

**PRIMA PROVA INTERMEDIA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO**

18 Aprile 2008

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

ESERCIZIO 1 (8 punti)

Si consideri una rete logica sequenziale avente un ingresso X e una uscita Z . L'uscita deve essere $Z = 1$ quando viene riconosciuta la sottosequenza 0100. Negli altri casi l'uscita deve essere $Z = 0$.

1. (3 punti) Disegnare il diagramma degli stati.
2. (2 punti) Codificare gli stati e scrivere la tabella di flusso. Si scriva poi la tabella delle transizioni qualora si usino flip-flop di tipo JK.
3. (3 punti) Calcolare le forme minime per le variabili di eccitazione dei flip-flop e per l'uscita impiegando le mappe di Karnaugh.

Soluzione

Grafo degli stati

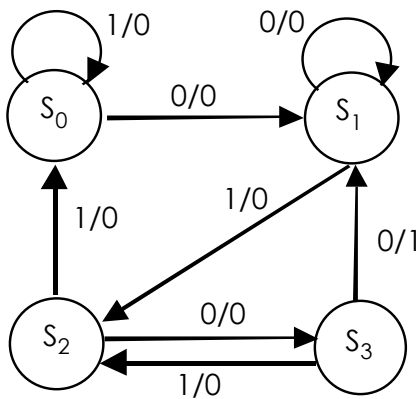


Tabella di flusso

Stato iniziale	Stato finale/uscita	
	$x = 0$	$x = 1$
S_0	$S_1/0$	$S_0/0$
S_1	$S_1/0$	$S_2/0$
S_2	$S_3/0$	$S_0/0$
S_3	$S_1/1$	$S_2/0$

Codifica degli stati
(2 bit - Y_1Y_0)

$S_0 \rightarrow 00$ $S_1 \rightarrow 01$ $S_2 \rightarrow 10$ $S_3 \rightarrow 11$

Tabella delle transizioni (FF-JK)

Y' indica lo stato futuro

Y_1	Y_0	x	Y'_1	J_1	K_1	Y'_0	J_0	K_0	z
0	0	0	0	0	d	1	1	d	0
0	0	1	0	0	d	0	0	d	0
0	1	0	0	0	d	1	d	0	0
0	1	1	1	1	d	0	d	1	0
1	0	0	1	d	0	1	1	d	0
1	0	1	0	d	1	0	0	d	0
1	1	0	0	d	1	1	d	0	1
1	1	1	1	d	0	0	d	1	0

Tabella di eccitazione
di un flip-flop JK

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

Uscita: $z = Y_1 Y_0 \bar{x}$

$J_1 = Y_0 x$		$K_1 = \bar{Y}_0 x + Y_0 \bar{x}$	
$J_0 = \bar{x}$		$K_0 = x$	

ESERCIZIO 2 (7 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificati utilizzando il codice di Hamming. Si consideri la parola di 10 bit 1010011101 (il bit meno significativo è a sinistra). **Spiegando bene ogni passo del ragionamento:**

1. (2 punti) calcolare il minimo numero di bit di controllo necessari per la codifica della parola;
2. (3 punti) codificare la parola data;
3. (2 punti) imporre un errore nel terzo e nel quarto bit della parola **inizialmente data**. **Applicare il procedimento di rilevazione di eventuali errori previsto** nella codifica di Hamming **e** spiegare se gli errori vengono rivelati e se possono essere corretti.

Soluzione.

- 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 $N = 10$, si evince dalla (1) che il numero minimo di bit di controllo richiesto è 4.

- 2) Per codificare la sequenza in ingresso occorre calcolare:

$$c_0 = b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 \oplus b_8 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$c_1 = b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 \oplus b_9 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$c_2 = b_1 \oplus b_2 \oplus b_3 \oplus b_7 \oplus b_8 \oplus b_9 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$c_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus b_8 \oplus b_9 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0$$

1	2	3	4	5	6	7	8	9	10	11	12	13	14
c_0	c_1	b_0	c_2	b_1	b_2	b_3	c_3	b_4	b_5	b_6	b_7	b_8	b_9
0	1	1	1	0	1	0	0	0	1	1	1	0	1

Dove $c_0 \dots c_3$ sono i 4 bit costituenti il vettore di controllo, e $b_0 \dots b_9$ i 10 bit trasmessi. La sequenza codificata è dunque 01110100011101.

