

Moltiplicazione 8 per 8 bit per numeri con segno

Emanuele Santoro <santoro@autistici.org> (Classe 4AI 2009/2010)

2 Ottobre 2009

Contents

1	Strumenti utilizzati	1
1.1	Z-80 Simulation	1
2	Descrizione dell'algoritmo "Moltiplicazione 8x8 per numeri con segno"	1
3	Moltiplicazione: descrizione dell'algoritmo shift-somma	2
4	Listato "Shift-Somma"	2
4.1	Registri utilizzati:	2
4.2	Codice sorgente dello shift-somma	2
4.3	Codice macchina dello shift-somma	3
4.4	Diagramma di flusso dello shift-somma	4
4.5	Codice sorgente del programma principale	5
4.6	Codice macchina del programma principale	6
4.7	Diagramma di flusso del programma principale	7
5	Test eseguiti	7
6	Conclusioni	8

1 Strumenti utilizzati

1.1 Z-80 Simulation

Il programma "Z80 Simulation" di Fulvio Ricciardi è un programma che simula il funzionamento del microprocessore Z80 permettendoci di testare

tutti i programmi senza operare fisicamente su uno z80.

2 Descrizione dell'algoritmo "Moltiplicazione 8x8 per numeri con segno"

Questo programma è una rivisitazione del già studiato programma di moltiplicazione di due numeri ad otto bit, con la differenza che questo può operare su numeri con segno.

I numeri negativi sono rappresentati in complemento a due, mentre i numeri positivi sono rappresentati normalmente.

Il principio di funzionamento del programma è questo: viene controllata la positività dei numeri che verranno trattati: se sono negativi li complementa e registra l'esecuzione di tale operazione, di modo che sia possibile decidere se è necessario complementare anche il risultato.

Infatti il risultato dovrà essere complementato solo se uno dei due fattori è negativo.

3 Moltiplicazione: descrizione dell'algoritmo shift-somma

Una volta resi positivi gli eventuali numeri negativi, essi verranno moltiplicati tramite il classico algoritmo di "shift-somma", ovvero il risultato verrà shiftato a sinistra e poi gli verrà sommato il moltiplicando, tante volte quanti sono il numero dei bit che rappresentano i dati.

4 Listato "Shift-Somma"

4.1 Registri utilizzati:

- B -> Contatore
- HL -> Risultato
- H -> Moltiplicatore
- E -> Moltiplicando

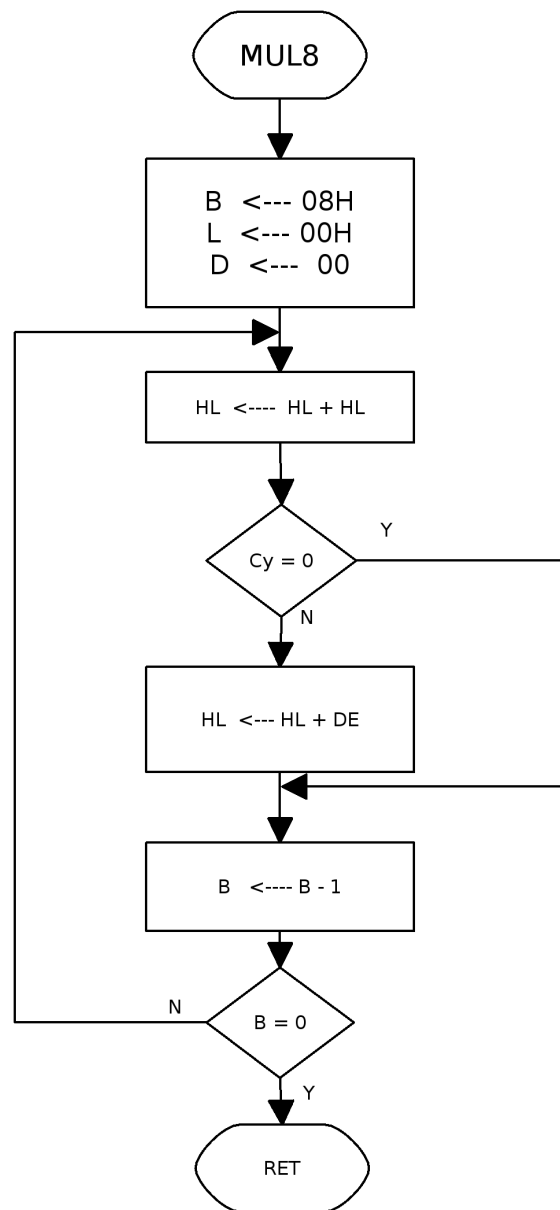
4.2 Codice sorgente dello shift-somma

```
1:  ORG #1800
2:          LD B, #08
3:          LD L, 00
4:          LD D, L
5:  LOOP    ADD HL, HL
6:          JR NC, LOOP1
7:          ADD HL, DE
8:  LOOP1   DJNZ LOOP
9:          END
```

4.3 Codice macchina dello shift-somma

Indirizzo	Codice
1800	06 08
1802	2E 00
1804	16 00
1806	29
1807	30 01
1809	19
180A	10 FA

4.4 Diagramma di flusso dello shift-somma



4.5 Codice sorgente del programma principale

```
1:
2:  ORG #1900
3:      LD C,00
4:      LD A,E
5:      SLA A
6:      JR NC, LOOP2
7:      INC C
8:      LD A, E
9:      NEG
10:     LD E,A
11: LOOP2 LD A,H
12:     SLA A
13:     JR NC,LOOP3
14:     INC C
15:     LD A,H
16:     NEG
17:     LD H,A
18: LOOP3 CALL MUL8
19:     RR C
20:     JR NC, LOOP4
21:     LD B,H
22:     LD C,L
23:     XOR A
24:     LD HL,0000
25:     SBC HL, BC
26: LOOP4 HALT
```

4.6 Codice macchina del programma principale

Indirizzo	Codice
1800	0E 00
1802	7B
1803	CB 27
1830	30 05
1807	0C
1808	7B
1809	ED 44
180B	5F
180C	7C
180D	CB 27
180F	30 05
1811	0C
1812	7C
1813	ED 44
1815	67
1816	CD 00 19
1819	CB 19
181B	30 08
181D	44
181E	4D
181F	AF
1820	21 00 00
1823	ED 42
1825	76

- E: 0x15h

- H: 0x02h

Risultato: 0x32h (CORRETTO)

- E: 0xE7h

- H: 0x02h

Risultato: 0xCEh (CORRETTO)

- E: 0x02h

- H: 0xE7h

Risultato: 0xCEh (CORRETTO)

- E: 0xFEh

- H: 0xE7h

Risultato: 0x32h (CORRETTO)

6 Conclusioni

I test confermano che il codice sopra riportato funziona con diverse combinazioni di numeri positivi e negativi.