

**PROVA SCRITTA DEL CORSO DI**  
**CALCOLATORI ELETTRONICI**  
**NUOVO E VECCHIO ORDINAMENTO DIDATTICO**  
15 Luglio 2008

**NOME:**

**COGNOME:**

**MATRICOLA:**

**ESERCIZIO 1 (NO: 9 punti – VO: 8 punti)**

Si consideri una rete sequenziale avente un ingresso X e un'uscita Z. L'uscita Z = 1 quando viene riconosciuta la sottosequenza 010011. Negli altri casi l'uscita Z = 0.

1. (NO: 4 punti – VO: 3 punti) Disegnare il diagramma degli stati.
2. (NO: 2 punti – VO: 2 punti) Codificare gli stati e scrivere la tabella di flusso. Si scriva poi la tabella delle transizioni qualora si usino flip-flop di tipo D.
3. (NO: 3 punti – VO: 3 punti) Calcolare le forme minime per le variabili di eccitazione dei flip-flop e per l'uscita impiegando le mappe di Karnaugh.

**ESERCIZIO 2 (NO: 10 punti – VO: 8 punti)**

1. (2 punti) Mostrare la suddivisione nei relativi campi TAG, Index, Offset di un indirizzo di memoria primaria di 1 Kbyte, disponendo di una cache da 32 byte, secondo il metodo diretto con blocchi da 4 byte.
2. (NO: 6 punti - VO: 4 punti) Si consideri la sequenza di riferimenti alla memoria indicati come indirizzi di parole (in formato decimale, il primo indirizzo è 0): 389, 719, 387, 697, 306, 308, 194, 198, 699, 310. Indicare i cache *hit* e il contenuto finale della cache, nella configurazione al punto 1. E' sufficiente indicare, per ciascun blocco di cache, l'indirizzo del corrispondente blocco di primaria in esso eventualmente presente.
3. (2 punti) Indicare almeno una possibile alternativa per aumentare il numero di hit in cache, senza alterare la dimensione di memoria primaria e di cache e il metodo di indirizzamento adottato.  
(Suggerimento. Ricordare il principio di località: chiamate vicine in ordine di tempo e di spazio)

**ESERCIZIO 3 (solo NO: 7 punti)**

Implementare in Assembler MIPS una funzione che, dato in ingresso un intero n, calcoli il fattoriale di n (indicato con n!). Sia  $n \rightarrow \$4$ ,  $n! \rightarrow \$5$ .

Si ricordi che:  $n! = \prod_{i=1}^n i$ , e che  $0! = 1$ .

Vincolo: si usi il numero minimo di registri possibile per implementare la funzione.

**ESERCIZIO 3 (solo VO: 7 punti)**

Un certo numero di periferiche sono collegate al calcolatore per mezzo di un bus sincrono. L'arbitraggio viene effettuato secondo lo schema "daisy chain".

1. (4 punti) Mostrare lo schema di collegamento delle periferiche al bus indicando le diverse linee che devono essere presenti nel bus per gestire l'arbitraggio e il trasferimento dati.
2. (2 punti) Descrivere il protocollo usato dal controller del bus per individuare la periferica che ha richiesto l'uso del bus.
3. (2 punti) Descrivere il protocollo per effettuare la lettura di un dato dalla memoria su un bus sincrono quando tale lettura è iniziata dalla periferica.

**ESERCIZIO 4 (NO: 7 punti – VO: 6 punti)**

La memoria di un calcolatore è gestita con una tecnica di 'paginazione su richiesta'. Si consideri la seguente richiesta di pagine: 1, 3, 2, 4, 5, 3, 1, 3, 3, 2, 4, 1. Mostrare il "page trace", calcolare il numero di "page faults" e l'"hit ratio" per una memoria di dimensione pari a quattro pagine, nei seguenti due casi:

- 1) (NO: 3,5 punti – VO: 3 punti) strategia di rimpiazzamento delle pagine FIFO.
- 2) (NO: 3,5 punti – VO: 3 punti) strategia di rimpiazzamento delle pagine LRU.

**ESERCIZIO 5 (solo VO: 4 punti)**

Spiegare in modo chiaro e sintetico la gestione della memoria in un sistema operativo, con particolare riferimento alla paginazione, alla segmentazione e ai vantaggi e svantaggi di queste due tecniche.

