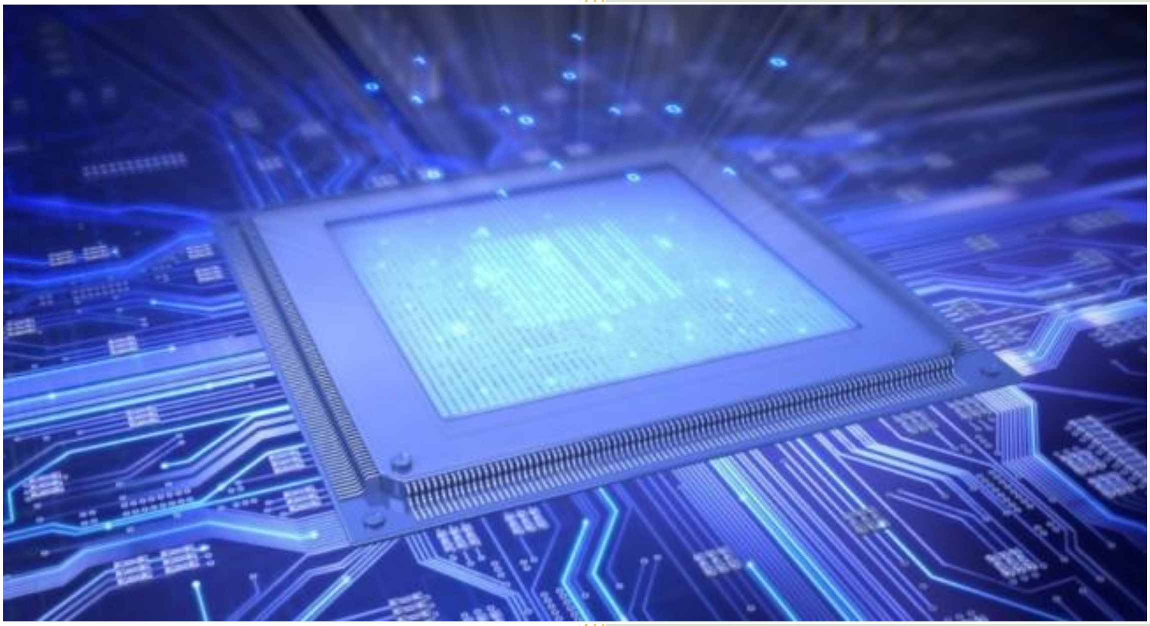
Tema 5. Electrónica digital



Víctor M. Acosta Guerrero

Profesor de Tecnología

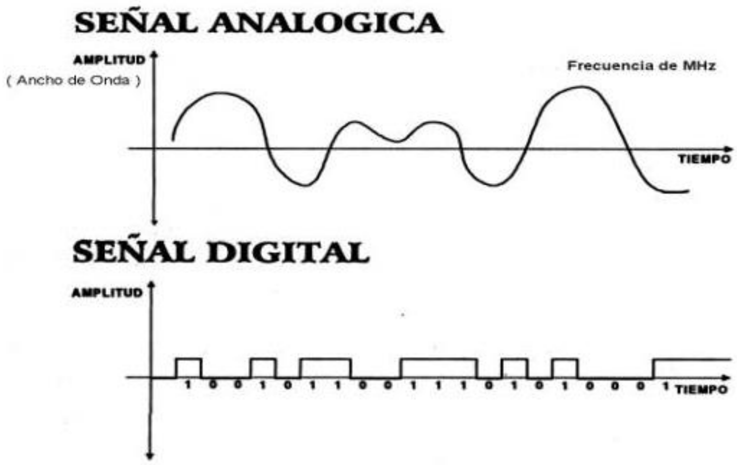
I.E.S.O. Matías Ramón Martínez

Tema 5. Electrónica digital.

1. INTRODUCCIÓN.

Antes de comenzar el tema es importante que sepamos distinguir entre señales analógicas y señales digitales.

- Las **señales analógicas** son continuas y toman un número infinito de valores.
- Las **señales digitales** son discretas, es decir, pasan de un valor a otro sin tomar valores intermedios. En el caso que nos ocupa, que es el de la electrónica digital, sólo existen dos estados: 0 y 1. Estos pueden ser impulsos eléctricos de baja tensión y una tensión más alta, interruptores abiertos o cerrados, etc…



Un ejemplo de señal analógica es la lectura de un termómetro de mercurio. Si nuestra vista fuera lo suficientemente precisa, podríamos distinguir infinitas lecturas entre dos puntos. La analogía digital la encontramos en un termómetro digital, ya que entre una centésima y la siguiente no detecta valores intermedios.

