



2º de Bachillerato

## Tecnología Industrial II

### Contenidos

### Circuitos y sistemas lógicos: Álgebra de Boole. Puertas y funciones lógicas

El álgebra de Boole se denomina así en honor a **George Boole**, matemático inglés 1815 - 1864, que fue el primero en definirla como parte de un sistema lógico, a mediados del siglo XIX. El álgebra de Boole fue un intento de utilizar las técnicas algebraicas para tratar expresiones de la lógica proposicional.

El álgebra opera con **variables booleanas**, que son aquellas que sólo pueden tomar dos valores (0 y 1), estos valores no representan números si no estados. Ejemplo: pueden simbolizar si un interruptor está abierto (0), o cerrado (1), si conduce o no conduce, si hay tensión o no.



George Boole

Imagen de Hask en [Wikimedia](#). Dominio público

En la actualidad, esta herramienta se aplica de forma generalizada en el ámbito del diseño de control electrónico de procesos industriales. Estos métodos de control se basan en el uso de sistemas

denominados puertas lógicas.

# 1. Operaciones básicas en el álgebra de Boole



En primer lugar definiremos unos conceptos que necesitamos utilizar:

## Importante

### **Función lógica:**

Toda aquella variable binaria cuyo valor depende de una **expresión algebraica** constituida por otras variables que se encuentran relacionadas entre si por determinadas operaciones.

Por ejemplo sea la expresión:

$$S = A \cdot B + C$$

La interpretación de esta función será que la salida S, tomará el valor "uno" cuando lo hagan las variables A y B o la variable C.

## Importante

### **Función canónica:**

Expresión lógica en la que todos sus términos contienen todas las variables de entrada, bien afirmadas o bien negadas.

Por ejemplo 1 :

La siguiente función lógica **S**, está compuesta por tres términos, y cada término tiene en su composición las tres variables del sistema (A, B y C) . Está función **S**, está expresada en **forma canónica**.

$$S = A \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$$

Por ejemplo 2 :

La siguiente función lógica **S**, está compuesta por tres términos, pero no todos estos términos tienen en su composición las tres variables del sistema (A, B y C). Esta función **S**, está expresada en forma **no canónica**.

$$S = A \cdot B \cdot C + A \cdot \bar{B} + B \cdot \bar{C}$$

## Importante

### Tabla de verdad:

Tabla formada por **n** columnas de entradas o variables (**A, B, C....**) y otra más de salida (**f**), y por **2<sup>n</sup>** filas, de modo que cada una de las filas representa cada una de las combinaciones posibles que pueden tener las variables de entrada, y el valor que toma la variable de salida para cada combinación de las variables de entrada.

Los posibles valores que pueden tomar las distintas variables representan estados diferentes de un dispositivo, y están representados por los valores lógicos "**cero**" (apagado, abierto, desconectado, ausencia, nivel bajo,...) o "**uno**" (encendido, cerrado, conectado, presencia, nivel alto,...).

Cuando en una tabla de verdad una variable está representada por el valor "x" quiere decir que su valor no está definido o que es indiferente el valor que tome para la respuesta de la función lógica.

En la figura adjunta se representa la tabla de verdad correspondiente a la función lógica:

$$S = A \cdot \bar{B} + \bar{A} \cdot B$$

ENTRADAS		SALIDA
A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

## Importante

Las variables lógicas pueden tomar exclusivamente los valores 0 y 1. Estos valores se pueden conseguir eléctricamente por dos procedimientos diferentes, según como se asignen los niveles de tensión se habla de lógica positiva o negativa.

Si se asigna al 1 lógico el valor más positivo de tensión y el 0 lógico al valor de menor tensión, estaremos ante **lógica positiva**.

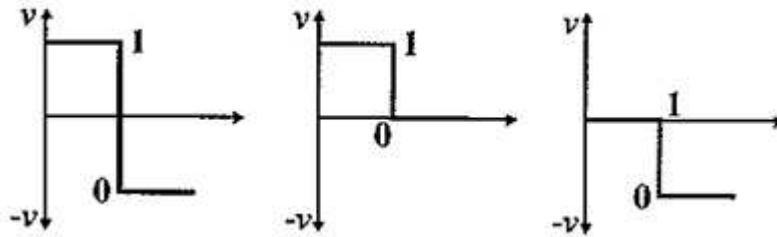


Imagen de elaboración propia

Si se asigna al 1 lógico el valor más negativo de tensión, estaremos ante **lógica negativa**.

