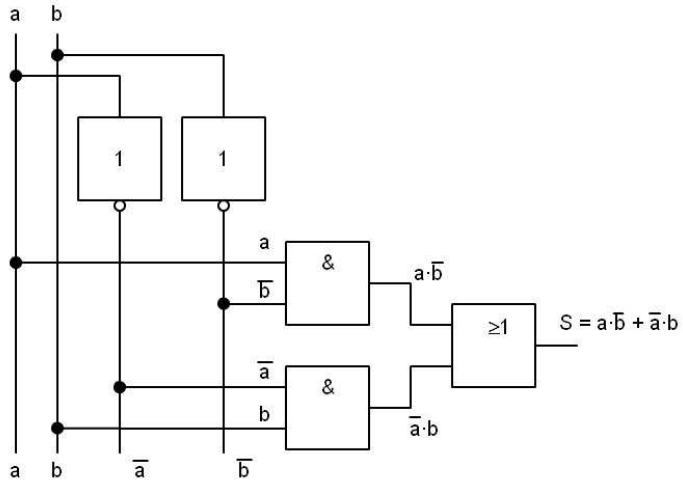
Autor: Antonio Bueno

Unidad didáctica: "Electrónica Digital"



CURSO 4º ESO

Autor: Antonio Bueno

Unidad didáctica: "Electrónica Digital"

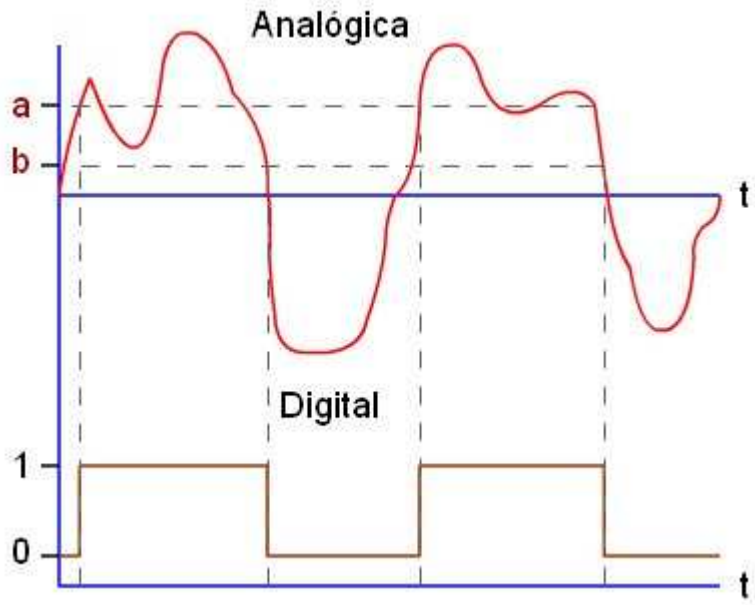
ÍNDICE

1.- Introducción.

La electrónica digital, se encuentra en pleno desarrollo, la mayor parte de los sistemas electrónicos se basan en ella.

En este tema estudiaremos las bases sobre las que se asienta. Sistemas de numeración y álgebra de boole. También obtendremos funciones, aprenderemos a simplificarlas y a crear circuitos que las implementan. Con todo esto obtendremos un diseño que servirá para resolver un problema real.

Existen una gran diversidad de sistemas digitales, tan solo estudiaremos una pequeña parte, con la que hacernos a la idea de su uso.



Señales analógica y digital

Una señal analógica es aquella que puede tener infinitos valores, positivos y/o negativos. Mientras que la señal digital sólo puede tener dos valores 1 o 0.

En el ejemplo de la figura, la señal digital toma el valor 1 cuando supera al valor a, y toma valor 0 cuando desciende por debajo del valor b. Cuando la señal permanece entre los valores a y b, se mantiene con el valor anterior.

Esto supone una gran ventaja, hace que la señal digital tenga un alto grado de inmunidad frente a variaciones en la transmisión de datos.

Pero tiene el inconveniente de que para transmitir una señal analógica debemos hacer un muestreo de la señal, codificarla y posteriormente transmitirla en formato digital y repetir el proceso inverso. Para conseguir obtener la señal analógica original todos estos pasos deben hacerse muy rápidamente. Aunque los sistemas electrónicos digitales actuales trabajan a velocidades lo suficientemente altas como para realizarlo y obtener resultados satisfactorios.

