

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI  
CALCOLATORI ELETTRONICI  
NUOVO E VECCHIO ORDINAMENTO DIDATTICO**  
13 Gennaio 2005

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

**ESERCIZIO 1 (NO: 8 punti – VO: 7 punti)**

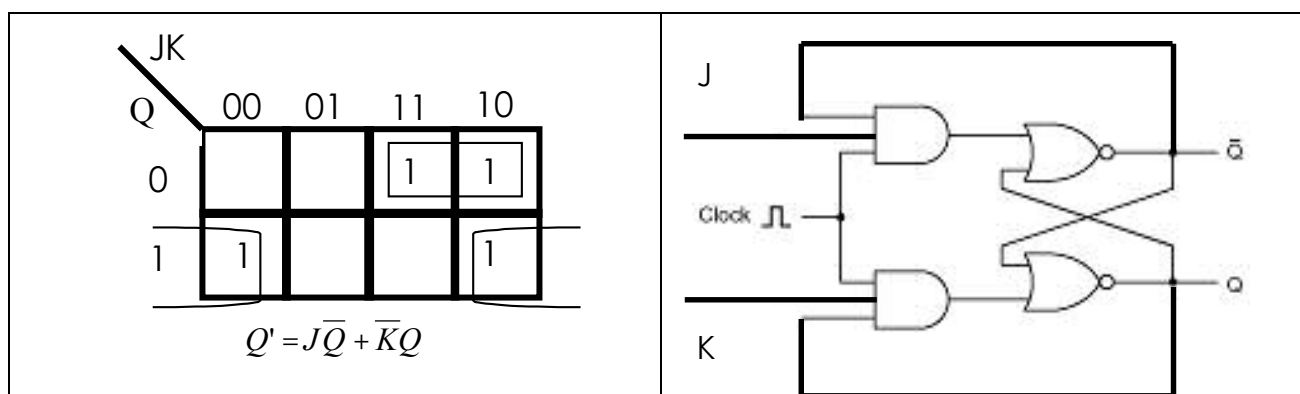
- (1) (NO: 5 punti – VO: 4 punti) Scrivere la tabella di transizione **completa**, ovvero con tutti i valori dello stato successivo dati quelli degli ingressi e dello stato presente, di un flip flop JK, e implementarlo con il numero minimo di porte logiche. Disegnare il circuito del flip flop nella configurazione "sincrona".
- (2) (3 punti) Ottenere un flip flop T da un flip flop JK.

**Soluzione.**

(1) Indicando con Q e Q' lo stato attuale e quello successivo, la tabella di transizione del flip flop JK è la seguente:

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

Per l'implementazione col numero minimo di porte logiche ci serviamo del metodo delle mappe di Karnaugh. A fianco abbiamo il disegno del relativo circuito nella configurazione "sincrona", realizzata con porte AND e NOR.



- (2) Per ottenere un flip flop T è sufficiente connettere fra loro gli ingressi J e K.

**ESERCIZIO 2 (NO: 9 punti – VO: 8 punti)**

Si consideri una memoria primaria costituita da 128 parole e una memoria cache costituita da 32 parole. Il metodo di indirizzamento della cache sia quello completamente associativo con blocchi di 8 parole.

- (1) (1 punto) Spiegare, precisando il significato e la funzione dei diversi campi, come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache.
- (2) (NO: 5 punti – VO: 4 punti) Si considerino le seguenti chiamate ad altrettante parole (indirizzi espressi in decimale): 125, 58, 60, 113, 70, 27, 59, 111, 5, 92. Si indichi il contenuto della cache, ovvero quali byte occupano i relativi blocchi di cache, dopo l'ultima chiamata, considerando la **LRU** come strategia di rimpiazzamento dei blocchi.
- (3) (2 punti) Si consideri una gerarchia di memoria a tre livelli costituita da: cache, primaria e disco. Se l'hit ratio di cache è pari a 0.9, l'hit ratio di primaria è pari a 0.95, i tempi di accesso a cache, primaria e disco valgono rispettivamente, 5 nsec, 50 nsec e 5 msec, esprimere il tempo medio di accesso alla gerarchia **in nanosecondi**.

**Soluzione.**

(a)

Con 128 parole, ogni indirizzo è formato da 7 bit così suddivisi secondo il metodo completamente associativo:

< Block frame 4 bit > < Offset 3 bit >

(b)

Blocco 0	Blocco 1	Blocco 2	Blocco 3
88 – 95	56 – 63	104 – 111	0 – 7

(c) Con tutti i dati a nostra disposizione è sufficiente valutare la formula:

$$\bar{T} = H_C T_C + (H_P - H_C)(T_P + T_C) + (1 - H_P)(T_D + T_P + T_C)$$

Quindi:

$$\bar{T} = 0.9 \cdot 5 + 0.05 \cdot 55 + 0.05 \cdot 5000000 = 250007.25ns$$

**ESERCIZIO 3 (solo NO: 8 punti)**

Si consideri la seguente area di memoria:

Indirizzo	Contenuto
1000	add \$4, \$5, \$6
1004	
1008	sub \$4, \$5, \$6
1012	

Si implementi in Assembler MIPS una funzione `esegui(k)` tale che per  $k=0$  esegua l'istruzione contenuta nell'indirizzo 1000 e per  $k=1$  esegua l'istruzione all'indirizzo 1008, ritornando infine al chiamante. Per tutti gli altri valori di  $k$ , la funzione ritorna al chiamante.

