



KSE s.r.l.

Via Sabbionara, 14/D
36045 Alonte (Vicenza)

Tel: 0444 834077

Fax: 0444 436820

URL: <http://www.kse.it>

e-mail: kse@kse.it

PLC KSE V:1.10

MANUALE DI PROGRAMMAZIONE

25 febbraio 2010

1.00 GENERALITA'	5
1.01 CARATTERISTICHE TECNICHE DEL PLC CN KSE	5
1.02 VARIABILE DI PROCESSO	6
1.03 MAPPA DEGLI INGRESSI E DELLE USCITE	7
1.03.1 Organizzazione degli ingressi.....	8
1.03.2 Organizzazione delle uscite.....	9
1.04 ELABORAZIONE CICLICA DEL PROGRAMMA	10
1.05 CAUSE DI ARRESTO DEL PLC	11
1.06 INIZIALIZZAZIONE DELLA MEMORIA DEL PLC	12
1.07 CONSIDERAZIONI SUL PROGRAMMA UTENTE	13
1.08 REGISTRI DI LAVORO DELLE ISTRUZIONI	14
1.09 I FLAG DEGLI OPERANDI	15
1.10 CONSIDERAZIONE SUGLI OPERANDI DELLE ISTRUZIONI	16
1.10.1 INDIRIZZAMENTO ESPLICITO	17
1.11 FLAG PLC	18
1.11.1 FB256 Flag speciali.....	19
1.11.2 FB260-FB263 Flag di tastiera.....	20
1.11.3 FB280-FB289 Flag generali di interfaccia con CN	21
FB280 - Flag pagina attiva.....	21
FB281 - Flag asse attivo.....	21
1.11.3 FB768-FB895 Flag attivazione messaggi utente.....	22
Il PLC settando ciascuno di questi 1024 Flag attiva un Messaggio da 0 a 1023.	22
Il messaggio attivato è visualizzato sul CN.....	22
Il testo del messaggio è editabile con l'apposito tool con cui viene poi inviato al CN.....	22
I messaggi non provocano l'arresto e non pongono il CN in allarme. Vengono solo visualizzati.....	22
1.12 COLLEGAMENTO TRA CN KSE E PERSONAL COMPUTER	23
2.00 ISTRUZIONI KSE CON ESEMPI DI PROGRAMMAZIONE	24
2.01 COMBINAZIONI AND	25
2.02 COMBINAZIONI OR	26
2.03 OR CON COMBINAZIONI SCRITTE TRA PARENTESI	27
2.04 AND CON COMBINAZIONI SCRITTE TRA PARENTESI	28
2.05 COMBINAZIONI VARIE	29
2.06 INTERROGAZIONE DEI SEGNALI SULLO STATO LOGICO ZERO	31
2.07 ISTRUZIONI DI ASSEGNAZIONE	32
2.08 ISTRUZIONI DI SET	33
2.09 ISTRUZIONI DI RESET	34
2.10 REALIZZAZIONE DI UNA MEMORIA SR SU UN FLAG	36
2.11 ISTRUZIONI DI SET E RESET INCONDIZIONATI	37
2.12 INTRODUZIONE ALL'USO DEI TEMPORIZZATORI	38
2.12.1 GENERAZIONE DI UN IMPULSO	40
2.12.2 GENERAZIONE DI UN IMPULSO PROLUNGATO	41

2.12.3 RITARDO ALL'INSERZIONE	42
2.12.4 RITARDO ALL'INSERZIONE MEMORIZZATO.....	43
2.12.5 RITARDO ALLA DISINSERZIONE.....	44
2.13 USO DEI CONTATORI	45
2.13.1 CONTEGGIO IN AVANTI.....	46
2.13.2 CONTEGGIO ALL' INDIETRO	47
2.13.3 IMPOSTAZIONE DELLA PAROLA DI CONTEGGIO.....	48
2.13.4 AZZERAMENTO DELLA PAROLA DI CONTEGGIO.....	49
2.14 ISTRUZIONI DI TEST BIT	50
2.15 OPERAZIONI DI CARICAMENTO E TRASFERIMENTO	52
2.15.1 ELENCO DEGLI OPERANDI PER LE ISTRUZIONI (LA) E (LB).....	53
2.15.2 LETTURA DI PAROLE DA 16 Bit.....	54
2.15.2.1 LETTURA DI DUE BYTE DI FLAG	55
2.15.2.2 LETTURA DI UNA PAROLA DI DATI.....	56
2.15.3 LETTURA DI UN BYTE.....	57
2.15.3.1 LETTURA DI UN BYTE DI FLAG	58
2.15.3.2 LETTURA DI UN BYTE DESTRO DI UNA PAROLA DATI	59
2.15.3.3 LETTURA DI UN BYTE SINISTRO DI UNA PAROLA DATI	60
2.15.4 ELENCO DEGLI OPERANDI CON ISTRUZIONE (T).....	61
2.15.5 TRASFERIMENTO SU PAROLE DA 16 BIT	62
2.15.5.1 TRASFERIMENTO SU DUE BYTE DI FLAG	63
2.15.5.2 TRASFERIMENTO SU DATI (DW)	64
2.15.6 TRASFERIMENTO SU BYTE	65
2.15.6.1 TRASFERIMENTO SU UN BYTE DI FLAG	66
2.15.6.2 TRASFERIMENTO SUL BYTE DESTRO DI DATI (DW)	67
2.15.6.3 TRASFERIMENTO SUL BYTE SINISTRO DI DATI (DW)	68
2.15.7 CARICAMENTO E TRASFERIMENTO INDIRETTO DI DATI (DW).....	69
2.16 OPERAZIONI DI CONFRONTO	70
2.16.1 CONFRONTO SU UGUALE	72
2.16.2 CONFRONTO SU DIVERSO	73
2.16.3 CONFRONTO SU MAGGIORE	74
2.16.4 CONFRONTO SU MINORE.....	75
2.16.5 CONFRONTO SU MAGGIORE O UGUALE.....	76
2.16.6 CONFRONTO SU MINORE O UGUALE	77
2.17 OPERAZIONI ARITMETICHE	78
2.17.1 ADDIZIONE	79
2.17.2 ADDIZIONE CON RIPORTO	80
2.17.3 SOTTRAZIONE	81
2.17.4 SOTTRAZIONE CON RIPORTO NEGATIVO.....	82
2.17.5 MOLTIPLICAZIONE	84
2.17.6 DIVISIONE	86
2.18 OPERAZIONI LOGICHE TRA GLI ACCUMULATORI	88
2.18.1 AND.....	89
2.18.2 OR	90
2.18.3 XOR.....	91
2.19 SCAMBIO DEGLI ACCUMULATORI	92
2.20 INCREMENTO DEGLI ACCUMULATORI	93
2.21 DECREMENTO DEGLI ACCUMULATORI	94
2.22 SOMMA DI UNA COSTANTE ALL'ACCUMULATORE (A)	95
2.23 COMPLEMENTO DELL'ACCUMULATORE (A)	96
2.24 COMPLEMENTO A 2 DELL'ACCUMULATORE (A)	97
2.25 ROTAZIONI DELL'ACCUMULATORE (A)	98
2.25.1 ROTAZIONE CIRCOLARE A SINISTRA	99

2.25.2	ROTAZIONE A SINISTRA ATTRAVERSO IL CARRY	100
2.25.3	ROTAZIONE CIRCOLARE A DESTRA	101
2.25.4	ROTAZIONE A DESTRA ATTRAVERSO IL CARRY	102
2.26	SHIFT DELL'ACCUMULATORE (A)	103
2.26.1	SHIFT A SINISTRA	104
2.26.2	SHIFT ARITMETICO A SINISTRA	105
2.26.3	SHIFT A DESTRA	106
2.26.4	SHIFT ARITMETICO A DESTRA	107
2.27	OPERAZIONI SUI FLAG	108
2.27.1	SET DEL FLAG (C)	109
2.27.2	RESET DEL FLAG (C)	110
2.27.3	COMPLEMENTO DEL FLAG (C)	111
2.27.4	MEMORIZZAZIONE DEI FLAG	112
2.27.5	CARICAMENTO DEI FLAG	113
2.28	ISTRUZIONI DI SALTO	114
2.28.1	SALTI INCONDIZIONATI	115
2.28.2	SALTI CONDIZIONATI DA RLC	116
2.28.3	SALTI CONDIZIONATI DAI FLAG	117

1.00 GENERALITA'

1.01 CARATTERISTICHE TECNICHE DEL PLC CN KSE

CAPACITA' DI INDIRIZZAMENTO

512 INPUT FISICI (LETTI DA SCHEDE A BORDO DEL CN O DA DISPISITIVI REMOTI)

512 OUTPUT FISICI (SCRITTI SU SCHEDE A BORDO DEL CN O SU DISPISITIVI REMOTI)

1536 INPUT DI INTERSCAMBIO CN - PLC

1536 OUTPUT DI INTERSCAMBIO PLC - CN

6144 FLAG UTENTE (0.0 -767.7)

1024 FLAG MESSAGGI (768.0 -895.7)

4096 DW (BLOCCO DATI)

256 TEMPORIZZATORI

256 CONTATORI

PROGRAMMAZIONE STRUTTURATA

Il programma applicativo è organizzato in moduli di programma e di dati come segue :

OB0 - Blocco per le inizializzazioni

OB1 - Blocco organizzativo

PB (PB1 – PB255) - Blocchi di Programma (richiamati da Blocco organizzativo)

INT0 - Blocco di interrupt (task controllo assi)

DB - Blocco dati (attivo solo un blocco)

1.02 VARIABILE DI PROCESSO

I 0.0 - I 255.7	2048 ingressi digitali da immagine di processo
Q 0.0 - Q 255.7	2048 uscite digitali su immagine di processo
F 0.0 - F 767.7	6144 memorie utente per l'utilizzo sul programma
F 768.0 - F 895.7	1024 memorie utente per i messaggi
T0 - T255	256 temporizzatori
Z0 - Z255	256 contatori
IB0 - IB255	256 byte di Ingresso
IW0 - IW254	128 word di Ingresso
QB0 - QB255	256 byte di Uscita
QW0 - QW254	128 word di Uscita
FB0 - FB767	768 byte di Flag
FW0 - FW766	384 word di Flag
FB768 – FB895	128 byte di Flag Messaggi
FW768 – FW894	64 word di Flag
DW0 - DW4095	4096 word di Dati
DR0 - DR4095	4096 byte parte bassa di Dati
DL0 - DL4095	4096 byte parte alta di Dati