# ESERCIZIO 1

## Soluzione.

Grafo degli stati

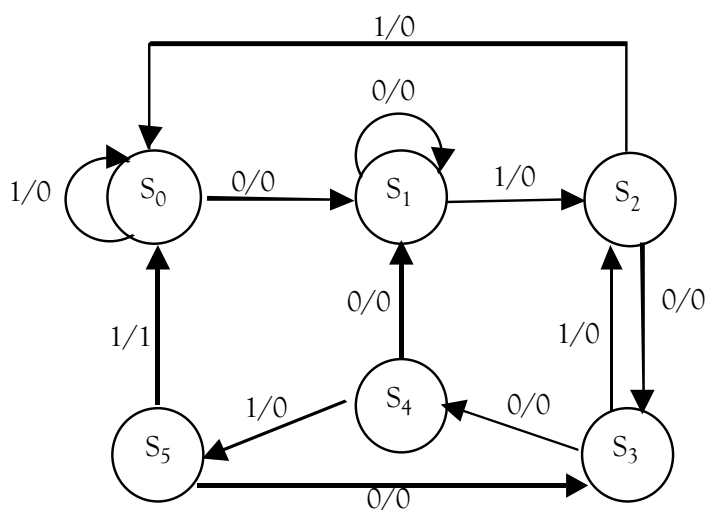


Tabella di flusso

Stato iniziale	Stato finale/uscita	
	x = 0	x = 1
S <sub>0</sub>	S <sub>1</sub> /0	S <sub>0</sub> /0
S <sub>1</sub>	S <sub>1</sub> /0	S <sub>2</sub> /0
S <sub>2</sub>	S <sub>3</sub> /0	S <sub>0</sub> /0
S <sub>3</sub>	S <sub>4</sub> /0	S <sub>2</sub> /0
S <sub>4</sub>	S <sub>1</sub> /0	S <sub>5</sub> /0
S <sub>5</sub>	S <sub>3</sub> /0	S <sub>0</sub> /1

Codifica degli stati (3 bit - Y<sub>2</sub>Y<sub>1</sub>Y<sub>0</sub>)

S<sub>0</sub> → 000      S<sub>1</sub> → 001  
S<sub>2</sub> → 010      S<sub>3</sub> → 011  
S<sub>4</sub> → 100      S<sub>5</sub> → 101

Tabella delle transizioni (FF-D) Y' indica lo stato futuro

Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	x	Y' <sub>2</sub>	D <sub>2</sub>	Y' <sub>1</sub>	D <sub>1</sub>	Y' <sub>0</sub>	D <sub>0</sub>	z
0	0	0	0	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1	0
0	0	1	1	0	0	1	1	0	0	0
0	1	0	0	0	0	1	1	1	1	0
0	1	0	1	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	1	1	0	0	0
1	0	0	0	0	0	0	0	1	1	0
1	0	0	1	1	1	0	0	1	1	0
1	0	1	0	0	0	1	1	1	1	0
1	0	1	1	0	0	0	0	0	0	1
1	1	0	0	d	d	d	d	d	d	d
1	1	1	1	d	d	d	d	d	d	d

Y <sub>2</sub> Y <sub>1</sub>	Y <sub>0</sub> x			
	00	01	11	10
00			d	
01			d	1
11			d	
10		1	d	

$D_2 = Y_2 \bar{Y}_0 x + Y_1 Y_0 \bar{x}$

Y <sub>2</sub> Y <sub>1</sub>	Y <sub>0</sub> x			
	00	01	11	10
00		1	d	
01			d	
11	1	1	d	
10			d	1

$D_1 = Y_1 Y_0 x + Y_2 Y_0 x + Y_2 Y_0 \bar{x}$

Y <sub>2</sub> Y <sub>1</sub>	Y <sub>0</sub> x			
	00	01	11	10
00	1	1	d	1
01			d	1
11			d	
10	1		d	1

$D_0 = Y_1 x + Y_0 x + Y_2 Y_0$

Y <sub>2</sub> Y <sub>1</sub>	Y <sub>0</sub> x			
	00	01	11	10
00			d	
01			d	
11			d	1
10			d	

$z = Y_2 Y_0 x$

## ESERCIZIO 2

### Soluzione:

1. <TAG 5 bit> <Cache Index 3 bit> <Offset 2 bit>
2. Applicando per ciascuna chiamata le formule per il calcolo di block frame e cache index, si ricavano i seguenti valori per ciascun set:

<b>Indirizzo</b>	389	719	387	697	306	308	194	198	699	310
<b>Blocco primaria</b>	97	179	96	174	76	77	48	49	174	77
<b>Blocco cache</b>	1	3	0	6	4	5	0	1	6	5
<b>Hit</b>									X	X

Il contenuto finale della cache è dunque:

<b>Blocco Cache</b>	0	1	2	3	4	5	6	7
<b>Blocco Primaria</b>	48	49		179	76	77	174	

3. La cosa più semplice per ridurre il numero di miss in cache è accrescere la grandezza dei blocchi. Ciò si può notare dal fatto che le richieste in primaria distano spesso meno di otto parole (e.g. prima e terza chiamata, quinta, sesta e ultima chiamata). Ad es. possiamo utilizzare blocchi da 8 byte. In questo caso la cache conterrà solo 4 blocchi, con la seguente sequenza:

<b>Indirizzo</b>	389	719	387	697	306	308	194	198	699	310
<b>Blocco primaria</b>	48	89	48	87	38	38	24	24	87	38
<b>Blocco cache</b>	0	1	0	3	2	2	0	0	3	2
<b>Hit</b>			X			X		X	X	X

Come si nota il numero di hit è passato da 2 a 5.