2. CÓDIGO BINARIO, DECIMAL Y HEXADECIMAL.

Un sistema electrónico maneja información en código binario, es decir ceros y unos: el cero quiere decir que no pasa corriente y el uno que sí pasa. Sin embargo, el sistema de numeración empleado por nosotros es el **sistema decimal**.

Éste se compone de diez dígitos (del 0 al 9) a los que otorga un valor dependiendo de la posición que ocupen en el número: unidades, decenas, centenas, millares, etc… El valor de cada dígito está asociado al de una potencia de base 10, y un exponente igual al valor que ocupa el dígito menos uno, contando desde la derecha. De esta forma, el número 634 sería:

$$634 = 6 \text{ centenas} + 3 \text{ decenas} + 4 \text{ unidades}$$
$$634 = 6\cdot10^2 + 3\cdot10^1 + 4\cdot10^0 = 600 + 30 + 4 = 634$$

En el caso de los números decimales, su resolución es la misma, con la diferencia de que existen algunas potencias de exponentes negativos, que se corresponderán con los dígitos colocados a la derecha del separador decimal. A modo de ejemplo, el número 9643,51 se calcularía como:

$$9643{,}51=9\cdot10^3+6\cdot10^2+4\cdot10^1+3\cdot10^0+5\cdot10^{-1}+1\cdot10^{-2} = 9000+600+40+3+0{,}5+0{,}01=9643{,}51$$

El **sistema de numeración binario** emplea sólo dos cifras: el 0 y el 1. En una cifra binaria, cada dígito tiene distinto valor dependiendo de la posición que ocupe. El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual a la posición que ocupe la cifra menos uno.

Como podemos apreciar, la base de la potencia utilizada es igual al número de dígitos empleados en la numeración (2), tal y como ocurría con las potencias en la numeración decimal (10).

De acuerdo con lo expuesto, el número binario 1011, tendría un valor:

$$1011 = 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 8 + 0 + 2 + 1 = 11$$

Para expresar que ambas cifras describen la misma cantidad, lo escribimos así:

$$1011_2 = 11_{10}$$

A los dígitos en el sistema de numeración binario se les llama **bit** (**B**inary dig**it**). Un bit es la unidad mínima de información empleada en informática. A un grupo de 8 bits se le conoce por el nombre de **byte**. Un **kilobyte** (kB) son 1024 bytes (2^{10}). Un **gigabyte** (GB) son 1024 kB, y un **terabyte** (TB) son 1024 GB.

2.1. Conversión de decimal a binario.

Convertir un número decimal al sistema binario es muy sencillo: basta con realizar divisiones repetidas por dos y escribir los restos obtenidos en cada división en orden inverso al que han sido obtenidos. Por ejemplo, para convertir al sistema binario el número 77_{10}, las divisiones arrojarán los siguientes restos:

- 77 / 2 = 38. Resto 1.
- 38 / 2 = 19. Resto 0.
- 19 / 2 = 9. Resto 1.
- 9 / 2 = 4. Resto 1.
- 4 / 2 = 2. Resto 0.
- 2 / 2 = 1. Resto 0.
- 1/ 2 = 0. Resto 1.

Por tanto:

$$77_{10} = 1001101_{2}$$

Es importante aclarar que para representar números en binario, es necesario utilizar más dígitos que en decimal. Así, en el ejemplo anterior hemos visto que para representar el número 77, que en el sistema decimal está compuesto por sólo dos dígitos, hemos necesitado siete dígitos en el sistema binario.

Para representar números más grandes del sistema decimal, necesitaremos aún más dígitos. Como regla general, con un número “n” de dígitos binarios o bits, pueden representarse 2^n números del sistema decimal. Por ejemplo, con 4 bits, se podrán representar un máximo de 2^4=16 números decimales, siendo el número decimal mayor que se puede representar el 15.

2.2. Conversión de binario a decimal.

Para convertir un número binario en uno decimal, habría que desarrollar el número teniendo en cuenta el valor de cada dígito en su posición, que es el de una potencia de base 2, cuyo exponente es el 0 en el bit más situado a la derecha, y se incrementa una unidad según vamos avanzando posiciones hacia la izquierda. A modo de ejemplo vamos a desarrollar el siguiente número: 10100112.

$$1010011_2 = 1 \cdot 2^6 + \ 0 \cdot^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 83 \rightarrow 1010011_2 = 83_{10}$$

2.3. El sistema hexadecimal.

En el sistema hexadecimal los números se representan con dieciséis símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. Se emplean las letras porque en el sistema decimal no existen dígitos mayores de 9.