Si modifichi il contenuto delle locazioni 1004 e 1012 in modo da completare l'implementazione di `esegui(k)`.

Si supponga che  $k$  sia contenuta nel registro `$7`.

**Soluzione.**

```
esegui:
    addi $29, $29, -8
    sw $8, 0($29)
    sw $1, 4($29)
    addi $1, $0, 1
    beq $7, $0, esegui_0:
    beq $7, $1, esegui_1:
exit:
    lw $8, 0($29)
    lw $1, 4($29)
    addi $29, $29, 8
    jr $31

esegui_0:
    add $8, $0, 1000
    jr $8

esegui_1:
    add $8, $0, 1008
    jr $8
```

Le locazioni 1004 e 1012 devono contenere l'istruzione `j exit`.

### ESERCIZIO 3 (solo VO: 7 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa **011110100101** (il bit meno significativo è a sinistra), ottenuta dalla codifica di Hamming di una stringa di N bit. **Spiegando bene ogni passo del ragionamento:**

- 1) (2 punti) calcolare N, sapendo che è stato usato il minimo numero di bit di controllo necessari per la codifica della parola di N bit;
- 2) (5 punti) decodificare la stringa data, rilevando eventuali errori.

#### 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  $N+K=12$ , il numero minimo di bit di controllo richiesto è 4.

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

$c_0$	$c_1$	$b_0$	$c_2$	$b_1$	$b_2$	$b_3$	$c_3$	$b_4$	$b_5$	$b_6$	$b_7$
0	1	1	1	1	0	1	0	0	1	0	1

Dove  $c_0 \dots c_3$  sono i quattro bit costituenti il vettore di controllo, e  $b_0 \dots b_7$  gli otto bit trasmessi. Tali bit si ottengono con le seguenti operazioni

$$c_0 = b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 = 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$c_1 = b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$c_2 = b_1 \oplus b_2 \oplus b_3 \oplus b_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$c_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

I bit di controllo effettivamente ricevuti sono invece, nell'ordine 0, 1, 1, 0. Calcolando il vettore di errore:

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

$$e_1 = c_1 \oplus c'_1 = 0$$

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

$$e_3 = c_3 \oplus c'_3 = 0$$

Poiché il vettore risultante 0001 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 primo, che essendo però un bit di controllo, non interferisce nel contenuto informativo della stringa ricevuta. La stringa decodificata è dunque: 11010101.

**ESERCIZIO 4** (NO: 8 punti – VO: 6 punti)

Si consideri un disco rigido con le seguenti caratteristiche: velocità di rotazione = 5400 giri/min, tempo medio di posizionamento = 5 ms, 34 settori per traccia di 512 byte ciascuno.

- a) Calcolare il tempo medio di trasferimento di un file da 8 kbyte considerando:
- il caso migliore (NO: 4 punti – VO: 2 punti);
  - il caso peggiore (NO: 4 punti – VO: 2 punti).
- b) Supporre che il file debba essere trasferito in memoria principale, attraverso un modulo DMA. Spiegare chiaramente quale è la strategia di trasferimento più appropriata per massimizzare la velocità di trasferimento. (solo VO: 2 punti)

**Soluzione**

$T_{rot} = 60/5400 \text{ sec} = 11.11 \text{ msec.}$

$T_{lat} = T_{rot}/2 = 5.555 \text{ msec.}$

$T_{ls} = T_{rot} / 34 = 0.327 \text{ msec.}$  (tempo di lettura di un settore)

$T_{pos} = 5 \text{ msec.}$

Numero di settori richiesti dal file  $N = 8\text{kbyte} / 512\text{byte} = 16$  settori.

- a) Caso migliore: il file è memorizzato in settori consecutivi della stessa traccia e la testina è posizionata all'inizio del primo settore. Dato che il file può essere memorizzato in una sola traccia:

$$T_{lett} = N * T_{ls} = 5.232 \text{ msec.}$$

Caso peggiore: il file è memorizzato in settori collocati in tracce diverse e la testina si trova in un punto qualsiasi del disco.

$$T_{lett} = N * (T_{lat} + T_{pos} + T_{ls}) = 16 * 10.882 = 174.112 \text{ msec}$$

- b) La strategia utilizzata è senz'altro la DMA "block transfer", per evitare tempi d'attesa sul disco. In questo caso infatti la periferica diventa master del bus e il trasferimento interrompe le operazioni della CPU fino a che non viene completato.

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

Descrivere la classificazione di Flynn delle architetture parallele, spiegando che tipo di calcolatore ricade in ciascuna architettura.

Illustrare le misure di "speedup" e "efficienza" usate per valutare le prestazioni di una architettura multiprocessore.

**Soluzione**

Vedi dispense del corso su "Architetture Parallele"