

**PROVA SCRITTA DEL CORSO DI**  
**CALCOLATORI ELETTRONICI**  
**NUOVO ORDINAMENTO DIDATTICO (7 CFU)**  
29 Giugno 2010

**NOME:**

**COGNOME:**

**MATRICOLA:**

**ESERCIZIO 1 (7 punti)**

Progettare una rete sequenziale che presenti un ingresso X e un'uscita Z posta a 1 ogni volta che viene riconosciuta la sequenza 01101.

Si richiede:

1. (4 punti) il diagramma degli stati, la tabella di flusso e la tabella delle transizioni;
2. (3 punti) il calcolo delle forme minime delle variabili di eccitazione dei flip flop con le mappe di Karnaugh. Si usino flip flop JK. Calcolare anche la rete combinatoria per l'uscita Z.

**ESERCIZIO 2 (6 punti)**

1. (3 punti) Sia dato un disco con le seguenti caratteristiche: velocità di rotazione pari a 6000 giri al minuto, tempo necessario alla testina per spostarsi da una traccia alla successiva uguale a 0.5 ms, settori da 1 KB, 200 settori per traccia. Calcolare il tempo medio di lettura di un file da 10 KB sapendo che la testina si trova inizialmente in un punto qualunque del disco e che la distanza media tra due settori successivi del file è pari a 2 tracce.
2. (3 punti) Utilizzando il tempo medio di accesso al disco calcolato al punto precedente, si calcoli l'hit ratio di cache minimo ( $H_c$ ), relativo ad una gerarchia a 3 livelli (cache, memoria primaria, disco), affinché il tempo medio di accesso alla gerarchia sia 30 ms. Siano dati i seguenti altri dati:  $H_p = 0.8$ ;  $T_p = 20$  ms;  $T_c = 10$  ms, dove  $H_p$  è l'hit ratio della memoria primaria,  $T_p$  il tempo medio di accesso alla memoria primaria e  $T_c$  il tempo medio di accesso alla memoria cache.

**ESERCIZIO 3 (7 punti)**

Si scriva il codice Assembly MIPS di una funzione che, dati tre vettori di N interi u, v, w, scriva nella posizione w(i) il valore  $u(i)-v(i)$ , se  $u(i) < v(i)$ , e il valore  $u(i)+v(i)$ , altrimenti. Si consideri che gli indirizzi iniziali dei vettori u, v, w siano memorizzati in \$4, \$5, \$6, rispettivamente, e che N sia memorizzato in \$7. In altri termini il codice MIPS può implementare la funzione C:

```
void elabora(int *u, int *v, int *w, int N) {
    int i;
    for(i=0; i<N; i++)
        if(u(i)<v(i))
            w(i)=u(i)-v(i);
        else
            w(i)=u(i)+v(i);
}
```

**ESERCIZIO 4 (7 punti)**

Si consideri un calcolatore in cui la CPU esegue  $10^5$  istruzioni/s. L'esecuzione di una istruzione richiede 5 cicli di clock, 3 dei quali tengono occupato il bus di sistema. Si ipotizzi che il 85% dell'Instruction Rate sia usato dalla CPU per eseguire programmi che non contengono trasferimenti di I/O. L'ampiezza della linea dati del bus è pari a 32 bit.

Si consideri il caso in cui il trasferimento dei dati avvenga mediante IO da programma, con le seguenti 4 istruzioni:

- a. LOAD parola dalla periferica al registro CPU
  - b. STORE parola da registro CPU a memoria
  - c. generazione indirizzo di memoria successivo
  - d. conteggio dati da trasferire.
1. (3 punti) Calcolare la massima frequenza di trasferimento dati ottenibile (espressa in kB/s) fra una periferica collegata al bus di sistema e la memoria principale.
  2. (2 punti) Calcolare la massima frequenza di trasferimento dati ottenibile (espressa in kB/s) nel caso in cui si usi la modalità "transparent" DMA. Si ipotizzi che una operazione di lettura/scrittura della memoria richieda un ciclo di clock.
  3. (2 punti) Spiegare quali 'passi' sostituiscono le istruzioni nel caso DMA.

**ESERCIZIO 5 (6 punti)**

1. (3 punti) Il sistema operativo di un calcolatore partiziona dinamicamente la memoria disponibile. Si supponga che ad un dato istante lo stato della memoria sia quello in figura, coi processi P1, ..., P4. Indicare una possibile azione per una gestione efficiente della memoria, supponendo che si debbano allocare anche i seguenti processi: P5 da 75KB, P6 da 40KB, P7 da 15KB.