1.03 MAPPA DEGLI INGRESSI E DELLE USCITE

1.03.1 Organizzazione degli ingressi

INGRESSI (IB)

Byte	7	bit	0
8	Input distribuito		
:			
95			
96	Celle dipendenti da applicazione		
:	Vedere documento specifico celle CN		
255			

1.03.2 Organizzazione delle uscite

USCITE (QB)

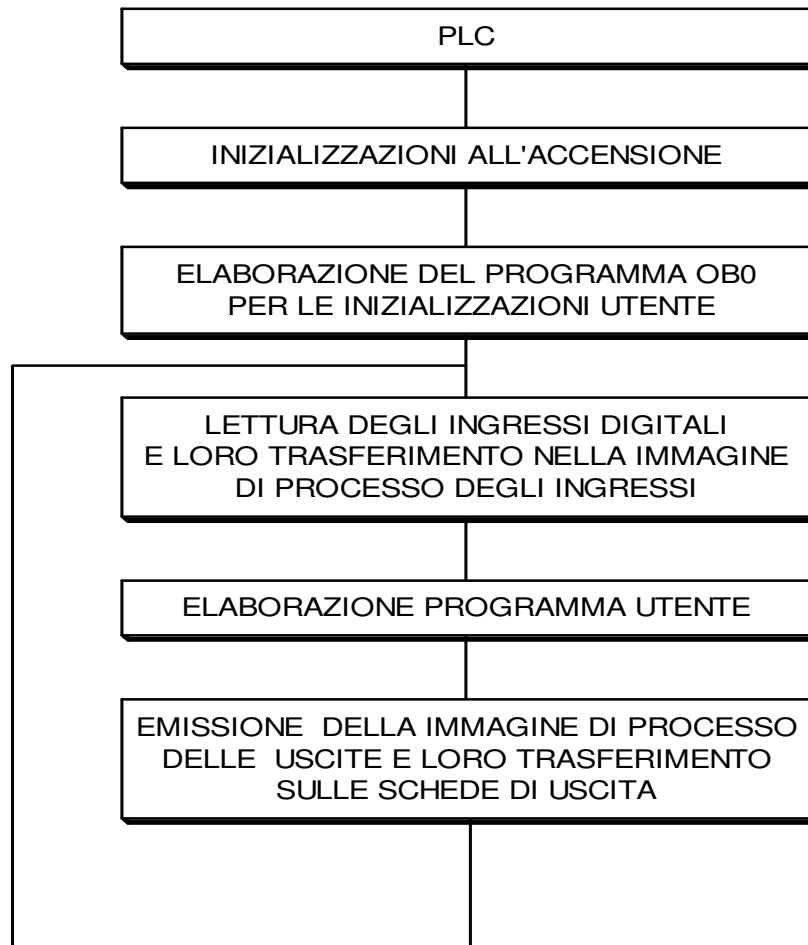
Byte	7	bit	0
0	Output distribuito		
:			
95			
96	Celle dipendenti da applicazione		
:	Vedere documento specifico celle CN		
255			

1.04 ELABORAZIONE CICLICA DEL PROGRAMMA

Il programma utente è costituito da una successione di istruzioni che vengono eseguite sequenzialmente dal microprocessore . Prima di iniziare l'elaborazione del programma utente, gli ingressi digitali vengono letti ed il loro valore viene introdotto nella immagine di processo: l'immagine di processo è una zona di memoria interna nella quale vengono riprodotti gli stati logici degli ingressi e delle uscite digitali. Dopo il trasferimento degli ingressi nella immagine di processo, inizia l'elaborazione del programma utente; le istruzioni che fanno riferimento agli ingressi prelevano il loro stato logico dalla immagine di processo degli ingressi, quelle che fanno riferimento alle uscite impostano lo stato di queste sulla immagine di processo delle uscite. Al termine dell'elaborazione del programma utente, l'immagine di processo delle uscite viene emessa sulle rispettive schede di uscita. L'elaborazione ciclica del programma prosegue riprendendo dall'aggiornamento della immagine di processo degli ingressi.

Rappresentazione schematica della elaborazione ciclica di un programma utente.

FLUSSO PLC



1.05 CAUSE DI ARRESTO DEL PLC

Il PLC può interrompere il ciclo per le seguenti cause:

- 1) Durata del programma superiore al ciclo PLC
- 2) Per un errore di programmazione (loop a seguito di salti)
- 3) Per problemi di memoria (controllo del PLC fallito)

Nel primo caso il problema può dipendere dall'utilizzo di istruzioni che impegnano il processore in modo aggressivo quali i timer. Può anche dipendere dall'utilizzo errato di alcune logiche che ritardano la fine del ciclo.

Nel secondo caso si utilizzano in modo errato i JUMP e si continua ad eseguire un salto indietro nel programma senza terminare mai il ciclo.

Nel terzo caso si hanno problemi alla memoria retentiva che contiene il programma PLC.

1.06 INIZIALIZZAZIONE DELLA MEMORIA DEL PLC

Parte della memoria del PLC viene inizializzata allo start-up della macchina e al trasferimento del programma nello strumento.

La parte di memoria inizializzata è:

- Tutti gli Ingressi
- Tutte le Uscite
- I flag dal 256.0 al 767.7
- I flag messaggi dal 768.0 al 895.7
- I primi 224 Contatori
- Tutti i Timer

N.B.: Il resto della memoria è da intendersi retentiva.

1.07 CONSIDERAZIONI SUL PROGRAMMA UTENTE

La struttura del programma PLC per il CN KSE è costituita essenzialmente da blocchi di programma e da uno di dati.

Questi blocchi sono rispettivamente OB0, OB1, INT0 i PB e DB.

La descrizione della funzionalità dei blocchi è la seguente:

BLOCCO OB0

Il blocco OB0 , se presente viene eseguito una sola volta alla accensione del controllore.

Tipicamente in questo blocco compaiono istruzioni di inizializzazione relative a variabili utente.

Non è indispensabile e può essere vuoto.

BLOCCO OB1

Viene eseguito ciclicamente a tempo PLC

Il blocco OB1 può contenere tutto il programma PLC senza nessuna limitazione sull'utilizzo delle istruzioni.

In genere viene però utilizzato per richiamare i vari blocchi di programma (PBnnn)

Questo blocco è assolutamente indispensabile.

BLOCCHI PBnnn (PB1 – PB255)

Questi blocchi contengono il programma PLC vero e proprio senza nessuna limitazione sull'utilizzo delle istruzioni.

Sono richiamati nel blocco OB1

BLOCCO INTO

Viene eseguito ciclicamente a tempo task Assi ed è quindi più rapido del ciclo PLC.

Questo blocco può contenere il programma PLC senza nessuna limitazione sull'utilizzo delle istruzioni ma deve essere utilizzato solo per operazioni "rapide" o che richiedano una perfetta sincronizzazione con il controllo assi.

BLOCCO DB (ATTIVO SOLO UN BLOCCO)

I blocco DB serve per la memorizzazione di dati.

Ha una lunghezza di 4096 WORD.

1.08 REGISTRI DI LAVORO DELLE ISTRUZIONI

Le istruzioni previste per il CN KSE sono suddivise in due categorie :

- a) Istruzioni logiche a bit
- b) Istruzioni logiche a byte/word

Le prime sono relative alla gestione del singolo ingresso, uscita e flag, tipica della gestione a contatti, mentre le seconde sono relative alla gestione dei blocchi dati, operazioni di mascheratura, conteggi, somme ecc..

I registri utilizzati sono rispettivamente l' "RLC" per la gestione a bit e "A" e "B" per la gestione byte/word.

L'RLC è un registro che rappresenta il risultato logico combinatorio della equazione corrente, esso viene impostato dalle istruzioni di tipo logico ed il suo valore può essere trasferito su variabili logiche oppure può condizionare istruzioni di salto, di chiamata blocchi ecc.

I registri A e B sono registri accumulatori a 16 Bit per la istruzioni PLC di tipo aritmetico e di movimento dati.

1.09 I FLAG degli Operandi

I flag influenzati dalle istruzioni sono i seguenti :

C	Carry
Z	Zero
S	Segno
O	Overflow

Per l'uso di questi flag occorre fare riferimento alla descrizione delle singole istruzioni.

1.10 CONSIDERAZIONE SUGLI OPERANDI DELLE ISTRUZIONI

1.10.1 INDIRIZZAMENTO ESPPLICITO

Il tipo di operando ed il suo indirizzo vengono specificati in modo esplicito. I tipi di operando ed il campo di variabilità dei relativi indirizzi sono elencati nel paragrafo "Variabili di processo".

