

L'ASSEMBLER dello



di Pierluigi Panunzi
Quinta parte

La scorsa puntata abbiamo analizzato le possibilità dello Z80 di effettuare operazioni aritmetiche e logiche tra due operandi dei quali in genere uno è l'accumulatore e l'altro è un registro o il contenuto di una locazione di memoria, indicata esplicitamente oppure indirettamente tramite i registri indice IX ed IY o la coppia HL.

Questa volta ci occuperemo delle istruzioni logiche di rotazione e di shift (in breve, istruzioni di "skew") e delle istruzioni di manipolazione dei singoli bit.

In tutto quanto segue ricordiamo, soprattutto ai meno esperti, che parleremo di byte formati da 8 bit; tali bit saranno indicati in modo tale da identificarne immediatamente la posizione nell'ambito del byte: il bit meno significativo (quello più a destra, tanto per intenderci) sarà indifferentemente indicato con "LSB" (Least Significant Bit) o con "bit 0", il successivo (proseguendo verso sinistra) sarà chiamato "bit 1" e così via fino all'ottavo bit posto più a sinistra e chiamato indifferentemente con "MSB" (Most Significant Bit) o "bit 7".

Per evitare ulteriori confusioni ricordiamo che nel contare i bit si parte da 0 per arrivare a 7 ... intesi !?

Le istruzioni di rotazione

Le istruzioni in esame, dette genericamente in gergo le "rotate" (è inglese!), consentono di effettuare un'operazione a prima vista insolita e apparentemente inutile in quanto, abituati come siamo alla notazione decimale, non vi troviamo un effettivo riscontro.

Esse si dividono in "rotazione verso destra" ed in "rotazioni verso sinistra", en-

trambe a loro volta suddivise a seconda se "attraverso il bit di carry" oppure no.

Ma andiamo con ordine iniziando dalle quattro istruzioni che si possono ottenere con le condizioni di cui sopra e relative all'accumulatore.

Analizziamo perciò l'istruzione RRA che sta per "Rotate Right Accumulator", facendo riferimento alla figura 1a.

In particolare, come indicato schematicamente nella figura grazie all'uso delle frecce, i singoli bit dell'accumulatore vengono traslati di una posizione verso destra: ciò potrebbe comportare la perdita del bit meno significativo.

Invece tale bit viene "salvato" nel Carry: inoltre, in questo caso, il posto lasciato vuoto dal bit 7 nel suo spostamento verso destra viene occupato dal valore "vecchio" del Carry, che a sua volta viene così "salvato" dal momento che è stato rimpiazzato precedentemente.

Attenzione: tutte queste "operazioni" di spostamento avvengono "istantaneamente" e cioè senza possibilità di errori.

L'analogia RLA ("Rotate Left Accumulator"), come si può vedere nella figura 1b, effettua una traslazione dei bit verso sinistra, portando il bit 7 nel Carry, ed il "vecchio" Carry al posto del "vecchio" bit 0, rimasto vuoto.

Abbiamo dunque visto le due "rotate" che lavorano "attraverso il Carry".

Viceversa esistono le analoghe che coinvolgono differenzialmente il Carry: l'istruzione RRCA ("Rotate Right Circular Accumulator"), come si vede in figura 1c, effettua una traslazione circolare dei bit dell'accumulatore verso destra ed in particolare il bit 0 viene reinserito al posto vacante lasciato dal "vecchio" bit 7 ed inoltre viene memorizzato nel Carry.

Ecco la differenza fondamentale: in questo caso il contenuto del Carry viene perso.

Analoghe considerazioni si possono effettuare per l'istruzione RLCA ("Rotate Left Circular Accumulator"), per la quale rimandiamo alla figura 1d.

Abbiamo dunque visto quattro tipi fondamentali di "rotate": ne abbiamo parlato all'inizio in quanto queste istruzioni erano quelle già presenti (e le uniche del genere!) nel set di istruzioni dell'8080, che è stato notevolmente ampliato dai progettisti della Zilog in fase di progettazione dello Z80.

In particolare sono state aggiunte tutte le istruzioni di cui parleremo nel resto dell'articolo.

Ecco introdotte le generalizzazioni delle quattro istruzioni viste finora, che si chiamano "RRr", "RLr", "RRCr" ed "RLCr".

