TEC.AZUAY
INSTITUTO UNIVERSITARIO

ALGEBRA BOOLEANA

UNIDAD 3

Mgtr. Vilma Duchi F

Compuertas Lógicas

- **Definición:** Las compuertas lógicas son dispositivos que operan con aquellos estados lógicos y funcionan igual que una calculadora, de un lado ingresas los datos, ésta realiza una operación, y finalmente, te muestra el resultado.

Cada una de las compuertas lógicas se las representa mediante un **Símbolo**, y la operación que realiza **(Operación lógica)** se corresponde con una tabla, llamada **Tabla de Verdad.**

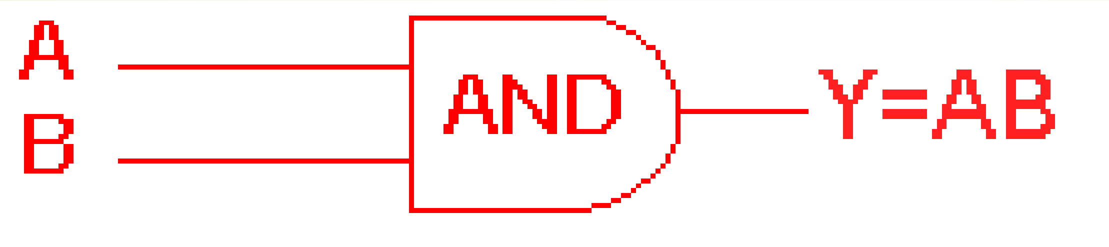
Puerta	Función	Descripción
NOT	$\sim A$	Complemento
AND	$A \bullet B$	Producto
OR	$A + B$	Suma
XOR	$A \bullet \sim B + \sim A \bullet B$	Suma exclusiva
NAND	$\sim(A \bullet B)$	Complemento del producto
NOR	$\sim(A + B)$	Complemento de la suma

4.1.1 COMPUERTA LOGICA AND

La operación realizada por la compuerta AND es el PRODUCTO LOGICO. Una compuerta AND de dos o más entradas entrega un nivel alto o 1 lógico en su salida cuando todas sus entradas están en nivel alto y un nivel bajo o 0 lógico, cuando por lo menos una de ellas, o todas, están en nivel bajo.

En el circuito eléctrico permite entender bien el funcionamiento de este tipo de puerta si consideramos que una corriente eléctrica pasa por el circuito formado por dos interruptores si ambos están cerrados (A y B valen 1).

Compuerta: AND



Valor de Y:

*	1	0
1	1	0
0	0	0

Tabla de Verdad:

A	B	Y=AB
0	0	0
0	1	0
1	0	0
1	1	1

4.1.2 COMPUERTA LOGICA OR

La operación realizada por la compuerta OR es la SUMA LOGICA. Una compuerta OR de dos o más entradas entrega un nivel bajo en una salida cuando todas sus entradas están en nivel bajo y un nivel alto cuando por lo menos una de ellas, o todas, están en nivel alto.

La representación del circuito eléctrico permite entender bien el funcionamiento si consideramos que el circuito dejará pasar la corriente eléctrica si cualquiera de los interruptores está cerrado.

Compuerta: **OR**

Valor de Y:

+	1	0
1	1	1
0	1	0

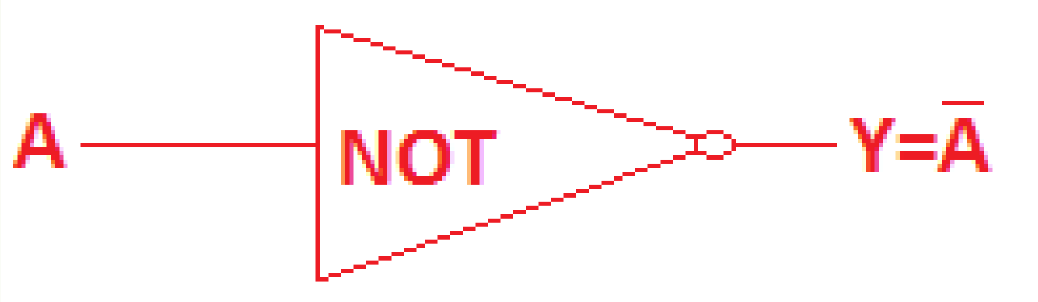
Tabla de Verdad:

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

4.1.3 COMPUERTA LOGICA NOT

Un inversor, niega o complementa el nivel lógico de la señal de entrada.

Compuerta: NOT



Valor de Y:

A	Salida
0	1
1	0

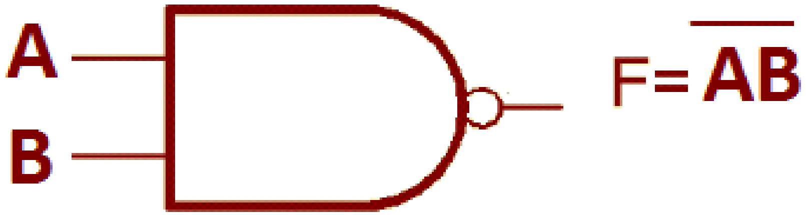
4.1.4 COMPUERTA LOGICA NAND

Una compuerta lógica NAND opera de forma contraria a una compuerta AND, entregando una salida baja cuando todas sus entradas son altas y una salida alta mientras exista por lo menos un bajo en cualquiera de ellas.

