

Esercizi Assembly 5

M. Sonza Reorda – M. Grosso

Politecnico di Torino
Dipartimento di Automatica e Informatica

Esercizio 1

- Scrivere un programma in Assembly 8086 che, dati due operandi *opa* e *opb* di tipo *word* in memoria, del valore rispettivo di 2043 e 5, esegua un'operazione tra interi scelta dall'utente e salvi il risultato nella variabile *word res*
- A seconda del carattere digitato dall'utente, il programma deve eseguire:
 - $1 \rightarrow res = a+b$
 - $2 \rightarrow res = a-b$
 - $3 \rightarrow res = a*b$
 - $4 \rightarrow res = a/b$ (divisione intera).

Implementazione

- Occorre implementare un costrutto *switch*:

```
switch (espressione)
{
  case val1:      sequenza1;
                  break;
  case val2:      sequenza2;
                  break;
  ...
  default:        sequenza_def;
}
```

- Si possono utilizzare:
 - operazioni di *compare* e salti condizionati a blocchi di istruzioni
 - una tabella di *jump* e un'unica istruzione di salto incondizionato.

Esercizio 2

- Scrivere un programma che acquisisca due variabili di tipo *byte* opA e opB in formato binario da tastiera (sequenza di '0' e '1' digitati dall'utente) e scriva il risultato di una operazione logica *bitwise* in una variabile di tipo *byte* ris. L'espressione logica bit-a-bit in questione è
- $C = \text{NOT}(A \text{ AND } (\text{NOT}(B))) \text{ OR } (A \text{ XOR } B).$

$$\begin{array}{c}
 \overline{(A \cdot \bar{B})} + (A \cdot \bar{B} + \bar{A} \cdot B) \\
 \begin{array}{ccc}
 \bar{A} + B & + & A\bar{B} + \bar{A}B \\
 \uparrow & \uparrow & \downarrow \\
 \text{OR} & \text{OR} & = 1
 \end{array}
 \end{array}
 \Rightarrow \boxed{C = 1}$$

Implementazione

- Acquisizione di valori binari
 - Necessaria conversione di caratteri ASCII letti da tastiera
 - L'utente introduce valori binari (es.: "00011100") separati da ENTER
- Operazioni logiche su *byte* effettuate con apposite istruzioni.

Esercizio 3

- Si scriva un programma in linguaggio Assembly 8086 che conti il numero di bit a 1 nella rappresentazione binaria di una variabile di tipo *byte*.

Esercizio 4

- Si scriva un programma in linguaggio Assembly 8086 che esegua un'operazione di AND logico tra gli elementi di due vettori di *byte* e ne memorizzi il risultato in un terzo vettore. Il programma deve inoltre contare gli elementi a parità pari nel vettore risultante, cioè tali per cui il numero di 1 nella rappresentazione binaria del numero è pari.

Esercizio 5

- Si scriva un programma in grado di determinare se ciascuno dei numeri naturali (≥ 2) contenuti in un vettore è primo oppure no. Si ricorda che un numero è primo quando è divisibile solamente per 1 e per se stesso. Siano dati:
 - un vettore di *byte* *numeri* contenente DIM elementi (DIM dichiarato come costante)
 - un vettore di *byte* *risultato* della stessa dimensione che dovrà contenere, per ogni numero analizzato, un valore logico 1 se il numero nella stessa posizione è primo e 0 se non lo è. Tale vettore sarà modificato dal programma.