Esempio :

A	I	10.5
=	Q	5.6
		Indirizzo
		Tipo di variabile
		Codice mnemonico istruzione

1.11 FLAG PLC

Come già descritto i flag indirizzabili a bit sono numerati da F0.0 a F895.7

Sono divisi per categorie:

- 1) FB000-FB255 Flag ritentivi (che mantengono il loro valore allo spegnimento)
- 2) FB256-FB259 Flag speciali
- 3) FB260-FB263 Flag di tastiera (dipendenti dal tipo di CN)
- 4) FB264-FB279 Flag liberi per l'utente (ex flag di attivazione allarmi e messaggi)
- 5) FB280-FB289 Flag generali di interfaccia con CN (dipendenti dall'applicazione)
- 6) FB290-FB767 Flag liberi per l'utente non ritentivi (vengono azzerati all'accensione)
- 7) FB768-FB895 Flag attivazione messaggi utente

1.11.1 FB256 Flag speciali

F	Significato
256.x	
0	Sempre "0"
1	Sempre "1"
2	Toggle ogni 4sec
3	Toggle ogni 2sec

1.11.2 FB260-FB263 Flag di tastiera

I Flag di tastiera possono differire da CN a CN

CN KSE PCA

	Bit: 7	6	5	4	3	2	1	0
FB260	Enter	Start	Stop	Reset	Help	Space		
FB261	Freccia Sinistra	Freccia Destra	Freccia Su	Freccia Giu	Tab	Ins	Del	
FB262	F8	F7	F6	F5	F4	F3	F2	F1
FB263			F16	F15	F14	F13	F12	F11

CN KSE 5V

	Bit: 7	6	5	4	3	2	1	0
FB260	Enter	Pag Up	Pag Dn	Home	Help	Space	Accens. led user	
FB261	Freccia Sinistra	Freccia Destra	Freccia Su	Freccia Giu	Tab	Back Space	Del	
FB262	F8	F7	F6	F5	F4	F3	F2	F1
FB263	Esc	Save	End	+	-		F10	F9

1.11.3 FB280-FB289 Flag generali di interfaccia con CN

FB280 - Flag pagina attiva

	Bit: 7	6	5	4	3	2	1	0
FB280	NUMERO PAGINA ATTIVA (0 - 255) VALORE STATICO							

FB281 - Flag asse attivo

	Bit: 7	6	5	4	3	2	1	0
FB281	ASSE W	ASSE V	ASSE Q	ASSE C	ASSE B	ASSE Z	ASSE Y	ASSE X

1.11.3 FB768-FB895 Flag attivazione messaggi utente

Il PLC settando ciascuno di questi 1024 Flag attiva un Messaggio da 0 a 1023.

Il messaggio attivato è visualizzato sul CN.

Il testo del messaggio è editabile con l'apposito tool con cui viene poi inviato al CN.

I messaggi non provocano l'arresto e non pongono il CN in allarme. Vengono solo visualizzati.

	7	6	5	4	3	2	1	0
FB 768	MP0007	MP0006	MP0005	MP0004	MP0003	MP0002	MP0001	MP0000

FB 895	MP1023	MP1022	MP1021	MP1020	MP1019	MP1018	MP1017	MP1016

1.12 COLLEGAMENTO TRA CN KSE E PERSONAL COMPUTER

Il collegamento tra l'unità di programmazione (personal computer) ed il controllore programmabile avviene tramite porta seriale RS232.

PLC CONNETTORE 9 POLI TIPO 'D' FEMMINA	PC CONNETTORE 9 POLI TIPO 'D' FEMMINA
PIN 3	PIN 2
PIN 2	PIN 3
PIN 5	PIN 5
CONTENITORE	CONTENITORE

Le linee di controllo CTS, TRS, DTR non sono utilizzate.
Non utilizzare cavi con lunghezza superiore a 5mt.



Attenzione:

**Assicurarsi che la terra del PC e dello strumento siano equipotenziali.
Se così non fosse eseguire un collegamento mediante un cavo tra la carcassa del PC e PLC.**

2.00 ISTRUZIONI KSE CON ESEMPI DI PROGRAMMAZIONE

2.01 COMBINAZIONI AND

Codice Mnemonico	Operando	Descrizione
A	I0.0 , 255.7	AND con un ingresso
A	Q0.0 , 255.7	AND con un'uscita
A	F0.0 , 895.7	AND con un flag
A	T0 , 255	AND con un timer
A	Z0 , 255	AND con un contatore

Flag influenzati : RLC

Esempio equazione in lista istruzioni CN KSE :

```
A    I    10.1
A    I    20.2
A    I    10.3
=    Q    5.6
```

L'uscita Q5.6 viene impostata ad '1' quando tutti gli ingressi assumono il valore logico '1'; viene impostata a '0' quando almeno uno degli ingressi assume il valore logico '0'.

2.02 COMBINAZIONI OR

Codice Mnemonico	Operando	Descrizione
O	I0.0 , 255.7	OR con un ingresso
O	Q0.0 , 255.7	OR con un'uscita
O	F0.0 , 895.7	OR con un flag
O	T0 , 255	OR con un timer
O	Z0 , 255	OR con un contatore

Flag influenzati : RLC

Esempio equazione in lista istruzioni CN KSE:

```
O    I    13.6
O    I    2.5
O    I    12.6
=    Q    20.5
```

L'uscita Q20.5 viene impostata ad '1' quando almeno uno degli ingressi assume il valore logico '1'; viene impostata a '0' quando tutti gli ingressi assumono il valore logico '0'.

2.03 OR CON COMBINAZIONI SCRITTE TRA PARENTESI

Codice Mnemonico	Operando	Descrizione
O(OR con una combinazione scritta tra parentesi
)		fine della combinazione tra parentesi

Flag influenzati : RLC

Esempio equazione in lista istruzioni CN KSE:

```
A    I    10.1
A    I    15.0
O(
A    I    30.2
A    I    30.3
)
=    Q    15.3
```

In maniera totalmente equivalente si può scrivere :

```
O(
A    I    10.1
A    I    15.0
)
O(
A    I    30.2
A    I    30.3
)
=    Q    15.3
```

L'uscita Q15.3 assume il valore logico '1' quando una delle due combinazioni AND assume il valore logico '1'. Il numero dei livelli di parentesi aperte innestate può arrivare fino a 10.

2.04 AND CON COMBINAZIONI SCRITTE TRA PARENTESI

Codice Mnemonico	Operando	Descrizione
A(AND con una combinazione scritta tra parentesi
)		fine della combinazione tra parentesi

Flag influenzati : RLC

Esempio equazione in lista istruzioni CN KSE:

```
O    I    10.1
O    I    10.2
A(
O    I    50.1
O    I    50.2
)
=    Q    50.3
```

In maniera totalmente equivalente si può scrivere :

```
A(
O    I    10.1
O    I    10.2
)
A(
O    I    50.1
O    I    50.2
)
=    Q    50.3
```

L'uscita Q50.3 assume il valore logico '1' quando entrambe le condizioni logiche di OR assumono il valore logico '1'. Il numero dei livelli di parentesi aperte innestate può arrivare fino a 10

2.05 COMBINAZIONI VARIE

Esempio equazione in lista istruzioni CN KSE:

```
A    I    10.1
A(
A    I    10.2
A    I    10.3
O    I    10.4
O    I    10.5
)
=    Q    10.0
```

In maniera totalmente equivalente si può scrivere :

```
O    I    10.4
O    I    10.5
O(
A    I    10.2
A    I    10.3
)
A    I    10.1
=    Q    10.0
```

oppure :

```
A    I    10.1
A(
O    I    10.4
O    I    10.5
O(
A    I    10.2
A    I    10.3
)
)
=    Q    10.0
```

Le tre rappresentazioni sono equivalenti.

2.06 INTERROGAZIONE DEI SEGNALI SULLO STATO LOGICO ZERO

AN	I	10.2
A	I	10.3
=	Q	10.5

Quando una istruzione logica "A" (and) ed "O" (or) è seguita dalla "N" (AN , ON), lo stato logico della variabile viene negato prima di essere elaborato. Nell'esempio riportato sopra, l'uscita Q10.5 assume il valore logico "1" quando I10.2 assume il valore "0" ed I10.3 il valore il valore "1".

2.07 ISTRUZIONI DI ASSEGNAZIONE

Codice Mnemonico	Operando	Descrizione
=	I0.0 , 255.7	Trasferimento su un ingresso
=	Q0.0 , 255.7	Trasferimento su un'uscita
=	F0.0 , 895.7	Trasferimento su un flag

Flag influenzati : NESSUNO

Esempio equazione in lista istruzioni CN KSE:

```
A    I    1.1
A    I    2.1
=    Q    10.3
O    I    10.3
O    F    10.3
=    Q    10.1
=    F    15.6
```

Sull'uscita Q10.3 viene trasferito il risultato dell'operazione AND tra I1.1 e I2.1.

Sull'uscita Q10.1 e sul flag F15.6 viene trasferito il risultato dell'operazione di OR tra I10.1 ed F10.3.

Dopo una istruzione di assegnazione, l' "RLC" rimane invariato ed è possibile fare, come nell'esempio riportato sopra, trasferimenti successivi su variabili diverse.

La prima istruzione logica "A", "AN", "A(", "O(", "O", "ON" dopo una istruzione di assegnazione, inizia una nuova elaborazione.

2.08 ISTRUZIONI DI SET

Codice Mnemonico	Operando	Descrizione
S	I0.0 , 255.7	Trasferimento su un ingresso
S	Q0.0 , 255.7	Trasferimento su un'uscita
S	F0.0 , 895.7	Trasferimento su un flag

Flag influenzati : NESSUNO

L'istruzione SET trasferisce alla variabile specificata il valore logico '1' solo se l' "RLC" della elaborazione precedente vale '1'; l' "RLC" vale '0', l'istruzione non ha alcun effetto.

Esempio equazione in lista istruzioni CN KSE:

```
A    I    10.3
S    F    20.1
A    F    30.5
S    F    15.6
S    Q    15.5
```

Il flag F20.1 viene impostato ad '1', se l'ingresso I10.3 vale '1'; il flag F15.6 e l'uscita Q10.5 vengono impostati ad '1' se F30.5 vale '1'.

Dopo una istruzione di SET, l' "RLC" rimane invariato ed è possibile fare, come nell'esempio riportato sopra, operazioni di SET successive su variabili diverse.

La prima istruzione logica "A", "AN", "A(", "O(", "O", "ON" dopo una istruzione di SET, inizia una nuova elaborazione.

2.10 REALIZZAZIONE DI UNA MEMORIA SR SU UN FLAG

Esempio equazione in lista istruzioni CN KSE:

```
A   I   1.0
S   F   100.1
A   I   1.1
R   F   100.1
```

Quando l'ingresso I1.0 vale "1" la memoria F100.1 viene impostata ad "1" e mantiene tale valore anche se I1.0 torna al valore "0".

Quando I1.1 vale "1", la memoria F100.1 viene azzerata e mantiene tale valore anche se I1.1 torna a "0".

Se I1.0 e I1.1 valgono contemporaneamente "1", la memoria F100.1 viene impostata a "0" perché il reset "R", essendo successivo rispetto al set "S", imposta per ultimo la memoria F100.1: la memoria SR così, realizzata è del tipo "a reset prioritario".