- 3) Imponiamo un errore sul terzo e sul quarto bit della parola inizialmente data (b_2 e b_3). La stringa risultante sarà:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
c_0	c_1	b_0	c_2	b_1	b_2	b_3	c_3	b_4	b_5	b_6	b_7	b_8	b_9
0	1	1	1	0	0	1	0	0	1	1	1	0	1

Per verificare la presenza di un errore, dobbiamo ricalcolare il vettore di controllo a partire dalla sequenza ricevuta. Si ha:

$$c'_0 = b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 \oplus b_8 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$c'_1 = b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 \oplus b_9 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$c'_2 = b_1 \oplus b_2 \oplus b_3 \oplus b_7 \oplus b_8 \oplus b_9 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$c'_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus b_8 \oplus b_9 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0$$

Il passo successivo è calcolare il vettore di errore come Exor dei vettori di controllo c e c' :

$$e_0 = c_0 \oplus c'_0 = 1$$

$$e_1 = c_1 \oplus c'_1 = 0$$

$$e_2 = c_2 \oplus c'_2 = 0$$

$$e_3 = c_3 \oplus c'_3 = 0$$

Poiché il vettore risultante 0001 non è nullo, viene rilevato un errore nella stringa di 14 bit e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato risulterebbe quindi il primo bit (c_0).

Ovviamente questo è un limite della codifica di Hamming, perché viene erroneamente rilevato un errore in c_0 quando invece si erano verificati 2 errori su bit diversi (b_2 e b_3).

ESERCIZIO 3 (10 punti)

- (3 punti) Si considerino una memoria primaria di dimensione pari a 256 byte e una memoria cache da 64 byte, indirizzabili al singolo byte. La dimensione dei blocchi che compongono le memorie è di 8 byte. Si mostri il formato logico dell'indirizzo nel caso in cui la cache sia gestita con il metodo diretto, completamente associativo oppure associativo con insiemi a 4 vie.
- (4 punti) Si considerino le seguenti chiamate ai singoli byte (indirizzi espressi in decimale): 65, 72, 18, 24, 81, 136, 82, 137, 49, 50. Si indichi il contenuto della cache, ovvero quali byte occupano i relativi blocchi di cache, dopo l'ultima chiamata, e il valore dell'hit ratio, nel caso in cui:
 - la cache sia gestita con il metodo diretto;
 - la cache sia gestita con il metodo associativo con insiemi a 4 vie, con strategia di rimpiazzamento LRU.
- (2 punti) Si consideri una gerarchia di memoria a tre livelli, costituita da cache, primaria e disco, che ha un tempo di accesso medio pari a 1 ms. E' noto il tempo di accesso al disco, pari a 5 ms, e si considerino trascurabili (rispetto al tempo di accesso al disco) i tempi di accesso in primaria e in cache ($T_c = T_p = 0$). La probabilità di trovare una parola in memoria primaria quando non è presente in cache (hit ratio condizionale) è pari a 0.7. Si stimi il valore dell'hit ratio per la cache presente in questa gerarchia.
- (1 punto) Si spieghi chiaramente perché le parole in memoria vengono accorpate in blocchi, e perché si trasferiscono, ad esempio dalla memoria primaria alla cache, interi blocchi di parole e non, volta per volta, le singole parole richieste.

Soluzione

- Visto che la memoria primaria è di 256 B e indirizzabile al singolo byte, l'ampiezza del campo indirizzi sarà di 8 bit. Il campo offset è composto di 3 bit, visto che si devono indirizzare 8 B/blocco.

Quindi: <Block Frame 5 bit><Offset 3 bit>

Per il metodo diretto occorre calcolare il numero di blocchi presenti in cache, dato da $64 \text{ B} / (8 \text{ B} / \text{blocco}) = 8$ blocchi. Quindi il cache index sarà composto di 3 bit, e il formato dell'indirizzo sarà: <TAG 2 bit><C.I. 3 bit><Offset 3 bit>

Nel caso di indirizzamento completamente associativo, ricordiamo TAG = Block frame, quindi banalmente: <TAG 5 bit><Offset 3 bit>

Infine, nel caso set-associativo con 4 blocchi/insieme, occorre calcolare il numero di insiemi in cache, da indirizzare con il campo set index. Il numero di insiemi in cache è dato dal numero di blocchi in cache diviso il numero di blocchi / insieme, ovvero: $8 \text{ blocchi} / (4 \text{ blocchi/insieme}) = 2$ insiemi. Quindi: <TAG 4 bit><S.I. 1 bit><Offset 3 bit>

- Metodo diretto: $BF = \text{int}(X/8)$, $CI = \text{mod}(BF/8)$

X	65	72	18	24	81	136	82	137	49	50
B.F.	8	9	2	3	10	17	10	17	6	6
C.I.	0	1	2	3	2	1	2	1	6	6
Blocchi cache										
0	8									
1		9				17		-		
2			2		10		-			
3				3						
4										
5										
6									6	-
7										
Hit							X	X		X