El muestreo de una señal consiste en convertir su valor en un valor binario, por lo que es necesario estar familiarizado con los sistemas de numeración.

Regresar al índice

2.- Sistemas de numeración.

Se define la base de un sistema de numeración como el número de símbolos distintos que tiene.

Normalmente trabajamos con el sistema decimal que tiene 10 dígitos: 0,1,2,3,4,5,6,7,8,9.

La representación de un número N en un sistema de base b, puede realizarse mediante el desarrollo en forma polinómica.

$N=a_n b^n + a_{n-1} b^{n-1} + ... + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + ...$

Donde:
b: base del sistema.
a_i: coeficientes que representan las cifras de los números.

Por ejemplo:

a) El número 723,54 en base 10, lo podemos expresar:

$$723,54 = 7x10^2 + 2x10^1 + 3x10^0 + 5x10^{-1} + 4x10^{-2}$$

b) El número 523,74 en base 8, lo podemos expresar:

$$523,74 = 5x8^2 + 2x8^1 + 3x8^0 + 7x8^{-1} + 4x8^{-2}$$

Regresar al índice

2.1- Sistema binario.

Consta de dos dígitos el 0 y el 1. A cada uno de ellos se le llama bit (binary digit). La forma de contar en este sistema es similar al decimal, es decir: 0, 1, 10, 11, 100, 101, 110, 111, 1000,...

Para cambiar un número de sistema binario a decimal se procede de la siguiente forma:

Primero se expresa el número binario en su polinomio equivalente, a continuación se calcula el polinomio y el resultado es el número en base 10.

$abcde,fg_{(2)} = N_{(10)}$

$N = a2^4 + b2^3 + c2^2 + d2^1 + e2^0 + f2^{-1} + g2^{-2}$

De la coma a la izquierda son los exponentes positivos y de la coma a la derecha son los exponentes negativos.

Por ejemplo:

a) El número 11010,11 en base 2, lo podemos expresar en base 10:

$$1x2^4 + 1x2^3 + 0x2^2 + 1x2^1 + 0x2^0 + 1x2^{-1} + 1x2^{-2} = 16 + 8 + 0 + 2 + 0 + 0,5 + 0,25 = 26,75$$

Observar como se calcula la parte de después de la coma.

b) El número 101011,101 en base 2, lo podemos expresar en base 10:

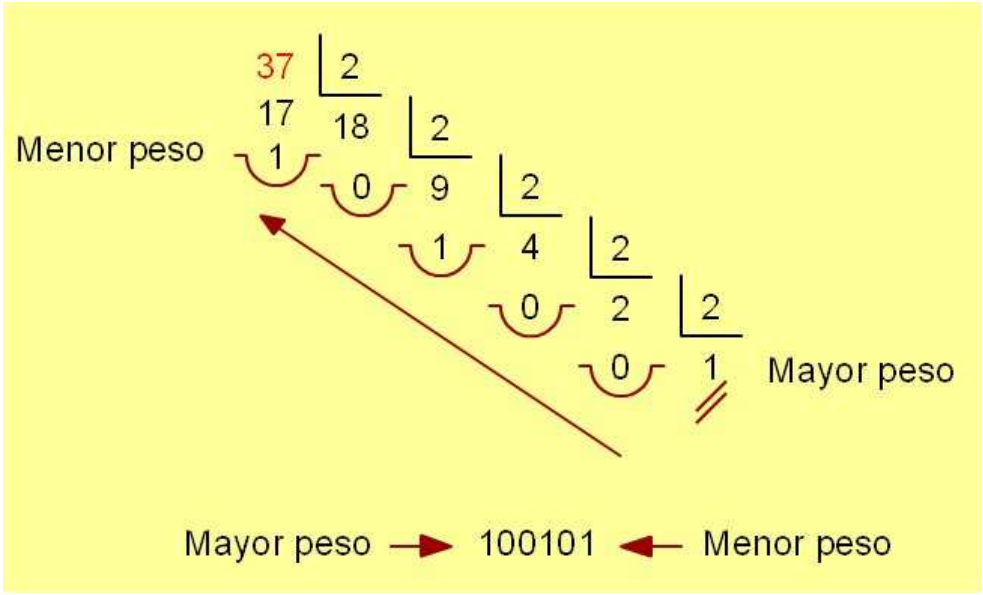
$$1x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 + 1x2^{-1} + 0x2^{-2} + 1x2^{-3} = 32 + 0 + 8 + 0 + 2 + 1 + 0,5 + 0 + 0,125 = 43,625$$

Para realizar el cambio de base decimal a base binaria de procede como se indica a continuación:

Se divide número decimal por dos, continuamente hasta que todos los restos y cocientes sean 0 o 1. El número binario será el formado por el último cociente (bit de mayor peso) y todos los restos.