Per realizzare una memoria del tipo a "set prioritario" le istruzioni devono essere scritte come segue:

```
A   I   1.0
R   F   100.1
A   I   1.1
S   F   100.1
```

2.11 ISTRUZIONI DI SET E RESET INCONDIZIONATI

L'istruzione SU imposta sulla variabile specificata il valore '1', mentre RU imposta il valore "0"; entrambe sono indipendenti dal valore di "RLC".

Dopo una istruzione SU o RU , la prima istruzione logica "(A", "AN", "O", "ON", "A(", "O(", ")" , dà inizio ad una nuova elaborazione.

Codice Mnemonico	Operando	Descrizione
SU	I0.0 , 255.7	SET di un ingresso
SU	Q0.0 , 255.7	SET di un'uscita
SU	F0.0 , 895.7	SET di un flag
RU	I0.0 , 255.7	RESET di un ingresso
RU	Q0.0 , 255.7	RESET di un'uscita
RU	F0.0 , 895.7	RESET di un flag

Flag influenzati : NESSUNO

Esempio equazione in lista istruzioni CN KSE:

```
RU   Q   10.5
RU   F   10.1
SU   F   20.0
SU   Q   15.1
```

2.12 INTRODUZIONE ALL'USO DEI TEMPORIZZATORI

Ad ognuno dei 256 temporizzatori è assegnata, nella memoria, una parola da 32 bit che contiene il valore attuale del conteggio espresso in unità di ciclo PLC.

L'inizio del conteggio del tempo avviene quando l' "RLC", relativo alla combinazione logica che precede l'istruzione di start, passa da "0" ad "1".

La durata del conteggio viene prelevata dall'accumulatore "A" secondo il seguente formato:

- Accumulatore "A" :

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 B B B C C C C C C C C C C C

C = Conteggio
 B = Base tempi

L'accumulatore "A" può essere facilmente caricato nel formato suddetto utilizzando la costante di tipo "KT" che permette di esprimere un valore numerico nella forma :

LA KT 999.5
 | |
 | Base Tempi
 Conteggio

Il conteggio e la base tempi espressi tramite la costante "KT" vengono automaticamente elaborati come segue:

$$\text{Conteggio} + \text{Base Tempi} * 4096$$

Il risultato di questo calcolo rappresenta il valore che viene caricato sull'accumulatore che è conforme con il formato richiesto per l'avviamento dei temporizzatori.

Nell'esempio riportato sopra l'accumulatore "A" viene caricato con il valore $999 + 5 * 4096 = 21479$

Il valore del conteggio deve essere compreso tra 0 e 999 mentre la base tempi può assumere i valori da 0 a 5.

La base tempi è soggetta al tempo di ciclo PLC secondo la tabella sottostante:

TABELLA PER CODIFICA BASE TEMPI

Codifica base tempi	Valore temporale
---------------------	------------------

0	Ciclo PLC
1	10 * Ciclo PLC
2	100 * Ciclo PLC
3	1000 * Ciclo PLC
4	6000 * Ciclo PLC
5	60000 * Ciclo PLC

I temporizzatori, quando sono attivi, vengono decrementati in modo sincrono rispetto al programma ciclico OB1, ciò significa che l'uscita di un temporizzatore può cambiare di stato solo dopo l'elaborazione di OB1.

L'avviamento di un temporizzatore può avvenire secondo le 5 modalità seguenti :

SI	T nnn	Generazione di un impulso
SV	T nnn	Generazione di un impulso
SE	T nnn	Ritardo all'inserzione
SS	T nnn	Ritardo all'inserzione memorizzato
SA	T nnn	Ritardo alla disinserzione

Affinché un temporizzatore venga avviato occorre che l'elaborazione della combinazione logica precedente l'istruzione di start (SI, SV,SE,SS,SA) produca. un "RLC"=1 e che l'elaborazione ciclica precedente abbia prodotto un "RLC"=0.

Lo stato di conteggio del temporizzatore può essere utilizzato nelle combinazioni logiche e quindi essere trasferito su uscite e su flags .

Esempio :

A T 10
= Q 35.1

2.12.1 GENERAZIONE DI UN IMPULSO

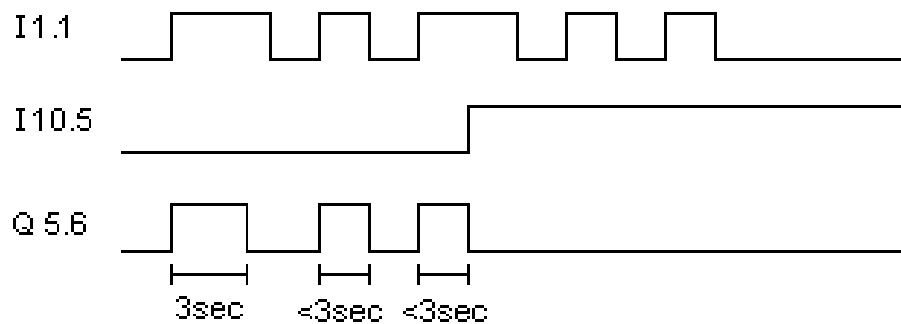
codice mnemonico: SI T nnn

Esempio :

A	I	1.1	Condizione di start
LA	KT	3.2	
SI	T	10	
A	I	10.5	Condizione di reset
R	T	10	
A	T	10	Trasferimento dello stato
=	Q	5.6	del timer su un'uscita

La condizione di reset potrebbe non servire e quindi non essere presente.

Rappresentazione temporale



2.12.2 GENERAZIONE DI UN IMPULSO PROLUNGATO

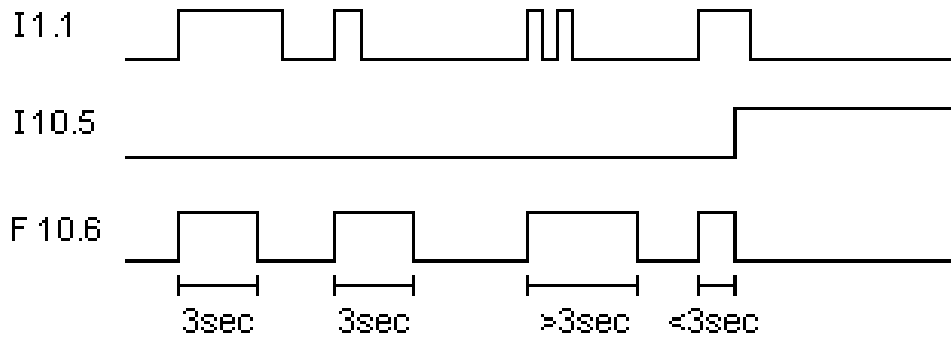
codice mnemonico: SV T nnn

Esempio :

A	I	1.1	Condizione di start
LA	KT	3.2	
SV	T	10	
A	I	10.5	Condizione di reset
R	T	10	
A	T	10	Trasferimento dello stato
=	F	10.6	del timer su un'uscita

La condizione di reset potrebbe non servire e quindi non essere presente.

Rappresentazione temporale



2.12.3 RITARDO ALL'INSERZIONE

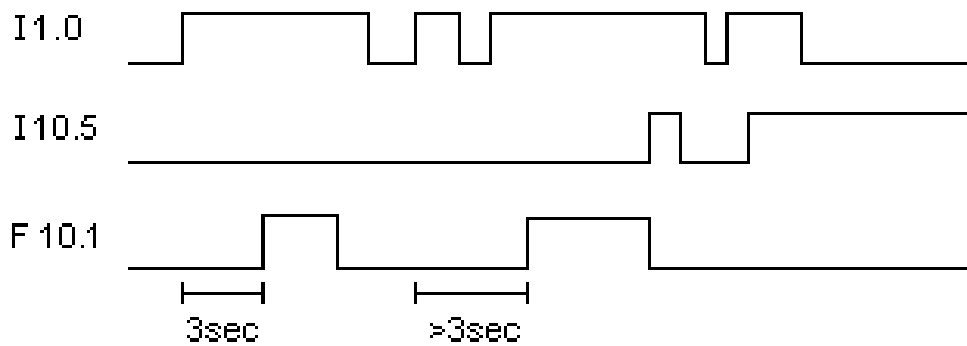
codice mnemonico: SE T nnn

Esempio :

LA	KT	5.2	
A	I	1.0	Condizione di start
SE	T	20	
A	I	10.5	Condizione di reset
R	T	20	
A	T	20	Trasferimento dello stato
=	F	10.1	del timer su un'uscita

La condizione di reset potrebbe non servire e quindi non essere presente.

Rappresentazione temporale



2.12.4 RITARDO ALL'INSERZIONE MEMORIZZATO

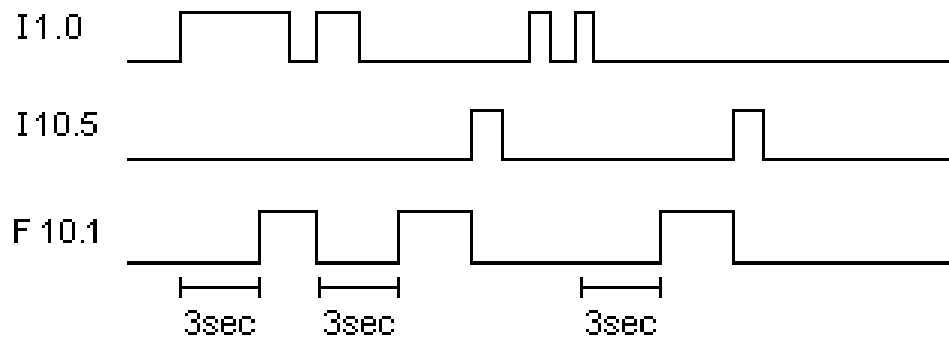
codice mnemonico: SS T nnn

Esempio :

LA	KT	3.2	
A	I	1.0	Condizione di start
SE	T	5	
A	I	10.5	Condizione di reset
R	T	5	
A	T	5	Trasferimento dello stato
=	F	10.1	del timer su un'uscita

La condizione di reset potrebbe non servire e quindi non essere presente.

