

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI  
CALCOLATORI ELETTRONICI  
NUOVO ORDINAMENTO DIDATTICO**

3 Luglio 2003

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

**ESERCIZIO 1 (10 punti)**

Progettare una rete sequenziale che ricevendo in ingresso una sequenza di bit  $X$ , produca un'uscita  $Z$  posta a 1 solo in corrispondenza della sequenza 01100101.

- (a) (6 punti) disegnare il grafo degli stati, la tabella di flusso, e la tabella delle transizioni con l'utilizzo di flip flop JK;
- (b) (4 punti) minimizzare le funzioni di transizione dello stato attraverso le mappe di Karnaugh. Indicare anche l'espressione algebrica dell'uscita  $Z$ .

**Soluzione**

Grafo degli stati:

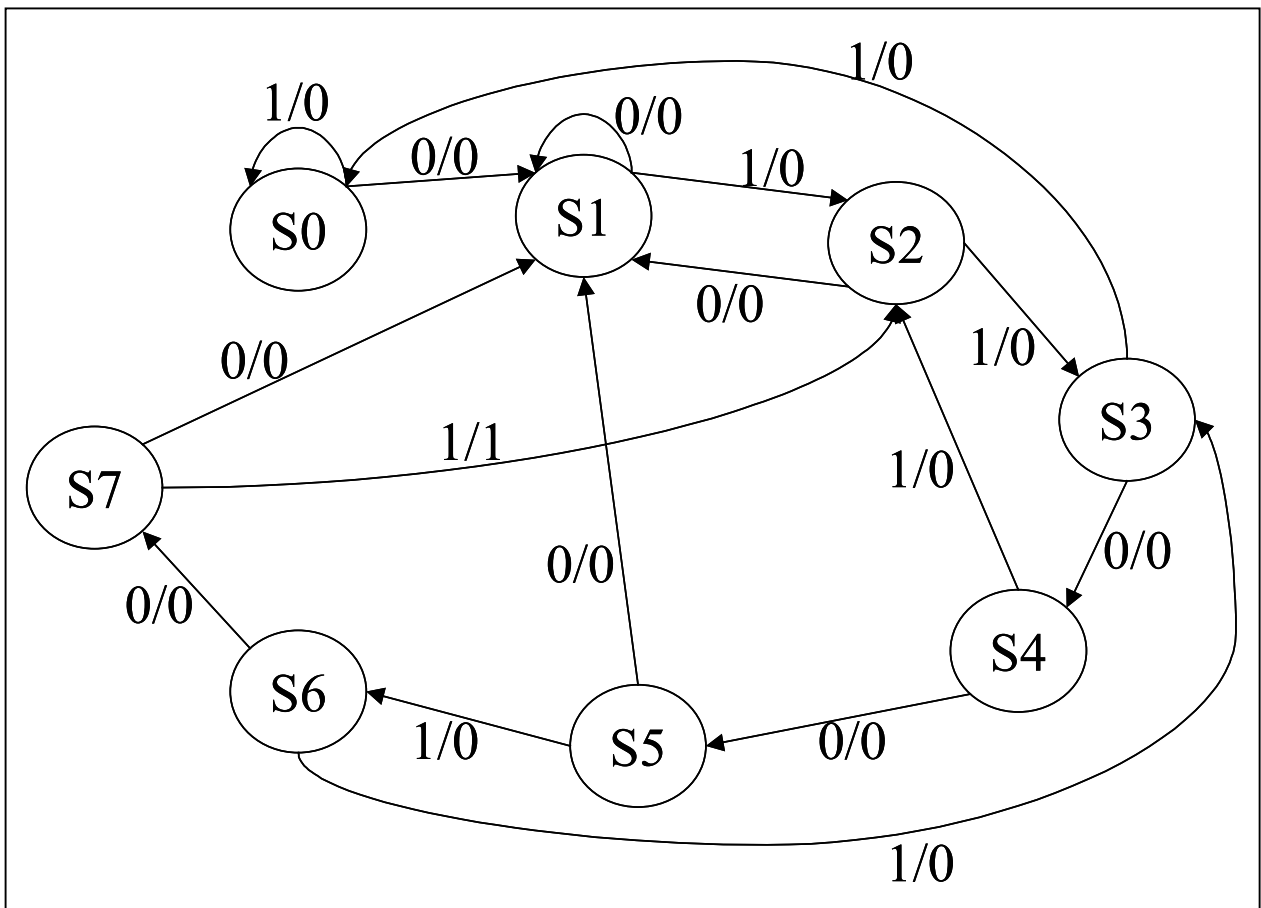


Tabella di flusso:

