttamente o se si esegue una istruzione SRXM quando c'è già una comunicazione in corso.

Vedere anche STXM

SRXMI RICEZIONE SERIALE DATI INDIRETTA (Mod. SBUS) (Serial Receive Media Indirect - Mode SBUS)

Per maggiori informazioni, consultare il "Manuale S-Bus" (Rif. 26/739)

Descrizione Questa istruzione opera nello stesso modo dell'istruzione SRXM; la differenza consiste nel fatto che l'istruzione SRXMI opera in modo indiretto. Questo significa che il numero dei dati riguardanti la sorgente e la destinazione viene fornito dal contenuto di un registro. SRXMI è disponibile solo per il trasferimento di dati. Le opzioni di trasferimento riguardanti l'Orologio hardware, il Registro Display e così via, non sono permesse.

L'istruzione SRXMI non opera in modalità indicizzata e parametrizzata.

| | |
|-----------------|--|
| Utilizzo | <p>SRXMI canale ; Numero porta (canale) seriale conteggio ; Conteggio o Conteggio + Posizione sorgente ; Tipo Sorgente e numero Registro destinazione ; Tipo Destinazione e numero Registro</p> |
|-----------------|--|

Canale

Questo parametro viene usato per specificare il numero della porta (canale) seriale (campo: 0..3).

Conteggio o Conteggio + Posizione

Questo parametro indica un numero di Registro. Quest'ultimo contiene il "Conteggio" per i dati standard oppure il "Conteggio" e la "Posizione" per i Blocchi di Dati. Per quanto riguarda questi ultimi, il "Conteggio" è indicato dalla parola (word) più significativa del registro e la "Posizione" dalla parola (word) meno significativa del registro stesso. L'inizializzazione del registro può essere agevolmente eseguita mediante le istruzioni LDL e LDH.

Tipo Sorgente e numero Registro

Tipo Destinazione e numero Registro

Questi parametri specificano la "Sorgente" e la "Destinazione" del trasferimento. Ciascun parametro è composto da un carattere indicante il tipo di dato (I/O/F/R/T/C/DB) e da un numero di Registro (0...4095). La sorgente e la destinazione devono rispettare la validità sorgente-destinazione descritta nella tabella dedicata alle istruzioni SRXM/STXM.

Esempio SRXMI 3 ; Porta (canale) seriale 3
 R 100 ; Numero di elementi in R 100
 O 101 ; Sorgente: uscite con indirizzo base in R 101
 F 102 ; Destinazione: flag con indirizzo base in R 102

Flag Il flag **Errore** (E) viene attivato se il canale non è stato inizializzato correttamente o se si esegue una istruzione SRXMI quando c'è già una comunicazione in corso.

Vedere anche SRXM, Istruzioni di Comunicazione, Flag Diagnostici.

SRXMI RICEZIONE SERIALE DATI INDIRETTA (PROFIBUS) (Serial Receive Media Indirect - PROFIBUS)

Per maggiori informazioni, consultare il "Manuale PROFIBUS"

Descrizione Legge i dati (oggetti) in modo indiretto dalla stazione remota specificata e li copia sul PCD Locale. E' possibile selezionare l'indirizzamento diretto o indiretto degli operandi legati al canale.

L'istruzione SRXMI non opera in modalità indicizzata e parametrizzata.

Utilizzo

| | | |
|--------------|---------------------|--|
| SRXMI | canale | ; Numero canale 10-99, R 0-4095 |
| | conteggio | ; Sottoindice R 0-4095 |
| | sorgente | ; indice oggetto sorgente K [R 0-4095] |
| | destinazione | ; Indice oggetto di destinazione K [R 0-4095] |

Dove:

| | | |
|--------------|--|--|
| canale | 10..99 | per PCD4.M445 |
| | 10..19 | per PCD2 con PCD7.F700 |
| conteggio | R 0..4095 | per indirizzamento indiretto |
| | R 0..4095 | Registro contenente il sottoindice (0 = nessun sottoindice) |
| sorgente | Registro contenente l'indice dell'oggetto sorgente (st.remota) | |
| | K [R 0..4095] | 0..16383 |
| destinazione | Registro contenente l'indice dell'oggetto di destinazione | |
| | K [R 0..4095] | 100..499 PCD4.M445 100..199 PCD2 con PCD7.F700 |

Esempio

SRXMI R 50 ; Numero canale seriale in R 50
R 51 ; sottoindice in R 51
K 52 ; indice oggetto sorgente in R 52
K 53 ; indice oggetto di destinazione in R 53

Flag

Il flag **Errore (E)** viene attivato se il canale non è stato inizializzato correttamente o se si esegue una istruzione SRXMI quando c'è già una comunicazione in corso.

Vedere anche SRXM

STXM TRASMISSIONE SERIALE DATI (Modalità D) (Serial Transmit Media - Mode D)

Descrizione Trasmette gli elementi dal PCD Locale verso elementi del PCD Remoto.
 I trasferimenti possibili sono: da I|O|F a O|F, da R|T|C a R|T|C.
 Il 1° operando è il numero di porta seriale (canale).
 Il 2° operando è il numero di elementi da trasferire.
 Il 3° operando è l'indirizzo più basso dell'elemento sorgente nel PCD Locale.
 Il 4° operando è l'indirizzo più basso dell'elemento di destinazione nel PCD Remoto.
 Durante l'esecuzione STXM, il flag TBSY viene posto al valore Alto (H) quindi ritorna al valore basso al termine dell'operazione.

| | | | |
|-----------------|----------------|-------------------------|---|
| Utilizzo | STXM[X] | canale | ; Numero porta (canale) seriale 0-3 |
| | | conteggio | ; Numero di elementi da trasmettere 1-16 |
| | | sorgente (i) | ; Elemento sorgente I O F, R T C |
| | | destinazione (i) | ; Elemento di destinazione O F, R T C |

Esempio STXM 0 ; Trasmette il contenuto di R 100-115
 16 ; in R 0-15 attraverso la porta seriale 0
 R 100
 R 0

Flags Il flag **Errore** (E) viene attivato se il canale non è stato inizializzato correttamente o se si esegue una istruzione STXM quando c'è già una comunicazione in corso

Vedere anche SRXM, Istruzioni di Comunicazione, Flag Diagnostici

Uso pratico Gli Ingressi 0..15 del PCD Locale devono essere copiati nelle Uscite 32..47 del PCD Remoto.
 PCD Remoto: Deve essere assegnata solo la porta (canale) seriale (vedere SRXM)
 PCD Locale:

```

XOB            16
SASI           1
                 15
TEXT 15      "UART:9600,7,E,1;MODE:MD0;DIAG:F1000,R1000;"
EXOB

COB            0
                 0
STH            F 1003 ; Se non è già occupato a comunicare (TBSY)
JR             H NEXT
STXM         1      ; Allora trasferisci attraverso la porta seriale 1
                 16      ;    16 Elementi
                 I 0      ;    dall'Ingresso 0 (fino a 15) del PCD Locale
                 O 32     ;    nell'Uscita 32 (fino a 47) del PCD Remoto
NEXT:         ECOB
    
```

STXM TRASMISSIONE SERIALE DATI (Modalità MM4) (Serial Transmit Media - Mode MM4)

Descrizione Trasferisce i registri attraverso la rete LAC/LAC 2 utilizzando il protocollo MM4. Il trasferimento può avvenire attraverso la rete LAC/LAC2 oppure in modalità punto-punto.

Il 1° operando indica il numero di porta seriale (canale).

Il 2° operando definisce la funzione di trasferimento.

Il 3° operando indica un Registro o un Contatore contenente il numero di caratteri da trasferire.

Il 4° operando indica l'indirizzo del primo Registro contenente i caratteri da trasmettere.

Un Registro può contenere un massimo di 4 caratteri e ogni carattere richiede 8 bit. Il carattere deve essere posizionato all'interno del Registro come segue:

reg 1: 11111111 22222222 33333333 44444444 Caratteri da 1 a 4

reg 2: 55555555 66666666 77777777 88888888 Caratteri da 5 a 8

...

L'indirizzo del partner è contenuto nel Registro <trpartner> definito nel testo SASI.

Dopo l'esecuzione dell'istruzione STXM, il flag XBSY è impostato al valore Alto (H); al termine dell'operazione (risposta positiva alla ricezione del telegramma da parte del partner) XBSY viene riportato al valore Basso (L).

Utilizzo

| | | |
|-------------|-----------------|--|
| STXM | canale | ; Numero porta (canale) seriale 0-3 |
| | funzione | ; Funzione da eseguire 0-4 |
| | reg 1 | ; Numero di caratteri da trasmettere R 0-4095 |
| | reg 2 | ; Indirizzo registro sorgente R 0-4095 |

dove:

| | |
|----------|---|
| funzione | Funzione da eseguire |
| 0 / 2 | Trasmissione dati |
| 4 | Diffusione |
| reg 1 | Registro contenente il numero di caratteri da trasmettere Può essere utilizzato anche un contatore. |
| reg 2 | Indirizzo del primo registro da cui devono essere trasferite le informazioni (un registro contiene fino a 4 caratteri). |

Esempio

STXM 1 ;Trasmissione sul canale 1
 0 ; indica una Trasmissione
C 100 ; numero di caratteri da trasmettere
R 20 ; primo Registro contenente i dati

Flag

Il flag **Errore** (E) viene impostato se l'inizializzazione della porta seriale non è stata eseguita correttamente oppure se si esegue STXM quando è già in corso una comunicazione

Vedere anche

Per ulteriori informazioni consultare la "Descrizione del Protocollo LAC MM4"

STXM TRASMISSIONE SERIALE DATI (Modalità SBUS) (Serial Transmit Media - Mode SBUS)

Per maggiori informazioni, consultare il "Manuale S-Bus"

Descrizione Trasmette gli elementi da una stazione PCD Master e li copia negli elementi di destinazione del PCD Slave. L'indirizzo della stazione Slave viene passato tramite un registro, come definito nel testo dell'istruzione SASI (vedere pag. 8-12).
I trasferimenti possibili sono: da I|O|F a O|F, da R|T|C a R|T|C.
Il 1° operando è il numero di porta seriale (canale).
Il 2° operando è il numero di elementi da trasferire.
Il 3° operando è l'indirizzo più basso dell'elemento sorgente nel PCD Master.
Il 4° operando è l'indirizzo più basso dell'elemento di destinazione nel PCD Slave.
Dopo aver eseguito l'istruzione STXM, il flag TBSY viene posto al valore Alto (H) quindi ritorna al valore basso al termine dell'operazione.

L'istruzione STXM può essere utilizzata solo sul PCD Master

| | | | |
|-----------------|----------------|---|--|
| Utilizzo | STXM[X] | canale conteggio sorgente (i) destinazione (i) | ; Numero porta (canale) seriale, 0-3 ; Numero di elementi da trasmettere ; Indirizzo base elemento sorgente ; Indirizzo base elemento di destinazione |
|-----------------|----------------|---|--|

Dove:

| | | | |
|--------------|--------|----------------------------------|--|
| canale | 0..3 | Interfaccia da utilizzare | |
| conteggio | 1..32 | Numero di R/T/C da leggere | |
| | 1..128 | Numero di I/O/F da leggere | |
| | 0 | Codice funzione speciale | |
| sorgente | I/O/F | 0..8191 | |
| | R | 0..4095 | Indirizzo base degli elementi nel PCD Master |
| | T/C | 0..1599 | |
| | DB | 0..7999 | |
| destinazione | K | 4000 | Funzione speciale |
| | I/O/F | 0..8191 | |
| | R | 0..4095 | Indirizzo base degli elementi nel PCD Slave |
| | T/C | 0..1599 | |
| | DB | 0..7999 | |
| K | 1000 | Scrittura orologio nel PCD Slave | |
| | K | 17, 18, 19 | Funzione speciale |

Esempio LD R 100 ; Registro come definito nel testo SASI per
22 ; il numero della stazione di destinazione
STXM 0 ; Trasmette attraverso il canale (porta) seriale 0
100 ; 100 elementi
F 100 ; da F100-199
O 32 ; a O32-131 della stazione 22

Flag Il flag **Errore** (E) viene impostato se l'inizializzazione della porta seriale non è stata eseguita correttamente oppure se si esegue STXM quando è già in corso una comunicazione

Vedere anche SRXM, Istruzioni di Comunicazione, Flag Diagnostici.
La seguente tabella illustra quali elementi possono essere copiati dalla stazione sorgente agli appropriati elementi della stazione di destinazione.

| | | PCD Slave (destinazione) | | | | | | |
|--------------------------|----|--------------------------|---|---|---|---|----|-------|
| | | O | F | R | C | T | DB | Clock |
| PCD Master (sorgente) | I | • | • | | | | | |
| | O | • | • | | | | | |
| | F | • | • | | | | | |
| | R | | | • | • | • | • | • |
| | C | | | • | • | • | • | |
| | T | | | • | • | • | • | |
| | DB | | | • | • | • | | |

In caso di scrittura dell'orologio, vengono trasmessi due registri. Per il formato dati dei registri in oggetto, vedere l'istruzione WTIME.

Funzioni speciali

E' possibile comandare l'esecuzione di un XOB in una stazione Slave utilizzando l'istruzione STXM con i seguenti argomenti:

STXM 0 . . 3 ; Numero porta (canale) seriale
 0 ; (deve essere 0)
 K 4000 ; Usato per indicare l'interrupt XOB
 K 17 | 18 | 19 ; numero dell'XOB da eseguire

Questa istruzione può essere utilizzata anche in modalità Broadcast; in questo modo è possibile effettuare la sincronizzazione di eventi.

Trasferimento di Blocchi di Dati - Data Block (Scrittura - Write)

Quando si opera con i Blocchi di Dati, il formato dell'istruzione STXM varia leggermente da quello convenzionale. Per indirizzare un elemento di un Blocco di Dati, è necessario specificare sempre il numero del Blocco di Dati in oggetto e la posizione dell'elemento all'interno del Blocco di Dati stesso.

STXM Canale
Conteggio + Posizione
Sorgente
Destinazione

Dove Conteggio + Posizione : Numero Registro.

Questo registro contiene il "Conteggio" ovvero il numero di elementi da trasferire (campo 1...32) e la "Posizione" all'interno del Blocco di Dati in cui inserire o da cui prelevare i dati. Il "Conteggio" è dato dalla parola (word) meno significativa del registro mentre la "Posizione" dalla parola (word) più significativa

**STXM TRASMISSIONE SERIALE DATI (PROFIBUS)
(Serial Transmit Media - PROFIBUS)**

Per maggiori informazioni, leggere il "Manuale PROFIBUS"

Description Trasmette i dati (oggetti) dal PCD Locale e li copia sulla stazione remota specificata.
 Il 1° operando è il numero della porta (canale) seriale.
 Il 2° operando è il sottoindice degli oggetti sorgente e di destinazione.
 Il 3° operando è l'indice dell'oggetto sorgente (stazione locale).
 Il 4° operando è l'indice dell'oggetto di destinazione (stazione remota).

| | | | |
|-----------------|-------------|---------------------|---|
| Utilizzo | STXM | canale | ; Numero porta (canale) seriale, 10-99 |
| | | conteggio | ; Sottoindice 0-255 |
| | | sorgente | ; Indice oggetto sorgente K 100-499 |
| | | destinazione | ; Indice oggetto di destinazione K 0-16383 |

Dove:

| | | |
|-------------|------------|--|
| canale | 10..99 | per PCD4.M445 |
| | 10..19 | per PCD2 con PCD7.F700 |
| conteggio | 0..255 | Sottoindice (0 = nessun sottoindice) |
| sorgente | | Indirizzo oggetto sorgente (stazione locale) |
| | K 100..499 | PCD4.M445 |
| | K 100..199 | PCD2 con PCD7.F700 |
| destination | | Indirizzo oggetto di destinazione (st. remota) |
| | K 0..16383 | |

Esempio **STXM** **11** ; Trasmette attraverso il canale 11
 3 ; il sottoindice (elemento) 3
 K 122 ; copia l'oggetto 122 dalla stazione locale
 K 150 ; nell'oggetto 150 della stazione remota

Flag Il flag **Errore (E)** viene impostato se l'inizializzazione della porta seriale non è stata eseguita correttamente oppure se si esegue STXM quando è già in corso una comunicazione.

Vedere anche SRXM

STXMI TRASMISSIONE SERIALE DATI INDIRETTA (SBUS) (Serial Transmit Media Indirect - Mode SBUS)

Per maggiori informazioni, consultare il "Manuale S-Bus"

Descrizione Questa istruzione opera nello stesso modo dell'istruzione STXM; la differenza consiste nel fatto che l'istruzione STXMI opera in modo indiretto. Questo significa che il numero dei dati riguardanti la sorgente e la destinazione viene fornito dal contenuto di un registro. STXMI è disponibile solo per il trasferimento di dati. Le opzioni di trasferimento riguardanti l'Orologio hardware, il Registro Display e così via, non sono permesse.

