Architettura degli Elaboratori 1 - Corso B

Terzo appello a.a. 2004-2005

7 giugno 2005

1) Compilare in assembler il seguente frammento di programma JAVA, in modo da *ottimizzare* il *tempo di esecuzione* del programma compilato sapendo che `A` è una matrice di N righe ed 2N colonne contenente interi di valore compreso tra 0 e 255, B e' un vettore di 256 posizioni di interi inizializzato a zero, gli interi occupano 4 bytes di memoria

```
k = 33;
if (A[0, 0] < A[1, 2])
  A[0, 0] = k;
i=1;
while (i < N-1)
{
 if ((A[i-1, 2(i-1)] < A[i, 2i]) && (A[i, 2i] < A[i+1, 2(i+1)]))
   A[i+1, 0] = A[i, 2i] + A[N-1, 2(N-1)];
 else
   A[i+1, 0] = A[i, 2i] + A[N-1, 0];

 B[A[i, 2i]]++;

 i++;
}
```

2) In un processore con la stessa PO vista a lezione ed in cui il prelievo e la decodifica delle istruzioni avvengono come visto a lezione, si consideri l'istruzione assembler

SHIFT R_1, R_2

che *sposta in avanti* di una posizione il contenuto di 64 locazioni di memoria, a partire da quella il cui indirizzo è contenuto nel registro di indice R_1. Dopo aver effettuato lo shift, se almeno uno dei valori spostati è risultato negativo, si salta alla istruzione il cui indirizzo è contenuto nel registro R_2, altrimenti si passa ad eseguire l'istruzione successiva. Scrivere il microprogramma per la fase di fetch e di esecuzione della istruzione. Illustrare le eventuali modifiche apportate alla PO per ottimizzare il tempo di esecuzione.

3) Rispondere alle seguenti domande:

a) Illustrare *i tipi di fault* che possono essere generati dalla MMU.

b) Nel caso di una parte controllo multiprogrammata, quali sono gli ingressi e quali le uscite della rete di condizionamento?

c) Descrivere i casi in cui in cui è necessario rimpiazzare un blocco in una cache con *organizzazione direct mapped.*

e) Descrivere come avviene *l'arbitraggio* del bus di I/O del processore