Por tanto representarían a los números que van desde el 10 hasta el 15. El valor de cada uno de estos símbolos depende de su posición, que se calcula mediante potencias de base 16. Así, por ejemplo, el número $1A3F_{16}$ sería:

$$1A3F_{16} = 1 \cdot 16^3 + 10 \cdot 16^2 + 3 \cdot 16^1 + 15 \cdot 16^0 = 6716_{10}$$

Para realizar la operación contraria, tendremos que realizar divisiones sucesivas entre 16, e irnos quedando con los restos.

Hay que tener en cuenta que si uno o varios de los restos están entre 9 y 15, habrá que sustituirlos por la letra que les corresponde, como en el ejemplo siguiente, en el que queremos convertir 1735_{10} a hexadecimal:

- 1735 / 16 = 108. Resto 7.
- 108 / 16 = 6. Resto 12 → Resto C.
- 6 / 16 = 0. Resto 6.

Ordenando los restos en orden inverso, tenemos que:

$$1735_{10} = 6C7_{16}$$

La razón para el uso del sistema hexadecimal es que su conversión a binario o la conversión de binario a hexadecimal son muy simples, puesto que, al ser dieciséis igual a dos elevado a cuatro, cuatro números binarios componen un número hexadecimal.

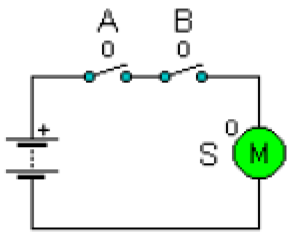
No obstante, no es objeto del presente curso el estudio de este tipo de conversiones.

3. LA TABLA DE LA VERDAD.

El objetivo de cualquier sistema electrónico, es que se produzca un determinado resultado (al que llamamos salida), siempre y cuando se cumplan unas determinadas condiciones (a las que llamamos entradas).

Vamos a analizar el siguiente ejemplo, que puede ser un sistema de seguridad electrónico. Una determinada máquina sólo debe funcionar si están accionados dos pulsadores normalmente abiertos al mismo tiempo.

De este modo el operario tendrá las manos alejadas de la zona peligrosa cuando la máquina se accione. El circuito será el siguiente:



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

La lectura de la tabla es la siguiente:

- Entradas: A y B.
- Salida: S.
- Pulsador cerrado: 1.
- Pulsador abierto: 0.
- Motor en marcha: 1.
- Motor parado: 0.

A la tabla que muestra la relación que existe entre las entradas y las salidas se le llama **tabla de la verdad**.

Por tanto, en la tabla de la verdad podemos apreciar que el motor sólamente se accionará en caso de que los dos pulsadores estén cerrados a la vez.

4. LAS FUNCIONES LÓGICAS.

En electrónica digital el objetivo es que nuestro sistema se comporte según la tabla de la verdad. Para conseguirlo, tenemos que reducir la tabla de la verdad a una única expresión que se llama **función lógica**.

Aunque las funciones lógicas pueden ser muy complejas, todas ellas son una combinación de las tres operaciones lógicas básicas:

Suma: Interruptores en paralelo; S = A + B + C

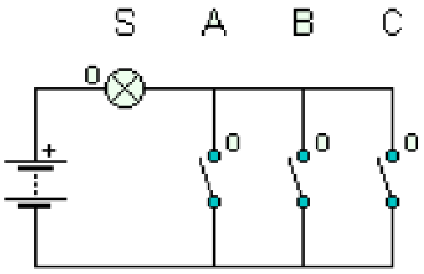
Producto: Interruptores en serie; S = A · B · C

Negación: Pulsador normalmente cerrado; S = A'

A estas operaciones lógicas básicas y a las que derivan de ellas se le conoce con el nombre de Álgebra de Boole. A continuación vamos a analizar detenidamente dichas operaciones.

4.1. La suma lógica.

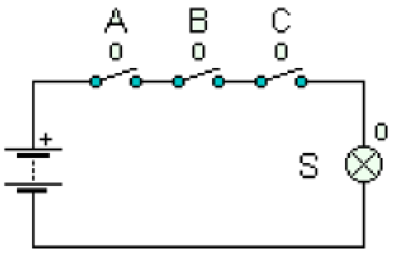
La salida se activa (es un 1), cuando cualquiera de las condiciones de entrada están activadas. Solamente no se activaría la salida si todas las entradas se encuentran desactivadas.