L'istruzione STXMI non opera in modalità indicizzata e parametrizzata.

Utilizzo

| | | |
|--------------|---------------------|--|
| STXMI | canale | ; Identificatore Canale |
| | conteggio | ; Conteggio o Conteggio + Posizione |
| | sorgente | ; Tipo Sorgente e numero Registro |
| | destinazione | ; Tipo Destinazione e numero Registro |

Canale

Questo parametro viene usato per specificare il numero della porta (canale) seriale (campo: 0..3).

Conteggio o Conteggio + Posizione

Questo parametro indica un numero di Registro. Quest'ultimo contiene il "Conteggio" per i dati standard oppure il "Conteggio" e la "Posizione" per i Blocchi di Dati. Per quanto riguarda questi ultimi, il "Conteggio" è indicato dalla parola (word) più significativa del registro e la "Posizione" dalla parola (word) meno significativa del registro stesso. L'inizializzazione del registro può essere agevolmente eseguita mediante le istruzioni LDL e LDH.

Tipo Sorgente e numero Registro

Tipo Destinazione e numero Registro

Questi parametri specificano la "Sorgente" e la "Destinazione" del trasferimento. Ciascun parametro è composto da un carattere indicante il tipo di dato (I/O/F/R/T/C/DB) e da un numero di Registro (0...4095). La sorgente e la destinazione devono rispettare la validità sorgente-destinazione descritta nella tabella dedicata all'istruzione STXM.

Esempio

```
STXMI      1
           R  100
           DB 101
           R  102
```

Flag

Il flag **Errorre** (E) viene impostato se l'inizializzazione della porta seriale non è stata eseguita correttamente oppure se si esegue STXMI quando è già in corso una comunicazione.

Vedere anche

STXM, Istruzioni di Comunicazione, Flag Diagnostici

STXMI TRASMISSIONE SERIALE DATI INDIRETTA (PROFIBUS) (Serial Transmit Media Indirect - PROFIBUS)

Per maggiori informazioni, consultare il "Manuale PROFIBUS"

Descrizione Trasmette i dati (oggetti) in modo indiretto dal PCD Locale e li copia sulla stazione remota specificata. E' possibile selezionare l'indirizzamento diretto o indiretto degli operandi legati al canale.

L'istruzione STXMI non opera in modalità indicizzata e parametrizzata.

Utilizzo

| | | |
|--------------|---------------------------|--|
| STXMI | canale | ; Numero canale seriale 10-99, R 0-4095 |
| | conteggio sorgente | ; Sottoindice R 0-4095 |
| | destinazione | ; Indice oggetto sorgente K [R 0-4095] |
| | | ; Indice oggetto di destinazione K [R 0-4095] |

Dove:

| | | |
|--------------|--|---|
| canale | 10..99 | per PCD4.M445 |
| | 10..19 | per PCD2 con PCD7.F700 |
| conteggio | R 0..4095 | per indirizzamento indiretto |
| | R 0..4095 | Registro contenente il sottoindice (0 = nessun sottoindice) |
| sorgente | Registro contenente l'indice dell'oggetto sorgente (st. locale) | |
| | K [R 0..4095] | 100..499 PCD4.M445 100..199 PCD2 con PCD7.F700 |
| destinazione | Registro contenente l'indice dell'oggetto destinaz. (st. remota) | |
| | K [R 0..4095] | 0..16383 |

Esempio **STXMI R 100** ; Numero canale in R 100
R 110 ; sottoindice in R 110
K 200 ; indice oggetto sorgente in R 200
K 300 ; indice oggetto di destinazione in R 300

Flag Il flag **Errore (E)** viene impostato se l'inizializzazione della porta seriale non è stata eseguita correttamente oppure se si esegue STXMI quando è già in corso una comunicazione.

Vedere anche STXM

SICL **SEGNALI DI CONTROLLO DEL CANALE SERIALE IN INGRESSO (Serial Input Control Line)**

Descrizione Legge un segnale di controllo dalla porta seriale specificata nel primo operando e ne memorizza lo stato nell'ACCU.

Il secondo operando è il segnale da leggere:

0 = CTS Clear To Send (Pronto a Trasmettere)
 1 = DSR Data Set Ready (Modem Pronto)
 2 = DCD Data Carrier Detect (Rilevazione di portante)

L'istruzione SICL è sempre ammessa per la porta seriale 0 (PGU) delle unità PCD1, PCD2, PCD4 e PCD6.M540 (indipendentemente dal fatto che la porta in oggetto sia assegnata o configurata).

Per qualsiasi altra porta seriale delle unità PCD1, PCD2, PCD4 o PCD6.M540, l'istruzione SICL è ammessa solo se eseguita per una porta configurata come PGU S-Bus. In caso contrario, l'istruzione SICL è ammessa solo dopo l'esecuzione di un'istruzione SASI.

Utilizzo

| | | |
|-------------|---------------------------|---|
| SICL | canale segnale | ; Numero porta (canale) seriale 0-3 ; Numero segnale CTS DSR DCD (0 1 2) |
|-------------|---------------------------|---|

Esempio

SICL 0 ; Se il segnale DSR della porta (canale) 0 è Alto (H)
 1
 CPB H 25 ; Allora chiama il PB 25

Flag

L'ACCU viene impostato in base allo stato dal segnale di controllo in ingresso. Il flag **Errore** (E) viene impostato se la porta seriale non esiste o non è stata inizializzata correttamente.

Vedere anche

SOCL, Istruzioni di Comunicazione

SOCL **SEGNALI DI CONTROLLO DEL CANALE SERIALE IN USCITA (Serial Output Control Line)**

Descrizione Imposta il segnale di controllo selezionato relativo alla porta seriale specificata nel primo operando in base allo stato dell'ACCU (H o L).

Il secondo operando rappresenta il segnale da gestire:

- 0 = RTS Request To Send (Richiesta di Trasmissione)
- 1 = DTR Data Terminal Ready (Terminale Pronto)
- 2 = Funzioni speciali

L'istruzione SOCL è sempre ammessa per la porta seriale 0 (PGU) delle unità PCD1, PCD2, PCD4 e PCD6.M540 (indipendentemente dal fatto che la porta in oggetto sia assegnata o configurata).
 Per qualsiasi altra porta seriale delle unità PCD1, PCD2, PCD4 o PCD6.M540, l'istruzione SOCL è ammessa solo se eseguita per una porta configurata come PGU S-Bus. In caso contrario, l'istruzione SOCL è ammessa solo dopo l'esecuzione di un'istruzione SASI.

| | |
|-----------------|---|
| Utilizzo | SOCL canale ; Numero porta (canale) seriale 0-3 signal ; Numero segnale RTS DTR (0 1 2) |
|-----------------|---|

Esempio SOCL 0 ; Imposta il segnale DTR della porta (canale) 0 in
 1 ; funzione dello stato dell'ACCU

Flag Il flag **Errorre** (E) viene impostato se la porta seriale non esiste o non è stata inizializzata correttamente.

Vedere anche SICL, Istruzioni di Comunicazione

Funzioni speciali:

Porta 0 su un'unità PCD2

Un'istruzione SASI per modalità SM1/SS1 presente all'interno del programma utente configurerà la porta 0 come RS 485. Se l'utente desidera utilizzare la porta 0 come interfaccia RS 232, è necessario eseguire le seguenti istruzioni dopo l'istruzione SASI:

```

ACC      L
SOCL     0
          2
    
```

Commutazione da RS485 a RS422

L'interfaccia seriale RS422/RS485 delle unità PCD4.C130, PCD4.C340, PCD7.F110/F150, PCD7.F520/530 commuta automaticamente in RS485 in base alla modalità assegnata:

| Modalità | Tipo |
|-----------------------|-------|
| MC0 .. MC3, MD0 / SD0 | RS422 |
| MC4, S-Bus | RS485 |

A volte è necessario forzare il PCD affinché utilizzi l'S-Bus in modalità RS422; in questo caso, è necessario eseguire le seguenti istruzioni dopo l'istruzione SASI:

```
ACC      L
SOCL     Nr._Porta
          2
```

E' anche possibile forzare la modalità RS485 con MC0..MC3 o MD0/SD0 eseguendo le seguenti istruzioni:

```
ACC      H
SOCL     Nr._Porta
          2
```

SCON CONNESSIONE SERIALE ALLA LAN 1 (Serial Connect to LAN 1)

Descrizione Apre o chiude una connessione virtuale verso altre stazioni collegate alla rete LAN1 SAIA.
 Il 1° operando è il numero della porta seriale (canale).
 Il 2° operando è il numero della stazione (1-250).
 Il 3° operando rappresenta lo stato della connessione (0 = Chiusa, 1 = Aperta).

Lo stato della connessione viene riportato nel Registro definito con l'istruzione SASI attraverso l'assegnazione MODE (Esempio: "MODE:SD0,R4000;").

Stato della connessione riportato nel registro:

| Valore | Descrizione |
|----------|--|
| 0 | Scollegato |
| 1 | Collegato |
| 2 | In coda |
| 3 | Destinazione occupata |
| 4 | Destinazione sconosciuta |
| 6 | PLC remoto non collegato |
| 10..2500 | Se la connessione è stata effettuata da un PLC remoto, il registro contiene il numero di tale PLC moltiplicato per 10. |

Per ulteriori informazioni consultare il manuale LAN 1 SAIA.

| | | | |
|-----------------|-------------|-----------------|---|
| Utilizzo | SCON | canale | ; Numero porta (canale) seriale, 0-3 |
| | | stazione | ; Numero stazione LAN 1, 1-250 |
| | | stato | ; Stato della connessione (0 = chiusa, 1 = aperta) |

Esempio SCON 0 ; Connessione aperta verso la stazione 100 sulla
 100 ; porta (canale) seriale 0
 1

Flag Il flag **Errore (E)** viene impostato se la porta seriale non esiste o non è stata inizializzata correttamente

Vedere anche SASI, Istruzioni di comunicazione

SCON APRI CANALE DI COMUNICAZIONE (PROFIBUS) (Open Communication Channel - PROFIBUS)

Per maggiori dettagli, consultare il "Manuale PROFIBUS"

Descrizione Apre o chiude un canale virtuale PROFIBUS.
Prima di avviare lo scambio delle informazioni, è necessario inizializzare (aprire) il canale virtuale di comunicazione per mezzo dell'istruzione SCON.
Il 1° operando è il numero del canale.
Il 2° operando è sempre zero (0 - non utilizzato).
Il 3° operando indica lo stato della connessione (0 = Chiusa, 1 = Aperta).

| | | | | |
|-----------------|------|--------|---|--|
| Utilizzo | SCON | canale | ; | Numero canale PROFIBUS, 10-99 |
| | | 0 | ; | |
| | | stato | ; | Stato della connessione (0 = chiusa, 1 = aperta) |

Esempio SCON 10 ; Apre il canale 10
0 ;
1

Flag Il flag **Errore (E)** viene impostato se il canale in oggetto non esiste.

SCONI CONNESSIONE SERIALE INDIRETTA ALLA LAN 1 (Serial Connect to LAN 1 Indirect)

Descrizione Apre o chiude una connessione virtuale verso altre stazioni collegate alla rete LAN1 SAIA in modo indiretto.
 Il 1° operando è il numero della porta seriale (canale) oppure un Registro contenente il numero della porta seriale.
 Il 2° operando è un Registro contenente il numero della stazione(1-250).
 Il 3° operando è un Registro contenente lo stato della connessione (0 = Chiusa, 1 = Aperta).

L'istruzione SCONI non opera in modalità indicizzata e parametrizzata.

Lo stato della connessione viene riportato nel Registro definito con l'istruzione SASI attraverso l'assegnazione MODE (Esempio: "MODE:SD0,R4000;").

Stato della connessione riportato nel registro:

| Valore | Descrizione |
|----------|--|
| 0 | Scollegato |
| 1 | Collegato |
| 2 | In coda |
| 3 | Destinazione occupata |
| 4 | Destinazione sconosciuta |
| 6 | PLC remoto non collegato |
| 10..2500 | Se la connessione è stata effettuata da un PLC remoto, il registro contiene il numero di tale PLC moltiplicato per 10. |

Per ulteriori informazioni consultare il manuale LAN 1 SAIA.

| | | | |
|-----------------|--------------|-----------------|---|
| Utilizzo | SCONI | canale | ; Numero porta (canale) seriale 0-3 o R 0-4095 |
| | | stazione | ; Numero stazione LAN 1 R 0-4095 |
| | | stato | ; Stato della connessione R 0-4095 |

Esempio SCONI 0 ; Apre la porta (canale) seriale 0
 R 100 ; numero stazione LAN 1 in R 100
 R 101 ; stato della connessione in R 101

Flag Il flag **Errore (E)** viene impostato se la porta seriale non esiste o non è stata inizializzata correttamente

Vedere anche SCON, SASI, Istruzioni di Comunicazione

SCONI APRI CANALE DI COMUNICAZIONE INDIRECTO (PROFIBUS) (Open Communication Channel Indirect - PROFIBUS)

Per maggiori informazioni, consultare il "Manuale PROFIBUS"

Descrizione Apre o chiude un canale virtuale PROFIBUS in modo indiretto. Prima di avviare lo scambio delle informazioni, è necessario inizializzare (aprire) il canale virtuale di comunicazione per mezzo dell'istruzione SCON. Il 1° operando è il numero del canale o un Registro contenente il numero del canale. Il 2° operando è un Registro che deve sempre essere a zero. Il 3° operando è un registro contenente lo stato della connessione (0 = Chiusa, 1 = Aperta).

L'istruzione SCONI non opera in modalità indicizzata e parametrizzata.

| | | | |
|-----------------|-------|-----------------------------|---|
| Utilizzo | SCONI | canale stazione stato | ; Num. canale PROFIBUS, 10-99 o R 0- 4095 ; R 0- 4095 (sempre a 0) ; Stato della connessione, R 0- 4095 |
|-----------------|-------|-----------------------------|---|

Esempio SCONI R 10 ; Apre il canale PROFIBUS indicato in R 10
R 11 ; Numero stazione in R 11 (deve essere 0)
R 12 ; Stato della connessione in R 12

Flags Il flag **Errore (E)** viene impostato se il canale in oggetto non esiste.

Vedere anche SCON

TABELLA DEI CARATTERI ASCII

| Primo Bit | Hex | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|------------|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Ultimo Bit | Binary | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| 0 | 0000 | NUL | | SP | 0 | @ | P | ` | p | Ç | É | á | ⋮ |  |  |  |  |
| 1 | 0001 | | XON | ! | 1 | A | Q | a | q | ü | æ | í | ⋮ |  |  |  |  |
| 2 | 0010 | STX | DC2 | " | 2 | B | R | b | r | é | Æ | ó | ⋮ |  |  |  |  |
| 3 | 0011 | ♥ | | # | 3 | C | S | c | s | â | ô | ú |  |  |  |  |  |
| 4 | 0100 | ♦ | DC4 | \$ | 4 | D | T | d | t | ä | ö | ñ |  |  |  |  |  |
| 5 | 0101 | ♣ |  | % | 5 | E | U | e | u | à | ò | Ñ |  |  |  |  |  |
| 6 | 0110 | ♠ | ACK | & | 6 | F | V | f | v | å | ù | ä |  |  |  |  |  |
| 7 | 0111 | BEL | | ' | 7 | G | W | g | w | ç | û | å |  |  |  |  |  |
| 8 | 1000 | BS | CAN | (| 8 | H | X | h | x | ê | ÿ | ì |  |  |  |  |  |
| 9 | 1001 | HT | |) | 9 | I | Y | i | y | ë | ÿ | í |  |  |  |  |  |
| A | 1010 | LF | | * | : | J | Z | j | z | è | Ü |  |  |  |  |  |  |
| B | 1011 | VT | ESC | + | ; | K | [| k | { | ì |  | ½ |  |  |  |  |  |
| C | 1100 | FF | | , | < | L | \ | l | | í |  | ¼ |  |  |  |  |  |
| D | 1101 | CR | | - | = | M |] | m | } | î |  |  |  |  |  |  |  |
| E | 1110 | SO | | . | > | N | ^ | n | ~ | ï |  |  |  |  |  |  |  |
| F | 1111 | SI | | / | ? | O | _ | o |  |  |  |  |  |  |  |  |  |

I caratteri 0 ... 127 (decimale) sono standard
I caratteri 128 ... 255 sono per PC IBM

9. Istruzioni LAN 2

La LAN2 SAIA è una Rete Locale basata sul principio del "Token Passing" che permette di interconnettere fino a 255 stazioni.

Per ogni stazione è richiesto l'utilizzo di un modulo LAN2 (PCD6.T1.. o PCD4.M340).

Attraverso la LAN2 è possibile trasmettere o ricevere gli stati di qualsiasi Ingresso, Uscita o Flag oppure di qualsiasi CPU, e i valori presenti in qualsiasi Registro, Temporizzatore o Contatore.

