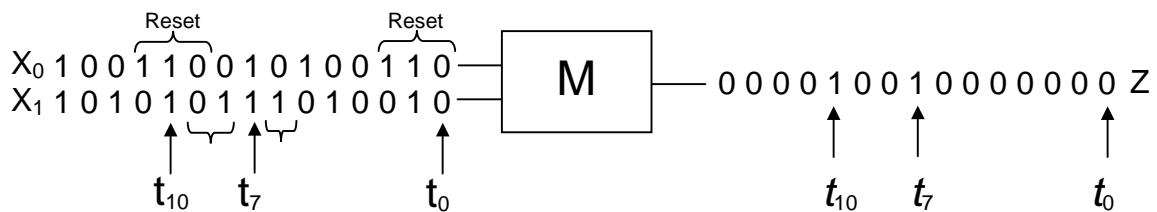


Corso di Calcolatori elettronici A

Compito scritto A – 3/9/2004

- 1) Progettare con un automa a stati finiti ed implementare con sole porte NAND, dopo aver minimizzato il numero delle porte stesse, la macchina sequenziale M, la cui uscita Z, inizialmente a 0, commuta a 1 se tra due 1 ricevuti sulla linea di ingresso X_0 , si è presentato un numero dispari di 1 sulla linea di ingresso X_1 . Il valore 1 dell'uscita è in corrispondenza del secondo 1 di X_0 , Z torna a 0 nel ciclo di clock seguente come illustrato nell'esempio in figura. Quando arriva uno 0 seguito da due 1 sulla linea X_0 , la macchina si riporta nello stato iniziale.



- 2) Un programma in assembler MIPS legge in sequenza i 10 byte in corrispondenza dei seguenti indirizzi: 0x100C8A00, 0x100C8A04, 0x100C8A1C, 0x100C8A18, 0x100C8A3D, 0x100C8A3E, 0x100C8A3F, 0x100C8A26, 0x100C8A27 e 0x100C8A28. Il computer che esegue il codice dispone di una cache per i dati di 64 byte, set-associative a 2 vie e con blocchi di 2 parole.

Supponendo la cache inizialmente vuota:

- Specificare per ognuna delle 10 letture se si tratta di un hit o un miss.
 - Per ogni accesso rappresentare graficamente lo stato della cache evidenziando i tag dei blocchi caricati.
- 3) Scrivere una procedura in assembler MIPS che riceva in ingresso l'indirizzo di un vettore di interi ed il numero dei suoi elementi e per ogni elemento del vettore stesso effettui la seguente assegnazione:

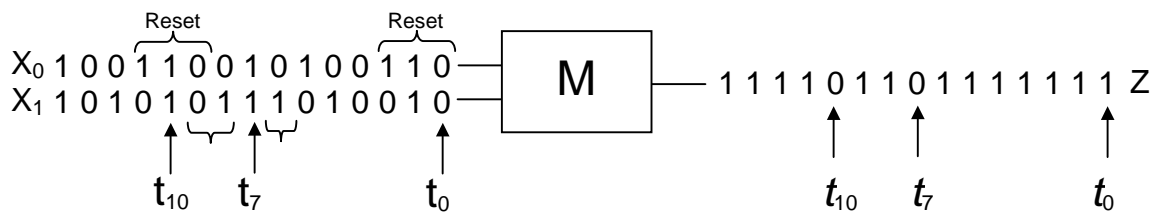
```
if a[i] > 200
    a[i] = 200
else
    a[i] = 5(200 - a[i])
```

Con $a[i]$ si è indicato l'i-esimo elemento del vettore passato alla procedura.

Corso di Calcolatori elettronici A

Compito scritto B – 3/9/2004

- 1) Progettare con un automa a stati finiti ed implementare con sole porte NAND, dopo aver minimizzato il numero delle porte stesse, la macchina sequenziale M, la cui uscita Z, inizialmente a 1, commuta a 0 se tra due 1 ricevuti sulla linea di ingresso X_0 , si è presentato un numero dispari di 1 sulla linea di ingresso X_1 . Il valore 0 dell'uscita è in corrispondenza del secondo 1 di X_0 , Z torna a 1 nel ciclo di clock seguente come illustrato nell'esempio in figura. Quando arriva uno 0 seguito da due 1 sulla linea X_0 , la macchina si riporta nello stato iniziale.



- 2) Un programma in assembler MIPS legge in sequenza i 10 byte in corrispondenza dei seguenti indirizzi: 0x100A8C00, 0x100A8C04, 0x100A8C1C, 0x100A8C18, 0x100A8C3D, 0x100A8C3E, 0x100A8C3F, 0x100A8C26, 0x100A8C27 e 0x100A8C28. Il computer che esegue il codice dispone di una cache per i dati di 64 byte, set-associative a 2 vie e con blocchi di 2 parole.

Supponendo la cache inizialmente vuota:

- Specificare per ognuna delle 10 letture se si tratta di un hit o un miss.
 - Per ogni accesso rappresentare graficamente lo stato della cache evidenziando i tag dei blocchi caricati.
- 3) Scrivere una procedura in assembler MIPS che riceva in ingresso l'indirizzo di un vettore di interi ed il numero dei suoi elementi e per ogni elemento del vettore stesso effettui la seguente assegnazione:

```
if a[i] > 500
    a[i] = 500
else
    a[i] = 2(500 - a[i])
```

Con $a[i]$ si è indicato l'i-esimo elemento del vettore passato alla procedura.