Por ejemplo:

a) El número 37 en base decimal, lo podemos expresar:



37 en base 10 = 100101 en base 2

Regresar al índice

2.2- Sistema hexadecimal.

Consta de dieciséis dígitos el 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y el F. La forma de contar en este sistema es similar al decimal, es decir: 0, 1, 2,..., E, F, 10, 11, 12,..., 1E, 1F, 20, 21, 22,..., 2E,2F, 30, 31, 32,..., 3E, 3F,...

La equivalencia entre hexadecimal y decimal es:

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Para cambiar un número de sistema hexadecimal a decimal se procede de la siguiente forma:

Primero se expresa el número hexadecimal en su polinomio equivalente, a continuación se calcula el polinomio y el resultado es el número en base 10.

$..abcde_{(16)} = N_{(10)}$

$N = ... a16^4 + b16^3 + c16^2 + d16^1 + e16^0$

Por ejemplo:

a) El número 3A1 en base 16, lo podemos expresar en base 10:

$$3x16^2 + (A)10x16^1 + 1x16^0 = 768 + 160 + 1 = 929$$

b) El número 3BF8 en base 16, lo podemos expresar en base 10:

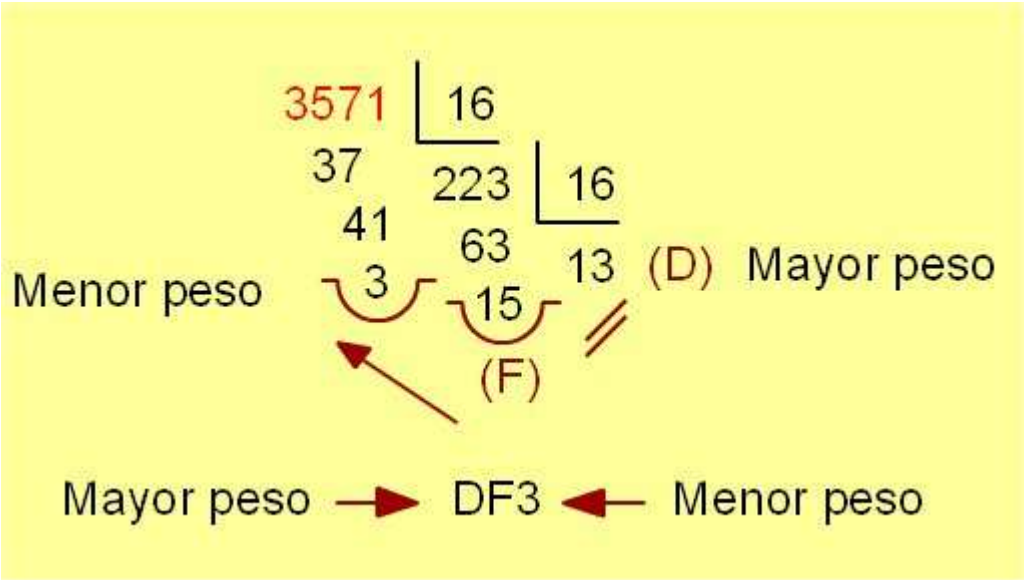
$$3x16^3 + (B)11x16^2 + (F)15x16^1 + 8x16^0 = 12288 + 2816 + 240 + 8 = 15352$$

Para realizar el cambio de base decimal a base hexadecimal de procede como se indica a continuación:

Se divide número decimal por 16, continuamente hasta que todos los restos y cocientes sean valores entre 0 y 15(F). El número hexadecimal será el formado por el último cociente (bit de mayor peso) y todos los restos.

Por ejemplo:

a) El número 3571 en base decimal, lo podemos expresar:



3571 en base 10 = DF3 en base 16

La fácil conversión que tiene este sistema con el binario lo hace muy atractivo.

La equivalencia entre Hexadecimal, decimal y binario es:

Hexadecimal	Decimal	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Para cambiar un número de sistema binario a hexadecimal se procede de la siguiente forma:

Primero se agrupa el número binario en bloques de cuatro bits empezando por el bit de menor peso. Luego se convierte cada uno de los grupos en su equivalente Hexadecimal.

