Preparador Informática

www.preparadorinformatica.com

MANUAL 1

CIRCUITOS LÓGICOS DIGITALES

1. Lógica de Circuitos

El término lógico se aplica a los circuitos digitales que se utilizan para implementar funciones lógicas. Existen varios tipos de circuitos lógicos, que son los elementos básicos que constituyen los bloques sobre los que se construyen los sistemas digitales más complejos, como por ejemplo un ordenador. A continuación, se abordan los conceptos previos necesarios para la descripción y diseño de estos circuitos:

- Operaciones lógicas
- Álgebra de Boole
- Teoremas de DeMorgan
- Funciones lógicas

2. Operaciones lógicas

Las puertas lógicas son el bloque fundamental de construcción de todos los circuitos lógicos digitales, de forma que las funciones lógicas se implementan interconectando puertas.

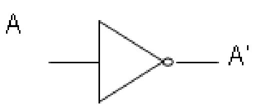
Para representar las puertas lógicas se utilizan las siguientes notaciones:

- Símbolos distintivos: corresponden a los utilizados comúnmente por la industria digital.
- Símbolos rectangulares: en 1984 se desarrolló una nueva norma para la representación de símbolos lógicos llamada IEEE/ANSI 91-1984. Esta norma es empleada entre otros organismos por el Dpto. de Defensa de los E.E.U.U.

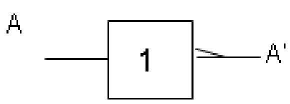
A continuación, se representan las principales puertas lógicas empleando la notación de símbolos distintivos (a) y la de símbolos rectangulares (b). Además, se muestra la tabla de verdad asociada a cada puerta lógica (c):

2.1. Puerta NOT

Produce un nivel lógico de salida opuesto al suministrado como entrada.



a)

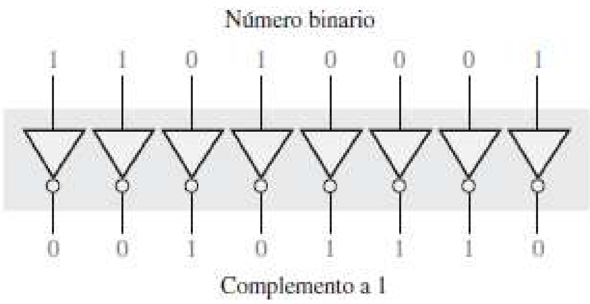


b)

a	s
0	1
1	0

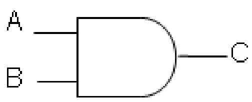
c)

Aplicación: En la imagen siguiente se muestra un circuito que genera el complemento a 1 de un número binario de 8 bits. Los bits del número binario se aplican a las entradas del inversor y el complemento a 1 se obtiene en las salidas.

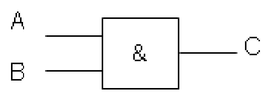


2.2. Puerta AND

Produce un nivel alto a la salida sólo cuando todas las entradas están a un nivel alto.



a)



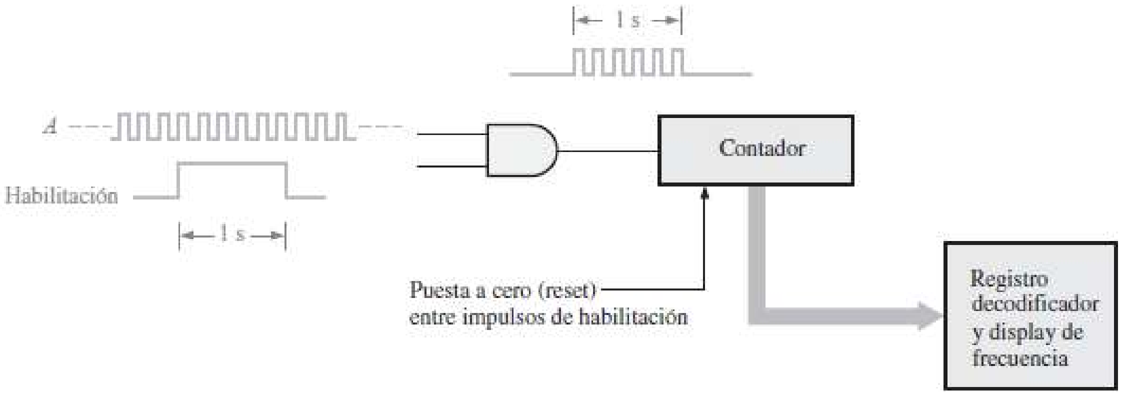
b)

a	b	s
0	0	0
0	1	0
1	0	0
1	1	1

c)

