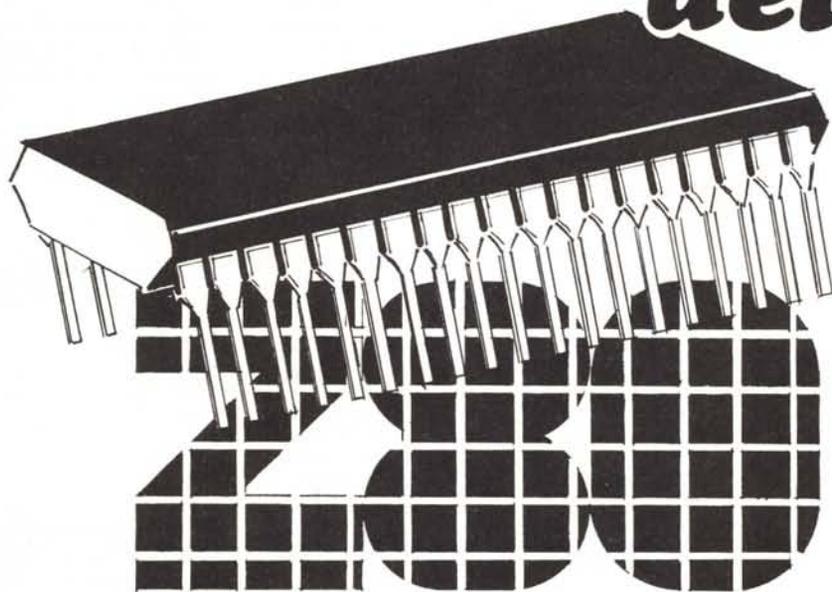


# L'ASSEMBLER dello



## I flag e le operazioni aritmetiche

di Pierluigi Panunzi

Abbiamo visto nelle precedenti puntate tutte le possibili variazioni di un'istruzione molto importante, la *LOAD (LD)*, importante perché come visto ci permette di trasferire da una generica sorgente ad una altrettanto generica destinazione dei dati, rappresentati da singoli byte o da coppie di byte (*word*), rispettivamente ad 8 e a 16 bit.

È chiaro che però la sola istruzione di spostamento non può bastare, in quanto non permette assolutamente di "manipolare" i dati di ingresso: ecco che infatti sono presenti parecchie istruzioni che ci consentono di gestire bit per bit il nostro bravo byte o word che sia, per effettuare, con relativa semplicità, le più svariate operazioni.

A questo punto magari i meno esperti si aspetterebbero, chissà, l'istruzione che effettua la radice quadrata, oppure il calcolo del seno di un certo angolo, influenzati dalle seppur piccole capacità della più infima delle calcolatrici tascabili.

Invece rimarranno delusi. Nello Z80 (e in tutti gli altri microprocessori ad 8 bit, tipo il 6502 e derivati, il 6800, l'8085, ecc) non esistono nemmeno le istruzioni di moltiplicazione e divisione!

Quasi sopraffatti dai "muggiti di disapprovazione" da parte dei meno esperti di cui sopra, diciamo che solo nei micro a 16 bit si cominciano a vedere istruzioni così complesse (!), quali la moltiplicazione e la divisione (vedasi ad esempio l'8088, il 68000, lo Z8000, ecc), mentre per arrivare alle fun-

zioni più elevate (trigonometriche, logaritmiche, ecc) bisogna passare ai cosiddetti co-processori aritmetici, un esempio dei quali è il famoso 8087, famoso non tanto perché usato, quanto perché in svariati personal a 16 bit c'è sempre uno zoccolo libero predisposto per tale integrato: il suo funzionamento però richiede una complessa programmazione in assembler 8086-8088 e cioè in ambiente MS-DOS o CP/M-86.

Comunque non distraiamoci con altre nozioni e torniamo al nostro Z80, per parlare innanzitutto del registro dei Flag e delle funzioni dei singoli bit di Flag.

Non è il caso di spaventarsi se alcune terminologie non saranno subito di facile comprensione: dei flag si parlerà più e più volte in questa e nelle prossime puntate.

