

**PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO
2 Luglio 2009**

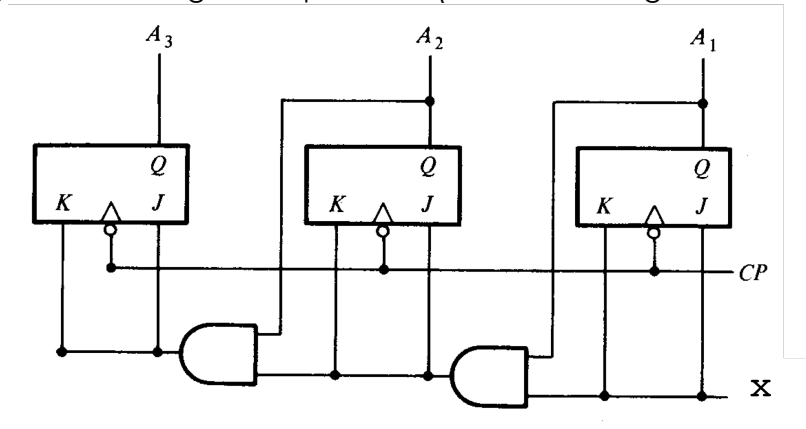
NOME:

COGNOME:

MATRICOLA:

ESERCIZIO 1 (9 punti)

Analizzare la seguente rete logica sequenziale (CP indica il segnale di clock, x l'ingresso):



Spiegare il ruolo dei singoli elementi e descrivere il funzionamento della rete per una durata pari a 8 cicli di clock. Qual è la funzione realizzata dalla rete?

ESERCIZIO 2 (8 punti)

Si consideri un calcolatore che dispone di una memoria principale di 256 Mbyte e di una memoria cache di 512 Kbyte. E' possibile accedere al singolo byte e la memoria è suddivisa in blocchi da 16 byte.

- 1) (4 punti) Spiegare, precisando il significato e la funzione dei diversi campi, come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache nel caso venga usata la modalità di indirizzamento
 - (a) Diretto
 - (b) "associativo su insiemi", e ciascun insieme contenga due blocchi
- 2) (4 punti) Si considerino le due parole di indirizzo rispettivamente 21A0x43 e 10y0F4D. Calcolare i valori delle cifre x e y tali per cui le due parole si trovino
 - (a) Nella stessa "linea" di cache nel caso di indirizzamento diretto
 - (b) Nello stesso insieme nel caso di indirizzamento associativo su insiemi a due vie

ESERCIZIO 3 (8 punti)

Scrivere una funzione Assembly MIPS che calcoli la varianza di un vettore x secondo la formula:

$$\sigma^2 = \frac{1}{N-1} \sum_i (x_i - \bar{x})^2$$

```
float varianza(int *x, int N) {
    float var = 0;
    float mean = media(x,N);
    for(int i=0; i<N; i++)
        var += (x[i]-mean)*(x[i]-mean);
    var = dividi(var,N-1);
    return var;
}
```

Il parametro x (ovvero l'indirizzo del primo elemento di x: &x[0]) viene passato in \$4, N (dimensione di x) è passato in \$5. Si supponga di poter utilizzare la funzione media che prende in ingresso un vettore x (in \$4) e la sua dimensione N (in \$5), e restituisce la media di x in \$6. Si supponga inoltre di poter utilizzare anche la funzione dividi che restituisce il valore della divisione tra i contenuti dei registri \$4 e \$5 in \$6.

ESERCIZIO 4 (8 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 75% 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:

- 1) LOAD *parola* dalla periferica al registro CPU
 - 2) STORE *parola* da registro CPU a memoria
 - 3) generazione indirizzo di memoria successivo
 - 4) conteggio dati da trasferire.
- a) (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 mediante I/O da programma, sapendo che una parola è pari a 32 bit.
- b) (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 e che una parola sia pari a 32 bit.
- c) (2 punti) Si descrivano, in modo chiaro e sintetico, le differenze tra il sistema di arbitraggio centralizzato e quello distribuito, facendo particolare riferimento ai sistemi di gestione delle periferiche "daisy chain" e "a richieste indipendenti".

ESERCIZIO 1

Soluzione

Il circuito è formato da flip-flop di tipo JK, da porte logiche AND. Il circuito è sincronizzato da un segnale di clock che abilita le transizioni di stato nei flip-flop. Gli ingressi ai flip-flop sono infine controllati da un segnale 'x'. La tabella di eccitazione di un flip-flop JK è riportata in basso.

$Q(t)$	$Q(t+1)$	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Ipotesizziamo che all'inizio tutti i flip-flop si trovino allo stato 0 e che il segnale x sia pari a 1. Dopo il primo colpo di clock, lo stato del primo flip flop passa da 0 a 1 ($J = 1$, $K = 1$), dunque l'uscita A1 è pari a 1. Al secondo colpo di clock lo stato del primo flip flop torna a 0, mentre lo stato del secondo flip flop passa a 1 perché gli ingressi sono entrambi pari a 1 dovuti all'uscita precedente del primo flip flop (N.B. ovviamente la durata del clock dovrà essere tale da consentire la transizione di stato di un flip flop ma non la propagazione del nuovo stato ai flip flop successivi. Questi ultimi invece vedono in ingresso lo stato posseduto dai flip flop a cui sono collegati relativo all'istante precedente). Se proseguiamo nel ragionamento per i successivi 6 colpi di clock, osserveremo che le uscite A3 A2 A1 effettuano le seguenti transizioni: $000 \Rightarrow 001 \Rightarrow 010 \Rightarrow 011 \Rightarrow 100 \Rightarrow 101 \Rightarrow 110 \Rightarrow 111$ (si osservi che il flip-flop A3 commuta solo quando lo stato precedente dei flip-flop A2 e A1 era pari a 1). Al nono colpo di clock tutte le variabili tornano a 0. La rete sequenziale dunque realizza un contatore binario sincronizzato dal clock.