A	B	C	S
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

4.2. El producto lógico.

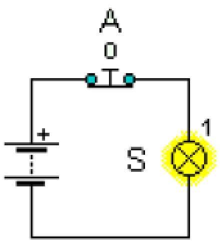
La salida se activa sólo cuando todas las condiciones de entrada están activadas. En el circuito del ejemplo, la salida sólo se activa cuando los tres interruptores están cerrados.



A	B	C	S
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

4.3. Negación o inversión lógica.

Al actuar sobre la entrada (A=1), la salida se detiene (S=0) y viceversa. En el circuito del ejemplo, al actuar sobre el pulsador normalmente cerrado, la luz se apaga, y si no actuamos, seguirá encendida (S=A’).



A	S
0	**1**
1	**0**

La inversión se puede representar mediante una barra encima de la función, o bien mediante un apóstrofe.

5. LAS FUNCIONES LÓGICAS A PARTIR DE LA TABLA DE LA VERDAD.

Partiendo de un sistema del que sólo conocemos la tabla de la verdad, si queremos obtener la función lógica tenemos que seguir los siguientes pasos:

- Localizar los valores 1 en la salida.

- Leer el valor de las variables de entrada para los casos en los que las salidas están activas (cuando valen 1).
- Asignar, por ejemplo para la variable A, A cuando vale 1, y A' cuando vale 0.
- Multiplicar los valores obtenidos en cada fila.
- Sumar todos los resultados.

Ejemplo: Dibujar la tabla de la verdad de un motor que se activa sólo cuando tenemos dos pulsadores normalmente abiertos activados al mismo tiempo. Obtener la función lógica a partir de ella.

A	**B**	**S**
Interruptor de marcha	**Interruptor de seguridad**	**Salida del motor**
0	0	0
1	0	0
0	1	0
1	1	1

Nos fijamos en la fila en la que S=1. Esta situación sólo se da cuando A=1 y B=1. Se trata por tanto de un producto lógico. La función lógica es por tanto:

$$S = AB$$

Ejemplo: Obtener la función lógica de la siguiente tabla de la verdad:

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

En este caso tenemos dos filas en las que S=1. En la primera, al ser 0 las dos entradas, equivaldría al producto de las dos negaciones (A'B'), mientras que en la segunda tendríamos AB'. Por tanto, la función lógica será:

$$S = A'B' + AB'$$

Ejemplo: Obtener la función lógica de la siguiente tabla de la verdad.

A	B	C	S
0	0	0	0
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	0

Teniendo en cuenta lo expuesto anteriormente:

$$S = A'B'C + A'BC' + AB'C$$

6. TABLA DE LA VERDAD A PATIR DE LA FUNCIÓN LÓGICA.

Es la operación contraria a la anterior, ya que en este caso sólo se conoce la función lógica y debemos determinar la tabla de la verdad. Para ello debemos seguir una serie de pasos:

- Construir una tabla con el número de variables que tiene la función, además de la variable de salida. Tendrá 2^n situaciones (n= número de variables de entrada).
- Introducir los valores de las entradas según el orden lógico.
- Interpretar en cada sumando, cuáles son los casos en los que la función vale 1.
- Completar con ceros.

Lo mejor es verlo con un ejemplo:

Ejemplo: Determinar la tabla de la verdad de la siguiente función lógica:

$$S = A' + BC + AB'C$$

Lo primero que debemos hacer es construir la tabla de la verdad con las tres variables de entrada (A, B y C). Para introducir las entradas de una forma lógica, en la última columna (última entrada) colocamos 0 y 1 alternativamente. En la anterior a ésta, 0 en las dos primeras filas y 1 en las otras dos, y así sucesivamente. En la anterior 0 en las cuatro primeras, y 1 en las otras 4. De esta forma quedan recogidos todos los posibles resultados.

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

Ahora tendremos que poner uno en los casos en los que S = A' + BC + AB'C, analizando los distintos sumandos:

- A'. Todos los casos en los que A está negado (es 0), es decir: 000, 001, 010 y 011.
- BC. Todos los casos en los que B y C sean 1 (011 y 111).
- AB'C. Cuando A=1, B=0 y C=1, o sea 101.
- En los demás casos se pone 0.

