

## 1. Ejercicio 1

## 2. Ejercicio 2

Se tiene la siguiente expresión en maxtérminos  $f(d; c; b; a) = \prod(M_0; M_1; M_5; M_7; M_8; M_{10}; M_{14}; M_{15})$

De aquí se desprende la siguiente tabla de verdad, de la cual se derivan las expresiones completas en función de las entradas.

d	c	b	a	-	f	$M_i$
0	0	0	0	-	0	$M_0$
0	0	0	1	-	0	$M_1$
0	0	1	0	-	1	$M_2$
0	0	1	1	-	1	$M_3$
0	1	0	0	-	1	$M_4$
0	1	0	1	-	0	$M_5$
0	1	1	0	-	1	$M_6$
0	1	1	1	-	0	$M_7$
0	0	0	0	-	0	$M_8$
0	0	0	1	-	1	$M_9$
0	0	1	0	-	0	$M_{10}$
0	0	1	1	-	1	$M_{11}$
0	1	0	0	-	1	$M_{12}$
0	1	0	1	-	1	$M_{13}$
0	1	1	0	-	0	$M_{14}$
0	1	1	1	-	0	$M_{15}$

Expandiendo la expresión,  $f = (d + c + b + a) \cdot (d + c + b + \bar{a}) \cdot (d + \bar{c} + b + \bar{a}) \cdot (d + \bar{c} + \bar{b} + \bar{a}) \cdot (\bar{d} + c + b + a) \cdot (\bar{d} + c + \bar{b} + a) \cdot (\bar{d} + \bar{c} + \bar{b} + a) \cdot (\bar{d} + \bar{c} + \bar{b} + \bar{a})$

1. Simplificamos mediante el uso del álgebra booleana:

Agrupando a los maxtérminos anteriores de a dos y en orden, aplicando la propiedad 14.b del álgebra de Boole de la página del libro "Fundamentals of Digital Logic with Verilog Design" propuesto por la cátedra, la expresión queda simplificada a:

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

Luego, aplicando los siguientes cambios de variable y la propiedad 17.a del libro:

$$\begin{cases} y = c + b \\ z = \bar{c} + \bar{a} \\ y' = c + a \\ z' = \bar{c} + \bar{b} \end{cases} \quad (1)$$

$$f = (d + y \cdot z) \cdot (\bar{d} + y' \cdot z')$$

Aplicamos propiedad distributiva y de nuevo la propiedad 17.a para llegar a:

$$f = d \cdot y' \cdot z' + \bar{d} \cdot y \cdot z + y' \cdot z' \cdot y \cdot z$$

$$f = d \cdot y' \cdot z' + \bar{d} \cdot y \cdot z$$

Volviendo a las variables originales:

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

que resulta ser la mínima expresión de f.

2. Simplificamos mediante el uso de mapas de Karnaugh:

$c,d a, b$	00	01	11	10
00	$M_0$	$M_2$	$M_3$	$M_1$
01	$M_8$	$M_{10}$	$M_{11}$	$M_9$
11	$M_{12}$	$M_{14}$	$M_{15}$	$M_{13}$
10	$M_4$	$M_6$	$M_7$	$M_5$

▪  $f$

$c,d a, b$	00	01	11	10
00	0	1	1	0
01	0	0	1	1
11	1	0	0	1
10	1	1	0	0

### 3. Ejercicio 3

Se desea realizar un circuito que convierta un número binario de 4 bits en su complemento a dos.

1. Expresamos el valor de cada bit de salida en función de los mintérminos de los bits de entrada.

Sean  $b_3, b_2, b_1$  y  $b_0$  los bits de entrada, donde  $b_3$  es el bit más significativo y  $b_0$  el menos significativo.

A su vez, sean  $y_3, y_2, y_1$  e  $y_0$  los bits de salida (complemento a dos de la entrada), donde  $y_3$  es el bit más significativo e  $y_0$  el menos significativo.

Luego, se considera cada bit de salida por separado como una función f de los bit de entrada de forma tal que  $y_j = f(b_3; b_2; b_1; b_0)$ , con  $j = 3, 2, 1, 0$ . Cada  $y_j$  tendrá 16 posibles valores, que serán identificados como  $y_{j,i}$ , con  $i=0;1;\dots;15$

Es sabido que cada  $y_j$  puede ser vista como una suma (operación lógica OR) de los mintérminos de los bits de entrada. Así,  $y_j = \sum_{i=0}^n m_{j,i} \cdot y_{j,i}$ , con  $n = 15$  por ser 16 las posibles entradas de 4 bits y siendo  $m_{j,i}$  el mintérmino correspondiente al i-ésimo valor posible del j-ésimo bit de salida.

