

# SOLUZIONI DELLA SECONDA PROVA INTERMEDIA DEL CORSO DI CALCOLATORI ELETTRONICI

13 Aprile 2000

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

### ESERCIZIO 1 (6 punti)

Si consideri un calcolatore che dispone di una memoria cache di 32 byte. Il metodo di indirizzamento della cache sia "associativo su insiemi", e ciascun insieme contenga due blocchi. L'indirizzamento usato è a 8 bit, è possibile accedere al singolo byte e la memoria sia suddivisa in blocchi da 4 byte.

1. Spiegare come vengono interpretati gli indirizzi logici a 8 bit per recuperare l'informazione contenuta nella cache. (2 punti)
2. Sia data la seguente sequenza di accessi in memoria primaria, in termini di indirizzi esadecimali delle parole della memoria primaria (il primo indirizzo sia 0):

31, 71, 58, 73, BE, C7, 5A, 33, 72, BD, 32, A0

Per ciascuna richiesta dire se c'è stato un cache miss o un cache hit, mostrando il contenuto della cache. (N.B.: si ipotizzi la cache inizialmente vuota). Usare la tecnica di rimpiazzamento dei blocchi nella cache LRU. (4 punti)

### Soluzione:

1. I due bit meno significativi vengono usati per indirizzare il singolo byte all'interno di un blocco. I restanti 6 bit costituiscono il "block offset". La cache di 32 byte con insiemi da due blocchi (8 byte) contiene esattamente 4 insiemi. Pertanto l'insieme nella cache viene indirizzato dai 2 bit meno significativi del "block offset". Una volta individuato l'insieme, si cerca la parola in uno dei due blocchi contenuti in ciascun insieme confrontando il "tag", formato dai 4 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. 

31 ⇒ 3 00 01 ⇒ insieme 0 (M);	71 ⇒ 7 00 01 ⇒ insieme 0 (M)
58 ⇒ 5 10 00 ⇒ insieme 2 (M);	73 ⇒ 7 00 11 ⇒ insieme 0 (H)
BE ⇒ B 11 10 ⇒ insieme 3 (M);	C7 ⇒ C 01 11 ⇒ insieme 1 (M)
5A ⇒ 5 10 10 ⇒ insieme 2 (H);	33 ⇒ 3 00 11 ⇒ insieme 0 (H)
72 ⇒ 7 00 10 ⇒ insieme 0 (H);	BD ⇒ B 11 01 ⇒ insieme 3 (H)
32 ⇒ 3 00 10 ⇒ insieme 0 (H);	A0 ⇒ A 00 00 ⇒ insieme 0 (M)

(M) Cache Miss (H) Cache Hit

Hit ratio =  $6/12 = 0,5$

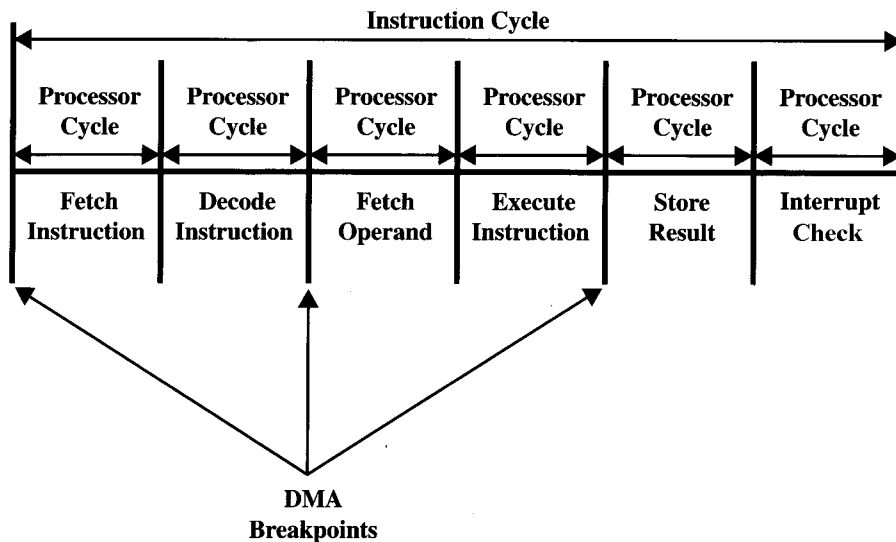
Nella sequenza di richieste data si ha un solo caso di sostituzione dei blocchi: la richiesta A0 provoca un cache miss e deve essere inserita nell'insieme 0 che contiene già due blocchi (tag 3 e 7). La sostituzione avviene secondo la politica LRU, per cui viene sostituito il blocco con tag 7.

