

SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
VECCHIO ORDINAMENTO
 6 Luglio 2000

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

ESERCIZIO 1 (8 punti)

Sia data una rete sequenziale caratterizzata da: due variabili di stato, **A** e **B**; un ingresso, **x**; un'uscita, **z**. Il circuito logico che implementa la rete è stato realizzato impiegando due flip-flop di tipo **J-K**, uno per ciascuna variabile di stato. Le funzioni logiche per gli ingressi dei due flip-flop e per l'uscita sono le seguenti:

flip-flop **A**: $J_A = \bar{x}B$ $K_A = \bar{x}\bar{B}$
 flip-flop **B**: $J_B = \bar{A}$ $K_B = \bar{x}\bar{A} + xA$
 uscita **z**: $z = xA$

1. Ricavare la tabella delle transizioni. (2 punti)
2. Ricavare, usando le mappe di Karnaugh, le espressioni minime che rappresentano lo stato futuro delle variabili di stato **A** e **B** in funzione dello stato attuale di **A**, **B** e dell'ingresso **x**. (4 punti)
3. Si supponga di voler realizzare lo stesso circuito con flip flop di tipo **T**. Calcolare le funzioni di eccitazione dei flip flop **T** sfruttando le espressioni di J_A , K_A , J_B , e K_B . (Suggerimento: ricavare prima l'espressione di **T** in funzione di **J**, **K** e dello stato presente). (2 pt)

Soluzione:

1. Per costruire la tabella delle transizioni occorre servirsi della tabella di eccitazione dei flip-flop e delle equazioni logiche dei flip flop **A** e **B** e dell'uscita **z**.

| Stato pres. | | Stato prossimo | | | | | | | | | | | | Uscita z | | Tabella di eccitazione di un flip-flop J-K: <table><tr><th>J</th><th>K</th><th>Q(t+1)</th></tr><tr><td>0</td><td>0</td><td>Q(t)</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>Q'(t)</td></tr></table> | J | K | Q(t+1) | 0 | 0 | Q(t) | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | Q'(t) |
|----------------|---|----------------|----------------|----------|----------------|----------------|----------|----------------|----------------|----------|----------------|----------------|----------|----------|---|--|---|---|--------|---|---|------|---|---|---|---|---|---|---|---|-------|
| | | J | K | Q(t+1) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Q(t) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Q'(t) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| x = 1 | | | | | | x = 0 | | | | | | x = 0 | x = 1 | | | | | | | | | | | | | | | | | | |
| A | B | J _A | K _A | A | J _B | K _B | B | J _A | K _A | A | J _B | K _B | B | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | | | | | | | | | | | |

2. Le equazioni logiche dello stato prossimo, $A(t+1)$ e $B(t+1)$, in funzione degli ingressi **x** e **y** e dello stato presente **A**, **B**, si ricavano facilmente dalle mappe di Karnaugh:

A B

| | | | | |
|-----|----|----|----|----|
| x \ | 00 | 01 | 11 | 10 |
| 0 | | 1 | 1 | |
| 1 | | | 1 | 1 |

$A(t+1) = \bar{x}B + xA$

A B

| | | | | |
|-----|----|----|----|----|
| x \ | 00 | 01 | 11 | 10 |
| 0 | 1 | | 1 | |
| 1 | 1 | 1 | | |

$B(t+1) = \bar{A}x + \bar{A}\bar{B} + \bar{x}AB$

3. Per sostituire un flip-flop **JK** con un flip-flop **T** è sufficiente osservare che l'eccitazione di un flip-flop **T** si ottiene dalle funzioni di eccitazione **J** e **K** e dallo stato presente $Q(t)$ come: $T = J\bar{Q}(t) + KQ(t)$ (compito del 10.08.1998). Usando questa espressione si calcola:

$$T_A = J_A\bar{A} + K_A A = \bar{x}\bar{A}B + \bar{x}A\bar{B} = \bar{x}(A \oplus B);$$

$$T_B = J_B\bar{B} + K_B B = \bar{A}\bar{B} + \bar{x}\bar{A}B + xAB$$

ESERCIZIO 2 (8 punti)

Si consideri un calcolatore che dispone di una memoria cache a due livelli, L1 e L2. La memoria cache L1 abbia dimensione pari a 32 kbyte e sia indirizzata in modalità "diretta", mentre la cache L2 abbia dimensione 512 Kbyte e sia indirizzata con il metodo "associativo su insiemi", e ciascun insieme contenga quattro blocchi. L'indirizzamento usato è a 32 bit, è possibile accedere al singolo byte e la memoria sia suddivisa in blocchi da 32 byte.

