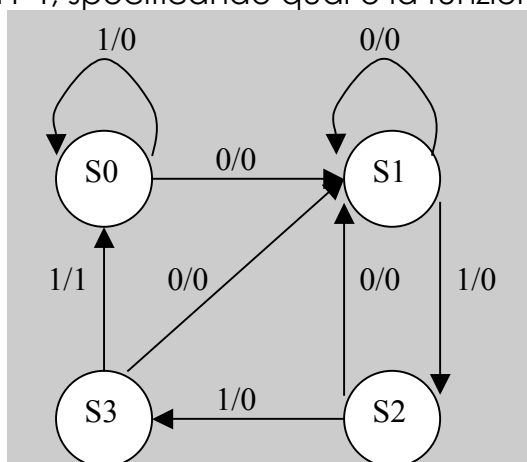


**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO**
14 Settembre 2004

MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI

ESERCIZIO 1 (10 punti)

- (1) (5 punti) Progettare un Decoder 3 to 8. Discutere in modo chiaro e sintetico qual è la funzione svolta dal Decoder nell'implementazione di un modulo di memoria.
- (2) (5 punti) Dato il seguente grafo degli stati, implementare la corrispondente rete sequenziale usando FF-T, specificando qual è la funzione svolta dalla rete data.



Soluzione.

(a) Vedi dispense del corso (cap. 2 pagg. 40-42; cap. 4 pag. 41).

(b)

La rete in oggetto presenta un solo ingresso e una sola uscita che indichiamo rispettivamente con X e Z.

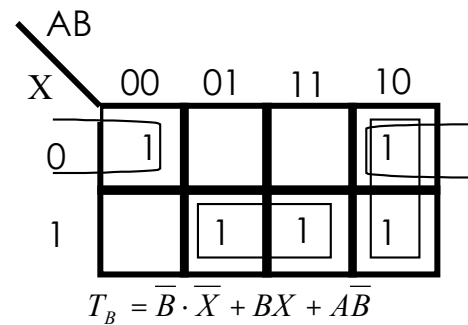
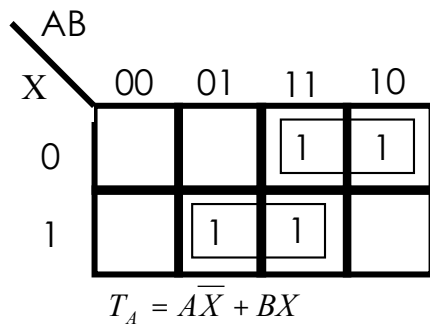
La tabella di flusso è data da:

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

Utilizzando flip flop T, e identificando con A e B i bit codificanti i quattro stati S0...S3 la tabella di transizione risulta:

A	B	X	A'	TA	B'	TB	Z
0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0
0	1	1	1	1	0	1	0
1	0	0	0	1	1	1	0
1	0	1	1	0	1	1	0
1	1	0	0	1	1	0	0
1	1	1	0	1	0	1	1

Le mappe di Karnaugh per la minimizzazione delle funzioni di transizione dello stato sono le seguenti:



La funzione svolta da questa rete logica è evidentemente il riconoscimento della sequenza 0111.

ESERCIZIO 2 (8 punti)

Scrivere una funzione Assembler MIPS tale che, dati tre vettori di interi u , v , w , di dimensione N , copi nella posizione i -esima del vettore w il valore $v[i]$, se esso è sottomultiplo di $u[i]$, altrimenti copi il valore 1. Il passaggio dei parametri avviene nel modo seguente: $\&u[0] \rightarrow \$4$, $\&v[0] \rightarrow \$5$, $\&w[0] \rightarrow \$7$, $N \rightarrow \$7$.

Si supponga di potere fare uso di una funzione $\text{div}(x,y)$ che riceva in ingresso due interi x e y e calcoli quoziente e resto della divisione x/y . Per questa funzione il passaggio dei parametri avviene nel modo seguente: $x \rightarrow \$4$, $y \rightarrow \$5$, quoziente $\rightarrow \$6$, resto $\rightarrow \$7$.

In altri termini il codice Assembler MIPS potrebbe implementare il seguente codice C:

```
void sottomultipli (int *u, int *v, int *w, int N)
{
    int i;
    int q, r;

    for(i=0; i<N; i++)
    {
        div(u[i], v[i], &q, &r); //q e r contengono quoziente e resto
        if(r==0)
            w[i]=v[i];
        else
            w[i]=1;
    }
}
```

Soluzione.

Nella seguente soluzione:

$\$10 \leftarrow$ copia di $\&u[0]$; $\$11 \leftarrow$ copia di $\&v[0]$; $\$12 \leftarrow$ copia di $\&w[0]$; $\$13 \leftarrow$ copia di N
 $\$8 \leftarrow i$, $\$1 \leftarrow 1$.