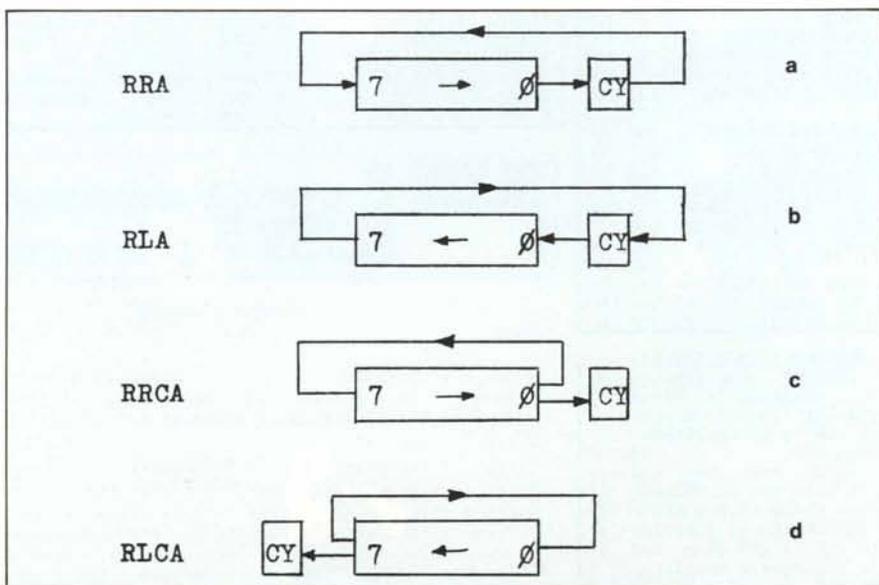


Figura 1 - Le istruzioni di "Rotate": il registro indicato è l'accumulatore e CY è il Carry.

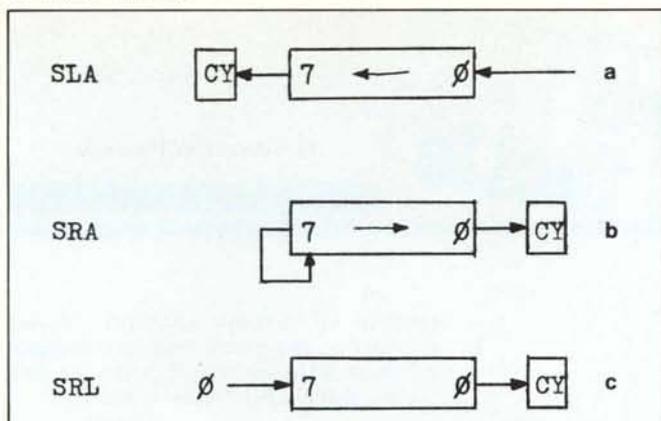


Figura 2 - Le istruzioni di "Shift": il registro indicato è "generalizzato" e CY è il Carry. In particolare nel caso (b) il bit 7, oltre che shiftato verso destra, viene lasciato anche al suo posto.

cioè rispettivamente "Rotate Right", "Rotate Left", "Rotate Right Circular" e "Rotate Left Circular" e dove la "r" sottintende un registro qualsiasi (B, C, D, E, H, L, nonché lo stesso A) oppure una locazione di memoria indirizzata da HL nonché indirettamente e con il meccanismo del "displacement" tramite i registri indice IX ed IY. In tutti questi casi il meccanismo è assolutamente identico alle corrispondenti istruzioni analizzate in precedenza.

Sono state aggiunte inoltre altre tre istruzioni di "shift", relative ad un operando "generalizzato" (come quello precedente, tanto per non ripeterci).

In particolare esiste la funzione "SLA r" ("Shift Left Arithmetical") la quale, come si può vedere dalla figura 2a, effettua una traslazione dei bit verso sinistra "aritmeticamente" e cioè riempiendo il posto precedentemente occupato dal bit 0 con uno 0 e "salvando" il "vecchio" bit 7 nel Carry, che così viene distrutto.

Vi è poi la "SRA r" ("Shift Right Arithmetical") che effettua, come si vede in figura 2b, una traslazione verso destra, salvando il "vecchio" bit 0 nel Carry (che si distrugge) e viceversa *mantenendo* il vecchio valore del bit 7 e cioè lasciando inalterato il "segno" del byte: ciò è molto utile per dividere ad esempio per due un numero che può essere anche negativo.

Se questo comportamento non è desiderato, allora si può usare la "SRL r" ("Shift Right Logical") che è differenza della precedente, come si può vedere in figura 2c, imposta un valore 0 nel bit più significativo dell'operando.

Facendo un po' di conti abbiamo, per quanto riguarda le istruzioni "nuove" dello Z80, quattro "Rotate" più tre "Shift", ognuna relativa ad un operando a scelta tra 10: in totale abbiamo la bellezza di 70 istruzioni in più rispetto al vecchio 8080, e non finiscono certo qui...

Due "Rotate" un po' particolari

Facendo riferimento alla figura 3 vediamo due altre istruzioni di "rotate", chiamate RLD e RRD (rispettivamente "Ro-

tate Left Digit" e "Rotate Right Digit"), veramente inconsuete.

In particolare operano tra l'accumulatore e la locazione di memoria puntata dalla coppia HL (chissà perché questa scelta, unica nel suo genere, da parte dei progettisti) ed effettuano una traslazione rispettivamente verso sinistra e verso destra del "nibble meno significativo" dell'accumulatore e dei due "nibble" della locazione di memoria già individuata.

