Esercizi di preparazione al secondo compitino

Esercizio 1 *Un bus é dotato delle seguenti linee: una linea* CLK *per trasportare il segnale di clock, un insieme di linee per gli indirizzi (*Address*), un insieme di linee (*Cmd*) per indicare il tipo di operazione (lettura/ scrittura, memoria/ IO), un insieme di linee per il trasporto dei dati (*Data*)), una linea* /REQ *pilotata dal master, una linea* /RDY *pilotata dallo slave.*

Il ciclo di lettura standard ha una durata di 6 periodi di clock ($T_1 - T_6$)*: a partire dal fronte in salita di* T_1 *il master pilota le linee degli indirizzi e le linee* Cmd *(con un ritardo* T_{ad}*).*

A partire dal fronte in discesa di T_1 *(a metà periodo) il master asserisce la linea* /REQ *(con un ritardo* T_{r1}*). Il master memorizza i dati in corrispondenza del fronte in discesa a metà di* T_6*. Affinché i dati vengano memorizzati correttamente é necessario che siano stabili da un tempo* T_{ds}*. Durante tutta l'operazione lo slave mantiene* /RDY *ad 1. Nel caso che lo slave sia in grado di rispondere in un tempo piú breve puó accorciare il ciclo asserendo la linea* /RDY.

Di quanti cicli di clock puó essere accorciato il ciclo di lettura standard nel caso che:

- *La frequenza* f *di clock sia pari a 33,3 MHz*
- $T_{ad} \leq 5ns$
- $T_{ds} \geq 5ns$
- *Lo slave é in grado di rispondere in 60 ns a partire dall'istante in cui vengono forniti gli indirizzi.*

Esercizio 2 *Sia dato un sistema di elaborazione con spazio di indirizzamento per l'I/ O pari a 4k. Disegnare lo schema logico della della rete di selezione di un'interfaccia con 4 registri collocati consecutivamente a partire dall'indirizzo esadecimale 4C.*

Esercizio 3 *Un elaboratore su cui é implementato il meccanismo della paginazione, ha 16 pagine di memoria virtuale ma solo quattro* page frames. *Inizialmente la memoria (fisica) é vuota. Un programma accede alle pagine virtuali in questo ordine:*

0,7,2,7,5,8,9,2,4

3.1 *Quali accessi causano un* page fault *se si usa come strategia di rimpiazzamento l'algoritmo LRU ?*

3.2 *Quali accessi causano un* page fault *se si usa come strategia di rimpiazzamento l'algoritmo FIFO ?*

In entrambi i casi specificare quali pagine virtuali sono in memoria fisica alla fine degli accessi.

Esercizio 4 *Consideriamo un elaboratore con spazio di indirizzamento virtuale a 32 bit e bus dati a 8 bit, per cui sia implementata la paginazione.*

4.1 *Se la dimensione della pagina $\mathcal{D}pag$ é pari a 16 Kbyte, quante sono le possibili pagine virtuali ?*

4.2 *Ipotizzando inoltre che la dimensione della memoria fisica $\mathcal{D}_{phys}$ sia pari a 64Mbyte, quanti bit sono necessari per indicizzare un* page frame*?*

4.3 *Calcolare la dimensione della tabella delle pagine, nell'ipotesi che ogni entry occupi 4 byte.*

Esercizio 5 *Sia dato un sistema con segmentazione su domanda che abbia la seguente tabella dei descrittori di segmento:*

Selettore	***Base***	***Limite***	***P***	***DPL***
1	*0101100001*	*1000*	*1*	*rw*
2	*0101100000*	*1000*	*0*	*rx*
3	*1111000000*	*1111*	*1*	*r*
4	*1011000010*	*1111*	*1*	*r*
...	...			

dove il bit P indica se il segmento é presente in memoria fisica, e il campo DPL specifica i diritti di accesso (r $\rightarrow$ lettura, w $\rightarrow$ scrittura, x $\rightarrow$ esecuzione).

Per ciascuno dei seguenti accessi alla memoria virtuale, dire quale indirizzo fisico é calcolato, nel caso non vengano sollevate eccezioni. Se viene sollevata un'eccezione, se ne specifichi il tipo.

a) Accesso in scrittura all'indirizzo con selettore 1 e offset 0101

b) Accesso in lettura all'indirizzo con selettore 1 e offset 1010

c) Salto all'indirizzo con selettore 2 e offset 0010

d) Accesso in scrittura all'indirizzo con selettore 3 e offset 1010

e) Accesso in lettura all'indirizzo con selettore 3 e offset 1010

f) Accesso in lettura all'indirizzo con selettore 4 e offset 1110

g) Salto all'indirizzo con selettore 4 e offset 1110

Soluzioni

Soluzione esercizio 1

La soluzione é 3. Infatti abbiamo

$$T = \frac{1}{f} = 30ns \quad e \quad (n \times T) - (T_{ad} + T_{ds} + 0.5 \times T) \geq 60ns;$$

avendo indicato con n il numero di cicli di clock necessari a completare il trasferimento. Sostituendo i valori dati otteniamo: $n \geq 2.833$ e poiché n deve essere intero, lo arrotondiamo a 3.

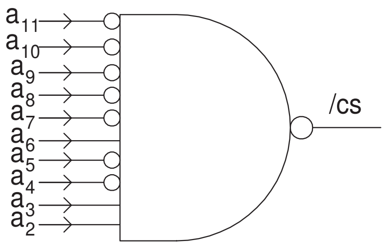
Il ciclo puó perció essere accorciato di 6-3=3 cicli.