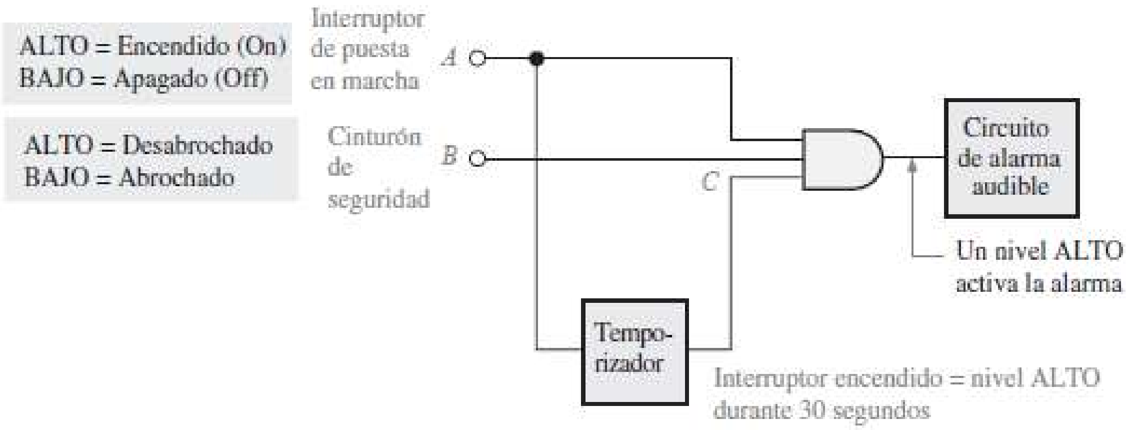
Aplicación: La puerta AND como un dispositivo de habilitación/inhibición. Una aplicación común de la puerta AND es habilitar (enable) o permitir el paso de una señal (tren de impulsos) de un punto a otro en determinados instantes, e inhibir o impedir el paso en otros instantes.

Un ejemplo sencillo de este particular uso de la puerta AND se muestra en la imagen siguiente:



La puerta AND controla el paso de una señal (A) a un contador digital. El propósito de este circuito es medir la frecuencia de la señal A. El impulso de habilitación tiene un ancho de exactamente 1 s. Cuando la entrada de habilitación está a nivel ALTO, la señal A pasa a través de la puerta hasta el contador, y cuando la entrada de habilitación está a nivel BAJO, se impide el paso de la señal a través de la puerta.

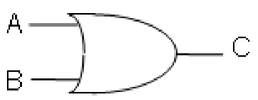
Otro ejemplo de aplicación lo podemos ver en la imagen siguiente:



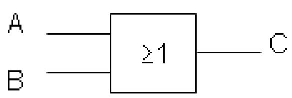
Es un sistema de alarma para el cinturón de seguridad. Se usa una puerta AND en un sencillo sistema de alarma para el cinturón de seguridad del coche, el cual detecta cuándo el interruptor de arranque se ha activado y el cinturón de seguridad no está abrochado. Si el interruptor de arranque se ha activado, la entrada A de la puerta AND se pone a nivel ALTO. Si el cinturón de seguridad no está correctamente abrochado, la entrada B de la puerta AND se pone a nivel ALTO. También cuando el interruptor de arranque se activa, se inicializa un temporizador que pone a nivel ALTO la entrada C durante 30 s. Si estas tres condiciones se cumplen, es decir, si el interruptor de arranque está activado y el cinturón de seguridad está desabrochado y el temporizador está corriendo, la salida de la puerta AND se pone a nivel ALTO, y una alarma audible se activa para advertir al conductor.

2.3. Puerta OR

Produce un nivel alto a la salida cuando cualquiera de sus entradas está a un nivel alto.



a)

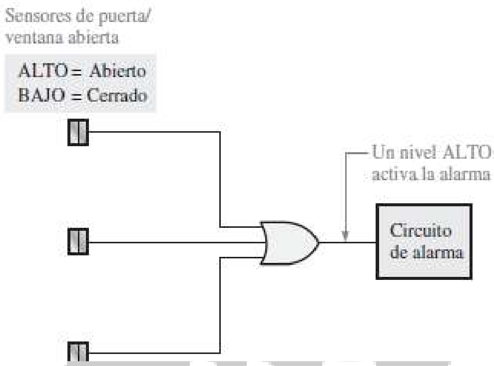


b)

a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

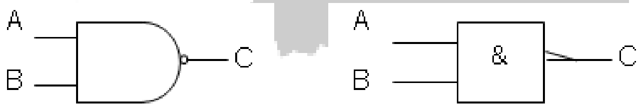
c)

Aplicación: En la siguiente figura se presenta parte de un sistema de alarma y detección de intrusos simplificado. Este sistema se podría utilizar en una habitación de una casa: una habitación con dos ventanas y una puerta. Los sensores son interruptores magnéticos que producen un nivel de salida ALTO cuando se abre la puerta (o las ventanas) y una salida a nivel BAJO cuando se cierra. Mientras que las ventanas y la puerta están aseguradas, los interruptores están cerrados y las tres entradas de la puerta OR están a nivel BAJO. Cuando se abre una de las ventanas o la puerta, en la entrada correspondiente de la puerta OR se genera un nivel ALTO y la salida de la puerta lógica se pone a nivel ALTO. Entonces se activa el circuito de alarma para advertir de la intrusión.



2.4. Puerta NAND

Realiza la función inversa a la Puerta AND.



