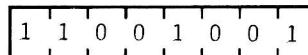


## RET

Operazione:  $PC_L \leftarrow (SP)$ ,  $PC_H \leftarrow (SP+1)$

Formato:      Codice operativo

RET



C9

### Descrizione:

Il controllo viene rimandato al flusso originale del programma estraendo il precedente contenuto del contatore di programma (PC) dalla parte alta dello stack di memoria esterna, nel quale era stato inserito dall'istruzione CALL. Questo avviene caricando prima il byte di ordine basso del PC con il contenuto dell'indirizzo di memoria indicato dallo stack pointer (SP), incrementando poi lo SP e caricando il byte di ordine superiore del PC con il contenuto dell'indirizzo di memoria ora indicato da SP. (Lo SP viene ora incrementato una seconda volta). Nel ciclo macchina seguente la CPU preleverà il successivo codice operativo del programma dalla locazione in memoria ora indicata dal PC.

CICLI M: 3    STATI T: 10 (4,3,3)    T.E. a 4 MHz: 2,50

Bit di condizione alterati: Nessuno

### Esempio:

Se il contenuto del contatore di programma è 3535H, il contenuto dello stack pointer è 2000H, il contenuto della locazione di memoria 2000H è B5H, e il contenuto della locazione di memoria 2001H è 18H, allora dopo l'esecuzione di

RET

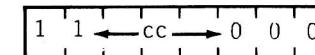
il contenuto dello stack pointer sarà 2002H e il contenuto del contatore di programma sarà 18B5H, che indica l'indirizzo del codice operativo del programma, che deve ora essere prelevato.

## RET cc

Operazione: Se cc è VERO:  $PC_L \leftarrow (SP)$ ,  $PC_H \leftarrow (SP+1)$

Formato:      Codice operativo      Operando

RET      cc



### Descrizione:

Se la condizione cc è vera, il controllo viene rimandato al flusso originale del programma estraendo il precedente contenuto del contatore di programma (PC) dalla parte alta dello stack di memoria esterna, in cui era stato inserito dall'istruzione CALL. Ciò avviene caricando prima il byte di ordine inferiore di PC con il contenuto dell'indirizzo di memoria indicato dallo stack pointer (SP), incrementando poi lo SP, e caricando il byte di ordine alto di PC con il contenuto dell'indirizzo di memoria ora indicato da SP. (Lo SP viene ora incrementato una seconda volta). Nel ciclo macchina successivo la CPU preleverà il successivo codice operativo del programma dalla locazione in memoria ora indicata da PC. Se la condizione cc è falsa, PC viene semplicemente incrementato come al solito, e il programma continua con l'istruzione che viene dopo in sequenza. La condizione cc si riferisce allo stato logico di uno tra i quattro flag (Z,C,P/V,S) del registro F.

Poichè ciascuno di essi può trovarsi nello stato 0 o nello stato 1, si ottiene un totale di otto condizioni possibili che sono indicate nella tabella seguente:

cc	Condizione	Flag
000	NZ non zero	Z
001	Z zero	Z
010	NC nessun riporto	C
011	C riporto	C
100	PO parità dispari	P/V
101	PE parità pari	P/V
110	P segno positivo	S
111	M segno negativo	S

Se cc è vera:

CICLI M: 3    STATI T: 11 (5,3,3)    T.E. a 4 MHz: 2,75

Se cc è falsa:

CICLI M: 1    STATI T: 5    T.E. a 4 MHz: 1,25

Bit di condizione alterati: Nessuno

Esempio:

Se il flag S nel registro F è settato, il contenuto del contatore di programma è 3535H, il contenuto dello stack pointer è 2000H, il contenuto della locazione di memoria 2000H è B5H, e il contenuto della locazione di memoria 2001H è 18H, allora dopo l'esecuzione di

RET M

il contenuto dello stack pointer sarà 2002H e il contenuto del contatore di programma sarà 18B5H, che indica l'indirizzo del codice operativo del programma, che deve venire successivamente prelevato.

## RETI

Operazione: Ritorno dall'interruzione

Formato:      Codice operativo

RETI

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

ED

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

4D

Descrizione:

Questa istruzione viene usata alla fine di una routine di servizio interruzione per:

1. Ripristinare di nuovo il contenuto del contatore di programma (PC) (analogia all'istruzione RET)
2. Segnalare ai dispositivi di I/O che la routine di interrupt è stata completata. L'istruzione RETI permette di avere una struttura d'interruzione a più livelli permettendo ai dispositivi di priorità più elevata di sospendere il servizio delle routine di priorità inferiore. Lo stato di IFF2 è copiato in IFF1.

Nota questa istruzione non riabilita gli interrupt, che erano stati disabilitati all'ingresso della routine di servizio dell'interruzione.

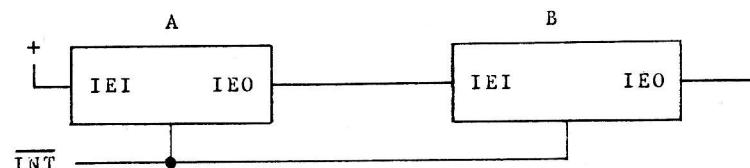
Per permettere il riconoscimento di interrupt quando la routine di servizio dell'interruzione è terminata, è necessario inserire, prima di RETI, l'istruzione di abilitazione dell'interruzione (EI).

CICLI M: 4    STATI T: 14 (4,4,3,3)    T.E. a 4 MHz: 3,50

Bit di condizione alterati: Nessuno

Esempio:

Sono dati due dispositivi d'interruzione, A e B collegati in una configurazione daisy chain, e A possiede una priorità più elevata di B.



B genera un'interrupt e viene riconosciuto. (L'uscita di abilitazione all'interrupt, IEO, di B si abbassa, impedendo a qualunque dispositivo di priorità inferiore di interrompere mentre viene servito B). Poi A genera un'interrupt, sospendendo il servizio di B. (L'IEO di A si abbassa, indicando che, in quel momento, viene servito un dispositivo di priorità più elevata). La routine A viene completata, e un'istruzione RETI risetta l'IEO di A, permettendo alla routine B di continuare. Una seconda istruzione RETI completa la routine B e l'IEO di B viene riportato alto permettendo l'accesso all'interrupt dei dispositivi di priorità inferiore.