I comandi LAN2 vengono definiti in specifici testi di comando. É possibile effettuare il trasferimento da un qualsiasi elemento a livello bit verso un altro elemento a livello bit (ad es.: da Ingressi a Flag), oppure da Temporizzatore/Contatore a Temporizzatore /Contatore, o da Registro a Registro

IMPORTANTE:

Le istruzioni e le funzionalità descritte nel presente manuale sono valide a partire dalla versione 005 (e successive) di LAN2.

Istruzioni LAN 2:

| | |
|-------------|--|
| LRXD | Receive data via LAN2 (Ricezione Dati via LAN2) |
| LTXD | Transmit data via LAN2 (Trasmissione Dati via LAN2) |
| LRXS | Receive status via LAN2 (Ricezione Stato via LAN2) |
| LTXS | Transmit status via LAN2 (Trasmissione Stato via LAN2) |

LRXD RICEZIONE DATI VIA LAN2 (Receive Data via LAN2)

Descrizione Ricezione dati da un PCD remoto attraverso la rete LAN2.
 Il 1° operando rappresenta la priorità (0 = Alta, 1 = Bassa). Per avvantaggiarsi di trasferimenti ad elevata priorità, è necessario che la lunghezza massima dei dati da trasferire non superi i 32 byte.
 Il 2° operando indica il numero di un Testo contenente l'indirizzo del partner remoto, gli elementi di tale partner che devono essere letti e dove tali elementi devono essere copiati (stazione locale).
 Il 3° operando indica l'indirizzo base di 10 Flag Diagnostici (o Uscite).
 Il primo elemento diagnostico è il flag EXEC. Inizialmente questo viene impostato a livello basso dall'istruzione LRXD, e rimane a tale livello per tutte le esecuzioni successive della stessa istruzione LRXD, finchè il trasferimento non viene completato. A questo punto, il flag EXEC viene posto a livello Alto. Se l'istruzione LRXD viene nuovamente eseguita, il trasferimento dei dati viene ripetuto. Lo stato del flag EXEC viene modificato solo quando viene eseguita l'istruzione LRXD.

Utilizzo

| | | |
|----------------|------------------|---|
| LRXD[X] | priorità | ; 0 = alta, 1 = bassa |
| | testo (i) | ; Numero del Testo contenente il comando, 0-7999 |
| | diag | ; Indirizzo base flag diagnostici F O 0-8182 |

Esempio

LRXD 1 ; Trasferimento a bassa priorità
 900 ; il Testo 900 contiene il comando
 F 200 ; I flag Diagnostici sono F 200-209

Flag

Invariati.

Vedere anche

LTXD, LRXS, LTXS, Flag Diagnostici LAN2, Testi di Comando LAN2

Uso pratico

Dopo la commutazione dell'ingresso I 1 sul PCD locale, gli ingressi 0-7 della stazione PCD remota numero 3 vengono letti e trasferiti nel PCD locale, sulle Uscite 32-39.

```

COB      5
          0
STH      I 1      ; Se l'ingresso 1 diventa Alto (H)
DYN      F 1
ORL      F 100   ;      Flag EXEC
CPB      H 50   ;      Allora chiama il PB 50
ECOB
PB       50
LRXD    1      ; Priorità
          15     ; Numero Testo.
          F 100  ; Flag EXEC
    
```

```

$LAN
TEXT 15 "3:I0-7:O32-39"
$ENDLAN
EPB
    
```

LTXD TRASMISSIONE DATI VIA LAN2 (Transmit Data via LAN2)

Descrizione Trasmissione dati a un PCD remoto attraverso la rete LAN2.
 Il 1° operando rappresenta la priorità (0 = Alta, 1 = Bassa). Per avvantaggiarsi di trasferimenti ad elevata priorità, è necessario che la lunghezza massima dei dati da trasferire non superi i 32 byte.
 Il 2° operando indica il numero di un Testo contenente l'indirizzo del partner remoto, gli elementi che devono essere trasferiti e dove tali elementi devono essere copiati (stazione remota).
 Il 3° operando indica l'indirizzo base di 10 Flag Diagnostici (o Uscite).
 Il primo elemento diagnostico è il flag EXEC. Inizialmente questo viene impostato a livello basso dall'istruzione LTXD, e rimane a tale livello per tutte le esecuzioni successive della stessa istruzione LTXD, finchè il trasferimento non viene completato. A questo punto, il flag EXEC viene posto a livello Alto. Se l'istruzione LTXD viene nuovamente eseguita, il trasferimento dei dati viene ripetuto. Lo stato del flag EXEC viene modificato solo quando viene eseguita l'istruzione LTXD.

Utilizzo

LTXD[X] **priorità** ; 0 = alta, 1 = bassa
 testo (i) ; Numero del Testo contenente il comando 0-7999
 diag ; Indirizzo base flag diagnostici F|O 0-8182

Esempio

```
LTXD      1      ; Trasferimento a bassa priorità
          900    ; Il Testo 900 contiene il comando
F 200    ; I flag Diagnostici sono F 200-209
```

Flag

Invariati.

Vedere anche

LRXD, LRXS, LTXS, Flag Diagnostici LAN2, Testi di Comando LAN2.

Uso pratico

Dopo la commutazione dell'ingresso I 8 sul PCD locale, gli ingressi 0-7 vengono letti una sola volta e vengono trasferiti alla stazione PCD remota numero 3, sulle uscite 40-47.

```
COB      5
          0
STH      I 8      ; Se l'ingresso 8 diventa Alto (H)
DYN      F 8
ORL      F 100   ;      Flag EXEC
CPB      H 51   ;      Allora chiama il PB 51
ECOB
PB       51
LTXD    1      ; Priorità
          16     ; Numero Testo
          F 100  ; Flag EXEC

$LAN
TEXT 16  " 3 : I0-7 : O40-47 "
$ENDLAN

EPB
```

LRXS RICEZIONE STATO VIA LAN2 (Receive Status via LAN2)

Descrizione Legge lo stato di un PCD remoto, ponendolo nei flag di stato definiti oppure legge le informazioni statistiche (controllo del traffico) relative alla stazione locale. Lo stato può essere:

- HALT** CPU bloccata
- RUN** CPU attiva
- DIS** Stazione scollegata dalla rete LAN2
- CON** Stazione collegata alla rete LAN2

Il primo operando è il numero di un Testo che contiene l'indirizzo della stazione di cui si richiede la lettura dello stato.

Il secondo operando è l'indirizzo base di 10 Flag Diagnostici (o Uscite).

Il primo elemento diagnostico è il flag EXEC. Inizialmente questo viene impostato a livello basso dall'istruzione LRXS, e rimane a tale livello per tutte le esecuzioni successive della stessa istruzione LRXS, finchè il trasferimento non viene completato.

A questo punto, il flag EXEC viene posto a livello Alto.

Lo stato del flag EXEC viene modificato solo quando viene eseguita l'istruzione LRXS.

L'istruzione LRXS ha sempre la priorità più elevata.

Se lo stato ricevuto è "Scollegato", si attiva solo il Flag "Scollegato" in quanto non sono stati segnalati errori.

Utilizzo

| | | | |
|----------------|--------------|------------|--|
| LRXS[X] | testo | (i) | ; Numero del Testo contenente il comando 0-3999 |
| | diag | | ; Indirizzo base flag diagnostici F O 0-8182 |

Esempio

```
LRXS            100    ; Il Testo 100 contiene il comando
                  O 32    ; Le Uscite Diagnostiche sono O 32-41
TEXT 100 " 020 " ; Legge lo stato della stazione 20
```

Flag

Invariati.

Vedere anche

LTXD, LRXD, LTXS, Flag Diagnostici LAN2, Testi di Comando LAN2

LTXS TRASMISSIONE STATO VIA LAN2 (Transmit Status via LAN2)

Descrizione Modifica lo stato di un PCD remoto.

Lo stato può essere modificato in:

| | |
|-----------------|---|
| HALT | CPU Bloccata |
| RUN | CPU Attiva |
| DIS | Disconnessione stazione dalla rete LAN 2 |
| CON | Conenssione stazione alla rete LAN 2 |
| TOUT:nnn | Impostazione valore di Time-out dove nnn = Valore in 1/10 sec (10..250). |

Il primo operando è il numero di un Testo che contiene sia l'indirizzo della stazione di cui si deve cambiare lo stato, sia il nuovo stato.

Il secondo operando è l'indirizzo base di 10 Flag Diagnostici (o Uscite).

Il primo elemento diagnostico è il flag EXEC. Inizialmente questo viene impostato a livello basso dall'istruzione LTXS, e rimane a tale livello per tutte le esecuzioni successive della stessa istruzione LTXS, finchè il trasferimento non viene completato.

A questo punto, il flag EXEC viene posto a livello Alto.

Lo stato del flag EXEC viene modificato solo quando viene eseguita l'istruzione LTXS.

L'istruzione LTXS ha sempre la priorità più elevata.

Utilizzo

| | | | |
|----------------|--------------|------------|--|
| LTXS[X] | testo | (i) | ; Numero del Testo contenente il comando 0-7999 |
| | diag | | ; Indirizzo base flag diagnostici F O 0-8182 |

Esempio

```
LTXS      1000      ; Il Testo 1000 contiene il comando
          F 100      ; I Flag Diagnostici sono F 100-109
TEXT 1000 "035:DIS" ; Disconnette la stazione 35
```

Flag

Invariati.

Vedere anche

LRXS, LRXD, LTXD, Flag Diagnostici LAN2, Testi di Comando LAN2

Flag Diagnostici LAN2

Per ogni istruzione SAIA LAN2, un operando fornisce l'indirizzo base per i 10 elementi diagnostici (Uscite o Flag).

| Flag | | Stato Alto (H) | Stato Basso (L) |
|------|-------------------------|--------------------------------|----------------------|
| 0 | EXEC | Comando eseguito | Comando in corso |
| +1 | FAIL | Anomalia | Nessuna Anomalia |
| +2 | Stato PCD Locale | Testo Comando non Valido | Testo Comando Valido |
| +3 | | Non connesso | Connesso |
| +4 | | Time Out (Errore trasmissione) | Nessun Timeout |
| +5 | Stato PCD Remoto | Non connesso | Connesso |
| +6 | | Non usato | Non usato |
| +7 | | Protetto da scrittura | Scrittura abilitata |
| +8 | | HALT (CPU 0) | RUN (CPU 0) |
| +9 | Watchdog ⁽¹⁾ | Riconfigurazione LAN2 | |

⁽¹⁾ a partire dalla versione V003 (PCD4) e V007 (PCD6)

Flag EXEC:

Il primo elemento diagnostico è rappresentato dal Flag EXEC. Fin tanto che EXEC è Basso (L), significa che l'istruzione LAN2 è ancora in esecuzione (ricezione o trasmissione dati). Il flag EXEC è impostato inizialmente al valore Basso (L) quando inizia l'esecuzione di un'istruzione LAN2 e rimane basso per tutte le esecuzioni successive della stessa istruzione, finchè il trasferimento non è terminato. Completato il trasferimento, il flag EXEC viene impostato al livello Alto (H).

I programmi devono essere strutturati in modo tale che le istruzioni LAN2 vengano eseguite in continuazione fin tanto che il Flag EXEC è Basso

Se l'istruzione LAN2 viene eseguita nuovamente, viene ripetuto il trasferimento dati. Lo stato del Flag EXEC viene alterato SOLO quando l'istruzione LAN2 è stata interamente eseguita.

PRIORITA':

Il trasferimento delle informazioni attraverso la rete LAN2 è orientato al trasferimento di "frame"; i frame sono delle strutture dati lunghe 32 byte e quindi possono contenere 8 R|C|T oppure 256 I|O|F.

Quando la stazione riceve il "Token" (gettone), viene trasmesso un solo frame ed il "token" passa alla stazione successiva. Un "Telegramma" più lungo di un frame verrà suddiviso in più frame e avrà sempre una priorità bassa (1); se è stata richiesta una priorità alta, il flag +2 (Testo di comando non valido) verrà impostato ed il telegramma non sarà inviato.

Un "Telegramma corto" (lunghezza inferiore ad un frame), può essere inviato con priorità alta (0) o bassa (1). Se la priorità definita è alta, il frame può essere inserito tra le varie frame di un "Telegramma lungo" (a priorità bassa).

I "telegrammi di stato" hanno sempre priorità alta (0).

Testi di Comando LAN2

Trasferimento Dati (LRXS/LTXS)

Le istruzioni LTXD e LRXD richiedono un testo di comando che definisca i dati da trasmettere o i dati ricevuti attraverso la rete LAN2.

In questi testi si devono utilizzare sempre CARATTERI MAIUSCOLI.

FORMATO:



Ciascun testo di comando contiene il numero della stazione remota, il tipo di sorgente con la lista degli indirizzi, e il tipo di destinazione con la relativa lista di indirizzi. Il numero della stazione e le liste degli indirizzi sorgente e destinazione devono essere separate con il carattere ':'.

La lista degli indirizzi può essere:

- Un **intervallo di valori**: due indirizzi separati da un '-'
es: "I100-200" (Indirizzi Ingressi da 100 a 200)
- Una **lista di valori**: fino a 8 elementi singoli separati da virgola
es: "R100,110,120".

Il numero di elementi per la sorgente e la destinazione deve coincidere.

NOTA:

Non tutte le possibili combinazioni di elementi sorgenti e destinazione sono valide (ad esempio non si possono trasferire Flag all'interno di Registri)

| Sorgente | Destinazione | | | | | Campo Indirizzi |
|----------|--------------|---|---|---|---|-----------------|
| | O | F | T | C | R | |
| I | ● | ● | | | | 0..8191 |
| O | ● | ● | | | | 0..8191 |
| F | ● | ● | | | | 0..8191 |
| T | | | ● | ● | | 0..450 |
| C | | | ● | ● | | 0..1599 |
| R | | | | | ● | 0..4095 |

Esempi di comando RICEZIONE DATI (LRXD):

TEXT 100 "015:O64-127:F1000-1063"

Trasferisce il valore delle Uscite da 64 a 127 della stazione PCD remota numero 15 nei flag 1000-1063 del PCD locale.

TEXT 101 "008:T11,13,25:C55,125,1231"

Trasferisce il contenuto dei Temporizzatori 11,13 e 25 dalla stazione PCD remota numero 8, nei Contatori 55, 125 e 1231 del PCD locale.

Esempio di comando TRASMISSIONE DATI (LTXD):

TEXT 75 "000:R11-22:R33-44"

Trasferisce il contenuto dei Registri da 11 a 22 dal PCD locale nei registri da 33 a 44 della stazione PCD remota numero 0.

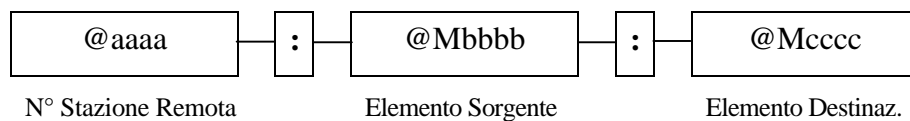
Indirizzamento indiretto

E' possibile ridurre il numero di testi impiegato per definire i trasferimenti via LAN2 utilizzando un indirizzamento indiretto: l'indirizzo effettivo è determinato dal contenuto di un registro.

Ogni singola parte del testo di comando LAN2 (numero di stazione, sorgente e destinazione) può essere indirizzata in modo indiretto utilizzando il carattere '@'.

Indirizzamento indiretto di un singolo elemento

FORMATO:



dove: @ indica che l'indirizzamento è indiretto.
 M indica il tipo di dato (I|O|F|T|C|R)
 aaaa è il registro contenente il numero della stazione remota
 bbbb è il registro contenente l'indirizzo dell'elemento sorgente
 cccc è il registro contenente l'indirizzo dell'elemento destinazione

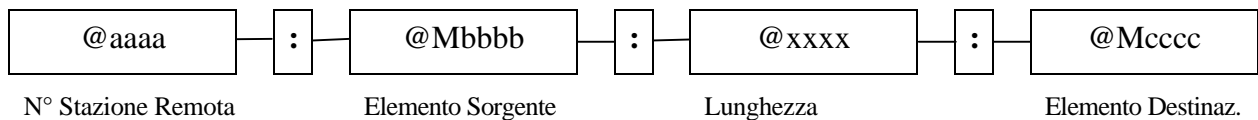
Esempio:

"@100:@I400:@F600"

Il numero della stazione remota è contenuto nel registro 100. L'ingresso, il cui indirizzo è fornito dal registro 400, sarà trasferito al flag il cui indirizzo è specificato dal registro 600.

Indirizzamento indireto di elementi multipli:

FORMATO:



dove: @ indica che l'indirizzamento è indiretto.
 M indica il tipo di dato (I|O|F|T|C|R)
 aaaa è il registro contenente il numero della stazione remota
 bbbb è il registro contenente l'indirizzo del 1° elemento sorgente
 xxxx è il registro contenente il numero degli elementi da trasferire
 cccc è il registro contenente l'indirizzo del 1° elemento destinazione

Esempio:

```
"@200:@C100-@101:@C500"
```

Il numero della stazione remota è nel Registro 200. L'indirizzo del primo contatore si trova nel registro 100, il numero dei contatori da trasferire si trova nel registro 101. Questi contatori verranno copiati nei contatori a partire dall'indirizzo presente nel registro 500.