### I Flag

Consideriamo il significato dei bit costituenti il registro di "flag" (F) che sono:

S, Z, X, H, X, P/V, N, C

Innanzitutto le "X" rappresentano bit non utilizzati, mentre per gli altri abbiamo che:

— C è il flag di Carry o riporto, che vale 1 se un'operazione di somma ha generato un riporto o se una sottrazione ha richiesto un prestito.

— N è il flag di "Add/subtract" e vale 1 se l'operazione effettuata è stata una sottrazione.

— P/V è il flag di "Parity/Overflow" ed è un flag condiviso da una serie di istruzioni: quelle logiche lo usano come bit di parità mentre quelle aritmetiche come bit di overflow.

— H è il flag di "Half Carry" ed indica, se pari ad 1, che c'è stato riporto tra il bit 3 e il bit 4 durante un'operazione aritmetica e serve per le operazioni di dati espressi in BCD (ne ripareremo in seguito...).

— Z è il flag di "Zero" che (attenzione!) vale 1 se il risultato di un'operazione è 0 e vale "0" se il risultato è "diverso da zero" (sembra un gioco di parole, ma non lo è)

— S infine è il flag di "Sign" ed indica il "segno" del risultato di una certa operazione ed è utile nel caso si lavori in logica complementata.

### Le operazioni aritmetiche

Ora che bene o male sappiamo che esistono i flag, li troveremo praticamente in tutte le istruzioni logico-aritmetiche.

Iniziamo dalle più semplici, delle quali abbiamo già parlato: l'incremento ed il decremento di un registro e di una coppia di registri.

Cominciamo dalla coppia di registri perché più semplice: sia che si tratti di incremento che di decremento, l'operazione che

INCREMENTO DI BYTE (INC)						
Valori		Flag				
Iniziale	Finale	C	Z	S	N	
00	01	/	0	0	0	0
7F	80	/	0	1	0	0
EE	EF	/	0	1	0	0
FF	00	/	1	0	0	0

DECREMENTO DI BYTE (DEC)						
Valori		Flag				
Iniziale	Finale	C	Z	S	N	
20	1F	/	0	0	1	1
01	00	/	1	0	1	1
00	FF	/	0	1	1	1
C4	C3	/	0	1	1	1
80	7F	/	0	0	1	1

**Tabella 1**

In questa e nelle altre tabelle sono riportati alcuni esempi delle varie istruzioni e i valori dei flag più importanti.

coinvolge un valore a 16 bit NON cambia lo stato di alcuno dei bit: in gergo si dice che "non affetta alcun flag".

(Attenzione, "affetta" si dice con la "e" aperta, dal momento che i flag non sono salami: deriva dall'inglese "affected").

Questo fatto di non affettare i flag, a detta di molti inspiegabile, obbliga il programmatore a dei trucchi per verificare il contenuto della coppia: ma ci spiegheremo meglio dopo.

Nel caso dei byte si può dire che tutti i flag meno uno (il carry) sono alterati.

In particolare l'incremento azzerava il flag N mentre viceversa il decremento lo pone ad 1: infatti se si pensa bene al significato di N, l'incremento è una somma mentre il decremento è una sottrazione.

Detto dunque che il carry (C) non viene alterato, vediamo come si comportano gli altri flag, in base al risultato dell'operazione.

In particolare il flag Z sarà posto ad 1 se il risultato dell'incremento o del decremento è 0.

Nel caso dell'incremento, il flag Z pari ad 1 significa che il contenuto era FFH, mentre un decremento con risultato pari a zero significa che il contenuto era 01H (ovvio, no?!).

Per quanto riguarda il flag S, questo è pari al bit più significativo del risultato, mentre il flag P/V in entrambi i casi ha il significato di "V" e cioè segnala un overflow (ma praticamente non è mai usato in questo caso). Tutto questo l'abbiamo sintetizzato in tabella 1, dove, in base al valore iniziale, si ottiene quello finale e l'alterazione dei vari flag (quelli più importanti perché più usati e cioè C, Z, S, ed N).

Passiamo ora alle Somme ed alle Sottrazioni.