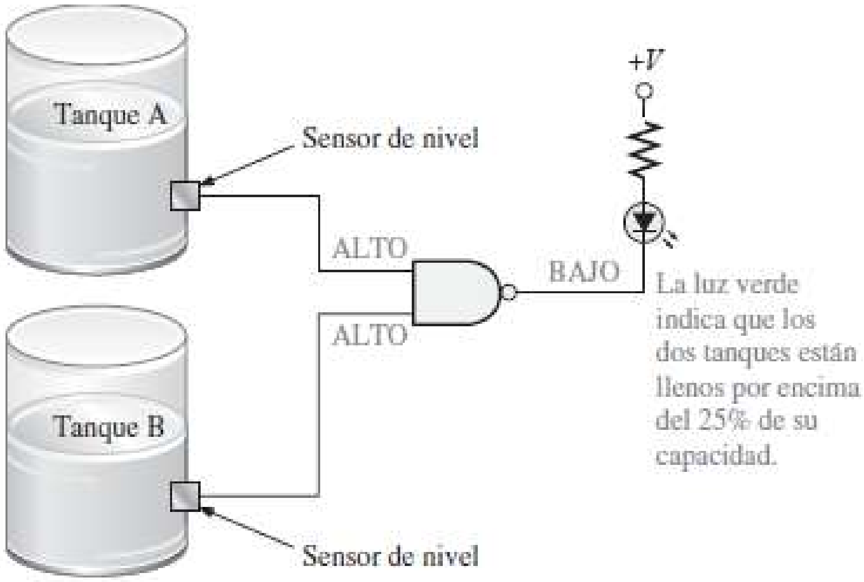
a	b	s
0	0	1
0	1	1
1	0	1
1	1	0

a) b) c)

Aplicación: Una planta de fabricación utiliza dos tanques para almacenar un determinado líquido químico que se requiere en un proceso de fabricación. Cada tanque dispone de un sensor que detecta cuándo el nivel del líquido cae al 25% del total. Los sensores generan una tensión de 5 V cuando los tanques están llenos por encima del 25%. Cuando el volumen de líquido en el tanque cae por debajo del 25%, el sensor genera un nivel de 0 V.

En el panel indicador se requiere un diodo emisor de luz (LED) verde que indique que el nivel de ambos tanques está por encima del 25%. Como se indica, se puede utilizar una puerta NAND para implementar esta función.

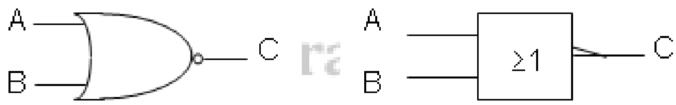
En la figura se muestra una puerta NAND de dos entradas conectadas a los sensores de nivel del tanque y la salida conectada al panel indicador. La operación puede definirse como sigue: si el nivel del tanque A y el nivel del tanque B están por encima de un cuarto del total, el LED se enciende.



Mientras que la salida de ambos sensores esté a nivel ALTO (5 V), quiere decir que el nivel de los tanques está por encima del 25% del volumen total, y la salida de la puerta NAND está a nivel BAJO (0 V). El circuito del LED verde se activa para una tensión a nivel BAJO.

2.5. Puerta NOR

Realiza la función inversa a la puerta OR.

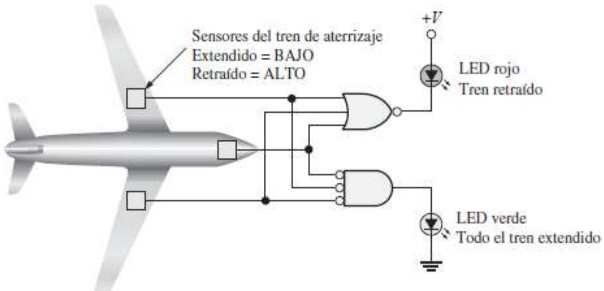


a	b	s
0	0	1
0	1	0
1	0	0
1	1	0

a) b) c)

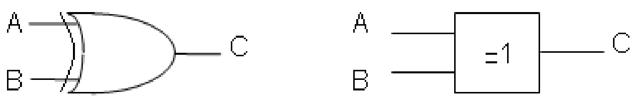
Aplicación: Como parte del sistema de monitorización funcional de un avión, se requiere un circuito para indicar el estado del tren de aterrizaje antes de tomar tierra. Se enciende un LED verde si los tres mecanismos de aterrizaje están correctamente extendidos cuando el interruptor para "bajar el tren de aterrizaje" se ha activado. Un LED rojo se enciende si cualquiera de los mecanismos falla al extenderse antes de aterrizar. Cuando uno de los mecanismos se extiende, el sensor correspondiente genera una tensión a nivel BAJO. Cuando uno de los mecanismos del tren de aterrizaje se retrae, su sensor genera una tensión a nivel ALTO.

