SOLUZIONI DEL PRIMO ESONERO di PROGETTAZIONE di SISTEMI DIGITALI
CANALE A-L prof. Gorla **FILA A**

Esercizio 1. Siano dati i seguenti numeri binari in rappresentazione con virgola mobile (1 bit di segno, 6 di mantissa e 3 di esponente): <0,100100,001> e <0,110000,111>.

a)Se ne effettui la somma, eventualmente normalizzando il risultato (3 punti):

Procedimento:

Per poter interpretare i due numeri bisogna convertire l'esponente, che è espresso in complemento a 2. Nel primo caso il bit più significativo è 0, quindi l'esponente è positivo e facilmente si ottiene 1 convertendo in decimale. Nel secondo caso il bit più significativo è 1, quindi l'esponente è negativo e per convertirlo sottraggo 1 (ottenendo 110), lo complemento bit a bit (ottenendo 001) che considerando il segno corrisponde a -1. Si prende il secondo numero che ha esponente più piccolo e si sposta la virgola in modo da unificare gli esponenti. Si effettua quindi l'addizione in colonna delle mantisse:

0,0011 0,1001
0,1100

Il risultato è già normalizzato. Pertanto: **< 0 , 110000 , 001 >**

b)Si converta il risultato ottenuto in base 10 (1 punto):

Procedimento:

Il risultato al punto precedente può essere scritto come $1,1_{(2)}$.
La parte intera in decimale è 1 e quella dopo la virgola è pari 1:2=0,5.
Risultato: **1,5**

c) Si converta il numero così ottenuto in base 5, fermandosi alla 4^ cifra dopo la virgola (2 punti):

Procedimento:

Convertiamo separatamente la parte intera (1) e la parte dopo la virgola (0,5).
La parte intera corrisponde a 1 anche in base 5.
Per 0,5 effettuiamo la procedura di conversione:

Operazione	**Parte Intera**	**Parte Decimale**
5 x 0,5 = 2,5	2	0,5
5 x 0,5 = 2,5	2	0,5
5 x 0,5 = 2,5	2	0,5
5 x 0,5 = 2,5	2	0,5

La prime 4 cifre della parte decimale sono quindi $0,2222_{(5)}$.
Si poteva anche evitare di effettuare le operazioni dopo la prima notando che la parte decimale che si ottiene è la stessa di partenza e quindi il numero è infinito periodico.

Risultato: **1,2222**

Esercizio 2. Si consideri la seguente stringa binaria: 0100.
a)*Se ne calcoli il bit di parità pari (1 punto):* Risposta: **1**

b)Si scriva la stringa come una matrice 2x2 e se ne calcolino i bit di parità pari longitudinale e trasversale (1 punto):

0	**1**	**1**
0	**0**	**0**
0	**1**	

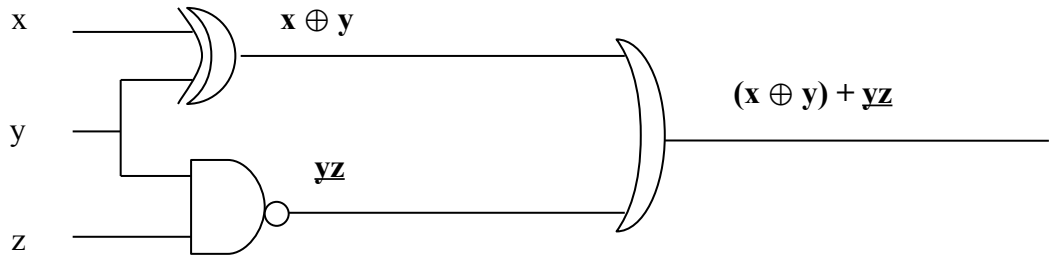
c)Si scriva la parola di codice di Hamming associata alla stringa di bit data (2 punti):

Procedimento:

La parola sarà XX0X100, col primo bit che controlla la parità dei bit 1, 3, 5, 7, il secondo quella dei bit 2, 3, 6, 7, il quarto quella dei bit 4, 5, 6, 7.

Risultato: **1001100**

Esercizio 3. Si consideri il seguente circuito combinatorio:



a)Si scriva sul disegno, in corrispondenza di ogni linea di uscita di ogni porta, l'espressione booleana calcolata dalla porta (1 punto)

b)Si calcoli la forma canonica SOP (o forma canonica disgiuntiva) associata all'espressione finale (5 punti)

Procedimento:

$(x \oplus y) + \underline{y}\,\underline{z} =$ (dalla definizione dello XOR)
$= \underline{x}\,y + x\,\underline{y} + \underline{y}\,\underline{z} =$ (applicando de morgan)
$= \underline{x}\,y + x\,\underline{y} + \underline{y} + \underline{z} =$ (propr. dell'elem. neutro e del complem.)
$= \underline{x}\,y\,(z + \underline{z}) + x\,\underline{y}\,(z + \underline{z}) + \underline{y}\,(x + \underline{x})\,(z + \underline{z}) + \underline{z}\,(x + \underline{x})\,(y + \underline{y}) =$ (propr. distributiva e assorbim.)
$= \mathbf{\underline{x}\,y\,z + \underline{x}\,y\,\underline{z} + x\,\underline{y}\,z + x\,\underline{y}\,\underline{z} + \underline{x}\,\underline{y}\,z + \underline{x}\,\underline{y}\,\underline{z} + x\,y\,\underline{z}}$

c)Si scriva la tavola di verità della funzione booleana associata (2 punti)

x	***y***	***z***	***f(x, y, z)***
0	**0**	**0**	**1**
0	**0**	**1**	**1**
0	**1**	**0**	**1**
0	**1**	**1**	**1**
1	**0**	**0**	**1**
1	**0**	**1**	**1**
1	**1**	**0**	**1**
1	**1**	**1**	**0**

Esercizio 4. Si vuole realizzare un circuito combinatorio associato alla funzione booleana di 4 variabili binarie di ingresso e due uscite binarie che restituisce in output la codifica binaria di quante linee in ingresso valgono 1. Si assuma che le variabili di ingresso non siano mai tutte contemporaneamente ad 1.

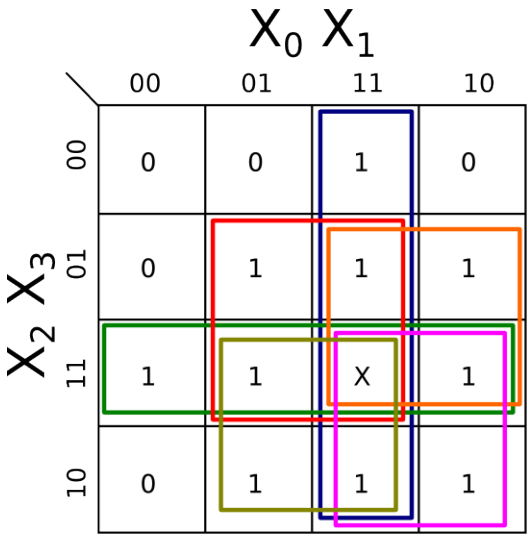
Esempio: se l'input è 0101, l'output sarà 10 (la codifica binaria del numero 2)

a)Si scriva la tavola di verità della funzione (2 punti)

X_0	X_1	X_2	X_3	Y_1	Y_0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	1	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	X	X

b)Si calcoli la forma normale SOP minimale per il bit più significativo della funzione (2 punti)

Procedimento:



$Y_1 = X_0X_1 + X_2X_3 + X_0X_2 + X_1X_2 + X_0X_3 + X_1X_3$

c)Si semplifichi l'espressione ottenuta usando algebra di Boole e altre porte logiche (2 punti)

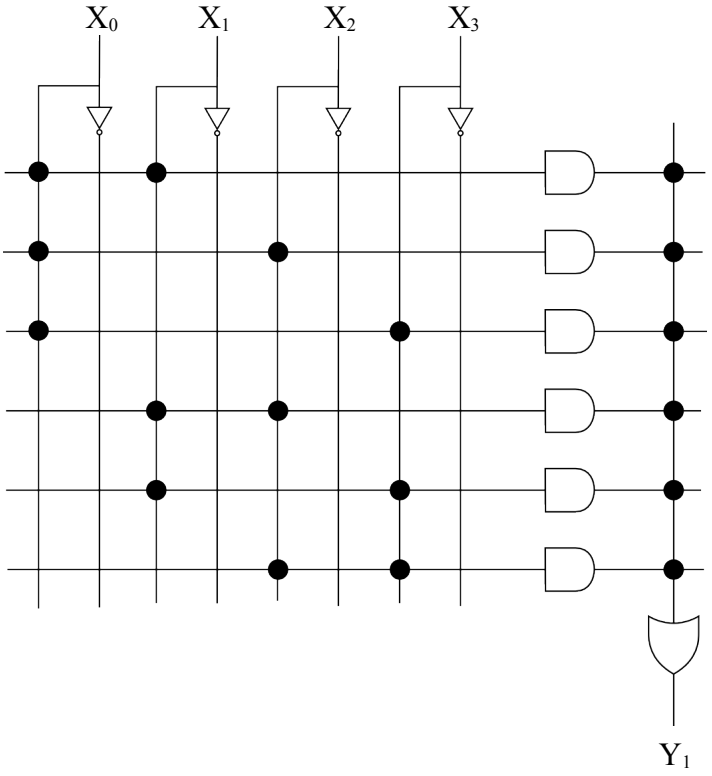
Procedimento:

$X_0 X_1 + X_2 X_3 + X_1 X_3 + X_0 X_3 + X_1 X_2 + X_0 X_2 =$
$= X_0 (X_1 + X_3) + X_2 (X_1 + X_3) + X_1 X_3 + X_0 X_2 =$
$= (X_0 + X_2) (X_1 + X_3) + X_1 X_3 + X_0 X_2$

