Esame di Progettazione di Sistemi Digitali

12 aprile 2021 - canale AL - prof. Pontarelli

Cognome ________________________ Nome____________________ Matricola_______________

Esercizio 1 (4 punti)

Verificare se le seguenti sequenze di bit sono parole di Hamming(7,4) [b3,b2,b1,b0,c2,c1,c0]. Individuare la posizione di eventuali errori e correggerli.

a) 1001011
b) 1100110

a)

$s_0 = b_0 \oplus b_1 \oplus b_2 \oplus c_0 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$
$s_1 = b_1 \oplus b_2 \oplus b_3 \oplus c_1 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$
$s_2 = b_0 \oplus b_1 \oplus b_3 \oplus c_2 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$

La parola è corretta.

b)

$s_0 = b_0 \oplus b_1 \oplus b_2 \oplus c_0 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$
$s_1 = b_1 \oplus b_2 \oplus b_3 \oplus c_1 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$
$s_2 = b_0 \oplus b_1 \oplus b_3 \oplus c_2 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$

c'è un errore su b_2 .

La parola corretta è 1000110

Esercizio 2 (6 punti)

Progettare, utilizzando il numero minimo di porte logiche AND/OR/NOT, un FF di tipo SR ed un flip-flop di tipo T il circuito sequenziale corrispondente alla seguente tabella di transizione di stato.

	x	$\bar{x}$
A	B/0	A/0
B	A/0	C/0
C	C/0	D/1
D	B/1	C/0

Tabella degli stati, utilizzando un flip-flop T per Q_1 e un flip-flop SR per Q_0

PS	Q_1	Q_0	x	NS	Q_1’	Q_0’	T_1	S_0	R_0	z
A	0	0	0	A	0	0	0	0	-	0
A	0	0	1	B	0	1	0	1	0	0
B	0	1	0	C	1	0	1	0	1	0
B	0	1	1	A	0	0	0	0	1	0
C	1	0	0	D	1	1	0	1	0	1
C	1	0	1	C	1	0	0	0	-	0
D	1	1	0	C	1	0	0	0	1	0
D	1	1	1	B	0	1	1	-	0	1

Le equazioni corrispondenti sono:

$$T_1 = \bar{Q}_1 Q_0 \bar{x} + Q_1 Q_0 x$$

$$S_0 = \bar{Q}_1 \bar{Q}_0 x + Q_1 \bar{Q}_0 \bar{x}$$

$$R_0 = \bar{Q}_1 Q_0 \bar{x} + \bar{Q}_1 Q_0 x + Q_1 Q_0 \bar{x} = \bar{Q}_1 Q_0 + Q_1 Q_0 \bar{x} = \bar{Q}_1 Q_0 + Q_0 \bar{x}$$

$$z = Q_1 \bar{Q}_0 \bar{x} + Q_1 Q_0 x$$

Esercizio 3 (4 punti)

Dati i numeri X= **45A00000** e Y= **C5100000** espressi nella rappresentazione in virgola mobile secondo lo standard IEEE-754. (a) convertire i numeri nel formato decimale, (b) eseguire l'operazione X+Y seguendo l'algoritmo previsto dallo standard IEEE-754, e (c) rappresentare il risultato sia in notazione decimale che in esadecimale IEEE-754.

(a) conversione

X = 0100_0101_1010_0000_0000_0000_0000_0000

Segno = +

esponente: 1000_1011 = 128+8+2+1= 139

esponente-bias= 139-127=12

mantissa = 1.01000000000000000000000 → 1.25

X= $1.25*2^{12}= (1+2^{-2}) *2^{12}= (2^{12}+2^{10}) =4096+1024= 5120$

Y = 1100_0101_0001_0000_0000_0000_0000_0000

Segno = -

esponente: 1000_1010 = 128+8+2= 138

esponente-bias= 138-127=11

mantissa = 1.00100000000000000000000 → 1.125

Y= $-1.125*2^{11}= -(1+2^{-3}) *2^{11}= -(2^{11}+2^{8}) = -(2048+256) = -2304$

(b) somma

1. allineo gli esponenti, scrivendo

$Y= -(1,001)_2 \cdot 2^{11} = -(0,1001)_2 \cdot 2^{12}$

2. eseguo il complemento a 2 della mantissa

$m_Y= -(0,1001) = 1,0111$

3. eseguo la somma, <u>estendendo</u> il segno

m_x	01.0100000000000000000000	+
m_y	11.0111000000000000000000	=
m_Z	00.1011000000000000000000	