Para implementar un circuito que cumpla esto la alimentación se aplica al circuito sólo cuando el interruptor para "bajar el tren de aterrizaje" se activa. Se debe utilizar una puerta NOR para cada uno de los dos requisitos, tal y como se muestra en la siguiente figura. Cuando las tres entradas de la puerta están a nivel BAJO, los tres mecanismos (ruedas) están correctamente extendidos y dan lugar a una salida a nivel ALTO en la puerta negativa–AND, lo que hace que se encienda el LED verde. La otra puerta NOR funciona como tal para detectar si una o más de las ruedas de aterrizaje permanecen retraídas cuando se activa el interruptor que baja el tren de aterrizaje. Cuando una o más de las ruedas de aterrizaje permanecen retraídas; la salida a nivel ALTO procedente del sensor se detecta mediante la puerta NOR, que da lugar a una salida a nivel BAJO que enciende el LED rojo de aviso.



2.6. Puerta XOR (OR-exclusiva)

Produce un nivel alto a la salida si y solo si una de las dos entradas está a un nivel alto.



a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

a) b) c)

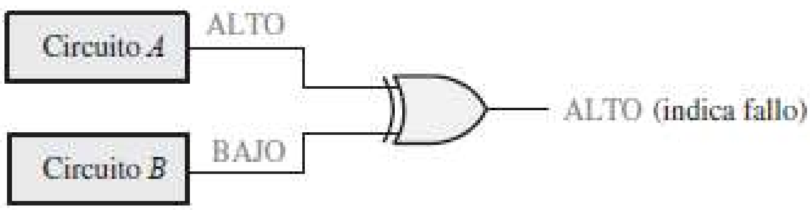
$$F = A \oplus B$$

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

Aplicación: Un cierto sistema está formado por dos circuitos idénticos que funcionan en paralelo. Mientras que ambos funcionan correctamente, las salidas de los dos circuitos son iguales. Si uno de los circuitos falla, las salidas serán niveles opuestos en ese instante. Establecer un método para detectar que se ha producido un fallo en uno de los circuitos.

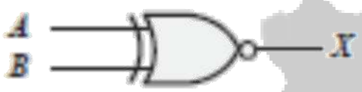
Para ello las salidas de los circuitos se conectan a las entradas de una puerta XOR, tal y como se muestra en la figura. Un fallo en cualquiera de los circuitos hace que las entradas de la puerta

XOR tengan niveles opuestos. Esta condición da lugar a nivel ALTO en la salida de la puerta XOR, que indica que uno de los circuitos ha fallado.

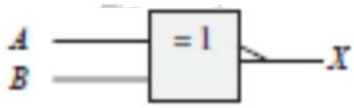


2.7. Puerta XNOR (NOR-exclusiva)

Produce un nivel alto a la salida si y solo las dos entradas están al mismo nivel.



a)



b)

a	b	s
0	0	1
0	1	0
1	0	0
1	1	1

c)

$$(A \cdot B)+(\bar{A} \cdot \bar{B})$$

Las puertas NAND y NOR se denominan puertas universales, ya que a partir de ellas es posible implementar cualquiera de las demás puertas lógicas.

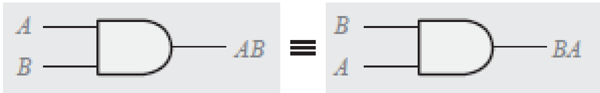
3. Álgebra de Boole

La base matemática en la que se fundamenta la electrónica digital es el álgebra de Boole. Para realizar de forma sistemática tanto el análisis como síntesis de los sistemas digitales se utiliza como eficaz herramienta el álgebra de Boole, que se puede definir como el conjunto de postulados siguientes:

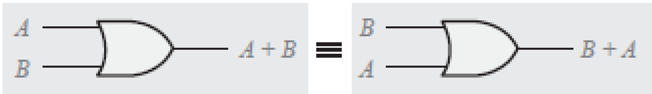
3.1. Leyes

- **Leyes conmutativas**

AB=BA

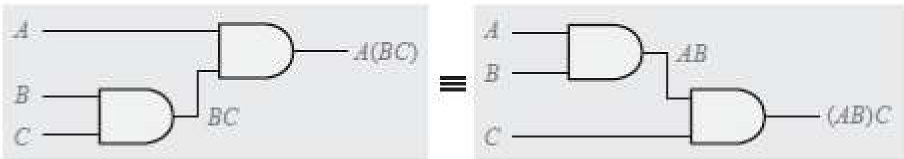


A+B=B+A

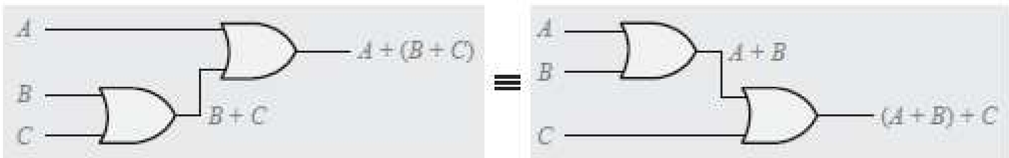


- **Leyes asociativas**

A(BC)=(AB)C

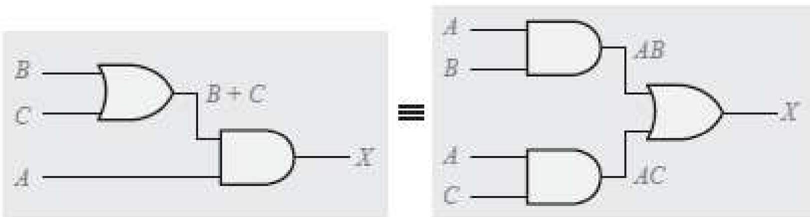


A+(B+C) =(A+B) +C



- **Ley distributiva**

A(B+C) =AB+AC



$X = A(B + C)$ $X = AB + AC$

3.2. Reglas

- **Primera Nomenclatura:**

1. $A + 0 = A$	7. $A \cdot A = A$
2. $A + 1 = 1$	8. $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	9. $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	10. $A + AB = A$
5. $A + A = A$	11. $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	12. $(A + B)(A + C) = A + BC$

