

**PRIMA PROVA INTERMEDIA DEL MODULO DI**  
**CALCOLATORI ELETTRONICI**  
**NUOVO E VECCHIO ORDINAMENTO DIDATTICO – A.A. 2014/15 e precedenti**  
22 aprile 2016

**MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI**

NOME: \_\_\_\_\_ COGNOME: \_\_\_\_\_ MATRICOLA: \_\_\_\_\_

**ESERCIZIO 1 (8 punti)**

Progettare un riconoscitore a singolo ingresso delle sequenze 0010 e 0011. Si usino FF-JK per il modulo di memorizzazione dello stato. Semplificare le espressioni delle reti di transizione dello stato con le mappe di Karnaugh. Si calcoli anche la rete logica di transizione dell'uscita.

**ESERCIZIO 2 (8 punti)**

Si consideri una gerarchia di memoria composta da una memoria cache di 512 parole ed una memoria principale di 2048 parole. Le parole sono raggruppate in blocchi da 64.

1. (3 punti) Illustrare come viene interpretato l'indirizzo di una generica parola nel caso in cui si utilizzi il metodo di indirizzamento diretto.
2. (5 punti) Mostrare lo stato finale della memoria e il relativo "cache hit", nel caso in cui la sequenza di chiamate sia la seguente:  
1788 – 1746 – 616 – 459 – 637 – 1198 – 865 – 1952 – 74 – 779 – 1705 – 137.

**ESERCIZIO 3 (7 punti)**

E' data un'unità a disco con i seguenti parametri: 6000 giri/min, 128 tracce, 64 settori per traccia, 4 KB/settore, tempo di posizionamento da una traccia a quella adiacente pari a 1 ms.

Si supponga di dover scrivere un file di 1 MB in modalità frammentata, con settori a due a due nella stessa traccia. I settori nella stessa traccia sono distanti mediamente quattro settori e le tracce distano invece tra loro mediamente 2 tracce.

Calcolare il tempo medio di scrittura se all'istante iniziale la testina è posizionata sul primo settore utile.

**ESERCIZIO 4 (6 punti)**

Durante un trasferimento da memoria a CPU con parole protette mediante codifica di Hamming, la CPU riceve la seguente stringa: 001011011101 (bit meno significativo a sinistra).

- (1) Indicare e spiegare il ruolo, mediante lo schema di principio dei codici a correzione d'errore mostrato a lezione, dei blocchi funzionali che permettono di rilevare e correggere eventuali errori presenti nella stringa ricevuta.
- (2) Applicare lo schema al caso specifico della stringa fornita dal testo, verificando e correggendo un eventuale errore se presente.

**ESERCIZIO 5 (4 punti)**

Implementare, tramite il set di istruzioni a singolo indirizzo fornito in tabella, l'espressione:

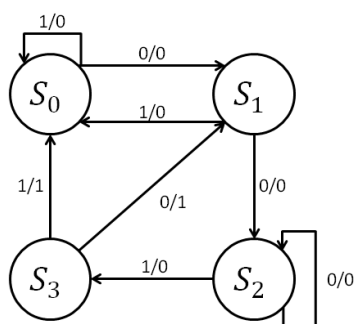
$$Z = A/B + C / (A+B)$$

Istruzione	Semantica
LOAD X	$M[X] \rightarrow ACC$
STORE X	$ACC \rightarrow M[X]$
ADD X	$ACC \leftarrow ACC + M[X]$
DIV X	$ACC \leftarrow ACC / D[X]$

M[X] indica il contenuto della locazione X. Si usino esclusivamente le quattro locazioni A, B, C, Z indicate e si faccia attenzione a non alterare i valori conservati in A, B, C durante l'esecuzione delle istruzioni.

# **ESERCIZIO 1 (8 punti)**

Risposta. Il grafo degli stati è dato da:



Che conduce alla seguente tabella delle transizioni dello stato, posto che ad ogni simbolo di stato si associ una coppia di bit in quest'ordine: S0=00, S1=01, S2=10, S3=11, che nel seguito indichiamo con la coppia AB. Indichiamo con Z l'uscita della rete, e con JA, KA e JB, KB gli ingressi di ciascuno dei flip flop rappresentanti i bit di stato A e B.