Las compuertas NAND son el complemento lógico de las compuertas AND. La expresión asociada a la función que realiza una compuerta XNOR se escribe como $Q = \overline{A \cdot B}$.
El valor de salida de una compuerta NAND será 1 cuando el valor de algunas de sus entradas sean 0 y será 0 cuando el valor de todas sus entradas sea 1. La siguiente figura interactiva muestra un diagrama de una compuerta NAND con su correspondiente tabla de verdad:

Compuerta:

NAND



Tablas de verdad: Para 2 y 3 variables

A	B	$Y=\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

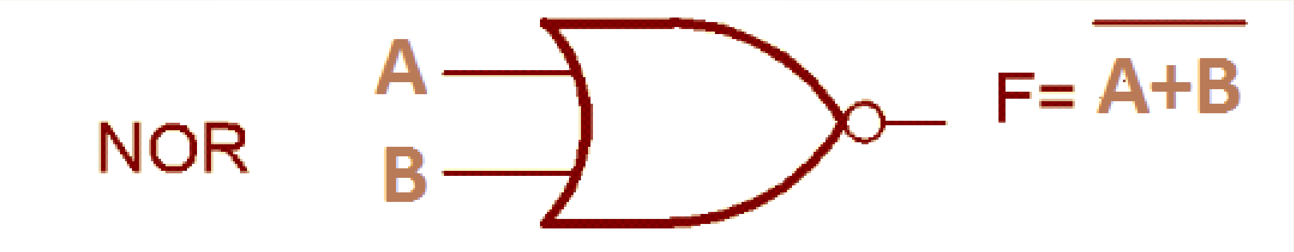
A	B	C	$Y=\overline{ABC}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

4.1.5 COMPUERTA LOGICA NOR

Una compuerta lógica NOR opera de forma contraria a una compuerta OR, entregando una salida alta cuando todas sus entradas son bajas y una salida baja cuando exista por lo menos un alto en cualquiera de ellas.

Las compuertas NOR son el complemento lógico de las compuertas OR. La expresión asociada a la función que realiza una compuerta XNOR se escribe como $Q = \overline{A + B}$.
El valor de salida de una compuerta NOR será 1 cuando el valor de todas sus entradas sean 0 y será 0 cuando el valor de alguna de sus entradas sea 1. La siguiente figura interactiva muestra un diagrama de una compuerta NOR con su correspondiente tabla de verdad:

Compuerta:



Tablas de Verdad: para dos y tres variables

A	B	$Y=\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

A	B	C	$Y=\overline{A + B + C}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

4.1.6 COMPUERTA XOR

Una compuerta OR exclusiva o XOR es un dispositivo digital con dos líneas de entrada y una línea de salida, que entrega una salida alta cuando una de sus entradas es baja y la otra alta, y una salida baja cuando sus entradas son ambas altas o ambas bajas.

Compuerta Lógica XOR

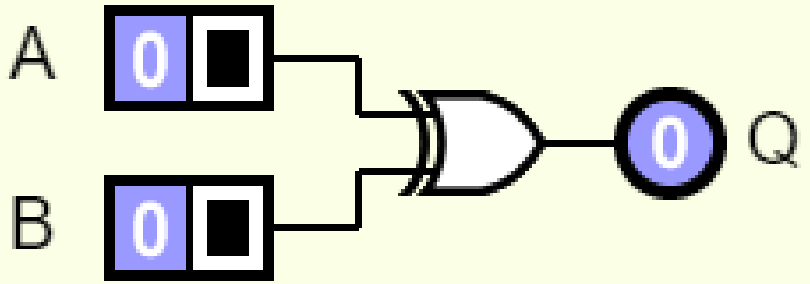


Tabla de Verdad

A	B	Q
0	0	0
1	0	1
0	1	1
1	1	0

Figura 4. Compuerta Lógica XOR y su tabla de verdad.

4.1.7 COMPUERTA LOGICA XNOR

Una compuerta NOR exclusiva o XNOR opera en forma exactamente opuesta a una compuerta XOR, entregando una salida baja cuando una de sus entradas es baja y la otra alta, y una salida alta cuando sus entradas son ambas altas o ambas bajas.

Las compuertas XNOR son el complemento lógico de las compuertas XOR. La expresión asociada a la función que realiza una compuerta XNOR se escribe como $Q = \overline{A \oplus B}$.

El valor de salida de una compuerta XNOR será 1 cuando el valor de sus entradas sean iguales y será 0 cuando el valor de sus entradas sean distintas. La siguiente figura interactiva muestra un diagrama de una compuerta XNOR con su correspondiente tabla de verdad:

Compuerta Lógica XNOR

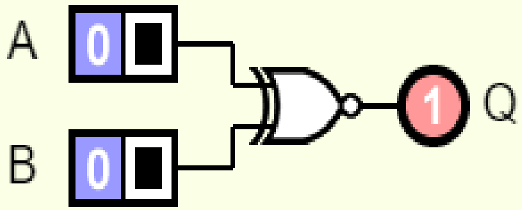


Tabla de Verdad

A	B	Q
0	0	1
1	0	0
0	1	0
1	1	1

Figura 5. Compuerta Lógica XNOR y su tabla de verdad.

TEOREMAS Y POSTULADOS DEL ALGEBRA DE BOOLE

Un sistema de elementos B y dos operaciones binarias cerradas (·) y (+) se denomina ALGEBRA de BOOLE siempre y cuando se cumplan las siguientes propiedades:

1. Propiedad conmutativa:

$$A + B = B + A$$
$$A \cdot B = B \cdot A$$

2. Propiedad distributiva:

$$A \cdot (B+C) = A \cdot B + A \cdot C$$
$$A + B \cdot C = (A+B) \cdot (A+C)$$

3. Elementos neutros diferentes

$$A + 0 = A$$
$$A \cdot 1 = A$$

4. Siempre existe el complemento de A, denominado A’

$$A + A' = 1$$
$$A \cdot A' = 0$$

PRINCIPIO DE DUALIDAD

Cualquier teorema o identidad algebraica deducible de los postulados anteriores puede transformarse en un segundo teorema o identidad válida sin más que intercambiar (+) por (·) y 1 por 0.

