IES NESTOR ALMENDROS
DPTO. DE TECNOLOGÍA

ALGEBRA DE BOOLE

INTRODUCCIÓN

(George Boole, matemático inglés, 1815 - 1864)

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.

CIRCUITOS DIGITALES FUNDAMENTALES Y ALGEBRA DE BOOLE

Los circuitos basados en el álgebra de boole son circuitos lógicos, y se pueden distinguir dos tipos de lógica:

Lógica positiva: Al nivel más elevado de tensión se le asigna el estado **1** y al más bajo el estado **0**.

Lógica negativa: Al nivel más elevado de tensión se le asigna el estado **0** y al más bajo el estado **1**.

Una función lógica es aquella cuyo valores son binarios y dependen de una expresión algebraica, formada por una serie de variables binarias relacionadas entre si por determinadas operaciones.

Las operaciones lógicas fundamentales en las que se basan los circuitos digitales son tres:

1. **Suma Lógica.**
2. **Producto Lógico.**
3. **Complementación.**

Los circuitos que las realizan son denominados circuitos lógicos o digitales. El soporte matemático de los circuitos lógicos o digitales, es el Algebra de Boole, un conjunto de reglas matemáticas que trata con variables binarias y se basa en las tres operaciones anteriormente indicadas.

Las expresiones se corresponden con un determinado circuito lógico, o sea expresan circuitos físicos.

OPERACIONES LOGICAS FUNDAMENTALES.

Suma lógica. Su expresión matemática en álgebra de Boole es: **A + B**

A y **B** son variables binarias, sólo pueden tomar los estados **1** y **0**. La operación se define por la tabla siguiente:

ENTRADAS		SALIDA
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

El resultado de la suma lógica vale **1** cuando las variables **A** y **B** valen **1**, o cuando las dos estén a **1**, a diferencia de la suma binaria aritmética, cuando **A** y B valen **1** el resultado es **1**; en cambio en la suma aritmética el resultado es **0** y se produce un acarreo.

La suma lógica y suma aritmética, no son lo mismo.

A este tipo de tabla se le denomina tabla de verdad y se indican las posibles combinaciones entre las variables y el estado correspondiente que toma la salida..

Producto lógico. Su expresión lógica es: **A x B.** La operación se define en la tabla adjunta.

ENTRADAS		SALIDA
A	**B**	**A x B**
0	**0**	**0**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**

El resultado vale **1** sólo cuando las dos variables están a **1**, basta que una de las dos variables esté a **0** para que el resultado también esté a **0**.

Complementación. Esta operación se aplica sobre una sola variable y su expresión lógica es igual a la variable: $\mathbf{B = \overline{A}}$. La tabla que define esta operación es la siguiente:

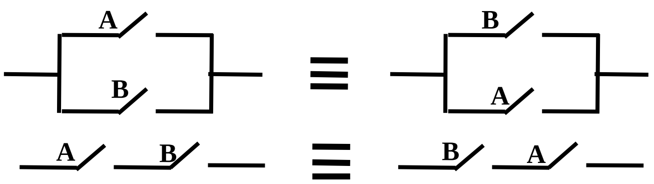
Entrada	Salida
A	**B**
0	**1**
1	**0**

La complementación también se suele denominar **inversión** o **negación**. Para la realización de las operaciones lógicas se puede utilizar cualquier dispositivo que pueda operar en binario, o que se pueda hacer funcionar en dos estados bien diferenciados: interruptores, relés, transistores, válvulas hidráulicas o neumáticas, etc.

La doble negación devuelve el valor a su estado no negado: $\mathbf{\overline{\overline{A}} = A}$. Aunque los operadores lógicos son normalmente dispositivos electrónicos. Estos dispositivos que realizan las operaciones lógicas se denominan **puertas lógicas,** y aparecen en forma integrada; son los denominados circuitos integrados digitales, en ellos se pueden contener varias puertas lógicas. Partiendo de estos integrados, se pueden realizar circuitos más complejos como decodificadores, multiplexores, contadores, etc. Mediante estos otros circuitos lógicos realizaremos otros aún más complejos: memorias, unidades lógico-aritméticas (ALU), microprocesadores, etc. Se puede decir que el circuito básico de los sistemas digitales es la puerta lógica.

