

SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI  
**CALCOLATORI ELETTRONICI**  
 NUOVO E VECCHIO ORDINAMENTO DIDATTICO  
 28 Giugno 2001

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

**ESERCIZIO 1 (NO: 8 punti – VO: 7 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. Esporre con la massima chiarezza il ragionamento seguito.

**Soluzione:**

Innanzitutto è necessario scrivere le tabelle di eccitazione del FF-JK e del FF-T:

*Tabella di eccitazione  
di un flip-flop JK*

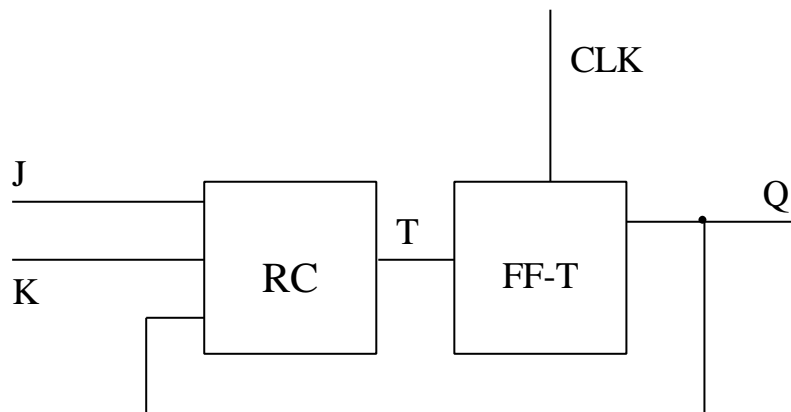
| $Q(t)$ | $Q(t+1)$ | $J$ | $K$ |
|--------|----------|-----|-----|
| 0      | 0        | 0   | d   |
| 0      | 1        | 1   | d   |
| 1      | 0        | d   | 1   |
| 1      | 1        | d   | 0   |

*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   |

Per realizzare il FF-JK a partire dal FF-T, dobbiamo far sì che una rete logica “converta” i segnali J e K nel segnale T appropriato. Per questo scopo retroazioniamo l’uscita Q utilizzando come terzo ingresso della rete logica suddetta (rete combinatoria RC).

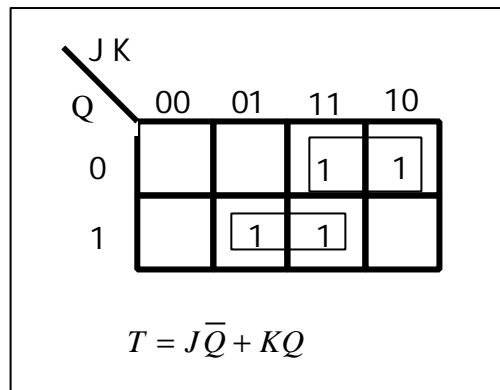
Lo schema logico è il seguente:



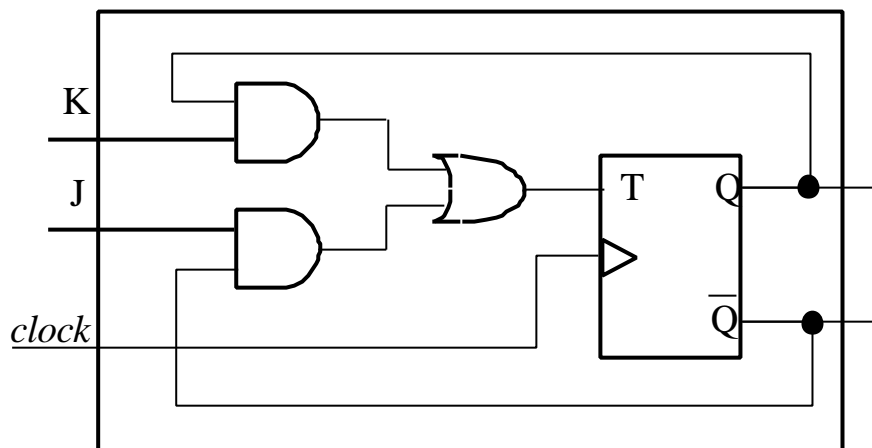
Ora scriviamo la tabella di verità di RC (l’apice indica lo stato futuro):

| 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 |

Minimizziamo la relativa funzione con le mappe di Karnaugh:



A questo punto possiamo disegnare il circuito:



## ESERCIZIO 2 (NO: 8 punti – VO: 7 punti)

Sia dato il seguente formato: rappresentazione in virgola mobile a 20 bit con mantissa frazionaria e normalizzata in segno e valore (1.M) ed esponente in eccesso 127.

