Istruzioni aritmetiche

Nelle istruzioni a due operandi il risultato è sempre memorizzato nel primo operando.

add op1, op2

Descrizione:

- somma il valore di op1 a quello di op2 ed il risultato è memorizzato al posto di op1.
- il primo operando può essere un registro o una locazione di memoria
- il secondo può essere o un registro o una locazione di memoria o un valore immediato
- i due operandi vengono espressi in complemento a 2.
- i due operandi devono avere la stessa dimensione e **non possono essere entrambi locazioni di memoria**

In base al risultato vengono impostati i flag:

z (zero) a 1 se il risultato è 0
s (segno) a 1 se il risultato è negativo
c (carry) a 1 se c'è stato il riporto sull'ultimo bit
a (auxiliary carry) a 1 se c'è stato il riporto tra semibyte (nei bit intermedi)
p (parità) a 1 quando il numero di bit posti a 1 nel risultato rappresentato in binario è pari (es. se il risultato è 11001001, p=1 perchè i bit uguali a 1 sono 4)
o (overflow) quando il risultato esce dal range

NON vengono modificati gli altri tre flag di controllo (d, i, t)

ESERCIZIO 1. Scrivere un programma in Assembly che esegua la somma tra due variabili num1=3 e num=2.
Osservazioni.

- Posso utilizzare l'istruzione **ADD num1,num2** ? Motivare la risposta.
- Esegui il programma con TD, i flag sono stati modificati?

ESERCIZIO 2. Scrivere un programma in Assembly che esegua la somma tra una variabile e un numero inserito in un registro, e che memorizzi il risultato nella variabile **ris.**
Osservazioni.
Modifica i valori degli operandi, scegli le seguenti coppie, esegui il programma con TD e osserva quali flag sono stati modificati:

- 3 e — 4 (nota eseguendo il programma con TD che i numeri negativi sono memorizzati in complemento a 2) [MODIFICATI: s=1, p=1]
- 3 e — 8 [MODIFICATI: s=1, p=0]
- 2 e —15 [MODIFICATI: s=1, p=1]
- —3 e —15 [MODIFICATI: s=1, p=1,c=1]
- 20 e 30 [MODIFICATI: a=1]
- 4 e —4 [MODIFICATI: c=1, z=1, a=1, p=1]
- 127 e 1 [MODIFICATI: o=1,s=1, a=1]
- 128 e 1 [MODIFICATI: o=0 (perché?), s=1, p=1]
- 128 e —1 [MODIFICATI: o=1(perché?), c=1]
- 128 e 128 [MODIFICATI: o=1, c=1, p=1,z=1]

Sub op1, op2

Descrizione: esegue la sottrazione, ha la stessa sintassi e le stesse regole di add

inc op

descrizione:

- per sommare 1 invece dell'istruzione add
- l'operando più essere un registro o una locazione di memoria

esempio: inc bl

dec op

descrizione:

- per sottrarre di 1 unità
- l'operando più essere un registro o una locazione di memoria