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

7. SIMPLIFICACIÓN DE FUNCIONES LÓGICAS. ÁLGEBRA DE BOOLE.

Existen casos en los que la función lógica que nos podemos encontrar, es bastante larga y compleja, por lo que es recomendable simplificarla. Para realizar esta simplificación debemos conocer una serie de reglas básicas, que son las Propiedades del Álgebra de Boole. Dichas propiedades son las siguientes:

Propiedad conmutativa	Inversión
$a + b = b + a$ $a \cdot b = b \cdot a$	$a + a' = 1$ $a \cdot a' = 0$
Propiedad asociativa	**Idempotencia**
$a + (b + c) = (a + b) + c = a + b + c$ $a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$	$a + a = a$ $a \cdot a = a$
Propiedad distributiva	**Absorción**
$a (b + c) = a \cdot b + a \cdot c$ $a + b \cdot c = (a + b) \cdot (a + c)$	$a + a \cdot b = a$ $a (a + b) = a$
Leyes de Morgan	**Involución**
$(a + b)' = a' \cdot b'$ $(a \cdot b)' = a' + b'$	$a'' = a$
Otras propiedades	
$a + 1 = 1$ $a \cdot 0 = 0$	

Las propiedades asociativa, distributiva y conmutativa son bastante intuitivas, puesto que existen igualmente en la suma de números naturales a la que estamos acostumbrados. Lo mismo ocurre con la propiedad $a \cdot 0 = 0$

Teniendo en cuenta las anteriores propiedades, para simplificar funciones lógicas es necesario seguir el siguiente procedimiento:

- Buscar factores comunes
- Aplicar la propiedad distributiva
- Eliminar términos aplicando a + a' = 1 · a + 1 = 1

Ejemplo: Simplificar la siguiente función lógica:

$$s = abc + a'b + ab'c + a'bc'$$

Si reordeno la función aplicando la propiedad conmutativa, la función nos quedaría:

$$s = abc + ab'c + a'b + a'bc'$$

Para buscar factores comunes, vemos que en los dos primeros términos se repite ab y en los dos últimos a'b. De esta forma puedo aplicar la propiedad distributiva:

$$s = ac\,(b + b') + a'b\,(1 + c')$$

Teniendo en cuenta las propiedades que hemos visto, b + b' = 1 y 1 + c' = 1. Entonces:

$$s = ac \cdot 1 + a'b \cdot 1 \rightarrow \mathbf{s = ac + a'b}$$

Si dibujáramos su tabla de la verdad:

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

Ejemplo: Simplificar la siguiente función lógica:

$$s = abc'd + a'bc'd + ab'c'd' + ab'$$

Buscando factores comunes tenemos que:

$$s = bc'd\ (a + a') + ab'\ (c'd' + 1)$$

Como $a + a' = 1$ y $c'd' + 1 = 1$:

$$\mathbf{s = bc'd + ab'}$$

a	**b**	**c**	**d**	**s**
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	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

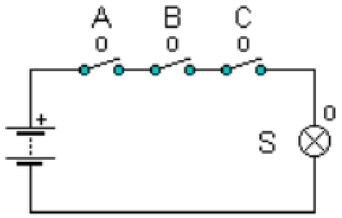
8. PUERTAS LÓGICAS.

Las puertas lógicas son circuitos electrónicos sencillos capaces de realizar las operaciones lógicas estudiadas anteriormente. Dichos elementos electrónicos son circuitos integrados que en apariencia no se diferencian unos de otros, salvo por el código que llevan escrito. A continuación vamos a estudiar dichos elementos:

8.1. Puerta AND (Producto lógico).

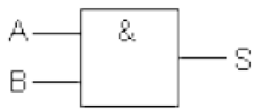
Equivale al producto lógico, por lo que la salida sólo se activa cuando todas las entradas se encuentran activadas:

Circuito y tabla de la verdad

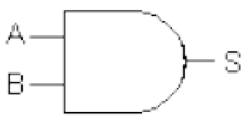


A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Símbolo DIN



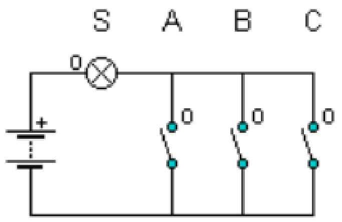
Símbolo ASA



8.2. Puerta OR (Suma lógica).

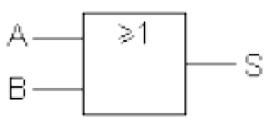
Equivale a la suma lógica, (la salida se activa si hay una entrada activada):

Circuito y tabla de la verdad

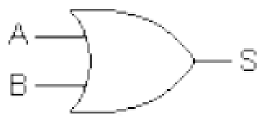


A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Símbolo DIN



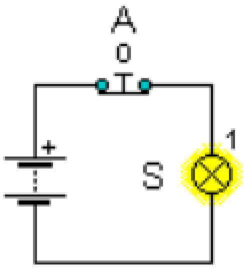
Símbolo ASA



8.3. Puerta NOT (Inversión o negación lógica).

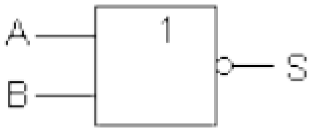
Corresponde a la inversión o negación lógica (la salida es la inversa de la entrada).