Indirizzamento misto

E' possibile utilizzare indirizzamenti sia diretti che indiretti all'interno dello stesso testo di comando.

In caso di indirizzamento misto è obbligatorio seguire la sintassi dell'indirizzamento indiretto.

Esempio:

```
"@5:R100-50:@99"
```

Il numero della stazione remota è 5. 50 Registri, a partire dal Registro 100, vengono trasferiti e copiati a partire dall'indirizzo presente nel Registro 99.

```
"25:I0-@55:F1000"
```

Il numero della stazione remota si trova nella stazione 25. Gli ingressi tra I 0 e il contenuto del Registro 55 vengono trasferiti nei Flag a partire dall'indirizzo 1000.

Note sull'indirizzamento indiretto e misto

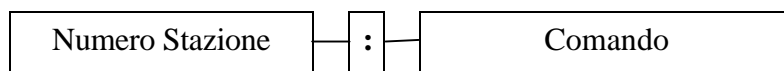
Poiché il contenuto dei registri utilizzati nell'indirizzamento indiretto o misto non può essere controllato dall'Assemblatore, l'utente deve prestare particolare attenzione a non superare il campo degli elementi indirizzati.

Il superamento del campo autorizzato può dar luogo ad effetti imprevedibili.

Trasmissione Stato (LTXS)

Attraverso la LAN2 è possibile trasferire lo stato di una stazione.

FORMATO:



| Numero Stazione | |
|-----------------|---|
| 0-254 | Il comando viene inviato alla stazione specificata |
| 255 | Il comando viene inviato a tutte le stazioni collegate alla rete ad eccezione della stazione che esegue il comando |
| Comando | |
| HALT | Pone tutti i processori della stazione specificata in HALT. Il programma utente si arresta e viene attivato il flag diagnostico 8 |
| RUN | Pone tutti i processori della stazione specificata in RUN |
| CON | Pone la stazione LAN2 in stato di connessione. Il flag diagnostico 3 o 5 viene posto a livello basso. |
| DIS | Pone la stazione LAN2 in stato di disconnessione. In questo caso non viene eseguita alcuna istruzione e non viene trasferito alcun dato. Il flag diagnostico 3 o 5 viene posto a livello alto |
| TOUT:NNN | Imposta il timeout per la stazione specificata. NNN è il numero della stazione nella rete (2-255) |

Esempio di testo per TRASMISSIONE STATO (LTXS):

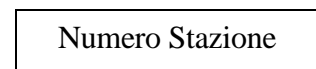
TEXT 213 "018:HALT"

Lo stato della stazione Remota 18 viene commutato in HALT.

Letture Stato (LRXS)

E' possibile leggere lo stato di una stazione:

FORMATO:



Lo stato viene riportato nei flag diagnostici (Flag 3, 5 o 8).

Esempio di testo per RICEZIONE STATO (LRXS):

TEXT 137 "015"

Lo stato della stazione 15 viene letto nel flag di stato:

"Scollegata" Flag EXEC + 5

"Bloccata (Halt)" Flag EXEC + 8

Il flag "Anomalia (Fail)" (Flag EXEC + 1) non viene influenzato.

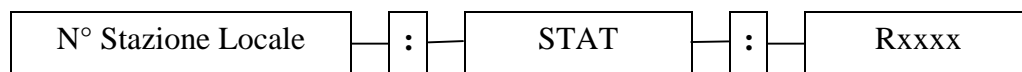
Se lo stato letto è "scollegato", viene attivato solo il flag "Scollegato", dal momento che non c'è alcun errore.

Nota: La comunicazione LAN2 è completa solo quando il flag EXEC ritorna = H, dopo la nuova esecuzione di LRXS.

Controllo del Traffico di Linea (LRXS)

Per mezzo dell'istruzione LRXS e di uno speciale testo di stato è possibile verificare cosa avviene sulla rete SAIA LAN2:

FORMATO:



dove xxxx è il numero di un registro (0-4095)

Questo testo di comando è utilizzato per leggere le informazioni statistiche di trasmissione/ricezione della rete LAN2 in 4 registri a partire dal registro xxxx. Questo comando è utile per la diagnosi degli errori hardware verificatisi sulla rete.

I valori restituiti sono i seguenti:

| Registro | Descrizione |
|----------|--|
| xxxx | Numero di frame ricevuti |
| xxxx + 1 | Numero di frame trasmessi |
| xxxx + 2 | Numero di tentativi (=numero di timeout * 3) |
| xxxx + 3 | Numero di messaggi respinti (=messaggi alterati) |

Nota: Durante la prima comunicazione con una stazione, viene incrementato il numero di tentativi: ciò significa che ogni stazione che sta comunicando ha almeno 1 tentativo. I valori nei registri delle statistiche vengono memorizzati utilizzando un formato a 16 bit (il valore massimo è 65535)

Esempio :

```
TEXT 123 "100:STAT:R20"
```

Legge le statistiche relative alla stazione locale (100) e ne memorizza i valori nei Registri da 20 a 23.

Sintassi dei Testi LAN2 nel file sorgente:

La sintassi dei testi LAN2 posti tra le direttive \$LAN (\$LLAN) e \$ENDLAN (\$NOLAN, \$NOLLAN), viene verificata durante la fase di assemblaggio. I caratteri minuscoli vengono convertiti in maiuscolo.

Inoltre, l'Assembler converte i testi LAN2 in formato binario in modo da velocizzare la comunicazione sulla rete LAN2. Il tempo risparmiato è circa 15ms per ogni comando; ciò risulta particolarmente efficace in caso di telegrammi corti.

Esempi:

```
$LAN  
TEXT 0 "2:I0-255:F1000-1255"  
TEXT 1 "5:r501-510:r101-110"  
TEXT 2 "7:f0-31:o96-127"  
TEXT 15 "2:con"  
TEXT 37 "15"  
$ENDLAN
```

Nota: I testi binari della LAN non possono essere visualizzati od editati con il "debugger" in quanto hanno un formato diverso.

Utilizzo di Simboli nei testi LAN2:

Nei testi LAN2 è possibile anche utilizzare dei simboli.

Il valore e opzionalmente il tipo del simbolo vengono inseriti all'interno del testo. Il simbolo viene scritto al di fuori del segmento di testo ASCII, tra virgolette, e deve essere separato dagli altri simboli per mezzo di una virgola. Dopo il simbolo, si possono specificare (opzionalmente) la larghezza del campo e il tipo di prefisso.

Formato:

| simbolo [. [[-] [0] larghezza] [t T]] | |
|--|---|
| simbolo | Nome del simbolo. Può essere in effetti una qualsiasi espressione che include un simbolo ad esempio: MotorOn + 100, ... Non sono ammessi simboli con valori in virgola mobile. |
| . | Il punto immediatamente dopo il simbolo indica che esiste un campo larghezza e/o un prefisso. |
| larghezza | É il campo larghezza: numero di cifre o di spazi richiesti dal numero. Se la larghezza inizia con 0, vengono inseriti gli zeri non significativi. |
| t T | Prefisso opzionale tipo 't' o 'T'. Se 't', il valore viene preceduto dal tipo del simbolo in minuscolo (o, f, r, ...); se 'T', il tipo del simbolo sarà in maiuscolo (O, F, R,...). |

Esempi:

```

SOURCE      EQU   R 55
DEST        EQU   R 66

$LAN
TEXT  25    "8:" , SOURCE.T , ":" , DEST.T
$ENDLAN
    
```

In questo modo viene generato il testo: "008:R55:R66"

10. Istruzioni di CONTROLLO

Queste istruzioni controllano l'esecuzione del programma.

Le istruzioni di salto sono spesso causa di errori (loop all'infinito, etc.); tali istruzioni devono pertanto essere utilizzate con particolare attenzione. E' preferibile utilizzare le istruzioni di salto in blocchi programma o blocchi funzione anziché nel programma principale.

Il GRAFTEC consente all'utente di programmare senza dover utilizzare istruzioni di salto (jump).

L'operando di queste istruzioni non può essere fornito come parametro di un Blocco Funzione.

| | |
|---------------|---|
| JR | Jump relative (Salto relativo) |
| JPD | Jump direct (Salto diretto) |
| JPI | Jump indirect (Salto indiretto) |
| HALT | Halts the CPU (Arresta la CPU) |
| LOCK | Lock semaphore (Attiva un semaforo) |
| UNLOCK | Unlock semaphore (Disattiva un semaforo) |

JR SALTO RELATIVO (Jump Relative)

Descrizione Salta in modo condizionato o incondizionato un numero specificato di righe di programma in avanti o indietro rispetto al numero di riga corrente.

Il numero di righe che si può saltare va da -4095 (salto indietro) a +4095 (salto avanti); la riga di programma a cui saltare viene calcolata sommando tale valore al numero di riga contenente l'istruzione JR. Non è consentito saltare fuori dal blocco corrente (COB, PB, FB, ST, TR o SB): la destinazione DEVE quindi trovarsi all'interno del blocco corrente.

I codici di condizione validi sono i seguenti:

| | |
|----------|--|
| - | Salto incondizionato (codice condizione vuoto) |
| H | Salta se ACCU = H (1) |
| L | Salta se ACCU = L (0) |
| P | Salta se Flag Positivo = H (Flag Negativo = L) |
| N | Salta se Flag Negativo = H |
| Z | Salta se Flag Zero = H |
| E | Salta se Flag Errore = H |

Se la condizione non è vera, il salto non viene effettuato; l'esecuzione continua con l'istruzione successiva a JR.

Quando si programma utilizzando il linguaggio Assembler, si utilizzano solitamente le Label (nomi simbolici) come destinazione dei salti. Le Label possono avere qualsiasi lunghezza, ma sono significativi solo i primi 8 caratteri; le label devono sempre iniziare con una lettera (A-Z).

Se si utilizza un editor diverso da SEDIT, è necessario includere un carattere ':' dopo ogni label.

Utilizzo

| | | |
|-----------|-------------|---|
| JR | [cc] offset | ; cc = codice condizione (H L P N Z E) ; offset indica il numero di righe relativo di ; righe da saltare (-4095.. + 4095) |
|-----------|-------------|---|

Esempio

```
JR      H  -2      ; Salta indietro di 2 righe
JR      H REPEAT   ; Salta alla label "REPEAT"
```

Flag

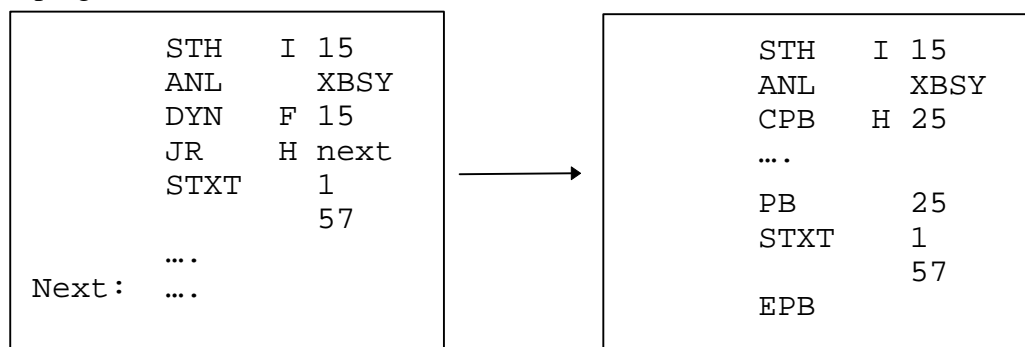
Invariati

Vedere anche

JPD, JPI, LD

Nota

I programmi ben strutturati non utilizzano istruzioni di salto.



JPD SALTO DIRETTO (Jump Direct)

Descrizione Salta in modo condizionato o incondizionato al numero della riga di programma relativo rispetto all'inizio del blocco corrente (COB, XOB, PB, FB, ST o TR). Il numero della riga di destinazione è sempre positivo, ed è compreso tra 0 e il numero di righe del blocco corrente (massimo 8191 righe). Si possono utilizzare le Label.

I codici di condizione validi sono i seguenti:

| | |
|----------|--|
| - | Salto incondizionato (codice condizione vuoto) |
| H | Salta se ACCU = H (1) |
| L | Salta se ACCU = L (0) |
| P | Salta se Flag Positivo = H (Flag Negativo = L) |
| N | Salta se Flag Negativo = H |
| Z | Salta se Flag Zero = H |
| E | Salta se Flag Errore = H |

Se la condizione non è vera, il salto non viene effettuato; l'esecuzione continua con l'istruzione successiva a JPD.

Utilizzo

| | | |
|------------|-------------|--|
| JPD | [cc] offset | ; cc = codice condizione (H L P N Z E) ; offset indica lo spostamento rispetto all'inizio ; del blocco (0..8191) |
|------------|-------------|--|

Esempio JPD L 10 ; Se ACCU è Basso, viene eseguito il salto
; alla 10^{ma} riga del blocco corrente.

Flag Invariati.

Vedere anche JR, JPI

JPI SALTO INDIRETTO (Jump Indirect)

Descrizione Simile a JPD; salta in modo condizionato o incondizionato al numero della riga di programma relativo rispetto all'inizio del blocco corrente (COB, XOB, PB, FB, ST o TR).

Il numero della riga di programma viene letto dal Registro specificato (sono considerati solo i 13 bit meno significativi). Dal momento che questa istruzione utilizza un codice di condizione, il codice relativo al tipo dati 'R' viene omesso.

I codici di condizione validi sono i seguenti:

| | |
|----------|--|
| - | Salto incondizionato (codice condizione vuoto) |
| H | Salta se ACCU = H (1) |
| L | Salta se ACCU = L (0) |
| P | Salta se Flag Positivo = H (Flag Negativo = L) |
| N | Salta se Flag Negativo = H |
| Z | Salta se Flag Zero = H |
| E | Salta se Flag Errore = H |

Se la condizione non è vera, il salto non viene effettuato; l'esecuzione continua con l'istruzione successiva a JPI.

Il valore di una label può essere caricato in un Registro utilizzando l'istruzione LD.

| | |
|-----------------|---|
| Utilizzo | JPI [cc] reg ; cc = codice condizione (H L P N Z E) ; reg è il numero del Registro contenente ; l'offset rispetto all'inizio del blocco (0..8191) |
|-----------------|---|

Esempio JPI H 300 ; Se ACCU è Alto, viene effettuato il salto
; alla riga del blocco corrente, specificata
; nel Registro 300

Flag Invariati

Vedere anche JR, JPI, LD

Nota



Occorre accertarsi che la destinazione del salto non sia esterna al blocco corrente !.

HALT ARRESTA LA CPU (Halts the CPU)

Descrizione Arresta in modo condizionato o incondizionato la CPU. Lo stato di Halt non è uguale allo stato di Stop. Dopo un comando di Halt, la CPU può essere messa in stato di Run solo eseguendo un riavvio (Restart) oppure spegnendo e riaccendendo il PCD.

Se la condizione non è vera, l'ARRESTO (HALT) non viene effettuato; l'esecuzione continua con l'istruzione successiva.

NOTA: Dopo un HALT della CPU0, è possibile eseguire solo una Ripartenza a freddo di tutte le CPU. Lo stato delle uscite, dopo un comando di HALT, è definito dalla configurazione hardware (ponticelli).

I **codici di condizione** validi sono i seguenti:

| | |
|----------|--|
| - | Salto incondizionato (codice condizione vuoto) |
| H | Salta se ACCU = H (1) |
| L | Salta se ACCU = L (0) |
| P | Salta se Flag Positivo = H (Flag Negativo = L) |
| N | Salta se Flag Negativo = H |
| Z | Salta se Flag Zero = H |
| E | Salta se Flag Errore = H |

Utilizzo `HALT [cc] ; cc = codice condizione: H|L|P|N|Z|E`

Esempio `HALT E ;` Arresta la CPU se viene impostato il flag di Errore (E)

Flag Invariati

Vedere anche Manuale Utente

Uso pratico In caso di errore, arresta il PCD e memorizza in un Registro alcune informazioni diagnostiche

```

XOB      13
DIAG     R 1000
HALT
EXOB
    
```



ATTENZIONE

L'istruzione HALT dovrebbe essere utilizzata principalmente durante la fase di verifica. In un programma applicativo definitivo essa dovrebbe essere utilizzata con la massima cautela !

LOCK ATTIVA UN SEMAFORO (Lock Semaphore)

Descrizione L'istruzione LOCK viene utilizzata, congiuntamente all'istruzione UNLOCK, per impedire conflitti di accesso dovuti al fatto che più CPU possono leggere o scrivere gli stessi elementi. Sono disponibili 100 Semafori (flag speciali), (0-99). L'istruzione LOCK controlla il Semaforo. Se è Alto (un'altra CPU ha eseguito in precedenza una LOCK), l'ACCU viene posto al livello Basso. Se il Semaforo è al livello Basso, sia l'ACCU che il Semaforo, vengono posti a livello Alto.