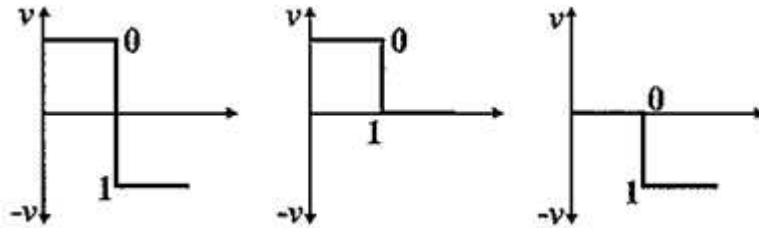


Imagen de elaboración propia

## 1.1. Operaciones básicas

En el algebra de Boole se definen tres operaciones básicas que son:

- Producto lógico, o intersección.
- Suma lógica o unión.
- Negación, complementación o inversión.

Las operaciones básicas del álgebra de Boole se suelen implementar por medio de circuitos electrónicos, a estos componentes se les llama **puertas lógicas**.

Veamos ahora cada una de ellas por separado:

### Importante

#### Producto lógico. Puerta AND. Y. &.

Función representada por la expresión:

$$S = A \cdot B$$

Esta función responde a la tabla de verdad que se acompaña, debe entenderse como: La salida correspondiente a la función lógica producto toma valor 1 solamente cuando también son 1 los valores de las entradas A y B.

dec	A	B	$S = A \cdot B$
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA (ANSI):

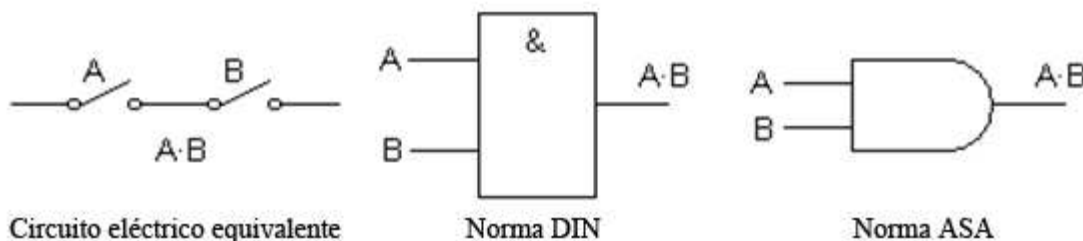


Imagen de elaboración propia

## Importante

### Suma lógica. Puerta OR. O.

Función representada por la expresión:

$$S = A + B$$

Esta función responde a la tabla de verdad que se acompaña, debe ser leída como: La salida correspondiente a la función lógica suma toma valor 1 cuando también son 1 cualquiera de las entradas A y B, o las dos simultáneamente.

dec	A	B	$S = A + B$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

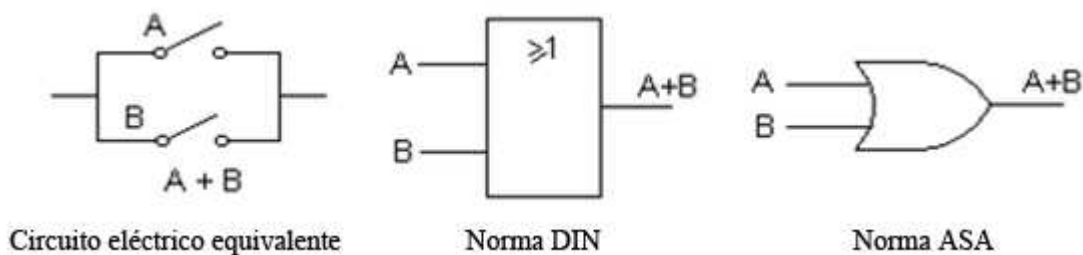


Imagen de elaboración propia

## Importante

### Negación, complementación o inversión. NOT.

Función representada por la expresión:

$$S = \bar{A}$$

entrada es 0 y viceversa.