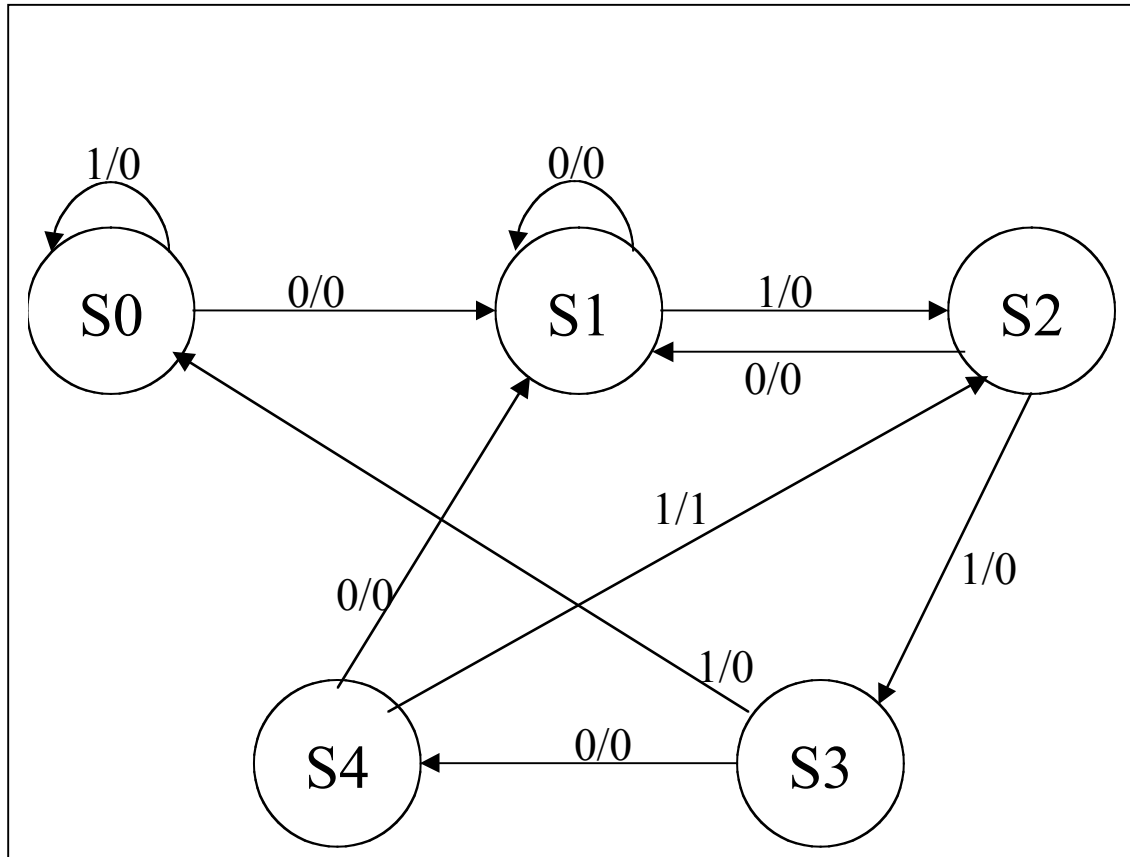
**Memoria**

P1 (100KB)
Vuoto (50KB)
P2 (200KB)
P3 (100KB)
Vuoto (30KB)
P4 (50 KB)
Vuoto (50KB)

2. (3 punti) La memoria di un calcolatore è gestita con una tecnica di 'paginazione su richiesta'. Si consideri la seguente richiesta di pagine: 2, 3, 4, 3, 2, 4, 3, 2, 4, 5, 6, 7, 5, 4, 2, 4, 5, 2, 6, 7, 2. Se la memoria contiene complessivamente quattro pagine calcolare il numero di 'page fault' nel caso di strategia di rimpiazzamento delle pagine LRU.