É responsabilità del programmatore accertarsi che la CPU non acceda ad un elemento se il Semaforo associato a quest'ultimo è Alto (ACCU = L (0) dopo l'istruzione LOCK.

L'istruzione UNLOCK azzera il Semaforo. L'istruzione LOCK DEVE essere seguita rapidamente da una istruzione UNLOCK in modo da non bloccare per troppo tempo l'accesso ad un elemento da parte di una CPU.

Utilizzo

| |
|---|
| LOCK semaforo ; Semaforo 0-99 |
|---|

Esempio

LOCK 1 ; Se il Semaforo 1 è Basso (nessuna altra CPU
CFB H 100 ; sta accedendo ai dati), allora chiama il FB 100.
; Il Semaforo 1 è utilizzato per proteggere i dati

Flag

ACCU impostato Alto (H)/Basso (L)

Vedere anche

UNLOCK

Uso pratico

vedere istruzione UNLOCK

UNLOCK DISATTIVA UN SEMAFORO (Unlock Semaphore)

Descrizione L'istruzione UNLOCK viene utilizzata, congiuntamente all'istruzione LOCK, per impedire conflitti di accesso dovuti al fatto che più CPU possono leggere o scrivere gli stessi elementi. Sono disponibili 100 Semafori (flag speciali), (0-99). L'istruzione UNLOCK azzerava il Semaforo.

Utilizzo **UNLOCK semaforo ; Semaforo 0-99**

Esempio UNLOCK 1 ; Il Semaforo 1 viene azzerato (posto a livello Basso - L)

Flag Invariati.

Vedere anche LOCK

Uso pratico Un PCD è equipaggiato con due CPU. La CPU 0 confronta il contenuto di 2 registri, mentre la CPU1 trasferisce informazioni BCD in uno di questi due registri.

| CPU 0 | | CPU 1 | |
|---------------|----------|---------------|----------|
| ... | | ... | |
| LOCK | 1 | LOCK | 1 |
| CFB | H 10 | CFB | H 100 |
| ... | | ... | |
| FB | 10 | FB | 100 |
| CMP | R 88 | DIGI | 2 |
| | R 89 | | I 16 |
| UNLOCK | 1 | | R 88 |
| EFB | | DIGI | 2 |
| | | | I 24 |
| | | | R 89 |
| | | UNLOCK | 1 |
| | | EFB | |

L'utilizzo del semaforo 1 assicura che la CPU0 non esegua mai il confronto tra i due registri mentre la CPU1 sta eseguendo l'istruzione DIGI e sta alterando il contenuto dei registri. Se la CPU0 confrontasse i registri nell'istante in cui la CPU1 li sta aggiornando, potrebbe confrontare un valore nuovo con uno vecchio. Il semaforo 1 impedisce anche alla CPU1 di eseguire le istruzioni DIGI finchè la CPU0 non ha terminato l'istruzione CMP.

11. Istruzioni di DEFINIZIONE

Queste istruzioni vengono eseguite una SOLA volta all'accensione.
Se una istruzione viene eseguita nuovamente, viene ignorata.

Normalmente queste istruzioni vengono inserite nel blocco di partenza XOB 16.

L'operando di queste istruzioni non può essere fornito come parametro di un Blocco Funzione.

| | |
|---------------|--|
| DEFVM | Define volatile memory (Flags) (Definizione memoria volatile - Flag) |
| DEFTC | Define Timers/Counters (Definizione Temporizzatori/Contatori) |
| DEFTB | Define timebase (Definizione base tempi) |
| DEFTR | Define Timer Resolution (Definizione risoluzione temporizzatori) |
| DEFWPH | Define write protected area (Halt) (Definizione area protetta da scrittura - Halt) |
| DEFWPR | Define write protected area (Run) (Definizione area protetta da scrittura -Run) |

DEFVM DEFINIZIONE MEMORIA VOLATILE (Flag) (Define Volatile Memory - Flags)

Descrizione Definisce l'area di memoria riservata ai Flag di tipo non-volatile (tamponati mediante batteria). I Flag non volatili mantengono i rispettivi valori, anche dopo lo spegnimento del PCD.
I Flag volatili vengono impostati tutti al valore 0 all'accensione del PCD. Tutti i Flag **AL DI SOPRA** di quello indicato nell'operando vengono definiti come non-volatili.

Se non viene utilizzata l'istruzione DEFVM, TUTTI i Flag vengono considerati non volatili per default.

Questa istruzione viene eseguita solo dalla CPU 0.

Utilizzo

| | |
|-------------------|--|
| DEFVM flag | ; 0-8190, partizione Flag vol/non-vol |
|-------------------|--|

Esempio

```
DEFVM    200    ; Flags    I flag 0 - 200 sono volatili,  
                 ;            mentre i flag 201 - 8191 sono non-volatili
```

Flag

Invariati.

Vedere anche

DEFTC, DEFTB, DEFWPR, DEFWPH

Uso pratico

Si supponga di voler dichiarare i Flag 0...200 come volatili.

```
XOB                    16    ; XOB di Partenza a freddo
```

```
DEFVM                200    ; I Flag 201..8191 sono non volatili
```

```
. . . .  
EXOB
```

DEFTC **DEFINIZIONE TEMPORIZZATORI/CONTATORI** **(Define Timer/Counters)**

Descrizione Definisce il numero di Temporizzatori per il PCD. I Temporizzatori e i Contatori occupano lo stesso spazio di indirizzi. Tutti gli elementi **AL DI SOTTO** del valore dell'operando vengono considerati come Temporizzatori, tutti gli altri vengono considerati come Contatori.

Nel caso in cui non venga utilizzata l'istruzione DEFTC, per default si avrà:

| | |
|-----------------|------------|
| Temporizzatori: | 0 - 31 |
| Contatori: | 32 - 1599. |

NOTA: Non è conveniente definire un numero di Temporizzatori maggiore di quello effettivamente richiesto dal programma. La gestione di ogni Temporizzatore riduce infatti la velocità di esecuzione del programma stesso. Il numero massimo di Temporizzatori ammessi è 450.

Questa istruzione viene eseguita solo dalla CPU 0.

Utilizzo

DEFTC ctr ; Limite inferiore per i Contatori (0-450)

Esempio DEFTC 64 ; Temporizzatori 0-63, Contatori 64-1599

Flag Invariati.

Vedere anche DEFTB, DEFTR, DEFVM, DEFWPR, DEFWPH

Uso pratico Si supponga che siano necessari 100 Temporizzatori per una applicazione

XOB 16 ; XOB di Partenza a freddo

DEFTC 100 ; 0..99 sono Temporizzatori
; 100..1599 sono Contatori

. . . .

EXOB

DEFTB DEFINIZIONE BASE TEMPI (Define Timebase)

Descrizione Definisce l'unità di tempo (base dei tempi) per il decremento dei Temporizzatori. L'operando indica il valore dell'unità di tempo in multipli di 10 ms. Sono ammessi valori da 1 a 1000 (da 10ms a 10 sec).

Se non viene definita l'unità di tempo (DEFTB non è utilizzata), il valore di default è 100 ms (1/10 di secondo).

L'istruzione DEFTB viene elaborata solo dalla CPU 0.

Per le altre CPU, DEFTB definisce la base tempi per i temporizzatori interni utilizzati dalle istruzioni “ritardate” SETD e RESD. Quindi, i temporizzatori interni della CPU 0 hanno sempre la stessa base tempi utilizzata per i Temporizzatori utente; la base tempi dei temporizzatori interni delle altre CPU può essere diversa rispetto a quella definita per i Temporizzatori utente.

Nota: Nota: occorre prestare attenzione a non definire una base dei tempi troppo bassa (es.: 10ms) quando è attivo un numero elevato di temporizzatori (in tal caso si ridurrebbe la velocità di esecuzione del programma).

Utilizzo

DEFTB basetempi ; base dei tempi in step da 10 ms (1- 1000)

Esempio DEFTB 100 ; Base dei tempi = 1 sec (100 * 10ms)

Flag Invariati.

Vedere anche DEFVM, DEFTC, DEFWPR, DEFWPH,SETD, RESD

Uso pratico Si supponga che la temporizzazione minima per un processo lento sia di 1 sec

XOB 16 ; XOB di Partenza a freddo

DEFTB 100 ; La base tempi è 100 * 10 ms = 1000 ms

.....
EXOB

DEFTR DEFINIZIONE RISOLUZIONE TEMPORIZZATORI (Define Timer Resolution)

Descrizione Permette di definire la risoluzione, **in millisecondi**, con cui verranno decrementati i Temporizzatori. Per esempio, se viene definita l'istruzione "DEFTR 100", tutti i Temporizzatori non a zero verranno decrementati di 100 ogni 100 ms. L'istruzione "DEFTR 1000" decremerà invece tutti i Temporizzatori di 1000 ogni 1000 ms e così via. Se all'interno dello stesso programma sono presenti le istruzioni DEFTR e DEFTB, all'interno dell'elenco storico (History List) verrà inserito il messaggio "**DOUBLE TIME BASE**" (Base dei Tempi Doppia) e la CPU all'accensione o in caso di riavvio a freddo commuterà automaticamente nello stato "HALT".

Il vantaggio dell'istruzione DEFTR, rispetto a DEFTB, è dato dal fatto che i valori specificati per i temporizzatori utilizzati sono indipendenti dalla base dei tempi o dalla risoluzione e sono sempre valori multipli di 10 ms. Per consentire all'istruzione DEFTR di influenzare i Temporizzatori, è necessario programmare tale istruzione sulla CPU0. L'istruzione DEFTR ammette una risoluzione temporizzatori massima di 10 ms: ciò significa che il valore specificato nelle varie istruzioni verrà arrotondato, se necessario.

Esempio: DEFTR 25: verrà definita una base dei tempi di 20 ms (25 arrotondato al valore inferiore 20). L'istruzione DEFTR, come l'istruzione DEFTB, influenza anche le istruzioni SETD, RESD e OUTD. Se all'interno del programma utente è presente l'istruzione DEFTR, la base dei tempi delle suddette istruzioni verrà fissata a **10ms** indipendentemente dal valore specificato mediante l'istruzione DEFTR.

Utilizzo

| | | |
|--------------|--------------------|------------------------------|
| DEFTR | risoluzione | ; risoluzione ≥ 10 ms |
|--------------|--------------------|------------------------------|

Esempio DEFTR 100 ; Risoluzione Temporizzatori = 100 msec

Flag Invariati.

Vedere anche DEFTB

DEFWPR DEFINIZIONE AREA PROTETTA DA SCRITTURA (RUN) (Define Write Protected Area - RUN)

Descrizione Definisce quali elementi devono essere protetti da scrittura da parte di stazioni della LAN2.

DEFWPR definisce gli elementi da proteggere quando la CPU è in Run (DEFWPH definisce gli elementi da proteggere quando la CPU è in Halt o in Stop).

Per entrambe le istruzioni l'operando definisce il tipo di elemento e il limite superiore del campo da proteggere. Gli elementi indirizzati compresi tra 0 e questo valore sono protetti da scrittura. Le istruzioni devono essere eseguite una volta sola per ciascun tipo di elemento da proteggere: O, F, T, C, R. Se si devono proteggere tutti i tipi di elementi, occorre eseguire 5 volte l'istruzione DEFWPR.

Se non vengono utilizzate queste istruzioni, NESSUN elemento sarà protetto da scrittura in stato RUN.

Questa istruzione viene eseguita solo dalla CPU 0.

Utilizzo

DEFWPR indirizzi ; O 0-8191, F 0-8191, T 0-450, C 0-1599, R 0-4095

Esempio

DEFWPR F 999 ; I Flag 0-999 sono protetti da scrittura (RUN)
 ; e quindi non possono essere modificati da un'altra
 ; stazione della LAN2

Flag

Invariati.

Vedere anche

DEFWPH, DEFTC, DEFTB, DEFVM, LAN2

Uso pratico

In una applicazione che utilizza la LAN2, si devono proteggere 1000 flag locali e 500 registri per impedirne la scrittura da parte di altre stazioni quando la CPU è in RUN.

XOB 16 ; XOB di Partenza a freddo

DEFWPR F 999 ; Flag 0..999 protetti

DEFWPR R 499 ; Registri 0..499 protetti

. . . .
 EXOB

DEFWPH DEFINIZIONE AREA PROTETTA DA SCRITTURA (HALT) (Define Write Protected Area - HALT)

Descrizione Definisce quali elementi devono essere protetti da scrittura da parte di stazioni della LAN2.

DEFWPH definisce gli elementi da proteggere quando la CPU è in Halt (DEFWPR definisce gli elementi da proteggere quando la CPU è in Run).

Per entrambe le istruzioni l'operando definisce il tipo di elemento e il limite superiore del campo da proteggere. Gli elementi indirizzati compresi tra 0 e questo valore sono protetti da scrittura. Le istruzioni devono essere eseguite una volta sola per ciascun tipo di elemento da proteggere: O, F, T, C, R.

Se si devono proteggere tutti i tipi di elementi, occorre eseguire 5 volte l'istruzione DEFWPH. Se non vengono utilizzate queste istruzioni, NESSUN elemento sarà protetto da scrittura in stato HALT.

Questa istruzione viene eseguita solo dalla CPU 0.

Utilizzo

| |
|---|
| DEFWPH indirizzi ; O 0-8191, F 0-8191, T 0-450, C 0-1599, R 0-4095 |
|---|

Esempio DEFWPH C 79 ; I Temporizzatori e i Contatori 0-79 sono
; protetti da scrittura (in stato Halt)

Flag Invariati.

Vedere anche DEFWPR, DEFTC, DEFTB, DEFVM, LAN2

Uso pratico In una applicazione che utilizza la LAN2, si devono proteggere 1000 flag locali e 500 registri per impedirne la scrittura da parte di altre stazioni quando la CPU è in RUN e in HALT.

XOB 16 ; XOB di Partenza a freddo

; Definizione della protezione quando la CPU è in RUN

DEFWPR F 999 ; Flag 0..999 protetti

DEFWPR R 499 ; Registri 0..499 protetti

; Definizione della protezione quando la CPU è in HALT

DEFWPH F 999 ; Flag 0..999 protetti

DEFWPH R 499 ; Registri 0..499 protetti

.....

EXOB

12. Istruzioni SPECIALI

| | |
|---------------|---|
| NOP | No operation (Nessuna operazione) |
| RTIME | Read time (Lettura ora) |
| WTIME | Write time (Scrittura ora) |
| PID | P.I.D. control (Controllo PID) |
| TEST | Test hardware |
| DIAG | Read XOB diagnostic (Lettura diagnostica) |
| SYSRD | System Read (Lettura dati sistema) |
| SYSWR | System Write (Scrittura dati sistema) |
| SYSCMP | System Compare (Comparazione dati sistema) |

Le seguenti istruzioni non devono più essere utilizzate ma sono state mantenute per ragioni di compatibilità:

| | |
|-------------|--|
| ALGI | Analogue input (Ingresso Analogico) |
| ALGO | Analogue output (Uscita Analogica) |

Queste due istruzioni operano esclusivamente con la scheda analogica PCA2.W1x. Per leggere o scrivere valori da/a schede analogiche PCD2, PCD4 e PCD6, consultare l'appropriato manuale hardware.

| | |
|-------------|---|
| STHS | Start high slow (Inizio lento su stato alto) |
| OUTS | Out slow (Imposta elemento da Accu - lento) |

Queste istruzioni erano utilizzate per l'accesso a moduli di I/O lenti quali il PCA2.W2x / W3x.

RTIME LETTURA ORA (Read Time)

Descrizione Legge il contenuto dell'orologio hardware interno e lo copia in due Registri. Il primo Registro è specificato nell'istruzione. Dopo l'istruzione RTIME i Registri sono configurati come segue:

| | | | | | | | | | | |
|--------------|---|---------|---------|---------|------|------|------|------|--------|--------|
| Cifra numero | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Registro | 0 | 0 | 0 | 0 | Ora | Ora | Min | Min | Sec | Sec |
| Registro + 1 | 0 | Sett.na | Sett.na | gg/Sett | Anno | Anno | Mese | Mese | Giorno | Giorno |

| | | |
|-----------|-----------------------|----------------------------------|
| Settimana | Numero settimana | 1..53 |
| gg/Sett. | N° Giorno della sett. | 1..7 (Lunedì = 1, Domenica = 7) |
| Anno | Anno | 0..99 |
| Mese | Mese dell'anno | 1..12 |
| Giorno | Giorno del mese | 1..28/29/30/31 (in base al mese) |
| Ora | Ora | 0..23 |
| Min | Minuti | 0..59 |
| Sec | Secondi | 0..59 |

I dati nei registri sono memorizzati in binario, NON in BCD, ma possono essere convertiti o emessi in BCD utilizzando l'istruzione DIGO.

