

**PRIMA PROVA INTERMEDIA DEL MODULO DI**  
**CALCOLATORI ELETTRONICI**  
CORSO DI LAUREA IN INGEGNERIA ELETTRICA, ELETTRONICA ED INFORMATICA  
CORSO DI LAUREA IN INGEGNERIA BIOMEDICA  
ISCRITTI A.A. 2016/17 e precedenti  
17 Aprile 2018

<b>NOME:</b>	<b>COGNOME:</b>	<b>MATRICOLA:</b>	<b>CFU:</b>
--------------	-----------------	-------------------	-------------

**ESERCIZIO 1 (8 punti)**

- 1) Si progetti una rete logica in grado di riconoscere in una sequenza di bit la stringa 0101 ponendo a 1 l'uscita solo quando si abbia il riconoscimento di tale stringa. Si sintetizzi la rete minima che realizza tale funzione utilizzando FF-T. Disegnare il circuito logico che rappresenta la rete minima sintetizzata.
- 2) Illustrare in modo chiaro e sintetico le caratteristiche di un MUX 4-1.

**ESERCIZIO 2 (14 punti)**

Si consideri la gerarchia di memoria costituita da una memoria primaria caratterizzata da 32 blocchi composti ciascuno da 4 parole di 32 B. E' possibile indirizzare il singolo byte. **Motivando** le risposte:

1. (2 punti) Calcolare la dimensione della memoria in B.
2. (2 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 di indirizzamento diretto con cache di 8 linee e set associativa a due vie.
3. (4 punti) Indicare il block frame di tutti i blocchi che possono essere mappati nell'insieme di indirizzo 0.
4. (4 punti) Ipotizzando che, a cache vuota, sia richiesto l'accesso sequenziale a tutti i byte dei blocchi 7 e 23 per due volte consecutive, indicare lo stato della cache ed il numero di miss e di hit e calcolare il relativo hit ratio di cache.
5. (2 punti) Se il tempo di accesso in cache è pari a 4 nsec mentre quello in primaria è pari a 40 nsec, calcolare il tempo medio di accesso alla gerarchia utilizzando l'hit ratio calcolato al punto precedente. Esprimere tutti i tempi **in nanosecondi**.

**ESERCIZIO 3 (4 punti)**

Si consideri un disco magnetico caratterizzato dai seguenti parametri: velocità 6000 rpm, tracce da 128 kB, settori da 256 B e tempo di posizionamento da traccia a quella adiacente di 1 ms. Si calcoli il tempo medio di lettura di un file da 16 kB supponendo che i settori del file si trovino **su tracce adiacenti**, con la testina posizionata all'istante iniziale sul primo settore utile.

**ESERCIZIO 4 (7 punti)**

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificati utilizzando il codice di Hamming. Ipotizzando di trasferire la stringa 00110011 e motivando la risposta:

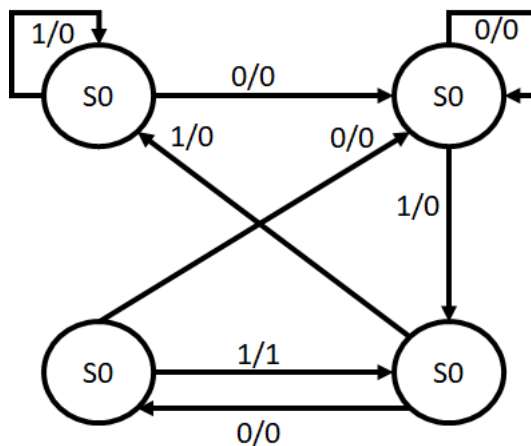
1. (1 punto) Si calcoli il numero di bit di controllo necessari alla codifica della stringa.
2. (3 punti) Codificare la stringa (il bit meno significativo è a sinistra), secondo il codice di Hamming, motivando ogni passaggio intermedio. Scrivere la stringa codificata.
3. (3 punti) Spiegare come viene rilevato e corretto un errore sul primo bit della stringa da trasferire (quella data).

**ESERCIZIO 5 (3 punti)**

Si consideri una CPU con set d'istruzioni a un indirizzo e registro accumulatore ACC, dotata delle seguenti istruzioni: LOAD X ( $M[X] \rightarrow ACC$ ), STORE X ( $ACC \rightarrow M[X]$ ), ADD X ( $ACC + M[X] \rightarrow ACC$ ), DIV X ( $ACC / M[X] \rightarrow ACC$ ), MUL X ( $ACC * M[X] \rightarrow ACC$ ). Scrivere la sequenza d'istruzioni necessaria per implementare l'espressione algebrica:  $Z = A/B + C*B$ .

## ESERCIZIO 1

1. Per prima cosa, occorre scrivere il grafo degli stati che corrisponde al seguente:



Utilizzando FF-T:

A	B	X	A'	DA	B'	JB	Z
0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0
0	1	1	1	1	0	1	0
1	0	0	1	0	1	1	0
1	0	1	0	1	0	0	0
1	1	0	0	1	1	0	0
1	1	1	1	0	0	1	1

Semplificando le espressioni delle rete logiche per la transizione dello stato:

AB	00	01	11	10
X				
0			1	
1		1		1

$$T_A = \bar{A}BX + A\bar{B}X + AB\bar{X}$$

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

$$T_B = \bar{B}\bar{X} + BX$$

L'espressione dell'uscita è invece data da:

$$Z = ABX$$

Si lascia allo studente il disegno del circuito.