### Addizioni e addizioni con il Carry

Si può effettuare la somma tra due byte di cui uno posto in accumulatore (A) e l'altro posto o in un registro qualsiasi (ADD A,B, ed anche ADD A,A) oppure

come valore immediato (ADD A,5) oppure nella locazione di memoria puntata dalla coppia HL o dai registri indice IX ed IY con uno spostamento (d), rispettivamente

ADD A,(HL) ADD A,(IX + d) ADD A,(IY + d).

Dato che una somma può comportare un riporto, in questo caso verrà settato il flag C (carry) ed inoltre si può effettuare una "somma con riporto" cioè tenendo in conto la presenza o meno del riporto (carry) da operazioni precedenti.

Facendo riferimento alla tabella 2 vediamo che negli ultimi tre casi si genera un riporto in quanto la somma finale supera il

SOMMA DI BYTE (ADD ADC)						
Addendi		Somma	Flag			
			C	Z	S	N
20	40	60	0	0	0	0
35	7E	B3	0	0	1	0
F0	54	44	1	0	0	0
FF	FF	FE	1	0	1	0
FF	01	00	1	1	0	0

SOMMA DI WORD (ADD)						
Addendi		Somma	Flag			
			C	Z	S	N
0100	5555	5655	0	/	/	0
FFFF	0000	FFFF	0	/	/	0
FFFF	0001	0000	1	/	/	0

SOMMA CON RIPORTO WORD (ADC)							
Addendi		Rip.	Somma	Flag			
				C	Z	S	N
0100	5555	1	5656	0	0	0	0
FFFF	0000	0	FFFF	0	0	1	0
FFFF	0000	1	0000	1	1	0	0
FFFF	FFFF	1	FFFF	1	0	1	0

**Tabella 2**

valore FFH (255) ed anche nel caso peggiore (somma di FFH e FFH) tale riporto vale 1 ed è posto nel Carry.

Per quanto riguarda la somma di word, cioè di valori a 16 bit, si possono avere solo le seguenti possibilità relative alla somma (ADD) e alla somma che tiene conto di un riporto precedente (ADC): un operando è il contenuto di HL e l'altro può essere il contenuto delle coppie BC, DE e HL oppure dello Stack Pointer SP; il risultato si avrà nella coppia HL.

Altri casi possibili sono quelli che prevedono un operando nel registro indice IX e l'altro in BC, DE, IX, SP, con risultato in IX e quelli con un operando in IY, il secondo in BC, DE, IY, SP e con il risultato in IY.

Come si vede non si possono sommare direttamente HL e uno dei due registri indice IX e IY, come neppure questi ultimi due tra loro.

Come da tabella 2 si ha che la somma (ADD) di valori a 16 bit non affetta il flag di zero, né quello di segno, in quanto ora si ha a che fare con 2 byte e non più con uno solo: è analogo al caso già visto dell'incremento e decremento di coppie di byte.

Viceversa la somma con carry può alterare i flag Z ed S mentre in ogni caso il carry tiene conto di eventuali riporti (il caso peggiore è questa volta la somma di FFFFH con FFFFH che dà FFFEH con carry ad 1).

### Sottrazioni e Sottrazioni con Riporto

Analogamente al caso delle somme, nel caso di sottrazione di byte si può tenere o meno in conto il riporto da un'operazione precedente: evidentemente si tratterà di un "prestito" che, se presente, farà decrementare di uno la differenza.

Anche in questo caso valgono regole

SOTTRAZIONE TRA BYTE (SUB SBC)						
Operandi		Differ.	Flag			
			C	Z	S	N
50	30	20	0	0	0	1
F5	10	E5	0	0	1	1
F5	F5	00	0	1	0	1
10	20	F0	1	0	1	1

**Tabella 3**

identiche a quella della somma, riguardanti la provenienza degli operandi: uno in accumulatore e l'altro in un registro (anche A stesso!) ecc.

Detto che in questo caso i nomi simbolici sono rispettivamente SUB e SBC, andiamo a vedere nella tabella 3 le caratteristiche riguardanti i flag: il flag N è sempre 1 e nell'ultimo esempio la differenza tra 10H e 20H richiede un prestito, fatto questo segnalato da un 1 nel bit C.