Por ejemplo:

a) El número 11101011011 en base 2, lo podemos expresar en base 16:

111,0101,1011 = 75B

b) El número 11011010110110 en base 2, lo podemos expresar en base 16:

11,0110,1011,0110 = 36B6

Para cambiar un número de sistema hexadecimal a binario se procede de manera similar:

Primero se convierte cada dígito hexadecimal en su equivalente binario de cuatro bits. Luego se agrupan y ya está.

Por ejemplo:

a) El número 15E8 en base 16, lo podemos expresar en base 2:

15E8 = 0001,0101,1110,1000 =0001010111101000

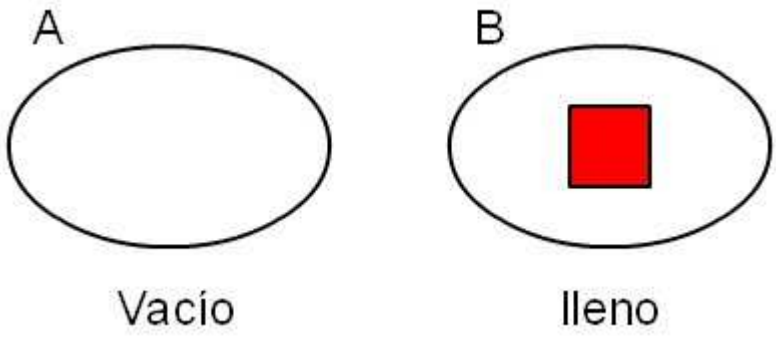
b) El número 123 en base 16, lo podemos expresar en base 2:

123 = 0001,0010,0011 = 000100100011

Regresar al índice

3.- Álgebra de Boole, álgebra de conjuntos.

En 1847 el matemático inglés George Boole desarrolló un álgebra que afecta a conjuntos de dos tipos, conjunto vacío y conjunto lleno.



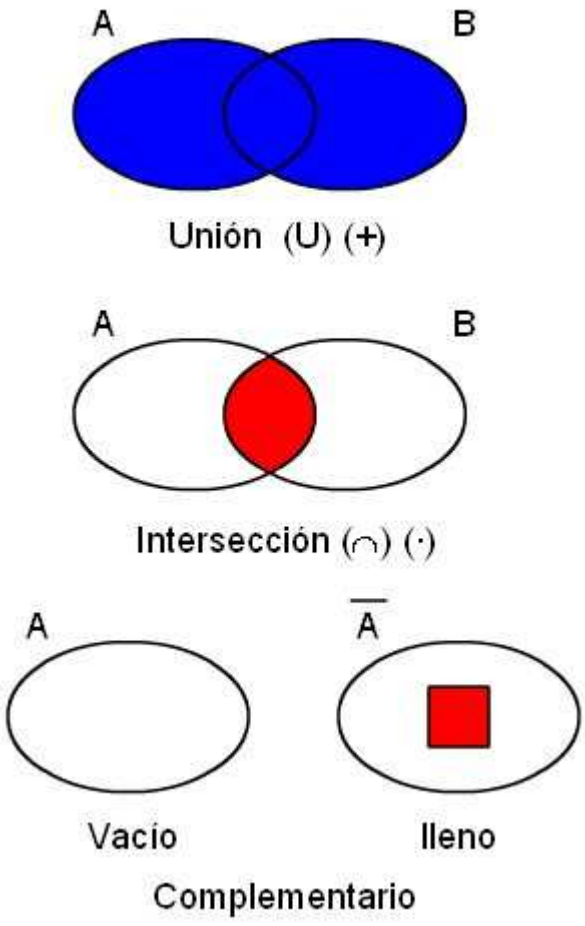
Conjunto vacío y conjunto lleno

Este álgebra se puede extrapolar a sistemas que tienen dos estados estables, "0" y "1", encendido y apagado, abierto y cerrado, ...

Regresar al índice

3.1- Operaciones lógicas.

El álgebra de conjuntos se desarrolló con las operaciones unión de conjuntos (U) (+), intersección de conjuntos (∩) (·) y el complementario.



Operaciones lógicas

De ahora en adelante denotaremos a la unión como (+) y a la intersección como (·). ¡Ojo! No son la suma y multiplicación ordinarias.

