

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO**
18 Febbraio 2004

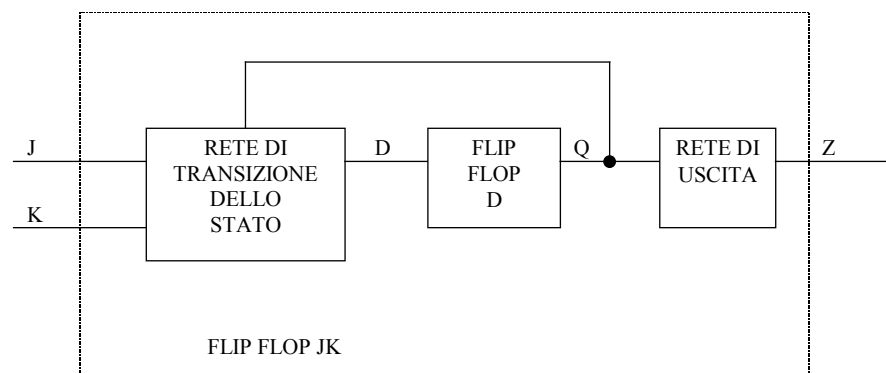
MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI

ESERCIZIO 1 (8 punti)

- (a) (NO: 5 punti – VO: 3 punti) Progettare un flip flop JK a partire da un flip flop D. Disegnare il circuito finale indicando chiaramente la funzione di transizione dello stato, il numero di stati, la funzione di uscita.
- (b) (NO: 3 punti - VO: 2 punti) Utilizzando flip flop JK, implementare un contatore binario che realizzi il conteggio da 0 a 15.
- (c) (solo VO: 3 punti) Spiegare in modo chiaro e sintetico la differenza tra una rete sequenziale sincrona e una asincrona.

Soluzione.

- (a)
Come in ogni rete logica sequenziale, lo schema del dispositivo è il seguente:

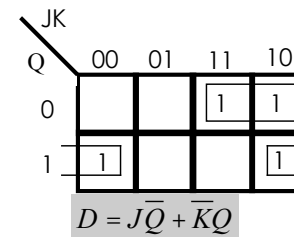


L'uscita di un flip flop JK corrisponde ai valori dello stato (due), per cui si ha $Z=Q$.

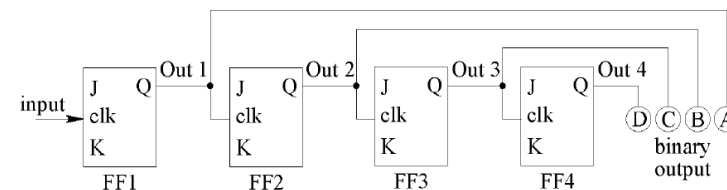
Rimane da definire solo la funzione di transizione dello stato, che si ottiene dalla tabella di transizione di un flip flop JK e dalla tabella di eccitazione del flip flop D:

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

Sintetizzando con le mappe di Karnaugh:



- (b)
Per implementare un contatore modulo 16 è sufficiente connettere in serie i vari flip flop JK:



- (c)
Vedi dispense del corso.

ESERCIZIO 2 (solo NO: 9 punti)

- a) (6 punti) Scrivere una funzione Assembler MIPS che ricevendo in ingresso due valori interi positivi x e y (con $x > y$), produca in uscita il valore z pari al Massimo Comune Divisore di x e y .
- Si supponga che $\$4 \leftarrow x$, $\$5 \leftarrow y$, $\$6 \leftarrow z$. Poiché il MIPS non è dotato di una esplicita istruzione di divisione, si ipotizzi di usare una funzione $\text{mod}(a, b)$ che memorizza il resto della divisione di a e b (che devono trovarsi rispettivamente in $\$4$ e $\$5$) nel registro $\$6$. Ad esempio, il codice MIPS potrebbe implementare la seguente funzione C:

```
int MCD(int x, int y)
{
    int r ;

    do
    {
        r=mod(x,y) ;
        x=y ;
        y=r ;
    }
    while (r>0);

    return x;
}
```

- b) (3 punti) Elencare i passi fondamentali del ciclo di esecuzione delle istruzioni in un calcolatore elettronico.

Soluzione.

a)

```

mcd:
    jal mod          #r = mod(x,y) con r → $6
    move $4, $5      #x = y;
    move $5, $6      #y = r;
    bne $6, $0, mcd  #if(r>0) goto mcd
    move $6, $4      #MCD in $6
    jr $31           #ritorno al chiamante

```

b) Vedi lucidi del corso.