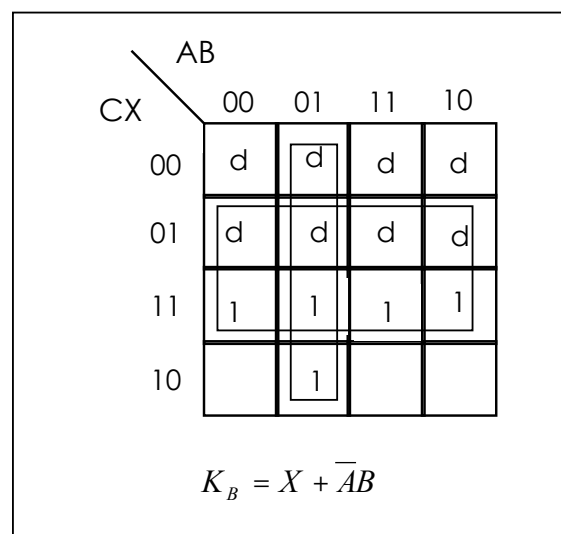
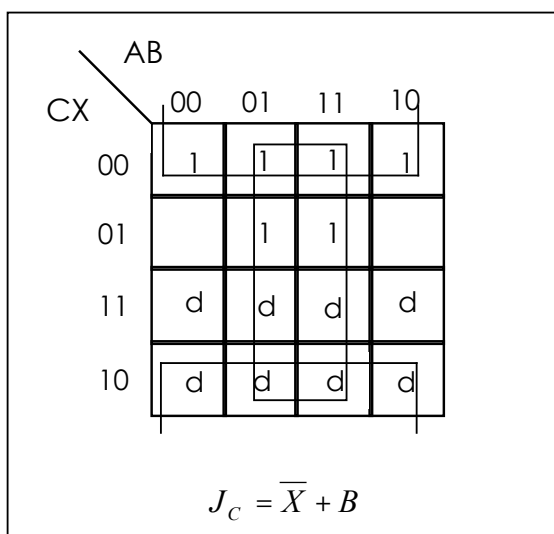
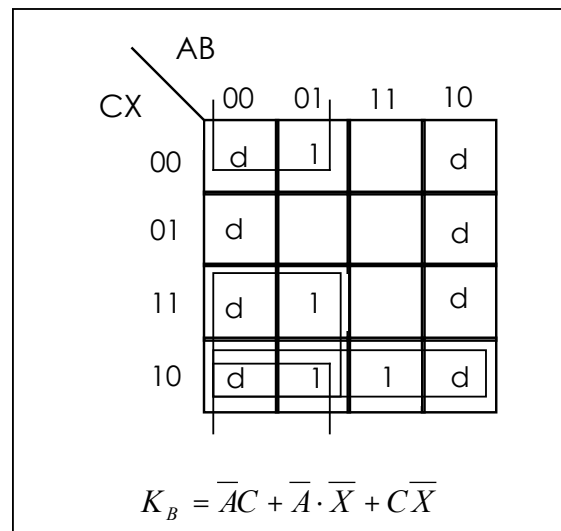
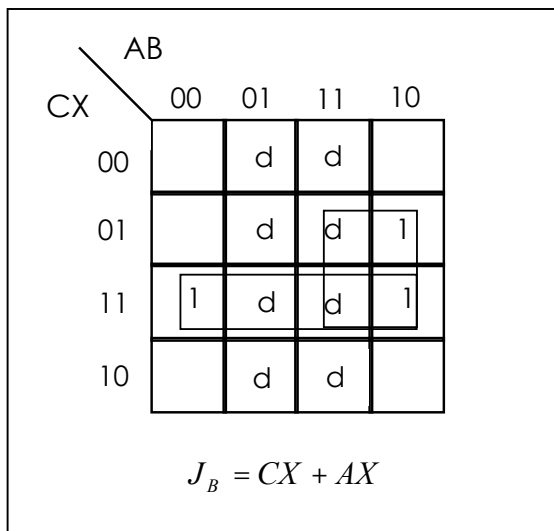
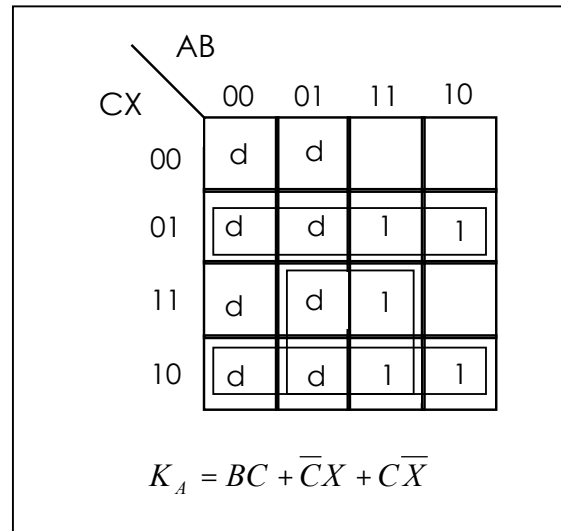
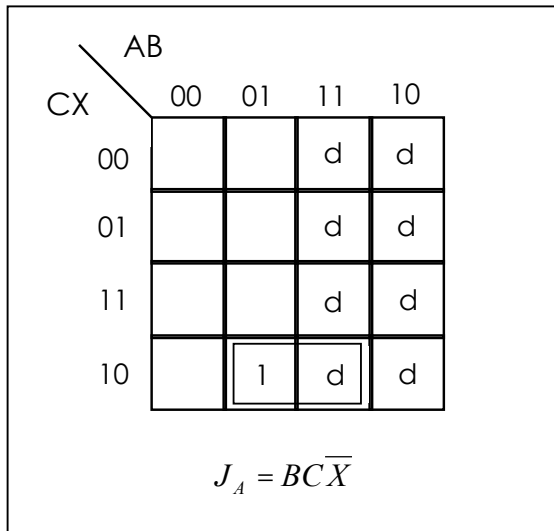
Stato presente	Stato futuro/Uscita	
	X=0	X=1
S0	S1/0	S0/0
S1	S1/0	S2/0
S2	S1/0	S3/0
S3	S4/0	S0/0
S4	S5/0	S2/0
S5	S1/0	S6/0
S6	S7/0	S3/0
S7	S1/0	S2/1

Tabella di eccitazione FF-JK:

Q	Q'	J	K
0	0	0	D
0	1	1	D
1	0	D	1
1	1	D	0

Tabella delle transizioni:

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



Infine, per quanto riguarda l'uscita:  $Z = ABCX$ .

## 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 **011110100101** (il bit meno significativo è a sinistra), ottenuta dalla codifica di Hamming di una stringa di N bit. **Spiegando bene ogni passo del ragionamento:**

- 1) (2 punti) calcolare N, sapendo che è stato usato il minimo numero di bit di controllo necessari per la codifica della parola di N bit;
- 2) (5 punti) decodificare la stringa data, rilevando eventuali errori.