**ESERCIZIO 1**  
**Soluzione**

Il diagramma degli stati è il seguente:



La tabella di flusso è data da:

Stato presente	Stato successivo/Uscita	
	X=0	X=1
S0	S1/0	S0/0
S1	S1/0	S2/0
S2	S1/0	S3/0
S3	S4/0	S0/0
S4	S1/0	S2/1

Per codificare 5 stati occorrono tre flip flop. La codifica è la seguente:

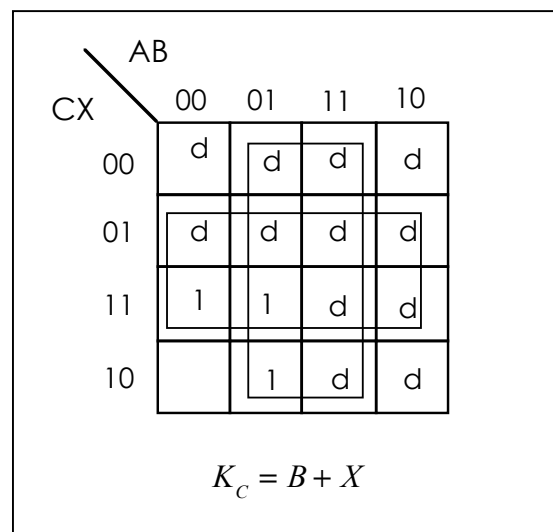
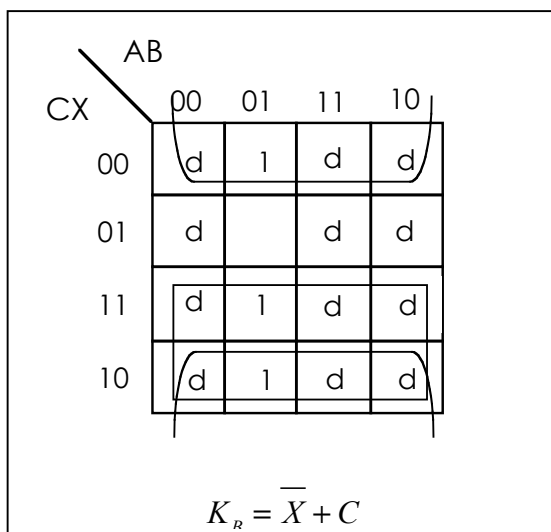
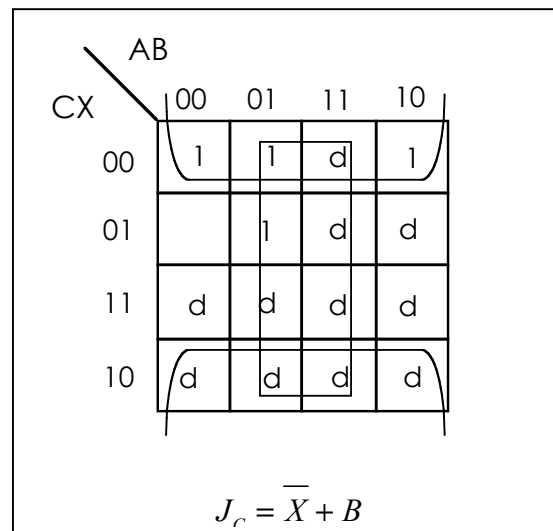
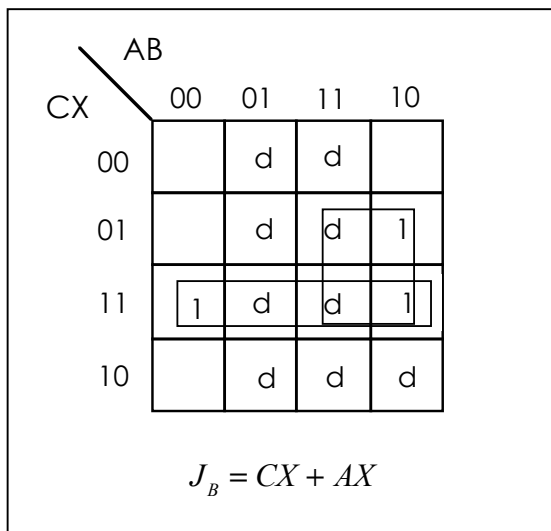
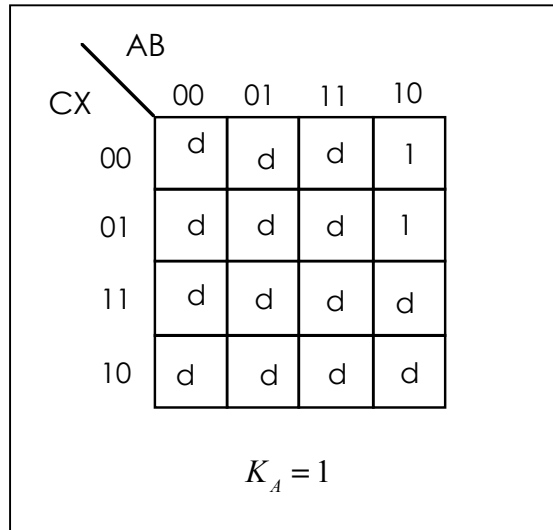
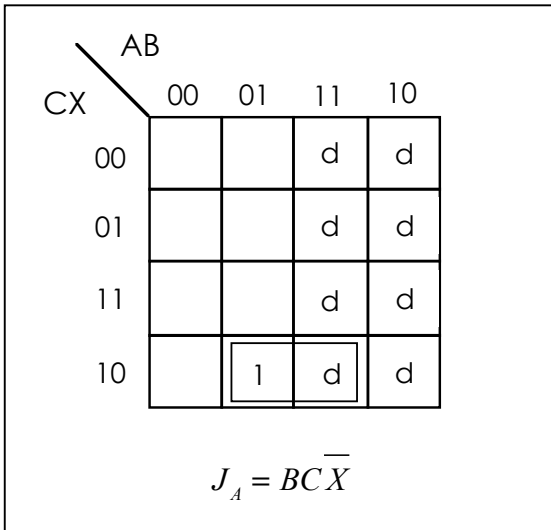
S0  $\rightarrow$  0 0 0; ...; S4  $\rightarrow$  1 0 0. Nel seguito indicheremo ciascun bit della codifica con le lettere A, B, C. L'apice indicherà il bit nell'istante successivo a quello considerato.