1. Spiegare come vengono interpretati gli indirizzi logici a 32 bit per leggere/scrivere i dati nelle due cache. (3 punti)
2. Calcolare il tempo medio di accesso alla gerarchia di memoria, sapendo che la cache L1 ha un tempo di accesso pari a 10ns, la cache L2 ha un tempo di accesso pari a 20 ns, e la memoria principale ha un tempo di accesso pari a 50 ns e che gli hit ratio rispettivamente della cache L1 e L2 sono pari a 0.8 e 0.9999. (3 punti)
3. Calcolare il valore che dovrebbe assumere l'hit ratio della cache L1 nel caso in cui si voglia ottenere lo stesso tempo medio di accesso calcolato al punto 2 ma usando una gerarchia a due livelli (cache L1 e memoria principale). (2 punti)

Soluzione

1. I 5 bit meno significativi vengono usati per indirizzare il singolo byte all'interno di un blocco ($32 = 2^5$). I restanti 27 bit costituiscono il <block frame>. La cache L1 di 32 kbyte è indirizzata con il metodo "diretto" e contiene esattamente $32k/32 = 1024$ blocchi. Il <block frame> viene dunque interpretato nel modo seguente: i 10 bit meno significativi identificano l'<index>, cioè il blocco in cache in cui deve essere letta/scrivita la parola indirizzata; i restanti 17 bit formano il <tag>. Per un dato indirizzo a 32 bit, si individua il blocco in cache attraverso l'<index> e si confronta il <tag> dell'indirizzo, formato dai 17 bit più significativi, con quello memorizzato nella cache. Se la parola non è presente in cache la si trasferisce dalla memoria primaria.

La cache L2 è suddivisa in insiemi di 4 blocchi ciascuno, cioè contiene $512k/(4 \cdot 32) = 4k$ (2^{12}) insiemi. L'indirizzo a 32 bit viene interpretato per la cache L2 nel modo seguente: 5 bit meno significativi per il <block offset>, i successivi 12 bit di <index> per individuare l'insieme e i restanti 15 bit per il <tag>. Una volta individuato l'insieme, si cerca la parola in uno dei quattro blocchi contenuti in ciascun insieme confrontando il <tag>, formato dai 15 bit più significativi. Se la parola non è presente la si trasferisce dalla memoria primaria e, se l'insieme è pieno, si sostituisce un blocco usando una delle politiche FIFO, LRU o LFU.

2. Il tempo medio di accesso ad una gerarchia a tre livelli, T_a , si calcola come:

$$T_a = H_{L1}T_{L1} + (H_{L2} - H_{L1})T_{L2} + (H_m - H_{L2})T_m$$

Sostituendo i dati forniti nel testo, si ottiene $T_a = 12 \text{ ns}$

3. In questo caso è necessario risolvere l'equazione

$$12\text{ns} = H_{\text{cache}} 10\text{ns} + (1 - H_{\text{cache}})50\text{ns}$$

da cui si ricava $H_{\text{cache}} = 0.95$. Questo valore può essere ottenuto, ad esempio, aumentando la dimensione della cache.

ESERCIZIO 3 (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. strategia di rimpiazzamento delle pagine FIFO.
2. strategia di rimpiazzamento delle pagine LRU.
3. Spiegare sinteticamente come avviene la gestione di un "page fault".