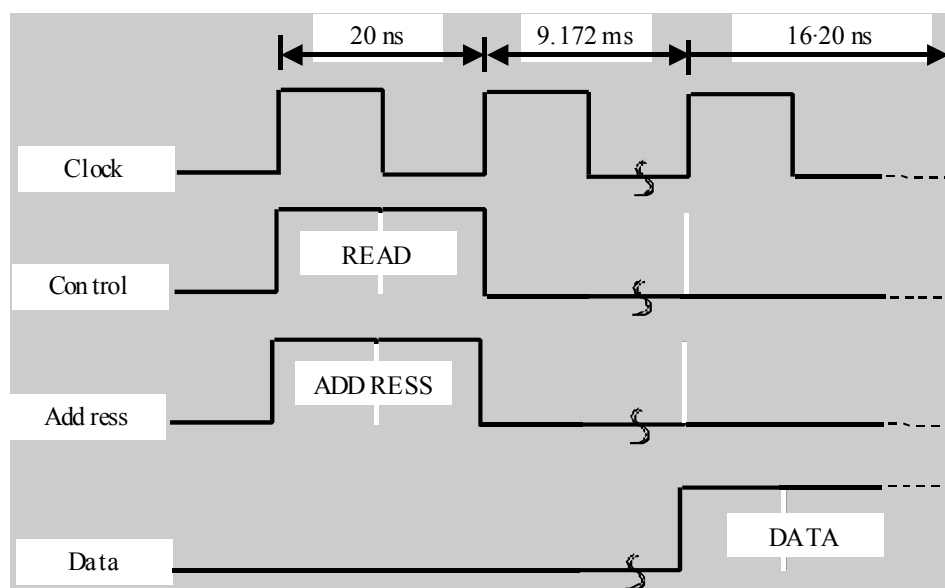
<pre>sottomultipli: addi \$29, \$29, -28 sw \$1, 0(\$29) sw \$8, 4(\$29) sw \$10, 8(\$29) sw \$11, 12(\$29) sw \$12, 16(\$29) sw \$13, 20(\$29) sw \$31, 24(\$29) move \$10, \$4 move \$11, \$5 move \$12, \$6 move \$13, \$7 addi \$1, \$0, 1 move \$8, \$0 for: beq \$8, \$13, exit lw \$4, 0(\$10) lw \$5, 0(\$11) sw \$1, 0(\$12)</pre>	<pre>jal div bne \$7, \$0, continue sw \$5, 0(\$12) continue: addi \$10, \$10, 4 addi \$11, \$11, 4 addi \$12, \$12, 4 addi \$8, \$8, 1 j for exit: lw \$1, 0(\$29) lw \$8, 4(\$29) lw \$10, 8(\$29) lw \$11, 12(\$29) lw \$12, 16(\$29) lw \$13, 20(\$29) lw \$31, 24(\$29) addi \$29, \$29, 28 jr \$31</pre>
---	--

ESERCIZIO 3 (7 punti)

- (a) (3 punti) Un disco rigido presenta le seguenti caratteristiche: velocità di rotazione 7200 giri/min; tempo medio di posizionamento 2 msec; capacità di una traccia 100 kbyte. Calcolare il tempo medio di lettura di un blocco di 64 byte.
- (b) (4 punti) I trasferimenti di dati dal disco descritto al punto precedente alla memoria avvengono su un bus sincrono, il cui clock ha frequenza pari a 50 MHz. Sapendo che ciascun trasferimento sul bus ha una durata pari a un ciclo di clock e che l'ampiezza della linea dati del bus è di 32 bit, calcolare il tempo totale necessario per il trasferimento di un blocco di 64 byte dal disco alla memoria principale. Nello svolgimento della soluzione, mostrare il diagramma temporale dei trasferimenti sul bus. [Nota: il disco trasferisce tutto il blocco richiesto in un *buffer* che è collegato al bus di sistema].

Soluzione

- 1) Tempo di rotazione $t_{rot} = 60/7200 = 8.33 \text{ ms}$
Tempo di latenza $t_{lat} = 1/2 t_{rot} = 4,167 \text{ ms}$
Tempo di trasferimento di un blocco da 64 byte $t_{trasf} = 64 \cdot 60 / (100k \cdot 7200) = 5.21 \text{ } \mu\text{s}$
Tempo di lettura totale $t_{lett} = t_{pos} + t_{lat} + t_{trasf} = 6.172 \text{ ms}$
(NOTA il tempo di trasferimento è praticamente irrilevante).
- 2) In un bus sincrono il trasferimento avviene in tre passi: invio del segnale di READ sulla linea di controllo e contemporaneamente l'indirizzo di memoria sulla linea indirizzi; tempo di "latenza" dovuto al ritardo con cui la periferica rende disponibili i dati nel buffer di uscita (nel caso del disco questo tempo è pari a 9.167 ms); trasferimento dei dati dal buffer alla memoria. In questo caso dovendo trasferire 64 byte su un bus con ampiezza dati di 32 bit, saranno necessari 16 trasferimenti consecutivi.
Tempo totale di trasferimento: $T(\text{READ \& ADDRESS}) + T_{disco} + T_{trasf} = 20\text{ns} + 9.172\text{ms} + 16 \cdot 20\text{ns} = 9,17234 \text{ ms}$. (NOTA: il "collo di bottiglia" è il disco).



ESERCIZIO 4 (8 punti)

Siano dati i seguenti processi:

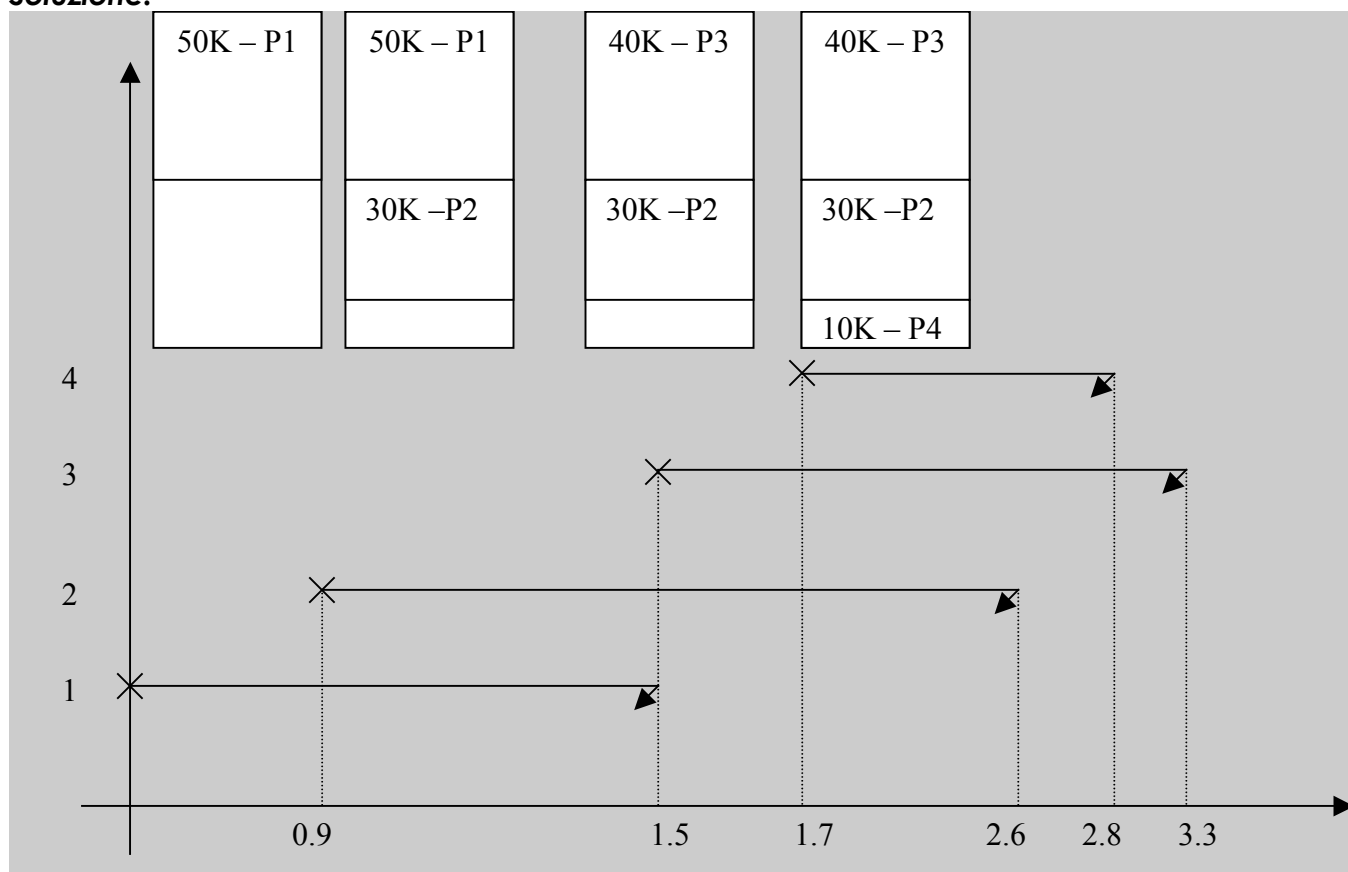
Processo	Tarrivo	TCPU	Memoria
1	0.0	1.2	50k
2	0.9	0.7	30k
3	1.5	1.0	40k
4	1.7	0.4	10k

La memoria disponibile, di 100K, è gestita dal sistema operativo mediante partizioni dinamiche non rilocabili. All'istante iniziale, si consideri la memoria vuota.

I processi sono gestiti con strategia FIFO multiprogrammata "round robin".

- Mostrare col metodo grafico lo sviluppo di ciascun processo e lo stato della memoria (6 punti).
- Definire e calcolare il tempo di turnaround medio e di turnaround pesato medio (2 punti).

Soluzione.



Processo	Arrivo	Avvio	Termine	Turnaround	W.Tourn.
1	0.0	0.0	1.5	1.5	1.3
2	0.9	0.9	2.6	1.7	2.4
3	1.5	1.5	3.3	1.8	1.8
4	1.7	1.7	2.8	1.1	2.8
Media				1.5	2.1