

PROVA SCRITTA DEL MODULO DI
CALCOLATORI ELETTRONICI
CORSO DI LAUREA IN INGEGNERIA ELETTRICA, ELETTRONICA ED INFORMATICA, BIOMEDICA
16 gennaio 2017

NOME:

COGNOME:

MATRICOLA:

ESERCIZIO 1 (4 punti)

Descrivere la tabella di verità e le caratteristiche di un flip flop JK, disegnando il circuito completo visto durante il corso.

ESERCIZIO 2 (12 punti)

1. (3 punti) Sia dato un disco con le seguenti caratteristiche: velocità di rotazione pari a 6000 giri al minuto, tempo necessario alla testina per spostarsi da una traccia alla successiva uguale a 0.5 ms, settori da 1 KB, 200 settori per traccia. Calcolare il tempo medio di lettura di un file da 10 KB sapendo che la testina si trova inizialmente in un punto qualunque del disco e che la distanza media tra due settori successivi del file è pari a 2 tracce.
2. (3 punti) Utilizzando il tempo medio di accesso al disco calcolato al punto precedente, si calcoli l'hit ratio di cache minimo (H_c), relativo ad una gerarchia a 3 livelli (cache, memoria primaria, disco), affinché il tempo medio di accesso alla gerarchia sia 30 ms. Siano dati i seguenti altri dati: $H_p = 0.8$; $T_p = 20$ ms; $T_c = 10$ ms, dove H_p è l'hit ratio della memoria primaria, T_p il tempo medio di accesso alla memoria primaria e T_c il tempo medio di accesso alla memoria cache.
3. (6 punti) L'indirizzo di memoria fisica di un calcolatore è partizionato per l'accesso alla cache (indirizzamento diretto) come segue: TAG 2 bit, cache index: 4 bit, offset: 2 bit. Sapendo che ogni singolo byte è indirizzabile, indicare, **motivando le risposte**:
 - a. (2 punti) la dimensione della memoria principale e della memoria cache in byte;
 - b. (2 punti) quante linee di cache sono disponibili e quali sono i valori di block frame corrispondenti a cache index pari a 0.
 - c. (2 punti) disegnare il circuito della rete logica di confronto dei TAG ed indicare a cosa serve.

ESERCIZIO 3 (8 punti)

Si scriva il codice Assembly MIPS di una funzione che, dato un vettore v (in \$4) di N elementi (in \$6), indichi quanti elementi (valore da memorizzare in \$7) pari ad un dato valore x (in \$5), siano presenti in v :

```
int conta_se(int *v, int x, int N) {  
    int i,m;  
    for(i=0, m=0; i<N; i++)  
        if(x==v[i])  
            m++;  
    return m;  
}
```

ESERCIZIO 4 (9 punti)

Si consideri un calcolatore in cui la CPU esegue 10^5 istruzioni/s. L'esecuzione di una istruzione richiede 5 cicli di clock, 3 dei quali tengono occupato il bus di sistema. Si ipotizzi che il 85% dell'Instruction Rate sia usato dalla CPU per eseguire programmi che non contengono trasferimenti di I/O. L'ampiezza della linea dati del bus è pari a 32 bit.

Si consideri il caso in cui il trasferimento dei dati avvenga mediante IO da programma, con le seguenti 4 istruzioni:

- a. LOAD parola dalla periferica al registro CPU
 - b. STORE parola da registro CPU a memoria
 - c. generazione indirizzo di memoria successivo
 - d. conteggio dati da trasferire.
1. (3 punti) Calcolare la massima frequenza di trasferimento dati ottenibile (espressa in kB/s) fra una periferica collegata al bus di sistema e la memoria principale.
 2. (3 punti) Calcolare la massima frequenza di trasferimento dati ottenibile (espressa in kB/s) nel caso in cui si usi la modalità "transparent" DMA. Si ipotizzi che una operazione di lettura/scrittura della memoria richieda un ciclo di clock.
 3. (3 punti) Spiegare quali 'passi' sostituiscono le istruzioni nel caso DMA.

ESERCIZIO 1

La tabella di eccitazione (tabella di verità) del flip flop JK è data da:

| Q | Q' | J | K |
|---|----|---|---|
| 0 | 0 | 0 | D |
| 0 | 1 | 1 | D |
| 1 | 0 | D | 1 |
| 1 | 1 | D | 0 |

Dove la "D" significa "don't care", ovvero la relazione stato presente/stato futuro è indipendente dal relativo segnale in ingresso. La tabella si può scrivere anche nella forma estesa indicata nelle slide del corso, che differisce da quella del flip flop SR per il fatto di ammettere la configurazione J=K=1 in ingresso, con effetto di commutazione dello stato futuro rispetto a quello presente.

Il circuito è indicato nelle slide del corso, con la componente latch di tipo "master" e la componente latch di tipo "slave" connesse da un segnale di sincronismo che attiva alternativamente le due componenti onde permettere il disaccoppiamento dei segnali di stato ovvero realizzare la memorizzazione dello stato futuro nel master mentre in uscita si presenta, ad un dato istante, lo stato presente attraverso lo slave.