Rappresentazione temporale



2.12.5 RITARDO ALLA DISINSERZIONE

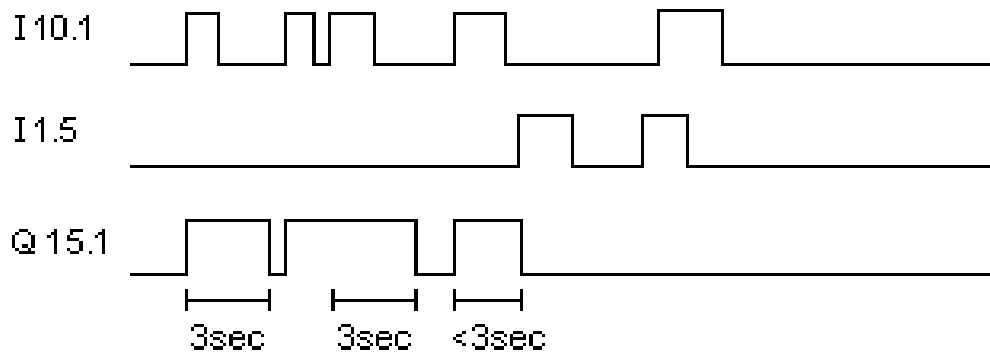
codice mnemonico: SA T nnn

Esempio :

A	I	1.0	Condizione di start
LA	KT	2.2	
SA	T	30	
A	I	1.5	Condizione di reset
R	T	30	
A	T	30	Trasferimento dello stato
=	Q	15.1	del timer su un'uscita

La condizione di reset potrebbe non servire e quindi non essere presente.

Rappresentazione temporale



2.13 USO DEI CONTATORI

Ad ognuno dei 256 contatori è assegnata, nella memoria, una parola da 16 bit che contiene il valore attuale del conteggio. Le operazioni previste per i contatori sono le seguenti:

ZV	Z nnn	Conteggio in avanti
ZR	Z nnn	Conteggio all'indietro
S	Z nnn	Impostazione di un valore sulla parola di conteggio
R	Z nnn	Azzeramento della parola di conteggio

Il valore contenuto nella parola di conteggio può anche essere caricato sugli accumulatori "A" e "B" per essere testato ed eventualmente trasferito su altre variabili. I singoli bit della parola di conteggio possono anche essere testati direttamente tramite le istruzioni BIT Znnn e BITN Znnn descritte in seguito.

Lo stato del conteggio del contatore può essere utilizzato nelle combinazioni logiche e quindi essere trasferito su uscite o flag.

Le interrogazioni dello stato del conteggio (A Znnn oppure 0 Znnn) forniscono un risultato logico "1" se il valore del conteggio è diverso da zero ed uno stato logico "0" se il conteggio vale zero.

2.13.1 CONTEGGIO IN AVANTI

codice mnemonico: ZV Z nnn

Il conteggio corrente del contatore specificato viene incrementato di una unità. L'incremento diventa attivo quando si verifica un fronte positivo di "RLC" tra due elaborazioni cicliche successive della combinazione logica che precede l'istruzione ZV.

Se il conteggio raggiunge il valore massimo di 65535 , con l'incremento successivo viene azzerato.

2.13.2 CONTEGGIO ALL' INDIETRO

codice mnemonico: ZR Z nnn

Il conteggio corrente del contatore specificato viene decrementato di una unità. Il decremento diventa attivo quando si verifica un fronte positivo di "RLC" tra due elaborazioni cicliche successive della combinazione logica che precede l'istruzione ZV.

Quando il conteggio corrente vale zero, con il decremento successivo viene impostato a 65535 -

2.13.3 IMPOSTAZIONE DELLA PAROLA DI CONTEGGIO

codice mnemonico: S Z nnn

La parola di conteggio del contatore specificato viene caricata con il valore presente sull'accumulatore "A".

Il caricamento diventa attivo quando si verifica un fronte positivo (da "0" a "1") di "RLC" tra due elaborazioni cicliche successive della combinazione logica che precede l'istruzione S.

2.13.4 AZZERAMENTO DELLA PAROLA DI CONTEGGIO

codice mnemonico: R Z nnn

Il reset di un contatore provoca l'azzeramento della sua parola di conteggio. Tale azzeramento è attivo quando il risultato "RLC" della combinazione che precede l'istruzione "R" vale "1"; se l'"RLC" vale "0" l'istruzione "R" non ha alcun effetto. Il reset di un contatore agisce staticamente quindi, fino a quando la condizione di reset è presente, non è possibile effettuare né il conteggio né l'impostazione:

Esempio :

Rappresentazione in linguaggio KSE/8A

A	I	10.1	Condizione per il conteggio avanti
ZV	Z	10	
A	I	10.2	Condizione per il conteggio indietro
ZR	Z	10	
LA	KH	5	
A	I	10.3	Condizione per il preset
S	Z	10	
A	I	10.4	Condizione per il reset
R	Z	10	
A	Z	10	
=	Q	10.0	

2.14 ISTRUZIONI DI TEST BIT

codice mnemonico : BIT operando

BITN operando

BIT operando: Imposta l' "RLC" allo stato logico "1" se il bit specificato dall'operando vale "1"

BITN operando: Imposta l' "RLC" allo stato logico "1" se il bit specificato dall'operando vale "0"

operandi :

T xxx.xx
 | |
 | 0 - 31 Numero del bit della parola di conteggio del temporizzatore
 0 - 127 Numero del temporizzatore

Z xxx.xx
 | |
 | 0 - 15 Numero del bit della parola di conteggio del contatore
 0 - 127 Numero del contatore

DW xxx.xx
 | |
 | 0 - 15 Numero del bit della parola di dati
 0 - 4094 Numero della parola dati

A xx
 |
 0 - 15 Numero del bit dell'accumulatore "A"

B xx
 |
 0 - 15 Numero del bit dell'accumulatore "B"

Le istruzioni BIT e BITN possono essere utilizzate all'inizio di una combinazione logica oppure prima di un salto condizionato.

Esempi:

LA	FW	400	Caricamento su "A" della parola FW300
BIT	A	3	Test del bit 3 di "A"
AN	I	1.1	
=	Q	10.5	

L'uscita Q10.5 viene impostata ad "1" se il bit 3 di FW400 vale "1" e se l'ingresso I1.1 vale "0".

bit 12	A	I	1.2		_____	Non hanno alcun effetto perché BITN imposta l' "RLC" del
	A	I	10.3			del timer 10 senza tener conto dell' "RLC" precedente .
	O	F	10.3			
	BITN	T	10.12			
	=	F	10.5			

2.15 OPERAZIONI DI CARICAMENTO E TRASFERIMENTO

Con le operazioni di caricamento e trasferimento è possibile effettuare scambi di informazioni tra schede di ingresso/uscita, immagini di processo di ingresso/uscita, flags, e blocchi dati.

L'operazione di caricamento può essere indirizzata verso l'accumulatore "A" (LA = Load A) oppure verso l'accumulatore "B" (LB = Load B) :

LA	FW	100	Carica su "A" la parola di flag numero 100
LB	IW	40	Carica su "B" l'immagine di processo dei 16 ingressi che vanno da I 40.0 a I 41.7

Le operazioni di trasferimento prevedono come sorgente dei dati solo l'accumulatore "A". I dati possono essere trasferiti verso le immagini di processo, verso i flag, verso i blocchi dati e verso la memoria. Il codice mnemonico per l'istruzione di trasferimento è rappresentato dalla lettera "T":

T	FW	100	Trasferisce il contenuto dell'accumulatore "A" sulla parola di flag numero 100 .
---	----	-----	--

Nelle operazioni di caricamento e di trasferimento gli operandi possono essere di 8 bit oppure di 16 bit .

2.15.1 ELENCO DEGLI OPERANDI PER LE ISTRUZIONI (LA) E (LB)

IB	0 : 255	Caricamento di un byte dall'immagine di processo degli ingressi
IW	0 : 254	Caricamento di due byte dall'immagine di processo degli ingressi
QB	0 : 255	Caricamento di un byte dall'immagine di processo delle uscite
QW	0 : 254	Caricamento di due byte dall'immagine di processo delle uscite
FB	0 : 895	Caricamento di un byte di flags
FW	0 : 766	Caricamento di due byte di flags
Z	0 : 255	Caricamento della parola di conteggio di un contatore
TL	0 : 255	Caricamento dei due byte meno significativi della doppia parola di conteggio di un
TH	0 : 255	Caricamento dei due byte più significativi della doppia parola di conteggio di un
DL	0 : 4095	Caricamento del byte sinistro (più, significativo) di una parola di dati
DR	0 : 4095	Caricamento del byte destro (meno significativo) di una parola di dati
DW	0 : 4095	Caricamento dei 16 bit di una parola dati
S	0 : 63	Caricamento di una parola di configurazione sistema

KH	0 : FFFF	Caricamento di una costante espressa in formato esadecimale
KB	0 : 127	Caricamento degli 8 bit meno significativi dell'accumulatore con una costante decimale con segno. Gli 8bit più significativi vengono azzerati
KI	0: 65535	Caricamento di una costante decimale senza segno
KS	0 : 32767	Caricamento di una costante decimale con segno
KT	0.0 : 999.5	Caricamento di una costante espressa nella forma conteggio base tempi
KC	AA : ZZ	Caricamento di una costante espressa mediante 2 caratteri con codifica ASCII

2.15.2 LETTURA DI PAROLE DA 16 Bit

Gli 8 bit meno significativi dell'accumulatore vengono caricati con il contenuto dell'indirizzo di memoria o di periferia specificato nel campo operandi, gli 8 bit più, significativi vengono caricati con il contenuto dell'indirizzo successivo.

2.15.2.1 LETTURA DI DUE BYTE DI FLAG

LA FW 100

ACCUMULATORE "A":

BIT :

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
-	F	B		1	0	1	-		-	F	B		1	0	0	-

LA FW 101

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
-	F	B		1	0	2	-		-	F	B		1	0	1	-

2.15.2.2 LETTURA DI UNA PAROLA DI DATI

LA DW 30

ACCUMULATORE "A":

1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
5	4	3	2	1	0										
-	-	-	-	-	-	D	W	3	0	-	-	-	-	-	-

2.15.3 LETTURA DI UN BYTE

Quando si effettua la lettura di un byte, gli 8 bit letti vengono trasferiti sugli 8 bit meno significativi dell'accumulatore, i rimanenti 8 bit più significativi dell'accumulatore vengono azzerati.