## RETN

Operazione: Ritorno dall'interruzione non mascherabile

Formato:

Codice operativo

RETN

1	1	1	0	1	1	0	1	ED
---	---	---	---	---	---	---	---	----

0	1	0	0	0	1	0	1	45
---	---	---	---	---	---	---	---	----

Descrizione:

Usata alla fine di una routine di servizio per un interrupt non mascherabile, questa istruzione esegue un rientro incondizionato che funziona in modo identico all'istruzione RET. Cioè, il contenuto del contatore di programma memorizzato in precedenza viene estratto dalla parte alta dello stack di memoria esterna; il byte di ordine inferiore di PC viene caricato con il contenuto della locazione di memoria indicata dallo stack pointer (SP), SP viene incrementato, il byte di ordine alto di PC è caricato con il contenuto della locazione di memoria ora indicata da SP, e SP viene incrementato di nuovo. Il controllo viene ora rimandato al flusso originale del programma: nel ciclo macchina seguente la CPU preleverà il codice operativo successivo dalla locazione in memoria ora indicata da PC. Inoltre lo stato di IFF2 viene ricopiatò in IFF1 nello stato che aveva prima di accettare NMI.

CICLI M: 4      STATI T: 14 (4,4,3,3)      T.E. a 4 MHz: 3,50

Bit di condizione alterati: Nessuno

Esempio:

Se il contenuto dello stack pointer è 1000H e il contenuto del contatore di programma è 1A45H quando viene ricevuto un segnale di interrupt non mascherabile (NMI), la CPU ignorerà l'istruzione successiva e ripartirà invece dall'indirizzo di memoria 0066H. Cioè, l'attuale contenuto del contatore di programma, 1A45H, verrà inserito nelle locazioni dello stack di indirizzo OFFFH e OFFEH, prima il byte di ordine alto, e 0066H sarà caricato nel contatore di programma.

Questo indirizzo dà inizio ad una routine di servizio interruzione che termina con l'istruzione RETN. Con l'esecuzione di RETN, il precedente contenuto del contatore di programma viene estratto dallo stack di memoria esterno, prima l'ordine basso, riportando SP al valore 1000H antecedente la routine di gestione dell'interruzione non mascherabile. Il flusso di programma continua da dove era stato lasciato con il prelevamento di un codice operativo dalla locazione 1A45H.

## RST p

Operazione:  $(SP-1) \leftarrow PC_H$ ,  $(SP-2) \leftarrow PC_L$ ,  
 $PC_H \leftarrow 0$ ,  $PC_L \leftarrow p$

<u>Formato:</u>	<u>Codice operativo</u>	<u>Operando</u>
	RST	p

### Descrizione:

L'attuale contenuto del contatore di programma (PC) viene inserito nello stack di memoria esterno, e la posizione di memoria di pagina zero data dall'operando p è caricata in PC. Inizia poi l'esecuzione del programma con il codice operativo nell'indirizzo ora indicato da PC. Il caricamento è eseguito decrementando prima il contenuto dello stack pointer (SP), caricando il byte di ordine superiore di PC nell'indirizzo di memoria ora indicato da SP, decrementando di nuovo SP, e caricando il byte di ordine basso di PC nell'indirizzo ora indicato da SP. L'istruzione ReSTart permette di saltare ad uno degli otto indirizzi, come mostra la tabella che segue. Nel codice oggetto sono presenti tre bit (campo t) che indicano codificati gli otto possibili valori per l'operando p secondo la corrispondenza definita.

Nota: dato che tutti gli indirizzi sono a pagina zero della memoria, il byte di ordine alto di PC è caricato con 00H. Nel byte di ordine basso di PC è invece caricato uno dei valori presenti nella colonna p della tabella.

p	t
00H	000
08H	001
10H	010
18H	011
20H	100
28H	101
30H	110
38H	111

CICLI M: 3    STATI T: 11 (5,3,3)    T.E. a 4 MHz: 2,75

### Esempio:

Se il contenuto del contatore di programma è 15B3H, dopo l'esecuzione di

RST 18H (Codice oggetto 1101111)

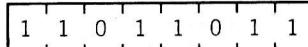
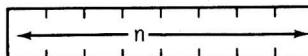
il PC conterrà 0018H, come l'indirizzo del codice operativo da prelevare successivamente.

## GRUPPO DI INPUT ED OUTPUT

## IN A, (n)

Operazione:  $A \leftarrow (n)$

Formato:	Codice operativo	Operandi
IN	A, (n)	

DB

Descrizione:

L'operando n è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare un dispositivo di I/O tra le 256 porte possibili. Contemporaneamente sulle linee da A8 ad A15 del bus degli indirizzi, viene posto il contenuto dell'accumulatore. In risposta a tale operazione la porta selezionata presenta sul bus dei dati un byte che viene così scritto nell'accumulatore (registro A) nella CPU.

CICLI M: 3    STATI T: 11 (4,3,4)    T.E. a 4 MHz: 2,75

Bit di condizione alterati: Nessuno

Esempio:

Se il contenuto dell'accumulatore è 23H e il byte 7BH è disponibile al dispositivo periferico di I/O corrispondente all'indirizzo 01H, allora dopo l'esecuzione di

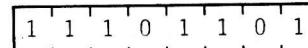
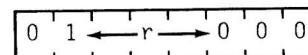
IN A, (01H)

l'accumulatore conterrà 7BH.

## IN r, (C)

Operazione:  $r \leftarrow (C)$

Formato:	Codice operativo	Operandi
IN	r, (C)	

ED

Descrizione:

Il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare un dispositivo di I/O tra le 256 porte possibili. Il contenuto del registro B viene posto sulla parte alta (da A8 ad A15) del bus degli indirizzi. Poi un byte della porta selezionata viene posto sul bus dei dati e scritto nel registro r nella CPU. Il registro r rappresenta uno qualunque dei registri della CPU come si vede nella seguente tabella, che mostra la corrispondenza tra i registri (B,C,D,E,H,L, o A) e valori del campo r nel codice oggetto.

L'esecuzione di questa istruzione modifica i flag secondo il valore del dato letto dalla parte di I/O.

Registro	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

CICLI M: 3    STATI T: 12 (4,4,4)    T.E. a 4 MHz: 3,00

Bit di condizione alterati:

- S: Settato se il dato letto è negativo;  
altrimenti resettato
- Z: Settato se il dato letto è zero;  
altrimenti resettato
- H: Resetto
- P/V: Settato in caso di parità pari;  
altrimenti resettato
- N: Resetto
- C: Inalterato

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 10H, e il byte 7BH è disponibile al dispositivo periferico di I/O corrispondente all'indirizzo 07H, dopo l'esecuzione di

IN D, (C)

il registro D conterrà 7BH.

INI

Operazione:  $(HL) \leftarrow (C), B \leftarrow B - 1, HL \leftarrow HL + 1$

Formato: Codice operativo

INI

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

ED

1	0	1	0	0	0	1	0
---	---	---	---	---	---	---	---

A2

Descrizione:

Il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare un dispositivo di I/O tra le 256 porte possibili. Il registro B può essere usato come contatore dei byte, ed il suo contenuto viene posto nella parte alta (da A8 ad A15) del bus degli indirizzi. Poi un byte della porta selezionata è posto sul bus dei dati e scritto nella CPU. Il contenuto della coppia di registri HL viene quindi posto sul bus degli indirizzi e il dato letto dal dispositivo di I/O viene scritto nella locazione di memoria corrispondente. Infine il contatore dei byte è decrementato, e incrementata invece la coppia di registri HL.

