

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO**
14 Giugno 2002

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

ESERCIZIO 1 (NO: 9 punti – VO: 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 di sei bit 100101.

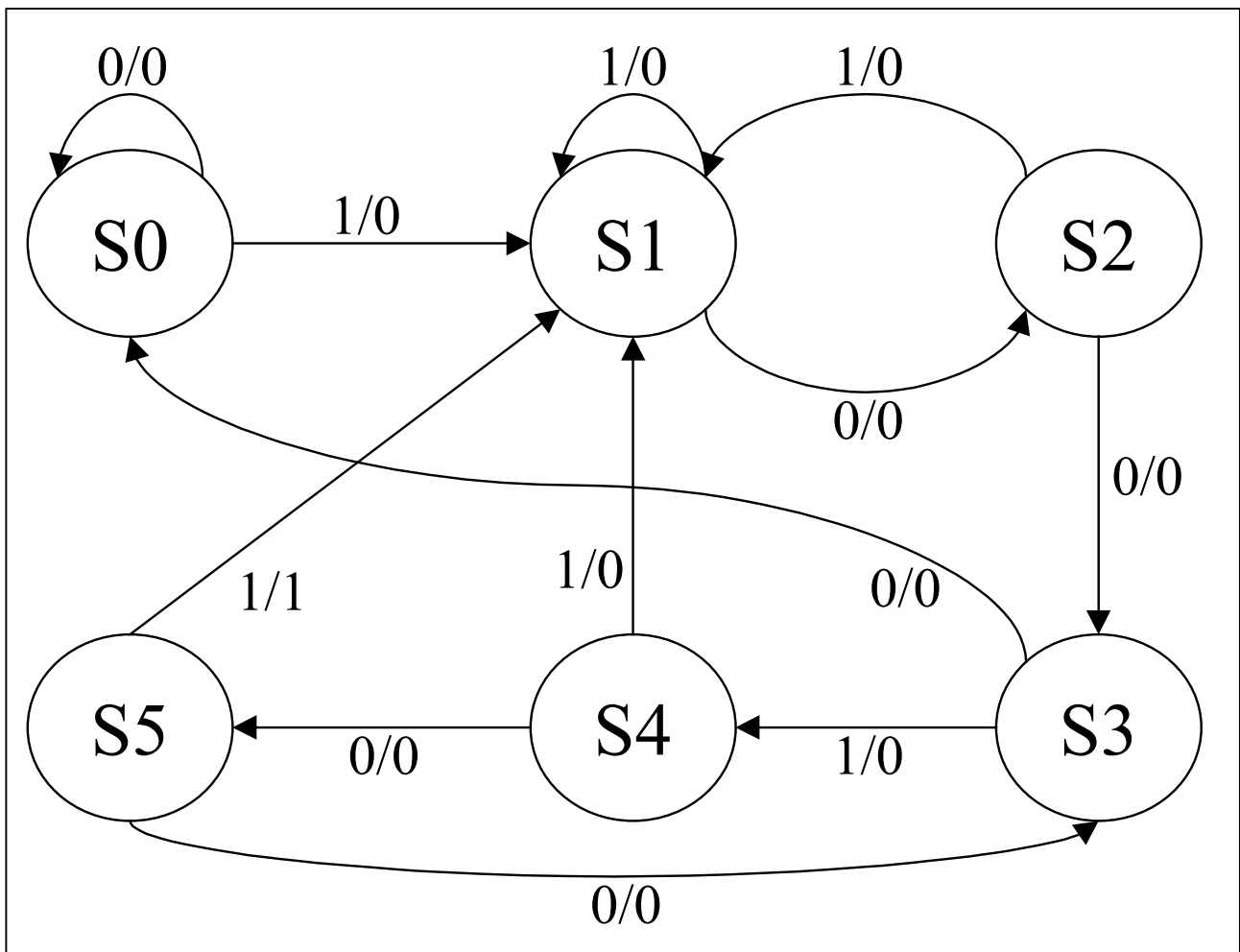
1. (VO: 6 punti) Si richiede:

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

2. (solo VO: 2 punti) Quanti sono i possibili grafi degli stati minimi (quindi senza stati ridondanti) riconoscitori di sequenze di sei bit ?