2.15.3.1 LETTURA DI UN BYTE DI FLAG

LA FB 100

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
0	0	0	0	0	0	0	0		-	F	B		1	0	0	-

2.15.3.2 LETTURA DI UN BYTE DESTRO DI UNA PAROLA DATI

LA DR 30

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
0	0	0	0	0	0	0	0		-	-	D	R	3	0	-	-

La parte bassa (8 Bit) della parola dati DW30 (parola composta da 16 Bit) viene trasferita nella parte bassa dell'accumulatore "A"

2.15.3.3 LETTURA DI UN BYTE SINISTRO DI UNA PAROLA DATI

LA DL 30

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
0	0	0	0	0	0	0	0		-	-	D	L	3	0	-	-

La parte alta (8 Bit) della parola dati DW30 (parola composta da 16 Bit) viene trasferita nella parte bassa dell'accumulatore "A" (la parte alta di "A" rimarrà con valore "0")

2.15.4 ELENCO DEGLI OPERANDI CON ISTRUZIONE (T)

IB	0 : 255	Trasferimento su un byte dell'immagine di processo degli
IW	0 : 254	Trasferimento su due byte dell'immagine di processo degli
QB	0 : 255	Trasferimento su un byte dell'immagine di processo delle
QW	0 : 254	Trasferimento su due byte dell'immagine di processo delle
FB	0 : 895	Trasferimento su un byte di flags
FW	0 :	Trasferimento su due byte di flags
DL	0 :	Trasferimento sul byte sinistro (più significativo) di una
DR	0 :	Trasferimento sul byte destro (meno significativo) di una
DW	0 :	Trasferimento sui 16 bit di una parola dati
S	0 : 63	Trasferimento nei 16 bit di una parola di configurazione

2.15.5 TRASFERIMENTO SU PAROLE DA 16 BIT

Gli 8 bit meno significativi dell'accumulatore vengono trasferiti all'indirizzo di memoria o di periferia specificato nel campo operandi mentre gli 8 bit più significativi vengono trasferiti all'indirizzo successivo.

2.15.5.1 TRASFERIMENTO SU DUE BYTE DI FLAG

T FW 100

ACCUMULATORE "A":

BIT :

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
-	F	B		1	0	1	-		-	F	B		1	0	0	-

T FW 101

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
-	F	B		1	0	2	-		-	F	B		1	0	1	-

2.15.5.2 TRASFERIMENTO SU DATI (DW)

T DW 30

ACCUMULATORE "A":

1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
5	4	3	2	1	0										
-	-	-	-	-	-	D	W	3	0	-	-	-	-	-	-

2.15.6 TRASFERIMENTO SU BYTE

Quando si effettua la lettura di un byte, gli 8 bit letti vengono trasferiti sugli 8 bit meno significativi dell'accumulatore, i rimanenti 8 bit più, significativi dell'accumulatore vengono azzerati.

2.15.6.1 TRASFERIMENTO SU UN BYTE DI FLAG

T FB 100

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
0	0	0	0	0	0	0	0		-	F	B		1	0	0	-

2.15.6.2 TRASFERIMENTO SUL BYTE DESTRO DI DATI (DW)

T DR 30

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
0	0	0	0	0	0	0	0		-	-	D	R	3	0	-	-

La parte bassa (8 Bit) dell'accumulatore "A" (composto da 16 Bit) viene trasferita nella parte bassa della parola dati (8bit)

2.15.6.3 TRASFERIMENTO SUL BYTE SINISTRO DI DATI (DW)

T DL 30

ACCUMULATORE "A":

1	1	1	1	1	1	9	8		7	6	5	4	3	2	1	0
5	4	3	2	1	0											
0	0	0	0	0	0	0	0		-	-	D	L	3	0	-	-

L'accumulatore "A" viene trasferito nella parte alta della parola dati DW30 (parola composta da 16 Bit)

2.15.7 CARICAMENTO E TRASFERIMENTO INDIRETTO DI DATI (DW)

La lettura e la scrittura di una parola di un blocco dati può avvenire anche in modo indiretto usando le seguenti istruzioni :

LDWI Caricamento indiretto
TDWI Trasferimento indiretto

La destinazione (LDWI) e l'origine (TDWI) dei dati è sempre l'accumulatore "A" mentre il numero della parola dati interessata allo scambio deve essere specificato nell'accumulatore "B".

esempi :

```
LA   KI   1000
LB   DW   5
TDWI
```

Il valore 1000 viene trasferito sulla parola il cui numero è specificato nella DW 5.

```
LB   DW   5
LDWI
```

Sull'accumulatore "A" viene caricata la parola dati il cui numero è specificato nella DW 5.

Queste istruzioni sono utili quando si devono creare tabelle oppure si devono fare trasferimenti di gruppi di parole da un blocco ad un altro in quanto, non facendo uso di indirizzi espliciti, possono essere utilizzate all'interno di un loop.

2.16 OPERAZIONI DI CONFRONTO

Nelle operazioni di confronto il contenuto dell'accumulatore "A" viene paragonato con quello dell'accumulatore "B" , il risultato del confronto imposta sia l' "RLC" che i flag "Z", "C", "S" ed "O". Il contenuto degli accumulatori viene considerato come numero binario dotato di segno secondo la rappresentazione in complemento a 2. i valori di "A" e di "B" rimangono invariati.

I flag "C", "S" ed "O" , contrassegnati con la "x" nelle tabelle riportate di seguito , assumono valori che non dipendono solo dalle condizioni relazionali "=", ">" e "<" , ma anche da altri fattori come il valore numerico di "A" e "B", il loro segno ed il tipo di operazione che viene effettuata per attivare i suddetti flag. L'operazione viene effettuata per l'impostazione dei flag nelle istruzioni di confronto è sempre una sottrazione tra l'operando di sinistra e quello di destra rispetto all'operatore :

con $A > B$ viene eseguita l'operazione $A - B$

con $B < A$ viene eseguito l'operazione $B - A$

Questa informazione è necessaria se si vuole fare uso dei flag contrassegnati con la "x" . Per esempio se si desidera effettuare il confronto tra "A" e "B" considerandoli come numeri binari senza segno, occorre utilizzare il flag "C". Per l'uso di tali flag fare riferimento alla tabella dei salti condizionati riportata al paragrafo "

2.28.3 SALTI CONDIZIONATI DAI FLAG".

2.16.1 CONFRONTO SU UGUALE

codice mnemonico : A=B

	RLC	Z	C	S	O
se A=B	1	1	X	X	X
se A<B	0	0	X	X	X
se A>B	0	0	X	X	X

2.16.2 CONFRONTO SU DIVERSO

codice mnemonico : $A > B$

	RLC	Z	C	S	O
se A=B	0	1	X	X	X
se A<B	1	0	X	X	X
se A>B	1	0	X	X	X

2.16.3 CONFRONTO SU MAGGIORE

codice mnemonico : A > B

	RLC	Z	C	S	O
se A=B	0	1	X	X	X
se A<B	0	0	X	X	X
se A>B	1	0	X	X	X

codice mnemonico : B > A

	RLC	Z	C	S	O
se A=B	0	1	X	X	X
se A<B	1	0	X	X	X
se A>B	0	0	X	X	X

2.16.4 CONFRONTO SU MINORE

codice mnemonico : A < B

	RLC	Z	C	S	O
se A=B	0	1	X	X	X
se A<B	1	0	X	X	X
se A>B	0	0	X	X	X

codice mnemonico : B < A

	RLC	Z	C	S	O
se A=B	0	1	X	X	X
se A<B	0	0	X	X	X
se A>B	1	0	X	X	X

2.16.5 CONFRONTO SU MAGGIORE O UGUALE

codice mnemonico : $A \geq B$

	RLC	Z	C	S	O
se A=B	1	1	X	X	X
se A<B	0	0	X	X	X
se A>B	1	0	X	X	X

codice mnemonico : $B \geq A$

	RLC	Z	C	S	O
Se A=B	1	1	X	X	X
Se A<B	1	0	X	X	X
Se A>B	0	0	X	X	X

2.16.6 CONFRONTO SU MINORE O UGUALE

codice mnemonico : A <= B

	RLC	Z	C	S	O
se A=B	1	1	X	X	X
se A<B	1	0	X	X	X
se A>B	0	0	X	X	X

codice mnemonico : B <= A

	RLC	Z	C	S	O
se A=B	1	1	X	X	X
se A<B	0	0	X	X	X
se A>B	1	0	X	X	X

2.17 OPERAZIONI ARITMETICHE

Gli operandi per le operazioni aritmetiche sono gli accumulatori "A" e "B". Il contenuto degli accumulatori viene considerato come un numero binario dotato di segno secondo la rappresentazione in complemento a 2.

2.17.1 ADDIZIONE

Codice mnemonico : A+B

Effettua la somma di "A" con "B" e memorizza il risultato in "A"; l'accumulatore "B" rimane invariato.

FLAG INFLUENZATI			
Z	C	S	O

Esempio :

LA	KH	300
LB	FW	280
A+B		
T	FW	290

2.17.2 ADDIZIONE CON RIPORTO

codice mnemonico : A++B

Effettua la somma di "A" con "B" tenendo conto del riporto (flag "C" del microprocessore) proveniente da un'altra operazione; il risultato viene memorizzato in "A" mentre "B" rimane invariato.

La addizione con riporto viene utilizzata per effettuare somme tra operandi con più di 16 bit.

FLAG INFLUENZATI			
Z	C	S	O

Esempio :

```
LA    FW    300
LB    FW    310
A+B
SF
T     FW    320
LA    FW    302
LB    FW    312
LF
A++B
T     FW    322
```

Nell'esempio viene calcolata la somma tra due operandi a 32 bit :

```
FW    300 , FW 302 addendo 1
FW    310 , FW 312 addendo 2
FW    320 , FW 322 risultato
```

Le istruzioni SF (Save Flag) ed LF (Load Flag) servono per salvare l'eventuale riporto (flag 'C') generato dalla prima somma che, altrimenti, verrebbe rovinato dalle operazioni di trasferimento e caricamento.

