

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

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

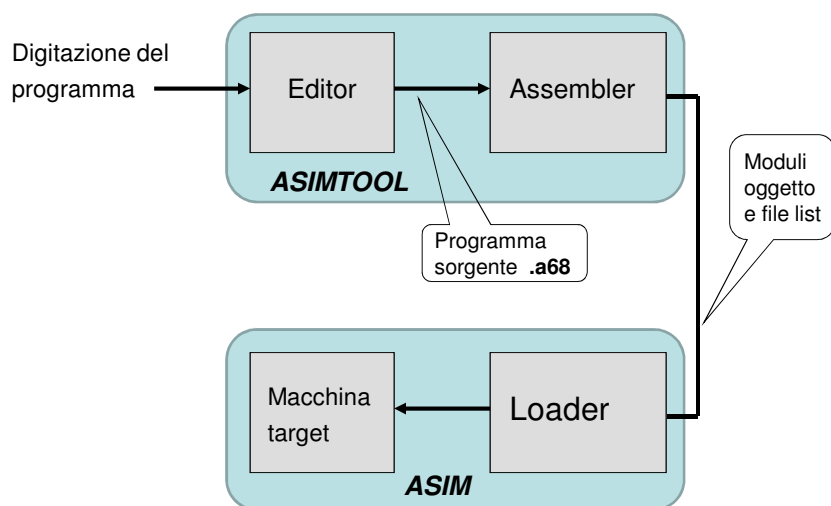
Lezione 22

Prof. Roberto Canonico



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

**Ciclo di sviluppo semplificato di programmi
assembly MC68000 nel sistema didattico ASIM**



Esercizio - Moltiplicazione 2 interi

```

* Programma per moltiplicare MCND e MPY
*
      ORG      $8000
*
MULT  CLR.W    D0          D0 accumula il risultato
      MOVE.W  MPY,D1      D1 e' il contatore di ciclo
      BEQ     DONE       Se il contatore e' zero e' finito
LOOP  ADD.W   MCND,D0     Aggiunge MCND al prodotto parziale
      ADD.W   #-1,D1      Decrementa il contatore
      BNE    LOOP        e ripete il giro
DONE  MOVE.W  D0,PROD     Salva il risultato
      STOP   #$2700      Arresta esecuzione programma
      ORG   $8200
PROD  DS.W    1          Riserva spazio di memoria per PROD
MPY   DC.W   3          Definisce il valore di MPY
MCND  DC.W   4          Definisce il valore di MCND
END   MULT          Fine ass., salto a entry point

```

AsimTool

The screenshot shows the AsimTool interface with the following content:

```

ASIM TOOL - OS.LIS
File Edit View Assemble Option Window Help
DS.A68
* os.a68: Simulates Operating System
*
      ORG      $9000
*
LOOP  00000000 1 * os.a68: Simulates Operating System
      00000000 2 *
      00009000 3      ORG   $9000
      00009000 4
      00009000 5 LOOP  NOP
      00009002 4EF9 00009000 6      JMP  LOOP

No errors detected
No warnings generated

```

The status bar at the bottom indicates "Ready".

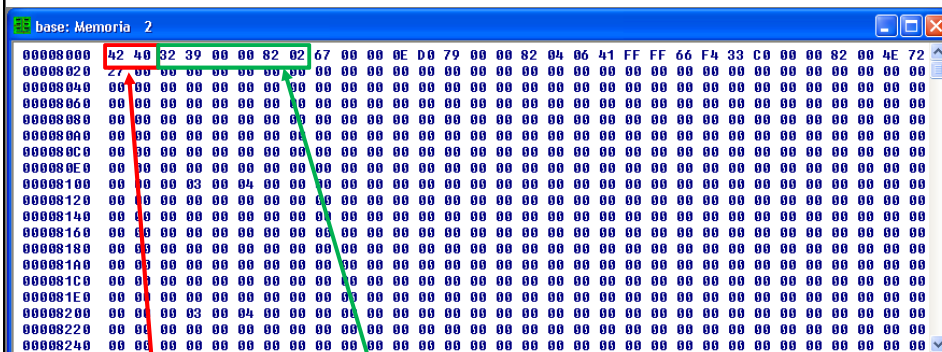
Analisi del file LIS

PLC	contenuto	label opcode operandi
00000000		1 * Programma per moltiplicare MCND e MPY
00000000		2 *
00008000		3 ORG \$8000
00008000		4 *
00008000	4240	5 MULT CLR.W D0
00008002	3239 00008202	6 MOVE.W MPY,D1
00008008	6700 000E	7 BEQ DONE
0000800C	D079 00008204	8 LOOP ADD.W MCND,D0
00008012	0641 FFFF	9 ADD.W #-1,D1
00008016	66F4	10 BNE LOOP
00008018	33C0 00008200	11 DONE MOVE.W D0,PROD
0000801E	4E72 2700	12 STOP #\$2700
00008200		13 ORG \$8200
00008200		14 PROD DS.W 1
00008202	0003	15 MPY DC.W 3
00008204	0004	16 MCND DC.W 4
00008206		17 END MULT

Symbol Table

MULT	\$8000	MPY	\$8202	DONE	\$8018
LOOP	\$800C	MCND	\$8204	PROD	\$8200

Asim: programma caricato in memoria



CLR.W D0

MOVE.W MPY,D1

Esercitazione

- Provare il programma mult2ints.a68 che moltiplica due interi attraverso un ciclo di addizioni ripetute
- Eseguire il programma sul simulatore ASIM e sperimentare:
 - L'effetto di DC e la convenzione big-endian del processore 68000
 - L'effetto dell'istruzione CLR .W su registro di 32 bit
 - L'effetto dell'istruzione MOVE da memoria a registro
 - L'effetto dell'istruzione BEQ sul PC
 - L'effetto dell'istruzione ADD tra memoria e registro
 - L'effetto dell'istruzione ADD tra immediato e registro
 - L'effetto dell'istruzione BNE sul PC
 - L'effetto dell'istruzione JMP sul PC
 - L'effetto dell'istruzione MOVE da registro a memoria
 - Confrontare la codifica in I/m delle istruzioni di salto

Esercitazione

- Nell'esempio precedente, effettuare le seguenti sostituzioni ed osservarne gli effetti

DONE	MOVE.W	D0,PROD	Salva il risultato
PROD	DS.W	1	Riserva spazio di memoria per
PROD			

↓

DONE	MOVE.L	D0,PROD	Salva il risultato
PROD	DS.L	1	Riserva spazio di memoria per
PROD			

L= long (32 bit) W= word (16 bit) B= byte (8 bit)