CONSTANTE: cualquier elemento del conjunto B

VARIABLE: símbolo que representa un elemento arbitrario del álgebra, ya sea constante o fórmula completa.

TEOREMAS:

Teorema 1: el elemento complemento A' es único.
Teorema de los elementos nulos: para cada elemento de B se verifica:

$$A+1 = 1$$
$$A \cdot 0 = 0$$

PRINCIPIO DE DUALIDAD

Teorema de idempotencia: para cada elemento de B, se verifica:

$$A+A=A$$
$$A\cdot A=A$$

Teorema de involución: para cada elemento de B, se verifica:

$$(A')' = A$$

Teorema de absorción: para cada par de elementos de B, se verifica:

$$A+A\cdot B=A$$
$$A\cdot(A+B) =A$$

Teorema 7: para cada par de elementos de B, se verifica:

$$A + A'\cdot B = A + B$$
$$A \cdot (A' + B) = A \cdot B$$

LEYES DE DEMORGAN: para cada par de elementos de B, se verifica:

$$(A+B)' = A'\cdot B'$$
$$(A\cdot B)' = A' + B'$$

Teorema de asociatividad: cada uno de los operadores binarios (+) y (·) cumple la propiedad asociativa:

$$A+(B+C) = (A+B) +C$$
$$A\cdot(B\cdot C) = (A\cdot B) \cdot C$$

RESUMEN

Postulados del Álgebra de Boole:

$$P1: \quad conmutativa \begin{cases} a+b=b+a \\ a\cdot b=b\cdot a \end{cases}$$

$$P2: \quad elementos\ \ neutros \begin{cases} 0+a=a \\ 1\cdot a=a \end{cases}$$

$$P3: \quad distributiva \begin{cases} a\cdot(b+c)=a\cdot b+a\cdot c \\ a+(b\cdot c)=(a+b)\cdot(a+c) \end{cases}$$

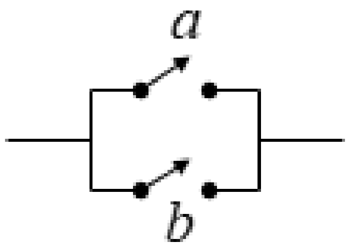
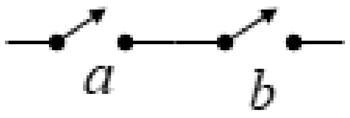
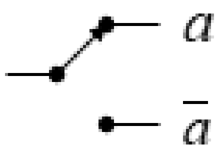
$$P4: \quad complementación \begin{cases} a+\overline{a}=1 \\ a\cdot\overline{a}=0 \end{cases}$$

PROPIEDADES Y TEOREMAS

6	Asociativa	$a + (b + c) = (a + b) + c$	$a.(b.c) = (a.b).c$
7	Distributiva	$a.b + a.c = a(b + c)$	$(a + b).(a + c) =$ $= a + (b.c)$
8	Absorción	$a + a.b = a$	$a.(a + b) = a$
9	Consenso	$a + \bar{a}.b = a + b$	$a.(\bar{a} + b) = a.b$
10	De Morgan	$\overline{a + b} = \bar{a}.\bar{b}$	$\overline{a.b} = \bar{a} + \bar{b}$
11	Involución ó doble negación	$\bar{\bar{a}} = a$	

RESUMEN

Tabla Resumen:

Boole	*Conjunto*	*Conmutación*	
+	$\cup$	paralelo	
$\cdot$	$\cap$	serie	
0	$\varnothing$	siempre abierto	
1	Universo	siempre cerrado	
$\bar{a}$	Lo contrario de a		

RESUMEN

Teoremas del Álgebra de Boole:

$$T1: \quad dualidad \begin{cases} + \leftrightarrow \cdot \\ 0 \leftrightarrow 1 \end{cases}$$

$$T2: \quad idempotencia \begin{cases} a + a = a \\ a \cdot a = a \end{cases}$$

$$T3: \quad absorción \begin{cases} a + a \cdot b = a \\ a \cdot (a + b) = a \end{cases}$$

$$T4: \quad asociativa \begin{cases} a + b + c = a + (b + c) = (a + b) + c \\ a \cdot b \cdot c = a \cdot (b \cdot c) = (a \cdot b) \cdot c \end{cases}$$

EJERCICIOS

Simplifica la expresión booleana $\overline{(\overline{A}+B)\cdot(\overline{B}+C)}$.

SOLUCIÓN

Aplique el teorema de Morgan $\overline{X\cdot Y}=\overline{X}+\overline{Y}$ con $X=\overline{A}+B$ y $Y=\overline{B}+C$:

$$\left(\overline{(\overline{A}+B)\cdot(\overline{B}+C)}\right)=\left(\overline{\overline{A}+B}+\overline{\overline{B}+C}\right)$$

Aplique el teorema de Morgan $\overline{X+Y}=\overline{X}\cdot\overline{Y}$ con $X=\overline{A}$ y $Y=B$:

$$\left(\overline{\overline{A}+B}\right)+\overline{\overline{B}+C}=\left(\overline{\overline{A}}\cdot\overline{B}\right)+\overline{\overline{B}+C}$$

Aplique la ley de la doble negación (involución) $\overline{\overline{X}}=X$ con $X=A$:

$$\left(\left(\overline{\overline{A}}\right)\cdot\overline{B}\right)+\overline{\overline{B}+C}=((A)\cdot\overline{B})+\overline{\overline{B}+C}$$

Aplique el teorema de Morgan $\overline{X+Y}=\overline{X}\cdot\overline{Y}$ con $X=\overline{B}$ y $Y=C$:

$$(A\cdot\overline{B})+\left(\overline{\overline{B}+C}\right)=(A\cdot\overline{B})+\left(\overline{\overline{B}}\cdot\overline{C}\right)$$