- (2 punti) Calcolare il minimo e il massimo valore rappresentabile (escluso lo zero).
- (2 punti) Rappresentare i valori  $(140.75)_{10}$  e  $(10.25)_{10}$ .
- (NO: 4 punti – VO: 3 punti) Sommare i due numeri al punto precedente con l'algoritmo usato nei calcolatori (senza dimenticare il bit implicito!).

### Soluzione.

- Minimo valore:  $2^{-127}$ .  
Massimo valore:  $(2-2^{-11}) * 2^{128}$ .
- $(140.75)_{10} = 10001100.11 = 1.000110011 * 2^7$   
 $(10.25)_{10} = 1010.01 = 1.01001 * 2^3$

I due numeri si rappresentano nel modo seguente:

| Segno | Esponente | Mantissa    |
|-------|-----------|-------------|
| 0     | 10000110  | 00011001100 |
| 0     | 10000010  | 01001000000 |

- L'algoritmo dei calcolatori per la somma di due numeri in virgola mobile è fatto di quattro fasi:

- confronto degli esponenti;
- denormalizzazione del numero ad esponente minore in modo da rappresentarlo con lo stesso esponente del maggiore;
- somma delle mantisse;
- eventuale normalizzazione del risultato.

Poiché il primo ha esponente maggiore del secondo ( $7 > 3$ ), di questo facciamo scorrere la mantissa a destra di quattro posizioni.

I due numeri da sommare sono dunque:

$$\begin{array}{r}
 1.000110011 + \\
 0.000101001 = \\
 \hline
 1.001011100 \quad (*2^7)
 \end{array}$$

| Segno | Esponente | Mantissa    |
|-------|-----------|-------------|
| 0     | 10000110  | 00101110000 |

Non è stato necessario normalizzare il risultato. Convertendo questo valore in decimale, si ottiene 151, che è appunto la somma di 140.75 con 10.25.

### ESERCIZIO 3 (NO: 8 punti – VO: 7 punti)

Si consideri un calcolatore che dispone di una memoria principale di 64 Kbyte e di una memoria cache di 8 Kbyte. E' possibile accedere al singolo byte e la memoria è suddivisa in blocchi da 16 byte.

- Spiegare come vengono interpretati gli indirizzi di memoria primaria per recuperare l'informazione contenuta nella cache nel caso venga usata la modalità di indirizzamento:
  - Diretto (NO: 2 punti – VO: 1 punto);
  - associativo su insiemi, e ciascun insieme contenga quattro blocchi (NO: 2 punti – VO: 1 punto).
- A che cosa corrispondono gli indirizzi CA72, B66E, A010, 2A7F, 3FA1, 801A espressi in esadecimale, usando l'uno e l'altro metodo (NO: 4 punti – VO: 3 punti) ?
- (solo VO: 2 punti) Spiegare in modo chiaro e sintetico le principali politiche di rimpiazzamento dei blocchi usate per la memoria cache.

#### Soluzione.

a)

Metodo Diretto: <tag 3 bit> <cache ind. 9 bit> <offset 4 bit>

Metodo Associativo su insiemi: < tag 5 bit> <cache ind. 7 bit> <offset 4 bit>

b)

Metodo Diretto:

CA72 = 110 | 0 1010 0111 | 0010 -> blocco 167, byte 2  
 B66E = 101 | 1 0110 0110 | 1110 -> blocco 358, byte 14  
 A010 = 101 | 0 0000 0001 | 0000 -> blocco 1, byte 0  
 2A7F = 001 | 0 1010 0111 | 1111 -> blocco 167, byte 15  
 3FA1 = 001 | 1 1111 1010 | 0001 -> blocco 506, byte 1  
 801A = 100 | 0 0000 0001 | 1010 -> blocco 1, byte 10

Metodo Associativo su Insiemi:

CA72 = 1100 1 | 010 0111 | 0010 -> insieme 39  
 B66E = 1011 0 | 110 0110 | 1110 -> insieme 102