4. normalizzo Z

$Z= 00.1011000000000000000000 \cdot 2^{12} = 1.011000000000000000000 \cdot 2^{11}$

(c) conversione di Z

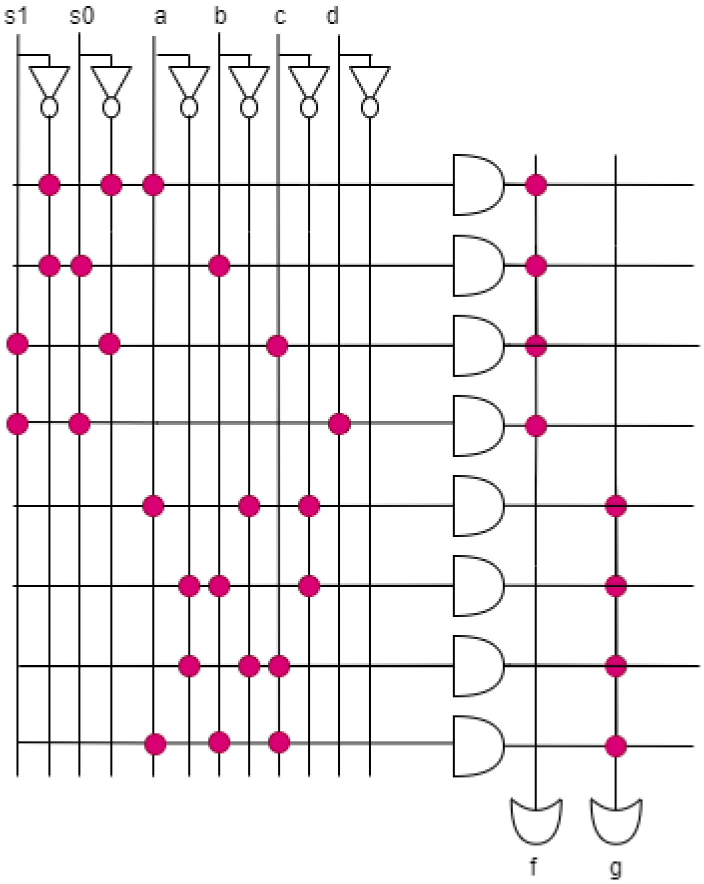
Segno = +
esponente-bias= 11
esponente= 127+11= 138= 1000_1010
mantissa = 1.01100000000000000000000 → 1.375

Z = 0100_0101_0011_0000_0000_0000_0000_0000 = **0x45300000**
Z = $1.375 * 2^{11}$ =2048+512+256=2816

Esercizio 4 (4 punti)

Utilizzare una PLA per realizzare una funzione *f* equivalente ad un mux con 4 ingressi ed una funzione *g* equivalente ad uno XOR a 3 ingressi

$$f = a\bar{s}_1\bar{s}_0 + b\bar{s}_1 s_0 + c s_1\bar{s}_0 + d s_1 s_0$$
$$g = a\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}\bar{b}c + abc$$



Esercizio 5 (4 punti)
Usando gli assiomi dell'algebra di Boole,

- **verificare la seguente identità:**

$$\overline{a+\bar{b}c} + \left(\bar{b}c \oplus \bar{a}b\right) = \bar{a} + \bar{b}c =$$
$$\bar{a}\overline{\bar{b}c} + \bar{b}c\overline{\bar{a}b} + \overline{\bar{b}c}\,\bar{a}b = \bar{a}(b+\bar{c}) + \bar{b}c\left(a+\bar{b}\right) + (b+\bar{c})\bar{a}b =$$
$$\bar{a}(b+\bar{c}) + \bar{b}c\left(a+\bar{b}\right) = \bar{a}(b+\bar{c}) + \bar{b}c(a+1) = \bar{a}(b+\bar{c}) + \bar{b}c = \bar{a}(b+\bar{c}) + \overline{(b+\bar{c})} = \bar{a} + \bar{b}c$$

per assorbimento:
$$\bar{a}(b+\bar{c}) + (b+\bar{c})\bar{a}b = \bar{a}(b+\bar{c})$$
$$\bar{a}(b+\bar{c}) + \overline{(b+\bar{c})} = \bar{a} + \overline{(b+\bar{c})}$$