Aplique la ley de la doble negación (involución) $\overline{\overline{X}}=X$ con $X=B$:

$$(A\cdot\overline{B})+\left(\left(\overline{\overline{B}}\right)\cdot\overline{C}\right)=(A\cdot\overline{B})+((B)\cdot\overline{C})$$

RESPUESTA

$$\overline{(\overline{A}+B)\cdot(\overline{B}+C)}=(A\cdot\overline{B})+(B\cdot\overline{C})$$

EJERCICIOS

Practiquemos

Expresión:

$$\overline{(A+\overline{B}\cdot C)}+(\overline{A}\cdot B\cdot \overline{C})$$

Expresión:

$$(A\cdot B\cdot \overline{C})+(A\cdot B\cdot C)+(A\cdot \overline{B}\cdot \overline{C})+(\overline{A}\cdot B\cdot \overline{C})$$

$$\overline{\bar{A}(C+D)+\bar{B}(A+D)+\bar{A}\cdot\bar{B}\cdot\bar{C}}$$

$$A(\bar{C}+B\bar{D}+DE)+D(BC+\bar{A}+B)+\bar{B}[A(E+CE)+A\bar{C}\bar{D}E]$$

Simplificar, utilizando los teoremas del álgebra de Boole:

$$\overline{\bar{A}(C+D)+\bar{B}(A+D)+\bar{A}\cdot\bar{B}\cdot\bar{C}}$$

$$A(\bar{C}+B\bar{D}+DE)+D(BC+\bar{A}+B)+\bar{B}[A(E+CE)+A\bar{C}\bar{D}E]$$

$$A(\bar{C} + B\bar{D} + DE) + D(BC + \bar{A} + B) + \bar{B}[A(E + CE) + A\bar{C}\bar{D}E] =$$
$$= A(\bar{C} + B\bar{D} + DE) + D(\bar{A} + B) + \bar{B}[A(E + C) + A\bar{C}\bar{D}E] =$$
$$= A(\bar{C} + B\bar{D} + DE) + D(\bar{A} + B) + \bar{B}(AE + AC) =$$
$$= A\bar{C} + AB\bar{D} + ADE + \bar{A}D + BD + A\bar{B}E + A\bar{B}C =$$
$$= A\bar{C} + B(A\bar{D} + D) + ADE + \bar{A}D + A\bar{B}E + A\bar{B}C =$$
$$= A\bar{C} + B(D + A)ADE + A\bar{D} + A\bar{B}E + A\bar{B}C =$$
$$= A\bar{C} + BD + BA + ADE + \bar{A}D + A\bar{B}E + A\bar{B}C =$$
$$= A\bar{C} + AB + A\bar{D} + ADE + A\bar{B}E + A\bar{B}C =$$
$$= A(\bar{C} + \bar{B}C) + A(B + \bar{B}E) + D(\bar{A} + AE) =$$
$$= A(\bar{C} + \bar{B}) + A(B + E) + D(\bar{A} + E) =$$
$$= A\bar{C} + A\bar{B} + AB + AE + \bar{A}D + DE =$$
$$= A\bar{C} + A + AE + \bar{A}D + DE =$$
$$= A + \bar{A}D + DE = A + D + DE =$$
$$= A + D(1 + E) = A + D$$

REPRESENTACIÓN DE FUNCIONES LÓGICAS

En los sistemas digitales se va a trabajar fundamentalmente con funciones lógicas (o booleana) que son
combinaciones de variables (que pueden tomar el valor 0, 1) y operadores +, ·, ¯ ó ‘.La forma habitual de trabajar a la hora diseñar o analizar un circuito digital es considerar que el circuito tiene:
Una o más entradas, que se suelen denominar con letras: a,b,c,… (a veces se usan mayúsculas A,B,C… o se usan W,X,Y,Z).
Una única salida, que se suele denominar como F, Z , S ó O dependiendo del autor y el contexto

REPRESENTACIÓN ALGEBRAICA

La notación algebraica es la que se compone de la escritura de variables y operadores del algebra booleana.

A continuación, se muestran dos ejemplos, uno para una función con 3 variables de entrada y otro para 4 variables de entrada:

$F = ABC' + (A+B'+C)' + AC'$

$Z = xy' + (y+w') + (z'+w)'$

Obsérvese dos detalles importantes:

Las funciones reflejan la estructura básica de un circuito digital, es decir, hay una salida (F, Z en este caso), que es resultado una serie de operaciones a un conjunto de entradas.

A priori, no se puede asegurar que la función sea la más simple posible (con menos términos y variables).

Para generar las tablas de verdad a partir de las funciones es importante identificar cuantas variables tenemos con ello se podrá calcular el número de filas..

Ejercicios

Realice las tablas de verdad para las siguientes funciones de salida.

$$F = A.B + A.\overline{B}$$

$$F = A \cdot B + C \cdot \overline{B}$$

$$F = \overline{\overline{(A + B)} + C}$$

$$Z = \overline{\overline{(A + B)} + \overline{(\overline{B} + C)} + \overline{(B + \overline{C})}}$$

$$Z = A.\overline{B}.\overline{C} + \overline{A}\overline{B}.C + A.\overline{B}.C + \overline{A}.B.C$$

$$F = A.B.C + \overline{A}.B.C + B.C$$

$$Z = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

REPRESENTACIÓN NUMÉRICA

Antes de entrar a explicar esta forma de representación es necesario explicar qué es una expresión canónica ya que esta representación es una forma simplificada de representar **expresiones canónicas.**

Una expresión canónica no es más que aquella que contiene en cada uno de sus términos todas las variables de entrada. Por ejemplo, si se tiene un circuito de **cuatro variables A, B, C, D:**

- ABCD es una expresión canónica
- ABC no es una expresión canónica

Sin embargo, es fácil encontrar la expresión canónica equivalente. Por ejemplo, en el caso anterior podríamos sustituir ABC por ABCD + ABCD’ sin que varié el resultado.