Contenuto finale della cache

	Insieme 0	Insieme 1	Insieme 2	Insieme 3
Blocco 0	30 ÷ 33	C4 ÷ C7	58 ÷ 5B	BC ÷ BF
Blocco 1	A0 ÷ A3			

**ESERCIZIO 2 (6 punti)**

Un modulo DMA trasferisce blocchi della dimensione di 1 byte da una periferica alla memoria alla velocità di 9600 bps, usando la tecnica del "furto di ciclo". Si consideri la suddivisione di una istruzione in fasi mostrata in figura (si supponga che la durata di un ciclo di clock del bus e del processore coincidano). In figura sono mostrati anche gli istanti in cui è possibile effettuare un 'furto di ciclo'. Se il microprocessore esegue istruzioni alla velocità di 1 MIPS, di quanto viene rallentato il microprocessore dai trasferimenti del DMA? (6 punti)

**Soluzione:**

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. Dalla figura si vede che 1 istruzione viene eseguita in 6 cicli di clock, e che i trasferimenti DMA possono avvenire al massimo con una periodo di due cicli di clock. Il DMA trasferisce dati con una frequenza pari a  $9600\text{bps}/8\text{bit} = 1200\text{ trasf./s.}$  Pertanto in 1 s deve effettuare 1200 'furti di ciclo'. Dal momento che ciascuna istruzione occupa sei cicli (vedi figura), 1200 trasferimenti equivalgono a  $1200/6 = 200$  istruzioni in meno. Pertanto la velocità di esecuzione delle istruzioni scende a 999800 istr./s, cioè si ha una riduzione dello 0,02%.

**ESERCIZIO 3 (7 punti)**

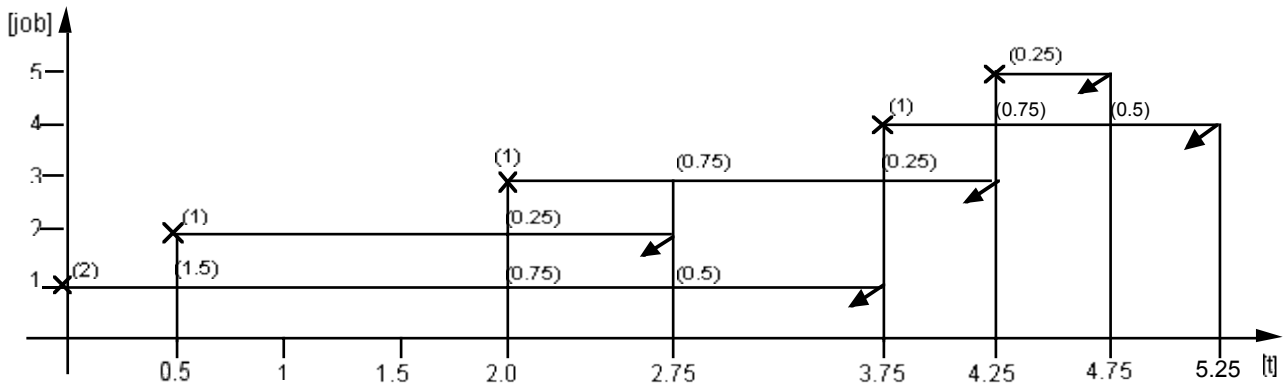
Sia data la seguente lista di processi (si supponga che l'istante iniziale sia 0):

Job	Tempo di Arrivo	Tempo di CPU Richiesto
1	0.0	2.0
2	0.5	1.0
3	2.0	1.0
4	3.75	1.0
5	4.25	0.25

1. Mostrare, utilizzando il metodo grafico, la sequenza di esecuzione dei job qualora si impieghi la politica di scheduling FIFO multiprogrammata "round robin" (5 punti)
2. Calcolare il tempo di *turnaround* medio e il tempo di *turnaround* pesato medio (2 punti).

