

**PRÁCTICO 3 - Lógica Combinacional***Minitérminos y maxitérminos para tres variables binarias*

<i>x</i>	<i>y</i>	<i>z</i>	<b>Minitérminos</b>		<b>Maxitérminos</b>	
			<b>Términos</b>	<b>Designación</b>	<b>Términos</b>	<b>Designación</b>
0	0	0	$x'y'z'$	$m_0$	$x + y + z$	$M_0$
0	0	1	$x'y'z$	$m_1$	$x + y + z'$	$M_1$
0	1	0	$x'yz'$	$m_2$	$x + y' + z$	$M_2$
0	1	1	$x'yz$	$m_3$	$x + y' + z'$	$M_3$
1	0	0	$xy'z'$	$m_4$	$x' + y + z$	$M_4$
1	0	1	$xy'z$	$m_5$	$x' + y + z'$	$M_5$
1	1	0	$xyz'$	$m_6$	$x' + y' + z$	$M_6$
1	1	1	$xyz$	$m_7$	$x' + y' + z'$	$M_7$

**Mapas de Karnaugh de 2, 3 y 4 variables:**

$m_0$	$m_1$
$m_2$	$m_3$

a)

		$y$	
		$0$	$1$
$x$	$0$	$x'y'$	$x'y$
	$1$	$xy'$	$xy$

b)

a)

b)

$m_0$	$m_1$	$m_3$	$m_2$
$m_4$	$m_5$	$m_7$	$m_6$

a)

		$y$			
		$yz$	$00$	$01$	$11$
$x$	$0$	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	$1$	$xy'z'$	$xy'z$	$xyz$	$xyz'$
		$z$			

b)

a)

b)

$m_0$	$m_1$	$m_3$	$m_2$
$m_4$	$m_5$	$m_7$	$m_6$
$m_{12}$	$m_{13}$	$m_{15}$	$m_{14}$
$m_8$	$m_9$	$m_{11}$	$m_{10}$

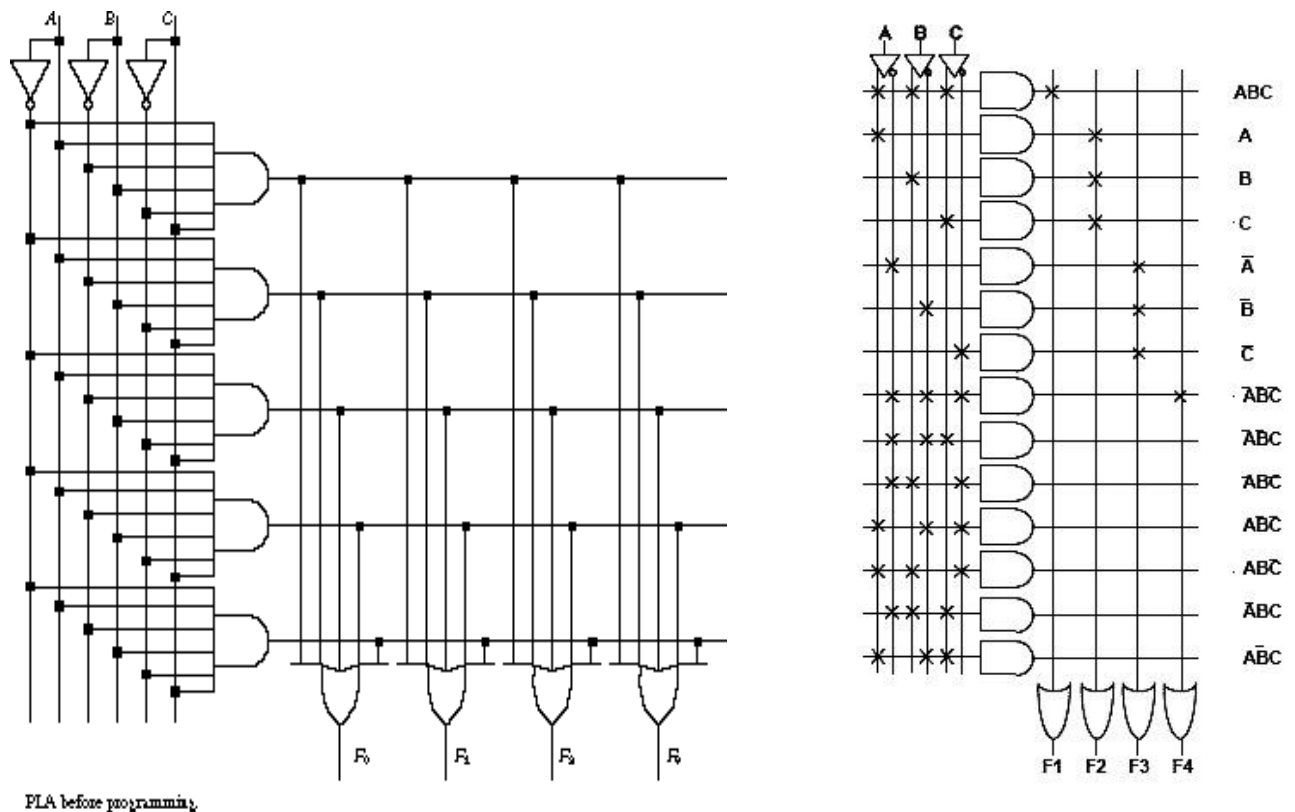
a)