2. Vedere le dispense del corso.

## ESERCIZIO 2

1. Essendo la memoria primaria costituita da 32 blocchi composti ciascuno di 4 parole da 32 B, il calcolo è dato da  $32 \cdot 4 \cdot 32 = 2^{12}$  B, con indirizzamento a 12 bit del singolo byte (ne occorrono 7 per la singola parola).
2. Di questi, i sette meno significativi compongono l'offset (i blocchi sono di 4 parole). Il block frame è invece composto da 5 bit (32 blocchi). Infine l'index da 2 bit (4 insiemi). Il TAG è quindi composto da 2 bit.
3. Per indicare i valori di block frame mappati nella linea 0, basta porre appunto a 0 il valore di index e valutare i relativi valori 0, 1, 2, 3, ..., 7 di TAG, che moltiplicati per 4 (numero insiemi di cache) ci dà appunto ciò che cerchiamo: 0, 4, 8, 12, 16, 20, 24, 28.
4. I blocchi 7 e 23, come tutti i blocchi di primaria, presentano 128 byte. A cache vuota, verrà copiato tutto il blocco 7 nella prima linea libera del set 3 e si verificherà un solo miss relativo alla prima parola dello stesso (l'indirizzo è  $7 \cdot 128 = 896$ ). Seguiranno 127 hit. Chiamata la prima parola del blocco 23 (indirizzo 2944) viene trasferito tutto il blocco nello stesso set 3, ma nella seconda linea libera. Anche in questo caso un miss, 127 hit. Un secondo ciclo di chiamate alle stesse parole produce esclusivamente hit per entrambi i blocchi. L'hit ratio è dunque  $254/256 = 0.99$  (circa).
5. La formula del tempo medio di accesso alla gerarchia di memoria data è:

$$\bar{T} = T_C + (1 - H_C) \cdot T_P$$

Sostituendo i valori di  $H_C$  trovati nel precedente esercizio si ha:

$$\bar{T}_{Diretto} = 4 + 0.01 \cdot 40 = 4.4nsec$$

## ESERCIZIO 3

Il blocco da leggere occupa 16 kB ed è costituito da  $214/28 = 26 = 64$  settori, posizionati ciascuno su tracce adiacenti. Letto il primo settore, per gli altri 63 sarà necessario muovere la testina di lettura sulla traccia adiacente (1ms) ed effettuare in media una mezza rotazione per entrare in contatto con il settore da leggere. In altre parole si può applicare la formula:

$$T = \frac{1}{512} \cdot T_R + 63 \cdot \left( \frac{1}{512} \cdot T_R + 0.5 \cdot T_R + 1 \right)$$

Essendo  $T_R = \frac{60}{6000} = 10ms$  il tempo di rotazione, una mezza rotazione necessita 5 ms, si ha  $T = 1.25 + 315 + 63 = 379.25ms$ .

## ESERCIZIO 4

- 1) Deve essere rispettata la condizione:

$$2^K \geq N + K + 1 \quad (1),$$

dove K è il numero di bit di controllo inseriti. Essendo  $N = 8$ , si evince dalla (1) che  $K = 4$ .

Nella codifica di Hamming, la sequenza in uscita deve presentare la seguente struttura:

	$c_0$	$c_1$	$b_0$	$c_2$	$b_1$	$b_2$	$b_3$	$c_3$	$b_4$	$b_5$	$b_6$	$b_7$
			1		1	0	0		1	1	0	0
$e_0$	1		1		1		1		1		1	
$e_1$		1	1			1	1			1	1	
$e_2$				1	1	1	1					1
$e_3$								1	1	1	1	1

Dove  $c_0 \dots c_3$  sono i quattro bit costituenti il vettore di controllo, e  $b_0 \dots b_7$  gli otto bit trasmessi. I quattro bit devono essere tali che il corrispondente vettore di errore  $e_3e_2e_1e_0$  indichi il bit alterato in caso di errore. L'intersezione fra gli uni presenti nel vettore di errore ci dice univocamente quale bit sia stato alterato. Scrivendo per ciascuno dei bit della sequenza intera le configurazioni di errore, otteniamo infatti i valori di parità che ciascun bit di errore deve assumere in corrispondenza di ciascuno dei bit della stringa completa. Calcolando i bit di controllo rispetto alla sequenza  $b_0 \dots b_7$  ricevuta si ottiene:

$c_0 = 1$   
 $c_1 = 0$   
 $c_2 = 1$   
 $c_3 = 0$

La stringa codificata è dunque **10110001100** (bit meno significativo a sinistra).

Un errore sul primo bit utile ( $b_0$ ), ovvero una sua inversione durante il trasferimento, è rilevato attraverso la corrispondente inversione dei bit che lo controllano,  $c_0$  e  $c_1$ . Di conseguenza la stringa di errore generata è pari a 1100 (bit meno significativo a sinistra), derivata dal ricalcolo dei bit di controllo e dal loro confronto con i bit di controllo ricevuti. La configurazione ottenuta è equivalente alla posizione del bit errato, appunto  $b_0$ .

## ESERCIZIO 5

A partire dalla semantica fornita nel testo, una possibile sequenza è:

Istruzione	Semantica
LOAD A	$M[A] \rightarrow ACC$
DIV B	$ACC / M[B] \rightarrow ACC$
STORE Z	$ACC \rightarrow M[Z]$
LOAD C	$M[C] \rightarrow ACC$
MUL B	$ACC * M[B] \rightarrow ACC$
ADD Z	$ACC + M[Z] \rightarrow ACC$
STORE Z	$ACC \rightarrow M[Z]$