Circuito y tabla de la verdad

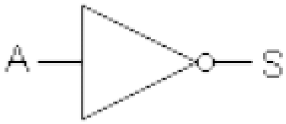


A	S
0	1
1	0

Símbolo DIN



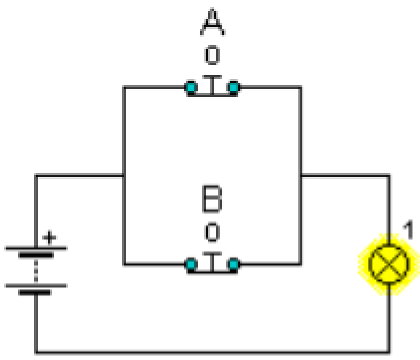
Símbolo ASA



8.4. Puerta NAND (Inversión del producto lógico).

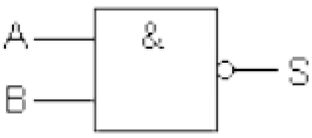
La señal de salida se activa siempre que no se activen todas las entradas. Equivale a combinar una puerta AND y una NOT. Es por tanto el inverso del producto lógico S = (AB)'

Circuito y tabla de la verdad

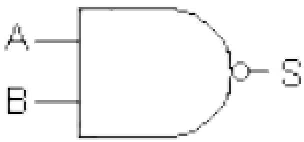


A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Símbolo DIN



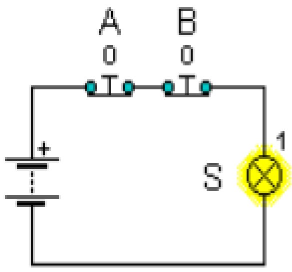
Símbolo ASA



8.5. Puerta NOR (Inversión de la suma lógica).

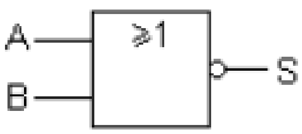
La señal de salida se activa siempre que todas las entradas estén inactivas. Equivale a combinar una puerta OR y una NOT. Es por tanto el inverso de la suma lógica S = (A+B)’

Circuito y tabla de la verdad

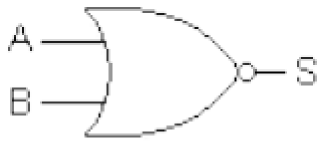


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

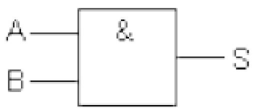
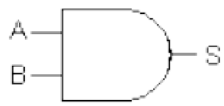
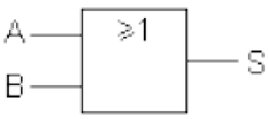
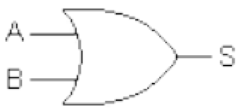
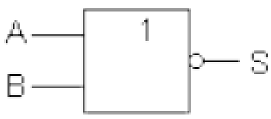
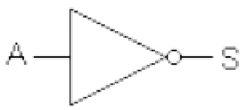
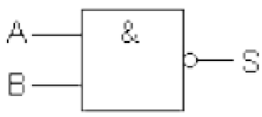
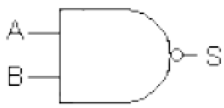
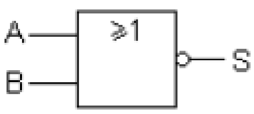
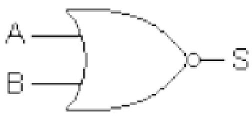
Símbolo DIN



Símbolo ASA



8.6. Resumen de puertas lógicas.

Puerta	Símbolo DIN	Símbolo ASA	Función
AND			Producto lógico S = A·B
OR			Suma lógica S = A+B
NOT			Inversión o negación S = A’
NAND			Inversión del producto S = (A·B)’
NOR			Inversión de la suma S = (A+B)’

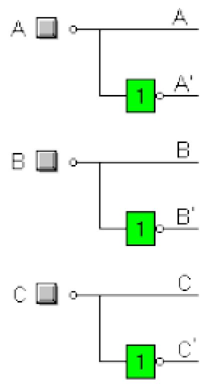
9. IMPLEMENTACIÓN DE UNA FUNCIÓN LÓGICA CON PUERTAS BÁSICAS.

