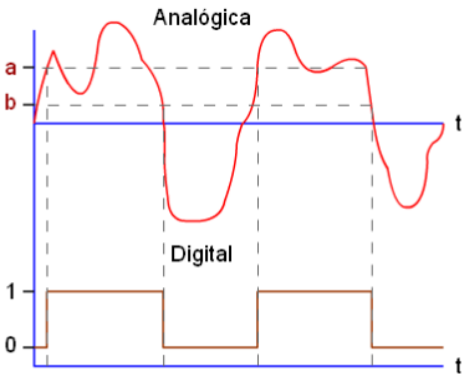
TEMA:

PUERTAS LÓGICAS. TÉCNICAS DE DISEÑO Y SIMPLIFICACIÓN DE FUNCIONES LÓGICAS

INDICE

1.- INTRODUCCIÓN A LA ELECTRÓNICA DIGITAL

El gran desarrollo experimentado por la Electrónica en los últimos años ha propiciado que la mayoría de los equipos actuales funcionen con sistemas digitales. Un sistema digital se caracteriza por utilizar **señales discretas**, es decir, señales que toman un número finito de valores en cierto intervalo de tiempo.
La comparación gráfica entre una **señal analógica** y **una digital** es la siguiente:



Una señal analógica puede tener infinitos valores, positivos y/o negativos. La señal va variando de manera progresiva a medida que pasa el tiempo.

La señal digital puede adquirir únicamente un número finito de valores, sólo dos valores concretos: uno máximo (que asignaremos el valor lógico 1) y uno mínimo (valor lógico 0), que pueden ser impulsos eléctricos de alta y baja tensión, interruptores cerrados o abiertos, etc. La gran ventaja es que la señal digital es más fiable en la transmisión de datos.

Comparación gráfica de una señal analógica frente a una señal digital.

La mayoría de las magnitudes que se miden o se transmiten en la vida cotidiana son señales analógicas: el sonido, la temperatura de un horno, la velocidad del viento… son señales que pueden tomar cualquier valor dentro de un intervalo. Sin embargo los sistemas digitales trabajan solamente con dos posibles valores (1 y 0).

Todas las telecomunicaciones modernas (Internet, telefonía móvil, etc.) están basadas en el uso de este tipo de sistemas, por lo que el estudio de las mismas resulta de gran importancia. El utilizar circuitos y sistemas que trabajen sólo con números tiene una ventaja muy importante: se pueden realizar manipulaciones con independencia de la señal que se esté introduciendo: datos, voz, vídeo... Un ejemplo muy claro es internet. Internet es una red digital, especializada en la transmisión de números. Y esos números pueden ser datos, canciones, vídeos, programas, etc... La red no sabe qué tipo de señal transporta, “sólo ve números”. La electrónica digital trabaja con números.

Un circuito digital realiza manipulaciones sobre los números de entrada y genera unos números de salida.

Números de entrada (2,5,8..) ⇒ Circuito digital ⇒ Números de salida (8,10,15,)

2.- SISTEMAS DE NUMERACIÓN

La información que se va a manejar en cualquier sistema digital tiene que estar representada numéricamente. Para ello, necesitaremos un sistema de numeración acorde con las características intrínsecas de este tipo de señales.

Un **sistema de numeración** se define como un conjunto de símbolos capaces de representar cantidades numéricas. A su vez, se define la **base del sistema** de numeración como la cantidad de símbolos distintos que se utilizan para representar las cantidades. Cada símbolo del sistema de numeración recibe el nombre de **dígito**.

Así, los **sistemas de numeración más utilizados** son:

- Sistema decimal o de base 10 - Consta de diez dígitos: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
- Sistema binario o de base 2 - Consta de dos dígitos: {0, 1}.
- Sistema octal o de base 8 - Consta de ocho dígitos: {0, 1, 2, 3, 4, 5, 6, 7}.
- Sistema hexadecimal o de base 16 - Consta de dieciséis dígitos: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.

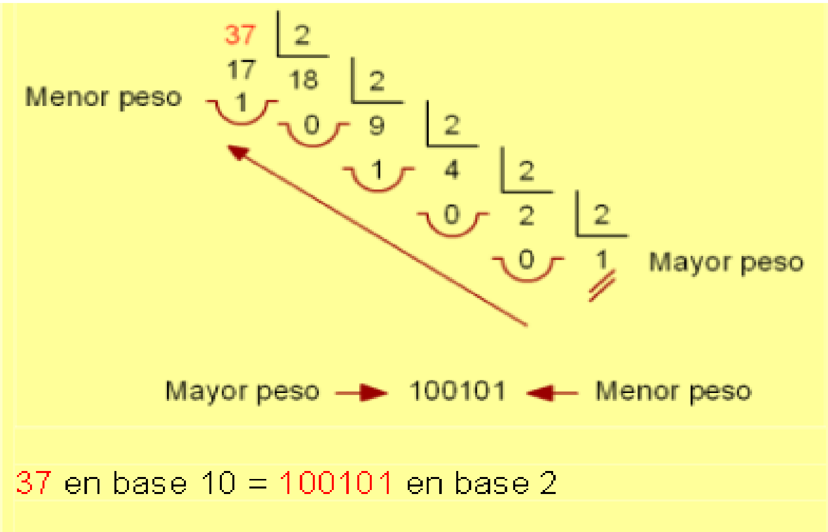
El sistema que utilizamos habitualmente es el sistema decimal, sin embargo los ordenadores y en general todos los sistemas que utilizan electrónica digital utilizan el **sistema binario**. En la electrónica digital sólo existen dos estados posibles (1 o 0) por lo que interesa utilizar un sistema de numeración en base 2, el sistema binario. Dicho sistema emplea únicamente dos caracteres, 0 y 1; que reciben el nombre de **bits**, que es la unidad mínima de información que se va a manejar en un sistema digital. Por ejemplo, el estado de una bombilla sólo puede tener dos valores (0 apagada, 1 encendida).