		$yz$		$y$	
		00	01	11	10
$w$	$wx$	00	01	11	10
	00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
	01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
	11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
	10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

b)

a)

b)

**Programmable Logic Array (PLA):****Ejercicio 1:**

Un detector de paridad impar de 4 entradas y una salida funciona de la siguiente manera: si la cantidad de entradas con valor '1' es impar la salida se pone en '1', en el resto de los casos la salida toma valor '0'.

- Construir la tabla de verdad para dicho sistema.
- Obtener la ecuación lógica como suma de minterminos y producto de maxiterminos (funciones canónicas).
- Implementar el sistema con compuertas NAND de la cantidad de entradas requeridas.
- Implementar el sistema con una PLA.

**Ejercicio 2:**

Un sistema digital recibe información en forma de palabras de 5 bits (**ABCDE**) en un código protegido contra errores, de tal forma que cualquier dato que se reciba debe contener 3 y sólo 3 bits en '1'. Diseñar un circuito con las entradas **ABCDE** y una salida **err** que se activa por bajo cuando se recibe un dato incorrecto.

- Construir la tabla de verdad para dicho sistema.
- Obtener la ecuación lógica como suma de minterminos y producto de maxiterminos (funciones canónicas).
- Implementar el sistema con una PLA.

**Ejercicio 3:**

Verificar los resultados obtenidos de cada función lógica en la Guía 2 - Ejercicio 1, mediante la utilización de mapas de Karnaugh, el cual garantiza la obtención de la mínima expresión.

- $x.y + x.y'$
- $(x + y).(x + y')$
- $x.y.z + x'.y + xyz'$
- $z.x + z.x'.y$
- $(A + B).(A' + B')$
- $y.(w.z' + w.z) + x.y$

**Ejercicio 4:**

Dadas la siguientes tablas de verdad para las funciones Fx:

 $(f_1)$ 

x3	x2	x1	x0	F(x3,x2,x1,x0)
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

 $(f_2)$ 

x3	x2	x1	x0	F(x3,x2,x1,x0)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
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	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

 $(f_3)$ 

x2	x1	x0	F(x2,x1,x0)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- Encontrar las expresiones canónicas de cada Fx como suma de minitérminos y como producto de maxitérminos.
- Encontrar la expresión minimizada de cada Fx utilizando mapas de Karnaugh.