Utilizzo

RTIME reg ; Numero Registro, R 0-4094

Esempio

RTIME R 10 ; Il contenuto dell'orologio viene copiato nei Registri 10 e 11

Flag

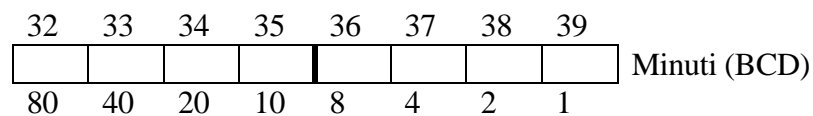
Invariati.

Vedere anche

WTIME, DIGO

Uso pratico

Dopo la commutazione dell'ingresso 3, si devono visualizzare i minuti dell'orologio in formato BCD sulle uscite 32-39



| | | | |
|------|------|--------------|-------------|
| COB | 0 | PB | 25 |
| | 0 | RTIME | R 20 |
| STH | I 3 | MOV | R 20 |
| DYN | F 3 | | D 2 |
| CPB | H 25 | | R 99 |
| ... | | | D 0 |
| ECOB | | MOV | R 20 |
| | | | D 3 |
| | | | R 99 |
| | | | D 1 |
| | | DIGOR | 2 |
| | | | R 99 |
| | | | O 32 |
| | | EPB | |

WTIME SCRITTURA ORA (Write Time)

Descrizione Scrive il contenuto di due Registri nell'orologio hardware interno. Il primo dei due Registri è specificato nell'istruzione. Il formato del contenuto dei Registri è identico a quello dell'istruzione RTIME:

Utilizzando l'istruzione DIGI, si possono caricare valori BCD nei Registri partendo da Flag ecc.

Utilizzo

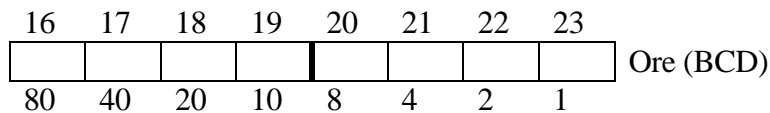
```
WTIME    reg                    ; Registro sorgente, R 0-4094
```

Esempio WTIME R 500 ; Carica l'orologio con i dati dei Registri 500 e 501

Flag Invariati.

Vedere anche RTIME, DIGI

Uso pratico Dopo la commutazione dell'ingresso 4, deve essere aggiornata l'ora dell'orologio. Il nuovo valore deve essere letto dagli switch BCD (ingressi 16-23).



| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--------------|---|--|--|---|--|-----|-----|--|-----|-----|--|-----|------|--|-----|--|--|------|--|--|--|---|----|----|--|-------|-------|--|-------|---|--|--|------|--|--|-------|--|-----|-------|--|--|-----|--|--|-------|--|--|-----|--|-----|-------|--|--|-----|--|--|-------|--|--|-----|--|--------------|--------------|--|-----|--|--|
| <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 30%;">COB</td><td style="width: 10%;">0</td><td style="width: 60%;"></td></tr> <tr><td></td><td>0</td><td></td></tr> <tr><td>STH</td><td>I 4</td><td></td></tr> <tr><td>DYN</td><td>F 4</td><td></td></tr> <tr><td>CPB</td><td>H 26</td><td></td></tr> <tr><td>...</td><td></td><td></td></tr> <tr><td>ECOB</td><td></td><td></td></tr> </table> | COB | 0 | | | 0 | | STH | I 4 | | DYN | F 4 | | CPB | H 26 | | ... | | | ECOB | | | | <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 30%;">PB</td><td style="width: 10%;">25</td><td></td></tr> <tr><td>RTIME</td><td>R 200</td><td></td></tr> <tr><td>DIGIR</td><td>2</td><td></td></tr> <tr><td></td><td>I 16</td><td></td></tr> <tr><td></td><td>R 199</td><td></td></tr> <tr><td>MOV</td><td>R 199</td><td></td></tr> <tr><td></td><td>D 2</td><td></td></tr> <tr><td></td><td>R 200</td><td></td></tr> <tr><td></td><td>D 4</td><td></td></tr> <tr><td>MOV</td><td>R 199</td><td></td></tr> <tr><td></td><td>D 1</td><td></td></tr> <tr><td></td><td>R 200</td><td></td></tr> <tr><td></td><td>D 5</td><td></td></tr> <tr><td>WTIME</td><td>R 200</td><td></td></tr> <tr><td>EPB</td><td></td><td></td></tr> </table> | PB | 25 | | RTIME | R 200 | | DIGIR | 2 | | | I 16 | | | R 199 | | MOV | R 199 | | | D 2 | | | R 200 | | | D 4 | | MOV | R 199 | | | D 1 | | | R 200 | | | D 5 | | WTIME | R 200 | | EPB | | |
| COB | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| STH | I 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DYN | F 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CPB | H 26 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ECOB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PB | 25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RTIME | R 200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DIGIR | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | I 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | R 199 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MOV | R 199 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | D 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | R 200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | D 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MOV | R 199 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | D 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | R 200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | D 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| WTIME | R 200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EPB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

PID ALGORITMO DI CONTROLLO PID (PID Control Algorithm)

Descrizione Implementa un algoritmo PID, utilizzando i dati definiti in un blocco di 13 Registri

| Registro | Utilizzo | Simbolo | | |
|----------|--------------------------|-----------|---|--------------------------|
| +0 | Nuovo Risultato | Y_n | * | dimensione di 'm' bit |
| +1 | Risultato Precedente | Y_{n-1} | * | |
| +2 | Nuova Var. Controllata | X_n | w | dimensione di 'm' bit |
| +3 | Var. Controllata Prec. | X_{n-1} | * | |
| +4 | Variabile di Riferimento | W_n | w | dimensione di 'm' bit |
| +5 | Var. di Riferim. Preced. | W_{n-1} | * | |
| +6 | Fattore Proporzionale | F_p | w | * 256 |
| +7 | Fattore Integrale | F_i | w | * 256 |
| +8 | Fattore Derivativo | F_d | w | * 256 |
| +9 | Banda Morta | D_r | w | |
| +10 | Partenza a Freddo Y | Y_s | w | Valore iniziale di Y_n |
| +11 | Precisione in Bit | m | w | m = 8, 12 o 16 bit |
| +12 | Area di Lavoro | Z_s | * | |

* Questi valori sono gestiti dall'istruzione PID.

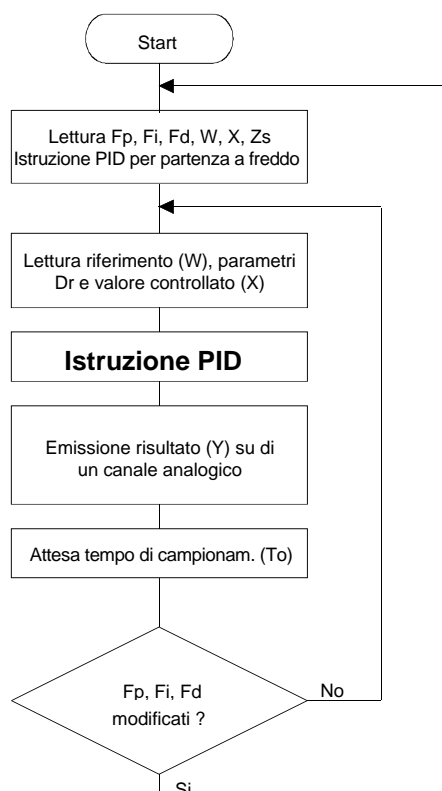
w Questi valori devono essere scritti nel registro dal programma utente.

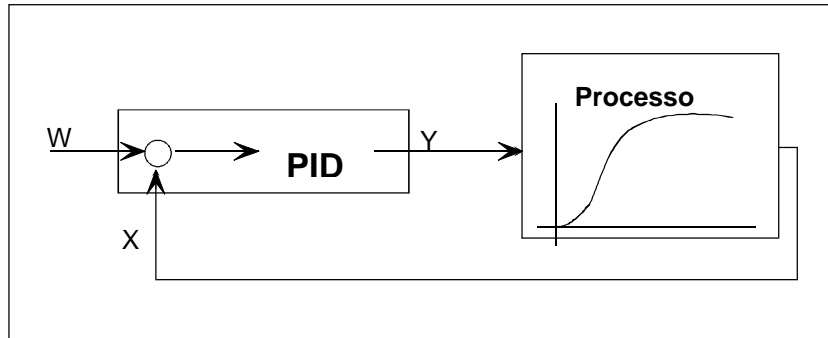
Utilizzo **PID reg ; reg indica l'indirizzo inferiore dei 13 Registri
; (R 0-4083)**

Esempio PID R 1000 ; Utilizza R 1000-1012 per i dati di controllo PID

Flag Invariati

Uso pratico Un tipico ciclo di controllo PID deve essere organizzato come segue:





Nuovo Risultato Y_n :

Questo è il risultato reale per controllare il processo determinato dal programma di sistema partendo dalla seguente equazione, con $Z_s = Z_s + (W_n - X_n)$:

$$Y_n = \frac{F_p}{256} * \{ (W_n - X_n) + \frac{F_i}{256} * Z_s + \frac{F_d}{256} * [(W_n - W_{n-1}) - (X_n - X_{n-1})] \}$$

Se il risultato supera la precisione dichiarata in termini di bit, verrà limitato al suo valore massimo (m bit) o, in caso di risultato negativo, verrà impostato al valore 0.

Risultato Precedente Y_{n-1} :

É il vecchio risultato determinato dall'operazione precedente.

Variabile Controllata X_n :

La variabile controllata X_n viene letta dal processo e scritta nel registro (R+2) dal programma utente. La variabile controllata deve essere al massimo di 'm' bit.

Variabile Controllata Precedente X_{n-1} :

É la vecchia variabile controllata, utilizzata nella precedente operazione calcolo.

Variabile di Riferimento W_n :

Il riferimento (setpoint) viene scritto nel registro (R+4) dal programma utente. Deve essere al massimo di 'm' bit.

Variabile di Riferimento Precedente W_{n-1} :

É il vecchio riferimento utilizzato nella precedente operazione aritmetica.

Fattore Proporzionale F_p :

Questo fattore determina la caratteristica proporzionale (amplificazione) del regolatore e viene scritto nel registro (R+6) dal programma utente. Durante il calcolo, vengono utilizzati solo i 16 bit meno significativi (0..65535). Il fattore proporzionale viene determinato nel modo seguente:

$$F_p = \frac{1}{X_p} * 256 \quad \text{con } X_p: \text{ Banda Proporzionale}$$

Nota: Per introdurre una banda proporzionale del 5%, il fattore F_p deve essere impostato come:

$$(1 / 0.05) * 256 = 5120$$

Una partenza a freddo del PID deve essere eseguita dopo una modifica di F_p o F_i

Fattore integrale F_i :

Questo fattore determina la caratteristica integrale del regolatore e viene scritto nel registro (R+7) dal programma utente.

Durante il calcolo, vengono utilizzati solo i 16 bit meno significativi (0..65535).

Il fattore Integrale viene determinato nel modo seguente:

$$F_i = \frac{T_0}{T_i} * 256 \quad \text{con} \quad \begin{array}{l} T_0: \text{Tempo di Campionamento istruzione PID} \\ T_i: \text{Tempo Integrale} \end{array}$$

Una partenza a freddo del PID deve essere eseguita dopo una modifica di F_p o F_i

Fattore Derivativo F_d :

Questo fattore determina la caratteristica derivativa del regolatore e viene scritto nel registro (R+8) dal programma utente.

Durante il calcolo, vengono utilizzati solo i 16 bit meno significativi (0..65535).

Il fattore Derivativo viene determinato nel modo seguente:

$$F_d = \frac{T_d}{T_0} * 256 \quad \text{con} \quad \begin{array}{l} T_0: \text{Tempo di Campionamento istruzione PID} \\ T_d: \text{Tempo Derivativo} \end{array}$$

Banda Morta D_i :

La banda morta, definisce l'intervallo in cui possono avvenire le variazioni della variabile controllata senza causare modifiche alla variabile Risultato (Y_n).

Partenza a Freddo Y_s :

È il valore utilizzato come valore di partenza per Y_n dal programma sistema. Non appena il programma utente scrive un valore diverso da 0 nel registro di partenza a freddo, viene eseguito il calcolo relativo:

$$\begin{array}{lcl} Y_n & = & Y_s \\ Y_{n-1} & = & Y_s \\ Z_s & = & [(Y_s * 256/F_p) - (W_n - X_n)] * 256/F_i \\ W_{n-1} & = & W_n \\ X_{n-1} & = & X_n \end{array}$$

Il valore di Y_s dopo essere stato utilizzato la prima volta, viene azzerato automaticamente dal programma di sistema e non verrà più riutilizzato.

Per una partenza a freddo con un valore in uscita pari a 0, il registro Y_s deve essere impostato a -1.

Quando $F_i = 0$, il valore Y_n non può essere inizializzato con una partenza a freddo. E' comunque consigliabile una partenza a freddo per inizializzare il registro di lavoro. In questo caso, il valore Y_s viene ignorato, il registro Z_s viene impostato a 0 e Y_n assume il valore della parte proporzionale dell'algorithm.

Nota: Il passaggio dal controllo manuale al controllo automatico è una tipica applicazione di calcolo della partenza a freddo. Per ottenere un passaggio dolce, Y_s può essere impostato allo stesso valore della variabile controllata corrente (Y_n).

Risoluzione m:

I valori massimi di X , W , Y_n e Y_s sono determinati dalla risoluzione.

Se $m = 8$: nel calcolo vengono utilizzati 8 bit (0..255)

Se $m = 12$: nel calcolo vengono utilizzati 12 bit (0..4095)

Se $m = 16$: nel calcolo vengono utilizzati 16 bit (0..65535)

La risoluzione viene definita normalmente dal modulo analogico utilizzato per l'emissione della variabile risultato. Se la risoluzione di ingresso e di uscita non sono uguali, il valore Y_n deve essere adattato dopo l'istruzione PID.

Tempo di Campionamento:

Il tempo di campionamento T_o deve essere fornito da un temporizzatore esterno all'istruzione PID

In pratica: $T_o \approx 0,1$ costante di tempo del processo (T_o deve essere almeno 80 ms)

Capacità di Calcolo:

Il registro di lavoro Z_s ha una capacità massima di 2^{31} .

Quando si utilizzano valori a 16 bit ($m = 16$), si può verificare un overflow; in questo caso il PID non opererà correttamente.

Per evitare questo problema, il fattore F_p deve essere ≥ 2 se $m = 16$ (quando $m = 8$ o 12 non sussistono problemi).

TEST TEST HARDWARE

Descrizione Verifica l'hardware selezionato del PCD in modo condizionato o incondizionato.

Se un qualsiasi test fallisce, il test stesso viene interrotto e l'ACCU viene impostato a livello basso (L).

Se tutti i test selezionati terminano con successo, l'ACCU viene impostato al livello Alto (H).

I singoli test possono essere selezionati nel modo seguente:

| Valore | Bit Numero | Descrizione del Test |
|--------|------------|---|
| | 11 | |
| 400 | 10 | Perdita Memoria Comune (Public Memory) |
| 200 | 9 | Memoria Estesa Danneggiata |
| 100 | 8 | Memoria RAM di sistema |
| | 7 | |
| 40 | 6 | Checksum Firmware di sistema |
| 20 | 5 | Porte (canali) seriali |
| 10 | 4 | Orologio hardware |
| | 3 | |
| 4 | 2 | Checksum Programma utente/Testi |
| 2 | 1 | RAM Programma utente/Testi |
| 1 | 0 | RAM Comune (Public) (F, T, C, R, Mailbox) |

I test vengono effettuati quando il bit relativo è impostato (1). I test 0 e 5 vengono eseguiti se la CPU controllata è l'unica in stato di RUN; Se ci sono altre CPU in stato di Run, tali test NON vengono eseguiti.

Nota: Alcuni test sono molto lenti e non dovrebbero essere eseguiti durante il normale funzionamento del PCD, ma solo all'avviamento del sistema o nei periodi di inattività.

Non è possibile fornire l'operando come parametro di un Blocco Funzione.

Utilizzo

| | | |
|-------------|--------------------|---|
| TEST | [cc] numero | ; cc = codice condizione (H L P N Z E) ; numero = numero identificativo del test |
|-------------|--------------------|---|

Esempio

TEST 50 ; Testa incondizionatamente il Checksum Firmware Sistema (40)
; + l'Orologio Hardware (10)

TEST L 4 ; Se ACCU = L (0), allora testa checksum Programma utente/testi

Flag

L'ACCU viene impostato a livello alto (H) se tutti i test vengono superati, viene impostato a livello basso (L) se uno qualsiasi dei test fallisce.