¿Cómo convertir un número decimal en un número binario?

Se divide el número del sistema decimal entre 2, el resultado entero se vuelve a dividir entre 2, y así sucesivamente hasta que el cociente sea menor que el divisor 2. Es decir, hasta que todos los restos sean 0 o 1.

El número binario será el formado por el último cociente (bit de mayor peso) y todos los restos. Los restos se ordenan empezando desde el último al primero. Éste será el número binario que buscamos.

Por ejemplo: El número 37 de base decimal lo podremos expresar en base binario.



$37_{(10} = 100101_{(2}$

¿Cómo convertir un número binario en un número decimal?

El sistema binario utiliza sólo dos dígitos diferentes para representar cualquier número. El peso de los dígitos es una potencia de 2.

Para realizar la conversión de binario a decimal, realice lo siguiente:

1. Inicie por el lado derecho del número en binario, cada cifra multiplíquela por 2 elevado a la potencia consecutiva (comenzando por la potencia 0, 2^0).
2. Después de realizar cada una de las multiplicaciones, sume todas y el número resultante será el equivalente al sistema decimal.

Ejemplos:

- (Los números de arriba indican la potencia a la que hay que elevar 2)

$$\overset{5}{1}\,\overset{4}{1}\,\overset{3}{0}\,\overset{2}{1}\,\overset{1}{0}\,\overset{0}{1}\,{}_2 = 1\cdot2^5+1\cdot2^4+0\cdot2^3+1\cdot2^2+0\cdot2^1+1\cdot2^0 = 32+16+0+4+0+1 = 53$$

$110101_{(2} = 53_{(10}$

3.- ÁLGEBRA DE BOOLE:

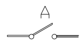
George Boole (1815-1864) fue un matemático y filósofo británico que inventó una serie de reglas para expresar y resolver problemas lógicos que sólo podían tomar dos valores. Estas reglas conforman lo que conocemos como el álgebra de Boole. El álgebra de Boole es un sistema matemático que utiliza variables y operadores lógicos. Las variables pueden valer 0 o 1. Y las operaciones básicas son la suma y la multiplicación.

George Boole, basándose en el sistema binario de numeración, desarrolló una teoría matemática a la que denominó álgebra de Boole cuyos postulados pueden aplicarse a todo tipo de elementos mecánicos, eléctricos, electrónicos, neumáticos y, en general a todos los dispositivos que trabajen con dos estados posibles, como es el caso de los componentes electrónicos utilizados en los ordenadores que, integrados de forma apropiada, son capaces de desarrollar todo tipo de tareas.

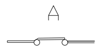
Dentro del microprocesador lo que realmente hay son innumerables interruptores que sólo pueden tomar dos posiciones " 0" o " 1 " según este apagado o encendido.

Explicamos a continuación los distintos valores que pueden tomar dichos interruptores.

Si el interruptor está en posición " 0" estará OFF (Abierto):



Si el interruptor está en posición " 1" estará ON (Cerrado):



Por lo tanto un **álgebra de Boole** es la estructura algebraica que corresponde a un conjunto de elementos, que pueden tomar los valores **0** y **1**, sobre los que se definen tres operaciones binarias: ***SUMA LÓGICA, PRODUCTO LÓGICO Y COMPLEMENTACIÓN O NEGACIÓN.***

SUMA LÓGICA:

Sobre los elementos **A** y **B** del conjunto se define una operación denominada **suma lógica S**, que se representa mediante el símbolo **+**.

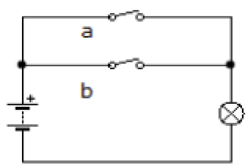
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

S = A + B (Suma lógica)

La tabla correspondiente a esta operación es la que aparece a la izquierda:

La función de salida S tomará el valor 1 si **A ó B** toman este valor **1.**

Si hacemos un símil con interruptores eléctricos, de forma que asignamos el valor 0 al interruptor abierto y el valor 1 al interruptor cerrado, la función de SUMA LÓGICA es equivalente a la conexión de ***dos interruptores en paralelo***, de forma que la salida del circuito se activa cuando uno o más interruptores están activados.



En el circuito de la figura se comprueba que la lámpara luce si y sólo si **"el interruptor "a" está pulsado"** o **"el interruptor "b" está pulsado"**

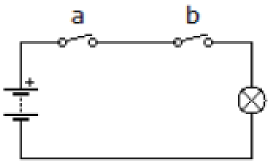
PRODUCTO LÓGICO:

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

Sobre los elementos **A** y **B** del conjunto se define una operación denominada **producto lógico F**, que se representa mediante el símbolo ***** o bien escribiendo una variable junto a la otra.   F = A * B = AB (Producto lógico)

La función de salida tomará el valor 1 si **A y B** toman este valor **1**.