**Ejercicio 5:**

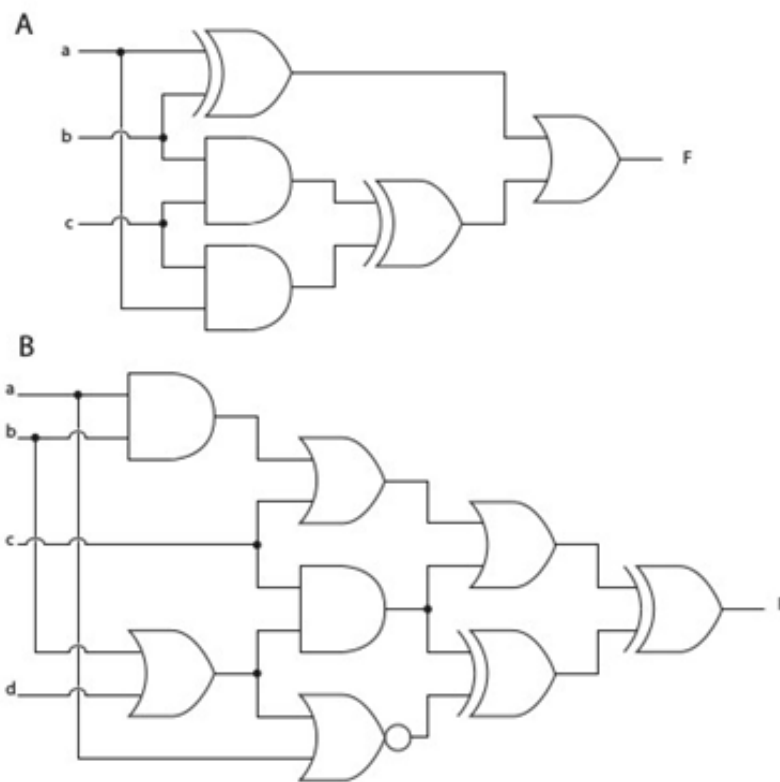
Un circuito combinacional comparador toma dos números de 2 bits,  $\mathbf{A} = (A_1, A_0)$  y  $\mathbf{B} = (B_1, B_0)$  y retorna tres salidas (" $\mathbf{A} > \mathbf{B}$ ", " $\mathbf{A} = \mathbf{B}$ " y " $\mathbf{A} < \mathbf{B}$ ") de 1 bit cada una.

Ej: si  $\mathbf{A} = (00)$  y  $\mathbf{B} = (10)$ , entonces " $\mathbf{A} > \mathbf{B}$ " = '0', " $\mathbf{A} = \mathbf{B}$ " = '0' y " $\mathbf{A} < \mathbf{B}$ " = '1'.

- Construir la tabla de verdad para dicho sistema.
- Obtener la ecuación lógica como suma de minterminos y producto de maxiterminos.
- Encontrar la función minimizada de cada salida como suma de productos usando mapas de Karnaugh.
- Implementar el sistema con compuertas lógicas básicas.

**Ejercicio 6:**

Analizar los circuitos de lógica combinacional de la figura. Para cada uno:



- Escribir la función booleana correspondiente.
- Encontrar la tabla de verdad para la función obtenida.
- Obtener la función minimizada como suma de productos a partir del mapa de Karnaugh.
- Dibujar el circuito de lógica combinacional resultante del punto (c).

**Ejercicio 7:**

Un DECODIFICADOR es un circuito combinacional que convierte información binaria de 'N' entradas codificadas ( $\mathbf{A}$ ), a ' $2^N$ ' salidas únicas ( $\mathbf{X}$ ). Esto quiere decir que sólo una salida  $\mathbf{X}$  está activa y representa el valor de las señales de entrada  $\mathbf{A}$ .

Considere un decodificador activo por bajo (salida activa = '0') con  $N=2$  y  $2^N=4$  (deco 2 x 4).

- Expresar las tablas de verdad de las cuatro salidas  $X_0$ ,  $X_1$ ,  $X_2$  y  $X_3$ .
- Encontrar las expresiones de  $X_0$ ,  $X_1$ ,  $X_2$  y  $X_3$  como suma de minterminos y como producto de maxiterminos.
- Encontrar expresiones minimizadas de  $X_0$ ,  $X_1$ ,  $X_2$  y  $X_3$  utilizando el método de Karnaugh o un método algebraico.
- Implementar las expresiones anteriores a través del uso de compuertas lógicas.
- Repetir el punto (d) agregando una entrada de HABILITACIÓN (**E**) activa por bajo, de tal forma que cuando **E**='1' ninguna señal de salida permanezca habilitada.