$A$	$S = \bar{A}$
0	<b>1</b>
1	<b>0</b>

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según las normas DIN y ASA:

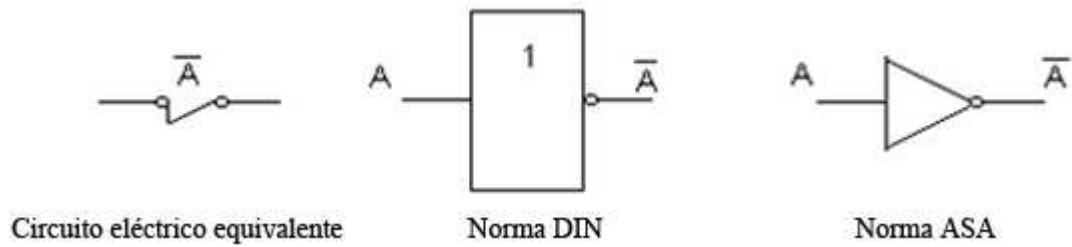


Imagen de elaboración propia



### Importante

#### Puerta NAND. No Y.

Función representada por la expresión:

$$S = \overline{A \cdot B}$$

Esta función responde a la tabla de verdad que se acompaña, debe interpretarse como: La salida correspondiente a la función lógica toma siempre valor 1 salvo cuando A y B tienen a la vez valor 1.

La salida es el producto negado de las variables de entrada. Es decir la función NAND es la asociación de una puerta AND y de la función NOT.

dec	A	B	$S = \overline{A \cdot B}$
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

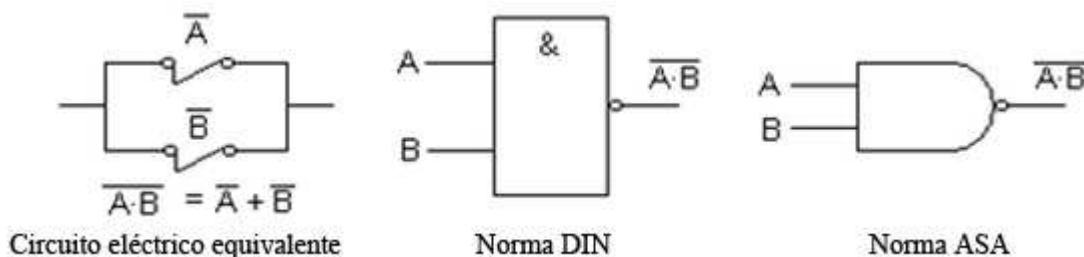


Imagen de elaboración propia

### Importante

### Puerta NOR. No O.

Función representada por la expresión:

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

Esta función responde a la tabla de verdad que se acompaña, debe interpretarse como: La salida correspondiente a la función lógica toma valor 1 solamente cuando A y B tienen a la vez valor 0.

La salida es la suma negada de las variables de entrada. Es decir la función NOR es la asociación de una puerta OR y de la función NOT.

dec	A	B	$S = \overline{A+B}$
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

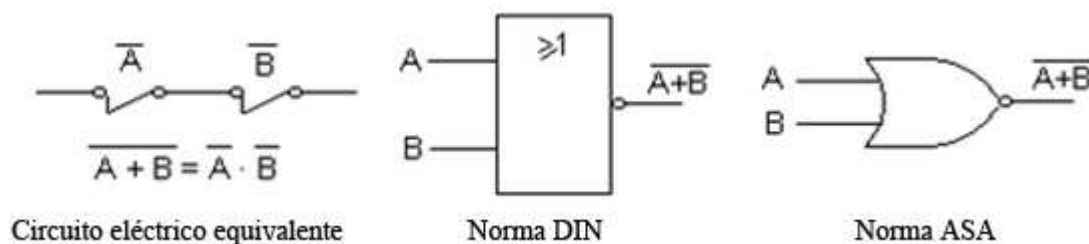


Imagen de elaboración propia

## Importante

### Puerta OR EXCLUSIVA. XOR.

Función representada por la expresión:

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

Esta función responde a la tabla de verdad que se acompaña, que debemos leer como: La salida correspondiente a la función lógica toma valor 1 cuando A ó B pero no las dos,

dec	A	B	$S = \bar{A} \cdot B + A \cdot \bar{B}$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