El PRODUCTO LÓGICO equivale a colocar ***dos interruptores en serie***, de forma que la salida se activa solamente si todos los interruptores están activados. En el circuito de la figura se comprueba que la lámpara luce **si y sólo si** los interruptores **“a” y “b”** estén pulsados.



<u>COMPLEMENTACIÓN O NEGACIÓN</u>:

a	$\overline{a}$
0	1
1	0

Si lo que queremos es realizar la operación contraria **a** lo representamos como " $\overline{\mathbf{a}}$ ", que viene a significar que la función toma el valor contrario a " a". Si **a** vale 1, $\overline{\mathbf{a}}$ vale 0 y viceversa.

Todo elemento **a** del conjunto posee un **elemento simétrico** $\overline{\mathbf{a}}$ de tal forma que la suma y el producto lógico de cada elemento con su simétrico determinan, respectivamente, los valores **0** y **1**.

$$a + \overline{a} = 1 \qquad a * \overline{a} = 0$$

La siguiente tabla permite establecer la relación que existe entre los valores “a” y su simétrico y comprobar la validez de la definición.

a	$\overline{a}$	$a+\overline{a}$	$a*\overline{a}$
0	1	1	0
1	0	1	0

4.-TABLA DE VERDAD

Se denomina función lógica a una función matemática cuyo estado depende de variables binarias relacionadas por medio de operaciones básicas (suma lógica (+), producto lógico (*) o negación) del álgebra de Boole.

A	B	F
0	0	
0	1	
1	0	
1	1	

La tabla de verdad de una función lógica es un cuadro formado por tantas columnas como variables contenga la función más la correspondiente a ésta y por tantas filas como combinaciones binarias sea posible construir con dichas variables.
Las combinaciones binarias posibles serán $\mathbf{2^n}$, siendo **n** el número de variables y debiéndose ordenar las combinaciones binarias de forma creciente con el fin de evitar repeticiones.

Por ejemplo, una función lógica F que esté compuesta por 2 variables **A** y **B** tendrá $\mathbf{2^n} = 2^2$ combinaciones posibles, o sea 4. Desde los números decimales 0 hasta el 3, en sus correspondientes combinaciones binarias: 0 =00; 1 =01; 2 =10 y 3 =11.

4.1.- CONSTRUCCIÓN DE UNA TABLA DE VERDAD:

Para construir la tabla de verdad de una función lógica, hay que descomponer ésta en una serie de operaciones lógicas simples y analizarlas una por una. Veamos un ejemplo.
Supongamos una función lógica **F**, que depende de dos variables **A** y **B**: $\mathbf{F= AB +\overline{A}B}$

Para confeccionar su tabla de verdad, procederemos del modo siguiente.

- Determinamos. en primer lugar, el número de variables **n** de las que depende la función. En este caso son dos **A** y **B**. Por tanto: **n=2**
- A partir de este dato, calculamos todas las posibles combinaciones de estados de las variables que intervienen, utilizando la expresión $\mathbf{2^n}$, y las escribimos.
  Número de combinaciones = $\mathbf{2^n} = 2^2 = 4$ Las combinaciones son: (00); (01); (10) y (11)
- Establecemos las operaciones simples en que puede descomponerse la función.
  Para nuestra función $\mathbf{F= AB + \overline{A}B}$, se aprecia que existe:
  - Un **producto lógico** entre A y B: **AB**
  - La **complementación o negación** de la variable A: $\overline{\mathbf{A}}$
  - Un **producto lógico** entre $\overline{A}$ y B: $\overline{\mathbf{A}}\mathbf{B}$
  - Una **suma lógica de los productos** anteriores: $\mathbf{AB + \overline{A}B}$

- Así preparamos una tabla e incluimos en ella las cuatro combinaciones de estados posibles y todas las operaciones simples en las que se ha descompuesto la tabla.

A	B	AB	$\overline{A}$	$\overline{A}B$	F =AB + $\overline{A}$B
0	0				
0	1				
1	0				
1	1				

A	B	AB	$\overline{A}$	$\overline{A}B$	F =AB + $\overline{A}$B
0	0	0	1	0	**0**
0	1	0	1	1	**1**
1	0	0	0	0	**0**
1	1	1	0	0	**1**

Así pues, **existirá señal de salida** (1) en este circuito lógico cuando exista señal de entrada en **B** o en **A y B.**

5.- FORMAS CANÓNICAS DE UNA FUNCIÓN

Se llama **primera forma canónica de una función Booleana** a toda suma de productos en las cuales aparecen todas las variables, en forma directa o complementada, en cada uno de los términos que conforma la expresión, recibe el nombre de **forma canónica.**

5.1.- OBTENCIÓN DE LAS FORMAS CANÓNICAS DE UNA FUNCIÓN A PARTIR DE LA TABLA DE VERDAD

De la tabla de verdad de una función lógica es fácil deducir las formas canónicas de una función.

Los productos de la primera forma canónica se formarán para cada fila en que la función de salida toma el valor **1**, asignando al 0 la variable complementada y al 1 la variable directa y sumándolos todos ellos.