A, B o C pueden representar una sola variable o una combinación de variables.

- **Segunda Nomenclatura:**

1. A+0=A	5. A+A=A	9. A=A´´
2. A+1=1	6. A+A´=1	10. A+AB=A
3. A*0=0	7. A*A=A	11. A+A´B= A+B
4. A*1=A	8. A*A´=0	12. (A+B)(A+C)= A +BC

Vamos a demostrar las reglas del Álgebra de Boole. Omitiremos las explicaciones de las primeras reglas ya que son básicas e inmediatas.

- **Regla 10: (A + AB = A)** Esta regla se puede obtener aplicando la ley distributiva y las reglas 2 y 4 de la siguiente forma:

$$
\begin{aligned}
A + AB &= A(1+B) && \text{Sacar factor común (ley distributiva)} \\
&= A \cdot 1 && \text{Regla 2: } (1+B) = 1 \\
&= A && \text{Regla 4: } A \cdot 1 = A
\end{aligned}
$$

La demostración se puede ver en la tabla siguiente:

A	B	AB	$A + AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

- **Regla 11:** Esta regla puede demostrarse de la siguiente forma:

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

$A+\overline{A}B=(A+AB)+\overline{A}B$	Regla 10: $A=A+AB$
$=(AA+AB)+\overline{A}B$	Regla 7: $A=AA$
$=AA+AB+A\overline{A}+\overline{A}B$	Regla 8: sumar $A\overline{A}=0$
$=(A+\overline{A})(A+B)$	Sacar factor común
$=1\cdot(A+B)$	Regla 6: $A+\overline{A}=1$
$=A+B$	Regla 4: eliminar el 1

La demostración se puede ver en la tabla siguiente:

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

igual

- **Regla 12: [(A+B)(A+C)=A+BC]** Esta regla puede demostrarse de la siguiente forma:

$(A+B)(A+C)$	$=AA+AC+AB+BC$	Ley distributiva
	$=A+AC+AB+BC$	Regla 7: $AA=A$
	$=A(1+C)+AB+BC$	Sacar factor común (ley distributiva)

$=A\cdot 1+AB+BC$	Regla 2: $1+C=1$
$=A(1+B)+BC$	Sacar factor común (ley distributiva)
$=A\cdot 1+BC$	Regla 2: $1+B=1$
$=A+BC$	Regla 4: $A\cdot 1=A$

La demostración se puede ver en la tabla siguiente:

A	B	C	$A+B$	$A+C$	$(A+B)(A+C)$	BC	$A+BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

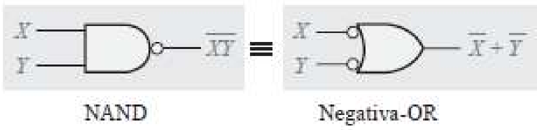
4. Teoremas de DeMorgan

El matemático DeMorgan propuso dos teoremas basados en las teorías de Boole que constituyen una parte muy importante del álgebra de Boole.

1. **El complemento de un producto de variables es igual a la suma de los complementos de las variables.**

(A*B)´= A´+B´

$$\overline{XY} = \overline{X} + \overline{Y}$$

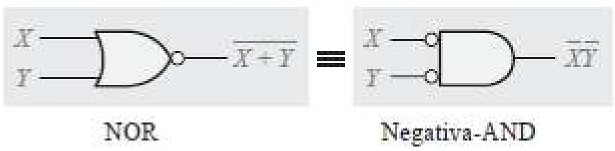


Entradas		Salida	
X	Y	$\overline{XY}$	$\overline{X}+\overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

2. **El complemento de una suma de variables es igual al producto de los complementos de las variables**

(A+B)´=A´*B´

$$\overline{X+Y} = \overline{X}\,\overline{Y}$$



Entradas		Salida	
X	Y	$\overline{X+Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

EJEMPLO:

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

Paso1: Identificamos los términos a los que se pueden aplicar los teoremas de DeMorgan y consideramos cada término como una única variable, por lo que establecemos

$\overline{A+B\overline{C}} = X$ y $D(\overline{E+\overline{F}}) = Y$.