Esta última una expresión ABCD + ABCD’ es canónica ya que sus dos términos contienen las cuatro variables. Se puede comprobar que si se aplica la ley de absorción a ABCD + ABCD’ la variable D que está en ambos términos es absorbida por el conjunto ABC. Este resultado ABC es precisamente el término original por lo que se constata que efectivamente ABC = ABCD + ABCD’.

Expresión en forma de “suma de productos”

Cualquier función se puede expresar en forma canónica y se puede pasar muy fácilmente de la tabla de verdad a la forma canónica. Por ejemplo, la tabla de verdad anterior se puede expresar como:

F = A’BC’ + A’BC + A’BC’ + ABC’

Para hacer esta conversión el procedimiento es el siguiente:

1. Se toman aquellas combinaciones de variables de entrada que dan lugar a una salida de valor igual a 1
2. Cada combinación da lugar a un término en forma de producto de las variables de entrada.

Para construir el término se observan los valores de las variables de entrada:

- Si el valor es 1 se escribe la variable tal cual (sin negar).
- Si el valor es 0 se escribe la variable negada.

3. Cada uno de los términos anteriores se concatenan mediante la operación **suma**.

Expresión en forma de “producto de sumas”

Existe una forma alternativa de expresarlo y para la tabla de verdad del ejemplo sería:

$F = (A+B+C) \cdot (A + B + C') \cdot (A + B' + C') \cdot (A' + B' + C')$

Para hacer esta conversión el procedimiento es el siguiente:

1. Se toman aquellas combinaciones de variables de entrada que dan lugar a una salida de valor 0.

2. Cada combinación da lugar a un término en forma de suma de las variables de entrada. Para construir el término se observan los valores de la variable de entrada:

- Si el valor es 0 se escribe la variable tal cual (sin negar).
- Si el valor es 1 se escribe la variable negada.

3. Cada uno de los términos anteriores se concatenan mediante la operación producto (·).

Expresiones Numéricas

Una vez comprendido lo que es una expresión canónica y la expresión en forma de producto de sumas y suma de productos, se puede representar la función descrita mediante la tabla de verdad anterior de la siguiente forma:

F = Σ (2, 4, 5, 6)

Donde,

- **F** es la función de salida que ya se ha indicado anteriormente
- **Σ** es la letra griega sigma que se suele usar también en matemáticas para indicar un sumatorio.
- **2, 4, 5, 6** son los valores numéricos en decimal de los términos que en la tabla de verdad valen 1, es decir los correspondientes a la **suma de productos** o minitérminos.

De forma similar, se puede hacer uso de la expresión de producto de sumas para representar la función descrita mediante la tabla de verdad anterior mediante la forma:

F = Π (0, 1, 3, 7)