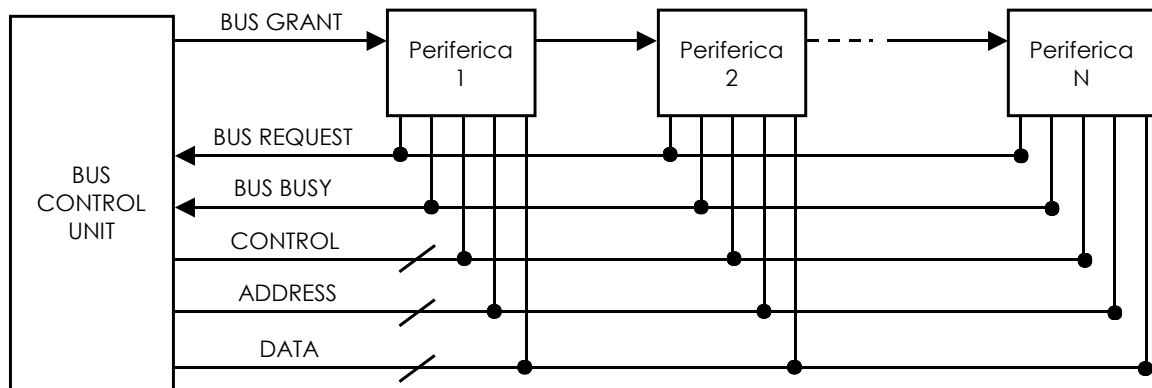
### ESERCIZIO 3 NO

#### Soluzione.

```
fattoriale:  addi $5, $0, 1
loop:       beq $0, $4, exit
            mul $5, $5, $4
            subi $4, $4, 1
            j  loop
exit:       jr $31
```

### ESERCIZIO 3 VO

#### Soluzione.



- 1) Le linee necessarie per il bus sono indicate in figura. Le linee di bus request, grant e busy, sono utilizzate per effettuare l'arbitraggio, mentre le linee control, address e data vengono usate per trasferire i dati. N.B. le linee di control, address e data sono in realtà "gruppi" di linee. La motivazione è contenuta nella risposta 3.
- 2) Arbitraggio in daisy chain: La periferica invia il segnale di richiesta del bus sulla linea bus request, se la linea bus busy indica che il bus è libero. Il controllore del bus invia in risposta il segnale di bus grant sull'apposita linea. Questo segnale viene propagato serialmente fra tutte le periferiche, fino a raggiungere la periferica che ha fatto richiesta, che blocca la propagazione del segnale di bus grant e attiva il segnale di bus busy. (in caso di richieste contemporanee la propagazione del bus grant viene bloccata dalla prima periferica che ha fatto richiesta). A questo punto la periferica può trasferire i dati.
- 3) Sulle linee di controllo viene inviato il segnale sul tipo di trasferimento, in questo caso lettura (ci sono più linee, ciascuna per un tipo di trasferimento come lettura, scrittura, trasferimento di blocchi, ecc.), e contemporaneamente l'indirizzo di memoria che deve essere letto sulla linea indirizzi (address). Trascorso il tempo di ciclo della memoria, il dato viene inviato sulla linea data che può consentire il trasferimento contemporaneo di un certo numero di byte.

**ESERCIZIO 4****Soluzione.**

Il "page trace" nel caso di strategia di rimpiazzamento delle pagine FIFO (x = hit)

Tempo	1	2	3	4	5	6	7	8	9	10	11	12
Richieste	1	3	2	4	5	3	1	3	3	2	4	1
Pagine	1	3	2	4	5	5	1	3	3	2	4	4
		1	3	2	4	4	5	1	1	3	2	2
			1	3	2	2	4	5	5	1	3	3
				1	3	3	2	4	4	5	1	1
Hit						x			x			x

Page faults: 9 - Hit ratio =  $3/12 = 1/4$

"Page trace" nel caso di strategia di rimpiazzamento delle pagine LRU (x = hit)

Tempo	1	2	3	4	5	6	7	8	9	10	11	12
Richieste	1	3	2	4	5	3	1	3	3	2	4	1
Pagine	1	3	2	4	5	3	1	3	3	2	4	1
		1	3	2	4	5	3	1	1	3	2	4
			1	3	2	4	5	5	5	1	3	2
				1	3	2	4	4	4	5	1	3
Hit						x		x	x			x

Page faults: 8 - Hit ratio =  $4/12 = 1/3$

**ESERCIZIO 5****Soluzione.**

Vedere le dispense del corso.