Ejemplo 1: La siguiente tabla de verdad muestra cuándo las combinaciones producen un **1** a la salida y por lo tanto la función se activa. Como la función consta de dos variables tendremos cuatro combinaciones posibles:

a	b	S
0	0	0
0	1	1
1	0	0
1	1	1

La forma canónica será la suma de productos donde la función de salida tenga el valor **1**, en nuestro caso tenemos en nuestra salida dos unos, por lo tanto nuestra función tendrá la suma de ***dos*** términos, por lo que quedaría como resultado:

$$\mathbf{S = \bar{a} \cdot b + a \cdot b}$$

El resultado sale de sumar las combinaciones donde S=1, cogiendo el producto de todas sus variables de forma que se coloca la variable normal si vale 1 y su complemento o negación si vale 0. Si se asigna a cada combinación binaria un número decimal, se puede escribir de forma abreviada: **S= Σ (1,3)**

6.- PUERTAS LÓGICAS

Comenzamos aquí el estudio de la electrónica digital. Nos centraremos a partir de ahora en componentes que sólo admitan dos estados: abierto o cerrado, apagado o encendido, 0 ó 1.

El desarrollo de la electrónica ha permitido sustituir los interruptores por componentes denominados puertas lógicas, que permiten realizar operaciones sencillas con ceros y unos. Se pueden aplicar a tecnología electrónica, eléctrica, mecánica, hidráulica y neumática. Son circuitos de conmutación integrados en un chip. Los elementos básicos con los que se construyen las puertas lógicas son componentes semiconductores como son el diodo y el transistor.

Una puerta lógica o compuerta lógica es un dispositivo electrónico capaz de realizar las operaciones lógicas básicas (suman, multiplican, niegan o afirman) del álgebra de Boole. Las puertas lógicas se encuentran disponibles en el mercado en **circuitos integrados**.

Según la función que represente las puertas lógicas serán distintas y su representación también. Las puertas lógicas más importantes se denominan: AND, OR, NOT, NAND, NOR, OR-EXCLUSIVA y NOR-EXCLUSIVA..

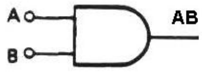
6.1.- TIPOS DE PUERTAS LÓGICAS

A continuación se irán explicando las distintas puertas lógicas acompañadas de su representación gráfica, su tabla de verdad y su función resultante forma canónica.

6.1.1.- *Puerta AND o función PRODUCTO LÓGICO:*

Realiza la función básica de intersección o producto lógico de dos variables de entrada llamadas A y B. Su símbolo algebraico es ***** o simplemente una variable junto a la otra.

La expresión lógica de la función de salida F es: **F = A * B = AB**

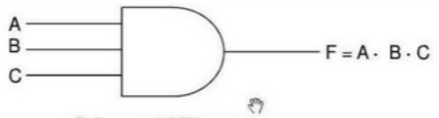


El símbolo lógico para la función **AND** o **PRODUCTO LÓGICO** es:

La tabla de verdad es la siguiente: La función **AND** tendrá el valor **1** a su salida, **si y sólo si** todas y cada una de las variables de entrada toman el valor 1.

A	B	**F= A * B**
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Las puertas lógicas pueden tener más de dos variables a la entrada:



3-Input AND gate

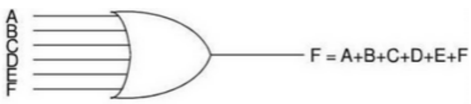
6.1.2.- Puerta OR o función SUMA LÓGICA:

Realiza la función básica de unión o suma lógica. Se representa mediante el signo "+" al igual que la suma aritmética. **F = A + B**

El símbolo lógico para la función **OR** o **SUMA LÓGICA** es:

La tabla de verdad es la siguiente: La función **OR** tendrá el valor **1** a su salida, cuando **al menos una** de las variables de entrada valga 1. Esto es válido para cualquier número de variables de entrada.

A	B	**F= A +B**
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**



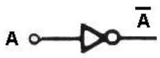
6-Input OR gate

6.1.3.- Puerta NOT o INVERSIÓN

Realiza la función básica de complementación, inversión o negación. La función NOT da como resultado el inverso del estado de la variable de entrada. Si A vale 1, $\overline{A}$ vale 0 y viceversa. Se representa por medio de una rayita colocada sobre el valor de la variable.

Es la *única* puerta lógica que tiene una sola entrada, el resto de puertas tendrá dos o más entradas.

El símbolo de la función **NOT** es:



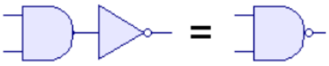
La tabla de verdad es la siguiente:

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

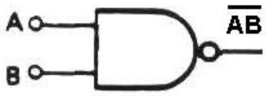
Partiendo de estas puertas (AND, OR y NOT) se obtienen otras funciones lógicas de gran importancia.

6.1.4.- Puerta NAND

Es la función complementaria o negación de la función AND. Es una puerta AND seguida de un inversor. Su símbolo algebraico se obtiene añadiendo una rayita horizontal en la parte superior de la expresión de la función AND. Su ecuación lógica será, por tanto:



$$\mathbf{F = \overline{A * B}}$$



Símbolo de la función **NAND**. El círculo indica la negación.

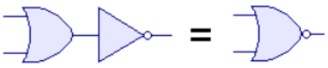
La tabla de verdad es la siguiente:

A	B	**F= $\overline{A * B}$**
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

6.1.5.- Puerta NOR

Es la función complementaria o negación de la función OR. Es una puerta OR seguida de un inversor. Su símbolo algebraico se obtiene añadiendo una rayita horizontal en la parte superior de la expresión de la función OR.