- **scrivere la forma POS normale**

$$\bar{a} + \bar{b}c = (\bar{a}+\bar{b})(\bar{a}+c)$$

- **riscrivere la funzione utilizzando esclusivamente l'operatore NOR**

$$\left(\bar{a}+\bar{b}\right)(\bar{a}+c) = \overline{\overline{\left(\bar{a}+\bar{b}\right)(\bar{a}+c)}} = \overline{\overline{\bar{a}+\bar{b}} + \overline{\bar{a}+c}} = \overline{\bar{a}+\bar{b}}\ NOR\ \overline{\bar{a}+c} = \left(\bar{a}\ NOR\ \bar{b}\right) NOR\left(\bar{a}\ NOR\ c\right) =$$
$$= \left((a\ NOR\ a)\ NOR\ (b\ NOR\ b)\right) NOR\left((a\ NOR\ a)\ NOR\ c\right)$$

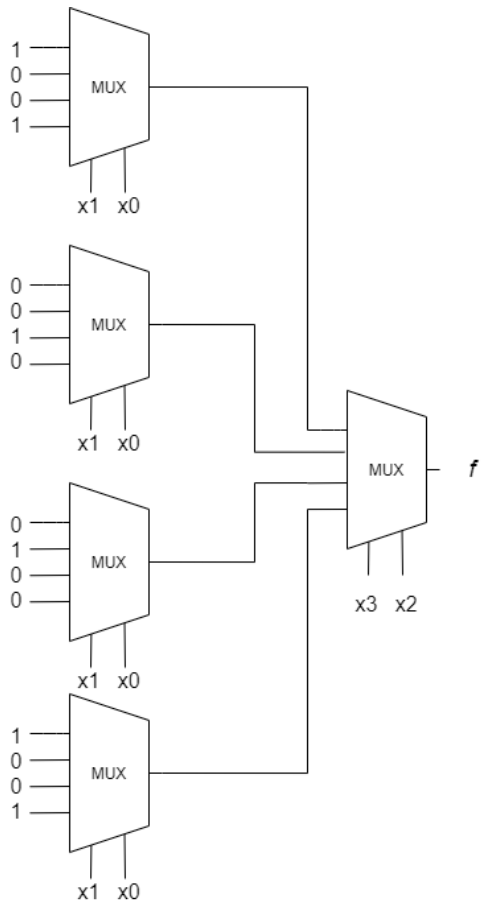
Esercizio 6 (5 punti)

Data una funzione logica $f(x_3, x_2, x_1, x_0)$ che dà vale 1 quando la codifica binaria dei suoi ingressi $x_3x_2x_1x_0$ è un multiplo di 3.

- disegnare, utilizzando soltanto mux a 4 ingressi, il circuito logico che la realizza.
- Realizzare la forma minima POS
- Realizzare la forma canonica SOP

Tabella della verità:

x4	x3	x2	x1	*f*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



Forma minima POS

Y

x1 x0 \ x3 x2	00	01	11	10
00	1	0	1	0
01	0	0	0	1
11	1	0	1	0
10	0	1	0	0

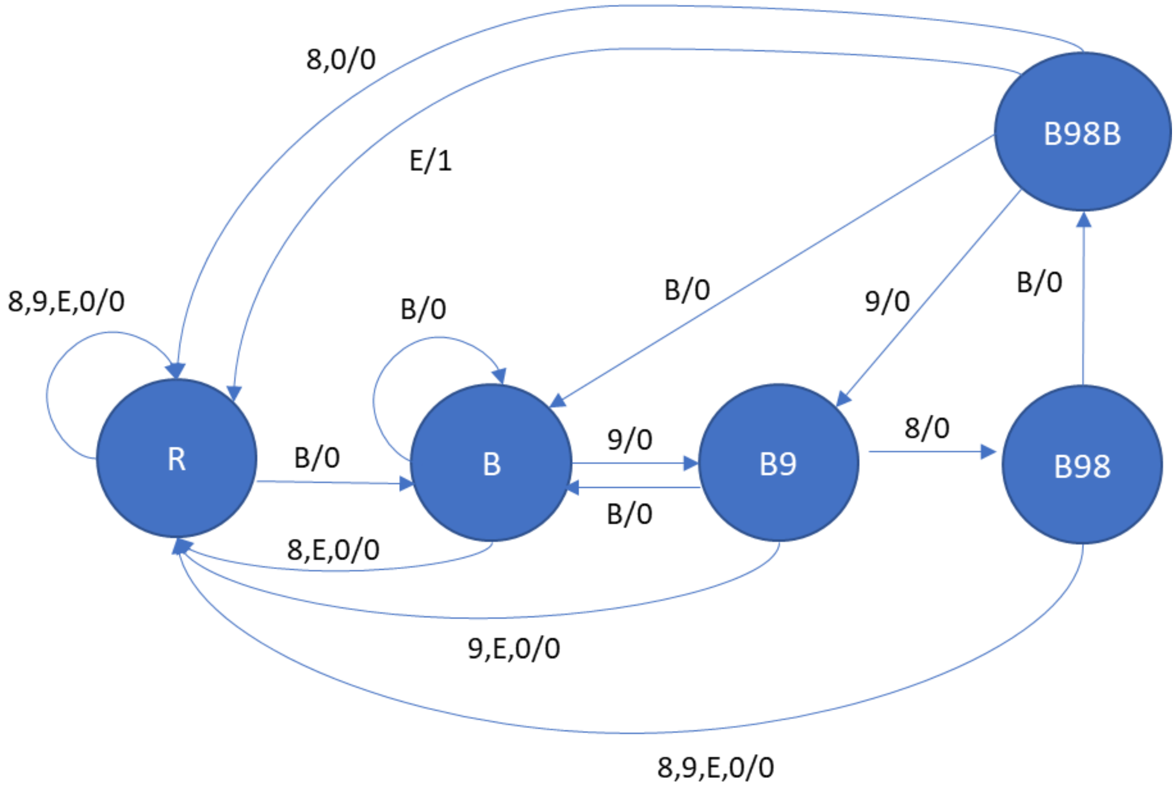
$$f = (x_3 + \bar{x}_2 + x_1)(x_3 + \bar{x}_2 + \bar{x}_0)(\bar{x}_3 + x_1 + \bar{x}_0)(\bar{x}_2 + x_1 + \bar{x}_0)(x_2 + \bar{x}_1 + x_0)(\bar{x}_3 + \bar{x}_1 + x_0)(\bar{x}_3 + x_2 + \bar{x}_1)(\bar{x}_3 + x_2 + x_0)$$

Forma canonica SOP

$$f = \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 + \bar{x}_3\bar{x}_2 x_1 x_0 + \bar{x}_3 x_2 x_1 \bar{x}_0 + x_3 \bar{x}_2 \bar{x}_1 x_0 + x_3 x_2 \bar{x}_1 \bar{x}_0 + x_3 x_2 x_1 x_0$$

Esercizio 7 (3 punti)

Un automa (FSM) riceve in ingresso 4 bits x_3,x_2,x_1,x_0 che rappresentano una cifra esadecimale, e fornisce in uscita il valore z_1=0 se riconosce la sequenza di cifre B98BE. Disegnare l'automa e scrivere la tabella di transizione di stato.
N.B.: per la tabella di transizione di stato è sufficiente descrivere i casi in cui i bit assumono i valori 8,9,B,E, e 0 (altro).



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
R	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	B/0	R/0	R/0	R/0
B	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	B9/0	R/0	B/0	R/0	R/0	R/0
B9	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	B98/0	R/0	R/0	B/0	R/0	R/0	R/0
B98	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	B98B/0	R/0	R/0	R/0
B98B	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	R/0	B9	R/0	B/0	R/0	R/0	R/1