A	B	X	A'	JA	KA	B'	JB	KB	Z
0	0	0	0	0	D	1	1	D	0
0	0	1	0	0	D	0	0	D	0
0	1	0	1	1	D	0	D	1	0
0	1	1	0	0	D	0	D	1	0
1	0	0	1	D	0	0	0	D	0
1	0	1	1	D	0	1	1	D	0
1	1	0	0	D	1	1	D	0	1
1	1	1	0	D	1	0	D	1	1

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

$$J_A = B\bar{X}$$

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

$$K_A = B$$

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

$$J_B = \bar{A}\bar{X} + AX$$

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

$$K_B = \bar{A} + X$$

Infine, la rete di transizione dell'uscita è  $Z = AB$ .

**ESERCIZIO 2 (8 punti)**

Risposta 1. Il campo indirizzi avrà ampiezza 11 bit (devo indirizzare 2048 parole). Avremo 32 blocchi in primaria e 8 in cache.

< TAG 2 bit > < Cache Index 3 bit > < Offset 6 bit >

Risposta 2.

	1788	1746	616	459	637	1198	865	1952	74	779	1705	1213
B.F.	27	27	9	7	9	18	13	30	1	12	26	2
C.I.	3	3	1	7	1	2	5	6	1	4	2	2
Hit		X			X							

Blocco 0	Blocco 1	Blocco 2	Blocco 3	Blocco 4	Blocco 5	Blocco 6	Blocco 7
	576...639	1152...1215	1728...1792	768...831	832...895	1920...1983	448...511
	64...127	1664...1727					
		1152...1215					
-	64...127	1152...1215	1728...1792	768...831	832...895	1920...1983	448...511

**ESERCIZIO 3 (7 punti)**

I parametri del disco sono:

$T_{rot} = 10 \text{ ms}$ ;  $T_{pos} = 1 \text{ ms}$ ;  $T_{lettsett} = T_{rot}/64 = 0.16 \text{ ms}$  (circa)

Poiché il file è da 1 MB =  $2^{10}$  KB, ciò implica la necessità di un numero di settori pari a 256, memorizzati a due a due su tracce diverse, quindi 128 tracce impegnate in tutto.

I passaggi sono:

- 1) Scrittura primo settore:  $T_{lettsett}$
- 2) Latenza verso secondo settore da scrivere:  $4 * T_{lettsett}$
- 3) Scrittura secondo settore:  $T_{lettsett}$

Sulle restanti 127 tracce, abbiamo:

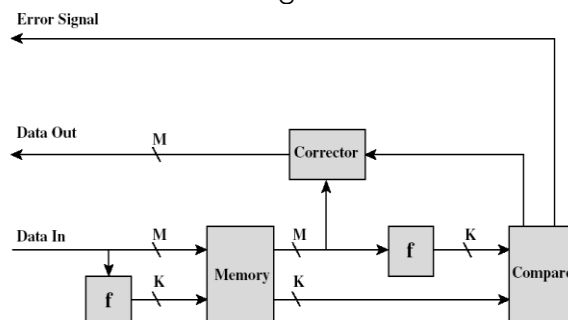
- 1) Posizionamento sulla traccia :  $2 * T_{pos}$
- 2) Latenza verso primo settore utile:  $T_{rot}/2$
- 3) Scrittura settore:  $T_{lettsett}$
- 4) Latenza verso secondo settore:  $4 * T_{lettsett}$
- 5) Scrittura settore:  $T_{lettsett}$

La formula è dunque:

$T = 6 * T_{lettsett} + 127 * (6 * T_{lettsett} + 2 * T_{pos} + T_{rot}/2) = 6 * 0.16 + 127 * (6 * 0.16 + 2 + 5) = 0.96 + 121.92 + 762 = 884.88 \text{ ms}$ .