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	S3/0	S1/0
S3	S0/0	S4/0
S4	S5/0	S1/0
S5	S3/0	S1/1

Per codificare 6 stati occorrono tre flip flop. La codifica è la seguente:
 $S0 \rightarrow 000$; ...; $S5 \rightarrow 101$. 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'	J	K
0	0	0	D
0	1	1	D
1	0	D	1
1	1	D	0

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

Ora possiamo disegnare le mappe di Karnaugh

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

$$J_A = BCX$$

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

$$K_A = X + C$$

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

$$J_B = C\bar{X}$$

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

$$K_B = X + C$$

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

$$J_C = A + B + X$$

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

$$K_C = B + \bar{A} \cdot \bar{X}$$

Infine, per quanto riguarda l'uscita Z:

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

Volendo utilizzare anche i don't care:

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

$$Z = ACX$$

Risposta n.2 VO. Il numero di grafi degli stati minimi si può calcolare considerando il numero possibile di sequenze di 6 bit (26) e dividendo tale numero per due. Infatti, dato il grafo per il riconoscimento di una certa stringa binaria, il grafo che riconosce la stringa complementare è esattamente lo stesso. Per cui il numero totale di grafi è pari a 32.

ESERCIZIO 2 (NO: 8 punti – VO: 6 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 11110110 risultata dalla codifica con il codice di Hamming (il bit meno significativo è a sinistra). **Spiegando bene ogni passo del ragionamento:**

- 1) (NO: 3 punti – VO: 2 punti) calcolare la lunghezza della stringa originale, sapendo che è stato fatto uso del minimo numero di bit di controllo per la codifica;
- 2) (NO: 5 punti - VO: 4 punti) verificare la presenza di un errore nel codice 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 $N+K=8$, il numero minimo di bit di controllo richiesto è 4. Quindi la stringa originale è formata da $N=8-K=4$ bit.

- 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
1	1	1	1	0	1	1	0

Dove $c_0 \dots c_3$ sono i quattro bit costituenti il vettore di controllo, e $b_0 \dots b_3$ gli otto bit trasmessi. Per verificare la presenza di un errore è necessario calcolare la

parità fra i bit di controllo e i relativi bit controllati (in questo caso c_3 è ininfluyente):

$$e_0 = c_0 \oplus b_0 \oplus b_1 \oplus b_3 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$e_1 = c_1 \oplus b_0 \oplus b_2 \oplus b_3 = 1 \oplus 1 \oplus 1 \oplus 1 = 0$$

$$e_2 = c_2 \oplus b_1 \oplus b_2 \oplus b_3 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

Poiché il vettore risultante 101 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 quinto (b_1), che può venire dunque corretto. La stringa originale è dunque 1111.

ESERCIZIO 3 (NO: 8 punti – VO: 7 punti)

Si consideri un calcolatore che dispone di una memoria principale di 128 Kbyte e di una memoria cache di 4 Kbyte. E' possibile accedere al singolo byte e la memoria è suddivisa in blocchi da 64 byte.

- 1) (2 punti) Spiegare come vengono interpretati gli indirizzi di memoria primaria per recuperare l'informazione contenuta nella cache nel caso venga usata la modalità di indirizzamento
 - Diretto
 - "associativo su insiemi", e ciascun insieme contenga quattro blocchi
- 2) (NO: 6 punti – VO: 5 punti) Si consideri la cache di cui alla domanda precedente indirizzata con la modalità "associativa su insiemi" con insiemi di quattro blocchi. Ipotizzare che il processore acceda ai byte di indirizzo 0, 1, 2, ..., 4095 in questo ordine. Si ipotizzi inoltre che la cache sia inizialmente vuota. Mostrare lo stato finale della memoria nel caso si utilizzino le due tecniche sopracitate. (4 punti)

Soluzione:

- 1) Per indirizzare 32 Kbyte occorre un indirizzo di almeno 15 bit. Per indirizzare il singolo byte all'interno di un blocco occorrono 6 bit ($64 = 2^6$), che coincidono con i 6 bit meno significativi dell'indirizzo di memoria primaria. I restanti 9 bit costituiscono l'indirizzo del "block frame". Per indirizzare la cache, il "block frame" viene interpretato diversamente a seconda che l'indirizzamento sia di tipo "diretto" o "associativo su insiemi".
- 2) Indirizzamento diretto. In questo caso devo poter indirizzare ciascuno dei 64 blocchi contenuti nella cache ($4\text{Kbyte}/(64\text{byte}/\text{blocco})$). Occorrono 6 bit che coincidono con i 6 bit meno significativi del "block frame" Pertanto i 15 bit di indirizzo della memoria primaria vengono interpretati come:

tag	cache index	offset
5 bit	6 bit	6 bit

- 3) Indirizzamento "associativo su insiemi". In questo caso devo poter indirizzare ciascuno dei 16 insiemi in cui sono suddivisi i blocchi contenuti nella cache ($\frac{4\text{Kbyte}}{4\text{blocchi} \cdot \frac{64\text{byte}}{\text{insieme blocco}}} = 16 \text{ insiemi}$). Occorrono 4 bit che coincidono con i 4 bit meno significativi del "block frame" Pertanto i 15 bit di indirizzo della memoria primaria vengono interpretati come:

tag	Cache index	offset
7 bit	4 bit	6 bit

2) La memoria è divisa in blocchi di 64 byte ciascuno in modo che la richiesta di un dato non presente in cache causa il trasferimento dalla memoria principale alla cache di blocco a cui appartiene il dato richiesto. Nel caso proposto i dati richiesti sono così suddivisi:

INDIRIZZAMENTO DIRETTO: blocco i -esimo di primaria – blocco i -esimo di cache.