Per quanto riguarda invece la sottrazione a 16 bit il minuendo deve per forza essere in HL mentre il sottraendo può stare solo in quattro coppie: in BC, DE, HL o SP e l'operazione avviene sempre tenendo in conto la presenza di un carry proveniente da una precedente operazione.

Il comportamento dei flag questa volta è alquanto scontato e non rimane altro da aggiungere che l'istruzione in merito si chiama SBC (dove la "C" ricorda che al risultato viene sottratto uno se "prima" di tale operazione il carry era settato, come negli altri casi già visti di SBC relativi a byte).

### Le operazioni logiche

Veniamo ora ad altre importantissime operazioni, che consentono di effettuare l'AND e l'OR, l'OR esclusivo e la Comparazione tra byte: in questo caso non esistono le rispettive a 16 bit ed in tutti e quattro i casi il primo operando è l'accumulatore ed il secondo segue le regole già viste nel caso dell'addizione, e della sottrazione.

A parte la Comparazione, della quale parliamo più tardi, le tre restanti funzioni logiche (le cui istruzioni si chiamano AND, OR e XOR rispettivamente), operano, sui due operandi, bit a bit e cioè se ad

AND (AND)						
Accum.	Oper.	Risult.	Flag			
			C	Z	S	N
10	73	10	0	0	0	1
38	00	00	0	1	0	1
DA	8B	8A	0	0	1	1
AA	55	00	0	1	0	1

OR (OR)						
Accum.	Oper.	Risult.	Flag			
			C	Z	S	N
10	73	73	0	0	0	0
DA	8B	DB	0	0	1	0
00	00	00	0	1	0	0

OR ESCLUSIVO (XOR)						
Accum.	Oper.	Risult.	Flag			
			C	Z	S	N
10	73	63	0	0	0	0
AA	55	FF	0	0	1	0
C7	C7	00	0	1	0	0
05	FF	FA	0	0	1	0
14	00	14	0	0	0	0

**Tabella 4**

esempio abbiamo l'AND tra l'accumulatore ed il registro E, il risultato (che va in accumulatore) avrà come bit 0 l'AND tra i bit 0 di A ed E, come bit 1 l'AND tra i bit 1 di A ed E e così via. I flag in questi tre casi varieranno in base al risultato e secondo considerazioni analoghe a quelle già fatte nel caso delle precedenti operazioni.

In particolare in tabella 4 possiamo ve-

COMPARAZIONE (CP)					
Accumul.	Operan.	Flag			
		C	Z	S	N
77	50	0	0	0	1
77	77	0	1	0	1
77	78	1	0	1	1
00	00	0	1	0	1

**Tabella 5**

dere alcuni esempi per tutte e tre le operazioni logiche: in tutti e tre i casi il flag C viene posto a 0, mentre l'N è posto ad 1 solo per l'AND e a 0 per l'OR e lo XOR.

È molto importante ricordarsi che il carry viene resettato!

Da notare il fatto che l'AND di un qualsiasi valore con 0 dà 0, mentre per avere il risultato di un OR pari a 0 i due operandi devono essere entrambi nulli.

Questo fatto è molto importante e si utilizza di solito in unione al decremento di una coppia di registri, che, come visto, non altera di per sé il flag Z: ne parleremo comunque nelle prossime puntate.

Passiamo ora alla Comparazione (CP) che avviene tra l'accumulatore ed un operando posto secondo le ormai solite regole.

A differenza di tutte le operazioni viste sinora, la "Compare" effettua praticamente la differenza tra l'accumulatore ed il secondo operando ma non pone da nessuna parte il risultato: cambieranno (e questo è fondamentale) soltanto i flag.

Vediamo nella tabella 5 il comportamento dei tre flag C, Z, S, dato che N è posto ad 1 (infatti si fa una sottrazione): nelle prossime puntate vedremo che la CP, l'OR e l'AND saranno molto utili per i salti condizionati. In particolare abbiamo che, tenendo fisso il valore in accumulatore, se tale valore è minore del termine di confronto si setta il carry; se i valori sono uguali il carry è resettato e lo zero è settato; se infine l'accumulatore è maggiore dell'operando sia il carry che lo zero sono tutti resettati.