ESERCIZIO 2

Soluzione

1. $TROT = 60 / 6000 = 10 \text{ ms}$
 $TLAT = TROT / 2 = 5 \text{ ms}$ (tempo di latenza)
 $Tlett = TROT / 200 = 50 \mu\text{s}$ (tempo di lettura di un settore)
 $Tpos = 0.5 \times 2 = 1 \text{ ms}$
Numero di settori richiesti per leggere 10 KB: $10\text{KB}/1\text{KB} = 10$
Tempo di lettura del file da 10 KB:
 $= 10 * (TLAT + TPOS + Tlett) = 10 * (5 \text{ ms} + 1 \text{ ms} + 50 \mu\text{s}) \cong 60 \text{ msec}$

2. Per calcolare H_c basta esplicitarlo dalla formula:

$$\bar{T} = H_c T_C + (H_P - H_c)(T_P + T_C) + (1 - H_P)(T_D + T_P + T_C)$$
$$H_c = (T_C + T_P + T_D - H_P T_D - \bar{T}) / T_P = 0.6$$

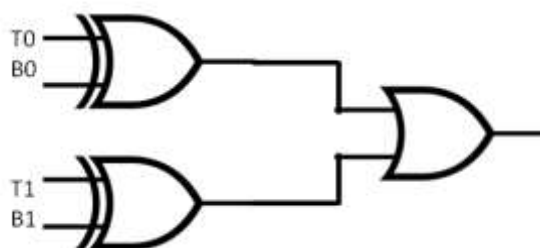
3. TAG 2 bit, cache index: 4 bit, offset: 2 bit

Con tale configurazione il numero di bit di indirizzo della memoria primaria è pari a 8 (TAG+cache index+offset), mentre il numero di bit di indirizzo della cache è pari a 6 (cache index+offset). Quindi la dimensione della memoria primaria è pari a 256 byte e quella della cache è pari a 64 byte.

I block frame con cache index pari a zero si ottengono facilmente moltiplicando i valori 0, 1, 2, 3, ovvero i quattro possibili valori di TAG, per $2^4=16$, pari al numero complessivo di linee di cache, ottenendo quindi i valori 0, 16, 32, 48.

Infine, il circuito di confronto TAG, necessario per verificare se un blocco vi è già presente e quindi possa verificarsi un "hit", è dato da due livelli di logica: il primo costituito da tante porte XOR quanti sono i bit del TAG. Queste porte si attivano esclusivamente se i bit di ingresso sono diversi. Le uscite del primo livello di logica diventano gli ingressi di una porta OR. Si ottiene quindi una rete logica combinatoria che ricevendo in ingresso due valori di TAG (quello relativo alla parola chiamata e quello relativo al blocco di primaria memorizzato nella corrispondente linea di cache), avrà in uscita il valore 1 se i due TAG sono diversi, e 0 se sono identici. Un "hit" si avrà in corrispondenza di un valore 0 in uscita da questa rete, disegnata come segue:

$\{T0, T1\} == \text{tag della parola chiamata}$
 $\{B0, B1\} == \text{tag del blocco presente nella linea di cache}$



ESERCIZIO 3

Soluzione

$\$8 \leftarrow i;$
 $\$9 \leftarrow v(i).$

```
elabora:  addi $29, $29, -8
          sw $8, 0($29)
          sw $9, 4($29)
          move $7, $0
          move $8, $0
for:      beq $8, $6, exit
          lw $9, 0($4)
          bne $9, $5, continue
          addi $7, $7, 1
continue: addi $8, $8, 1
          addi $4, $4, 4
          j for
exit:     lw $8, 0($29)
          lw $9, 4($29)
          addi $29, $29, 8
          jr $31
```

ESERCIZIO 4

Soluzione.

1. Nel caso di trasferimento mediante I/O da programma, per trasferire una parola occorrono 4 istruzioni. La CPU è impegnata per l'85% del tempo a eseguire istruzioni che non coinvolgono l'I/O, dunque può usare solo il 15% del tempo per eseguire istruzioni di trasferimento dati con periferiche. In termini di istr./sec questo tempo è pari a $0.15 \times 10^5 \text{ istr./s} = 1.5 \times 10^4 \text{ istr./s}$. Dal momento che per trasferire una parola servono due istruzioni, la velocità di trasferimento è pari a:
 $1.5 \times 10^4 \text{ istr./s} / (4 \text{ istr./parola}) = 3750 \text{ parole/s}$. La dimensione di una parola è pari a 32 bit (4 byte), da cui si ricava la velocità di trasferimento di **14.65 kB/s**.
2. Nel caso di 'trasparent DMA' posso trasferire i dati tutte le volte che il bus di sistema è libero. Nel caso in esame questo tempo è pari alla somma del 15% del tempo lasciato libero dall'esecuzione di istruzioni che non coinvolgono I/O, più i due cicli/istruzione in cui il bus è libero. Pertanto durante l'85% del tempo posso trasferire due parole/istr.:
 $0.85 \times 2 \text{ parole/istr} \times 10^5 \text{ istr./s} = 1.7 \times 10^5 \text{ parole/s}$
Nel restante 15% del tempo posso trasferire 5 parole/istr.:
 $0.15 \times 5 \text{ parole/istr.} \times 10^5 \text{ istr./s} = 0.75 \times 10^5 \text{ parole/s}$
In **totale**, nel caso di trasferimento con DMA la velocità totale di trasferimento è pari a: **$(1.7 + 0.75) \times 10^5 \text{ parole/s} = 2.45 \times 10^5 \text{ parole/s} = 239 \text{ kB/s}$** .
3. Il 'controller' DMA esegue le operazioni di generazione indirizzi e conteggio dati trasferiti usando registri interni al controller oltre, ovviamente, a trasferire i dati dalla periferica alla memoria.