2.17.3 SOTTRAZIONE

codice mnemonico : A - B

Effettua la differenza di "A" con "B" e memorizza il risultato in "A" ; l'accumulatore "B" rimane invariato

codice mnemonico : B - A

Effettua la differenza di "B" con "A" e memorizza il risultato in "A" ; l'accumulatore "B" rimane invariato

FLAG INFLUENZATI			
Z	C	S	O

Esempio :

```
LA  FW  300
LB  KI   10
A-B
T   FW  310
```

Sulla FW 310 viene memorizzato il valore contenuto in FW 300 diminuito di 10.

2.17.4 SOTTRAZIONE CON RIPORTO NEGATIVO

codice mnemonico : A -- B

Effettua la differenza di "A" con "B" tenendo conto del riporto negativo (flag "C") proveniente da un'altra sottrazione; il risultato viene memorizzato in "A" mentre "B" rimane invariato.

codice mnemonico : **B--A**

Effettua la sottrazione di "B" con "A" tenendo conto del prestito (flag 'C') proveniente da un'altra operazione ; il risultato viene memorizzato in "A" e "B" rimane invariato.

La sottrazione con prestito viene utilizzata per effettuare differenze tra operandi con più di 16 bit.

FLAG INFLUENZATI			
Z	C	S	O

Esempio :

```
LA   FW   300
LB   FW   310
A-B
SF
T    FW   320
LA   FW   302
LB   FW   312
LF
A--B
T    FW   322
```

Nell'esempio viene calcolata la differenza tra due operandi di 32 bit

```
FW 300 , FW 302   minuendo
FW 310 , FW 312   sottraendo

FW 320 , FW 322   risultato
```

Le istruzioni SF (Save Flag) ed LF (Load Flag), servono per salvare l'eventuale riporto negativo (flag 'C') generato dalla prima differenza che , altrimenti , verrebbe rovinato dalle operazioni di trasferimento e caricamento.

2.17.5 MOLTIPLICAZIONE

codice mnemonico : **A * B**

Effettua la moltiplicazione di "A" per "B" considerando i rispettivi valori come numeri binari con segno. Il risultato di 32 bit viene memorizzato in "A" (16 bit meno significativi) e in "B" (16 bit più significativi).

codice mnemonico : **A ** B**

Effettua la moltiplicazione di "A" per "B" considerando i rispettivi valori come numeri binari senza segno. Il risultato di 32 bit viene memorizzato in "A" (16 bit meno significativi) e in "B" (16 bit più significativi).

FLAG INFLUENZATI			
	C		O

I flag "C" ed "O" vengono entrambi impostati ad "1" se il risultato della moltiplicazione supera i 16 bit. Quando si superano i 16 bit , occorre considerare anche il valore presente su "B"

Esempi :

```
LA   DW   3
LB   KI   10
A*B
T    DW   4
```

La DW 3 del blocco dati corrente viene moltiplicata per 10.

```
LA   KI   1500
LB   KI   80
A*B
T    DW   4
XCHG
T    DW   5
```

Viene memorizzato anche l'accumulatore "B" in quanto il risultato supera i 16 bit.

2.17.6 DIVISIONE

codice mnemonico : A / B

Effettua la divisione di "A" per "B" considerando i rispettivi valori come numeri binari con segno.

codice mnemonico : B / A

Effettua la divisione di "B" per "A" considerando i rispettivi valori come numeri binari con segno.

codice mnemonico : A/B

Effettua la divisione di "A" per "B" considerando i rispettivi valori come numeri binari senza segno.

codice mnemonico : B//A

Effettua la divisione di "B" per "A" considerando i rispettivi valori come numeri binari senza segno.

Il quoziente viene sempre memorizzato in "A" ed il resto in "B".

FLAG INFLUENZATI		
sono	tutti	indefiniti

Se il quoziente non può essere contenuto nell'accumulatore "A" perché troppo grande (troppo piccolo se negativo), viene settato il flag F255.5 e i valori di "A" e di "B" rimangono invariati.

Esempio senza gestione dell'overflow :

```
LA   FW   300
LB   KI   12
A/B
T    FW   320
```

Esempio con gestione dell'overflow :

```
RU   F    255.5
LA   FW   300
LB   FW   302
```

A/B

A F 255.5

JR Overflow

T FW 320

JMP Ok

Overflow: Gestione overflow

.

.

Ok:

A I 10.1 Proseguimento programma

2.18 OPERAZIONI LOGICHE TRA GLI ACCUMULATORI

2.18.1 AND

codice mnemonico : AW

Ognuno dei 16 bit di "A" viene messo in AND con il corrispondente bit di "B" , il risultato viene memorizzato in "A" mentre "B" rimane invariato.

FLAG INFLUENZATI			
Z	C	S	O

Esempio

```
LA    KH    3142
LB    KH    A322
AW
T     FW    400
```

```
"A"   0011 0001 0100 0010
"B"   1010 0011 0010 0010
AW    _____
"A"   0010 0001 0000 0010
```

Nella FW 400 viene memorizzato il valore esadecimale 2102 .

2.18.2 OR

codice mnemonico : OW

Ognuno dei 16 bit di "A" viene messo in OR con il corrispondente bit di "B" ; il risultato viene memorizzato in "A" mentre "B" rimane invariato.

FLAG INFLUENZATI			
Z	C	S	O

Esempio:

```
LA    KH    3142
LB    KH    A322
OW
T     FW    400
```

```
"A"   0011 0001 0100 0010
"B"   1010 0011 0010 0010
OW    _____
"A"   1011 0011 0110 0010
```

Nella FW 400 viene memorizzato il valore esadecimale B362 .

2.18.3 XOR

codice mnemonico : XOW

Ognuno dei 16 bit di "A" viene messo in OR esclusivo con il corrispondente bit di "B" ; il risultato viene memorizzato in "A" mentre "B" rimane invariato.

FLAG INFLUENZATI			
Z	C	S	O

Esempio

```
LA    KH    3142
LB    KH    A322
XOW
T     FW    400
```

```
"A"   0011 0001 0100 0010
"B"   1010 0011 0010 0010
XOW   _____
"A"   1001 0010 0110 0000
```

Nella FW400 viene memorizzato il valore esadecimale 9260.

2.19 SCAMBIO DEGLI ACCUMULATORI

codice mnemonico: XCHG

Il contenuto di "A" viene scambiato con quello di "B"

FLAG INFLUENZATI
NESSUNO

Esempio :

```
LA   FW   300
LB   FW   320
A+B
T    FW   400
XCHG
T    FW   420
```

La FW 320 viene sommata alla FW 300 e poi trasferita in FW 420.

2.20 INCREMENTO DEGLI ACCUMULATORI

codice mnemonico : INC A

Il contenuto di "A" viene incrementato di una unità.

codice mnemonico : INC B

Il contenuto di "B" viene incrementato di una unità.

2.21 DECREMENTO DEGLI ACCUMULATORI

codice mnemonico : DEC A

Il contenuto di "A" viene decrementato di una unità.

codice mnemonico : DEC B

Il contenuto di "B" viene decrementato di una unità.

FLAG INFLUENZATI			
Z	C	S	O

2.22 SOMMA DI UNA COSTANTE ALL'ACCUMULATORE (A)

codice mnemonico : ADD costante

La costante specificata viene sommata all'accumulatore "A".

codice mnemonico : ADC costante

La costante specificata ed il riporto (flag "C") vengono sommati all'accumulatore "A".

La costante può essere espressa nei seguenti formati :

KH	esadecimale
KI	decimale senza
KS	decimale con
KB	byte decimale con

FLAG INFLUENZATI			
Z	C	S	O

esempio:

```
LA  FW  300
ADD KI  15
T   FW  300
```

LA FW 300 viene incrementata di 15.

2.23 COMPLEMENTO DELL'ACCUMULATORE (A)

codice mnemonico : CM1

I 16 bit dell'accumulatore "A" vengono complementati : gli "0" diventano "1" e gli "1" diventano "0".

FLAG INFLUENZATI			
Z	C	S	O

Esempio :

```
LA   FW   400
CM1
T    FW   400
```

Supponendo che la FW 400 contenga il valore esadecimale 3942 , nella FW 400 viene memorizzato il valore esadecimale C6BD

```
"A"  0011 1001 0100 0010
CM1  _____
"A"  1100 0110 1011 1101
```

2.24 COMPLEMENTO A 2 DELL'ACCUMULATORE (A)

codice mnemonico : CM2

I 16 bit dell'accumulatore "A" vengono complementati (gli "0" diventano "1" e gli "1" diventano "0") e al risultato viene aggiunto il valore 1.

FLAG INFLUENZATI			
Z	C	S	O

Esempio :

```
LA   FW   400
CM2
T    FW   400
```

Supponendo che la FW 400 contenga il valore esadecimale 63 (99 decimale), nella FW 400 viene memorizzato il valore esadecimale FF9D che corrisponde a 99 .

```
"A"  0000 0000 0110 0011
CM2  _____
"A"  1111 1111 1001 1100
+    0000 0000 0000 0001
_____
"A"  1111 1111 1001 1100
```

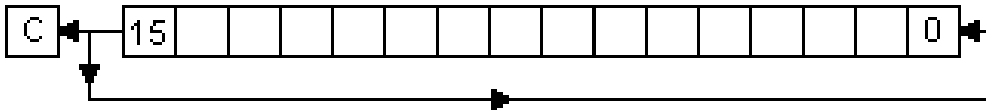
L'istruzione CM2 viene quindi utilizzata per trasformare il valore numerico dell'accumulatore "A" da positivo a negativo e viceversa.

2.25 ROTAZIONI DELL'ACCUMULATORE (A)

2.25.1 ROTAZIONE CIRCOLARE A SINISTRA

ROLW 0 : 15 Rotazione di "A" a sinistra. Il parametro rappresenta il numero delle posizioni di cui ruotare l'accumulatore e può assumere i valori da 0 a 15.