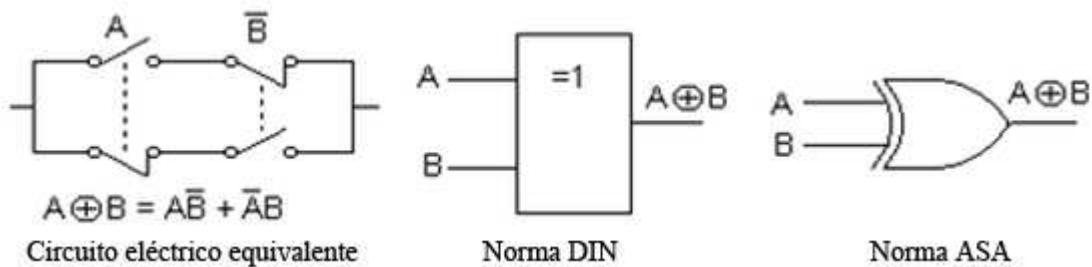


Imagen de elaboración propia

## Importante

### Puerta NOR EXCLUSIVA. XNOR.

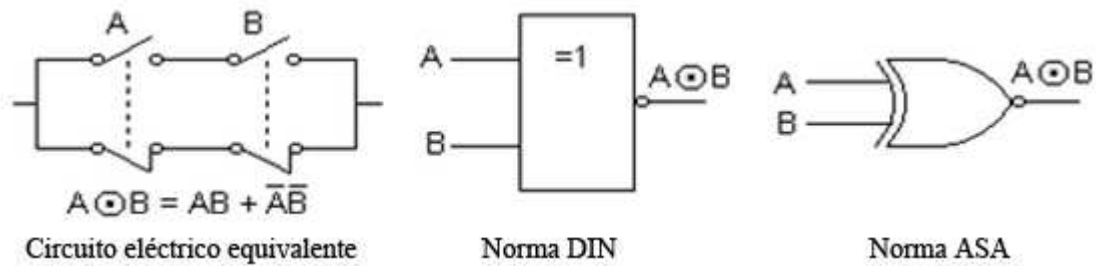
Función representada por la expresión:

$$S = A \cdot B + \bar{A} \cdot \bar{B}$$

Esta función responde a la tabla de verdad que se acompaña, que debemos leer como: La salida toma el valor 1 cuando las dos entradas son 1 o las dos entradas son 0.

dec	A	B	$S = A \cdot B + \bar{A} \cdot \bar{B}$
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

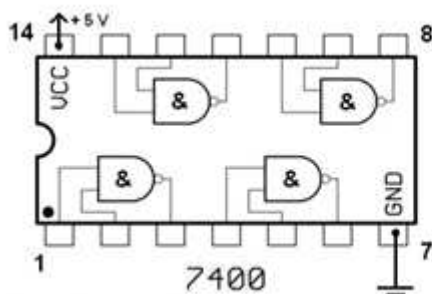
Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:



## Para saber más

- Comercialmente las puertas lógicas se distribuyen en circuitos integrados (CI), que disponen de catorce patillas o pines.
- Todos ellos presentan una muesca que se debe ver a la izquierda para empezar a contar las patillas desde abajo de la muesca y en sentido contrario a las agujas de un reloj.
- La patilla número 7 se conecta a masa y la número 14 a tensión de alimentación +Vcc, por lo que quedan 12 patillas para utilizar distintas entradas y salidas.
- Cada circuito integrado se identifica mediante un código, que indica el número y el tipo de puertas de que consta.

En la fotografía se muestra un circuito integrado que contiene cuatro puertas NAND de dos entradas, con las conexiones como se puede observar en el esquema adjunto.



### 3. Álgebra de Boole, postulados y teoremas



Vamos a enumerar las distintas operaciones, postulados, leyes y teoremas, para familiarizarnos con el manejo del álgebra de Boole.

#### Postulados:

Respecto a la suma	Respecto al producto
$A + 0 = A$	$A \cdot 0 = 0$
$A + 1 = 1$	$A \cdot 1 = A$
$A + A = A$	$A \cdot A = A$
$A + \bar{A} = 1$	$A \cdot \bar{A} = 0$
$\bar{\bar{A}} = A$	

Imagen de elaboración propia

#### Leyes y teoremas:

Leyes/Teoremas	Respecto a la suma	Respecto al producto
Commutativa.	$A + B = B + A$	$A \cdot B = B \cdot A$
Asociativa	$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Distributiva	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
Cancelación	$(A \cdot B) + A = A$	$(A + B) \cdot A = A$
De Morgan	$\overline{A + B} = \bar{A} \cdot \bar{B}$	$\overline{A \cdot B} = \bar{A} + \bar{B}$

Imagen de elaboración propia

#### Reflexiona

Comprueba que se cumplen las leyes de DeMorgan, mediante el método de las tablas de verdad, e implementa con los distintos tipos de puertas lógicas los resultados que has obtenido.

#### Mostrar retroalimentación

Es este un caso particular del **principio de dualidad**.

A toda relación o ley lógica le corresponderá su dual, que se forma permutando las operaciones suma lógica por producto lógico, los 0 por 1 y los 1 por 0, y las variables afirmadas por negadas y viceversa.



## 4. Obtención de la función lógica a partir de la tabla de verdad



Una vez que hemos obtenido la tabla de verdad de un sistema o problema, existen dos métodos distintos para obtener la función lógica que la representa. En ambos métodos obtenemos una forma canónica. Como recordarás una forma canónica es la expresión de la función lógica sin ninguna simplificación, es decir, en cada término estarán presentes todas las variables de entrada, negadas o sin negar.

Los dos métodos son:

### *Importante*

#### **Productos lógicos "Minterms"** (implementación por unos)

El proceso consiste en tomar todas las combinaciones de las variables de entrada que provocan que la función lógica presente un 1 en la salida.

Cada una de estas combinaciones de las variables de entrada será un sumando, constituido por el producto de todas las variables de entrada, en el que las variables estarán negadas cuando tomen el valor 0 y estarán afirmadas cuando tomen el valor 1.

La expresión de la función canónica será la suma de todos los productos equivalentes a las combinaciones de las variables de entrada que hacen que la salida sea un 1.

#### **Sumas lógicas "Maxterms"** (implementación por ceros)

El proceso consiste en tomar todas las combinaciones de las variables de entrada que provocan que la función lógica presente un 0 en la salida.

Cada una de estas combinaciones de las variables de entrada será un producto, constituido por la suma de todas las variables de entrada, en el que las variables estarán negadas cuando tomen el valor 1 y estarán afirmadas cuando tomen el valor 0. La expresión de la función canónica será el producto de todas las sumas equivalentes a las combinaciones de las variables de entrada que hacen que la salida sea un 0.

Cualquiera de los dos métodos es igual de eficaz y rápido, aunque suele tenerse tendencia a resolver los problemas utilizando el método de minterms, es decir suma de productos, o el método de los unos, aunque realmente deberíamos escoger un método u otro según el que produjese funciones canónicas más cortas.

Parece complicado, verás que es mucho más sencillo de entender si se hace un ejemplo.

## Ejercicio resuelto

Obtén la función lógica que se corresponde con la siguiente tabla de verdad:

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	1	*
1	1	0	0	
1	1	1	1	*

Imagen de elaboración propia

### Mostrar retroalimentación

En primer lugar aplicaremos el método de **minterms**, o por el método de los "unos", o por suma de productos.

Para ello observamos todas las combinaciones de las variables de entrada que hacen que la salida sea un 1 (señaladas con un asterisco) y obtendremos como función canónica:

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

Es decir la suma de los productos de las cinco combinaciones de las variables de entrada que dan un 1 en la salida, en donde las variables aparecen negadas si son 0 y afirmadas si son 1.

Vamos a repetir el ejercicio, pero ahora lo haremos por el método de **maxterms**, o por el método de los "ceros", o por productos de sumas.

Para ello observamos todas las combinaciones de las variables de entrada que hacen que la salida sea un 0 (sin macar con asterico) y obtendremos como función canónica:

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

Es decir, el producto de las sumas de las tres combinaciones de las variables de entrada que dan un 0 en la salida, en donde las variables aparecen negadas si son 1 y afirmadas si son 0.

En este ejemplo sería más sencillo obtener la función canónica por el segundo método, ya que da como resultado una función canónica más corta.

Las dos funciones son equivalentes, ya que se obtienen de la misma tabla de verdad.



## *Ejercicio resuelto*

---

Obten la función canónica en el siguiente ejercicio:

[Descarga pdf](#)

## 5. Cronogramas



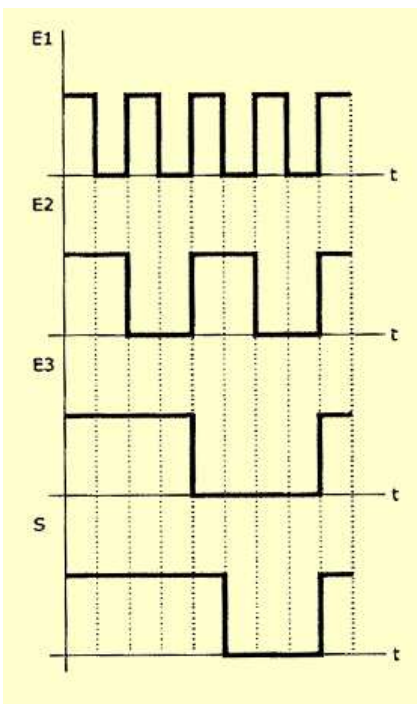
En ocasiones en vez de emplearse la tabla de verdad para indicar las combinaciones de las variables de entrada que provocan un determinado valor de la salida, se emplea un método gráfico en el que con un eje de tiempo coincidente para todas las variables de una función lógica, se representa por niveles altos y bajos (1 y 0) que van tomando las variables.

Vamos a verlo con un ejemplo.

Para transformar el siguiente cronograma de una determinada función lógica, en el que E1, E2 y E3 corresponde a las entradas y S es la salida, en primer lugar vamos a obtener su tabla de verdad, y a partir de esta información conseguimos la función canónica.

### *Ejercicio resuelto*

En el siguiente cronograma de una función lógica, E1, E2 y E3 corresponden con las entradas y S la salida. Obtén su función canónica.



1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Imagen de elaboración propia

A partir de esta información conseguimos la función canónica:

$$S = \overline{E_1} \cdot \overline{E_2} \cdot E_3 + \overline{E_1} \cdot E_2 \cdot E_3 + E_1 \cdot \overline{E_2} \cdot E_3 + E_1 \cdot E_2 \cdot \overline{E_3} + E_1 \cdot E_2 \cdot E_3$$

## Ejercicio resuelto

1. A partir del cronograma adjunto, escribe su tabla de verdad e implementa el circuito lógico resultante con cualquier tipo de puertas.

[Descargar pdf](#)

2. La salida Y de un circuito lógico depende de dos entradas A Y B y su función puede describirse mediante el cronograma de la figura adjunta.

[Descargar pdf](#)

## 6. Implementación de circuitos con puertas NAND y NOR



Para la implementación de circuitos lógicos se pueden utilizar cualquier tipo de puertas. Sin embargo la tendencia más común es implementar un circuito empleando solamente un tipo de puertas. De este modo se abaratan costes.

Este método de implementación solo se puede realizar con puertas NAND o NOR, ya que solo estas dos puertas lógicas son **universales**, es decir se puede realizar cualquier circuito lógico y sustituir cualquier puerta empleando únicamente este tipo de puertas, para ello debemos seguir un cierto protocolo aprovechando que una doble negación es igual a una afirmación ( $A = \overline{\overline{A}}$ ).

Siempre podemos negar una expresión lógica dos veces, tantas veces como necesitemos y ésta quedará inmutable, si luego aplicamos el teorema de DeMorgan a una de las dos negaciones anteriores, conseguimos que un producto negado se convierta en una suma de variables negadas, o bien que una suma negada se convierta en un producto de variables negadas.

Ejemplo de ejecución de cualquier puerta empleando solamente puertas NOR.


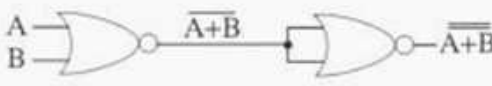
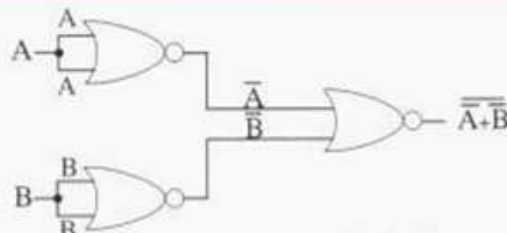
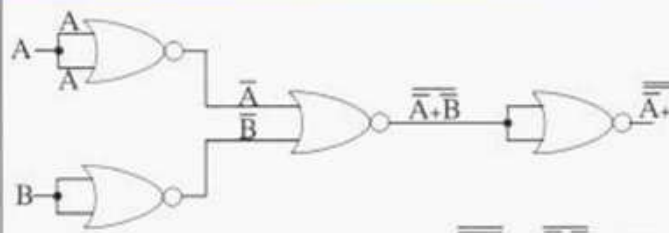
puerta NOT	puerta OR
 <p>Nota: <math>\overline{\overline{A}} = A</math></p>	 <p><math>\overline{\overline{A+B}} = A+B</math></p>
puerta AND	puerta NAND
 <p>(i) <math>\overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}} = A \cdot B</math></p>	 <p>(i) <math>\overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}} = A \cdot B</math></p>

Imagen de elaboración propia

De igual forma podemos utilizar puertas lógicas NAND:

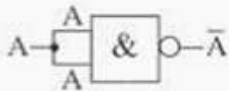
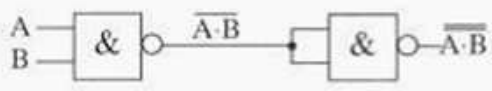
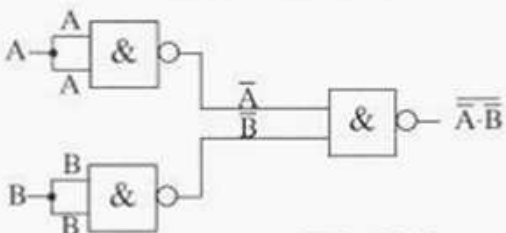
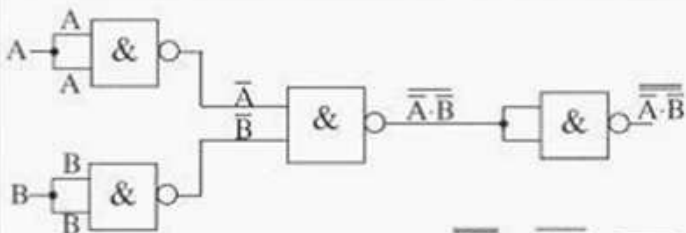
puerta NOT	puerta AND
 <p>Nota: <math>\overline{\overline{A}} = A</math></p>	 <p><math>\overline{\overline{A \cdot B}} = A \cdot B</math></p>
puerta OR	puerta NOR
 <p>(1) <math>\overline{\overline{A \cdot B}} = \overline{A \cdot B} = A + B</math></p>	 <p>(1) <math>\overline{\overline{A \cdot B}} = \overline{A \cdot B} = A + B</math></p>

Imagen de elaboración propia

## Ejercicio resuelto

La expresión adjunta corresponde a una determinada función lógica:

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

Se desea expresar esta función lógica en forma de productos negados (puertas NAND) o en forma de sumas negadas (Puertas NOR)

### Mostrar retroalimentación

La función permanece inmutable si la negamos dos veces:

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

Aplicando el teorema de Morgan a la negación:

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

Así habremos conseguido expresar la función en forma de productos negados, es decir de modo que se pueda implementar con puertas NAND.

Si lo que quisiéramos es expresar la función en forma de sumas negadas, (puertas NOR), actuaríamos del siguiente modo:

Se niega toda la función dos veces, y se niegan dos veces todos los productos que aparecen en la expresión lógica:

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

Y así obtenemos la expresión lógica en función de sumas negadas, que es lo se pretendía.

## Ejercicio resuelto

Implementar la función lógica que se adjunta empleando únicamente puertas NAND.  
Repetir el ejercicio empleando únicamente puertas NOR.

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

### Mostrar retroalimentación

En primer lugar obtendremos la solución **utilizando puertas NAND**.

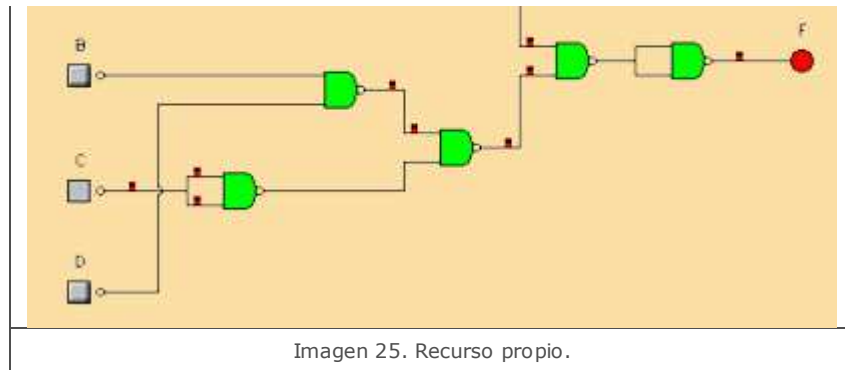
Se aplica una doble negación a toda la expresión y además se niega dos veces la suma:

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

Aplicamos ahora el teorema de Morgan a una de las negaciones de la suma y obtenemos:

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

Que al implementarse quedará:



Resolveremos ahora el problema **utilizando puertas NOR**.

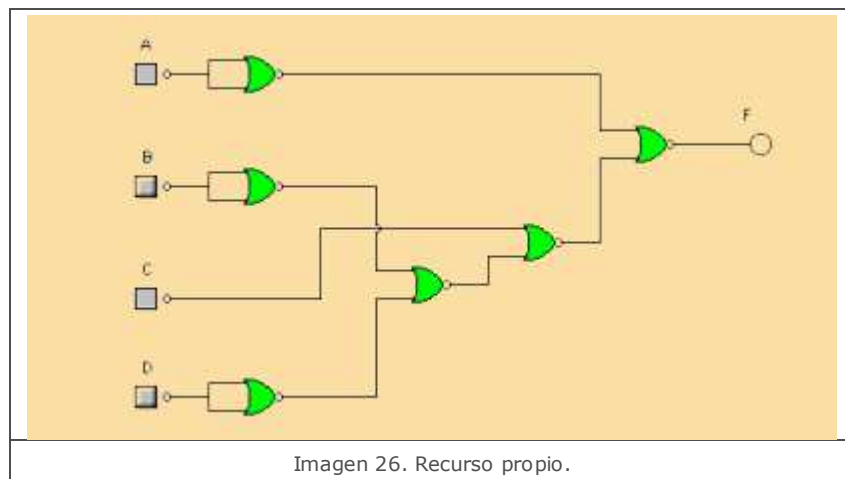
Negamos dos veces toda la expresión y el producto interior del paréntesis:

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

Aplicando el teorema de DeMorgan por un lado a una de las negaciones que abarca a toda la expresión y por otro al producto interior, tendremos:

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

Que al implementarse quedará:



## *Ejercicio resuelto*





## Importante

### 1. Operaciones básicas

Operación	Símbolo	Salida
Producto lógico	$a \cdot b = c$	La salida c toma el valor 1 si a y b también lo son.
Suma lógica	$a + b = c$ $\bar{a} = b$	La salida c toma el valor 1 si a, b o ambas toman el valor 1
Negación		La salida b toma el valor 1 si a toma el valor 0

## Importante

### 2. Otras operaciones lógicas

Operación	Símbolo	Salida
NAND	$S = \overline{A \cdot B}$	La salida toma el valor 1 si A y B no toman simultáneamente el valor 1.
NOR	$S = \overline{A + B}$	La salida toma el valor 1 si A y B toman a la vez el valor 0
XOR	$S = \overline{A} \cdot B + A \cdot \overline{B}$	La salida toma el valor 1 si una entrada es 0 y la otra es 1.
XNOR	$S = A \cdot B + \overline{A} \cdot \overline{B}$	La salida toma el valor 1 si las dos entradas son 0 ó 1 simultáneamente

## Importante

### 3. Álgebra de Boole

#### Postulados

Respecto a la suma	Respecto al producto
$A + 0 = A$	$A \cdot 0 = 0$
$A + 1 = 1$	$A \cdot 1 = A$
$A + A = A$	$A \cdot A = A$
$A + \bar{A} = 1$	$A \cdot \bar{A} = 0$
$\bar{\bar{A}} = A$	

#### Teoremas

Leyes/Teoremas	Respecto a la suma	Respecto al producto
Commutativa.	$A + B = B + A$	$A \cdot B = B \cdot A$
Asociativa	$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Distributiva	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
Cancelación	$(A \cdot B) + A = A$	$(A + B) \cdot A = A$
De Morgan	$\overline{A + B} = \bar{A} \cdot \bar{B}$	$\overline{A \cdot B} = \bar{A} + \bar{B}$

## Importante

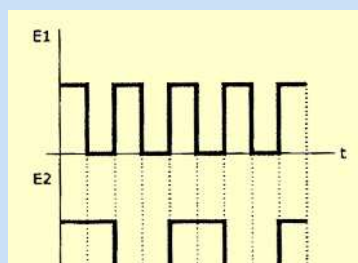
### 4. Función lógica a partir de la tabla de verdad

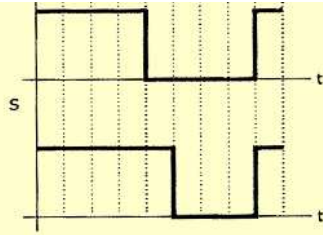
Una vez obtenida la tabla de verdad de un sistema, existen dos métodos para obtener la función lógica en la forma canónica que la representa:

- Minterms: implementación por unos.
- Maxterms: implementación por ceros.

### 5. Cronogramas

Es un método gráfico en el que se representa por niveles altos y bajos (1 y 0) que van tomando las variables. Permite obtener la tabla de verdad y la función lógica.





## *Importante*

### **6. Implementación de circuitos con puertas NAND y NOR**

Se puede implementar un circuito empleando solamente un tipo de puertas abarata costes.

Este método de implementación solo se puede realizar con puertas NAND o NOR, ya que solo estas dos puertas lógicas son universales

# Uf. Tenemos problemas para encontrar ese sitio.



No podemos conectar al servidor en [adistancia.ced.junta-andalucia.es](http://adistancia.ced.junta-andalucia.es).

**Si esa dirección es correcta, aquí hay otras tres cosas que puede probar:**

- Vuelva a intentarlo más tarde.
- Compruebe su conexión de red.
- Si está conectado a través de un cortafuegos, compruebe que Firefox tiene permiso para acceder a la web.

Reintentar