Las operaciones lógicas se pueden representar como funciones:

Para la unión, S = A + B.
Para la intersección, S = A · B.
Complementario o negación, S = Ā

Donde los conjuntos A y B (variables) pueden tener los dos estados 0, 1.

Función unión o suma lógica (+):

$$S = a + b$$

La función toma valor lógico "1" cuando **a** o **b** valen "1". También se la conoce como función Or (O).
Otra forma de representarlo es en la llamada tabla de verdad.

a	b	S = a+b
0	0	0
0	1	1
1	0	1
1	1	1

La tabla de verdad, representa en el lado izquierdo todas las combinaciones que se pueden dar de las variables y en la parte derecha el valor que toma la función para cada uno de ellos.

Función intersección o multiplicación lógica (·):

$$S = a \cdot b$$

La función toma valor lógico "1" cuando **a** y **b** valen "1". También se la conoce como función And (Y).
Otra forma de representarlo es en la llamada tabla de verdad.

a	b	S = a·b
0	0	0
0	1	0
1	0	0
1	1	1

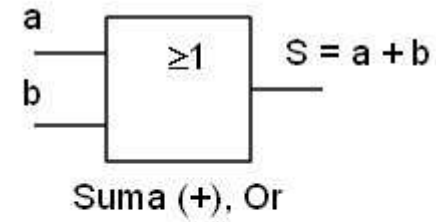
Función negación lógica o complementario (¯):

$$S = \bar{a}$$

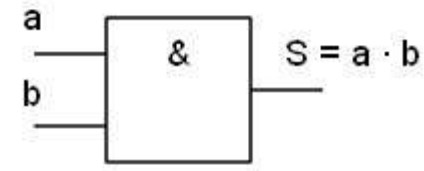
La función toma valor lógico "1" cuando **a** vale "0" y toma el valor "0" cuando **a** vale "1". También se la conoce como función Inversión.
Otra forma de representarlo es en la llamada tabla de verdad.

a	S = ā
0	1
1	0

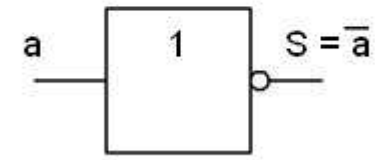
Los símbolos que representan estas funciones se pueden ver a continuación:



Suma (+), Or



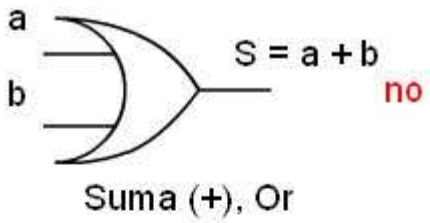
Multiplicación (·), And



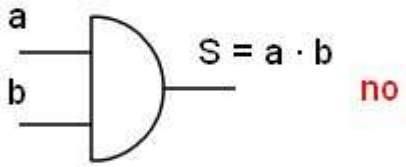
Complementario, inversión

Símbolos normalizados de la suma, multiplicación e inversión

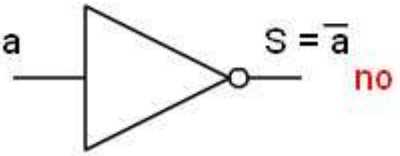
Los símbolos antiguos todavía se pueden ver en numerosos lugares por lo que se representan aquí, pero ya no deben utilizarse.