ESERCIZIO 2 (solo VO: 8 punti)

Un calcolatore ha un sistema di memoria virtuale a tre livelli costituita da: cache, memoria primaria e disco. La lettura di una parola che si trova già memorizzata nella cache richiede 15 ns. La lettura di una parola dalla memoria primaria e il suo trasferimento in cache richiedono complessivamente 40 ns. La lettura di una parola dal disco e il suo trasferimento in memoria primaria richiedono complessivamente 10 ms. La probabilità che una parola si trovi già in cache è pari a 0.95 mentre la probabilità che una parola si trovi in memoria primaria quando non è presente nella cache è pari a 0.6. Calcolare il tempo medio di accesso al sistema di memoria.

Soluzione:

Si devono considerare tre possibili situazioni:

Posizione della parola richiesta	Probabilità	Tempo di accesso totale
In cache	0.95	15 ns
Non in cache ma in memoria primaria	$(1 - 0.95) * 0.6 = (0.05) * 0.6 = 0.03$	$40\text{ns} + 15\text{ ns} = 55\text{ ns}$
Non in cache né in memoria primaria	$(1 - 0.95) * (1 - 0.6) = 0.05 * 0.4 = 0.02$	$10\text{ ms} + 40\text{ ns} + 15\text{ ns} = 10.000.055\text{ ns}$

Pertanto il tempo medio di accesso alla gerarchia di memoria è dato da:

$$T_m = 0.95 * 15 + 0.03 * 55 + 0.02 * 10.000.055 = 200.017\text{ ns}$$

ESERCIZIO 3 (NO: 8 punti – VO: 6 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa di 12 bit 101001101110 (il bit meno significativo è a sinistra), risultato della codifica di una parola di N bit secondo il codice di Hamming.

- 1) (VO: 1 punto – NO: 2 punti) calcolare N, supponendo di aver fatto uso del numero minimo di bit di controllo necessari.
- 2) (VO: 2 – NO: 3 punti) scrivere la parola di N bit a partire dalla stringa data;
- 3) (3 punti) indicare eventuali errori nella stringa codificata, specificando quale dei bit è stato alterato.

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 $N + K = 12$, si evince dalla (1) che il numero minimo di bit di controllo richiesto è 4. Da cui $N = 8$.

- 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	0	1	0	0	1	1	0	1	1	1	0

Dove $c_0...c_3$ sono i quattro bit costituenti il vettore di controllo, e $b_0...b_7$ gli otto bit trasmessi. La sequenza ricevuta è 10111110.

- 3) Per verificare la presenza di un errore, dobbiamo ricalcolare il vettore di controllo a partire dalla sequenza ricevuta. Si ha:

$$c'_0 = b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 = 0$$

$$c'_1 = b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 = 1$$

$$c'_2 = b_1 \oplus b_2 \oplus b_3 \oplus b_7 = 0$$

$$c'_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 1$$

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):

$$e_0 = c_0 \oplus c'_0 = 1$$

$$e_1 = c_1 \oplus c'_1 = 1$$

$$e_2 = c_2 \oplus c'_2 = 0$$

$$e_3 = c_3 \oplus c'_3 = 1$$

Poiché il vettore risultante 1011 non è nullo, vi è un errore nella stringa di 12 bit data e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato è quindi l'undicesimo (b_6), e la parola corretta è 10111100.

ESERCIZIO 4 (NO: 8 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:

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 faults' nei seguenti due casi:

- 1) (NO: 2 punti – VO: 2 punti) strategia di rimpiazzamento delle pagine FIFO.
- 2) (NO: 3 punti – VO: 2 punti) strategia di rimpiazzamento delle pagine LRU.
- 3) (NO: 3 punti – VO: 2 punti) spiegare in modo chiaro e sintetico cos'è, a cosa serve e come è strutturata una Page Map Table (PMT).

Soluzione:

1) 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	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	4	4	4	4	4	4	5	6	7	7	7	2	4	5	5	6	7	2
		2	3	3	3	3	3	3	3	4	5	6	6	6	7	2	4	4	5	6	7
			2	2	2	2	2	2	2	3	4	5	5	5	6	7	2	2	4	5	6
										2	3	4	4	4	5	6	7	7	2	4	5
Hit				x	x	x	x	x	x					x	x				x		

Hit ratio = 9/21; Page faults = 12/21

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 = 9/21

ESERCIZIO 5 (solo VO: 5 punti)

L'esecuzione di programmi su elaboratori con architettura RISC può originare le cosiddette "bolle di ritardo". Spiegare in modo chiaro e sintetico cosa si intende con questo termine ed elencare le principali tecniche impiegate per eliminarle.

Soluzione.

Vedi dispense su architetture RISC.