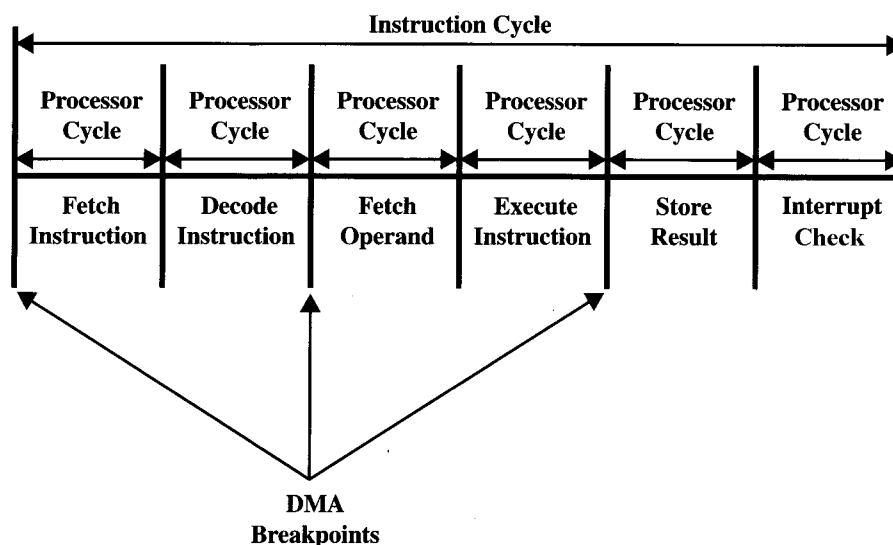
A010 = 1010 0 | 000 0001 | 0000 -> insieme 1  
 2A7F = 0010 1 | 010 0111 | 1111 -> insieme 39  
 3FA1 = 0011 1 | 111 1010 | 0001 -> insieme 122  
 801A = 1000 0 | 000 0001 | 1010 -> insieme 1.

c) Si rimanda alle dispense del corso.

#### ESERCIZIO 4 (NO: 9 punti – VO: 7 punti)

Un modulo DMA trasferisce blocchi della dimensione di 1 byte da una periferica alla memoria usando la tecnica del "furto di ciclo". La durata di un ciclo di clock del bus e del processore coincidono.

Si consideri la suddivisione in fasi mostrata in figura, con gli istanti in cui è possibile fare un "furto di ciclo" (*DMA breakpoints*):



Sapendo che il microprocessore esegue istruzioni alla velocità di 1 MIPS:

- (NO: 4 punti – VO: 3 punti) calcolare il massimo numero di furti di ciclo al secondo;
- (NO: 5 punti – VO: 4 punti) se la velocità di trasferimento *effettiva* è di 28800 bps, calcolare di quanto viene rallentato il microprocessore dai trasferimenti DMA.

#### Soluzione.

- Il massimo numero è calcolabile ipotizzando che i tre istanti in cui è possibile fare un furto di ciclo vengano usati sistematicamente. Quindi il numero di cicli per eseguire un istruzione passa *da 6 a 9*, causando un decremento dell'istruzione rate pari a 1/3 MIPS (circa 33%).

Per il numero cicli "rubati" basta considerare il numero iniziale di cicli/istruzione richiesto e moltiplicarlo per il decremento dell'istruzione rate (si ricordi che ogni furto corrisponde a un ciclo):

$$\text{Numero di furti/secondo} = 6 \text{ (cicli/istr)} * 1/3 \text{ (Mistr/sec)} = 2000000 \text{ cicli/secondo.}$$

- Il ragionamento è inverso al precedente. Qui viene fornito il numero di furti al secondo, facilmente calcolabile considerando che:

$$28800 \text{ bps} = 3600 \text{ byte/secondo, ovvero } 3600 \text{ furti di ciclo al secondo.}$$

Poiché ci sono 6 cicli/istr, il numero istruzioni al secondo "sottratto" al processore è:

$$3600/6 = 600 \text{ istr/secondo.}$$

L'istruzione rate scende pertanto a 999400 istr/sec, con un decremento dello 0.06%.

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

- 1) Descrivere la classificazione di Flynn delle architetture parallele, spiegando che tipo di calcolatore ricade in ciascuna architettura. (3 punti)
- 2) Illustrare le misure di "speedup" e "efficienza" usate per valutare le prestazioni di una architettura multiprocessore. (2 punti)

**Soluzione**

Vedi dispense del corso.