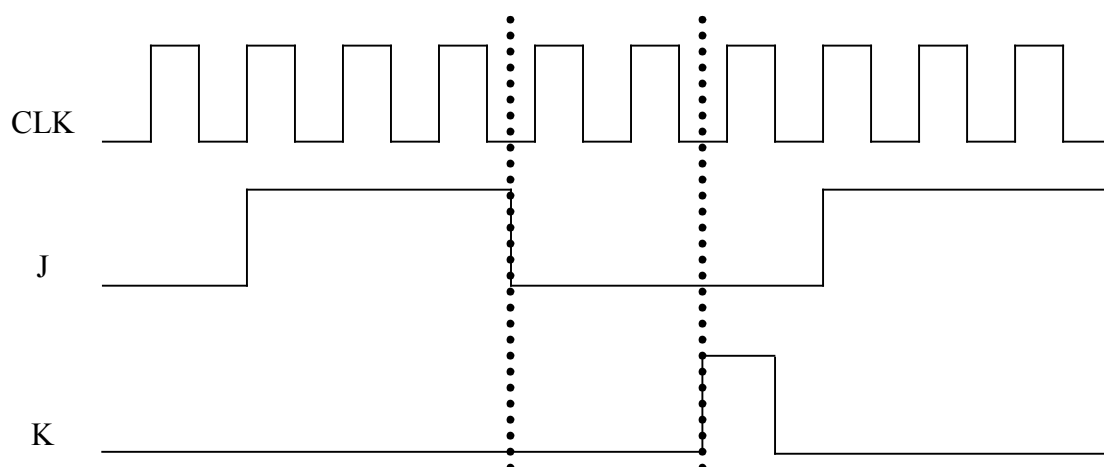


**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO**
2 Luglio 2004

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

ESERCIZIO 1 (9 punti)

- (a) (5 punti) Realizzare un flip flop JK a partire da un flip flop T e una opportuna rete logica. Sintetizzare la rete logica minima usando le mappe di Karnaugh e disegnare il relativo circuito.
- (b) (4 punti) Esprimere le uscite di un latch JK asincrono e di uno sincrono secondo l'andamento dei segnali in figura (CLK è il segnale di sincronismo).



Soluzione.

Esercizio (a)

Tabelle di eccitazione dei flip flop JK e T:

Tabella di eccitazione di un flip-flop JK

$Q(t)$	$Q(t+1)$	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Tabella di eccitazione di un flip-flop T

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

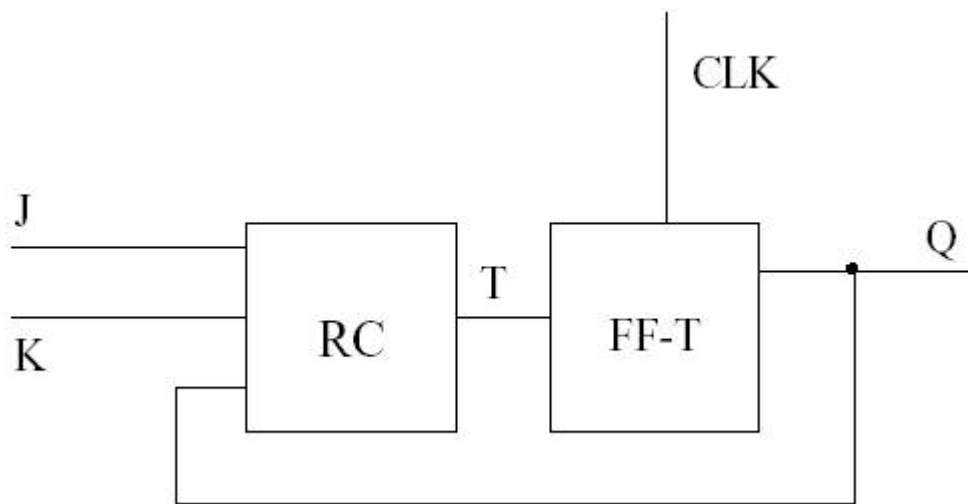
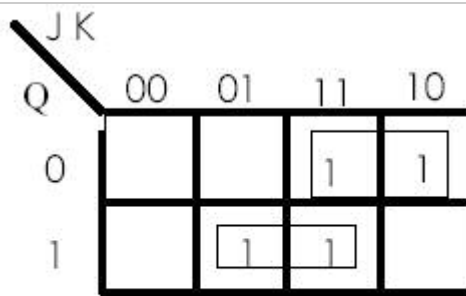


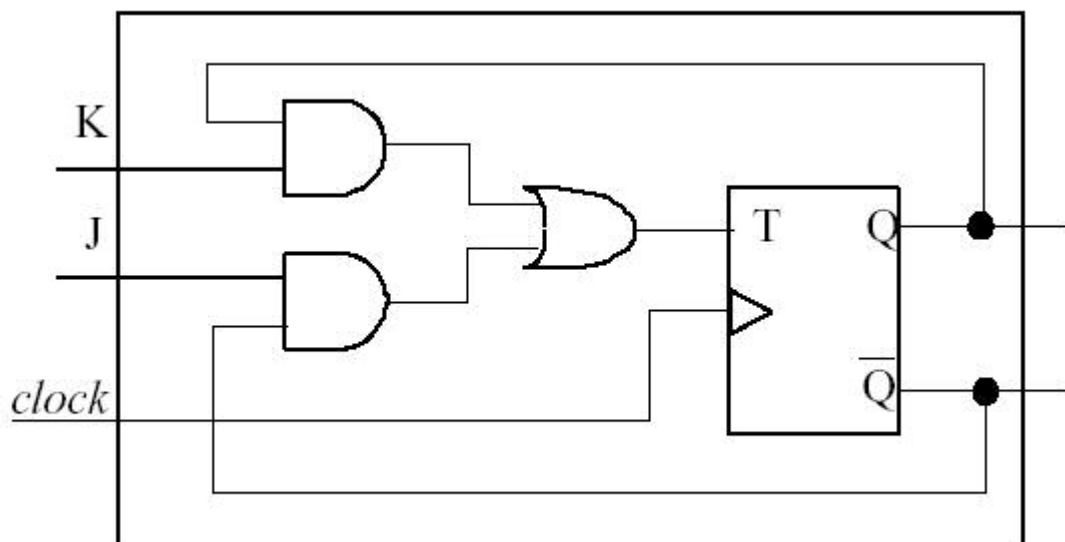
Tabella di verità e mappa di Karnaugh:

J	K	Q	Q'	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

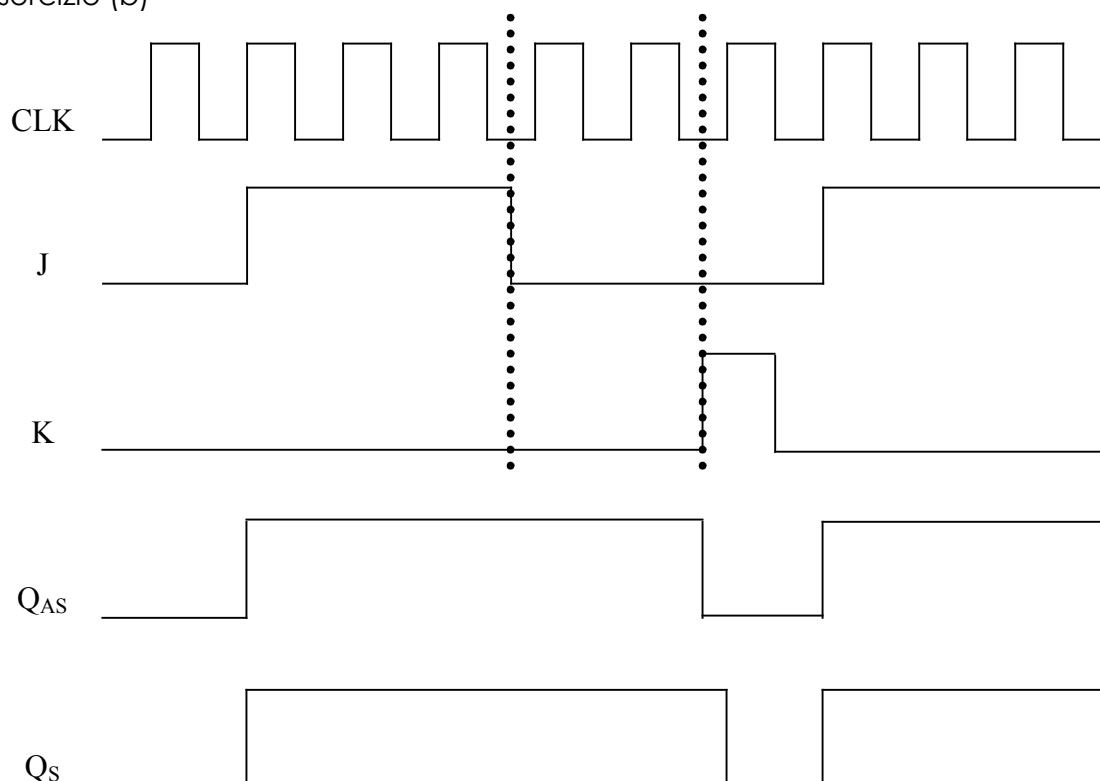


$$T = J\bar{Q} + KQ$$

Circuito finale:



Esercizio (b)



ESERCIZIO 2 (8 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificati utilizzando il codice di Hamming. Si consideri la stringa di 8 bit 11101011 (il bit meno significativo è a sinistra).

Spiegando bene ogni passo del ragionamento:

- 1) (1 punto) calcolare il minimo numero di bit di controllo necessari per la codifica della parola;
- 2) (2 punti) codificare la stringa data;
- 3) (3 punti) imporre un errore sul quinto bit della stringa di 8 bit e spiegare come l'errore viene rivelato e corretto per mezzo della codifica di Hamming.
- 4) (2 punti) indicare a quale classe di codici appartiene il codice di Hamming e spiegare in modo chiaro e sintetico qual è la sua utilità.

Soluzione.

- 1) Deve venire rispettata la condizione:

$$2^K \geq N + K + 1 \quad (1),$$

dove K è il numero di bit di controllo inseriti. Essendo N=8, il numero minimo di bit di controllo richiesto è 4.

- 2) Nella codifica di Hamming, la sequenza in ingresso presenta la seguente struttura:

c ₀	c ₁	b ₀	c ₂	b ₁	b ₂	b ₃	c ₃	b ₄	b ₅	b ₆	b ₇
		1		1	1	0		1	0	1	1

Dove c₀...c₃ sono i quattro bit costituenti il vettore di controllo, e b₀...b₇ gli otto bit trasmessi. Tali bit si ottengono con le seguenti operazioni

$$\begin{aligned}
c_0 &= b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0 \\
c_1 &= b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1 \\
c_2 &= b_1 \oplus b_2 \oplus b_3 \oplus b_7 = 1 \oplus 1 \oplus 0 \oplus 1 = 1 \\
c_3 &= b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1
\end{aligned}$$

La stringa codificata è 011111011011.

- 3) Nell'ipotesi di un errore sul quinto bit della stringa iniziale, la stringa ricevuta risulta: 011111010011. Per rivelare questo errore, bisogna ricalcolare i bit di controllo:

$$\begin{aligned}
c'_0 &= b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1 \\
c'_1 &= b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1 \\
c'_2 &= b_1 \oplus b_2 \oplus b_3 \oplus b_7 = 1 \oplus 1 \oplus 0 \oplus 1 = 1 \\
c'_3 &= b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \oplus 0 \oplus 1 \oplus 1 = 0
\end{aligned}$$

Il passo successivo è calcolare il vettore di errore dato dalla differenza dei vettori di controllo c e c' (ricordiamo che somma e differenza tra bit producono lo stesso risultato):

$$\begin{aligned}
e_0 &= c_0 \oplus c'_0 = 1 \\
e_1 &= c_1 \oplus c'_1 = 0 \\
e_2 &= c_2 \oplus c'_2 = 0 \\
e_3 &= c_3 \oplus c'_3 = 1
\end{aligned}$$

Poiché il vettore risultante 1001 non è nullo, vi è un errore nella stringa di 12 bit e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato nella stringa codificata è quindi il sesto (b_4), che può venire dunque corretto.

- 4) Vedi dispense del corso (cap.4, pagg.112 e segg.).

ESERCIZIO 3 (8 punti)

Si scriva il codice Assembler MIPS di una funzione che, ricevendo in ingresso l'indirizzo iniziale di un vettore di interi v , e la sua dimensione $N \geq 1$, calcoli la somma degli scarti quadratici di ogni elemento $v[i]$ dalla media m di v . In altri termini il codice MIPS deve implementare la seguente formula:

$$S = \sum_{i=1}^N (v_i - m)^2$$

$$\text{con } m = \frac{1}{N} \sum_{i=1}^N v_i.$$

Allocazione dei parametri di ingresso e uscita: $\&v[0] \rightarrow \$4$, $N \rightarrow \$5$, $S \rightarrow \$6$.

Si ipotizzi di disporre di una funzione "media" per il calcolo di m i cui parametri di ingresso ed uscita siano i seguenti $\&v[0] \rightarrow \$4$, $N \rightarrow \$5$, $m \rightarrow \$6$.

Es. il codice MIPS potrebbe implementare il seguente codice C:

```
int scarti(int v[], int N)
{
    int i, s, m;

    m=media(v,N);
    s=0; i=0;
    do
    {
        s += (v[i]-m)*(v[i]-m);
        i++;
    }
    while(i<N);
    return s;
}
```

Soluzione.

$\$8 \leftarrow i$; $\$9 \leftarrow v[i]$, $v[i]-m$, $(v[i]-m)^2$
 $\$10 \leftarrow$ copia di $\&v[0]$, perché $\$4$ potrebbe essere alterato da "media"
 $\$11 \leftarrow m$, perché al termine $\$6$ deve contenere s

```
scarti:      addi $29, $29, -20      #salvataggio contesto
            sw $8, 0($29)
            sw $9, 4($29)
            sw $10, 8($29)
            sw $11, 12($29)
            sw $31, 16($29)         #è necessario salvare anche $31
            move $10, $4
            jal media
            move $11, $6
            move $6, $0
            move $8, $0             #i=0
do:          lw $9, 0($10)           #carico v[i]
            sub $9, $9, $11         #v[i]-m
            mul $9, $9, $9          #s=s+(v[i]-m)2
            add $6, $6, $9
            addi $8, $8, 1
            addi $10, $10, 4
            bne $8, $5, do          #while(i<N)
exit:        lw $8, 0($29)           #ripristino contesto
            lw $9, 4($29)
            lw $10, 8($29)
            lw $11, 12($29)
            lw $31, 16($29)
            addi $29, $29, 20
            jr $31                  #ritorno a chiamante
```

ESERCIZIO 4 (8 punti)

(a) (3 punti) Un disco presenta le seguenti caratteristiche: 7200 giri/min, 100 settori per traccia, tempo di posizionamento medio 4 ms, 32 B per settore.

Calcolare il tempo medio di lettura di un blocco di 1 KB da disco, nell'ipotesi che i settori del blocco siano situati ciascuno in tracce diverse.

(b) (5 punti) Un modulo DMA trasferisce blocchi della dimensione di 1 byte da una periferica alla memoria alla velocità di 14000 bps, usando la tecnica del "furto di ciclo". Si consideri la suddivisione di una istruzione in 6 cicli (si supponga che la durata di un ciclo di clock del bus e del processore coincidano). Se il microprocessore esegue istruzioni alla velocità di 1 MIPS, di quanto viene rallentato il microprocessore dai trasferimenti del DMA?

Soluzione.

Esercizio (a)

$TROT = 60 / 7200 = 0.0083$ secondi

$TLAT = TROT / 2 = 0.00415$ secondi (tempo di latenza)

$Tlett = TROT / 100 = 0.0833$ ms (tempo di lettura di un settore)

$Tpos = 4$ ms.

Numero di settori richiesti per il blocco da 1 KB: $1024B/32B = 32$.

Tempo di lettura del blocco da 1KB:

$$= 32 * (TLAT + TPOS + Tlett) = 32 * (4.15 + 4.00 + 0.0833) = 263.47 \text{ msec}$$

Esercizio (b)

Con la tecnica di trasferimento DMA con "furto di ciclo" i dati vengono trasferiti dalla periferica alla memoria intercalando i trasferimenti DMA con il normale flusso di trasferimento delle istruzioni dalla memoria al processore. E' possibile intercalare i trasferimenti DMA solo prima che il processore abbia bisogno del bus. Il DMA trasferisce dati con una frequenza pari a $14000\text{bps}/8\text{bit} = 1750$ trasf./s. Pertanto in 1 s deve effettuare 1750 'furti di ciclo'. Dal momento che ciascuna istruzione occupa sei cicli, 1750 trasferimenti equivalgono a $1750/6 = 292$ istruzioni in meno (circa). Pertanto la velocità di esecuzione delle istruzioni scende a 999708 istr./s, cioè si ha una riduzione dello 0,03%.