Suma (+), Or



Multiplicación (·), And



Complementario, inversión

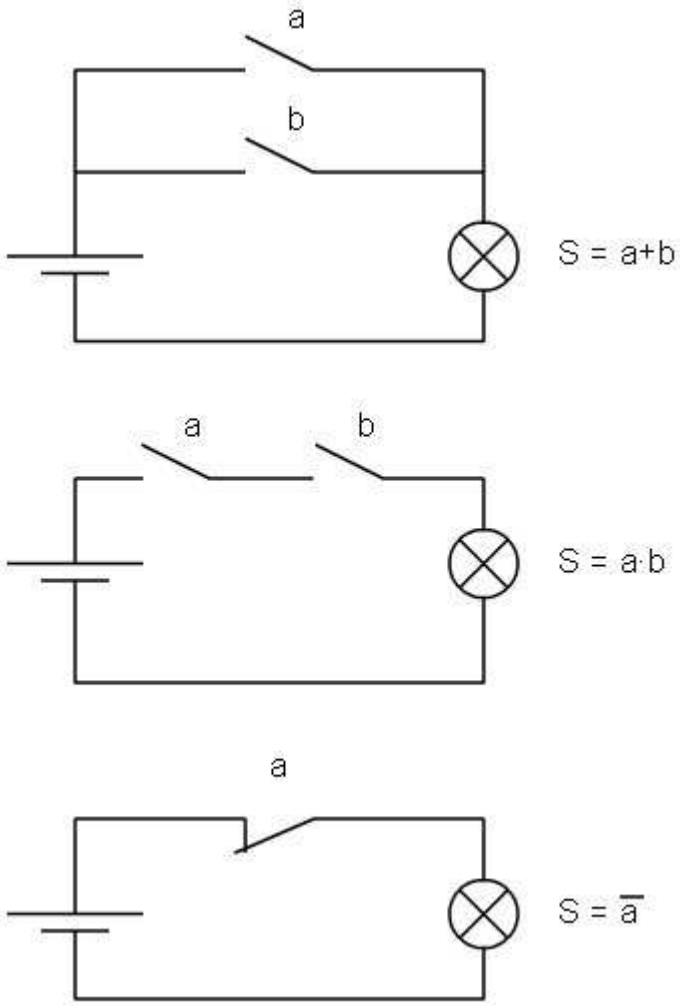
Símbolos antiguos de la suma, multiplicación e inversión en desuso no se deben utilizar

Regresar al índice

3.2- Puertas lógicas.

Las puertas lógicas son componentes físicos (electrónicos, eléctricos, mecánicos, neumáticos...) capaces de realizar las operaciones lógicas.

A continuación se implementan las tres puertas lógicas con interruptores.



Puertas Suma, multiplicación e inversión con interruptores

En la puerta suma (OR), cuando se cierra el interruptor a o el b, o los dos, luce la bombilla.

En la puerta multiplicación (AND), sólo cuando se cierra el interruptor a y el b luce la bombilla.

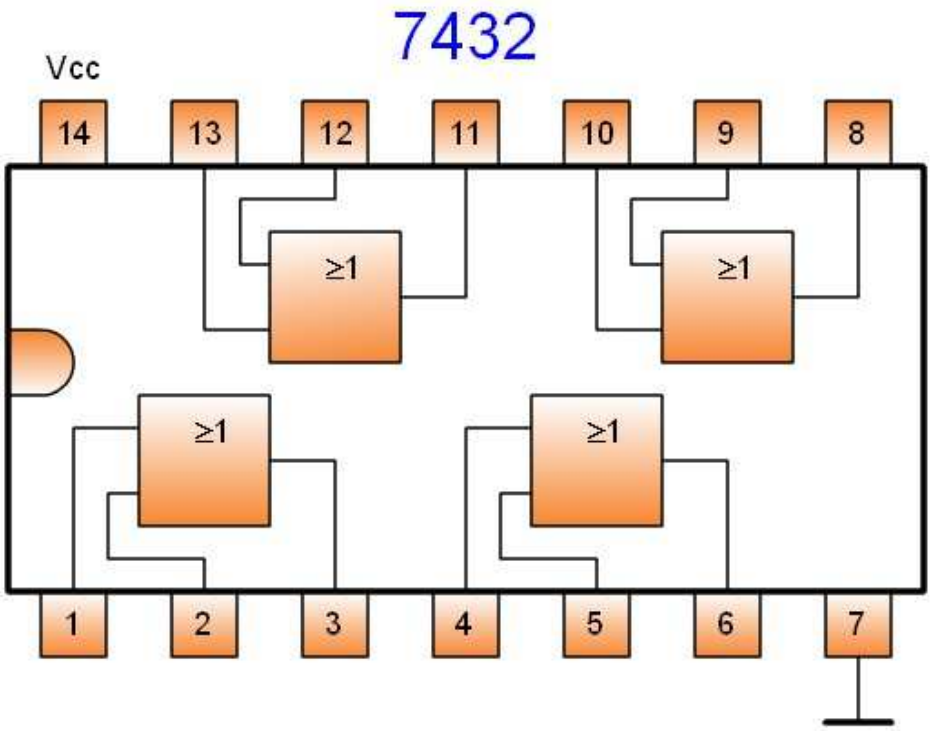
La puerta inversora tiene encendida la bombilla, y deja de estarlo cuando actuamos sobre el interruptor a, normalmente cerrado.

Las puertas lógicas se encuentran comercializadas en diversos formatos.

