

RAPPRESENTAZIONE E CONVERSIONE IN ASSEMBLER

Il problema che dobbiamo risolvere in linguaggio assembler è il seguente: dato un numero codificato in una stringa (ASCII) come faccio per convertirlo in un numero in rappresentazione decimale ? (stringa di bit) .Oppure dato un numero codificato in bit, come faccio per convertirlo in una stringa ? (ASCII)

Il problema non è difficile da risolvere, basta ricordarsi dei metodi che si utilizzano per convertire i numeri dalle e nelle vari basi.

1° CASO Conversione da ASCII in BINARIO

mettiamo il caso di avere in un vettore (buffer) la cifra “1674” che ovviamente è codificata in ASCII, essendo una stringa:

'1' = 31h
'6' = 36h
'7' = 37h
'4' = 34h

Ecco come sono posti in memoria i dati :

31h
36h
37h
34h

Per fare la conversione dobbiamo prendere dal buffer un dato alla volta, convertirlo nel suo corrispondente numero in binario, moltiplicarlo per la rispettiva base e sommarlo con i numeri precedentemente convertiti. In questo caso il nostro numero andrebbe trattato così :

$$N = 10^3 * 1 + 10^2 * 6 + 10^1 * 7 + 10^0 * 4$$

vediamo ora una semplice MACRO per convertire il numero :

CONVERSIONE MACRO buffer

```
mov ch, lunghezza_buffer ; metto in CH la lunghezza del buffer (contatore)
mov si,offset buffer      ; SI mi punta alla prima locazione del buffer
mov cl,0
xor ax,ax                  ; azzero il contenuto di ax
```

LOOP:

```
mov ax,[si]           ; metto in AX il primo dato del buffer
and ax,0Fh           ; converto il dato da ASCII in BINARIO
mov dx,pesi[cl]      ; metto in dx la base per la quale devo moltiplicare il numero
mul dx               ; AX * DX e risultato in AX
add ax,N             ; sommo il numero con i precedenti già trovati
mov N,ax             ; metto la somma in N
inc cl
dec ch
cmp ch,0             ; controllo se il buffer sia finito o no
JNZ loop             ; se non è finito ricomincio da loop se no termino

ENDM
```

Evidentemente nel buffer chiamato pesi ho i seguenti dati precaricati

1000
100
10
1

Ovviamente in Binario e queste non sono altro le basi per le quali andrò a moltiplicare volta volta i miei numeri codificati da ASCII.

Molto più semplice se devo convertire un numero con una sola cifra decimale (0-9)..L'operazione da fare si riduce in

```
and ax,0Fh
```

NOTA BENE:

Vi sono altri modi alternativi per la conversione da ASCII in BINARIO :

```
sub ax,'0' oppure sub ax,30H
```

2° CASO Conversione da BINARIO in ASCII

In questo caso abbiamo il problema opposto al precedente. Il modo di procedere è esattamente l'inverso. Devo successivamente dividere il mio numero per la base 10 in modo da trovare le singole cifre decimali da convertire in ASCII

1674:10 -> 167 e resto 4 (converto il 4 in ASCII -> 34h e lo inserisco nel buffer p.s ultima posizione)

167:10 -> 16 e resto 7 (converto il 7 -> 37h e lo inserisco nella penultima posizione)

e così via fino all'ultima cifra, che dovrà essere 1

in questo modo alla fine della conversione avrò un buffer con inseriti i seguenti dati

31h
36h
37h
34h

Non è altro che il mio numero codificato in ASCII ,ovvero in stringa "1674"

Supponiamo di avere nel registro AX in numero 1674 (in binario ovviamente).

AX
0000101110001010

Vediamo adesso una Macro per convertire il numero in ASCII:

CONVERSIONE MACRO buffer

```
mov si,offset buffer
add si,4                ; faccio puntare SI all'ultima locazione di buffer
mov cl,10              ;carico in CL la base per la quale devo dividere
mov ch, lunghezza_buffer ; inizializzo il contatore con la lunghezza del buffer (4)
```

LOOP :

```
div ax,cl              ; AX/CL  quoziente in AL e resto in AH
or ah,30H
mov[si],ah
```

```

dec si
mov N,al           ;N variabile di appoggio che utilizzo per rimettere il quoziente
mov ax,N          ;in AX per la divisione successiva
dec ch
cmp ch,0
JNZ loop

```

ENDM

Il passaggio nel quale aggiungo 4 al registro SI per farlo puntare all'ultima locazione di buffer è dovuto al fatto che nei vari passaggi di conversione (le varie divisioni per la base 10) trovo sempre l'ultimo elemento del numero corrente e quindi devo inserire dal fondo e non dalla cima.

Molto più semplice se devo convertire un numero con una sola cifra decimale (0-9)..L'operazione da fare si riduce in

```
or ax,30h
```

NOTA BENE:

Vi sono anche altri modi per convertire un binario in ASCII:

```
add ax,30h oppure add ax,'0'
```