

**Corso di Calcolatori Elettronici I
A.A. 2010-2011**

**Esempi di programmi
assembly per MC68000
(seconda serie)**

Lezione 23

Prof. Roberto Canonico



Università degli Studi di Napoli Federico II
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica (allievi A-DE)
Corso di Laurea in Ingegneria dell'Automazione

Esercizio: somma n interi

- Scrivere un programma che sommi i primi n interi
 - Assemblare il programma con ASIMTOOL ed eseguirlo sul simulatore ASIM
 - Sperimentare:
 - L'effetto dell'istruzione CLR in memoria
 - L'effetto dell'istruzione MOVE da memoria a registro
 - L'effetto dell'istruzione ADD tra memoria e registro
 - L'effetto delle varie istruzioni sui codici di condizione
 - L'effetto dell'istruzione BNE sul PC
 - L'effetto dell'istruzione JMP sul PC
-

Soluzione - sumnnums.a68

```

START      ORG      $8000
           CLR.W    D0
           MOVE.W   N, D1
LOOP       ADD.W    D1, D0
           ADD.W    #-1, D1
           BNE     LOOP
           MOVE.W   D0, SUM
           STOP    #2700
           ORG     $8200
N          DC.W    17
SUM        DS.W    1
           END     START

```

Esercizio: prodotto scalare

- Scrivere un programma che esegua il prodotto scalare tra due vettori di interi word A e B
 - A e B allocati staticamente ed inizializzati con DC
 - Assemblare il programma con ASIMTOOL ed eseguirlo sul simulatore ASIM
-

DBcc: Test condition, decrement, and branch

Operazione: IF (*cc* false) THEN
 [Dn] ← [Dn] - 1
 IF [Dn] = -1 THEN [PC] ← [PC] + 2
 ELSE [PC] ← [PC] + *d*
 ELSE [PC] ← [PC] + 2

Sintassi: DBcc Dn,<label>
Attributi: Size = word

Descrizione:

Fintantoché la condizione *cc* rimane falsa, decrementa il registro *Dn*, e se questo non era zero prima del decremento (ovvero se non vale -1) salta all'istruzione a distanza *d*. Negli altri casi, passa all'istruzione seguente.

Fornisce un modo sintetico per gestire i cicli, sostituendo con un'unica istruzione il decremento di un registro di conteggio e la verifica di una condizione normalmente fatti con istruzioni separate.

Supporta tutti i *cc* usati in Bcc. Inoltre, ammette anche le forme DBF e DBT (F = false, e T = true) per ignorare la condizione ed usare solo il registro di conteggio.

Soluzione – scalprod.a68

```

ORG      $8000
START  MOVE.L  #A,A0
        MOVE.L  #B,A1
        MOVE.L  #N-1,D0
        CLR     D2
LOOP   MOVE   (A0)+,D1
        MULS  (A1)+,D1
        ADD   D1,D2
        DBRA  D0,LOOP
        MOVE  D2,C
DONE   STOP   #$2700
N      EQU    $000A
A      DC.W   1,1,1,1,1,1,1,1,1,1
        ORG   $80D0
B      DC.W   1,1,1,1,1,1,1,1,1,1
C      DS.W   1
        END   START

```