CICLI M: 4      STATI T: 16 (4,5,3,4)      T.E. a 4 MHz: 4,00

Bit di condizione alterati:

- S: Non definito
- Z: Settato se  $B-1=0$ ;  
altrimenti resettato
- H: Non definito
- P/V: Non definito
- N: Settato
- C: Inalterato

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 10H, il contenuto della coppia di registri HL è 1000H, e il byte 7BH è disponibile al dispositivo periferico di I/O corrispondente all'indirizzo 07H, allora dopo l'esecuzione di

INI

la locazione di memoria 1000H conterrà 7BH, la coppia di registri HL conterrà 1001H, e il registro B conterrà 0FH.

# INIR

Operazione:  $(HL) \leftarrow (C)$ ,  $B \leftarrow B - 1$ ,  $HL \leftarrow HL + 1$

Formato: Codice operativo

INIR

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

ED

1	0	1	1	0	0	1	0
---	---	---	---	---	---	---	---

B2

Descrizione:

Il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili. Il registro B viene usato come contatore dei byte, ed il suo contenuto è posto nella parte alta (da A8 ad A15) del bus degli indirizzi. Poi un byte della porta selezionata è posto sul bus dei dati e scritto nella CPU. Il contenuto della coppia di registri HL è posto sul bus degli indirizzi e il dato letto dal dispositivo di I/O è scritto nella locazione di memoria corrispondente. Poi la coppia di registri HL viene incrementata, e il contatore dei byte decrementato.

Se il decremento fa sì che B vada a zero, l'istruzione è terminata.

Se B non è zero, il PC è decrementato di due e l'istruzione viene ripetuta. Si noti che se B è settato a zero prima che venga eseguita l'istruzione, verranno prelevati dal dispositivo di I/O 256 byte di dati. Inoltre le interruzioni verranno riconosciute e saranno eseguiti due cicli di refresh dopo ogni trasferimento dei dati.

Se  $B \neq 0$ :

CICLI M: 5      STATI T: 21 (4,5,3,4,5)      T.E. a 4 MHz: 5,25

Se  $B = 0$ :

CICLI M: 4      STATI T: 16 (4,5,3,4)      T.E. a 4 MHz: 4,00

Bit di condizione alterati:

S:	Non definito
Z:	Settato
H:	Non definito
P/V:	Non definito
N:	Settato
C:	Inalterato

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 03H, il contenuto della coppia di registri HL è 1000H, e al dispositivo periferico di I/O corrispondente all'indirizzo 07H è disponibile la sequenza di byte seguente:

51H

A9H

03H

allora dopo l'esecuzione di

INIR

la coppia di registri HL conterrà 1003H, il registro B conterrà zero, e le locazioni di memoria avranno i seguenti contenuti:

Locazione	Contenuto
1000H	51H
1001H	A9H
1002H	03H

## IND

Operazione:  $(HL) \leftarrow (C)$ ,  $B \leftarrow B - 1$ ,  $HL \leftarrow HL - 1$

Formato:      Codice operativo

IND

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

ED

1	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---

AA

Descrizione:

Il contenuto del registro C viene posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili. Il registro B può essere usato come contatore dei byte, ed il suo contenuto viene posto sulla parte alta (da A8 ad A15) del bus degli indirizzi. Poi un byte della porta selezionata è letto sul dispositivo di I/O e scritto nella CPU. Il contenuto della coppia di registri HL è posto sul bus degli indirizzi e il dato letto sul dispositivo di I/O è scritto nella locazione di memoria corrispondente. Infine, il contatore dei byte e la coppia di registri HL vengono decrementati.

CICLI M: 4      STATI T: 16 (4,5,3,4)      T.E. a 4 MHz: 4,00

Bit di condizione alterati:

- S: Non definito
- Z: Settato se  $B-1=0$ ;  
altrimenti resettato
- H: Non definito
- P/V: Non definito
- N: Settato
- C: Inalterato

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 10H, il contenuto della coppia di registri HL è 1000H, e il byte 7BH è disponibile al dispositivo periferico di I/O corrispondente all'indirizzo 07H, allora dopo l'esecuzione di

IND

la locazione di memoria 1000H conterrà 7BH, la coppia di registri HL conterrà 1001H, e il registro B conterrà 0FH.

## INDR

Operazione:  $(HL) \leftarrow (C)$ ,  $B \leftarrow B - 1$ ,  $HL \leftarrow HL - 1$

Formato:      Codice operativo

INDR

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

ED

1	0	1	1	1	0	1	0
---	---	---	---	---	---	---	---

BA

Descrizione:

Il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili. Il registro B viene usato come contatore di byte, ed il suo contenuto è posto sulla parte alta (da A8 ad A15) del bus degli indirizzi. Poi un byte della porta selezionata è posto sul bus degli indirizzi e scritto nella CPU. Il contenuto della coppia di registri HL è posto sul bus degli indirizzi e il dato letto sul dispositivo di I/O è scritto nella locazione di memoria corrispondente. Poi HL ed il contatore dei byte vengono decrementati. Se il decremento fa sì che B vada a zero, l'istruzione è terminata. Se B non è zero, PC è decrementato di due e l'istruzione viene ripetuta. Si noti che, se B è settato a zero prima che l'istruzione venga eseguita, verranno prelevati dal dispositivo di I/O 256 byte di dati. Inoltre, le interruzioni verranno riconosciute e saranno eseguiti due cicli di refresh dopo ogni trasferimento dei dati.

Se  $B \neq 0$ :

CICLI M: 5      STATI T: 21 (4,5,3,4,5)      T.E. a 4 MHz: 5,25

Se  $B = 0$

CICLI M: 4      STATI T: 16 (4,5,3,4)      T.E. a 4 MHz: 4,00

Bit di condizione alterati:

S:	Non definito
Z:	Settato
H:	Non definito
P/V:	Non definito
N:	Settato
C:	Inalterato

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 03H, il contenuto della coppia di registri HL è 1000H, e al dispositivo periferico di I/O corrispondente all'indirizzo 07H è disponibile la sequenza di byte seguente:

51 H  
A9H  
03 H

allora dopo l'esecuzione di

INDR

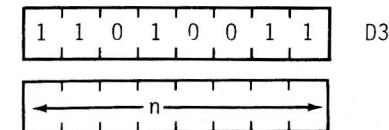
la coppia di registri HL conterrà OFFDH, il registro B conterrà zero, e le locazioni di memoria avranno i seguenti contenuti:

Locazione	Contenuto
OFFEH	03H
0FFFH	A9H
1000H	51H

OUT (n), A

Operazione: (n)  $\leftarrow$  A

<u>Formato:</u>	<u>Codice operativo</u>	<u>Operandi</u>
	OUT	(n), A



Descrizione:

L'operando n è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili. Il contenuto dell'accumulatore (registro A), inoltre, è contemporaneamente presentato sulla parte alta (da A8 ad A15) del bus degli indirizzi. Poi il byte contenuto nell'accumulatore è posto nel dispositivo periferico selezionato.