#### ESERCIZIO 4 (6 punti)

Risposta 1. Lo schema presentato a lezione è il seguente:



Come si evince dallo schema, la stringa codificata in ingresso (Data In), viene partizionata nel sottoinsieme dei bit di controllo (blocco 'f' in basso a sinistra) e dei bit di informazione. Dopo essere stati depositati in memoria, i bit di informazione vengono utilizzati per ricalcolare i bit di controllo (blocco 'f' a destra) che vengono confrontati con quelli ricevuti: il comparatore emette un segnale di errore (ovvero un bit pari a 0 o ad 1 a seconda che non ci sia o ci sia errore) e contemporaneamente il codice di errore relativo che diviene ingresso, assieme alla stringa di informazione, del modulo di correzione (Corrector), in uscita del quale abbiamo la stringa corretta (Data Out).

Risposta 2. Nel caso fornito dalla stringa in esame, i passi sono:

1. Separazione della stringa di informazione dai bit di controllo, ottenendo i sottoinsiemi:  $M = \{11101101\}$ , stringa di informazione e  $K = \{0011\}$ , stringa di controllo.
2. Ricalcolo della stringa di controllo  $K' = \{1111\}$ . L'ottenimento della stringa di controllo  $K'$  deriva dalle formule che prevedono la seguente associazione tra bit di controllo e bit di informazione:

$$k'_0 = m_0 \oplus m_1 \oplus m_3 \oplus m_4 \oplus m_6 = 1$$

$$k'_1 = m_0 \oplus m_2 \oplus m_3 \oplus m_5 \oplus m_6 = 1$$

$$k'_2 = m_1 \oplus m_2 \oplus m_3 \oplus m_7 = 1$$

$$k'_3 = m_4 \oplus m_5 \oplus m_6 \oplus m_7 = 1$$

3. Lo XOR bit a bit di  $K'$  e  $K$  fornisce la configurazione di errore  $e = \{1100\}$  che è ingresso del modulo di correzione. Essa impone la correzione del primo bit della stringa di informazione, ed il fatto che sia diversa da  $\{0000\}$  consente la generazione del bit di segnalazione d'errore.

Per ottenere questo risultato, il sistema ovviamente conosce già il numero di bit controllo ed il numero di bit di informazione che si ottengono mediante la formula:  $2^K \geq N+K+1$ , dove con  $N$  e  $K$  abbiamo qui indicato il numero di bit di controllo ( $K$ ) e il numero di bit di informazione ( $N$ ).

Inoltre la loro posizione è fissa e nota al sistema grazie all'associazione progressiva (in neretto il bit di controllo di esempio  $k_0$  controlla i bit di informazione associati alla sua parità nella stringa di errore:  $m_0, m_1, m_3, m_4, m_6$  - similmente per gli altri):

<b>1</b>	0001	<b>k0</b>
2	0010	k1
<b>3</b>	0011	<b>m0</b>
4	0100	k2
<b>5</b>	0101	<b>m1</b>
6	0110	m2
<b>7</b>	0111	<b>m3</b>
8	1000	k3
<b>9</b>	1001	<b>m4</b>
10	1010	m5
<b>11</b>	1011	<b>m6</b>
12	1100	m7

**ESERCIZIO 5 (4 punti)**

La sequenza è:

Istruzione	Semantica	Contenuto dell'accumulatore
LOAD A	$M[A] \rightarrow ACC$	$M[A]$
ADD B	$ACC \leftarrow ACC + M[B]$	$M[B] + M[A]$
STORE Z	$M[Z] \leftarrow ACC$	$M[B] + M[A]$
LOAD C	$M[C] \rightarrow ACC$	$M[C]$
DIV Z	$ACC \leftarrow ACC / M[Z]$	$M[C] / (M[B] + M[A])$
STORE Z	$M[Z] \leftarrow ACC$	$M[C] / (M[B] + M[A])$
LOAD A	$M[A] \rightarrow ACC$	$M[A]$
DIV B	$ACC \leftarrow ACC / M[B]$	$M[A] / M[B]$
ADD Z	$ACC \leftarrow ACC + M[Z]$	$M[A] / M[B] + M[C] / (M[B] + M[A])$
STORE Z	$M[Z] \leftarrow ACC$	$M[A] / M[B] + M[C] / (M[B] + M[A])$