### 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  $N+K=12$ , il numero minimo di bit di controllo richiesto è 4.

- 2) Nella codifica di Hamming, la sequenza in ingresso presenta la seguente struttura:

$c_0$	$c_1$	$b_0$	$c_2$	$b_1$	$b_2$	$b_3$	$c_3$	$b_4$	$b_5$	$b_6$	$b_7$
0	1	1	1	1	0	1	0	0	1	0	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. Tali bit si ottengono con le seguenti operazioni

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

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

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

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

I bit di controllo effettivamente ricevuti sono invece, nell'ordine 0, 1, 1, 0. Calcolando il vettore di errore:

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

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

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

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

Poiché il vettore risultante 0001 non è nullo, 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 primo, che essendo però un bit di controllo, non interferisce nel contenuto informativo della stringa ricevuta. La stringa decodificata è dunque: 11010101.

### ESERCIZIO 3 (8 punti)

Scrivere una funzione Assembler MIPS che, ricevendo in ingresso due interi  $x$  e  $y$ , restituisca in uscita il quoziente (parte intera) e il resto della divisione  $x/y$ . Si assuma che  $x \rightarrow \$4$ ,  $y \rightarrow \$5$ , quoziente  $\rightarrow \$6$ , resto  $\rightarrow \$7$ .

Per esempio il codice MIPS potrebbe implementare il seguente codice C:

```
void dividi(int x, int y, int *quoziente, int *resto)
{
    *quoziente=0;
    while (x>=y)
    {
        x -= y;
        (*quoziente)++;
    }
    *resto=x;
}
```

### Soluzione.

```
1.
dividi:      addi $29, $29, -4      #salvataggio contesto
             sw $8, 0($29)
             move $6, $0           #*quoziente=0
while:       slt $8, $4, $5        #if(x<y) exit
             bne $8, $0, exit
             sub $4, $4, $5        #x -= y
             addi $6, $6, 1        #(*quoziente)++
             j while
exit:        move $7, $4           #*resto=x
             lw $8, 0($29)         #ripristino del contesto
             addi $29, $29, 4
             jr $31               #ritorno al prog.principale
```

#### **ESERCIZIO 4 (8 punti)**

Si consideri un disco rigido con le seguenti caratteristiche: velocità di rotazione = 5400 giri/min, tempo medio di posizionamento = 5 ms, 34 settori per traccia di 512 byte ciascuno.

- a) (4 punti) Calcolare il tempo medio di trasferimento di un file da 8 kbyte considerando il file posizionato su settori collocati in tracce diverse e la testina all'istante iniziale sul primo settore del file.
- b) (4 punti) Supporre che il file debba essere trasferito in memoria principale, attraverso un modulo DMA. Ipotizzando una linea dati di 8 bit, e una velocità di clock di 2 Ghz, e che il buffer del DMA sia stato caricato di tutto il file, qual è il tempo minimo impiegato per il trasferimento?

#### **Soluzione**

a)

$$T_{rot} = 60/5400 \text{ sec} = 11.11 \text{ msec.}$$

$$T_{lat} = T_{rot}/2 = 5.555 \text{ msec.}$$

$$T_{ls} = T_{rot} / 34 = 0.327 \text{ msec. (tempo di lettura di un settore)}$$

$$T_{pos} = 5 \text{ msec.}$$

$$\text{Numero di settori richiesti dal file } N = 8\text{kbyte} / 512\text{byte} = 16 \text{ settori.}$$

$$T_{lett} = (N-1) * (T_{lat} + T_{pos} + T_{ls}) + T_{ls} = 15 * 10.882 + 0.327 = 163.557 \text{ msec}$$

b) In tal caso viene trasferito un byte ogni 0.5 nsec (block transfer). Quindi il tempo complessivo è  $8 * 1024 * 0.5 \text{ nsec} = 4096 \text{ nsec} = 4.1 \text{ } \mu\text{sec}$  (circa).