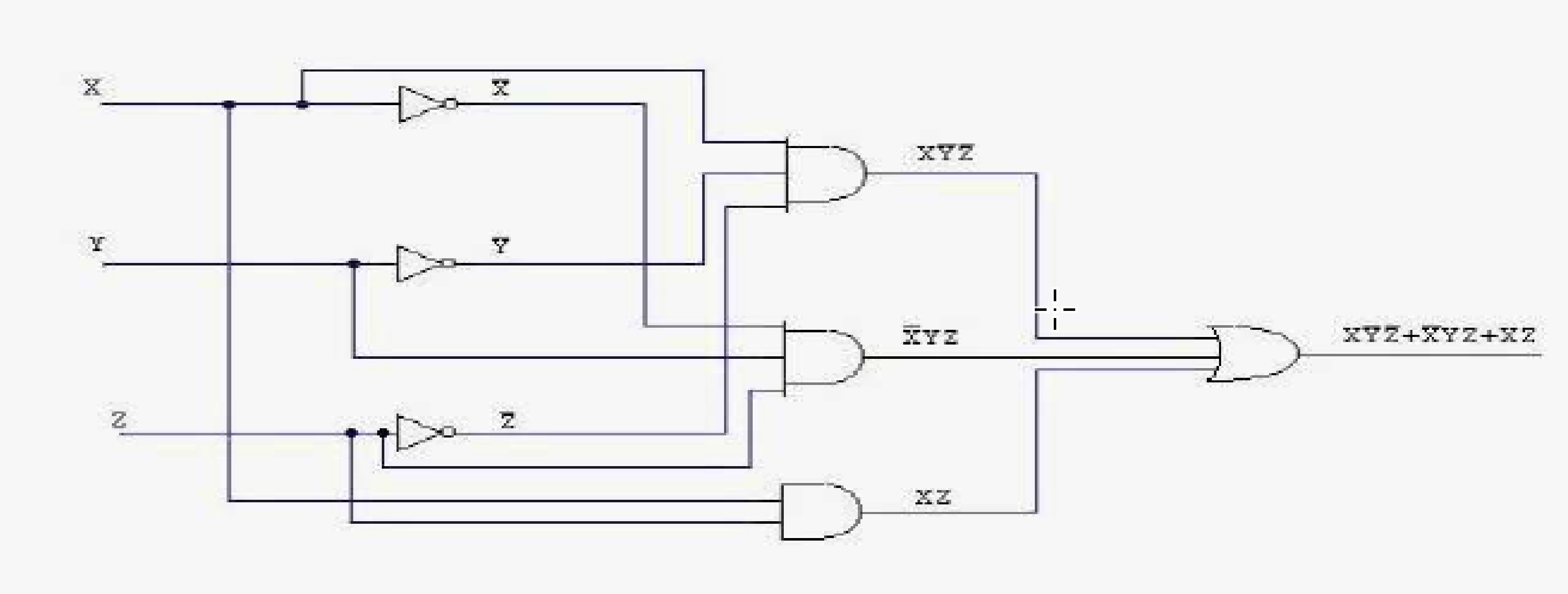
Donde,

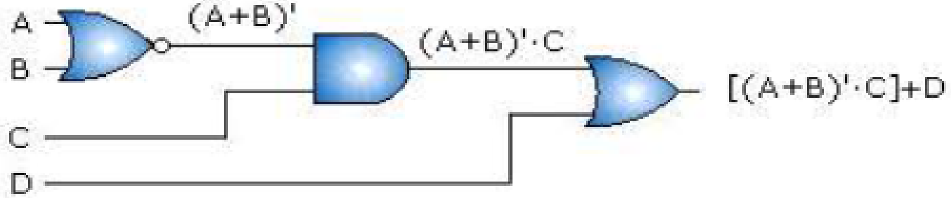
- **F** es la función de salida que ya se ha indicado anteriormente
- **Π** es la letra griega pi mayúscula que se suele usar también en matemáticas para indicar un productorio
- **0, 1, 3, 7** son los valores numéricos en decimal de los términos que en la tabla de verdad valen 0, es decir los correspondientes al **producto de sumas** o maxitérminos

4.3.5 REPRESENTACIÓN GRÁFICA

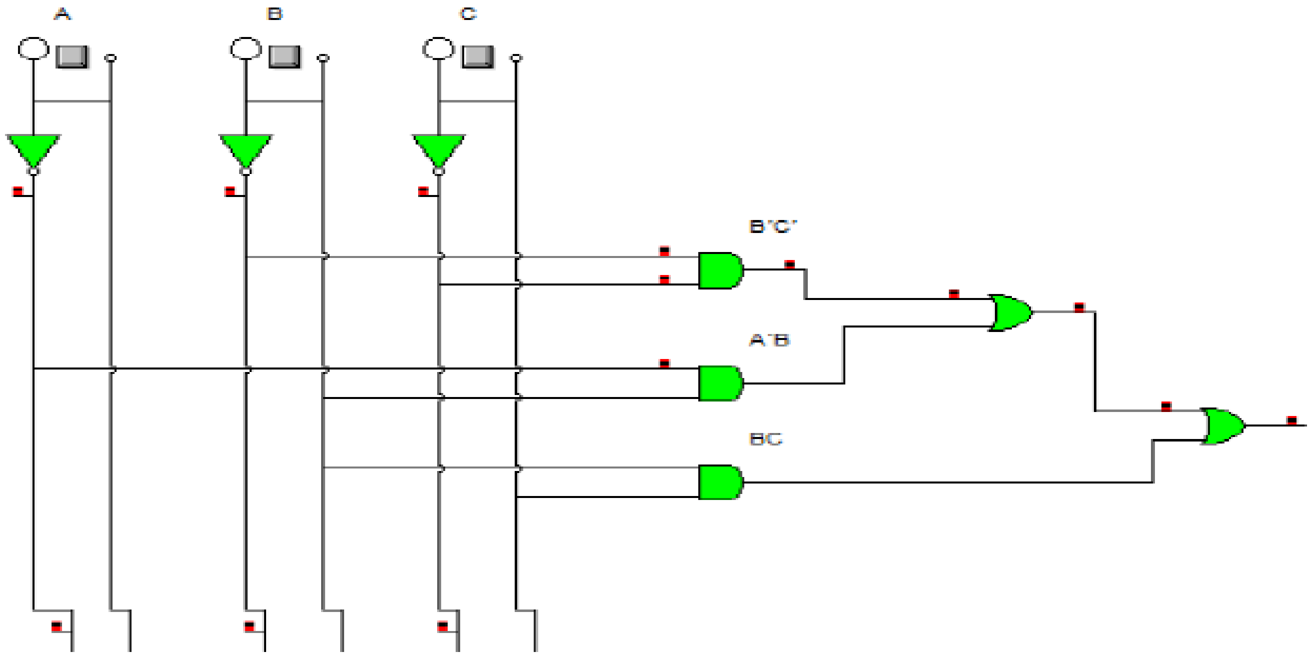
La representación gráfica es la que se utiliza en circuitos y esquemas electrónicos. En la siguiente figura se representan gráficamente dos funciones algebraicas, una con símbolos no normalizados, superior, y la otra con normalizados, inferior (véanse los símbolos de las puertas lógicas).