Detto che ancora una volta tutte le operazioni di traslazione avvengono contemporaneamente, queste due strane istruzioni servono più che altro quando si ha a che fare con numeri espressi in BCD, ma non ci soffermeremo oltre sull'argomento.

Istruzioni di manipolazione dei singoli bit

Sono queste delle istruzioni veramente potenti ed interessanti, che ogni programmatore desidererebbe vedere nel set di istruzioni del proprio microprocessore preferito, ma che invece anche nei "micro" a 16 bit è raro trovare.

In parole povere consentono di testare lo stato di uno degli 8 bit di un operando, come pure di settare o resettare uno degli 8 bit.

In dettaglio abbiamo le "istruzioni BIT b, r" dove "b" può assumere a piacere un valore tra 0 e 7 ed "r" è il solito "operando generalizzato" che ormai conosciamo molto bene.

Abbiamo parlato di "istruzioni BIT" in quanto, facendo un po' di conti, in questo caso abbiamo come al solito 8 scelte (il numero di bit) moltiplicate per i 10 operandi: un totale di 80 istruzioni BIT

Per vedere il funzionamento di questa funzione bisogna ragionare come segue: se il bit in esame è zero (resettato) allora il "flag di zero" è posto "correttamente" ad 1, come nei casi in cui, ad esempio, l'accumulatore è nullo.

In termini tecnici, indubbiamente molto oscuri e che sottintendono quasi un sottile gioco di parole, "il negato del bit in esame è posto nel flag di Zero": il gioco di parole

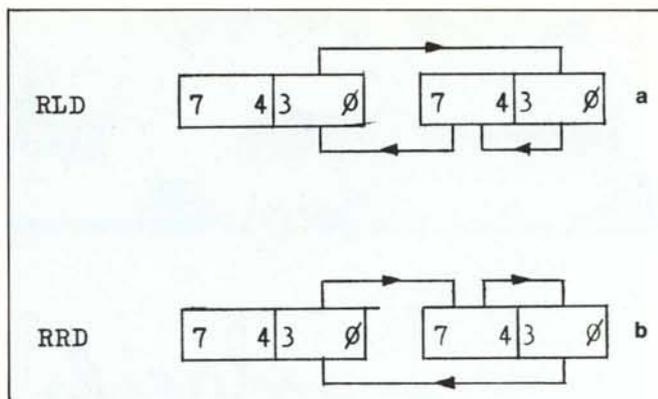


Figura 3 - Le due istruzioni di "Digit Rotate": il registro a sinistra è l'accumulatore e quello a destra è la locazione (HL), cioè quella puntata dalla coppia di registri HL. In entrambi i registri si considerano il "nibble più significativo", formato dai bit 7=4 ed il "nibble meno significativo", formato dai bit 3=0.

può nascere dal fatto che "se il bit è 0, il flag di Zero è 1" e viceversa "se il bit è 1, il flag di Zero è 0". Sembra buffo, ma basta ragionarci un attimo per convincersi che in realtà non potrebbe essere altrimenti, in quanto l'operazione è strettamente logica...

Altre ottanta istruzioni provengono dalle "SET b, r" ed altre ottanta ancora dalle "RES b, r" che rispettivamente pongono ad 1 e a 0 il bit "b" (specificato nell'istruzione) dell'"operando generalizzato" "r".

E gli altri flag?

Abbiamo sinora visto il comportamento del flag di Carry nelle "Rotate" e dello "Zero" nelle BIT: per completezza indicheremo ora come si comportano gli altri bit di flag, relativamente ai più importanti.

Le prime quattro "Rotate", originarie dell'8080, ovviamente influenzano il Carry, mentre non alterano minimamente lo "Zero" ed il "Sign", viceversa pongono a 0 il flag "N".

Invece le "rotate generalizzate" e le "shift generalizzate" influenzano il "Carry" (ovviamente!), lo "Zero" ed il "Sign" e pongono al solito il flag "N" a zero.

Per quanto riguarda le "digit rotate", queste si comportano come le precedenti ad eccezione del flag di Carry che non viene minimamente sfiorato.

Dopo aver detto che le SET e RES non alterano alcun flag di sorta, non rimangono che le BIT: come detto alterano il flag "Zero", mentre non toccano il Carry, azzerano il flag "N", mentre infine lasciano sconosciuto e perciò casuale lo stato del flag di segno.

Con questo abbiamo terminato la rassegna di queste istruzioni "bit oriented", delle quali lo Z80 può ben vantarsi nel paragone con gli altri microprocessori ad 8 bit, quali il 6502 o il 6800, tanto per citare al solito i più noti.

La prossima volta conosceremo altre "perle" del ricchissimo set di istruzioni dello Z80: i salti e le chiamate di subroutine, anche queste nettamente superiori come numero e qualità rispetto alla concorrenza.