PROPIEDADES DEL ÁLGEBRA DE BOOLE

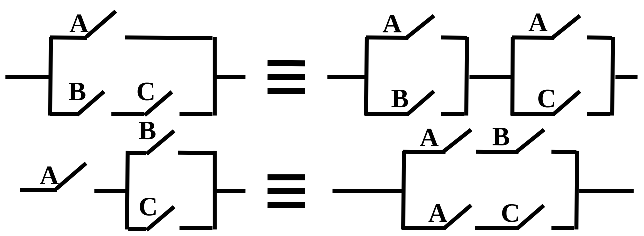
- **Propiedad interna.** El resultado de una operación de dos variables booleanas es otra variable booleana.
- **Propiedad idempotencia.** Si **A** es una variable booleana se cumple: **A + A = A**
  A x A = A
- **Ley de involución.** Para una variable booleana se cumple $\mathbf{\overline{\overline{A}} = A}$
- **Propiedad conmutativa.** Si **A** y **B** son dos variables booleanas, se verifica:



A + B = B + A

A x B = B x A

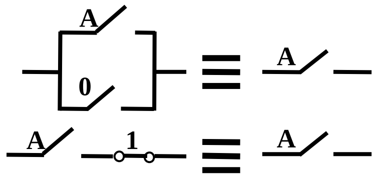
- **Propiedad asociativa.** Si **A**, **B** y **C** son tres variables booleanas, se cumple:
  A + B + C = (A + B) + C = A + (B + C)
  A x B x C = (A x B) x C = A x (B x C)
- **Propiedad distributiva.** Si A, B y C son tres variables booleanas, se cumple:



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

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

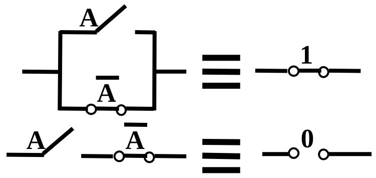
- **<u>Existencia de elemento neutro.</u>** Existe un elemento neutro para la suma y otro para el producto.



$$A + 0 = A$$

$$A \times 1 = A$$

- **<u>Existencia de elemento opuesto.</u>** Existe un elemento opuesto para la suma y otro para el producto.



$$A + \overline{A} = 1$$

$$A \times \overline{A} = 0$$

- **<u>Ley de absorción.</u>** Para las variables booleanas **A** y **B**, se cumple:

$$A + A \times B = A \quad \text{y} \quad A \times (A + B) = A$$

- **<u>Ley simplificativa.</u>** Para las variables booleanas **A** y **B**, se cumple:

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

- **<u>Leyes de Morgan.</u>** Las leyes de Morgan se pueden generalizar en más de dos variables. Demostración:

Entradas		Salidas	
A	B	$\overline{A + B}$	$\overline{A} \times \overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

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

Entradas		Salidas	
A	B	$\overline{A \times B}$	$\overline{A} + \overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

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

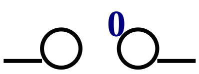
<u>PUERTAS LOGICAS</u>

Aspectos fundamentales sobre lógica de contactos

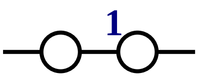
Cuestiones importantes sobre la realización de las funciones lógicas mediante circuitos con interruptores, ya que la asociación de circuitos eléctricos a las expresiones lógicas facilita enormemente la comprensión de los circuitos lógicos.

Las premisas básicas de asimilación son:

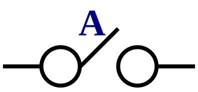
1- Cero lógico **0**: circuito abierto (no hay paso de corriente).



2- Uno lógico **1**: Circuito cerrado (pasa la corriente).

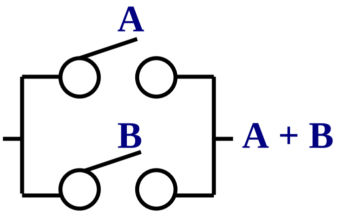


3- Variable: interruptor (abierto: **0**, cerrado: **1**).

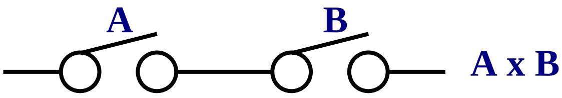


4- Complementación (variable negada): Interruptor normalmente cerrado (NC); en reposo circula corriente (el paso de corriente se interrumpe al activarlo).

5- Operación suma lógica: Interruptores en paralelo.



6- Operación producto lógico: Interruptores en serie.