CICLI M: 3      STATI T: 11 (4,3,4)      T.E. a 4 MHz: 2,75

Bit di condizione alterati: Nessuno

Esempio:

Se il contenuto dell'accumulatore è 23H, dopo l'esecuzione di

OUT (01H), A

il byte 23H sarà stato scritto nel dispositivo periferico di I/O corrispondente all'indirizzo 01H.

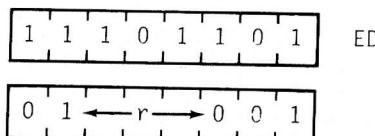
## OUT (C), r

Operazione:  $(C) \leftarrow r$

Formato: Codice operativo      Operandi

OUT

(C), r



Descrizione:

Il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili.

Il contenuto del registro B è posto sulla parte alta (da A8 ad A15) del bus degli indirizzi. Poi il byte contenuto nel registro r è posto sul bus dei dati e scritto nel dispositivo periferico selezionato. Il registro r rappresenta uno qualunque dei registri della CPU come si vede nella tabella seguente, che mostra la corrispondenza tra i registri (B,C,D,E,H,L o A) e i valori del campo r nel codice oggetto:

Registro	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

CICLI M: 3   STATI T: 12 (4,4,4)   T.E. a 4 MHz: 3,00

Bit di condizione alterati: Nessuno

Esempio:

Se il contenuto del registro C è 01H e il contenuto del registro D è 5AH, dopo l'esecuzione di

OUT (C), D

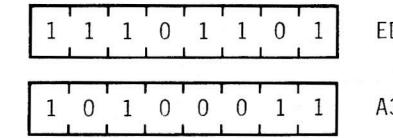
il byte 5AH sarà stato scritto nel dispositivo periferico di I/O corrispondente all'indirizzo 01H.

## OUTI

Operazione:  $(C) \leftarrow (HL)$ ,  $B \leftarrow B-1$ ,  $HL \leftarrow HL + 1$

Formato: Codice operativo

OUTI



Descrizione:

Il contenuto della coppia di registri HL è posto sul bus degli indirizzi per selezionare una locazione in memoria. Il byte contenuto in questa locazione di memoria viene temporaneamente memorizzato nella CPU. Quindi, dopo che il contatore dei byte (B) è stato decrementato, il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili. Il registro B può essere usato come contatore dei byte, ed il suo valore decrementato viene posto sulla parte alta (da A8 ad A15) del bus degli indirizzi. Il byte da mettere in uscita è posto sul bus dei dati e scritto nel dispositivo periferico selezionato. Infine viene decrementata la coppia di registri HL.

CICLI M: 4   STATI T: 16 (4,5,3,4)   T.E. a 4 MHz: 4,00

Bit di condizione alterati:

- |      |  |
|------|--|
| S:   | Non definito                                     |
| Z:   | Settato se $B - 1 = 0$ ;<br>altrimenti resettato |
| H:   | Non definito                                     |
| P/V: | Non definito                                     |
| N:   | Settato  |
| C:   | Inalterato                                       |

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 10H, il contenuto della coppia di registri HL è 1000H, e il contenuto dell'indirizzo di memoria 1000H è 59H, dopo l'esecuzione di

OUTI

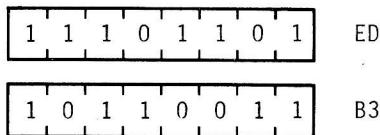
il registro B conterrà 0FH, la coppia di registri HL conterrà 1001H, e il byte 59H sarà stato scritto nel dispositivo periferico di I/O corrispondente all'indirizzo 07H.

## OTIR

Operazione:  $(C) \leftarrow (HL)$ ,  $B \leftarrow B - 1$ ,  $HL \leftarrow HL + 1$

Formato: Codice operativo

OTIR



### Descrizione:

Il contenuto della coppia di registri HL è posto sul bus degli indirizzi per selezionare una locazione in memoria.

Il byte contenuto in questa locazione di memoria è temporaneamente memorizzato nella CPU. Poi, dopo che il contatore dei byte (B) è stato decrementato, il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili. Il registro B può essere usato come contatore dei byte, ed il suo valore decrementato è posto sulla parte alta (da A8 ad A15) del bus d'indirizzo. Quindi il byte da mettere in uscita è posto sul bus dei dati e scritto nel dispositivo periferico selezionato. Poi la coppia di registri HL viene incrementata. Se il registro decrementato B non è zero, il contatore di programma (PC) viene decrementato di 2 e l'istruzione viene ripetuta. Se B è andato a zero, l'istruzione è terminata. Notate che se B è settato a zero prima che l'istruzione venga eseguita, l'istruzione presenterà al dispositivo periferico 256 byte di dati.

Inoltre, le interruzioni saranno riconosciute e verranno eseguiti due cicli di refresh dopo ogni trasferimento dei dati.

Se  $B = 0$ :

CICLI M: 5      STATI T: 21 (4,5,3,4,5)      T.E. a 4 MHz: 5,25

Se  $B \neq 0$ :

CICLI M: 4      STATI T: 16 (4,5,3,4)      T.E. a 4 MHz: 4,00

### Bit di condizione alterati:

S:	Non definito
Z:	Settato
H:	Non definito
P/V:	Non definito
N:	Settato
C:	Inalterato

### Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 03H, il contenuto della coppia di registri HL è 1000H, e le locazioni di memoria hanno i seguenti contenuti:

Locazione	Contenuto
1000H	51H
1001H	A9H
1002H	03H

dopo l'esecuzione di OTIR la coppia di registri HL conterrà 1003H, il registro B conterrà zero, e nel dispositivo periferico di indirizzo 07H sarà stata scritta la sequenza di byte:

51H  
A9H  
03H

## OUTD

Operazione:  $(C) \leftarrow (HL)$ ,  $B \leftarrow B-1$ ,  $HL \leftarrow HL-1$

Formato: Codice operativo

OUTD

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

ED

1	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---

AB

Descrizione:

Il contenuto della coppia di registri HL è posto sul bus degli indirizzi per selezionare una locazione in memoria. Il byte contenuto in questa locazione di memoria viene temporaneamente memorizzato nella CPU. Poi, dopo che il contatore dei byte (B) è stato decrementato, il contenuto del registro C viene posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare un dispositivo di I/O tra le 256 porte possibili. Il valore del registro B, registro che può essere utilizzato come contatore dei byte da trasferire, è contemporaneamente posto sulla parte alta del bus degli indirizzi (da A8 a A15).