Paso 2: Dado que

$$\overline{X+Y}=\overline{X}\,\overline{Y}.$$

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

Paso 3: Utilizamos la regla 9 para eliminar la barra doble sobre el término de la izquierda

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

Paso 4: Aplicando el teorema de DeMorgan al segundo término

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

Paso 5: Empleamos la regla 9 para cancelar las barras dobles sobre la parte del segundo término

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

5. Funciones Lógicas

Las funciones lógicas describen cada una de las salidas de un sistema digital para todas las posibles combinaciones de entradas.

Las funciones lógicas se pueden expresar de varias formas:

A. Expresiones booleanas, de la forma:

$$x=A'B'C'+A'B'C+A'BC+ABC'$$

B. Circuitos lógicos: (representado mediante puertas lógicas).

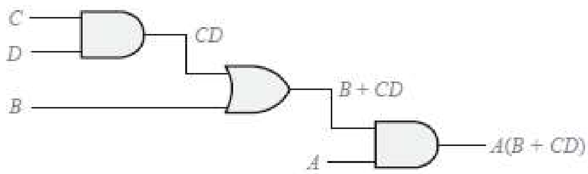
C. Tablas de verdad: es una tabla de las posibles combinaciones de valores de las variables de entrada y sus correspondientes valores de salida.

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

D. Mapas de Karnaugh: es una matriz de celdas que proporciona un método sistemático de simplificación de expresiones booleanas.

C \ A B	00	01	11	10
0	1	1	1	0
1	1	0	0	0

6. Expresión booleana de un circuito lógico



Para obtener la expresión booleana de un determinado circuito lógico, la manera de proceder consiste en comenzar con las entradas situadas más a la izquierda e ir avanzando hasta las líneas de salida, escribiendo la expresión para cada puerta.

Para el circuito ejemplo de la imagen, su expresión booleana se determina de la siguiente manera:

1. La expresión de la puerta AND situada más a la izquierda cuyas entradas son C y D es CD.

2. La salida de la puerta AND situada más a la izquierda es una de las entradas de la puerta OR y B es su otra entrada. Por tanto, la expresión para la puerta OR es B+CD.

3. La salida de la puerta OR es una de las entradas de la puerta AND situada más a la derecha, siendo A su otra entrada. Por tanto, la expresión de esta puerta AND será A(B+CD), que es la expresión final de salida del circuito completo.

7. Construcción de una tabla de verdad para un circuito lógico

Una vez que se ha determinado la expresión booleana de un circuito dado, puede desarrollarse una tabla de verdad que represente la salida del circuito lógico para todos los valores posibles de las variables de entrada. El procedimiento requiere que se evalúe la expresión booleana para todas las posibles combinaciones de valores de las variables de entrada. En el caso del circuito de la imagen anterior, existen cuatro variables de entrada (A, B, C y D) y, por tanto, hay dieciséis ($2^4 = 16$) posibles combinaciones de valores.

Para evaluar la expresión A(B + CD), en primer lugar hallamos los valores de las variables que hacen que la expresión sea igual a 1, utilizando las reglas de la suma y la multiplicación booleanas.

En este caso, la expresión es igual a 1 sólo si A = 1 y B +CD = 1.

Ahora hay que determinar cuándo el término B + CD es igual a 1. El término B + CD = 1 si B = 1 o CD = 1 o si ambas variables son igual a 1, ya que:

B + CD = 1 + 0 = 1

B + CD = 0 + 1 = 1

B + CD = 1 + 1 = 1

El término CD = 1 sólo si C = 1 y D = 1.

Resumiendo, la expresión A (B + CD) = 1 cuando A = 1 y B = 1, independientemente de los valores de C y D, o cuando A = 1 y C = 1 y D = 1, independientemente del valor de B. La expresión A (B + CD) = 0 para todas las restantes combinaciones de valores de las variables.

Para hacer la representación de los resultados en una tabla de verdad, el primer paso consiste en enumerar las dieciséis combinaciones de unos y ceros de las variables de entrada en una secuencia binaria, como muestra la tabla siguiente. A continuación, se pone un 1 en la columna de salida para las combinaciones de variables de entrada que se han determinado en la evaluación de la expresión. Finalmente, se escribe un 0 en la columna de salida para el resto de las combinaciones de las variables de entrada. Los resultados serán:

Entradas				**Salida**
A	**B**	**C**	**D**	**A(B + CD)**
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0

0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

8. Simplificación mediante el Álgebra de Boole

Muchas veces, a la hora de aplicar el álgebra booleana, hay que reducir una expresión a su forma más simple o cambiarla a una forma más conveniente para conseguir una implementación más eficiente. Veamos este método que utiliza las reglas, leyes y teoremas del álgebra de Boole para manipular y simplificar una expresión. Este método requiere un profundo conocimiento del álgebra booleana y una considerable experiencia en su aplicación, por no mencionar también un poquito de ingenio y destreza.

Una expresión booleana simplificada emplea el menor número posible de puertas en la implementación de una determinada expresión. Veamos un ejemplo:

EJEMPLO: Simplificar la siguiente expresión utilizando técnicas del álgebra de Boole:

AB + A(B + C) + B(B + C)

El método que se sigue no es necesariamente el único método posible.