#### Ejercicio 8:

Implementar un decodificador de 3 x 8 y otro de 4 x 16 a partir de decodificadores 2 x 4 activos por bajo, con entrada de habilitación (**E**) activa por bajo y compuertas lógicas.

#### Ejercicio 9:

- Diseñar un circuito SUMADOR COMPLETO (3 entradas: **X**, **Y**, **C<sub>IN</sub>**; 2 salidas: **S**, **C<sub>OUT</sub>**) mediante el uso de un Decodificador de salida activa por alto y compuertas OR. Tip: La salida que vale 1 representa el mintermino equivalente al número binario que está a la entrada.
- Diseñar un sumador completo usando dos semisumadores y una compuerta.

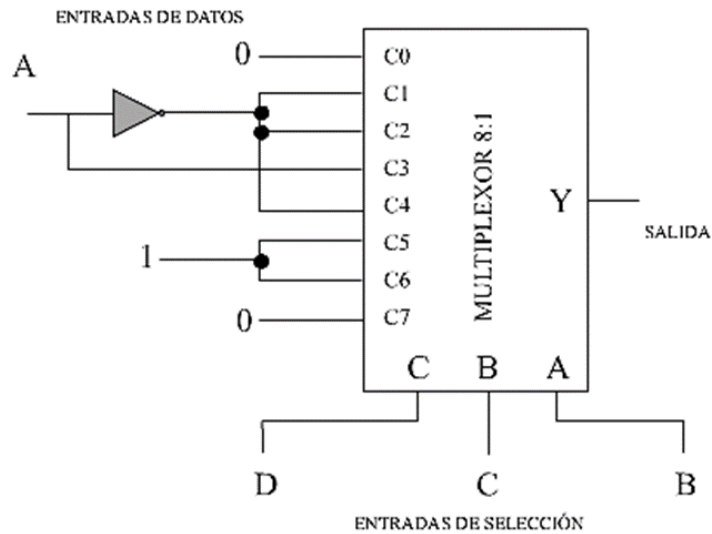
#### Ejercicio 10:

Un MULTIPLEXOR (MUX) es un circuito combinacional que selecciona información binaria de muchas entradas y la dirige a una única salida (**Y**), conforme al estado de las señales de selección. Si un MUX posee ' $2^N$ ' entradas de información (**D**) requiere ' $N$ ' señales de selección (**S**).

- Expresar la tabla de verdad de un MUX de 2 entradas (y una salida) y su implementación mediante el uso de compuertas lógicas (AND, OR, NOT, NOR, NAND, etc.)
- Mostrar cómo se puede usar un MUX para obtener una compuerta NOT.
- ¿Cómo obtener un MUX de 4 entradas (y una salida) en base a multiplexores de 2 entradas?
- ¿Cómo obtener un multiplexor de ' $N$ ' entradas con multiplexores de 2 entradas?

**Ejercicio 11:**

Dado el siguiente circuito con entradas (DCBA):