ESERCIZIO 2

Soluzione

1) Per indirizzare 256 Mbyte occorre un indirizzo di almeno 28 bit. Per indirizzare il singolo byte all'interno di un blocco occorrono 4 bit ($16 = 2^4$), che coincidono con i 4 bit meno significativi dell'indirizzo di memoria primaria. I restanti 24 bit costituiscono l'indirizzo del "block frame". Per indirizzare la cache, il "block frame" viene interpretato diversamente a seconda che l'indirizzamento sia di tipo "diretto" o "associativo su insiemi".

(a) Indirizzamento diretto. In questo caso devo poter indirizzare ciascuno dei 32K blocchi contenuti nella cache ($512\text{Kbyte}/(16\text{byte}/\text{blocco})$). Occorrono 15 bit che coincidono con i 15 bit meno significativi del "block frame". Pertanto i 28 bit di indirizzo della memoria primaria vengono interpretati come:

tag	cache index	Offset
9 bit	15 bit	4 bit

(b) Indirizzamento "associativo su insiemi". In questo caso devo poter indirizzare ciascuno dei 16 insiemi in cui sono suddivisi i blocchi contenuti nella cache ($\frac{512\text{Kbyte}}{2\text{blocchi} \cdot 16\text{byte}} = 16\text{K insiemi}$). Occorrono 14 bit che coincidono con i 14 bit meno significativi del "block frame"

Pertanto i 28 bit di indirizzo della memoria primaria vengono interpretati come:

tag	cache index	offset
10 bit	14 bit	4 bit

2. Perché le due parole si trovino nella stessa "linea" di cache o nello stesso insieme è necessario eguagliare l'index delle due parole. Faremo riferimento alla suddivisione in campi dell'indirizzo mostrato nella soluzione della prima parte dell'esercizio.

Nel caso di indirizzamento diretto avremo:

tag	cache index	offset
9 bit	15 bit	4 bit
001000011	0100000xxxx0100	0011
00010000y	yyy000011110100	1101

Cioè, in esadecimale, $x = F$, mentre y può essere pari a 2 od A perché la cifra più significativa è nel tag.

Analogamente, per il caso di indirizzamento associativo su insiemi a due vie, si ha:

tag	cache index	offset
10 bit	14 bit	4 bit
0010000110	100000xxxx0100	0011
00010000yy	yy000011110100	1101

Cioè, in esadecimale, $x = F$, mentre y può essere pari a 2, 6, A, E perché le due cifre più significative sono nel campo tag.

ESERCIZIO 3

Soluzione

```
varianza:    addi $29, $29, -24
             sw $31, 0($29)
             sw $8, 4($29)
             sw $9, 8($29)
             sw $10, 12($29)
             sw $4, 16($29)
             sw $5, 20($29)

             jal media          #in $6 trovo la media, non altera $4, $5
             move $8, $0        #i=0;
             move $10, $0       #var=0;
for_var:     beq $8, $5, exit_var
             lw $9, 0($4)       #$9<-v[i]
             sub $9,$9,$6       #v[i]-media(v)
             mul $9,$9,$9       #(v[i]-media(v))^2
             add $10, $10, $9   #var+=(v[i]-media(v))^2
             addi $8,$8,1       #i++
             addi $4, $4, 4     #&v[i] (next...)
             j for_var

exit_var:    move $4, $10        #sposto la varianza in $6
             subi $5, $5, 1     #calcolo N-1
             jal dividi
             lw $31, 0($29)
             lw $8, 4($29)
             lw $9, 8($29)
             lw $10, 12($29)
             lw $4, 16($29)
             lw $5, 20($29)
             addi $29, $29, 24
             jr $31
```

ESERCIZIO 4

Soluzione

- a) Nel caso di trasferimento mediante I/O da programma, per trasferire una parola occorrono 4 istruzioni. La CPU è impegnata per il 75% del tempo a eseguire istruzioni che non coinvolgono l'I/O, dunque può usare solo il 25% del tempo per eseguire istruzioni di trasferimento dati con periferiche. In termini di istr./sec questo tempo è pari a **$0.25 \times 10^5 \text{ istr./s} = 2.5 \times 10^4 \text{ istr./s}$** . Dal momento che per trasferire una parola servono quattro istruzioni, la velocità di trasferimento è pari a:
- $2.5 \times 10^4 \text{ istr./s} / (4 \text{ istr./parola}) = 6250 \text{ parole/s}$** . La dimensione di una parola è pari a 32 bit (4 byte), da cui si ricava la velocità di trasferimento di **24.41 kB/s**,
- b) 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 25% 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 il 75% del tempo posso trasferire una parola/istr.:
- $0.75 \times 2 \text{ parole/istr} \times 10^5 \text{ istr./s} = 1.5 \times 10^5 \text{ parole/s}$**
Nel restante 25% del tempo posso trasferire 5 parole/istr.:
- $0.25 \times 5 \text{ parole/istr.} \times 10^5 \text{ istr./s} = 1.25 \times 10^5 \text{ parole/s}$**
In **totale**, nel caso di trasferimento con DMA la velocità totale di trasferimento è pari a: **$(1.5 + 1.25) \times 10^5 \text{ parole/s} = 2.75 \times 10^5 \text{ parole/s} = 1074.22 \text{ kB/s}$**
- c) si vedano le dispense del corso.