

**Corso di CALCOLATORI ELETTRONICI I
CdL INGEGNERIA INFORMATICA**

A.A: 2000-2001

Prof. Bruno FADINI

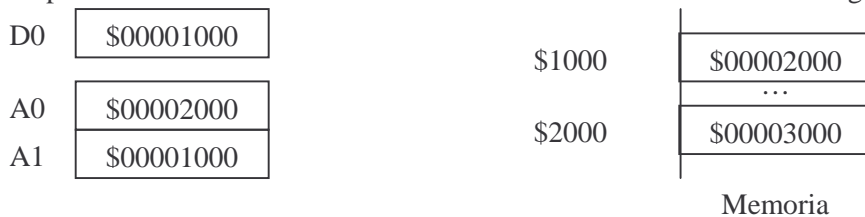
Prova intracorso numero 3

1. Determinare il contenuto del registro D0 e dei flag N (negative), Z (zero), V (overflow) e C (carry) al termine di ciascuno dei seguenti segmenti di codice:

- a) A EQU \$62A0
 MOVE.W #\$1234, D0
 ADD.W #A, D0
- b) A EQU \$62A0
 MOVE.W #\$4321, D0
 ADD.W #A, D0
- c) A EQU \$62A0
 MOVE.W #\$9D60, D0
 ADD.W #A, D0
- d) A EQU \$62A0
 MOVE.W #\$CFA0, D0
 ADD.W #A, D0

2. Scrivere un programma assembler che valuti l'espressione: $A*B + C*D$ in un processore ad accumulatore, che dispone delle istruzioni di Load, Store, Multiply e Add. Si supponga che tutti i valori siano compatibili con le dimensioni dell'accumulatore.

3. Sia dato un processore Motorola 68000 che si trova nello stato illustrato dalla figura seguente.



- a) Qual è l'effetto dell'esecuzione di ciascuna delle seguenti istruzioni, a partire da questo stato iniziale?
- a.1) ADD.L D0, (A0)
a.2) ADD.L (A1, D0), D0
a.3) ADD.L #\$2000, (A0)
- b) Quanti accessi alla memoria richiedono le fasi di costruzione degli operandi e di esecuzione di ciascuna istruzione?
4. Disponendo di addizionatori in modulo 16 e di porte logiche AND, OR e NOT, si progetti un addizzatore in modulo 12.

Posto:

$$A = a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

$$B = b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

$$S = s_3 \cdot 2^3 + s_2 \cdot 2^2 + s_1 \cdot 2^1 + s_0 \cdot 2^0$$

con: $0 \leq A < 12, \quad 0 \leq B < 12$

deve essere:

$$\begin{cases} S = (A + B + c_{in}) \bmod 12 \\ c_{out} = (A + B + c_{in}) \geq 12 \end{cases}$$