EJERCICIOS RESUELTOS



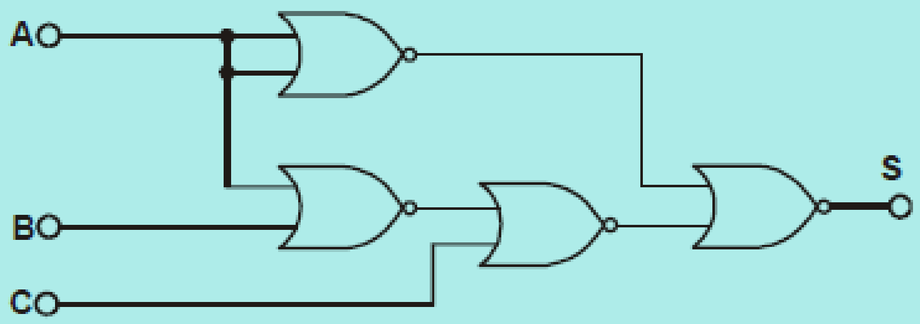


$\bar{B}\bar{C} + \bar{A}B + BC$



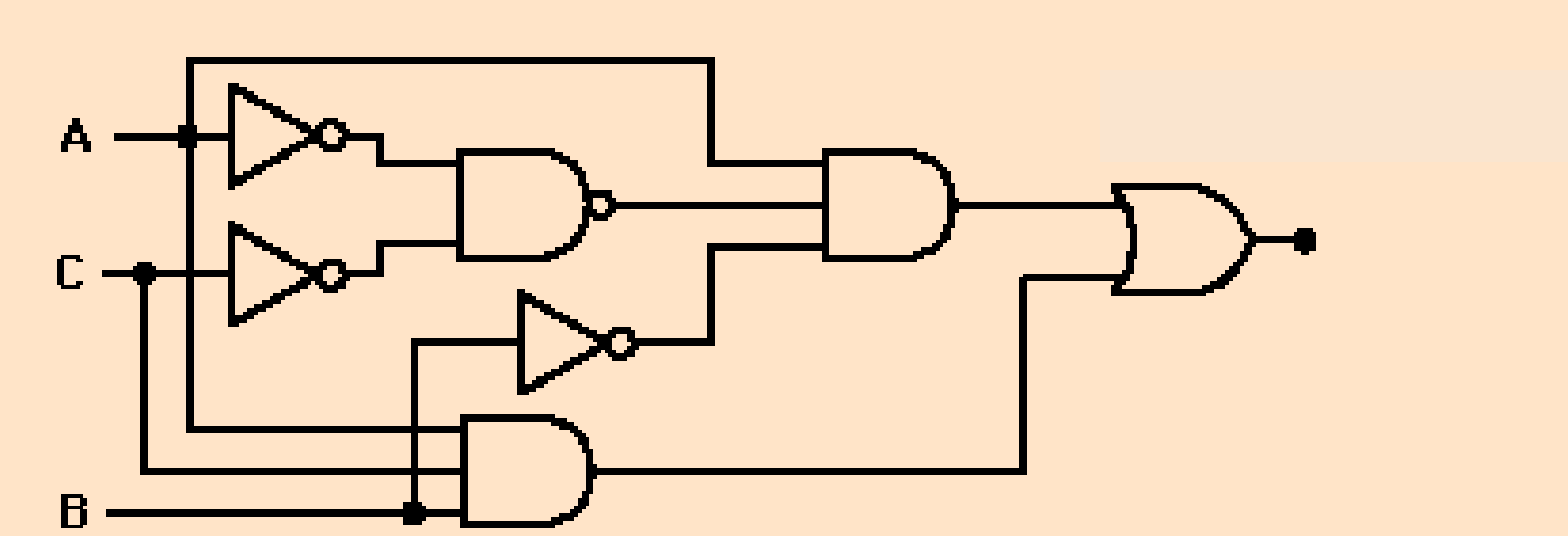
ACTIVIDAD

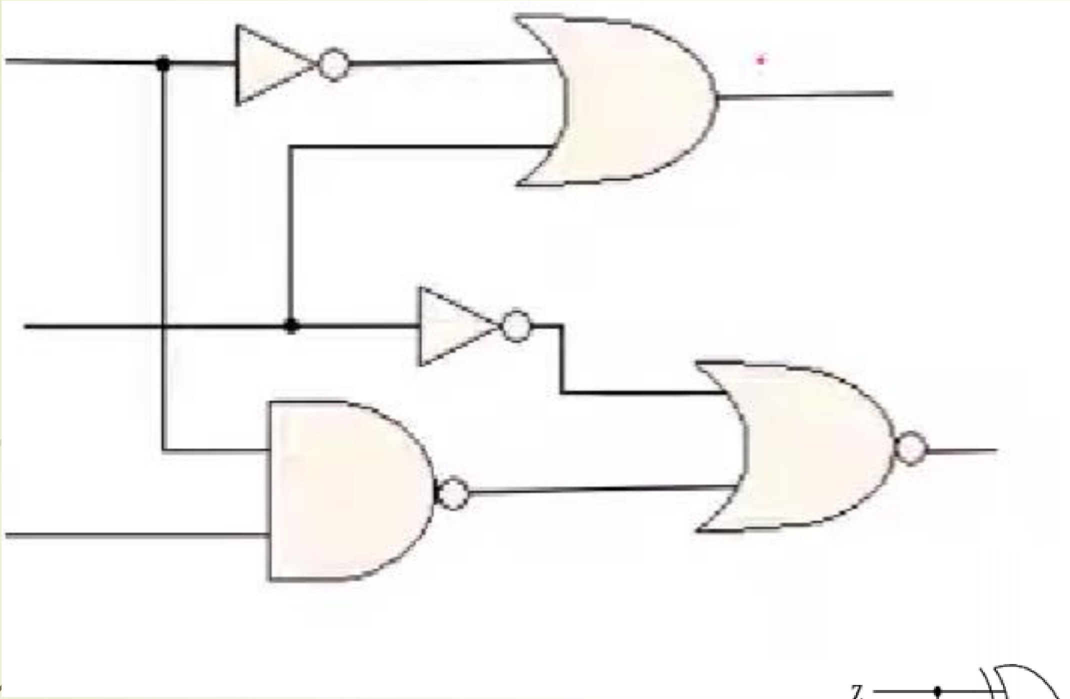
Dado el siguiente esquema, obtenga la función de salida (S) y simplifíquela.