- **Paso 1.** Aplicar la ley distributiva al segundo y tercer término del siguiente modo:

$$AB + AB + AC + BB + BC$$

- **Paso 2.** Aplicar la regla 7 (BB = B) al cuarto término.

$$AB + AB + AC + B + BC$$

- **Paso 3.** Aplicar la regla 5 (AB + AB = AB) a los dos primeros términos.

$$AB + AC + B + BC$$

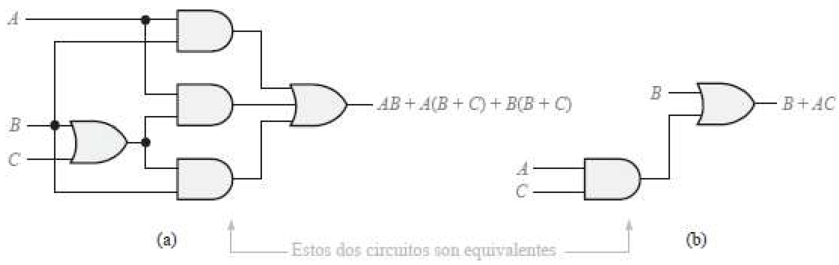
- **Paso 4.** Aplicar la regla 10 (B + BC = B) a los dos últimos términos.

AB + AC + B

- **Paso 5.** Aplicar la regla 10 (AB + B = B) al primero y tercer término.

B + AC

En este punto, la expresión ya no puede seguir simplificándose.



Vemos en la anterior imagen cómo el proceso de simplificación del ejemplo anterior ha reducido significativamente el número de puertas lógicas necesarias para implementar la expresión. En la parte (a) se puede ver que son necesarias cinco puertas para implementar dicha expresión en su forma original, mientras que sólo se requieren dos para hacerlo una vez simplificada, como se muestra en la parte (b). Es importante resaltar que estos dos circuitos de puertas son equivalentes, es decir, para cualquier combinación de valores en las entradas A, B y C, obtenemos siempre la misma salida en ambos circuitos.

9. Formas estándar de las expresiones booleanas

Todas las expresiones booleanas, independientemente de su forma, pueden convertirse en cualquiera de las dos formas estándar: suma de productos o producto de sumas. La estandarización posibilita que la evaluación, simplificación e implementación de las expresiones booleanas sea mucho más sistemática y sencilla.

9.1. Suma de productos

Cuando dos o más productos se suman mediante la adición booleana, la expresión resultante se denomina suma de productos. Algunos ejemplos son:

$$AB + ABC$$
$$ABC + CDE + \overline{B}C\overline{D}$$
$$\overline{A}B + \overline{A}B\overline{C} + AC$$

Una suma de productos puede contener también términos de una única variable como en $A+\overline{A}\overline{B}C+BC\overline{D}$.

En una expresión con formato de suma de productos, una barra no puede extenderse sobre más de una variable; sin embargo, más de una variable puede tener una barra encima. Por ejemplo, una suma de productos puede contener el término $\overline{A}\overline{B}\overline{C}$ pero no el término $\overline{ABC}$.

La implementación de una suma de productos simplemente requiere aplicar la operación OR a las salidas de dos o más puertas AND. Una operación AND da lugar a un producto, y la adición de dos o más productos se realiza mediante puertas OR. Por tanto, una expresión suma de productos puede implementarse mediante un circuito lógico AND-OR en el que las salidas de las puertas AND, cuyo número es igual al de productos que contenga la expresión, son las entradas de una puerta OR.

9.2. Producto de sumas

Cuando dos o más términos suma se multiplican, la expresión resultante es un producto de sumas. Algunos ejemplos son:

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

Un producto de sumas puede contener términos con una única variable como en

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

En una expresión producto de sumas, una barra no puede extenderse nunca sobre más de una variable, aunque más de una variable puede tener una barra encima. Por ejemplo, un producto de sumas puede contener el término $\overline{A}+\overline{B}+\overline{C}$ pero no el $\overline{A+B+C}$.

La implementación de un producto de sumas requiere simplemente la aplicación de la operación AND a las salidas de dos o más puertas OR. Un sumando se origina mediante la operación OR y el producto de varios términos suma se realiza por medio de la operación AND. Por tanto, un producto de sumas puede implementarse a partir de puertas lógicas OR (cuyo número será igual al de sumandos de la expresión) cuyas salidas se conectan a las entradas de una puerta AND.

10. Conversión de una suma de productos estándar en un producto de sumas estándar

Los valores binarios de los términos producto en una suma de productos estándar dada no aparecen en su producto de sumas estándar equivalente. Asimismo, los valores binarios que no están representados en una suma de productos sí aparecen en el producto de sumas equivalente. Por tanto, para pasar de la suma de productos estándar al producto de sumas estándar hay que realizar los siguientes pasos:

- **Paso 1.** Evaluar cada término producto de la expresión suma de productos. Es decir, determinar los números binarios que representan estos términos.
- **Paso 2.** Determinar todos los números binarios no incluidos al realizar la evaluación del paso 1.
- **Paso 3.** Escribir los términos suma equivalente para cada valor binario del paso 2 y expresarlos en forma producto de sumas.