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



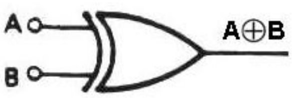
Símbolo de la función **NOR**. El círculo indica la negación.

La tabla de verdad es la siguiente:

A	B	**F= $\overline{A+B}$**
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**

6.1.6.- Puerta ORex, OR-exclusiva o X-OR

La función OR exclusiva de dos variables es aquella que toma el valor 1 cuando una de las variables toma el valor 1 y la otra el valor 0 ó viceversa. Su símbolo algebraico es ⊕.



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

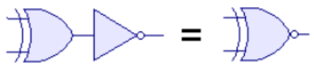
A	**B**	**$\overline{A}$**	**$\overline{B}$**	**$A\overline{B}$**	**$\overline{A}B$**	**F = A ⊕ $\overline{B}$**
0	0	1	1	0	0	**0**
0	1	1	0	0	1	**1**
1	0	0	1	1	0	**1**
1	1	0	0	0	0	**0**

Su ecuación lógica se deriva de una combinación de funciones AND, OR y NOT, aunque se considera una función elemental. Su tabla de verdad puede deducirse combinando adecuadamente las funcionen elementales que la forman, es decir:

La función XOR vale **1** cuando el número de entradas de valor 1 es impar.

6.1.7.- Puerta NOR-exclusiva o X-NOR

La puerta **XNOR** es una puerta lógica digital cuya función es la inversa de la puerta OR exclusiva (XOR). La puerta X-NOR ofrece un “1” lógico a la salida cuando las entradas son iguales. En el caso de puertas de dos entradas, ésta se define como:

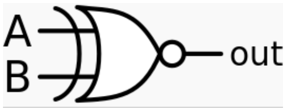


$F = \overline{A \oplus B}$

INPUT		OUTPUT
A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

La tabla de verdad indica las siguientes combinaciones:

El símbolo lógico utilizado para este tipo de puertas es:



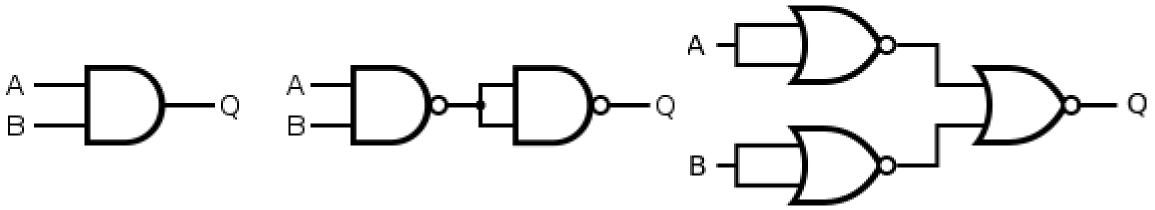
6.1.8.- Puertas que sirven para todo:

Las puertas **NAND** y **NOR** se conocen como **“PUERTAS UNIVERSALES”** lo que significa que utilizando exclusivamente cualquiera de ellas como base, se pueden implementar el resto de puertas AND, OR, NOT, XOR etc, es decir mediante ellas es posible obtener cualquiera de las otras puertas.

Los siguientes circuitos muestran cómo conseguir puertas AND, NOT y OR a partir de puertas NAND o puertas NOR:

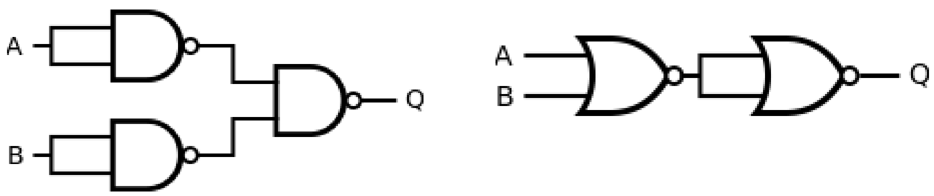
En caso de no estar disponibles puertas AND específicas, estas pueden ser implementadas usando puertas NAND o NOR.

Puerta deseada **Construcción NAND** **Construcción NOR**

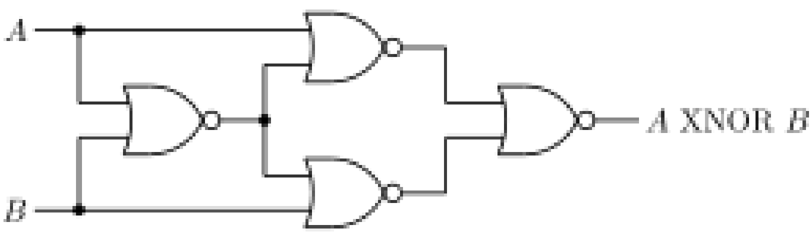


En caso de no estar disponibles puertas OR específicas, se puede hacer de NAND o NOR en la configuración que se muestra en la imagen.

Construcción de NAND **Construcción de NOR**

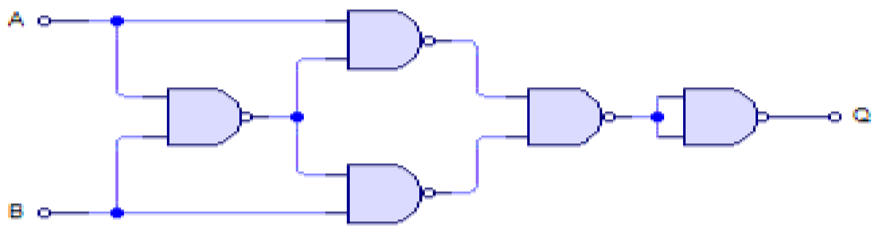


En caso de no estar disponibles puertas XNOR específicas, se puede hacer de cuatro puertas NOR o cinco puertas NAND en las configuraciones que se muestran a continuación.