Test RAM Comune (Public) (Valore = 1)

Esegue un test della memoria RAM che contiene gli F/R/T/C mediante un'operazione di salvataggio-scrittura-lettura-confronto-ripristino. Questo test non viene eseguito se, in un sistema multiprocessore, vi è un'altra CPU in stato di RUN.

| ACCU | Flag Errore | Descrizione |
|------|-------------|-------------------------------------|
| 0 | 1 | Un'altra CPU è in modo RUN |
| 0 | 0 | Rilevato un errore sulla RAM comune |
| 1 | x | RAM Comune OK |

Test RAM Programma Utente/Testi (Valore = 2)

Esegue un test della memoria RAM che contiene il Programma Utente/Testi mediante un'operazione di salvataggio-scrittura-lettura-confronto-ripristino. Se la memoria in oggetto non è di tipo RAM o se la RAM è protetta da scrittura, viene eseguito il test Checksum Programma Utente/Testi.

| ACCU | Flag Errore | Descrizione |
|------|-------------|-----------------------------------|
| 0 | 1 | Intestazione Programma non valida |
| 0 | 0 | RAM guasta |
| 1 | 0 | RAM OK |

Test Checksum Programma Utente/Testi (Valore = 4)

Calcola la somma dell'intera area Programma + Testi e confronta il risultato con il checksum presente nell'intestazione.

| ACCU | Flag Errore | Descrizione |
|------|-------------|-----------------------------------|
| 0 | 1 | Intestazione Programma non valida |
| 0 | 0 | Checksum non OK |
| 1 | 0 | Checksum OK |

Test Orologio Hardware (Valore = 10)

Verifica l'esistenza dell'Orologio (RTC) ed esegue un test per accertare se l'orologio si incrementa in modo corretto. Qualsiasi altra CPU che tenti di accedere all'Orologio durante l'esecuzione di questo test verrà bloccata per un periodo massimo di 15 ms.

| ACCU | Flag Errore | Descrizione |
|------|-------------|-----------------------|
| 0 | 1 | Orologio non presente |
| 0 | 0 | Orologio guasto |
| 1 | 0 | Orologio OK |

Test Porte (canali) Seriali (Valore = 20)

Esegue un test delle porte seriali inizializzando la porta in oggetto in modalità loopback locale, trasmettendo una sequenza di test e quindi verificando la ricezione di tale sequenza.

Se è stata assegnata una qualsiasi delle altre porte seriali, questo test non viene effettuato.

| ACCU | Flag Errore | Descrizione |
|------|-------------|-------------------------------------|
| 0 | 1 | Una delle porte seriali è assegnata |
| 0 | 0 | La porta non è OK |
| 1 | 0 | Le porte seriali sono OK |

Test Checksum Firmware di Sistema (Valore = 40)

Viene controllata la memoria EPROM contenente il firmware di sistema

| ACCU | Flag Errore | Descrizione |
|------|-------------|---------------------|
| 0 | 0 | Checksum non valido |
| 1 | 0 | Cheksum OK |

Test Memoria RAM di Sistema (Valore = 100)

Viene controllata la memoria RAM di sistema

| ACCU | Flag Errore | Descrizione |
|------|-------------|---------------------|
| 0 | 0 | Checksum non valido |
| 1 | 0 | Cheksum OK |

Test Danneggiamento Memoria Estesa (Valore = 200)

Viene effettuato un test della memoria estesa all'accensione e, in caso risulti danneggiata, tale anomalia viene indicata utilizzando un flag interno.

| ACCU | Flag Errore | Descrizione |
|------|-------------|----------------------------|
| 0 | 0 | Memoria estesa danneggiata |
| 1 | 0 | Nessun danno rilevato |

Test Perdita memoria Comune (Public) (Valore = 400)

Se la sequenza di test memorizzata nella mailbox risulta non valida quando viene verificata durante la routine di partenza, il sistema presuppone che la memoria RAM comune (public) sia stata danneggiata (peridta dati) durante lo spegnimento, a causa di un non sufficiente grado di carica delle batterie.

| ACCU | Flag Errore | Descrizione |
|------|-------------|-------------------------------------|
| 0 | 0 | Memoria comune (public) danneggiata |
| 1 | 0 | Nessun danno rilevato |

DIAG LETTURA DIAGNOSTICA (Read XOB Diagnostic)

Descrizione Compila un blocco di 12 registri con le informazioni diagnostiche relative all'ultimo Blocco ad Organizzazione Esclusiva (XOB) eseguito. L'operando è il numero di registro più basso nel blocco di 12 Registri. L'istruzione DIAG viene normalmente utilizzata all'interno di un XOB.

Uso del Blocco Registri:

| Registro | | |
|-----------------|---------------------------------|------------------------------------|
| 0 | Numero XOB | N° ultimo XOB (o attuale) |
| +1 | Riga di Programma | Riga di progr. con chiamata XOB |
| +2 | Registro Indice | Valore Reg. Indice alla chiamata |
| +3 | Riga di Programma COB | Riga progr. con chiamata Livello 0 |
| +4 | Riga Progr. 1° Liv. di Nidific. | Riga progr. con chiamata Livello 1 |
| +5 | Riga Progr. 2° Liv. di Nidific. | Riga progr. con chiamata Livello 2 |
| +6 | Riga Progr. 3° Liv. di Nidific. | Riga progr. con chiamata Livello 3 |
| +7 | Riga Progr. 4° Liv. di Nidific. | Riga progr. con chiamata Livello 4 |
| +8 | Riga Progr. 5° Liv. di Nidific. | Riga progr. con chiamata Livello 5 |
| +9 | Riga Progr. 6° Liv. di Nidific. | Riga progr. con chiamata Livello 6 |
| +10 | Riga Progr. 7° Liv. di Nidific. | Riga progr. con chiamata Livello 7 |
| +11 | Non Usato | Riservato |

Il numero di righe di programma delle chiamate al blocco (informazioni sul livello di nidificazione) fornisce la riga di programma in cui è avvenuta la chiamata precedente (CFB, CPB). Da questa si può stabilire esattamente dove era il programma quando è stato eseguito l'XOB.

Nota: Le informazioni più importanti sono contenute nei Registri R e R+1.

Non è possibile fornire l'operando come parametro di un Blocco Funzione.

Utilizzo **DIAG reg ; R 0-4084, Indirizzo inferiore dei 12 Registri**

Esempio DIAG R 1000 ; Memorizza le info diagnostiche nei Registri 1000-1011

Flag Invariati

Vedere anche Manuale Utente

Uso pratico Deve essere stampato l'indirizzo della riga in cui si è verificato un errore.

```

XOB            13
DIAG        R 1000
STXT           1
                 100
TEXT 100 "$D $ H : "
                "Flag Errore attivato all'indirizzo $R1001
                <CR><LF> "
EXOB
    
```

SYSRD LETTURA DATI SISTEMA (System Read)

Descrizione Legge vari parametri legati al sistema PCD quali: tipo Dispositivo PCD, tipo CPU, versione Firmware, nome Programma Utente, parametri S-Bus, ...

Utilizzo

| | | |
|--------------|---------------------------|--|
| SYSRD | Funzione Risultato | ; Codice funzione, codice K, R 0..4095 ; Risultato della lettura, R 0..4095 |
|--------------|---------------------------|--|

Funzione

K x o R x: Costante o Registro contenente un codice funzione. Questa istruzione può essere sia diretta, indicando una Costante di identificazione del codice funzione, che indiretta, servendosi di un Registro. Essa permette all'utente di accedere a informazioni utili sul sistema attraverso il programma utente.

Risultato

R 0..4095 Registro in cui porre il risultato o primo di 2 registri (vedere codice 5400) oppure primo di un blocco di registri (vedere codici 65xx)

Esempio

SYSRD K 5000 ; Legge il tipo di PCD in formato ASCII
R 20 ; e pone il risultato in R 20

Flag

Vedere anche SYSWR

Codici funzione

| Codice | Descrizione della funzione | Risultato | | |
|--------------------------------------|---|---|-----|----------|
| 2000 2001 2002 2003 2004 | Legge EEPROM Utente Registro 0 Registro 1 Registro 2 Registro 3 Registro 4 | Valore presente nella EEPROM | | |
| 5000 5010 | Legge Tipo Apparecc. in ASCII in decimale | ASCII | Dec | Tipo |
| | | " D1" | 1 | PCD1 |
| | | " D2" | 2 | PCD2 |
| | | " D4" | 4 | PCD4 |
| | | " D6" | 6 | PCD6 |
| 5100 5110 | Legge Tipo CPU in ASCII in decimale | ASCII | Dec | Tipo |
| | | " M1_" | 10 | PCD2.M1 |
| | | " M1_" | 10 | PCD2.M1 |
| | | " M11" | 11 | PCD4.M11 |
| | | " M12" | 12 | PCD4.M12 |
| | | " M14" | 14 | PCD4.M14 |
| | | " M24" | 24 | PCD4.M24 |
| | | " M34" | 34 | PCD4.M34 |
| | | " M44" | 44 | PCD4.M44 |
| | | " M1_" | 10 | PCD6.M1 |
| | | " M2_" | 20 | PCD6.M2 |
| | | " M54" | 54 | PCD6.M5 |
| 5200 5210 | Legge versione Firmware in ASCII in dec | Esempi di risposte valide: " \$4C", " 004", " X41" | | |
| | | Es: 5 dec per Versione 005 -1 dec per ogni '\$', 'X', 'β' | | |
| 5400 | Legge nome Programma Utente in ASCII Il nome del programma utente contiene sempre 8 caratteri ASCII | R x contiene i primi 4 byte del nome programma in ASCII R x+1 contiene i 4 byte inferiori del nome programma in ASCII | | |
| 6000 | Legge numero stazione S-Bus | Esempi di risultato: 2 Numero stazione = 2 -1 Stazione non configurata | | |
| 6010 6020 | Legge ritardo TN PGU S-Bus Legge ritardo TS PGU S-Bus | Esempi di risultato: 10 Ritardo in mS -1 S-Bus non configurato | | |
| 6030 | Legge Time-out PGU S-Bus | | | |
| 6040 | Legge baudrate PGU S-Bus | Esempi di risultato: 9600bps -1 S-Bus non configurato | | |

| Codice | Descrizione funzione | Risultato | |
|---------------|--|---|-----|
| 6050 | Legge modalità PGU S-Bus | Stato | Dec |
| | | BREAK senza modem | 0 |
| | | PARITA' senza modem | 1 |
| | | BREAK con modem | 10 |
| | | PARITA' con modem | 11 |
| | S-Bus non configurato | -1 | |
| 6060 | Legge numero porta PGU S-Bus | Esempi di risultato: 1 porta PGU S-Bus configurata sulla porta 1 -1 S-Bus non configurato | |
| 6070 | Legge livello S-Bus | Stato | Dec |
| | | S-Bus Livello 1 (ridotto) | 1 |
| | | S-Bus Livello 2 (completo) | 2 |
| | | S-Bus non configurato | -1 |
| 6080 | Legge gestore porta attuale PGU (S-Bus o protocollo P8) | CPU 0 | 0 |
| | | CPU 1 | 1 |
| 6100 | Legge byte di stato modem Legge lo stato corrente del collegamento modem. Queste informazioni indicano all'utente lo stato del modem durante la procedura di inizializzazione. Risultati: 2 PCD in attesa del collegamento modem. 6..39 Il PCD è in fase di inizializzazione del modem. 40 Riassegnazione porta seriale per Modo SS1/SS0. 45..49 Il collegamento con il modem è stato interrotto. Rappresenta uno stato intermedio prima della reinizializzazione del modem. 50 Il collegamento è corretto ed il PCD è online in modalità SS0/1. | | |
| 6500 | Legge stringa tipo di modem | | |
| 6510 | Legge stringa di reset modem | | |
| 6520 | Legge stringa di inizializzazione modem | | |
| | Legge la stringa modem indicata dall'intestazione estesa del programma utente e la pone nel blocco di registri a partire dall'indirizzo base R x | | |
| 7000 | Legge contatore sistema | 0.. 2.147.483.647 | |
| | Un Contatore di Sistema interno viene incrementato ogni millisecondo. Questo Contatore di Sistema viene azzerato ad ogni accensione quindi, per esempio, una "Ripartenza a Freddo" non è in grado di influenzarlo. Il periodo massimo di questo contatore è: 24 giorni 20 ore 31 min 23 sec 647 ms Per un esempio esplicativo, vedere l'istruzione SYSCMP. | | |

SYSWR SCRITTURA DATI SISTEMA (System Write)

Descrizione Rappresenta il complemento dell'istruzione SYSRD e permette di modificare alcune informazioni del sistema o di inizializzare delle funzioni di sistema mediante il programma utente.

Utilizzo

| | | | |
|--------------|-----------------|----------|---|
| SYSWR | Funzione | ; | Codice funzione, codice K, R 0..4095 |
| | Valore | ; | Valore da scrivere |

Funzione
K x o R x: Costante o Registro contenente un codice funzione. Questa istruzione può essere sia diretta, indicando una Costante di identificazione del codice funzione, che indiretta, servendosi di un Registro. Essa permette all'utente di accedere a informazioni utili sul sistema attraverso il programma utente.

Valore
K y Valore da scrivere
R 0..4095 Registro contenente il valore da scrivere

Esempio SYSWR K 4014 ; Inizializza l'XOB 14 con una frequenza
 K 10 ; di 10 ms

Flag Il flag **Errore (E)** viene impostato se il codice funzione indicato non esiste.

Vedere anche SYSRD

| Codice | Descrizione funzione |
|--------------------------------|---|
| 1000 | <p>Watchdog di Sistema (solo PCD1) Valori ammessi per K y o R y: 0 Disattiva il WDOG 1 Attiva il WDOG ed esegue una ripartenza a freddo se la funzione non viene ripetuta almeno ogni 200 ms 2 Attiva il WDOG e chiama l'XOB 0 prima di eseguire una ripartenza a freddo se la funzione non viene ripetuta almeno ogni 200 ms. Non appena il watchdog viene attivato, l'istruzione deve essere ripetuta almeno ogni 200 ms. Un XOB 0 attivato dal watchdog si distingue da un XOB 0 relativo ad una caduta di tensione dal differente messaggio inserito nella tabella storica. Se attivato da WDOG viene infatti scritto il messaggio "XOB0 WDOG START" mentre in caso di XOB 0 conseguente a una caduta di tensione viene scritto il messaggio "XOB 0 START EXEC".</p> |
| 2000 2100 | <p>Scrittura registri non volatili in EEPROM I SAIA PCD sono equipaggiati con dei registri non-volatili residenti in memoria EEPROM. Il PCD1 ha 5 registri utenti EEPROM 0 .. 4, indirizzabili dai codici funzione 2000 - 2004. Il PCD2 (dalla versione hardware H) ha 50 registri utente EEPROM 0 .. 49, indirizzabili dai codici funzione 2000 - 2049. Il PCD6.M3 ha 101 registri utente EEPROM 0 .. 100, indirizzabili dai codici funzione 2000 - 2100. Valori ammessi per K y o R y: R y: Registro sorgente contenente i valori da scrivere sulla EEPROM. <u>Attenzione:</u> La memoria EEPROM può essere riscritta dall'utente per max. 100.000 volte. Si raccomanda quindi di non eseguire frequentemente questa istruzione all'interno del programma utente. L'esecuzione di SYSWR richiede 20 ms perciò essa NON DEVE essere utilizzata all'interno dell'XOB 0. Essa serve per configurare valori legati all'inizializzazione del sistema.</p> |

| Codice | Descrizione funzione |
|--------------|---|
| 4000 | <p>Impostazione Limite di Overflow XOB</p> <p>Gli XOB 14/15/17/18/19/20/25 operano tutti secondo un meccanismo basato sul principio delle code. Se un XOB è attivo, l'XOB in attesa viene posto in una coda avente una dimensione massima di 127 inserimenti <i>per ogni</i> XOB. Se questo limite viene superato allora viene richiamato l'XOB 7 e la coda viene azzerata.</p> <p>Nella tabella storica (History table) viene inserito il messaggio "SYSTEM OVERLOAD". Il limite di 127 inserimenti può, a volte, essere troppo grande per applicazioni real time, così ora è possibile, grazie a questa istruzione, definire un limite diverso. Il limite definito sarà comune a tutti gli XOB che possono essere accodati.</p> <p>Valori ammessi per R y o K y: 0 .. 127</p> |
| 4005 4013 | <p>Abilitazione/Disabilitazione XOB 5/13</p> <p>Consente di abilitare o disabilitare gli XOB 5/13. In alcuni casi, l'esecuzione immediata di questi XOB dopo il loro richiamo può creare delle complicazioni nell'esecuzione del programma utente. Per questa ragione è ora possibile disabilitarli. Se un XOB viene richiamato una o più volte mentre è disabilitato, esso verrà richiamato ancora non appena verrà riabilitato.</p> <p>Codice funzione: 4005 XOB 5 4013 XOB 13</p> <p>Valori ammessi per R y o K y:</p> <p>0 Disabilita l'XOB 1 Abilita l'XOB 2 Azzerà il flag Error nel COB corrente e nell'XOB attivo (solo per K 4013)</p> |
| 4014 4015 | <p>Installazione XOB 14/15</p> <p>Consente di configurare gli XOB periodici con la frequenza definita in Ky od Ry. E' possibile configurare due XOB periodici con una frequenza che va da 10 mS a 1000 S.</p> <p>Il valore contenuto in Ky od Ry è espresso in mS e se pari a zero l'XOB viene disattivato. Questa istruzione può essere eseguita in un qualsiasi momento. Se un XOB è già attivo quando un secondo XOB viene richiamato, quest'ultimo verrà accodato finchè non vi sono più XOB attivi e può quindi essere eseguito. Gli XOB possono essere eseguiti solo se la CPU è in modalità RUN oppure RUN CONDIZIONATO.</p> <p>Codice funzione 4014 Configura l'XOB 14 4015 Configura l'XOB 15</p> <p>Valori ammessi per R y o K y: 10 .. 1000000</p> |