Utilizando un procedimiento similar, se puede pasar de un producto de sumas a una suma de productos.

EJEMPLO: Convertir la siguiente suma de productos en como producto de sumas:

$$\overline{A}\overline{B}\overline{C}+\overline{A}B\overline{C}+\overline{A}BC+A\overline{B}C+ABC$$

El resultado de la evaluación es la siguiente 000 + 010 + 011 + 101 + 111.

Puesto que son tres las variables que conforman el dominio de esta expresión, existe un total de ocho (2^3) posibles combinaciones. La suma de productos contiene cinco de estas combinaciones, luego la expresión producto de sumas debe contener las otras tres que son 001, 100 y 110. Recuerde que estos son los valores binarios que hacen que cada término suma sea igual a cero. La expresión producto de sumas equivalente es la siguiente:

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

11. Expresiones booleanas y tablas de verdad

Todas las expresiones booleanas pueden convertirse fácilmente en tablas de verdad utilizando los valores binarios de cada término de la expresión. La tabla de verdad es una forma muy común, en un formato muy conciso, de expresar el funcionamiento lógico de un circuito.

Además, las expresiones suma de productos y producto de sumas pueden determinarse mediante las tablas de verdad.

11.1. Conversión de una suma de productos a tabla de verdad

Una suma de productos es igual a 1 sólo si y sólo si al menos uno de los productos es igual a 1. Una tabla de verdad es sencillamente la lista de las posibles combinaciones de valores de las variables de entrada y sus correspondientes valores de salida (1 o 0).

El primer paso para construir una tabla de verdad consiste en enumerar todas las posibles combinaciones de los valores de las variables de la expresión. A continuación, hay que pasar la suma de productos a su formato estándar, si no lo está ya. Por último, se escribe un 1 en la columna de salida para cada valor binario que hace que la suma de productos estándar sea 1, y se escribe un 0 para los restantes valores.

EJEMPLO: Desarrollar una tabla de verdad para la expresión suma de productos estándar

$$\overline{A}\overline{B}C + A\overline{B}\overline{C} + ABC.$$

Existen tres variables en el dominio, por lo que hay ocho posibles combinaciones de valores binarios de las variables

Entrada			Salida
A	**B**	**C**	**X**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

11.2. Conversión de un producto de sumas a tabla de verdad

Recuerde que un producto de sumas es igual a 0 sólo si y sólo si al menos uno de los términos suma es igual a 0. Para construir la tabla de verdad de un producto de sumas, basta con enumerar todas las posibles combinaciones de valores binarios de las variables del mismo modo que se hace para una suma de productos. A continuación, hay que pasar el producto de sumas a su formato estándar, si no lo está ya. Por último, se escribe un 0 en la columna de salida para cada valor binario que hace que la suma de productos estándar sea 0, y se escribe un 1 para los restantes valores binarios.

EJEMPLO: Desarrollar una tabla de verdad para la expresión producto de sumas siguiente:

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

Entrada			Salida
A	**B**	**C**	**X**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

11.3. Determinación de las expresiones estándar a partir de una tabla de verdad

Para determinar la expresión de la suma de productos estándar representada por una tabla de verdad se enumeran todos los valores de las variables de entrada para los que la salida es 1. Cada valor binario se convierte en el correspondientes término producto, reemplazando cada 1 por la variable y cada 0 por la variable complementada. Por ejemplo, el valor binario 1010 se transforma en un término producto de la manera siguiente:

$$1010 \rightarrow A\overline{B}C\overline{D}$$

Para determinar el producto de sumas estándar representado por una tabla de verdad se enumeran todos los valores binarios para los que la salida es 0. A continuación, se convierte cada valor binario en el correspondiente término suma, reemplazando cada 1 por la variable complementada y cada 0 por la variable. Por ejemplo, el número binario 1001 se pasa a término suma de la manera siguiente:

$$1001 \rightarrow \overline{A}+B+C+\overline{D}$$

EJEMPLO: A partir de la siguiente tabla de verdad, determinar la expresión suma de productos estándar y la expresión producto de sumas estándar equivalente:

Entradas			Salida
A	*B*	*C*	*X*
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

En la columna de salida hay cuatro 1s y los correspondientes valores binarios son 011, 100, 110 y 111. Convertir estos valores binarios a términos producto como sigue:

$$011 \rightarrow \overline{A}BC$$
$$100 \rightarrow A\overline{B}\overline{C}$$
$$110 \rightarrow AB\overline{C}$$
$$111 \rightarrow ABC$$

La expresión suma de productos estándar resultante para la salida X es:

$$X = \overline{A}BC + A\overline{B}\overline{C} + AB\overline{C} + ABC$$

Para el producto de sumas, la salida es 0 para los valores binarios 000, 001, 010 y 101. Estos valores binarios se convierten en términos suma como sigue:

$$000 \rightarrow A+B+C$$
$$001 \rightarrow A+B+\overline{C}$$
$$010 \rightarrow A+\overline{B}+C$$
$$101 \rightarrow A+B+\overline{C}$$

La expresión producto de sumas estándar resultantes para la salida X es:

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