### Soluzione:

FIFO multiprogrammata 'round robin'



Job	$t_{\text{arrivo}}$	$t_{\text{start}}$	$t_{\text{finish}}$	Turnaround time	Weighted Turnaround time
1	0	0	3.75	3.75	1.875
2	0.5	0.5	2.75	2.25	2.25
3	2	2	4.25	2.25	2.25
4	3.75	3.75	5.25	1.5	1.5
5	4.25	4.25	4.75	0.5	2
Media				2.05	1.975

### ESERCIZIO 4 (7 punti)

Si consideri la seguente 'page map table' relativa a un processo in esecuzione. Tutti i numeri sono espressi nel formato decimale e il primo indirizzo è 0. Tutti gli indirizzi sono relativi a un singolo byte nella memoria principale. La dimensione della pagina è 1024 byte.

Indirizzo di pagina virtuale	Bit di validità	Bit di modifica	Indirizzo di pagina fisica
0	1	0	4
1	1	1	7
2	0	0	10
3	1	0	2
4	0	0	5
5	1	1	0

- Descrivere come viene utilizzato ciascuna colonna della tabella per tradurre un indirizzo virtuale generato dalla CPU in un indirizzo fisico (4 punti).
- Quali indirizzi fisici vengono generati dagli indirizzi virtuali: 1052, 2221, 5499? (3 punti)

### Soluzione

- Un indirizzo virtuale, espresso in decimale, deve essere diviso per 1024 e il quoziente indica il numero di pagina virtuale (se esprimiamo l'indirizzo in formato binario, questa operazione equivale a considerare i bit più significativi dell'indirizzo, a partire dall' 11°, dal momento che i 10 bit meno significativi indicano uno dei 1024 byte che formano una pagina). Nella tabella si controlla che la riga relativa alla pagina virtuale richiesta contenga un numero di pagina fisico valido, cioè che la pagina sia caricata in memoria. Questa informazione si trova nella colonna relativa al bit di validità: se è uguale a 1, la pagina è in memoria e l'indirizzo di pagina fisico si legge nell'ultima colonna; altrimenti la pagina non è in memoria e

viene generato un page fault. La terza colonna indica se la pagina è stata modificata durante la permanenza nella memoria: in caso di un page fault la pagina che viene rimossa deve essere riscritta sul disco solo se è stata modificata.

2. Dalla risposta precedente si ricava:

$1052/1024 \Rightarrow$  indirizzo pagina virtuale 1, la pagina è presente in memoria all'indirizzo di pagina virtuale 7, e il byte cercato all'interno della pagina è il  $1052 - 1024 = 28^\circ$ . Il corrispondente indirizzo fisico è  $7 \cdot 1024 + (1052 - 1024) = 7196$ .

$2221/1024 \Rightarrow$  indirizzo pagina virtuale 2, la pagina non è presente in memoria (bit di validità uguale a 0) e viene generato un page fault.

$5499/1024 \Rightarrow$  indirizzo pagina virtuale 5, la pagina è presente in memoria all'indirizzo di pagina virtuale 0, e il byte cercato all'interno della pagina è il  $5499 - 5 \cdot 1024 = 379^\circ$ . Il corrispondente indirizzo fisico è  $0 \cdot 1024 + (5499 - 5120) = 379$ .

---

### SOLO PER GLI STUDENTI DEL VECCHIO ORDINAMENTO DIDATTICO

#### ESERCIZIO 5 (7 punti)

1. Spiegare che cosa siano le "bolle di ritardo" che si possono generare durante l'esecuzione delle istruzioni in pipeline e quali tecniche vengano impiegate per eliminarle. (4 punti)
2. Descrivere sinteticamente le differenze fra il Pentium II e l'UltraSPARC II relativamente all'unità di caricamento/decodifica delle istruzioni. (3 punti)

#### Soluzione

Vedi dispense del corso

---

### PUNTEGGIO DEGLI ESERCIZI PER GLI STUDENTI DEL NUOVO ORDINAMENTO DIDATTICO

Esercizio 1: 3 punti per la domanda (1) e 6 punti per la domanda (2)

Esercizio 2: 8 punti

Esercizio 3: 6 punti per la domanda (1) e 2 punti per la domanda (2)

Esercizio 4: 4 punti per ciascuna domanda.