$b_3$	$b_2$	$b_1$	$b_0$	-	$y_3$	$y_2$	$y_1$	$y_0$	
0	0	0	0	-	0	0	0	0	$m_{j,0}$
0	0	0	1	-	1	1	1	1	$m_{j,1}$
0	0	1	0	-	1	1	1	0	$m_{j,2}$
0	0	1	1	-	1	1	0	1	$m_{j,3}$
0	1	0	0	-	1	1	0	0	$m_{j,4}$
0	1	0	1	-	1	0	1	1	$m_{j,5}$
0	1	1	0	-	1	0	1	0	$m_{j,6}$
0	1	1	1	-	1	0	0	1	$m_{j,7}$
1	0	0	0	-	1	0	0	0	$m_{j,8}$
1	0	0	1	-	0	1	1	1	$m_{j,9}$
1	0	1	0	-	0	1	1	0	$m_{j,10}$
1	0	1	1	-	0	1	0	1	$m_{j,11}$
1	1	0	0	-	0	1	0	0	$m_{j,12}$
1	1	0	1	-	0	0	1	1	$m_{j,13}$
1	1	1	0	-	0	0	1	0	$m_{j,14}$
1	1	1	1	-	0	0	0	1	$m_{j,15}$

De la tabla anterior, se observa que:

$$\begin{cases} y_3 = m_{3,1} + m_{3,2} + m_{3,3} + m_{3,4} + m_{3,5} + m_{3,6} + m_{3,7} + m_{3,8} \\ y_2 = m_{2,1} + m_{2,2} + m_{2,3} + m_{2,4} + m_{2,9} + m_{2,10} + m_{2,11} + m_{2,12} \\ y_1 = m_{1,1} + m_{1,2} + m_{1,5} + m_{1,6} + m_{1,9} + m_{1,10} + m_{1,13} + m_{1,14} \\ y_0 = m_{0,1} + m_{0,3} + m_{0,5} + m_{0,7} + m_{0,9} + m_{0,11} + m_{0,13} + m_{0,15} \end{cases} \quad (2)$$

- Expresamos el valor de cada bit de salida en forma simplificada. El método elegido para realizar la simplificación es el de mapas de Karnaugh:

Así, para cada  $y_j$ , el mapa de Karnaugh de 4 variables/bits queda definido como:

■  $y_j$

$b_2, b_3   b_0, b_1$	00	01	11	10
00	$m_{j,0}$	$m_{j,2}$	$m_{j,3}$	$m_{j,1}$
01	$m_{j,8}$	$m_{j,10}$	$m_{j,11}$	$m_{j,9}$
11	$m_{j,12}$	$m_{j,14}$	$m_{j,15}$	$m_{j,13}$
10	$m_{j,4}$	$m_{j,6}$	$m_{j,7}$	$m_{j,5}$

Los siguientes mapas aparecerán con los valores de sus mintérminos reemplazados y los grupos ya formados:

■  $y_3$

$b_2, b_3   b_0, b_1$	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	0	0	0	0
10	1	1	1	1

Figura 1: Mapa de Karnaugh para  $y_3$

De este mapa se puede obtener  $y_3 = \overline{b_3} \cdot b_2 + \overline{b_3} \cdot b_1 + \overline{b_3} \cdot b_0 + b_3 \cdot \overline{b_2} \cdot b_1 \cdot b_0$   
Así,  $y_3 = \overline{b_3} \cdot (b_2 + b_1 + b_0) + b_3 \cdot \overline{b_2} \cdot b_1 \cdot b_0$

■  $y_2$

$b_2, b_3   b_0, b_1$	00	01	11	10
00	0	1	1	1
01	0	1	1	1
11	1	0	0	0
10	1	0	0	0

Figura 2: Mapa de Karnaugh para  $y_2$

De este mapa se puede obtener  $y_2 = \overline{b_2} \cdot b_1 + \overline{b_2} \cdot b_0 + b_2 \cdot \overline{b_1} \cdot \overline{b_0}$   
Así,  $y_2 = \overline{b_2} \cdot (b_1 + b_0) + b_2 \cdot \overline{b_1} \cdot \overline{b_0}$

■  $y_1$

$b_2, b_3   b_0, b_1$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

Figura 3: Mapa de Karnaugh para  $y_1$

De este mapa se puede obtener  $y_1 = \overline{b_1} \cdot b_0 + \overline{b_0} \cdot b_1$   
Así,  $y_1$  resulta ser la xor entre  $b_1$  y  $b_0$ .

- $y_0$

$b_2, b_3   b_0, b_1$	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

Figura 4: Mapa de Karnaugh para  $y_0$

De este mapa se puede obtener  $y_0 = b_0$

Así, el bit menos significativo de la entrada resulta ser el bit menos significativo de la salida (conexión directa).

#### 4. Ejercicio 5

#### 5. Ejercicio 6