### Altre operazioni sull'accumulatore ed i flag

Abbiamo altre istruzioni che agiscono

COMPLEMENTO ACCUM. (CPL)					
Accumul.	Risult.	Flag			
		C	Z	S	N
50	AF	/	/	/	1

COMPLEMENTO A 2 ACCUM. (NEG)					
Accumul.	Risult.	Flag			
		C	Z	S	N
FF	01	0	0	0	1
00	00	1	1	0	1
23	DD	0	0	1	1

COMPLEMENTO DEL CARRY (CCF)					
Carry	Flag				
	C	Z	S	N	
0	1	/	/	0	
1	0	/	/	0	

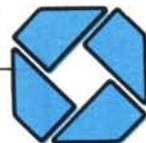
SET CARRY (SCF)					
Flag					
C	Z	S	N		
1	/	/	0		

**Tabella 6**

nel contenuto dell'accumulatore: la CPL ("complement") effettua il complemento di A e la NEG ("negation") invece effettua il "complemento a 2" di A cioè, come visto le scorse puntate, prima fa il complemento e poi incrementa di uno il risultato.

Infine due istruzioni permettono di manipolare lo stato del flag di Carry: l'istruzione CCF complementa lo stato del carry, mentre la SCF lo setta ad 1. Un'occhiata alla tabella 6 ci consente dunque di vedere il comportamento dei flag in questi ultimi casi.

A risentirci la prossima puntata con altre istruzioni molto importanti. 



## DISITACO s.r.l.

DIVISIONE INFORMATICA  
Via Poggio Moiano, 34/C  
00199 Roma (Italia)  
Tel. 06/8310756-8391557

### VENDITA PER CORRISPONDENZA

#### SINCLAIR: linea QL

QL a prezzo di lancio ..... telefonate  
Disk drives 200 - 400 - 800 KB ..... telefonate  
Espansione memoria 128 - 256 .....  
512 K RAM ..... telefonate  
CP/M 68K ..... telefonate  
Monitor QL 14" 85 col ..... telefonate  
Stampante Brother 4 nera 80 col ..... telefonate  
Interfaccia parallela ..... telefonate  
Software (Business and utilities) ..... telefonate

#### SINCLAIR: linea Spectrum

Spectrum 48K PLUS ..... L. 424.000  
Spectrum 48K ..... L. 320.000  
Microdrive ..... L. 118.000  
Interfaccia 1 ..... L. 118.000  
Interfaccia 2 ..... L. 48.000  
Tastiera DK Tronics ..... L. 119.000  
Disk drives 175-350-700 K RAM ..... telefonate  
... e molti altri prodotti ..... telefonate

#### COMMODORE: Hard e Software

CBM 64 OFFERTA SPECIALE ..... telefonate  
CBM PLUS 4 ..... L. 630.000  
CBM C16 ..... L. 220.000  
Drive 1541 ..... L. 440.000  
Stampante MP 802 ..... L. 500.000

#### SHARP SERIE MZ 700:

Hard e software  
MZ 700 + Registratore ..... L. 549.000  
(Mod. 721) .....  
MZ 700 + Registr. + Stampante ..... L. 749.000  
(Mod. 731) .....  
MZ 700 interfaccia Centronics .... L. 115.000

#### EPSON:

Stampante RX-80 in offerta ..... telefonate  
Stampante RX-80 F/T ..... L. 695.000  
Stampante RX-100 ..... L. 975.000  
Stampante FX-80 ..... L. 1.025.000  
Stampante FX-100 ..... L. 1.250.000

#### Olivetti M24 .... L. 3.999.000

#### APPLE - APRICOT - OLIVETTI

a prezzi interessanti  
(chiedere quotazioni)  
Su tutti i prezzi è esclusa l'IVA del 18%.

#### CONDIZIONI DI VENDITA

Il pagamento dovrà essere effettuato in forma anticipata, a mezzo vaglia telegrafico o assegno circolare.  
Le spese sono a carico del destinatario. La spedizione è prevista entro 15 gg.  
Le riparazioni e le sostituzioni del materiale in garanzia sono previste entro 10 gg.

#### VENDITA ALL'INGROSSO

Punto Vendita:  
Via Massaciuccoli 25/A - Tel. 8390100