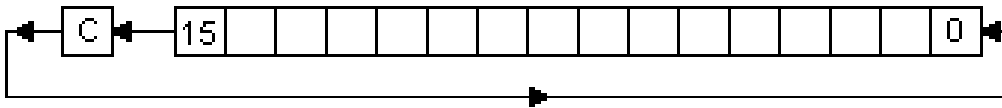
ROLI Rotazione di "A" a sinistra. Il numero delle posizioni di cui ruotare l'accumulatore "A" è contenuto in "B".



2.25.2 ROTAZIONE A SINISTRA ATTRAVERSO IL CARRY

RCLW 0 : 15 Rotazione di "A" a sinistra. Il parametro rappresenta il numero delle posizioni di cui ruotare l'accumulatore e può assumere i valori da 0 a 15.

RCLI Rotazione di "A" a sinistra. Il numero delle posizioni di cui ruotare l'accumulatore "A" è contenuto in "B".

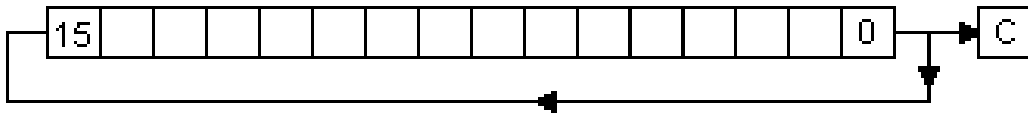


FLAG INFLUENZATI			
	C		O

2.25.3 ROTAZIONE CIRCOLARE A DESTRA

RORW 0 : 15 Rotazione di "A" a destra : Il parametro rappresenta il numero delle posizioni di cui ruotare l'accumulatore e può' assumere i valori da 0 a 15.

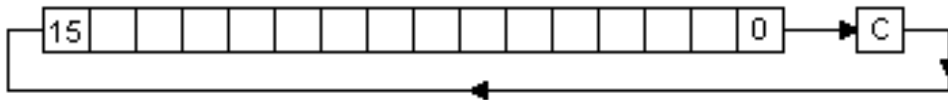
RORI Rotazione di "A" a destra. Il numero delle posizioni di cui ruotare l'accumulatore "A" e' contenuto in "B".



2.25.4 ROTAZIONE A DESTRA ATTRAVERSO IL CARRY

RCRW 0 : 15 Rotazione di "A" a destra . Il parametro rappresenta il numero delle posizioni di cui ruotare l'accumulatore e può, assumere i valori da 0 a 15.

RCRI Rotazione di "A" a destra. Il numero delle posizioni di cui ruotare l'accumulatore 'A' e' contenuto in "B".



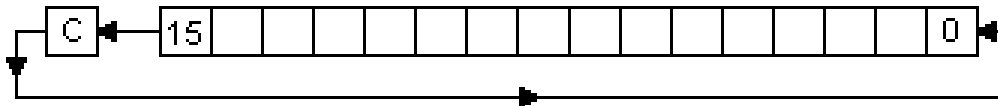
FLAG INFLUENZATI			
	C		O

2.26 SHIFT DELL'ACCUMULATORE (A)

2.26.1 SHIFT A SINISTRA

SHLW 0 : 15 Shift di "A" a sinistra. Il parametro rappresenta il numero delle posizioni di cui shiftare l'accumulatore e può assumere i valori da 0 a 15.

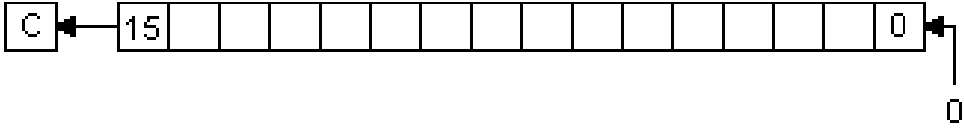
SHLI Shift di "A" a sinistra. Il numero delle posizioni di cui shiftare l'accumulatore "A" è contenuto in "B".



2.26.2 SHIFT ARITMETICO A SINISTRA

SALW 0 : 15 Come SHLW

SALI Come SHLI

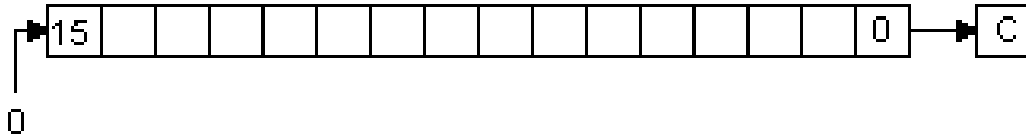


FLAG INFLUENZATI			
Z	C	S	O

2.26.3 SHIFT A DESTRA

SHRW 0 : 15 Shift di "A" a destra. Il parametro rappresenta il numero delle posizioni di cui shiftare l' accumulatore e può assumere i valori da 0 a 15 .

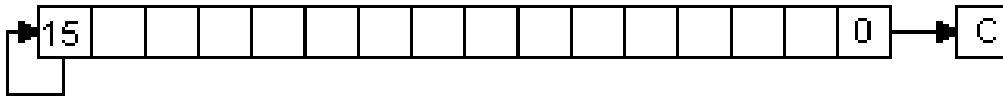
SHRI
"A" è Shift di "A" a destra. Il numero delle posizioni di cui shiftare l' accumulatore contenuto in "B".



2.26.4 SHIFT ARITMETICO A DESTRA

SARW Shift aritmetico di "A" a destra. Il parametro rappresenta il numero delle posizioni di cui shiftare l'accumulatore e può, assumere i valori da 0 a 15 .

SARI Shift aritmetico di "A" a destra. Il numero delle posizioni di cui shiftare l'accumulatore "A" è contenuto in "B"



FLAG INFLUENZATI			
Z	C	S	O

2.27 OPERAZIONI SUI FLAG

2.27.1 SET DEL FLAG (C)

codice mnemonico : STC

Sul flag "C" (Carry) viene impostato il valore "1"

Esempio :

```
LA   FW   400
STC
RCRW   1
T     FW   400
```

La FW 400 viene ruotata a destra di una posizione e sul bit più significativo viene inserito il valore "1" proveniente da flag "C".

2.27.2 RESET DEL FLAG (C)

codice mnemonico : CLC

Sul flag "C" (Carry) viene impostato il valore "0"

2.27.3 COMPLEMENTO DEL FLAG (C)

codice mnemonico : CMC

Il valore del flag "C" viene complementato .

FLAG INFLUENZATI			
	C		

2.27.4 MEMORIZZAZIONE DEI FLAG

codice mnemonico : SF

I flag "Z", "C", "S", "O", vengono memorizzati.

2.27.5 CARICAMENTO DEI FLAG

codice mnemonico : LF

I flag "Z", "C", "S", "O", vengono prelevati dalla memoria e trasferiti al microprocessore. L'istruzione LF ha senso solo se precedentemente è stata effettuata una SF.

FLAG INFLUENZATI			
Z	C	S	O

Le istruzioni SF (Save Flag) ed LF (Load Flag) vengono utilizzate quando si eseguono operazioni aritmetiche su operandi con più' di 16 bit .

Esempio : Somma di due operandi di 32 bit

```
LA    FW    300
LB    FW    310
A+B
SF
T     FW    320
LA    FW    302
LB    FW    312
LF
A++B
T     FW    322
```

Le istruzioni SF ed LF permettono di salvare il eventuale riporto (flag "C") generato dalla prima somma che , altrimenti , verrebbe rovinato dalle operazioni di trasferimento e caricamento.

2.28 ISTRUZIONI DI SALTO

Le istruzioni di salto possono essere incondizionate oppure condizionate dall' "RLC" e dai flag "Z" , "C" , "S" ed "O" del microprocessore. Per utilizzare i salti condizionati occorre che prima della istruzione di salto siano presenti le istruzioni che impostano il flag corrispondente. La destinazione del salto viene indicata con un nome simbolico di 8 caratteri. Non sono ammessi salti che oltrepassano il segmento. Nell'ambito del segmento non c'è alcuna limitazione alla lunghezza del salto.

2.28.1 SALTII INCONDIZIONATI

codice mnemonico JMP "destinazione"

esempio:

```
      A   I   2.1
      =   Q   10.3
      JMP SALTO_2
SALTO_1:
      O   I   10.1
      O   F   10.3
      =   Q   10.1
SALTO_2:
      A   I   50.3
      =   F   15.6
```

2.28.2 SALTII CONDIZIONATI DA RLC

Codice mnemonico : JR "destinazione"

Il salto viene eseguito solo se 1 "RLC" vale "1" altrimenti l'elaborazione prosegue.

codice mnemonico : JNR "destinazione"

Il salto viene eseguito solo se 1 "RLC" vale "0" altrimenti l'elaborazione prosegue.

Esempio :

```

A    I    1.1
JR   SALTO_1
O    I    10.1
O    F    10.3
=    Q    10.1
JNR  SALTO_2
SALTO_1:
A    I    50.3
=    F    15.6
SALTO_2:
LA   KH   50
```

2.28.3 SALTII CONDIZIONATI DAI FLAG

Codice	operando	Il salto viene eseguito	Interpretazione
JC	destinazion	C=1	se è più piccolo
JNC	destinazion	C=0	se non è più
JG	destinazion	(S xor O) or Z = 0	se >
JNG	destinazion	(S xor O) or Z = 1	se <=
JGE	destinazion	(S xor O)= 0	se >=
JL	destinazion	(S xor O)= 1	se <
JLE	destinazion	(S xor O) or Z = 1	se <=
JZ	destinazion	Z=1	se uguale o se
JNZ	destinazion	Z=0	se diverso o non
JS	destinazion	S=1	se negativo
JNS	destinazion	S=0	se non negativo
JO	destinazion	O=1	se c'è overflow
JNO	destinazion	O=0	se non c'è

Il termine "piccolo" sta ad indicare "minore" riferito a numeri senza segno . Gli altri operatori relazionali (>, >=, <, <=) si riferiscono sempre a valori dotati di segno. Gli operatori "or" ed "xor" stanno rispettivamente per OR ed OR esclusivo.