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

3. (Descrizione "page fault": vedi dispense del corso sui "Sistemi Operativi")

ESERCIZIO 4 (6 punti)

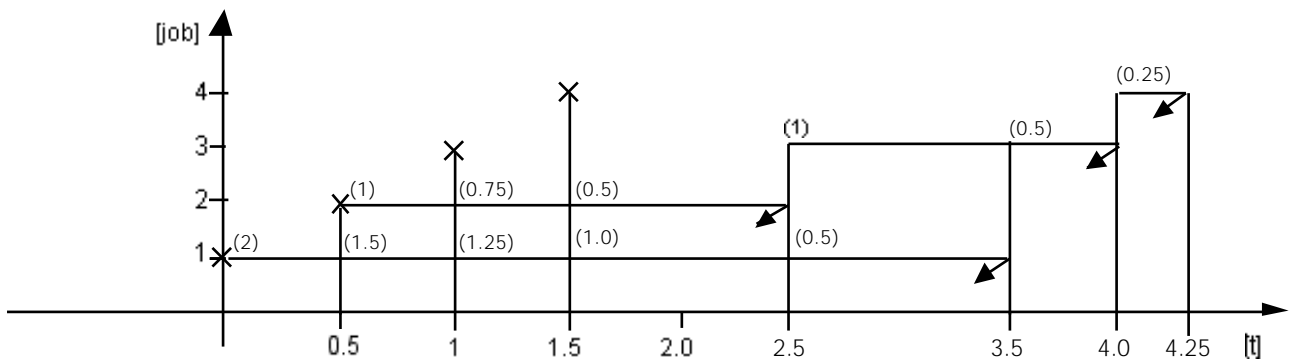
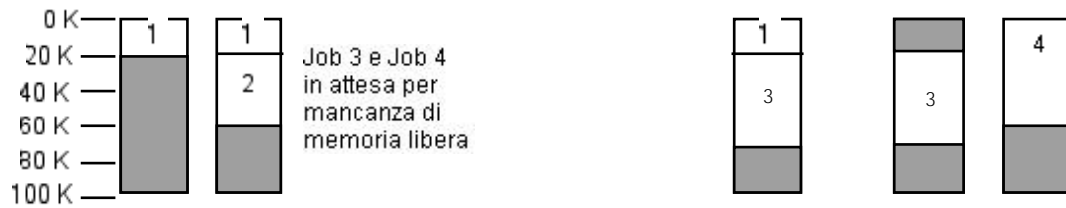
Sia data la seguente lista di processi (si supponga che l'istante iniziale sia 0):

| Job | Tempo di Arrivo | Tempo di CPU Richiesto | Richiesta di Memoria |
|-----|-----------------|------------------------|----------------------|
| 1 | 0.0 | 2.0 | 20K |
| 2 | 0.5 | 1.0 | 40K |
| 3 | 1.0 | 1.0 | 50K |
| 4 | 1.5 | 0.25 | 60K |

Il sistema ha 100K di memoria disponibile, e viene gestita con partizioni dinamiche non rilocabili.

- Mostrare, utilizzando il metodo grafico, la sequenza di esecuzione dei job qualora si impieghi la politica di scheduling FIFO multiprogrammata "round robin" (3 punti)
- Indicare, negli istanti in cui un job va in esecuzione e termina, la partizioni di memoria occupate da ciascun job e le partizioni libere, giustificando l'eventuale "messa in attesa" dei job (2 punti).
- Calcolare il tempo di *turnaround* medio e il tempo di *turnaround* pesato medio (1 punto).

Soluzione:



| Job | t_{arrivo} | t_{start} | t_{finish} | Turnaround time | Weighted Turnaround time |
|--------------|---------------------|--------------------|---------------------|-----------------|--------------------------|
| 1 | 0 | 0 | 3.5 | 3.5 | 1.75 |
| 2 | 0.5 | 0.5 | 2.5 | 2 | 2 |
| 3 | 1 | 2.5 | 4 | 3 | 3 |
| 4 | 1.5 | 4 | 4.25 | 2.75 | 11 |
| Media | | | | 2.8125 | 4.4375 |

ESERCIZIO 5 (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"