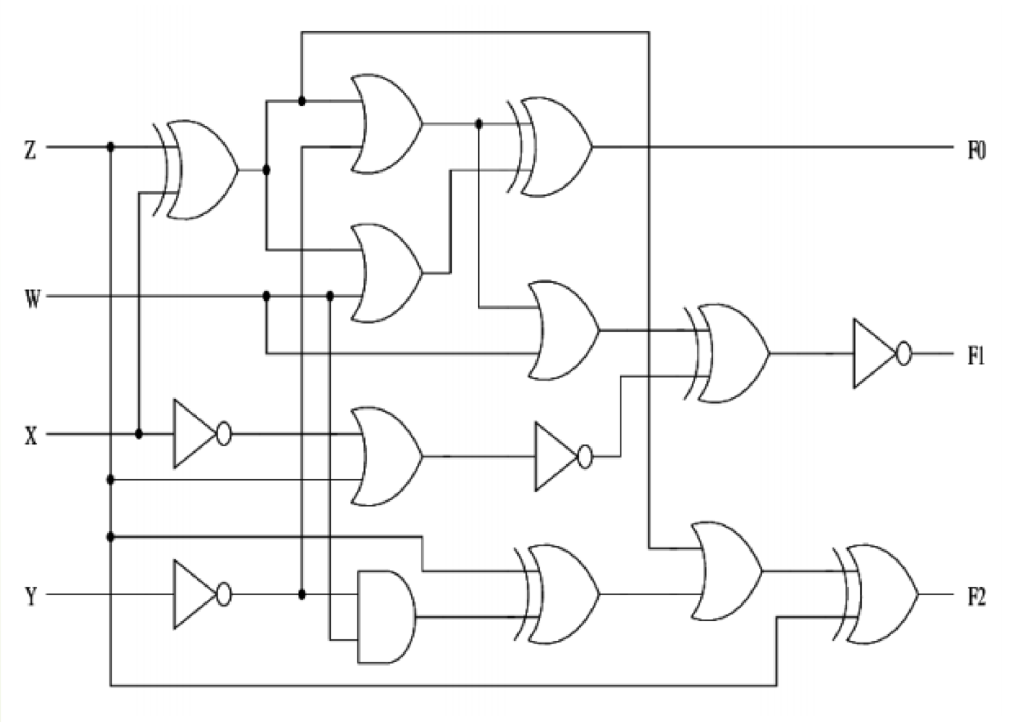


ACTIVIDAD PRÁCTICA

Determina la expresión de salida del circuito y simplifique mediante algebra de Bool y realice la tabla de verdad






Z
W
X
Y
F0
F1
F2

MAPAS DE KARNAUGH

Los mapas de Karnaugh son una herramienta utilizada para la simplificación de funciones lógicas booleanas y a diferencia de la resolución por algebra de Boole, este es un método gráfico que implica conocer las representaciones canónicas de las funciones.

El Mapa de Karnaugh tiene la característica de que puede ser visto como una representación bidimensional de una tabla de verdad. En la tabla de verdad, se colocan las variables por columnas y las combinaciones de tales variables determinan un valor de salida, 0 o 1, sin embargo, en el mapa las variables se colocan como si de un plano cartesiano se tratara, respetando cada una de las combinaciones que de ellas se generan, y colocando en la intersección de las combinaciones de las variables, el valor de salida.

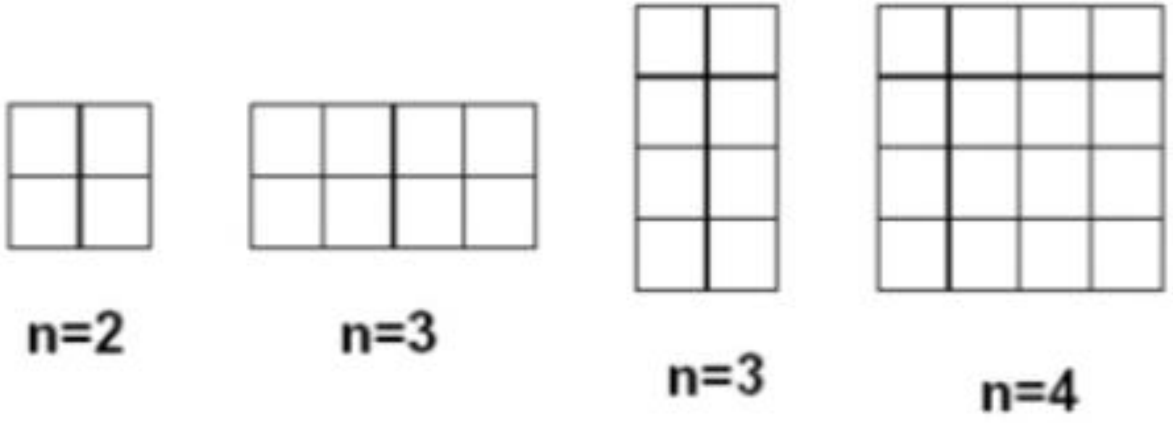
Mapas de Karnaugh

- **Los mapas de Karnaugh (MK) son formas modificadas de Tablas de Verdad que permiten minimizar funciones de hasta 5 variables.**
- **Los MK permiten diseño rápido de circuitos combinacionales de mínimo costo, es decir, con el mínimo número de compuertas.**

Construcción de Mapas de Karnaugh

- **Para construir MK se siguen los siguientes pasos:**

1) Para una función de n variables, el MK tiene 2^n celdas



...Construcción de Mapas de Karnaugh

2) En las coordenadas se anotan las combinaciones de variables según código Grey

	0	1
0		
1		

n=2

	00	01	11	10
0				
1				

n=3

	00	01	11	10
00				
01				
11				
10				

n=4

Se trata que de una celda a la siguiente sólo cambie una variable

MAPAS DE KARNAUGH

$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC$

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Simplifique haciendo uso de los Mapas de Karnaugh y cree el circuito.

A	B	C	D	F
0	0	0	0	**1**
0	0	0	1	**0**
0	0	1	0	**1**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	**1**
1	0	0	0	**0**
1	0	0	1	**1**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**0**
1	1	1	1	**1**