

Calcolatori Elettronici

Esercitazione 4

M. Sonza Reorda – M. Monetti

M. Rebaudengo – R. Ferrero

L. Sterpone – E. Vacca

Politecnico di Torino

Dipartimento di Automatica e Informatica

Esercitazione 4 - Obiettivi

- Salti e controllo del flusso del programma
- Array (vettori e matrici)

Esercizio 1

- Si scriva un programma in linguaggio Assembly MIPS che scriva in un vettore definito di 20 elementi di tipo *word* i primi 20 valori della serie di Fibonacci.
- Serie di Fibonacci
 - $\text{vet}[i] = \text{vet}[i-1] + \text{vet}[i-2] \Rightarrow \text{vet} = 1, 1, 2, 3, 5, 8, \dots$

Esercizio 2

- Scrivere un programma 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 dell'intero digitato dall'utente, il programma deve eseguire:
 - $0 \rightarrow \text{res} = a+b$
 - $1 \rightarrow \text{res} = a-b$
 - $2 \rightarrow \text{res} = a*b$
 - $3 \rightarrow \text{res} = a/b$ (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

```
                .data
tab:            .word somma, sottrazione, multiplic, divisione
                ...
                .code
                ...
                lw $t2, tab($t0)
                jr $t2
somma:          ...
sottrazione:    ...
```

Esercizio 3

- Si scriva un programma MIPS che, dati due vettori di 4 *word* ciascuno come matrici riga e colonna, ne calcoli il prodotto.
- Si ricorda che
Se $x = (x_1, x_2, \dots, x_n)$ e $y = (y_1, y_2, \dots, y_n)$ sono due vettori a n componenti, il prodotto fra il vettore colonna x e il vettore riga y coincide con la matrice di ordine $n \cdot n$ in cui l'elemento di indice ij è dato dal prodotto tra la i -esima componente di x e la j -esima componente di y . In formule:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} (y_1 \quad y_2 \quad \cdots \quad y_n) = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n y_1 & x_n y_2 & \cdots & x_n y_n \end{pmatrix}$$

Esercizio 4

- Si scriva un programma in grado di generare una tavola pitagorica (10x10) e memorizzarla.

Esercizio 5

- Sia data la seguente tabella di *word*:

| | | | | | |
|-----|-----|------|----|-----|---|
| 154 | 123 | 109 | 86 | 4 | ? |
| 412 | -23 | -231 | 9 | 50 | ? |
| 123 | -24 | 12 | 55 | -45 | ? |
| ? | ? | ? | ? | ? | ? |

- Implementare in Assembly MIPS il programma che scriva la somma di ciascuna riga e colonna rispettivamente nell'ultima colonna e riga.