Sul bus dei dati viene quindi presentato il byte che deve essere scritto nel dispositivo periferico; da ultimo viene incrementata la coppia di registri HL.

CICLI M: 4      STATI T: 16 (4,5,3,4)      T.E. a 4 MHz: 4,00

Bit di condizione alterati:

- S: Non definito
- Z: Settato se  $B - 1 = 0$ ;  
altrimenti resettato
- H: Non definito
- P/V: Non definito
- N: Settato
- C: Inalterato

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 10H, il contenuto della coppia di registri HL è 1000H, e il contenuto della locazione di memoria 1000H è 59H, dopo l'esecuzione di

OUTD

il registro B conterrà 0FH, la coppia di registri HL conterrà OFFFH, e il byte 59H sarà stato scritto nel dispositivo periferico di I/O corrispondente all'indirizzo 07H.

## OTDR

Operazione:  $(C) \leftarrow (HL)$ ,  $B \leftarrow B-1$ ,  $HL \leftarrow HL-1$

Formato: Codice operativo

OTDR

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

ED

1	0	1	1	1	0	1	1
---	---	---	---	---	---	---	---

BB

Descrizione:

Il contenuto della coppia di registri HL è posto sul bus degli indirizzi per selezionare una locazione in memoria. Il byte contenuto in questa locazione di memoria è temporaneamente memorizzato nella CPU. Quindi, dopo che il contatore dei byte (B) è stato decrementato, il contenuto del registro C è posto sulla parte bassa (da A0 ad A7) del bus degli indirizzi per selezionare il dispositivo di I/O tra le 256 porte possibili. Il registro B può essere usato come contatore dei byte, e il suo valore decrementato è posto sulla parte alta (da A8 ad A15) del bus degli indirizzi. Poi il byte da mettere in uscita è posto sul bus di dati e scritto nel dispositivo periferico selezionato. Quindi la coppia di registri HL è decrementata, e se il registro decrementato B non è zero, il contatore di programma (PC) è decrementato di 2 e l'istruzione è ripetuta. Se B è andato a zero, l'istruzione è terminata. Si noti che, se B è settato a zero prima che l'istruzione sia eseguita, l'istruzione presenterà al dispositivo periferico 256 byte di dati. Inoltre, le interruzioni saranno riconosciute e verranno eseguiti due cicli di refresh dopo ogni trasferimento dei dati.

Se  $B \neq 0$ :

CICLI M: 5      STATI T: 21 (4,5,3,4,5)      T.E. a 4 MHz: 5,25

Se  $B = 0$ :

CICLI M: 4      STATI T: 16 (4,5,3,4)      T.E. a 4 MHz: 4,00

Bit di condizione alterati:

S:	Non definito
Z:	Settato
H:	Non definito
P/V:	Non definito
N:	Settato
C:	Inalterato

Esempio:

Se il contenuto del registro C è 07H, il contenuto del registro B è 03H, il contenuto della coppia di registri HL è 1000H, e le locazioni di memoria hanno i seguenti contenuti:

Locazione	Contenuto
0FFEH	51H
0FFFH	A9H
1000H	03H

dopo l'esecuzione di

OTDR

la coppia di registri HL conterrà 0FFDH, il registro B contiene zero, e nel dispositivo di indirizzo 07H sarà stata scritta la sequenza di byte:

03H  
A9H  
51H

FLAGS

## Indicatori di stato del microprocessore Z 80 (FLAG)

Il registro dei flag (F ed F') fornisce all'utilizzatore delle informazioni sullo stato dello Z-80 in un certo istante di tempo.

Di seguito è data la posizione dei bit di ciascun flag:

dove

C =	RIPORTO (CARRY)
N =	SOMMA/SOTTRAZIONE (ADD/SUBTRACT)
P/V =	PARITÀ/SOVRACCARICO (PARITY/OVERFLOW)
H =	RIPORTO PARZIALE (HALF CARRY)
Z =	ZERO
S =	SEGNO (SIGN)
X =	NON UTILIZZATO

ciascuno dei due registri di flag dello Z-80, contiene 6 bit di informazione di stato. Tali bit sono settati o resettati dalle operazioni della CPU.

(I bit 3 e 5 non sono utilizzati).

Allo stato di uno tra i quattro flag C, P/V, Z e S fanno riferimento le istruzioni di salto condizionato, chiamata condizionata di subroutine e ritorno condizionato; gli altri due flag invece (H e N) sono utilizzati solo per l'esecuzione di operazioni in aritmetica BCD.

### Flag di CARRY (C)

Il flag di Carry è in generale un indicatore di riporto.

Viene utilizzato da alcuni tipi di istruzione e, puramente, alcuni tipi di istruzione possono alterarne lo stato.

Nel caso dell'istruzione di somma ADD, per esempio, lo stato del Carry sarà uguale a 1 se l'esecuzione dell'istruzione ha generato un riporto, sarà uguale a 0 se viceversa non è stato generato riporto.

Un comportamento simile si ha per l'istruzione di sottrazione SUB, dove il flag di Carry, in questo caso, avrà il significato di riporto negativo (borrow, cioè prestito) proprio delle operazioni decimali usuali.

Considerando che una sottrazione è in effetti effettuata dalla CPU come un'operazione di somma tra il minuendo e il complemento a 2 della sottraendo, si può osservare che il riporto negativo sarà il complemento del riporto generato dalla somma accennata. Le istruzioni di confronto, essendo di fatto delle operazioni di sottrazione che non gene-

rano risultato, opereranno sul flag di Carry in modo identico alle istruzioni di sottrazione.

L'istruzione di somma con Carry ADC (e in modo del tutto simile l'istruzione di sottrazione con Carry SBC), oltre ad alterare lo stato del flag in modo identico a quanto detto per l'istruzione ADD, utilizza il Carry come operando ed il risultato dell'operazione sarà così la somma dei due addendi e del Carry.

Grazie a queste caratteristiche del flag C diventa possibile realizzare routines matematiche che operino su dati di dimensione superiori al byte.

Sempre tra le istruzioni aritmetiche l'esecuzione di DAA comporterà l'alterazione del Carry se durante l'operazione di aggiustamento decimale verrà generato un riporto. Una estensione del significato del flag di riporto sia ha nel caso delle istruzioni del gruppo di shift e di rotazione alla descrizione delle quali si rimanda.

Infine, il flag di Carry può essere alterato direttamente utilizzando le istruzioni SCF, che pone lo stato del flag a 1, e CCF che complementa lo stato dello stesso.

### Flag di ADD/SUBTRACT (N)

Questo flag è usato dall'istruzione di aggiustamento decimale dell'accumulatore (DAA) per distinguere tra le istruzioni di somma e di sottrazione.

Dopo l'esecuzione di una operazione di somma, N si troverà nello stato 0, mentre dopo una sottrazione N si troverà nello stato 1.