Soluzione esercizio 2

Uno spazio di indirizzamento per l'I/O pari a 4k significa avere 12 linee per indirizzare i registri di I/O. I registri dell'interfaccia occupano i 4 indirizzi che vanno da `4C` a `4F`. Esprimendoli in binario sono gli indirizzi seguenti:

- 000001001100
- 000001001101
- 000001001110
- 000001001111

quindi la maschera che genera il *chip select* avrá in ingresso solamente i 10 bit piú significativi. Ricordandoci che *chip select* é attivo basso abbiamo:



Soluzione esercizio 3

Illustriamo con questa tabella la storia degli accessi del programma:

tempo	1	2	3	4	5	6	7	8	9
pag. acceduta	0	7	2	7	5	8	9	2	4

3.1

Nell'algoritmo LRU, all'atto di un page fault, se é necessario rimpiazzare una pagina, si opera lo *swap out* della pagina che non é stata riferita da piú tempo.

I primi tre accessi comportano un page fault ciascuno, ma senza rimpiazzamento, in quanto inizialmente la memoria fisica é vuota. Al tempo 4 viene riferita la pagina 7 che é giá in memoria. Al tempo 5 si porta in memoria la pagina 5 senza rimpiazzare alcuna pagina: a questo punto la memoria fisica é piena, e contiene le pagine $\mathcal{M}_5 = \{0, 7, 2, 5\}$. Al tempo 6 si tenta di accedere alla pagina 8: é necessario rimpiazzare una pagina e la scelta cade sulla pagina 0, riferita al tempo 1. Avremo perció $\mathcal{M}_6 = \{8, 7, 2, 5\}$. Al tempo 7 é necessario rimpiazzare una pagina per poter portare in memoria fisica la pagina 9. La scelta cade sulla pagina 2, che pur essendo stata caricata dopo la 7, é stata riferita meno recentemente. Continuando con l'applicazione dell'algoritmo avremo:

$$\mathcal{M}_7 = \{8, 7, 9, 5\} \quad \mathcal{M}_8 = \{8, 2, 9, 5\} \quad \mathcal{M}_9 = \{8, 2, 9, 4\}$$

per un totale di 8 page fault rispettivamente agli istanti $\{1, 2, 3, 5, 6, 7, 8, 9\}$.

3.2

Nell'algoritmo FIFO, all'atto di un page fault, se é necessario rimpiazzare una pagina, si opera lo *swap out* della pagina che é stata caricata in memoria meno recentemente.

I primi tre accessi comportano un page fault ciascuno, ma senza rimpiazzamento, in quanto inizialmente la memoria fisica é vuota. Al tempo 4 viene riferita la pagina 7 che é giá in memoria. Al tempo 5 si porta in memoria la pagina 5 senza rimpiazzare alcuna pagina: a questo punto la memoria fisica é piena, e contiene le pagine $\mathcal{M}_5 = \{0, 7, 2, 5\}$. Al tempo 6 si tenta di accedere alla pagina 8: é necessario rimpiazzare una pagina e la scelta cade sulla pagina 0, caricata al tempo 1. Avremo perció $\mathcal{M}_6 = \{8, 7, 2, 5\}$. Al tempo 7 é necessario rimpiazzare una pagina per poter portare in memoria fisica la pagina 9. La scelta cade sulla pagina 7, che pur essendo stata riferita dopo la 2, é stata caricata prima di essa. Continuando con l'applicazione dell'algoritmo avremo:

$$\mathcal{M}_7 = \mathcal{M}_8 = \{8, 9, 2, 5\} \quad \mathcal{M}_9 = \{8, 9, 4, 5\}$$

per un totale di 7 page fault rispettivamente agli istanti $\{1, 2, 3, 5, 6, 7, 9\}$.

Soluzione esercizio 4

Un tale elaboratore avrá uno spazio di indirizzamento virtuale di dimensione $\mathcal{D}_{\text{virt}}$ pari a 4Gbyte.

4.1

Abbiamo che il numero delle pagine virtuali é pari a:

$$\mathcal{N}_{\text{virt}} = \frac{\mathcal{D}_{\text{virt}}}{\mathcal{D}_{\text{pag}}} = 2^{32-14} = 2^{18} = 256\text{K}$$

4.2

Abbiamo che il numero delle pagine fisiche é pari a:

$$\mathcal{N}_{\text{phys}} = \frac{\mathcal{D}_{\text{phys}}}{\mathcal{D}_{\text{pag}}} = 2^{26-14} = 2^{12} = 4\text{K}$$

per cui sono necessari 12 bit per indirizzare un page frame.

4.3

Poiché abbiamo 256K pagine virtuali possibili la dimensione della tabella delle pagine é di 256k * 4 Byte = 1Mbyte.

Soluzione esercizio 5

Prima di leggere il campo Base dalla tabella dei descrittori di segmento si controlla il bit P e il campo DPL. Se le condizioni di accesso sono verificate si controlla che l'offset non ecceda il campo Limite, dopodiché se ancora non sono sollevate eccezioni si sommano Base e offset per ottenere l'indirizzo fisico. Per cui avremo:

a) Indirizzo fisico $\rightarrow 0101100001 + 0101 = 0101100110$

b) Violazione limite del segmento

c) Segment Fault

d) Violazione di protezione, segmento in sola lettura

e) Indirizzo fisico $\rightarrow 1111000000 + 1010 = 1111001010$

f) Indirizzo fisico $\rightarrow 1011000010 + 1110 = 1011010000$

g) Violazione di protezione, il segmento non ha il permesso di esecuzione