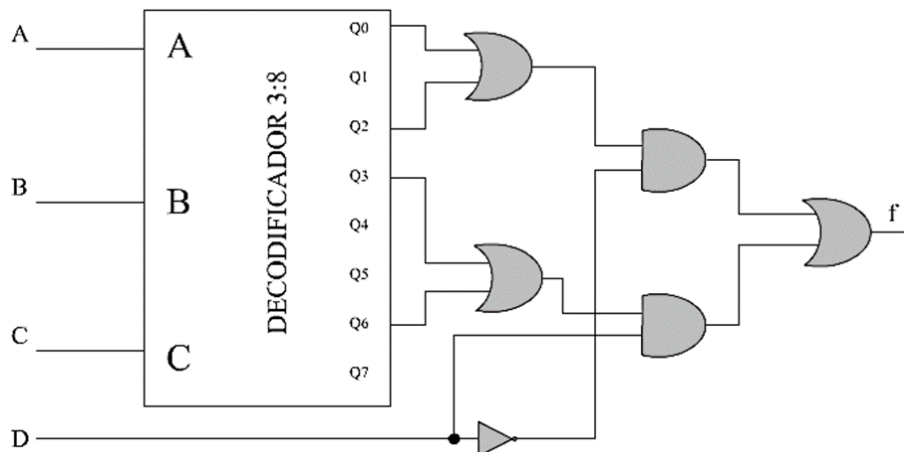


- a. Elabore la tabla de verdad de "SALIDA" y responda: ¿Para cuántas combinaciones posibles de entradas la salida es igual a 1?

ATENCIÓN: no confundir el nombre de las señales del enunciado con las de selección del decodificador!

**Ejercicio 12:**

Dado el siguiente circuito con entradas (DCBA):

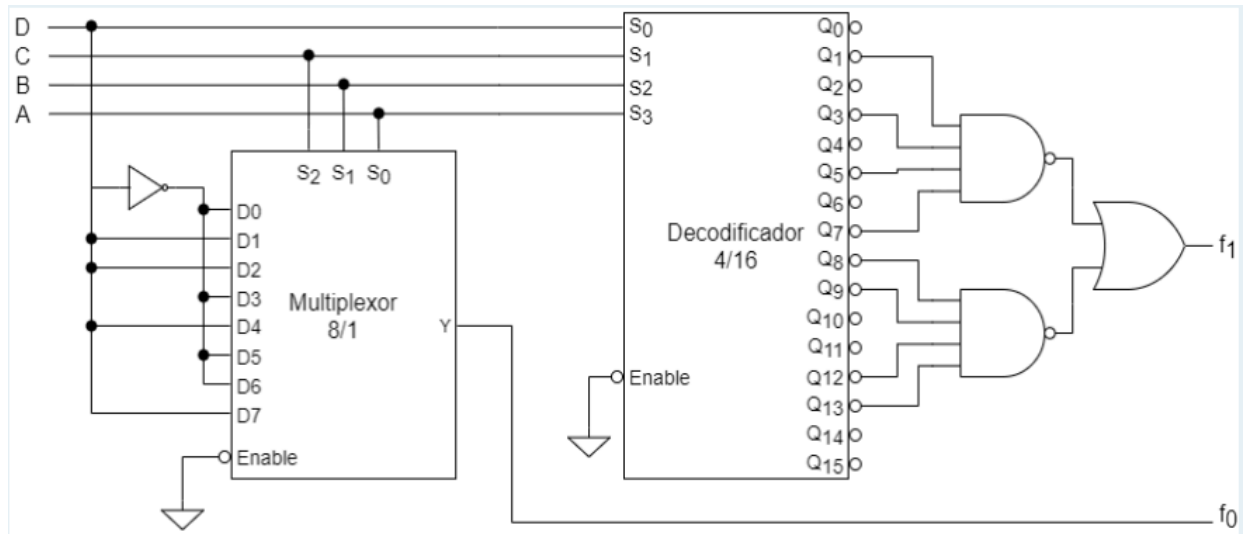


Asumir que las salidas Q del decodificador son activas por alto, y que las salidas Q(1,4,5,7) no se conectan a ninguna compuerta.

- a. a.Elaborar la tabla de verdad de la función f y responder: ¿Para cuántas combinaciones posibles de entradas la salida f es igual a 1?

**Ejercicio 13:**

Dado el siguiente circuito con entradas (ABCD):



- Elaborar la tabla de verdad de la función  $f$  y responder: ¿Para cuántas combinaciones posibles de entradas la salida  $f$  es igual a 1?
- Repetir el punto a) pero conectando la salida  $Y$  del Multiplexor a la entrada Enable del Decodificador. ¿Qué diferencias se obtuvieron? ¿Por qué?