### Flag di PARITY/OVERFLOW

Il flag di Parity/Overflow può essere alterato o meno nel corso dell'esecuzione di alcuni tipi di istruzione a seconda del risultato dell'operazione.

Per le operazioni aritmetiche, questo flag indica una condizione di overflow quando il risultato nell'accumulatore è maggiore di +127 o minore di -128.

Questa condizione di overflow può essere prevista esaminando i bit di segno degli operandi ed eventualmente del risultato.

Nel caso di somme, operandi con segni differenti non daranno mai luogo ad overflow.

Nel caso di somma tra due numeri di stesso segno con risultato di segno differente, il flag di overflow è settato.

Ad esempio:

$$\begin{array}{r}
 +120 = 0111\ 1000 \quad \text{ADDENDO} \\
 +105 = 0110\ 1001 \quad \text{ADDENDO} \\
 \hline
 +225 = 1110\ 0001 \quad (-95) \text{ SOMMA}
 \end{array>$$

I due numeri sommati tra loro danno un numero superiore a +127 e dai due operandi positivi risulta un numero negativo (-95), cosa questa certamente errata.

Il flag di overflow è quindi settato.

Nel caso di sottrazione, può verificarsi overflow nel caso di operandi di segno opposto. Operandi dello stesso segno non danno mai overflow.

Ad esempio:

$$\begin{array}{r}
 +127 \quad 0111 \ 1111 \quad \text{MINUENDO} \\
 (-) -64 \quad 1100 \ 0000 \quad \text{SOTTRAENDO} \\
 \hline
 +191 \quad 1011 \ 1111 \quad \text{DIFFERENZA}
 \end{array}$$

I due operandi sono di segno opposto: il minuendo positivo e il sottraendo negativo. Il risultato scorretto ottenuto ha il bit 7 di segno uguale a 1, a indicare un valore negativo.

La condizione segni opposto a minuendo e risultato indica appunto una condizione di overflow. Un altro modo per prevedere una condizione di overflow consiste nell'osservare il riporto che entra come operando nell'operazione binaria tra il bit di segno e quello che ne esce come risultato dalla stessa. Secondo tale metodo sono indicazioni di overflow sia il fatto che il riporto "entri" ma non "esca" sia il viceversa.

Questo flag è anche utilizzato nel caso di operazioni logiche e di rotazione, per indicare la parità del risultato.

Viene contato il numero di bit nello stato 1 presenti nel byte.

Se il totale è dispari, è indicata una parità dispari, «ODD» ( $P=0$ ).

Se il totale è pari, è indicata una parità dispari, «EVEN» ( $P=1$ ).

Durante le istruzioni di ricerca (CPI, CPIR, CPD, CPDR) e trasferimento blocchi (LDI, LDIR, LDD, LDDR) il flag P/V è legato allo stato del registro byte counter (BC). Durante un decremento, se il byte counter va a 0, allora il flag è posto a 0, altrimenti il flag è settato a 1. Con le istruzioni LD A, I e LD A, R il flag P/V rispecchierà lo stato del flip-flop di interrupt enable (IFF2) perché possa essere eventualmente memorizzato o testato.

Quando è introdotto un byte da una periferica mediante l'istruzione IN, r, (C), il flag sarà aggiustato in modo da indicare la parità del dato.

#### Flag di HALF CARRY (H)

Questo flag sarà settato o resettato a seconda che in istruzione aritmetica a 8 bit venga o meno generato un riporto (carry) o in prestito (borrow) tra i bit 3 e 4.

Questo flag è usato dall'istruzione di aggiustamento decimale dell'accumulatore (DAA) per correggere il risultato nel caso di operazione di somma o sottrazione su dati BCD compattati (packed) cioè con due cifre BCD in ogni byte.

La tabella seguente mostra le condizioni secondo cui il flag H è settato o resettato:

H	SOMMA	SOTTRAZIONE
1	Vi è un riporto dal bit 3 al bit 4	Vi è un riporto negativo dal bit 4
0	Non c'è riporto dal bit 3 al bit 4	Non c'è un riporto negativo dal bit 4

#### Flag di ZERO (Z)

Il flag di zero è alterato in conformità col fatto che il risultato generato da determinate operazioni sia o meno uguale a zero.

Nel caso di istruzioni aritmetiche o logiche che operano su dati a 8 bit, il flag Z sarà settato ad '1' se il byte contenuto nell'Accumulatore ad operazione completata è zero. Se il byte è diverso da zero, il flag Z sarà resettato a 0.

Nel caso di istruzioni di confronto (ricerca) il flag Z sarà settato ad 1 nel caso di corrispondenza tra il valore dato in Acc. e quello nella locazione di memoria puntata dalla coppia di registri HL.

Nelle operazioni che testano lo stato di un bit di un registro o di una locazione di memoria, il flag Z conterrà lo stato complementato del bit indicato (si veda l'istruzione BIT b, s).

Nei trasferimenti di byte tra una locazione di memoria ed un dispositivo di I/O (INI; IND; OUTI e OUTD) se il risultato di B-1 è zero, il flag Z è settato, altrimenti è resettato.

Ancora, nel caso di lettura di un byte da un dispositivo di I/O effettuata attraverso l'istruzione IN r, (C) il flag Z è settato per indicare l'input di un byte zero.

#### Flag di SEGNO (S)

Il flag di segno (S) memorizza lo stato del bit più significativo dell'acc. (bit 7).

Quando lo Z 80 realizza un'operazione aritmetica su numeri con segno, si usa la notazione binaria complemento a due. Un numero positivo è identificato con «0» nel bit 7. Un numero negativo è identificato da un «1». Nei bit da 0 a 6 è posto il valore binario del numero che si vuole rappresentare se esso è positivo (da 0 a 127), mentre viene posto il complemento a due del valore binario del numero se esso è negativo (da -1 a -128).

Nell'operazione di trasferimento di un byte da un dispositivo di I/O a un registro (istruzione IN r, (C)), il flag S indicherà il segno del dato trasferito.

