

**SOLUZIONI DEL PRIMO COMPITINO DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO
11 Maggio 2002**

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

ESERCIZIO 1 (8 punti)

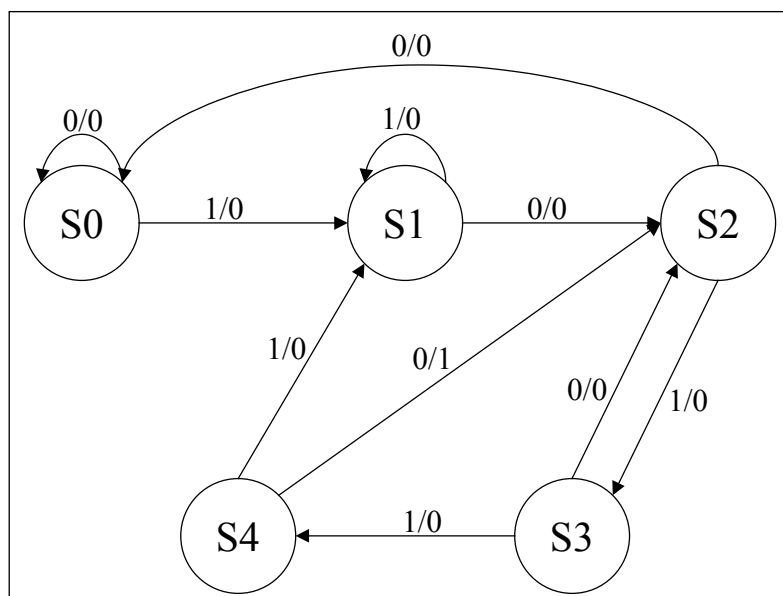
Progettare una rete sequenziale che presenti un ingresso X e un'uscita Z posta a 1 ogni volta che viene riconosciuta la sequenza 10110.

Si richiede:

- il diagramma degli stati, la tabella di flusso e la tabella delle transizioni;
- il calcolo delle forme minime delle variabili di eccitazione dei flip flop con le mappe di Karnaugh. Si usino flip flop T.

Soluzione.

Il diagramma degli stati è il seguente:



La tabella di flusso è data da:

Stato presente	Stato successivo/Uscita	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S3/0
S3	S2/0	S4/0
S4	S2/1	S1/0

Per codificare 5 stati occorrono tre flip flop. La codifica è la seguente:

S0 → 0 0 0; ...; S4 → 1 0 0. Nel seguito indicheremo ciascun bit della codifica con le lettere A, B, C. L'apice indicherà il bit nell'istante successivo a quello considerato.

A partire dalla tabella di eccitazione del flip flop T:

Q	Q'	T
0	0	0
0	1	1
1	0	1
1	1	0

			A'	T _A	B'	T _B	C'	T _C	Z	A'	T _A	B'	T _B	C'	T _C	Z
A	B	C	X=0							X=1						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	1	1	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0
0	1	1	0	0	1	0	0	1	0	1	1	0	1	0	1	0
1	0	0	0	1	1	1	0	0	1	0	1	0	0	1	1	0
1	0	1	D	D	D	D	D	D	0	D	D	D	D	D	D	0
1	1	0	D	D	D	D	D	D	0	D	D	D	D	D	D	0
1	1	1	D	D	D	D	D	D	0	D	D	D	D	D	D	0

Ora possiamo disegnare le mappe di Karnaugh

AB		00	01	11	10
CX	00			d	1
	01			d	1
	11		1	d	d
	10			d	d

$$T_A = A + BCX$$

AB		00	01	11	10
CX	00		1	d	1
	01			d	
	11		1	d	d
	10	1		d	d

$$T_B = A\bar{X} + B\bar{C} \cdot \bar{X} + BCX + \bar{B}C\bar{X}$$

		AB			
		00	01	11	10
CX	00			d	
	01	1	1	d	1
	11		1	d	d
	10	1	1	d	d

$$T_c = \overline{C}X + C\overline{X} + BX$$

Infine, per quanto riguarda l'uscita Z:

$$Z = A \cdot \overline{B} \cdot \overline{C} \cdot \overline{X}$$

Volendo utilizzare anche i don't care:

		AB			
		00	01	11	10
CX	00			d	1
	01			d	
	11			d	d
	10			d	d

$$Z = A\overline{X}$$

ESERCIZIO 2 (7 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa di 8 bit 10101110 (il bit meno significativo è a sinistra). **Spiegando bene ogni passo del ragionamento:**

- 1) (1 punto) calcolare il minimo numero di bit di controllo necessari per la codifica della parola;
- 2) (3 punti) codificare la stringa data;
- 3) (3 punti) imporre un errore nel terzo bit della stringa data e spiegare come l'errore viene rivelato e corretto per mezzo della codifica di Hamming.

Soluzione.

- 1) Deve venire rispettata la condizione:

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

dove K è il numero di bit di controllo inseriti. Essendo N=8, il numero minimo di bit di controllo richiesto è 4.

- 2) Nella codifica di Hamming, la sequenza in ingresso presenta 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		0	1	0		1	1	1	0

Dove c₀...c₃ sono i quattro bit costituenti il vettore di controllo, e b₀...b₇ gli otto bit trasmessi. Tali bit si ottengono con le seguenti operazioni

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

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

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

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

La stringa codificata è 101101011110.

- 3) Nell'ipotesi di un errore sul terzo bit della stringa iniziale, la stringa ricevuta risulta: 101100011110. Per rivelare questo errore, bisogna ricalcolare i bit di controllo:

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

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

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

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

Il passo successivo è calcolare il vettore di errore dato dalla differenza dei vettori di controllo c e c' (ricordiamo che somma e differenza tra bit producono lo stesso risultato):

$$e_0 = c_0 \oplus c'_0 = 0$$

$$e_1 = c_1 \oplus c'_1 = 1$$

$$e_2 = c_2 \oplus c'_2 = 1$$

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

Poiché il vettore risultante 0110 non è nullo, vi è un errore nella stringa di 12 bit e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato nella stringa codificata è quindi il sesto (b₂), che può venire dunque corretto.

ESERCIZIO 3 (8 punti)

Si abbia a disposizione un campo di 25 bit per la rappresentazione dei numeri nei seguenti formati:

- (1 punto) Numeri interi senza segno.
- (3 punti) Numeri reali in virgola fissa rappresentati in segno e valore con 13 bit per la parte intera (compreso il segno).
- (4 punti) Numeri in virgola mobile con 8 bit per rappresentare l'esponente in eccesso 128 e la mantissa, frazionaria e normalizzata (1.M) rappresentata in segno e valore.

Calcolare il minimo e il massimo numero positivo rappresentabile escluso lo zero, per i formati dati.

Soluzione.

- 1)

Formato a)

Minimo: 1.

Massimo: $2^{25}-1$.

Corrispondenti alle configurazioni:

Minima configurazione:																								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Massima configurazione:																								
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Formato b)

Minimo: 2^{-12} .

Massimo: $2^{12} \cdot 1 + 1 \cdot 2^{-12} = 2^{12} \cdot 2^{-12}$.

Corrispondenti alle configurazioni:

S	Parte Intera										Parte frazionaria										
Minima configurazione:																					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Massima configurazione:																					
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Formato c)

Minimo: 2^{-128} .

Massimo: $2^{127} * (2-2^{-16})$.

Corrispondenti alle configurazioni:

S	Esponente										Mantissa																			
Minima configurazione:																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Massima configurazione:																														
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

ESERCIZIO 4 (10 punti)

Si consideri una gerarchia di memoria a tre livelli costituita da: cache, primaria e disco.

