

**PROVA SCRITTA DEL CORSO DI**  
**CALCOLATORI ELETTRONICI**  
**NUOVO ORDINAMENTO DIDATTICO (5 CFU)**

20 Aprile 2010

**NOME:**

**COGNOME:**

**MATRICOLA:**

**ESERCIZIO 1 (7 punti)**

1. (4 punti) Progettare una rete logica che calcoli il complemento a 2 di un operando a tre bit.
2. (3 punti) Indicare a quale tipo di reti logiche appartiene la rete progettata e spiegarne il motivo.

**ESERCIZIO 2 (10 punti)**

E' data una gerarchia di memorie cache-primaria. La memoria primaria è di 512 KB mentre la cache è di 64 KB. E' possibile indirizzare il singolo byte, e la memoria primaria è suddivisa in blocchi di 32 B.

1. (2 punti) Indicare, sapendo che l'indirizzo della prima parola è pari a 0, l'indirizzo della prima e dell'ultima parola del blocco di memoria primaria con block frame pari a 16.
2. (3 punti) Indicare, specificando l'ampiezza e la funzione dei diversi campi, come vengono interpretati gli indirizzi di memoria primaria secondo il metodo di indirizzamento associativo, set-associativo a otto vie, e diretto.
3. (3 punti) Ipotizzando la cache piena, indicare in quale linea di cache viene allocato il blocco indicato nel punto 1 con i metodi di indirizzamento esaminati nel punto 2. Si indichi e descriva, dove necessario, almeno un algoritmo di rimpiazzamento.
4. (2 punti) Calcolare il tempo medio di accesso alla gerarchia sapendo che l'hit ratio di cache è pari a 0.95, il tempo di accesso alla cache è pari a 4 ns, il tempo di accesso alla primaria è pari a 40 ns.

**ESERCIZIO 3 (8 punti)**

Si scriva una *procedura* Assembly MIPS che esegua il prodotto scalare tra 2 vettori,  $v$  e  $w$  (di uguale dimensione  $N$ ), passati per indirizzo rispettivamente in  $\$4$  e  $\$5$ . La dimensione  $N$  dei vettori viene passata alla procedura usando  $\$6$ . Si ritorni il valore del prodotto scalare in  $\$7$ . La procedura deve occuparsi del salvataggio e del ripristino del contesto, secondo la modalità "callee save". Si ricorda che il prodotto scalare di 2 vettori si calcola come

$\sum_{i=1}^N v_i w_i$ , dove l'indice  $i$  rappresenta l' $i$ -esimo elemento di ciascun vettore.

**ESERCIZIO 4 (8 punti)**

Si consideri un calcolatore dotato di un sistema di memorizzazione su disco avente le seguenti caratteristiche: 100 tracce per superficie, tracce da 16 Kbyte, velocità di rotazione pari a 7200 giri/minuto, tempo impiegato dalla testina per spostarsi dalla traccia più interna a quella più esterna (o viceversa) pari a 120 ms.

I file sono memorizzati in settori da 4 Kbyte. I settori di uno stesso file sono memorizzati in maniera tale che la distanza media fra due settori successivi è pari a 12 tracce.

1. (5 punti) Calcolare le dimensioni di un file il cui tempo di caricamento è di 300 ms.
2. (3 punti) Calcolare il tempo di caricamento del programma al punto 1 nel caso la distanza media fra due settori successivi sia pari a 1 traccia.

Nota bene: Si ipotizzi che ciascun settore sia contenuto in un'unica traccia e che la testina si trovi inizialmente in un punto qualsiasi del disco.

## ESERCIZIO 1

### Soluzione

Tabella di verità della rete in oggetto :

Ingressi			Uscite		
A	B	C	A'	B'	C'
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

AB \ C	00	01	11	10
0		1		1
1	1	1		

$$A' = \overline{A}B + \overline{A}C + AB \cdot \overline{C}$$

AB \ X	00	01	11	10
0		1	1	
1	1			1

$$B' = BC + \overline{B}C$$

AB \ C	00	01	11	10
0				
1	1	1	1	1

$$C' = C$$

Si tratta ovviamente di una rete logica combinatoria, in quanto le uscite dipendono soltanto dai valori degli ingressi ad un dato istante.