| Codice | Descrizione funzione |
|----------------------|---|
| 4017 4018 4019 | <p>Esecuzione XOB 17/18/19 Esegue l'XOB specificato in R x o K x sulla CPU specificata in K y od R y. Gli XOB 17/18/19 sono XOB definiti dall'utente che possono essere richiamati tramite l'S-BUS od il programma utente. Gli XOB possono essere eseguiti solo se la CPU è in modalità RUN o RUN CONDIZIONATO.</p> <p>Codice funzione : 4017 Esegue l'XOB 17 4018 Esegue l'XOB 18 4019 Esegue l'XOB 19</p> <p>Valori ammessi per R y o K y: 0 .. 6 CPU sulla quale verrà eseguito l'XOB 7 Esegue l'XOB sulla sola CPU locale 8 Esegue l'XOB su tutte le CPU.</p> |
| 6000 | <p>Scrittura Numero Stazione S-Bus Modifica il numero di stazione S-BUS in base al valore contenuto in K y od R y. Questa istruzione può operare con programmi utente memorizzati sia su EPROM che su RAM. Valori ammessi per K y o R y: 0 .. 254</p> |
| 7000 | <p>Conversione FFP-IEEE Consente di convertire valori in virgola mobile dal formato FFP (Fast Floating Point) al formato IEEE e viceversa. Lo standard FFP è utilizzato da TUTTE le istruzioni in virgola mobile dei SAIA PCD. Quando un valore viene convertito in formato IEEE esso non può più essere utilizzato con alcuna istruzione in virgola mobile.</p> <p>Codice funzione 7000 da formato FFP a formato IEEE 7001 da formato IEEE e formato FFP</p> <p>Valori ammessi per R y: R y contiene il valore da convertire. Il risultato verrà posto nello stesso registro.</p> |

SYSCMP COMPARAZIONE DATI SISTEMA (System Compare)

Descrizione L'istruzione SYSCMP consente di trasformare un qualsiasi registro in uno pseudo-Temporizzatore. Il suo compito è quello di confrontare la somma del primo e del secondo operando con il Contatore di Sistema e quindi impostare l'ACCU in base al risultato.

Se il risultato dell'addizione è maggiore del Contatore di Sistema, l'ACCU verrà impostato al livello Alto (H). Se il risultato dell'addizione è inferiore od uguale al Contatore di Sistema allora l'ACCU verrà impostato al livello Basso (L).

Il vantaggio offerto da questa istruzione in combinazione con l'istruzione SYSRD K 7000 è quello di rendere possibile l'uso di Temporizzatori con una risoluzione pari ad 1 mS. E' anche possibile misurare l'intervallo tra due eventi con una risoluzione di 1 mS.

Utilizzo

| | | |
|---------------|------------------|---------------------------------|
| SYSCMP | R x | ; R 0..4095 |
| | K y o R y | ; K 0..16383 o R 0..4095 |

Esempio

SYSCMP R 100 ; Confronta il contenuto del Registro 100 + 1500 con il
K 1500 ; Contatore di Sistema ed imposta l'ACCU in conseguenza

SYSCMP R 100 ; Confronta il contenuto di R100 + R101 con il
R 101 ; Contatore di Sistema ed imposta l'ACCU in conseguenza

Flag

ACCU

Vedere anche

SYSRD

Uso pratico

Programmazione di un temporizzatore ad alta risoluzione (1ms) con le istruzioni SYSRD e SYSCMP

Questo esempio illustra come programmare un temporizzatore ad alta risoluzione (1ms) con le istruzioni SYSRD e SYSCMP.

```

COB          0
              0
...
LD          R 100 ; Carica il periodo di attesa (1500)
              1500 ; in R 100

SYSRD       K 7000 ; Legge il Contatore di Sistema in R 101
              R 101

wait:       SYSCMP R 101 ; Confronta Contatore di Sistema e R100 + R101
              R 100 ; e imposta l'ACCU in base al risultato
JR          H wait ; Se ACCU = Alto (1) allora cicla (loop)
...
ECOB

```


ALGO USCITA ANALOGICA (Analogue Output)

Descrizione Emette verso il modulo analogico **PCA2.W1x** un valore binario a 12 bit prelevato dal registro specificato.
 Il primo operando rappresenta il Registro da emettere
 Il secondo operando contiene sia il numero del canale D/A che l'indirizzo di base del modulo.

Se il secondo operando viene fornito come parametro di un FB, il numero del canale D/A e l'indirizzo di base devono essere forniti sulla stessa riga.

Utilizzo

| | | |
|----------------|---------------------|--|
| ALGO[X] | registro (i) | ; Registro sorgente R 0-4095 |
| c base | | ; c = canale 0-3, base = 0-8176 |

Esempio

ALGO R 100 ; Emette il valore contenuto in R 100
 3 128 ; sul canale 3 del modulo all'indirizzo di base 128

Flag

Invariati

Vedere anche

ALGI

Nota

Questa istruzione non può essere utilizzata per i moduli PCD4.Wxxx e PCD6.Wxxx (consultare i rispettivi manuali hardware).

**STHS INIZIO LENTO SU STATO ALTO
(Start High Slow)**

Descrizione L'ACCU viene impostato con lo stato logico dell'elemento indirizzato, solitamente un Ingresso. É analoga all'istruzione STH, ad eccezione del fatto che la temporizzazione sul bus I/O PCD è più lenta, e quindi adatta per moduli di I/O lenti. La velocità di esecuzione del programma non viene influenzata in termini significativi.

É consigliabile utilizzare questa istruzione per accedere ai moduli Analogici PCA2.W2x/W3x.

| | |
|-----------------|--|
| Utilizzo | STHS[X] elemento (i) ; I 0-8191, O 0-8191, F 0-8191 |
|-----------------|--|

Esempio STHS I 25

Flag L'ACCU viene impostato allo stato logico dell'elemento specificato.

Vedere anche OUTS, STH, Istruzioni a Livello Bit

OUTS IMPOSTA ELEMENTO LENTO DA ACCUMULATORE (Set Element From Accumulator Slow)

Descrizione L'elemento specificato, di solito una Uscita, viene impostato con lo stato dell'ACCU. È analoga all'istruzione OUT, ad eccezione del fatto che la temporizzazione sul bus I/O PCD è più lenta, e quindi adatta per moduli di I/O lenti. La velocità di esecuzione del programma non viene influenzata in termini significativi.

È consigliabile utilizzare questa istruzione per accedere ai moduli Analogici PCA2.W2x/W3x.

Utilizzo

OUTS[X] elemento (i) ; I 0-8191, O 0-8191, F 0-8191

Esempio OUTS O 32

Flag L'ACCU viene impostato allo stato logico dell'elemento specificato.

Vedere anche OUT, OUTD, STHS

Uso pratico E' necessario leggere il valore analogico del canale 0, deve essere letto da un PCA2.W2x (indirizzo base 96) e memorizzarlo nel Registro 100.

Dopo aver effettuato la conversione con l'istruzione OUTS, si possono leggere 8 bit binari a partire dall'indirizzo base del modulo + 8 (=104).

```

COB          0
              0
...
ACC          H          ; Verifica che l'ACCU sia Alto
OUTS       96       ; Selezione il canale analogico
...          ; Attendi ± 100 ms *)
CPB          RD_VAL    ; Chiama il blocco di programma RD_VAL
...
ECOB
PB           RD_VAL
BITIR        8          ; Legge 8 bit binari in ordine inverso
              I 104     ; dagli indirizzi 104 .. 111
              R 100     ; nel Register 100
EPB
    
```

*) Il modulo analogico PCA2.W2x ha un tempo di conversione ≤ 100 ms. Questa funzione di attesa può essere realizzata inserendo varie istruzioni NOP consecutive (il numero di istruzioni NOP dipende dal tipo di CPU).

13. Tabella Storica (Hystory List)

Qui di seguito viene riportata una dettagliata descrizione di tutti gli errori che possono essere inseriti nella TABELLA STORICA (History List) o REGISTRO DELLE CAUSE DI BLOCCO (Halt Reason Register). Ogni volta che la CPU rileva un errore, all'interno della TABELLA STORICA viene inserito un messaggio che può essere visualizzato con il debugger PG3 utilizzando il comando "Display History" (Visualizza Tabella Storica).

Errori che determinano l'esecuzione di un XOB (se programmato)

| Messaggio | HALT | XOB | Sistema *) | Significato |
|------------------|------|-----|------------|---|
| XOB START EXEC | N | 0 | Tutti | E' stato avviato l'XOB 0 |
| XOB 0 EXECUTED | N | 0 | Tutti | L'XOB 0 è stato terminato durante una caduta di tensione |
| XOB 0 WDOG START | N | 0 | 1 | Attivazione watchdog di sistema |
| EXTERN PWR FAIL | N | 1 | 2, 5, 6 | Caduta di tensione sul rack di estensione |
| PARITY FAILURE | N | 4 | 5, 6 | Errore sul bus principale PCD6 |
| SYSTEM OVERLOAD | N | 7 | Tutti | Il meccanismo di gestione delle code per gli XOB livello 3 è sovraccarico |
| ILLEGAL OP CODE | N | 8 | Tutti | Esegue l'XOB 8 quindi una ripartenza a freddo dopo la rilevazione di un'istruzione non valida |
| >32 ST/TR ACTIVE | N | 9 | Tutti | Troppi task GRAFTEC attivi |
| >7 CALL LEVELS | N | 10 | Tutti | Superamento livello massimo di nidificazione PB/FB |

I seguenti errori hanno un posizionamento fisso all'interno della tabella storica e sono corredati con un contatore di errori

| Message | HALT | XOB | Sistema | Significato |
|------------------|------|-----|------------|--|
| BATT FAIL 000 | N | 2 | 2, 4, 5, 6 | La batteria è scarica |
| IO QUIT FAIL 000 | N | 5 | 4, 5, 6 | Tentato accesso a una locazione di I/O non installata |
| IR OVERFLOW 000 | N | 12 | 2, 4, 5, 6 | Registro Indice incrementato oltre il valore max. 8191 |
| ERROR FLAG 000 | N | 13 | 2, 4, 5, 6 | Flag Errore impostato |

*) Sistema:

- 1: PCD1
- 2: PCD2
- 4: PCD4
- 5: PCD6.M540
- 6: PCD6.M1.., M2..

Errori rilevati all'accensione del sistema

Tutti i seguenti errori vengono rilevati all'ACCENSIONE del PCD

| Messaggio | HALT | Sistema | Significato |
|------------------|------|---------|---|
| RTC FAILURE | S | Tutti | Orologio hardware (RTC) presente ma malfunzionante |
| DUART HW ERROR | S | Tutti | Una DUART è difettosa |
| CHECKSUM FAIL | S | Tutti | Checksum errata per EPROM Programma Utente |
| BAD TXT/DB TABLE | S | Tutti | Causato da un'errata 'creazione tabella testo' all'accensione |
| TXT/DB HW ERROR | S | Tutti | Causato da un'errata 'creazione tabella testo' all'accensione |
| BAD MEM EXT INIT | S | Tutti | Causato da un'errata 'creazione tabella testo' per la memoria estesa all'accensione |
| USR MEM HW ERROR | S | 6 | Causato da un errato 'test programma utente' all'accensione |
| CPU SYNCH ERROR | S | 6 | Causato dal time-out della 2 ^{da} CPU |
| CPU FIRMWARE MIX | S | 4, 6 | Sistema multiprocessore equipaggiato con versioni incompatibili |

Errori di sistema gravi

Questi errori vengono scritti nel registro CAUSA DEL BLOCCO (Halt Reason) che può essere letto utilizzando il Debugger PG3 quando viene rilevata una condizione di HALT del sistema. Questi errori possono verificarsi all'ACCENSIONE oppure quando il PCD è in RUN.

| Messaggio | HALT | Sistema | Significato |
|------------------|------|---------|--|
| BUS QUIT FAILURE | S | Tutti | Il FW ha tentato di accedere ad un indirizzo non esistente |
| 68K INVALID OPC | S | Tutti | E' stata eseguita un'istruzione assembler 68000 non valida |
| 68K ADDR ERROR | S | Tutti | Tentativo di accesso ad un indirizzo dispari |
| ZERO DIVIDE | S | Tutti | Errore FW interno |
| 68K CHK INSTR | S | Tutti | ... |
| 68K TRAPV INSTR | S | Tutti | ... |
| PRIVILEGE VIOL | S | Tutti | ... |
| TRACE | S | Tutti | ... |
| ILLEGAL AUTO VEC | S | Tutti | ... |
| INTERRUPT ERROR | S | Tutti | ... |
| RESERVE INT | S | Tutti | ... |

Messaggi di errore legati alla programmazione o alla configurazione

I seguenti errori verranno rilevati all'ACCENSIONE del PCD

| Messaggio | HALT | Sistema | Significato |
|------------------|-------------|----------------|--|
| EVERYTHING IS OK | N | Tutti | Normale messaggio generato all'accensione |
| MODIFIED PROGRAM | N | Tutti | Il Programma utente è stato modificato dal Debugger PG3. Generato solo se il programma utente è protetto da scrittura. |
| CPU NUMBER > 6 | S | 6 | Il numero CPU impostato con i DIL switch non è valido |
| CPU 0 START FAIL | S | 4, 5, 6 | Solo CPU 1-6 : Nessuna CPU può essere posta in Run senza un programma caricato sulla CPU 0 |
| INIT-FAILURE | S | Tutti | Sono stati definiti più di 32 Step Iniziali GRAFTEC |
| HEADER FAIL | S | Tutti | Intestazione Programma Utente danneggiata |
| NO PROGRAM | S | Tutti | Nella CPU non vi è alcun programma da eseguire |
| MEM-EXT CORRUPT | S | Tutti | Mem. RAM estesa danneggiata |
| INVALID OPCODE | S | Tutti | E' stata caricata nella CPU una istruzione IL (AWL) non valida |
| MEDIA CORRUPTION | S/N | Tutti | Causato da anomalia della batteria |
| DOUBLE TIME BASE | S | Tutti | Istruzioni DEFTB e DEFTR all'interno dello stesso programma |
| BAD MODEM STRING | S | 1 | Stringa modem in EEPROM troppo lunga |

I seguenti messaggi verranno rilevati mentre il PCD è in RUN; quelli che causano un "HALT" del PCD verranno scritti anche nel registro CAUSA DEL BLOCCO.

| Messaggio | HALT | Sistema | Significato |
|------------------|-------------|----------------|--|
| BLOC NONEXISTENT | S | Tutti | Chiamata a PB, FB, SB, ST, TR non esistenti |
| HALTED BY LAN-2 | S | 4, 6 | Il coprocessore LAN-2 ha posto il PCD in HALT |
| LAN-2 WATCHDOG | N | 4, 6 | E' stato attivato il watchdog FW LAN-2 |
| HALT INSTRUCTION | S | Tutti | E' stata eseguita un'istruzione HALT comandata dall'utente |
| MANUAL HALT | S | 4, 5, 6 | La CPU è stata arrestata mediante lo switch HALT |
| HALTED BY CPU 0 | S | 4, 6 | Il coprocessore(i) è stato arrestato dalla CPU 0 |
| SBUS-PGU ERROR | N | Tutti | Assegnazione PGU S-BUS non valida su una porta |

Menù generale

Gamma di prestazioni dal PCD1 fino al PCD6



PCD1:
un concentrato di potenza



PCD2:
il modulare compatto dalle prestazioni eccezionali



Series xx7:
il PLC compatibile con SIMATIC® S7



SAIA®PCD
per l'automazione industriale



PCD4:
la flessibilità del PLC di media grandezza



PCD6:
il PLC di alta gamma per sistemi multiprocessore



PCD2.M250:
PLC + PC integrati in una unità industriale



Dai piccoli terminali di testi ai terminali «Touch Screen»



SAIA®PCD
per la Building automation

Strumenti di programmazione

Manuali

Informazioni generale