<b>INDICE MNEMONICO DELLE ISTRUZIONI</b>	<b>OPERAZIONI</b>	<b>pag.</b>
ADC HL,ss	Somma la coppia di Reg. ss e il Carry a HL .....	115
ADC A,s	Somma l'Oper. s e il Carry all'Acc. ....	76
ADD A,n	Somma il numero n all'Acc. ....	72
ADD A,r	Somma il Reg. r all'Acc. ....	70
ADD A,(HL)	Somma la locaz. (HL) all'Acc. ....	73
ADD A,(IX+d)	Somma la locaz. (IX+d) all'Acc. ....	74
ADD A,(IY+d)	Somma la locaz. (IY+d) all'Acc. ....	75
ADD HL,ss	Somma la coppia di Reg. ss a HL .....	114
ADD IX,pp	Somma la coppia di Reg. pp a IX .....	117
ADD IY,rr	Somma la coppia di Reg rr a IY .....	118
AND s	"AND" logico tra l'Oper. s e l'Acc. ....	82
BIT b,(HL)	Test del BIT b di locaz. (HL) .....	161
BIT b,(IX+d)	Test del BIT b di locaz. (IX+d) .....	162
BIT b,(IY+d)	Test del BIT b di locaz. (IY+d) .....	164
BIT b,r	Test del BIT b del Reg. r .....	160
CALL cc,nn	Chiama la subroutine di locaz. nn se la condizione cc è vera .....	194
CALL nn	Chiama la subroutine di locaz. nn in modo incondiziona- to .....	192
CCF	Complementa il flag di Carry .....	103
CP s	Confronta l'Oper. s con l'Acc. ....	88
CPD	Confronta la locaz. (HL) con l'Acc., decrementa HL, BC .....	65
CPDR	Confronta la locaz. (HL) con l'Acc., decrementa HL, BC; ripeti finchè BC=0 .....	66
CPI	Confronta la locaz. (HL) con l'Acc., incrementa HL, decrementa BC .....	62
CPIR	Confronta la locaz. (HL) con l'Acc., incrementa HL, decrementa BC; ripeti finchè BC=0 .....	63
CPL	Complementa a 1 l'Acc. ....	100
DAA	Aggiustamento decimale dell'Acc. ....	98
DEC m	Decrementa l'Oper. m .....	94
DEC IX	Decrementa IX .....	123
DEC IY	Decrementa IY .....	124
DEC ss	Decrementa la coppia di Reg. ss .....	122
DI	Disabilita le interruzioni .....	107
DJNZ e	Decrementa B e salta a PC+e se B=0	188
EI	Abilita le interruzioni .....	108
EX (SP),HL	Scambia le locaz. (SP) e (SP+1) con HL .....	53
EX (SP),IX	Scambia le locaz. (SP) e (SP+1) con IX .....	54
EX (SP),IY	Scambia le locaz. (SP) e (SP+1) con IY .....	55
EX AF, AF'	Scambia AF con AF' .....	51
EX DE,HL	Scambia DE con HL .....	50

EXX	Scambia BC,DE,HL rispettivamente con BC',DE',HL' .	52			
HALT	HALT (attesa di interruzione o reset) .	106	LD IX,nn	Carica il valore nn in IX .	29
IM 0	Setta l'interruzione nel modo 0 .	109	LD IX,(nn)	Carica le locaz. (nn) e (nn+1) in IX .	33
IM 1	Setta l'interruzione nel modo 1 .	110	LD (IX+d),n	Carica il valore n nella locaz. (IX+d) .	15
IM 2	Setta l'interruzione nel modo 2 .	111	LD (IX+d),r	Carica il Reg. r nella locaz. (IX+d) .	12
IN A,(n)	Carica la porta d'ingresso (n) nell'Acc.	204	LD IY,nn	Carica il valore nn in IY .	30
IN r,(C)	Carica la porta d'ingresso (c) nel Reg. r .	205	LD IY,(nn)	Carica le locaz. (nn) e (nn+1) in IY .	34
INC (HL)	Incrementa la locaz. (HL) .	91	LD (IY+d),n	Carica il valore n nella locaz. (IY+d) .	16
INC IX	Incrementa IX .	120	LD (IY+d),r	Carica il Reg. r nella locaz. (IY+d) .	13
INC (IX+d)	Incrementa la locaz. (IX+d) .	92	LD (nn),A	Carica l'Acc. nella locaz. (nn) .	22
INC IY	Incrementa IY .	121	LD (nn),dd	Carica la coppia di Reg. dd nelle locaz. (nn) e (nn+1) .	36
INC (IY+d)	Incrementa la locaz. (IY+d) .	93	LD (nn),HL	Carica HL nelle locaz. (nn) e (nn+1) .	35
INC r	Incrementa il Reg. r .	90	LD (nn),IX	Carica IX nelle locaz. (nn) e (nn+1) .	37
INC ss	Incrementa la coppia di Reg. ss .	119	LD (nn),IY	Carica IY nelle locaz. (nn) e (nn+1) .	38
IND	Carica la porta d'ingresso (C) nella locaz. (HL), decremente HL, B .	210	LD R,A	Carica l'Acc. in R .	26
INDR	Carica la porta d'ingresso (C) nella locaz. (HL), decremente HL, B; ripeti finchè B = 0 .	211	LD r,(HL)	Carica la locaz. (HL) nel Reg. r .	8
INI	Carica la porta d'ingresso (C) nella locaz. (HL), incrementa HL, decrementa B .	207	LD r,(IX+d)	Carica la locaz. (IX+d) nel Reg. r .	9
INIR	Carica la porta d'ingresso (C) nella locaz. (HL), incrementa HL, decrementa B; ripeti finchè B=0 .	208	LD r,(IY+d)	Carica la locaz. (IY+d) nel Reg. r .	10
JP (HL)	Salta a (HL) in modo incondizionato .	185	LD r,n	Carica il valore n nel Reg. r .	7
JP (IX)	Salta a (IX) in modo incondizionato .	186	LD r,r'	Carica il Reg. r' nel Reg. r .	6
JP (IY)	Salta a (IY) in modo incondizionato .	187	LD SP,HL	Carica HL in SP .	39
JP cc,nn	Salta a nn se la condizione cc è vera .	175	LD SP,IX	Carica IX in SP .	40
JP nn	Salta a nn in modo incondizionato .	174	LD SP,IY	Carica IY in SP .	41
JR C,e	Salta a PC+e se flag di Carry=1 .	178	LDD	Carica la locaz. (HL) nella locaz. (DE), decremente DE,HL,BC .	59
JR e	Salta a PC+e in modo incondizionato .	177	LDDR	Carica la locaz. (HL) nella locaz. (DE), decrementa DE,HL,BC; ripeti finchè BC=0 .	60
JR NC,e	Salta a PC+e se flag di Carry = 0 .	180	LDI	Carica la locaz. (HL) nella locaz. (DE), incrementa DE,HL, decrementa BC; ripeti finchè BC=0 .	56
JR NZ,e	Salta a PC+e se flag di Zero = 0 .	183	LDIR	Complementa a 2 l'Acc. .	101
JR Z,e	Salta a PC+e se flag di Zero=1 .	181	NEG	Nessuna operazione .	105
LD A,(BC)	Carica la locaz. (BC) nell'Acc. .	17	NOP	"OR" logico tra l'Oper. s e l'Acc. .	84
LD A,(DE)	Carica la locaz. (DE) nell'Acc. .	18	OR s	Carica la locaz. (HL) nella porta d'uscita (C), decremente HL, B; ripeti finchè B=0 .	219
LD A,I	Carica I nell'Acc. .	23	OTDR	Carica la locaz. (HL) nella porta d'uscita (C), incrementa HL, decremente B; ripeti finchè B=0 .	216
LD A,(nn)	Carica la locaz. (nn) nell'Acc. .	19	OTIR	Carica la locaz. (HL) nella porta d'uscita (C) .	214
LD R,A	Carica R nell'Acc. .	26	OUT (C),r	Carica il Reg. r nella porta d'uscita (C) .	213
LD (BC),A	Carica l'Acc. nella locaz. (BC) .	20	OUT (n),A	Carica l'Acc. nella porta d'uscita (n) .	218
LD (DE),A	Carica l'Acc. nella locaz. (DE) .	21	OUTD	Carica la locaz. (HL) nella porta d'uscita (C), decremente HL, B .	215
LD (HL),n	Carica il valore n nella locaz. (HL) .	14	OUTI	Carica la locaz. (HL) nella porta d'uscita (C), incrementa HL, decremente B .	215
LD dd,nn	Carica il valore nn nella coppia di Reg. dd .	28	POP IX	Carica la porta alta dello stack in IX .	46
LD dd,(nn)	Carica le locaz. (nn) e (nn+1) nella coppia di Reg. dd .	32	POP IY	Carica la porta alta dello stack in IY .	47
LD HL,(nn)	Carica le locaz. (nn) e (nn+1) in HL .	31			
LD (HL),r	Carica il Reg. r nella locaz. (HL) .	11			
LD I,A	Carica l'Acc. in I .	25			