- Il disco presenta le seguenti caratteristiche: 7200 giri/min, 120 settori per traccia, tempo medio di posizionamento 2 ms, 256 B per settore.
- L'indirizzamento di una parola della memoria primaria è a 24 bit.
- La memoria cache è di 1024 KB, è suddivisa in blocchi da 512 B (ovvero 512 parole), ed è indirizzata secondo il metodo diretto.

Si richiede:

- (3 punti) Il calcolo del tempo medio di lettura di un blocco di 512 B da disco, nell'ipotesi che la testina si trovi in un punto qualsiasi del disco all'istante iniziale e che il blocco sia registrato su settori contigui della stessa traccia.
- (2 punti) Il partizionamento dell'indirizzo di primaria secondo il metodo di indirizzamento suddetto.
- (3 punti) La stima del "hit ratio" di cache, considerando la cache vuota all'istante iniziale e che vengano prodotte le seguenti richieste di accesso espresse in esadecimale:

21A5BC
21A4F1
F98018
21A492
21A5AA
F9815B
21A5B6
F981EA

- (2 punti) Il calcolo del tempo medio di accesso alla gerarchia, in nanosecondi, utilizzando i valori trovati al punto 1 (per il tempo di accesso al disco) e al punto 3 (per l' "hit ratio" di cache), e considerando che l' "hit ratio" di primaria sia pari a 0.9, il tempo di accesso in cache sia pari a 4 ns e quello in primaria pari a 50 ns.

Soluzione.

- E' sufficiente valutare il tempo di latenza e il tempo di posizionamento una volta sola, in quanto il blocco è registrato su 2 settori contigui.

$$TROT = 60 / 7200 = 0.0083 \text{ secondi}$$

$$TLAT = TROT / 2 = 0.00415 \text{ secondi}$$

$$Tlett = TROT / 120 = 0.0694 \text{ ms}$$

$$\text{Tempo di lettura di un blocco} = TLAT + TPOS + 2 * Tlett = 4.15 + 2 + 2 * 0.0694 = 6.2888 \text{ ms}$$

- La cache è formata da 2^{20} parole, suddivise in blocchi da 2^9 parole, pertanto il numero di blocchi in cache è 2^{11} , indirizzabili con 11 bit. Per indirizzare le parole di ogni blocco occorrono invece 9 bit. Il partizionamento è il seguente:

$$< \text{TAG } 4 \text{ bit} > < \text{Cache Index } 11 \text{ bit} > < \text{Offset } 9 \text{ bit} >$$

- Per stimare l' "hit ratio" di cache, è sufficiente eseguire il rapporto fra numero di "hit" e numero totale di accessi. Se all'istante iniziale la cache è vuota, la sequenza di richieste produce i risultati in tabella. Indichiamo ciascuna cifra esadecimale con il relativo quartetto di bit evidenziando, in neretto, il campo "cache index":

Parola	$a_3a_2a_1a_0$	$b_3b_2b_1b_0$	$c_3c_2c_1c_0$	$d_3d_2d_1d_0$	$e_3e_2e_1e_0$	$f_3f_2f_1f_0$	Hit?
21A5BC	2	0001	1010	0101	B	C	
21A4F1	2	0001	1010	0100	F	1	Sì
F98018	F	1001	1000	0000	1	8	
21A492	2	0001	1010	0100	9	2	Sì
21A5AA	2	0001	1010	0101	A	A	Sì
F9815B	F	1001	1000	0001	5	B	Sì
21A5B6	2	0001	1010	0101	B	6	Sì
F981EA	F	1001	1000	0001	E	A	Sì

$$L' \text{ "hit ratio" di cache stimato è pari a: } 6/8 = 0.75.$$

- Con tutti i dati a nostra disposizione è sufficiente valutare la formula:

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

Quindi:

$$\bar{T} = 0.75 \cdot 4 + 0.15 \cdot 54 + 0.1 \cdot 6288854 = 628896.5 \text{ ns}$$