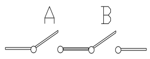
Puerta XNOR construida utilizando sólo puertas NOR

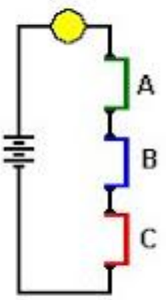
Puerta XNOR construida utilizando solo puertas NAND



5.1.9.- Aplicaciones de las puertas lógicas:

Una puerta AND puede utilizarse, por ejemplo, para controlar el motor de una taladradora de columna; dicho motor sólo funcionaría cuando estuvieran cerrados dos interruptores: el interruptor del motor propiamente dicho y otro interruptor que podría activarse o desactivarse mediante un pedal de emergencia.

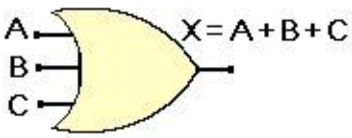




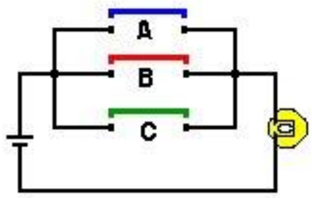
A	B	C	Lámpara
A	A	A	Apagada
A	A	C	Apagada
A	C	A	Apagada
A	C	C	Apagada
C	A	A	Apagada
C	A	C	Apagada
C	C	A	Apagada
C	C	C	Encendida

En este circuito se puede apreciar una compuerta AND aplicada en un circuito. Se puede ver que no habrá paso de corriente hasta que todas las entradas o interruptores estén cerrados, es decir tenga señal alta.

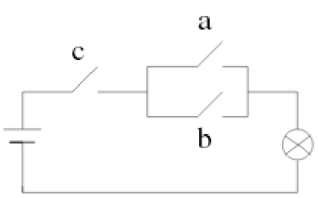
- Una puerta OR puede utilizarse, por ejemplo, para activar un dispositivo desde dos o tres lugares diferentes. De esta manera, conectando adecuadamente un timbre y una puerta OR podríamos conseguir que la alarma de una casa con dos o tres puertas se activara cuando estuviera abierta cualquiera de ellas.



A	B	C	X
0	0	0	0
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	1



En la figura anterior podemos apreciar el comportamiento de una puerta OR de 3 entradas y su única salida, junto con su tabla de verdad y un ejemplo de su aplicación real en un circuito.



$L = f(a,b,c) = c\,(a+b)$

a	b	c	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Ejemplo de función booleana, tabla de verdad y circuito con interruptores.

6.- Álgebra de Boole. Simplificación de funciones lógicas.

Cuando se plantea el problema de realizar físicamente un sistema digital, que implemente una función lógica a partir de las puertas que hemos visto, el resultado debe ser lo más simple posible, es decir, que necesite el menor número de puertas posibles. Las funciones lógicas pueden simplificarse utilizando las propiedades, leyes y teoremas del álgebra de Boole.

<u>Propiedades del Álgebra de Boole</u>:

Para toda variable a, b, c que pertenecen al conjunto de álgebra de Boole se cumple:

1) Propiedad conmutativa de la suma y del producto: $a+b = b+a$ $a \cdot b = b \cdot a$

2) Propiedad asociativa de la suma y del producto: $a+b+c = a+(b+c)$ $a \cdot b \cdot c = a \cdot (b \cdot c)$

3) Propiedad distributiva de la suma y del producto entre sí:
$a \cdot (b+c) = a \cdot b + a \cdot c$ $a+(b \cdot c) = (a+b) \cdot (a+c)$

Los paréntesis indican el orden en el que se han de realizar las operaciones.

4) Elementos neutros: son el "0" para la suma y el "1" para el producto. $\boxed{a + 0 = a}$ $\boxed{a \cdot 1 = a}$

5) Del elemento complementario u opuesto: $\boxed{a + \bar{a} = 1}$ $\boxed{a \cdot \bar{a} = 0}$

6) Teorema del doble complemento. $(\bar{\bar{a}}) = a$

7) Elementos absorbentes: son el "1" para la suma y el "0" para el producto: $\boxed{a + 1 = 1}$ $\boxed{a \cdot 0 = 0}$

8) Idempotente: $\boxed{a + a = a}$ $\boxed{a \cdot a = a}$

9) Simplificativa: $a + a \cdot b = a$ $a \cdot (a+b) = a$

10) Leyes o Teoremas de De Morgan:

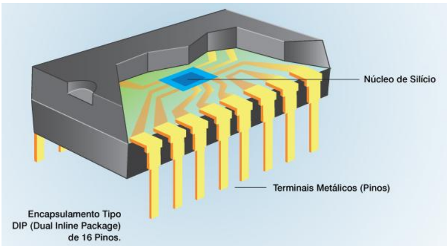
$$\overline{a+b} = \bar{a} \cdot \bar{b}$$
$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

$$\overline{a + b + c} = \bar{a} \cdot \bar{b} \cdot \bar{c}$$
$$\overline{a \cdot b \cdot c} = \bar{a} + \bar{b} + \bar{c}$$

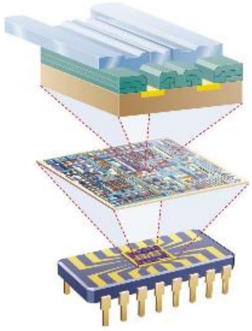
Todas estas propiedades, leyes y teoremas se pueden comprobar sin más que realizando las tablas de verdad de cada lado de las igualdades y observar que el resultado es el mismo.

