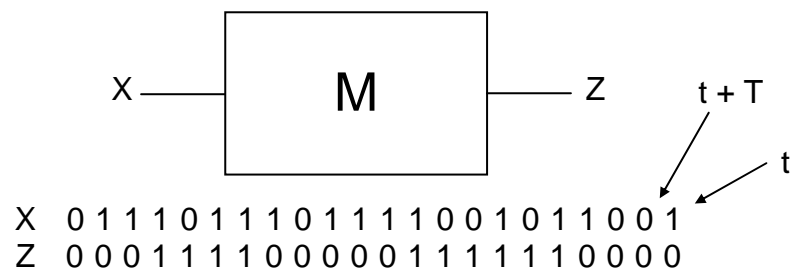


Corso di Calcolatori elettronici A

Compito scritto A – 5/7/2005

- 1) Progettare con un automa di Moore ed implementare con sole porte NAND, dopo aver minimizzato il numero delle porte stesse, la macchina sequenziale M che, dopo aver ricevuto uno 0 seguito da due 1 sulla linea d'ingresso X, commuta il valore sull'uscita Z come evidenziato nell'esempio seguente.



- 2) Un programma in assembler MIPS legge in sequenza i 10 byte in corrispondenza dei seguenti indirizzi: 0x7FADEC98, 0x70BB5717, 0x12BDCEF3, 0x2A2E7E51, 0x7FADEC99, 0x2B7C65B8, 0x2C8D9A57, 0x70BB5718, 0x12BDCEF4 e 0x2A2E7E52. Il computer che esegue il codice dispone della cache per i dati, rappresentata in figura, con blocchi di 4 parole.

0	0x3FD6F64
	0x15BE32D
	0x1646CD2
	0x15173F2
1	0x3FD6F64
	0x385DAB8
	0x95EE77
	0x15173F2

Indicare innanzi tutto il nome della cache utilizzata.

Supponendo la cache inizialmente piena e con i tag dei blocchi caricati indicati in figura, si richiede di specificare per ognuna delle 10 letture se si tratta di un hit o un miss, indicando per gli hit il blocco che contiene il byte richiesto e per i miss il blocco da sostituire considerando la tecnica LRU.

- 3) Scrivere una funzione in assembler MIPS che riceva in ingresso due interi e calcoli il valore restituito secondo il seguente frammento di pseudocodice:

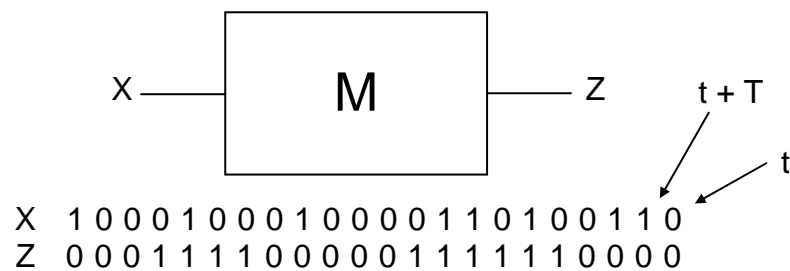
```
if f(a - 470.523) >= g(b) + 534.256 then
    return 0
else
    return 1
```

Dove con $f()$ e $g()$ si sono indicate due funzioni a disposizione del programmatore.

Corso di Calcolatori elettronici A

Compito scritto B – 5/7/2005

- 4) Progettare con un automa di Moore ed implementare con sole porte NAND, dopo aver minimizzato il numero delle porte stesse, la macchina sequenziale M che, dopo aver ricevuto un 1 seguito da due 0 sulla linea d'ingresso X, commuta il valore sull'uscita Z come evidenziato nell'esempio seguente.



- 5) Un programma in assembler MIPS legge in sequenza i 10 byte in corrispondenza dei seguenti indirizzi: 0x7FADEC88, 0x70BB5707, 0x12BDC EE3, 0x2A2E7E41, 0x7FADEC89, 0x2B7C65A8, 0x2C8D9A47, 0x70BB5708, 0x12BDC EE4 e 0x2A2E7E42. Il computer che esegue il codice dispone della cache per i dati, rappresentata in figura, con blocchi di 4 parole.

0	0x3FD6F64
	0x385DAB8
	0x95EE77
	0x15173F2
1	0x3FD6F64
	0x15BE32D
	0x1646CD2
	0x15173F2

Indicare innanzi tutto il nome della cache utilizzata.

Supponendo la cache inizialmente piena e con i tag dei blocchi caricati indicati in figura, si richiede di specificare per ognuna delle 10 letture se si tratta di un hit o un miss, indicando per gli hit il blocco che contiene il byte richiesto e per i miss il blocco da sostituire considerando la tecnica LRU.

- 6) Scrivere una funzione in assembler MIPS che riceva in ingresso due interi e calcoli il valore restituito secondo il seguente frammento di pseudocodice:

```
if h(a) + 577.546 <= k(b - 469.427) then
    return 0
else
    return 1
```

Dove con $h()$ e $k()$ si sono indicate due funzioni a disposizione del programmatore.