INDIRIZZAMENTO ASSOCIATIVO SU INSIEMI:

indirizzi $0 \div 63 \Rightarrow$ blocco 0; indirizzi $64 \div 127 \Rightarrow$ blocco 1; ... ; $4032 \div 4095 \Rightarrow$ blocco 63. I primi quindici blocchi vengono allocati in altrettanti insiemi. Per verificare il calcolo basta estrarre il resto della divisione $\text{blockframe}/\text{numero di insiemi}$. Alla fine la cache è piena con la seguente situazione:

Insieme 0 Blocchi: 0, 16, 32, 48	Insieme 1 blocchi: 1, 17, 33, 49	Insieme 2 blocchi: 2, 18, 34, 50	Insieme 3 blocchi: 3, 19, 35, 51
Insieme 4 Blocchi: 4, 20, 36, 52	Insieme 5 blocchi: 5, 21, 37, 53	Insieme 6 blocchi: 6, 22, 38, 54	Insieme 7 blocchi: 7, 23, 39, 55
Insieme 8 Blocchi: 8, 24, 40, 56	Insieme 9 blocchi: 9, 25, 41, 57	Insieme 10 blocchi: 10, 26, 42, 58	Insieme 11 blocchi: 11, 27, 43, 59
Insieme 12 Blocchi: 12, 28, 44, 60	Insieme 13 blocchi: 13, 29, 45, 61	Insieme 14 blocchi: 14, 30, 46, 62	Insieme 15 blocchi: 15, 31, 47, 63

ESERCIZIO 5 (solo VO: 5 punti)

Descrivere la classificazione di Flynn delle architetture parallele, spiegando che tipo di calcolatore ricade in ciascuna architettura.

Illustrare le misure di "speedup" e "efficienza" usate per valutare le prestazioni di una architettura multiprocessore.

Soluzione

Vedi dispense del corso su "Architetture Parallele"

ESERCIZIO 4 (NO: 8 punti – VO: 7 punti)

Si consideri un sistema operativo con gestione della memoria segmentata e paginata. Ciascun job viene suddiviso al massimo in 3 segmenti. Ciascun segmento ha una dimensione massima di 128 pagine. Le pagine hanno dimensione 1KB. Si abbia ad esempio la seguente situazione:

Tabella dei Segmenti	
Segmento	Tabella delle Pagine
00	2
01	0
10	1

Tabella delle Pagine 0			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0001010	1	0
0000001	0010001	0	0
0000010	1011001	1	0
0000011	1011010	1	0
0000100	0011101	1	0
0000101	0101111	0	0
0000110	0110111	1	0
0000111	0100000	1	0
0001000	0100100	1	0
0001001	0001001	1	0

Tabella delle Pagine 1			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0000011	0	0
0000001	0010110	0	0
0000010	1001001	1	0
0000011	1001010	1	0
0000100	1010101	1	0
0000101	0011101	1	0
0000110	0111111	1	0
0000111	1011101	1	0
0001000	1010011	1	0
0001001	0001111	1	0
0001010	0011011	1	0
0001011	0100010	1	0

Tabella delle Pagine 2			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0100001	1	1
0000001	0101110	1	1
0000010	0110110	0	1
0000011	0000110	1	1
0000100	1100011	0	1
0000101	1000011	1	1
0000110	0010101	1	1

Il bit di validità indica se la pagina virtuale richiesta è presente (1) o no (0) nella memoria principale. Si considerino le seguenti richieste e mostrare: a quale indirizzo fisico corrispondono gli indirizzi virtuali, se l'operazione può essere conclusa con successo o se viene generato un errore (segmento non valido, pagina non valida, violazione di protezione) o un page fault:

- 1) Read 0000000010101001101
- 2) Write 1000001000101011001
- 3) Read 0100001011001110011
- 4) Write 0000000111100100110
- 5) Read 0100010000001100001
- 6) Write 1110000100000100111

Soluzione:

- 1) Indirizzo fisico valido 01011100101001101
- 2) Indirizzo fisico valido 10101010101011001
- 3) Indirizzo fisico non valido (pagina non valida)
- 4) Violazione di scrittura sull'indirizzo fisico 00001101100100110
- 5) Indirizzo fisico valido 01001000001100001

6) Segmento non valido