7.- Circuitos integrados:

Históricamente las primeras puertas lógicas se hicieron con relés. Después con válvulas de vacío (ya en desuso) y finalmente, con transistores. Las puertas lógicas no se comercializan individualmente, sino que se presentan empaquetados en un circuito integrado.



Los circuitos integrados (I.C. **I**ntegrated **C**ircuits) son circuitos que están formados por componentes electrónicos (transistores, diodos, resistencias, condensadores,…), integrados en una **oblea de silicio** de reducidas dimensiones (miniaturizados), estos chips de silicio están protegidos por una funda o carcasa de plástico con unas patillas para realizar las conexiones. También se les llama chip o microchip. El número aproximado de puertas lógicas que contiene un circuito integrado es de 4.



En un chip, los elementos del circuito son tan pequeños que se necesita un buen microscopio para verlo. En un microchip de un par de centímetros de largo por un par de centímetros de ancho pueden caber millones de transistores además de resistencias, condensadores, diodos, etc. Un ejemplo muy bueno sería el microprocesador de un ordenador. El Pentium IV de Intel, sacado en el mercado en el 2001, integraba unos 42 millones de transistores.

Los IC se pueden implementar con diferentes técnicas o tecnologías, según sean los métodos de fabricación de los componentes. Existen una gran variedad de **tecnologías de fabricación** de circuitos integrados. Las tecnologías más conocidas y usadas son las **TTL** (**T**ransistor-**T**ransistor **L**ogic) y **CMOS** (**C**omplementary **M**etal **O**xide **S**emiconductor), aunque existen otras. Las características que buscan los fabricantes en todas las familias lógicas fundamentales son fundamentalmente conseguir un gran número de componentes por mm^2, el mínimo consumo, un bajo coste y velocidades de trabajo muy rápidas.

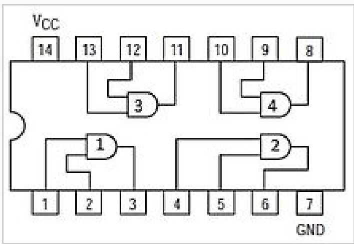
Familia TTL: Los circuitos que forman las puertas lógicas están constituidos por resistencias, diodos y transistores bipolares. Esta familia es una de las de mayor aplicación y más comercializada para la realización de circuitos lógicos por su facilidad de manipulación y bajo precio. Esta familia dispone de una gran variedad de circuitos integrados, la más conocida es la serie **74.** Vamos a ver a continuación los esquemas de algunos de los circuitos integrados TTL de la serie **74xx** más comunes, formados por puertas lógicas. El circuito integrado básico de la familia TTL es el 7400, circuito que contiene puertas lógicas NAND.

CIRCUITO INTEGRADO	FUNCIÓN
7400	4 NAND
7402	4 NOR
7404	4 NOT
7408	4 AND
7410	4 NAND
7432	4 OR
7486	4 ORex

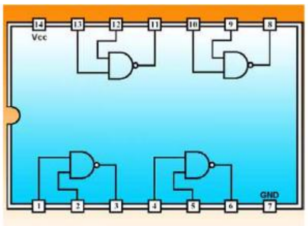
La tensión de **alimentación es de 5 voltios**. En tecnología TTL se trabaja generalmente en lógica positiva. Al estado 1 se le asigna 5 voltios y al estado 0 se le asignan 0 voltios. El número de terminales o pins es de 14, estas patillas constituirán las entradas, las salidas y la alimentación.

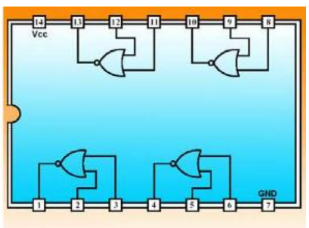
<u>DESCRIPCIÓN DE LOS TERMINALESDEL CI 7408</u>

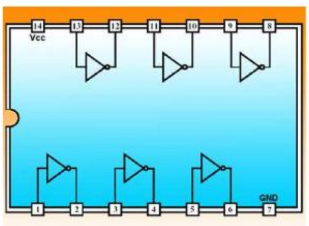
- **Pin 1:** La entrada A de la compuerta 1.
- **Pin 2:** La entrada B de la compuerta 1.
- **Pin 3:** Aquí veremos el resultado de la operación de la primer compuerta.
- **Pin 4:** La entrada A de la compuerta 2.
- **Pin 5:** La entrada B de la compuerta 2.
- **Pin 6:** Aquí veremos el resultado de la operación de la segunda compuerta.
- **Pin 7 Normalmente GND:** Es el polo negativo de la alimentación, generalmente tierra.
- **Pin 8:** Aquí veremos el resultado de la operación de la cuarta compuerta.
- **Pin 9:** La entrada B de la compuerta 4.
- **Pin 10:** La entrada A de la compuerta 4.
- **Pin 11:** Aquí veremos el resultado de la operación de la tercer compuerta.
- **Pin 12:** La entrada B de la compuerta 3.
- **Pin 13:** La entrada A de la compuerta 3.
- **Pin 14 Normalmente VCC:** Alimentación, es el pin donde se conecta el voltaje de alimentación de 5 ± 0.25 voltios