Hit rate = 3/10

Stato finale della cache

Blocchi	BF	Words
0	8	64, 65, 66, 67, 68, 69, 70, 71
1	17	136, 137, 138, 139, 140, 141, 142, 143
2	10	80, 81, 82, 83, 84, 85, 86, 87
3	3	24, 25, 26, 27, 28, 29, 30, 31
4		
5		
6	6	48, 49, 50, 51, 52, 53, 54, 55
7		

Metodo set-associativo: $BF = \text{int}(X/8)$, $SI = \text{mod}(BF/2)$

X	65	72	18	24	81	136	82	137	49	50
B.F.	8	9	2	3	10	17	10	17	6	6
S.I.	0	1	0	1	0	1	0	1	0	0
Blocchi cache										
Set 0	8									
			2							
					10		-			
									6	-
Set 1		9								
				3						
						17		-		
Hit							X	X		X

Hit rate = 3/10

Stato finale della cache

Set	BF	Words
Set 0	8	64, 65, 66, 67, 68, 69, 70, 71
	2	16, 17, 18, 19, 20, 21, 22, 23
	10	80, 81, 82, 83, 84, 85, 86, 87
	6	48, 49, 50, 51, 52, 53, 54, 55
Set 1	9	72, 73, 74, 75, 76, 77, 78, 79
	3	24, 25, 26, 27, 28, 29, 30, 31
	17	136, 137, 138, 139, 140, 141, 142, 143

3. $T = 1 \text{ ms}$; $T_d = 5 \text{ ms}$;

$$H_{P|\bar{C}} = 0.7$$

$$H_{P|\bar{C}} = \frac{H_P - H_C}{1 - H_C}$$

$$H_P = H_C + (1 - H_C)H_{P|\bar{C}}$$

Occorre scrivere la formula per il calcolo del tempo medio di accesso alla gerarchia ed esplicitarla in funzione di H_C :

$$\bar{T} = T_C + (1 - H_C)T_P + (1 - H_P)T_D \approx (1 - H_P)T_D$$

$$H_P = H_C + (1 - H_C)H_{P|\bar{C}}$$

$$\bar{T} = (1 - H_P)T_D$$

$$H_P = 1 - \frac{\bar{T}}{T_D} = 0.8$$

$$H_C + (1 - H_C)H_{P|\bar{C}} = 0.8$$

$$H_C - 0.7H_C = 0.8 - 0.7$$

$$H_C = 0.33$$

4. La memoria è gestita in blocchi di parole per sfruttare il principio di località, in cui si assume che sia molto probabile che si richiedano in modo sequenziale (località temporale) locazioni adiacenti di memoria (località spaziale). Per cui, per velocizzare il tempo di accesso alla gerarchia, conviene trasferire direttamente in cache (o comunque a livelli più alti e più rapidi della gerarchia di memorie) blocchi di parole memorizzate consecutivamente.

ESERCIZIO 4 (8 punti)

1. (6 punti) Un disco presenta le seguenti caratteristiche: 6000 giri/min, 10 settori per traccia, tempo di posizionamento da una traccia a quella adiacente 2 ms, 64 B per settore. Calcolare il tempo medio di lettura di un blocco da 1 KB da disco, nell'ipotesi che i settori da trasferire si trovino, mediamente, uno ogni due tracce. Si supponga inoltre che la testina sia posizionata sul primo settore da trasferire.
2. (2 punti) Calcolare di quanto si ridurrebbe il tempo di trasferimento del blocco da 1 KB, qualora i settori fossero disposti lungo le tracce in modo da poter trasferire il blocco con la massima velocità.

Soluzione

1. Numero di settori da trasferire: $1 \text{ KB} / (64 \text{ B/settore}) = 16$ settori

$$T_{\text{rot}} = 60 / 6000 \text{ s} = 10 \text{ ms}$$

$$T_{\text{lat}} = T_{\text{rot}} / 2 = 5 \text{ ms}$$

$$T_{\text{pos}} = T_{\text{sp}} * 2 = 2 \text{ ms} * 2 = 4 \text{ ms}$$

$$T_{\text{lett}} = T_{\text{rot}} / 10 = 1 \text{ ms}$$

$$T = T_{\text{lett}} + 15 * (T_{\text{lat}} + T_{\text{pos}} + T_{\text{lett}}) = 1 + 15 * (5 + 4 + 1) \text{ ms} = 151 \text{ ms}$$

2. Caso migliore:

Tutti i settori sono contigui, ma saranno contenuti su due tracce adiacenti (sono 16 settori da trasferire, ma posso posizionarne al massimo 10 a traccia), quindi dovrò includere un tempo di spostamento:

$$T = 16 * T_{\text{lett}} + T_{\text{sp}} = 16 + 2 \text{ ms} = 18 \text{ ms}$$

Il tempo si riduce da 151 ms a 18 ms, quindi dell'88% ($1 - 18/151 = 0.88$)