## ESERCIZIO 2

### Soluzione

1. Indirizzo della prima parola:  $\text{Block Frame} * \text{DimBlocco} = 16 * 32 = 512$ . Indirizzo dell'ultima = Indirizzo della prima +  $\text{DimBlocco} - 1 = 512 + 32 - 1 = 543$ .
2. Memoria indirizzabile  $2^{19}$  B  $\rightarrow$  19 bit di indirizzamento, di cui 5 per l'offset (i blocchi sono di  $2^5$  B).

Nel caso del metodo associativo, si ha:

< Block Frame (TAG) 14 bit > < Offset 5 bit >

La cache è invece formata da  $2^{16}$  B, da cui

< TAG 3 bit > < C.I. 11 bit > < Offset 5 bit >

Infine, nel caso del metodo set-associativo a otto vie:

< TAG 6 bit > < C.I. 8 bit > < Offset 5 bit >.

3. Metodo associativo: a meno che in una delle linee della cache non sia presente un blocco con medesimo TAG, il blocco dovrà essere allocato nel blocco soddisfacente ad esempio l'algoritmo di rimpiazzamento FIFO, per il quale il primo blocco da rimpiazzare è il primo ad essere stato allocato in cache.

Metodo diretto: è necessario calcolare il resto della divisione  $16/2^{11}$ . Tale operazione da evidentemente 16. Quindi il blocco 16 di primaria viene allocato, sostituendo il blocco precedente, se con TAG diverso, nella linea 16 di cache.

Metodo set-associativo: è necessario calcolare il resto della divisione  $16/2^8$ . Anche in questo caso il valore ottenuto è 16, che però corrisponde all'indirizzo dell'insieme. Il blocco 16 sostituirà, a meno di hit, uno degli otto blocchi allocati in altrettante linee, ad esempio con strategia FIFO.

4. E' sufficiente applicare la formula del tempo medio di accesso:

$$T = T_c + (1 - H_c) * T_p = 4 + 0.05 * 40 = 6 \text{ ns.}$$

### ESERCIZIO 3

#### Soluzione

scalar\_product:

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

        move $7, $0
        move $8, $0
for:     beq $8, $6, exit_for
        lw $9, 0($4)
        lw $10, 0($5)
        mul $9, $9, $10
        add $7, $7, $9
        addi $4, $4, 4
        addi $5, $5, 4
        addi $8, $8, 1
        j for

        #scalar_prod = 0
        #i=0
        #i==N, exit
        #$9 <- v[i]
        #$10 <- w[i]
        #$9 <- v[i] * w[i]
        #scalar_prod += v[i] * w[i]
        #incremento indirizzo v
        #incremento indirizzo w
        #i++

exit_for:
        lw $4, 0($29)
        lw $5, 4($29)
        lw $8, 8($29)
        lw $9, 12($29)
        lw $10, 16($29)
        addi $29, $29, 20
        jr $31
```

#### ESERCIZIO 4

##### Soluzione.

1.

Poiché la testina impiega 120 ms per spostarsi di 99 tracce,  $(120/99)$  ms è il tempo di posizionamento da una traccia a quella contigua. Due settori di uno stesso file distano 12 tracce, quindi il tempo medio di posizionamento è dato da  $(120/99) \cdot 12 = 14.54$  ms.

Una rotazione, ovvero la lettura di una traccia, richiede  $60/7200$  s = 8.33 ms, quindi  $T_L = 4.17$  ms.

Infine, un settore occupa  $\frac{1}{4}$  di traccia. Dunque viene letto in  $T_R = (1/4) \cdot 8.33$  ms = 2.08 ms. Indicando con x la dimensione del file in Kbyte, e ricordandosi che ogni settore richiede 4 Kbyte, l'espressione  $(x/4)$  rappresenta il numero totale di settori di cui è composto il file. Si ha allora:

$$300 \text{ ms} = (x/4) \cdot (T_P + T_L + T_R) = 5.1975x \text{ ms}$$

quindi  $x = 57$  Kbyte circa.

2.

Il tempo di posizionamento si riduce a  $(120/99)=1.21$  ms. Il tempo di caricamento del file è allora dato da:

$$(57/4) \cdot (1.21 + 4.17 + 2.08) = 106.305 \text{ ms.}$$