PUERTAS LÓGICAS ELEMENTALES

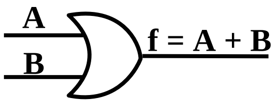
Las puertas lógicas elementales son aquellas que realizan operaciones de álgebra con variables booleanas de suma, producto e inversión. Puertas: **OR, AND** y **NOT.**

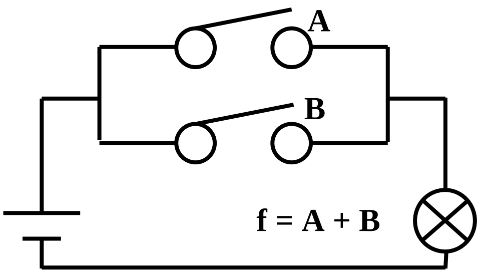
PUERTA O (OR)

Esta puerta realiza la operación suma lógica. Su expresión lógica booleana es: **f = A + B**
La **"f"** indica función lógica; es otra variable binaria que depende del resultado de la operación con las variables A y B. Tabla de funcionamiento de la puerta **O**

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

La puerta **O** proporciona pues, el estado lógico uno **1** siempre que alguna de las dos entradas se encuentre a uno **1** , o las dos a la vez.





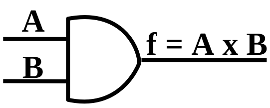
También representamos el circuito eléctrico de la función **O/OR**.
Un ejemplo de Circuito Integrado que contiene puertas **"O/OR"** es el **7432**, este dispone de cuatro puertas **O/OR** de dos entradas cada una.

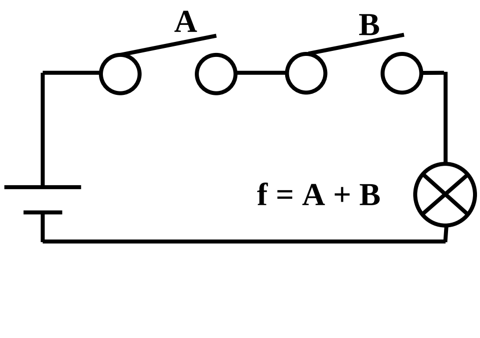
PUERTA Y (AND)

La puerta **Y (AND)** realiza el producto lógico y su expresión lógica es: **f = A x B**
Tabla de la verdad:

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

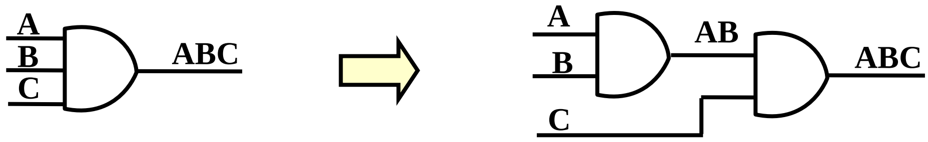
También representaremos la correspondencia como circuito eléctrico que realiza la función **"Y/AND"**





El circuito integrado **7408** sería el ejemplo de un Circuito Integrado de cuatro puertas **Y/AND** de dos entradas.

También cabe reseñar, que para obtener una puerta **Y/AND** de tres entradas podríamos utilizar dos puertas **Y/AND** de dos entradas y conexionarlas según esquema.



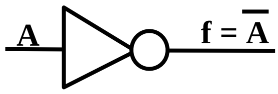
PUERTA NO (NOT): inversor

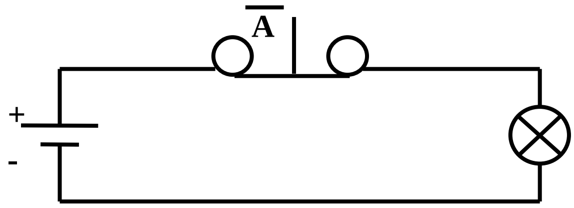
Esta puerta hace la operación de complementación, también conocida como **negación** o simplemente **inversión**. Es el único operador lógico que tiene sólo una entrada. Su expresión lógica es: $f = \overline{A}$. Esto significa que la función "f" siempre toma el estado contrario de la variable **"A"**. $A = 0,\quad f = \overline{A} = \overline{0} = 1,\qquad A = 1,\quad f = \overline{A} = \overline{1} = 0$

Siendo la tabla de la verdad la siguiente:

Entrada	Salida
A	f
0	1
1	0

La equivalencia eléctrica correspondiente al inversor se basa en un pulsador del tipo NC (normalmente cerrado), como se muestra





Al pulsar la lámpara se apaga. Sin pulsar, la lámpara está encendida. Es un pulsador del tipo NC. Un ejemplo de circuito integrado sería el **7404**, compuesto por seis puertas lógicas del tipo **NO/NOT**.