POP qq	Carica la parte alta dello stack nella coppia di Reg. qq	45
PUSH IX	Carica IX nella parte alta dello stack .....	43
PUSH IY	Carica IY nella parte alta dello stack .....	44
PUSH qq	Carica la coppia di Reg. qq nella parte alta dello stack	42
RES b,m	Reset del bit b dell'Oper. m.....	170
RET	Ritorno da subroutine in modo incondizionato .....	196
RET cc	Ritorno da subroutine se la condizione cc è vera .....	197
RETI	Ritorno dall'interruzione in modo incondizionato .....	199
RETN	Ritorno dall'interruzione non mascherabile in modo incondizionato .....	201
RL m	Ruota a sinistra attraverso il Carry l'Oper. m	137
RLA	Ruota a sinistra attraverso il Carry l'Acc. ....	127
RLC (HL)	Ruota a sinistra in modo circolare la locaz. (HL) .....	132
RLC (IX+d)	Ruota a sinistra in modo circolare la locaz. (IX+d) .....	133
RLC (IY+d)	Ruota a sinistra in modo circolare la locaz. (IY+d) ..	135
RLC r	Ruota a sinistra in modo circolare il Reg. r .....	130
RLCA	Ruota a sinistra in modo circolare l'Acc. ....	126
RLD	Ruota a sinistra in modo circolare il digit di ordine bas so dell'Acc. con i due digit della locaz. (HL) .....	155
RR m	Ruota a destra attraverso il Carry l'Oper. m .....	143
RRA	Ruota a destra attraverso il Carry l'Acc. ....	129
RRC m	Ruota a destra in modo circolare l'Oper. m .....	140
RRCA	Ruota a destra in modo circolare l'Acc. ....	128
RRD	Ruota a destra in modo circolare il digit di ordine basso dell'Acc. con i due digit della locaz. (HL) .....	157
RST p	Riparti (Restart) dalla locaz. p .....	202
SBC A,s	Sottrai l'Oper. s e il Carry dall'Acc. ....	80
SBC HL,ss	Sottrai la coppia di Reg. ss e il Carry da HL .....	116
SCF	Set del flag di Carry (C=1) .....	104
SET b,(HL)	Set del BIT b di locaz. (HL) .....	167
SET b,(IX+d)	Set del BIT b di locaz. (IX+d) .....	168
SET b,(IY+d)	Set del BIT b di locaz. (IY+d) .....	169
SET b,r	Set del BIT b del Reg. r .....	166
SLA m	Shift aritmetico a sinistra dell'Oper. m .....	146
SRA m	Shift aritmetico a destra dell'Oper. m .....	149
SRL m	Shift logico a destra dell'Oper. m .....	152
SUB s	Sottrai l'Oper. s dall'Acc. ....	78
XOR s	"OR" esclusivo tra l'Oper. s e l'Acc. ....	86

Information furnished is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor  
for an infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise  
under any patent or patent rights of SGS-ATES. This publication supersedes and substitutes all information previously supplied.

**SGS – ATES GROUP OF COMPANIES**  
**Italy – France – Germany – Singapore – Sweden – United Kingdom – U.S.A.**  
 Printed in Italy by permission of Zilog Inc.

**SGS-ATES COMPONENTI ELETTRONICI SpA**

**DIREZIONE GENERALE E  
AMMINISTRATIVA**

Via C. Olivetti, 2  
**20041 Agrate Brianza**  
Tel.: 65.03.41  
Telex: 330141-330131

**DIREZIONE COMMERCIALE ITALIA**

Via Correggio, 1/3  
**20149 Milano**  
Tel.: 46.95.651

**PUNTO DI VENDITA**

Via Larga  
ang. V. Brolo/V. Verziere  
**20122 Milano**  
Tel. 86.90.047

**UFFICI VENDITA**

Via G. Del Pian dei Carpini, 96/1

**50127 Firenze**  
Tel.: 43.77.763

Via Correggio, 1/3  
**20149 Milano**  
Tel.: 46.95.651

Piazza Gondar, 11

**00199 Roma**  
Tel.: 83.92.848

Via La Loggia, 51/7  
**10134 Torino**  
Tel.: 63.45.72

**DISTRIBUTORI E RIVENDITORI IN ITALIA**

Giulio Ballarin  
Via Japelli, 9  
**35100 Padova**  
Tel.: 65.45.00

DE DO Electronic Fitting  
Via Trieste, 26  
**64019 Tortoreto Lido (TE)**  
Tel.: 78.134

Diselco S.p.A.  
Via Dei Cignoli, 9  
**20151 Milano**  
Tel.: 30.86.141

Electronic System S.r.l.  
Via Davanzati, 39  
**00137 Roma**  
Tel.: 82.72.860/82.74.610

Fanton di M. Lago & C. S.a.s.  
Via Venezia, 106  
**35100 Padova**  
Tel.: 65.50.33

G.B.C. Italiana S.p.A.  
Via Matteotti, 66  
**20092 Cinisello B. (MI)**  
Tel. 61.89.391/61.81.801

Marcucci S.p.A.  
Via Cadore, 24  
**20135 Milano**  
Tel.: 57.64.16

Carlo Restelli  
Via Nizza, 34  
**10125 Torino**  
Tel.: 65.57.65

B. Rossi S.p.A.  
Via Archimede, 43/2  
**16142 Genova**  
Tel. 51.65.75

Adriano Zaniboni  
Via Tasso, 13/4  
**40129 Bologna**  
Tel. 36.89.13

L. 6.000  
(IVA compresa)