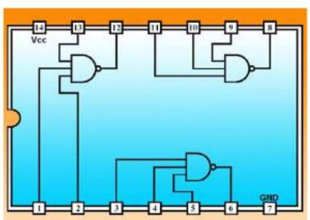


Configuración 7408

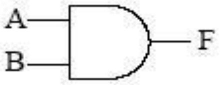
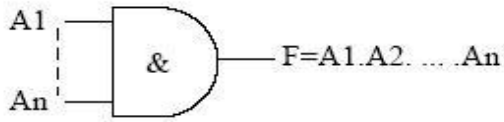
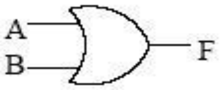
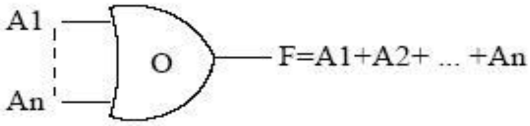
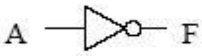
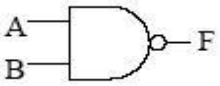
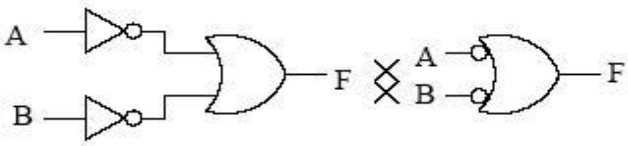
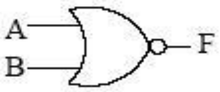
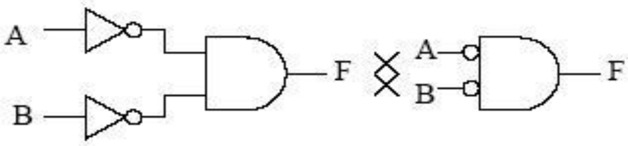
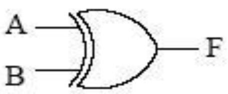
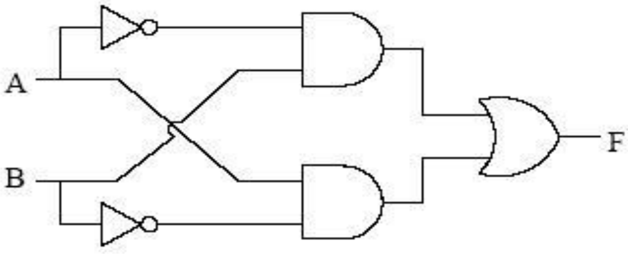
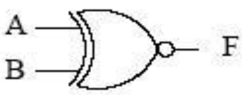
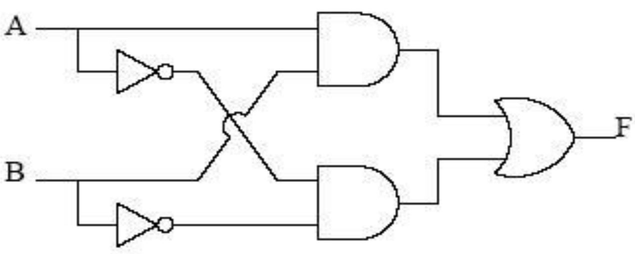

Chip TTL, SN7400


Chip TTL, SN7402


Chip TTL, SN7404


Chip TTL, SN7410

RESUMEN: EQUIVALENCIA DE LAS PUERTAS LÓGICAS:

<table>
<tr><th></th><th>SIMBOLO</th><th>FUNCION</th><th>TABLA</th><th>DATOS DE INTERES</th></tr>
<tr><td>AND</td><td></td><td>$F = A.B$</td><td>A B F
0 0 0
0 1 0
1 0 0
1 1 1</td><td></td></tr>
<tr><td>OR</td><td></td><td>$F = B+A$</td><td>A B F
0 0 0
0 1 1
1 0 1
1 1 1</td><td></td></tr>
<tr><td>NOT</td><td></td><td>$F = \overline{A}$</td><td>A F
0 1
1 0</td><td></td></tr>
<tr><td>NAND</td><td></td><td>$F = \overline{A.B} = \overline{A} + \overline{B}$</td><td>A B F
0 0 1
0 1 1
1 0 1
1 1 0</td><td></td></tr>
<tr><td>NOR</td><td></td><td>$F = \overline{A+B} = \overline{A}.\overline{B}$</td><td>A B F
0 0 1
0 1 0
1 0 0
1 1 0</td><td></td></tr>
<tr><td>XOR</td><td>OR EXCLUSIVA</td><td>$F = A \oplus B =$
$\overline{A}.B + A.\overline{B}$</td><td>A B F
0 0 0
0 1 1
1 0 1
1 1 0</td><td></td></tr>
<tr><td>NXOR</td><td>NOR EXCLUSIVA</td><td>$F = \overline{A \oplus B} =$
$= \overline{\overline{A}B+A\overline{B}} =$
$=(A+\overline{B})(\overline{A}+B)=$
$= A.B + \overline{B}.\overline{A}$</td><td>A B F
0 0 1
0 1 0
1 0 0
1 1 1</td><td></td></tr>
</table>