PUERTAS LÓGICAS UNIVERSALES

Las puertas lógicas universales son aquellas que realizan operaciones de negación de las puertas elementales y además son capaces de sustituirlas. Puertas: **NOR** y **NAND.**

PUERTA NO-O (NOR).

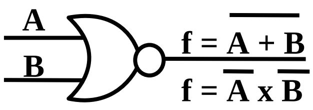
Esta puerta realiza la operación suma lógica negada. Su expresión lógica es: $f = \overline{A + B}$

Como podemos observar esta es la expresión de la puerta **"O/OR"** complementada o negada, su operación es por tanto, la suma lógica complementada.

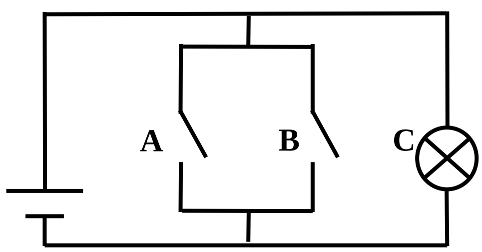
Tabla de la verdad:

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

Para que en la salida el valor sea **1**, no tiene que estar ninguna entrada activada a **1**, si la hay su salida será **0**.

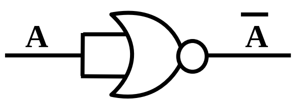


Esta puerta se basa en la combinación de una puerta **O/OR** y una **NO/NOT.** La puerta **NO-O** también se puede comportar como un inversor, basta con unir las entradas o conectar a masa las entradas no utilizadas. Como ejemplo de CI damos al **7402**, compuesto por cuatro puertas de dos entradas.



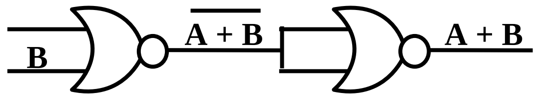
Si unimos las dos entradas en una sola se obtiene la función **NOT**

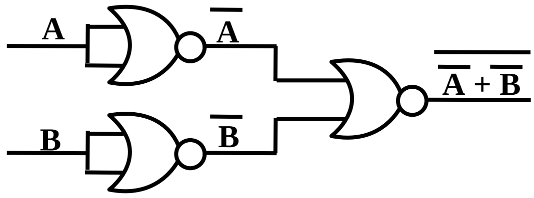
Cuando ambas entradas son iguales, como se demuestra en la 1ª fila y última de la tabla, se obtiene la función **NOT**



En la función **NOR**, sólo cuando ambos interruptores están abiertos **C** estará encendida (**1**)

La función **OR**, se obtiene de la siguiente forma:





La función **AND** se puede obtener a partir de a partir de puertas **NOR**, para ello es necesario recurrir a las leyes de Morgan.

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

PUERTA NO-Y (NAND).

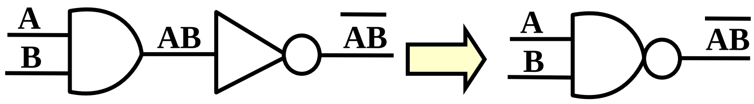
Esta puerta realiza la función de producto lógico negado, o sea, el producto lógico seguido de una complementación o inversión. Su expresión lógica es: $f = \overline{A \times B}$

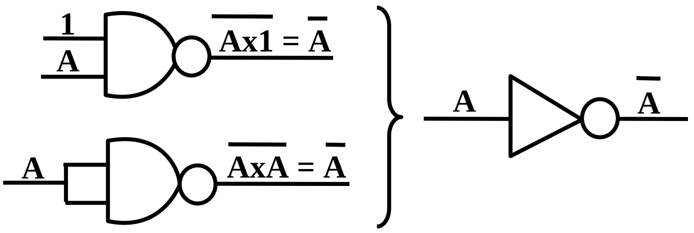
Tabla de la verdad

ENTRADAS		SALIDA
A	**B**	**f**
0	0	1
0	1	1
1	0	1
1	1	O

La puerta **NO-Y/NAND** se caracteriza porque se precisa que todas sus entradas estén a **1** para que la salida sea **0**, siempre que una entrada esté a **0** su salida será **1**.

La puerta **NO-Y/NAND** es la combinación de una puerta **Y** y una **NO**.

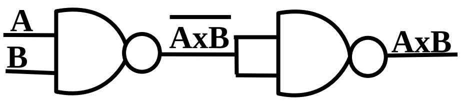


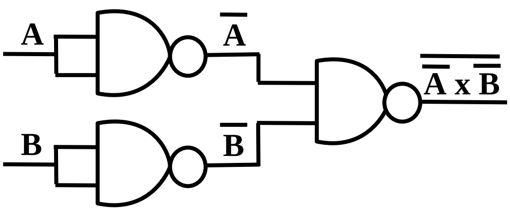


Entre otras utilidades, la puerta **NO–Y (NAND)**, también se puede utilizar como un inversor **NO/NOT**.

El circuito integrado **7400** (puertas **NO-Y/NAND**) compuesto por cuatro puertas de 2 entradas es el más popular.

La función **AND** se consigue por la definición de la puerta **NAND**





Para la función **OR** es necesario recurrir a las leyes de Morgan:

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

OTRAS PUERTAS

Además de las puertas elementales y universales, existen otras puertas menos utilizadas, pero no por ello menos importante. **XOR (O-EXCLUSIVA)** y **XNOR (EQUIVALENCIA)**.

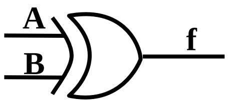
PUERTA O-exclusiva (Exclusive - OR)/ XOR.

Esta puerta también se conoce por **XOR** y **EXOR**. Realiza la operación que lleva su nombre, que es una variante de la **O**. Su expresión lógica es: $f = A \oplus B = \overline{A} \times B + A \times \overline{B}$

Esta función puede obtenerse partiendo de otras funciones básicas.

Tabla de la verdad:

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



Esta puerta responde a: "*La salida de un **XOR** de dos entradas permanece en estado **1** si una y sólo una de las entradas está en estado **1***"

ENTRADAS			SALIDA
A	**B**	**C**	**f**
0	**0**	**0**	**0**
0	**0**	**1**	**1**
0	**1**	**0**	**1**
0	**1**	**1**	**0**
1	**0**	**0**	**1**
1	**0**	**1**	**0**
1	**1**	**0**	**0**
1	**1**	**1**	**1**

En general una puerta **XOR** se caracteriza porque su salida es **1** sólo cuando el número de entradas activadas es impar. Por esta razón, entre otras aplicaciones, se utiliza como detector de imparidad.

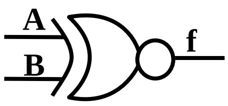
La siguiente tabla corresponde a una puerta **XOR (O-exclusiva)** de tres entradas.

Un circuito integrado compuesto por cuatro puertas **XOR** de dos entradas es el **7486**.

PUERTA DE EQUIVALENCIA / XNOR

Esta puerta es la negada de la puerta **XOR**, se le llama **XNOR** o **EQUIVALENCIA**.

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



Esta puerta responde a: "*La salida de una puerta **XNOR/ EQUIVALENCIA** de dos entradas, permanece en estado **1** si ámbas entradas son iguales*"

$$f = \overline{A \oplus B} = A \times B + \overline{A} \times \overline{B}$$

REPRESENTACIÓN DE FUNCIONES LÓGICAS

Una misma función lógica puede representarse mediante varias formulaciones equivalentes, aunque la tabla de verdad será siempre la misma.

Lo ideal será usar para una misma tabla la formulación matemática más sencilla y simplificada y así tendremos el circuito con menos puertas lógicas y por lo tanto el más económico, el que menos espacio ocupa y el que menos energía consume.

Para obtener la expresión algebraica de una función, a partir de su tabla de verdad, se escribe la ecuación cómo una suma de productos, de los términos cuyo valor de salida en la tabla sea 1. Las variables de la función se expresan de forma directa cuando aparezca un 1, y de forma negada cuando aparezca un 0

Ejemplo: Obtener la expresión algebraica y el circuito lógico más económico de la función que viene dada por la tabla de verdad:

ENTRADAS			SALIDA
A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	1 *
0	1	1	1 *
1	0	0	1 *
1	0	1	1 *
1	1	0	0
1	1	1	0

El primer término con valor de salida **1** será:

$$(0 , 1 , 0) = \overline{A} \times B \times \overline{C}$$

$$f = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C$$

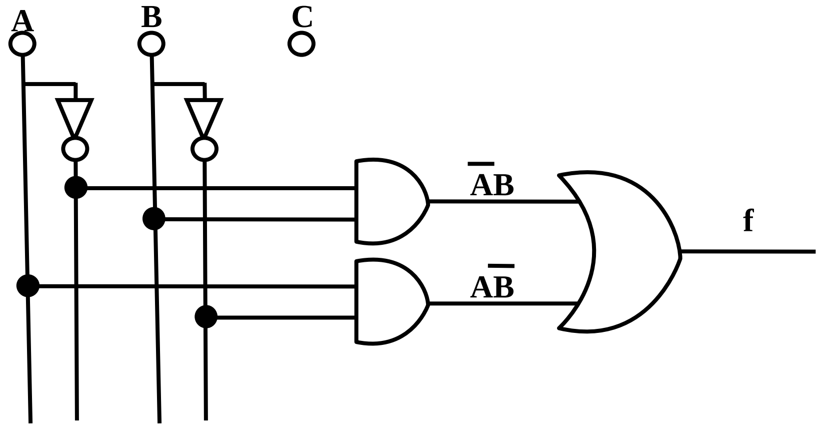
Simplifiquemos algebraicamente:

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

$$= \overline{A}B(\overline{C}+C) + A\overline{B}(\overline{C}+C) \text{ como } (\overline{C}+C) = 1$$

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

Si comprobamos su tabla de verdad veremos que es la misma. El circuito lógico será:



Observa que el valor de **C**, en este caso no hace cambiar la tabla de verdad, por lo que a la función obtenida no le afecta el valor de entrada de esta variable.

Problema 1º

Construye la tabla de verdad y el circuito lógico, correspondiente a la función:

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

FORMAS CANÓNICAS DE UNA FUNCIÓN

Entre las diversas representaciones matemáticas que puede tomar una función, existen dos de gran utilidad, que se denominan **Formas Canónicas**.

- **Primera forma canónica de una función lógica:** - Es una suma de productos lógicos , en los que intervienen todas las variables de la función, ya sea de forma directa o negada.
  - Cada termino o producto canónico se representa por m_i siendo **i** = Valor decimal de la combinación binaria, obtenida al sustituir por **1** las variables que aparecen de forma directa en el producto, y por **0** las que lo hacen de forma negada.

  Ejemplo: $f(A , B , C , D) \longrightarrow m_5$ cuyo valor binario es **(0 1 0 1)** representa el producto de : $(\overline{A} \times B \times \overline{C} \times D)$

- **Segunda forma canónica de una función lógica** - Es el producto de sumas lógicas, donde intervienen las variables de forma directa o negada.
  - Cada término se representa por M_i teniendo **i** = Valor decimal de la 1ª forma canónica.

Ejemplo: $f(A, B, C, D) \longrightarrow M_5$ cuyo valor binario es **(0 1 0 1)** representa la suma de : $(\overline{A} + B + \overline{C} + D)$

- Las dos formas canónicas son expresiones matemáticas de la función, las dos son equivalentes y poseen la misma tabla de verdad.

- Para obtener las formas canónicas a partir de la tabla de verdad, tendremos en cuenta:

- **- 1ª Forma Canónica.** Aparecen aquellos términos que corresponden a una salida (combinación determinada de las entradas) de valor **1**, cuando no figuran los términos correspondientes al valor **0**. La función se representa de forma directa para las variables que representan un **1** en la tabla de verdad, y de forma negada las que presentan un **0**.

- **- 2ª Forma Canónica.** Aparecen Aquellos términos cuya salida es **0**, no apareciendo los de valor **1**. En este caso, se escriben de forma directa aquellas variables que representan un **0** en la tabla de verdad, y de forma negada las que presentan un **1**.

Ejemplo: Obtener las formas canónicas de la función que viene dada por la tabla de verdad:

ENTRADAS			SALIDA	
A	**B**	**C**	**f**	**dec**
0	0	0	0	0
0	0	1	0	1
0	1	0	1	2
0	1	1	1	3
1	0	0	1	4
1	0	1	1	5
1	1	0	1	6
1	1	1	0	7

1ª Forma Canónica: Suma de los productos con salida **1**

$$f(A,B,C,) = m_2 + m_3 + m_4 + m_5 + m_6$$
$$f = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C}$$

2ª Forma Canónica: Producto de sumas con salida **0**

$$f(A,B,C,) = M_0 + M_1 + M_7$$
$$f = (A+B+C) \times (A+B+\overline{C}) \times (\overline{A}+\overline{B}+\overline{C})$$

Simplificación 1ª Forma canónica:

$$f = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} = \overline{A}B(\overline{C}+C) + A\overline{B}(\overline{C}+C) + AB\overline{C}$$
$$f = \underbrace{\overline{A}B + A\overline{B} + AB\overline{C}}_{(a)} = \underbrace{\overline{A}B + A(\overline{B}+B\overline{C})}_{(b)}$$

Si comprobamos la tabla de verdad, veremos que coincide.

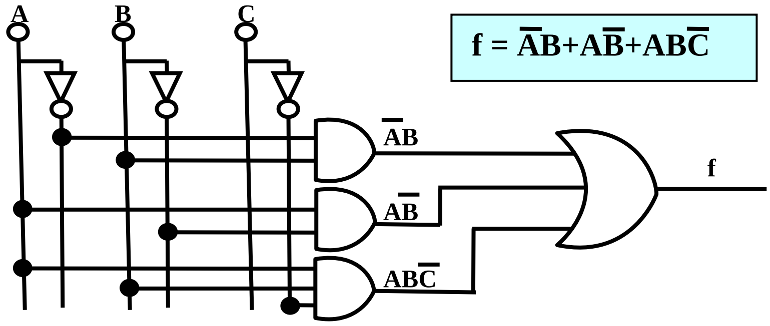
Simplificación 2ª Forma canónica:

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

Si comprobamos la tabla de verdad, veremos que coincide.
La función matemática más simplificada resulta ser la de la 2ª forma canónica.

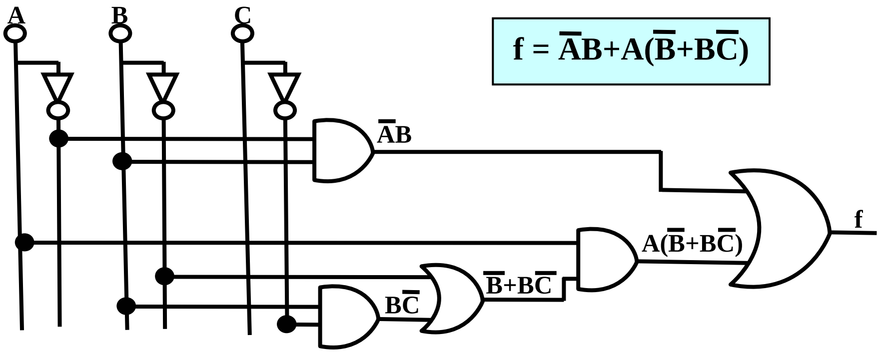
- Veamos que circuito lógico lleva menos componentes:

<u>Circuito lógico de la 1ª Forma Canónica (a):</u>



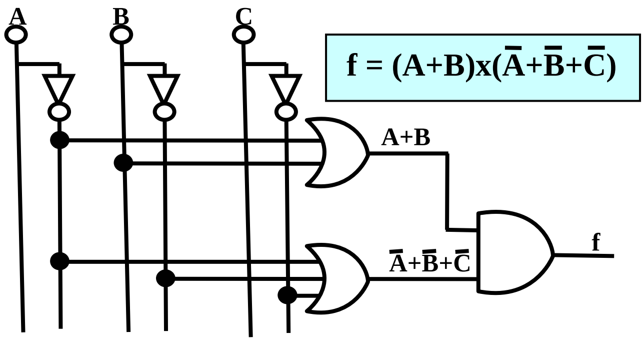
Es un circuito sencillo, pero no el más económico

<u>Circuito lógico de la 1ª Forma Canónica (b):</u>



Resulta ser la forma más cara y compleja

<u>Circuito lógico de la 2ª Forma Canónica:</u>



Resulta ser la forma más económica

<u>Problema 2º</u>

Simplifica la ecuación lógica por métodos algebraicos, representa su tabla de verdad y dibuja el circuito lógico más económico.

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

<u>Problema 3º</u>

Simplifica la ecuación lógica por métodos algebraicos, representa su tabla de verdad y dibuja el circuito lógico más económico.

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

<u>Problema 4º</u>

Dibujar el circuito lógico, usando puertas **NAND**, de la función:

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