Una vez obtenida y simplificada la función que relaciona la salida con las entradas en un sistema electrónico, dicha función puede implementarse, es decir, llevarse a la práctica, mediante un circuito de puertas lógicas básicas. La simplificación de la función es importante porque nos ahorra el uso de puertas lógicas y el circuito tendrá un funcionamiento mucho más eficiente. Los pasos a seguir para ello son:

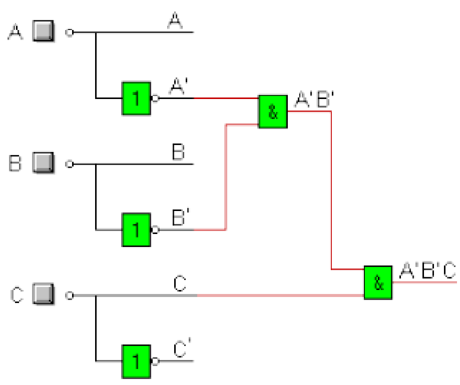
- Dibuja las entradas y añade puertas NOT para negar las variables necesarias.
- Realiza las multiplicaciones mediante puertas AND.
- Realiza las sumas mediante puertas OR.

Ejemplo. Obtener el circuito de la función S = A’B’C + AB’C’

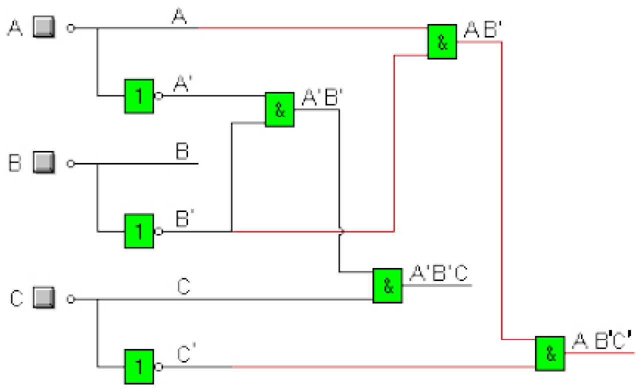
Comenzamos dibujando las entradas y junto a ellas las puertas NOT que nos permitan invertir las señales:



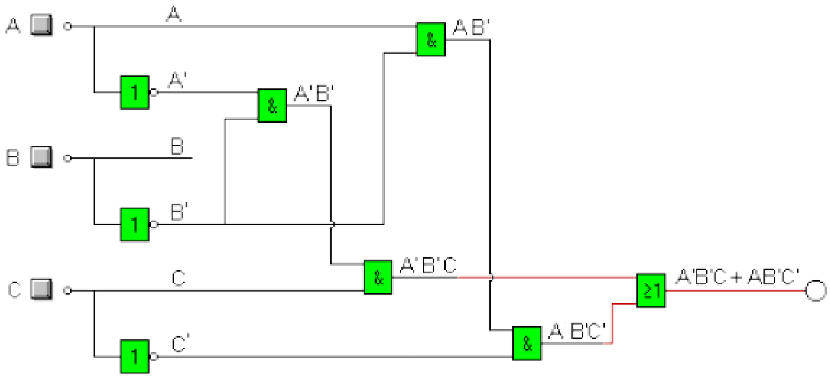
Para obtener el primer sumando, multiplicamos las variables de entrada correspondientes mediantes puertas AND.



Repetimos el último paso para el otro sumando:

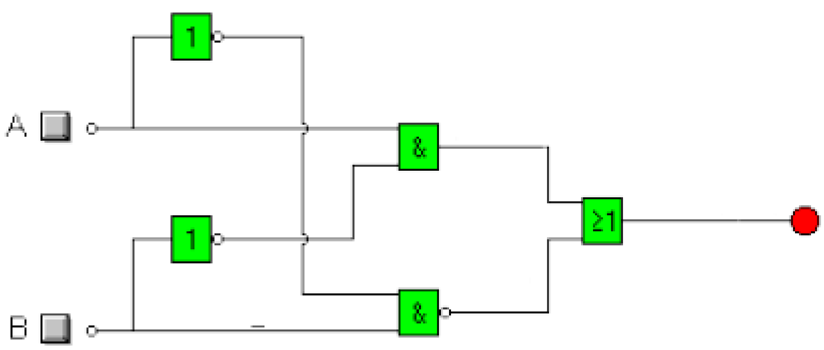


Finalmente, mediante una puerta OR, sumamos A’B’C y AB’C’:

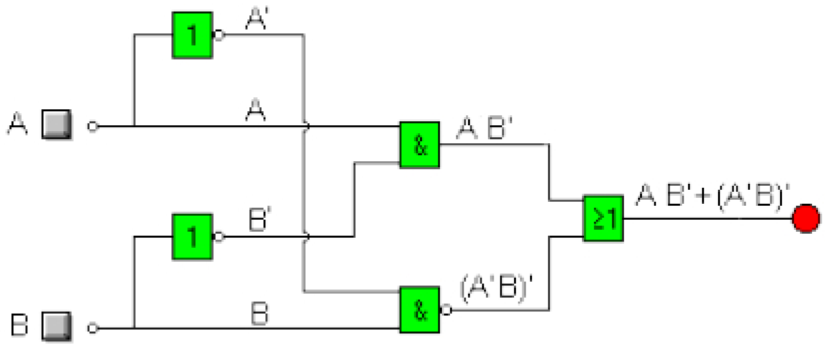


10. OBTENCIÓN DE LA TABLA DE LA VERDAD DE UN CIRCUITO YA DISEÑADO.

La tabla de la verdad, como hemos visto, sirve para obtener la función lógica y con ella poder diseñar el circuito electrónico. Pero es frecuente lo contrario, que nos den el circuito electrónico ya diseñado y que necesitemos obtener su tabla de la verdad para comprender su funcionamiento. Supongamos que nos piden la tabla de la verdad en el siguiente circuito con dos entradas A y B:



Debemos ir siguiendo el recorrido del circuito y obteniendo la función en cada cable hasta llegar a la salida S. Sabiendo ya la función de salida, podemos obtener la tabla de la verdad, tal y como estudiamos anteriormente:



$$S = AB' + (A'B)' = AB' + A + B' = A\,(B' + 1) + B' = A + B'$$

Teniendo en cuenta la función lógica obtenida, la tabla de la verdad será:

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

11. ANÁLISIS DE UN SISTEMA MEDIANTE BLOQUES.

Teniendo en cuenta todo lo aprendido, podemos reducir cualquier sistema electrónico por muy complejo que sea a tres bloques:

- Bloque de entrada. Está formado por todas las variables de entrada: las que activan o paran el funcionamiento del sistema.
- Bloque de proceso. Es el que genera las respuestas a partir de las variables de entrada. Está formado por las puertas lógicas que relacionan las entradas con las salidas permitiendo que el sistema cumpla con la tabla de la verdad.
- Bloque de salida. Mediante este tercer bloque el sistema actúa de acuerdo con la función de transferencia.

La forma de diseñar el sistema electrónico es tener claras cuántas y cuáles son las señales de entrada del sistema, cuál es la señal de salida, y a continuación, por medio de la tabla de la verdad, obtener la función lógica que nos permite diseñar el bloque de proceso, el cual constará de las puertas lógicas que permitan implementar esa función. Lo mejor es verlo mediante un ejemplo:

Ejemplo. Obtener el circuito lógico que se ajuste a las siguientes especificaciones:

- Un sistema de aire acondicionado se puede poner en marcha mediante un interruptor (A) manual.
- Se encenderá de forma automática, aunque el interruptor está apagado, cuando un termostato (B) detecte que la temperatura exterior pasa de 30 ºC.
- Existe también un detector (C) que desconecta el sistema, incluso estando el interruptor encendido, cuando la ventana está abierta.

Diseña el sistema electrónico que permite el control del aire acondicionado.

Vamos a determinar en primer lugar los bloques de entrada y de salida:

- A. Interruptor manual. 1=Encendido; 0=Apagado.
- B. Termostato. 1=T>30ºC; 0=T<30ºC.
- C. Detector. 1=Ventanas abiertas; 0=Ventanas cerradas.
- S=Salida. Puesta en marcha o apagado del sistema en función de las condiciones.

Determinadas las entradas y las salidas, diseñaremos la tabla de la verdad que explique el funcionamiento del sistema:

A	**B**	**C**	**S**
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0

1	1	0	1
1	1	1	0

El sistema no funcionará (S = 0) cuando haya ventanas abiertas (C = 1) o cuando el interruptor esté apagado y no haya temperatura alta en el exterior (A y B = 0). El resto de los casos la salida será 1.

Teniendo en cuenta la tabla de la verdad, ya podemos obtener la función lógica teniendo en cuenta las filas en las que la salida está activada:

$$S = A'BC' + AB'C' + ABC'$$

Si simplificamos la función:

$$S = A'BC' + AB'C' + ABC' = A'BC' + AC'(B'+B) = A'BC' + AC'$$

Implementando la función mediante puertas lógicas obtendríamos: