Corso di Calcolatori Elettronici I

Motorola 68000: primi programmi assembly

ing. Alessandro Cilardo

Corso di Laurea in Ingegneria Biomedica

Organizzazione dei dati in memoria

File: `programma006.a68`

- Un *vettore* è una sequenza ordinata di dati dello stesso tipo presenti in memoria
- Scrivere un programma che allochi in memoria un vettore contenente le 5 word (due byte) elencate di seguito:

  2A11, **2FB1**, **CC92**, **3E40**, **A119**

  (*suggerimento: usare la direttiva* `DC`)
- Compilare il programma e, osservando il file .LIS, riconoscere il tipo di organizzazione dei dati in memoria adottato dal processore

Organizzazione dei dati in memoria

File: `programma007.a68`

- Estendere il precedente programma aggiungendo una seconda area dati avente la stessa dimensione del primo vettore, ma non inizializzata
  - (*suggerimento: usare la direttiva* `DS`)
- Usando le opportune operazioni di trasferimento (`MOVE`), copiare il vettore iniziale nella seconda area allocata dal programma, in modo che il vettore risultante sia ordinato secondo una convenzione *little-endian*

Operazioni di shift

- Il processore **68000** mette a disposizione le seguenti istruzioni di shift (consultare il manuale per i dettagli della sintassi)
  - **`ASL`**: shift aritmetico a sinistra
  - **`ASR`**: shift aritmetico a destra
  - **`LSL`**: shift logico a sinistra
  - **`LSR`**: shift logico a destra
  - **`ROL`**: ruota a sinistra
  - **`ROR`**: ruota a destra

Operazioni di shift

File: `programma008.a68`

- Scrivere un programma che contenga una variabile di tipo Word (due byte) in cui sia memorizzato il seguente valore: **B155**
- Il programma deve caricare il dato in un registro ed applicargli uno shift a destra.
- Compilare e simulare il programma. Osservare cosa cambia usando uno shift a destra *logico* piuttosto che *aritmetico*
- Ripetere le stesse prove con lo shift a sinistra

Operazioni di shift

File: `programma009.a68`

- Scrivere un programma che contenga una variabile di tipo Long (4 byte) in cui sia memorizzato il seguente valore: **B155**
- Il programma deve moltiplicare il precedente dato per il valore **19** = (**10011**)$_2$, facendo uso soltanto di operazioni di shift e di addizione.

  suggerimento*: ricordare un esempio simile presentato nella parte sulla rappresentazione dei numeri*

Swap

File: `programma010.a68`

- Scrivere un programma che contenga due variabili di tipo Word (due byte) etichettate come **A** e **B**.
- Scegliere arbitrariamente il contenuto delle due locazioni di memoria
- Facendo uso di operazioni di spostamento, il programma deve eseguire lo *swap*, ovvero deve invertire i contenuti delle due locazioni di memoria **A** e **B**.

Costrutti condizionali

File: `programma011.a68`

- Scrivere un programma che contenga due variabili di tipo Word (due byte) etichettate come **A** e **B**.
- Scegliere arbitrariamente il contenuto delle due locazioni di memoria
- Facendo uso di operazioni di spostamento, di confronto (**`CMP`**) e di salto condizionato (**`Bcc`**), il programma deve eseguire lo *swap*, ovvero deve invertire i contenuti delle due locazioni di memoria, ma soltanto nel caso in cui il contenuto di **A** sia **minore** del contenuto di **B**

Costrutti condizionali e cicli

File: `programma012.a68`

- Scrivere un programma che calcola la moltiplicazione tra due variabili **A** e **B** secondo il seguente approccio:

  somma **A** *un numero di volte indicato dal valore di* **B**

  Ad esempio, per calcolare 5 x 3, *calcola* 5 + 5 + 5= 15

suggerimento: usare costrutti di confronto (**`CMP`**) e salto condizionato (**`Bcc`**) per creare un ciclo. Il ciclo deve essere ripetuto **B** volte. Usare un registro dati **Dx**, da inizializzare con il valore di **B** e decrementare fino a quando diventa **0**, mentre in un altro registro dati **Dy** si somma ripetutamente il valore di **A**.