A partire dalla tabella di eccitazione del flip flop JK:

Q	Q'	J	K
0	0	0	D
0	1	1	D
1	0	D	1
1	1	D	0

A	B	C	X	A'	Ja	Ka	B'	Jb	Kb	C'	Jc	Kc	Z
0	0	0	0	0	0	D	0	0	D	1	1	D	0
0	0	0	1	0	0	D	0	0	D	0	0	D	0
0	0	1	0	0	0	D	0	0	D	1	D	0	0
0	0	1	1	0	0	D	1	1	D	0	D	1	0
0	1	0	0	0	0	D	0	D	1	1	1	D	0
0	1	0	1	0	0	D	1	D	0	1	1	D	0
0	1	1	0	1	1	D	0	D	1	0	D	1	0
0	1	1	1	0	0	D	0	D	1	0	D	1	0
1	0	0	0	0	D	1	0	0	D	1	1	D	0
1	0	0	1	0	D	1	1	1	D	0	0	D	1
1	0	1	0	D	D	D	D	D	D	D	D	D	D
1	0	1	1	D	D	D	D	D	D	D	D	D	D
1	1	0	0	D	D	D	D	D	D	D	D	D	D
1	1	0	1	D	D	D	D	D	D	D	D	D	D
1	1	1	0	D	D	D	D	D	D	D	D	D	D
1	1	1	1	D	D	D	D	D	D	D	D	D	D

Ora possiamo disegnare le mappe di Karnaugh



Infine, per quanto riguarda l'uscita Z:

AB		00	01	11	10
CX	00			d	
	01			d	1
	11			d	d
	10			d	d

$$Z = AX$$

## ESERCIZIO 2

### Soluzione

1.  $TROT = 60 / 6000 = 10 \text{ ms}$   
 $TLAT = TROT / 2 = 5 \text{ ms}$  (tempo di latenza)  
 $Tlett = TROT / 200 = 50 \mu\text{s}$  (tempo di lettura di un settore)  
 $Tpos = 0.5 \times 2 = 1 \text{ ms}$   
Numero di settori richiesti per leggere 10 KB:  $10\text{KB}/1\text{KB} = 10$   
Tempo di lettura del file da 10 KB:  
 $= 10 * (TLAT + TPOS + Tlett) = 10 * (5 \text{ ms} + 1 \text{ ms} + 50 \mu\text{s}) \approx 60 \text{ msec}$
2. Per calcolare  $H_c$  basta esplicitarlo dalla formula:

$$\bar{T} = H_c T_c + (H_p - H_c)(T_p + T_c) + (1 - H_p)(T_d + T_p + T_c)$$

$$H_c = (T_c + T_p + T_d - H_p T_d - \bar{T}) / T_p = 0.6$$

### ESERCIZIO 3

#### Soluzione

$\$8 \leftarrow i; \$9 \leftarrow u(i) < v(i)$

$\$10 \leftarrow u(i); \$11 \leftarrow v(i); \leftarrow \$12 \leftarrow w(i)$

```
elabora:    addi $29, $29, -20
            sw $8, 0($29)
            sw $9, 4($29)
            sw $10, 8($29)
            sw $11, 12($29)
            sw $12, 16($29)
            move $8, $0
```

```
for:        beq $8, $7, exit
            lw $10, 0($4)
            lw $11, 0($5)
            slt $9, $10, $11
            bne $9, $0, sottrai
            add $12, $10, $11
```

```
continue:   sw $12, 0($6)
            addi $8, $8, 1
            addi $4, $4, 4
            addi $5, $5, 4
            addi $6, $6, 4
            j for
```

```
exit:       lw $8, 0($29)
            lw $9, 4($29)
            lw $10, 8($29)
            lw $11, 12($29)
            lw $12, 16($29)
            addi $29, $29, 20
            jr $31
```

```
sottrai:    sub $12, $10, $11
            j continue
```



## ESERCIZIO 4

### Soluzione.

1. Nel caso di trasferimento mediante I/O da programma, per trasferire una parola occorrono 4 istruzioni. La CPU è impegnata per l'85% del tempo a eseguire istruzioni che non coinvolgono l'I/O, dunque può usare solo il 15% del tempo per eseguire istruzioni di trasferimento dati con periferiche. In termini di istr./sec questo tempo è pari a  $0.15 \times 10^5 \text{ istr./s} = 1.5 \times 10^4 \text{ istr./s}$ . Dal momento che per trasferire una parola servono due istruzioni, la velocità di trasferimento è pari a:  
 $1.5 \times 10^4 \text{ istr./s} / (4 \text{ istr./parola}) = 3750 \text{ parole/s}$ . La dimensione di una parola è pari a 32 bit (4 byte), da cui si ricava la velocità di trasferimento di **14.65 kB/s**.
2. Nel caso di 'trasparent DMA' posso trasferire i dati tutte le volte che il bus di sistema è libero. Nel caso in esame questo tempo è pari alla somma del 15% del tempo lasciato libero dall'esecuzione di istruzioni che non coinvolgono I/O, più i due cicli/istruzione in cui il bus è libero. Pertanto durante l'85% del tempo posso trasferire due parole/istr.:  
 $0.85 \times 2 \text{ parole/istr} \times 10^5 \text{ istr./s} = 1.7 \times 10^5 \text{ parole/s}$   
Nel restante 15% del tempo posso trasferire 5 parole/istr.:  
 $0.15 \times 5 \text{ parole/istr.} \times 10^5 \text{ istr./s} = 0.75 \times 10^5 \text{ parole/s}$   
In **totale**, nel caso di trasferimento con DMA la velocità totale di trasferimento è pari a:  **$(1.7 + 0.75) \times 10^5 \text{ parole/s} = 2.45 \times 10^5 \text{ parole/s} = 239 \text{ kB/s}$** .
3. Il 'controller' DMA esegue le operazioni di generazione indirizzi e conteggio dati trasferiti usando registri interni al controller oltre, ovviamente, a trasferire i dati dalla periferica alla memoria.

## ESERCIZIO 5

### Soluzione.

1. La memoria è evidentemente frammentata. Poiché viene utilizzato un partizionamento dinamico, che dobbiamo considerare rilocabile in assenza di altre informazioni provenienti dal testo, per gestire efficientemente la memoria è necessario provvedere a rilocare le partizioni esistenti e compattare le lacune presenti in memoria. Come si può notare sono disponibili complessivamente 130KB, quindi tutti e tre i processi citati nel testo P5...P7 possono essere allocati senza problemi, ma solo dopo le operazioni suddette.

2. 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	13	14	15	16	17	18	19	20	21
Richieste	2	3	4	3	2	4	3	2	4	5	6	7	5	4	2	4	5	2	6	7	2
Pagine	2	3	4	3	2	4	3	2	4	5	6	7	5	4	2	4	5	2	6	7	2
		2	3	4	3	2	4	3	2	4	5	6	7	5	4	2	4	5	2	6	7
			2	2	4	3	2	4	3	2	4	5	6	7	5	5	2	4	5	2	6
										3	2	4	4	6	7	7	7	7	4	5	5
Hit				x	x	x	x	x	x				x	x		x	x	x			x

Hit ratio = 12/21; Page faults = 9