Risultato: $\mathbf{(X_0 + X_2) (X_1 + X_3) + X_1 X_3 + X_0 X_2}$

d)Si realizzi tramite PLA l'espressione ottenuta al punto (b) (2 punti)

(N.B.: è necessario visualizzare la matrice di AND e di OR)



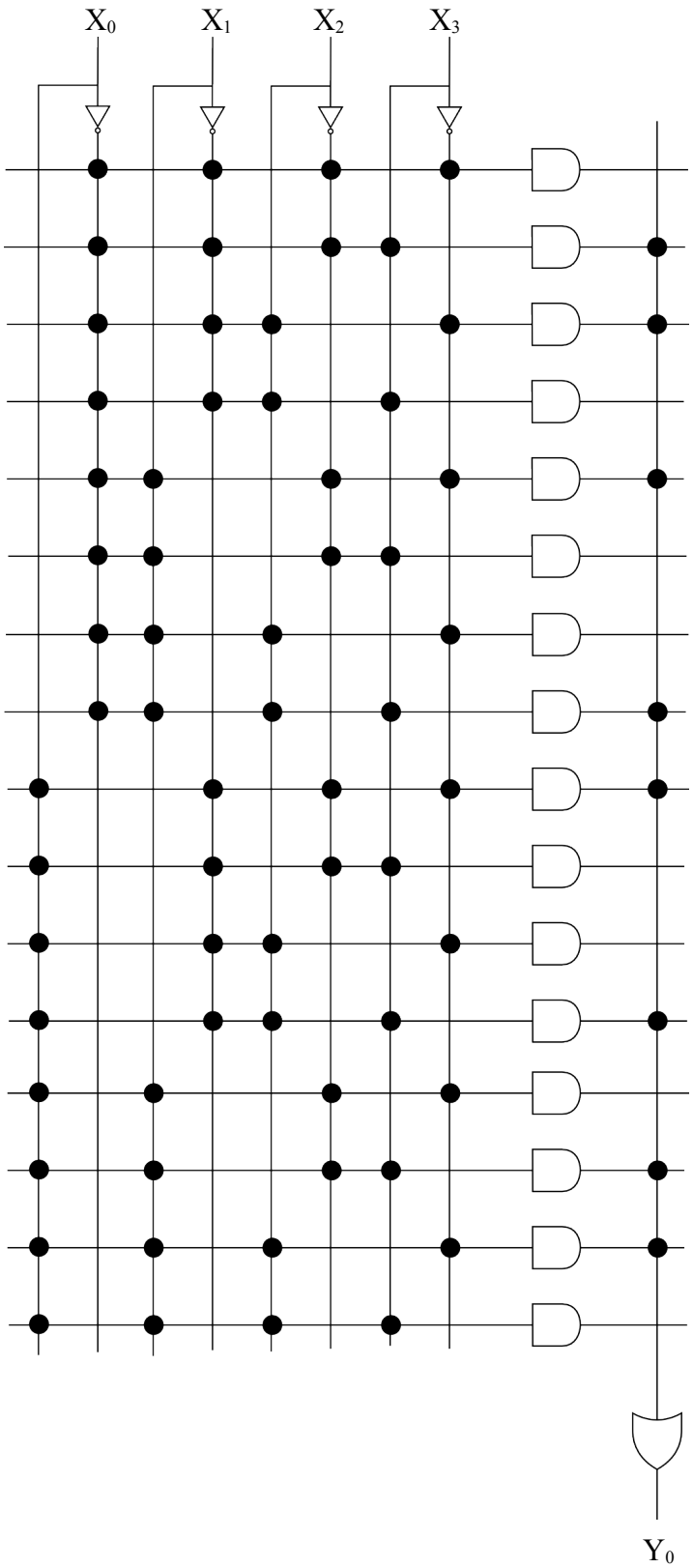
e)Si scriva la forma canonica disgiuntiva (SOP) per il bit meno significativo della funzione (2 punti)

Risultato:

$\mathbf{Y_0 = \underline{X_0}\underline{X_1}\underline{X_2}X_3 + \underline{X_0}\underline{X_1}X_2\underline{X_3} + \underline{X_0}X_1\underline{X_2}\underline{X_3} + \underline{X_0}X_1X_2X_3 + X_0\underline{X_1}\underline{X_2}\underline{X_3} + X_0\underline{X_1}X_2X_3 + X_0X_1\underline{X_2}X_3 + X_0X_1X_2\underline{X_3}}$

f)Si realizzi tramite ROM l'espressione ottenuta al punto (e) (2 punti)

(N.B.: è necessario specificare la matrice di AND e di OR)



g)Si realizzi tramite un MULTIPLEXER 16-a-1 il bit meno significativo della funzione (2 punti)
(N.B.:si può assumere un modulo MUX che si comporta come un multiplexer 16-a-1; basta scriverne le entrate)

I bit di selezione corrisponderanno alle entrate X_0 X_1 X_2 X_3.
I bit dei 16 canali di input saranno invece fissati ai valori della colonna Y_0 della tabella di verità:
0110100110010110 (l'ultimo bit può indifferentemente essere messo a 1)