El más famoso es el formato electrónico, puesto que ocupa muy poco espacio y su coste es muy bajo. Se comercializan múltiples formatos, tecnologías y características eléctricas. No es el objetivo de esta unidad entrar en tanto detalle, por lo que mostraré un ejemplo sin entrar demasiado en los detalles.

Las puertas electrónicas corresponden a familias lógicas, una de las más utilizadas es la TTL (Transistor Transistor Logic). El circuito 7432 en sus distintas versiones (L, LS, S...), integra cuatro puertas suma (OR) de dos entradas en un encapsulado de 14 patillas, dos de las cuales son la de alimentación +5V(14) y masa (7).

El aspecto de dicho integrado puede verse a continuación:

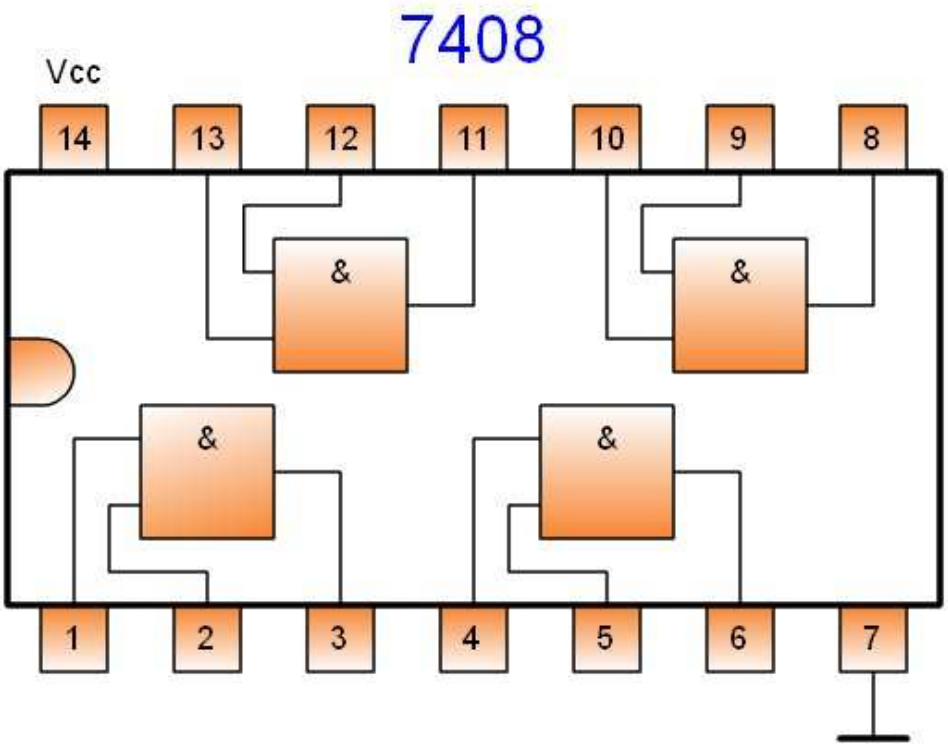


Circuito integrado 7432

Regresar al índice

Por otra parte el circuito 7408 integra también cuatro puertas, pero ahora multiplicación (AND) y sus terminales de alimentación.

Este es su aspecto:

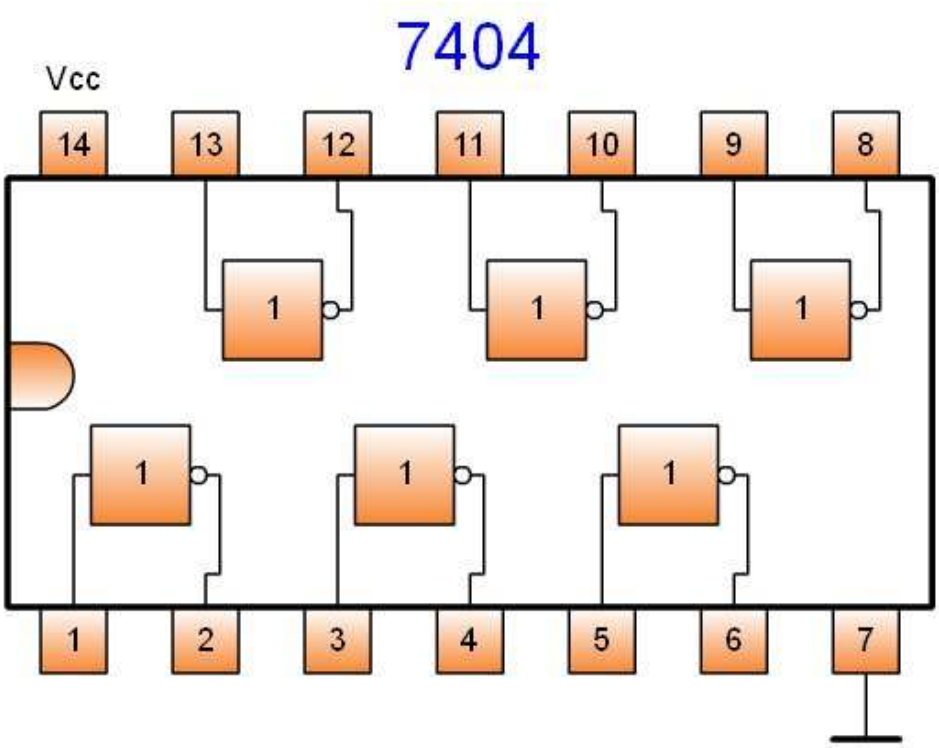


Circuito integrado 7408

Regresar al índice

El circuito 7404 integra 6 puertas inversoras con los terminales de alimentación.

Este es su aspecto:



Circuito integrado 7404

Para utilizar una de estas puertas se debe alimentar el circuito a 5 Voltios y conectar los terminales de dicha puerta. Cada una de ellas es independiente del resto.

Regresar al índice

Existen otras puertas que son combinación de las anteriores, la NOR y la NAND, que también se comercializan.

Función NOR:

$$S = \overline{a + b}$$

La función toma valor lógico "1" cuando **a** y **b** valen "0". Es la negación de la OR.
Esta es su tabla de verdad.

a	b	$S = \overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

Función NAND:

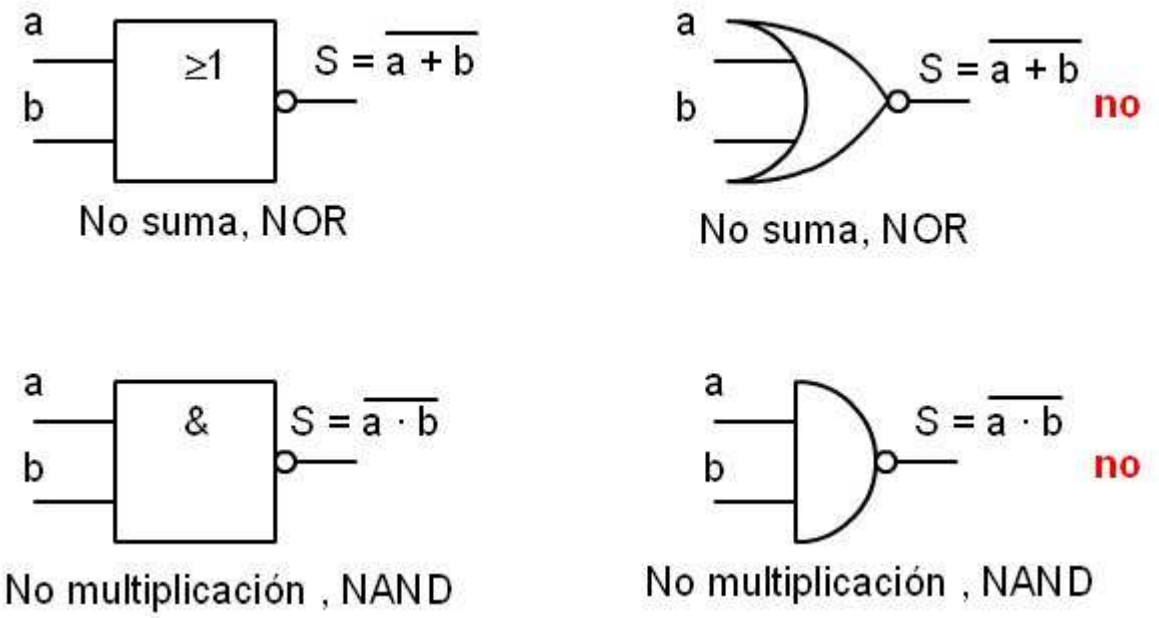
$$S = \overline{a \cdot b}$$

La función toma valor lógico "1" cuando **a** o **b** valen "0". Es la negación de la AND.
Esta es su tabla de verdad.

a	b	$S = \overline{a \cdot b}$
0	0	1

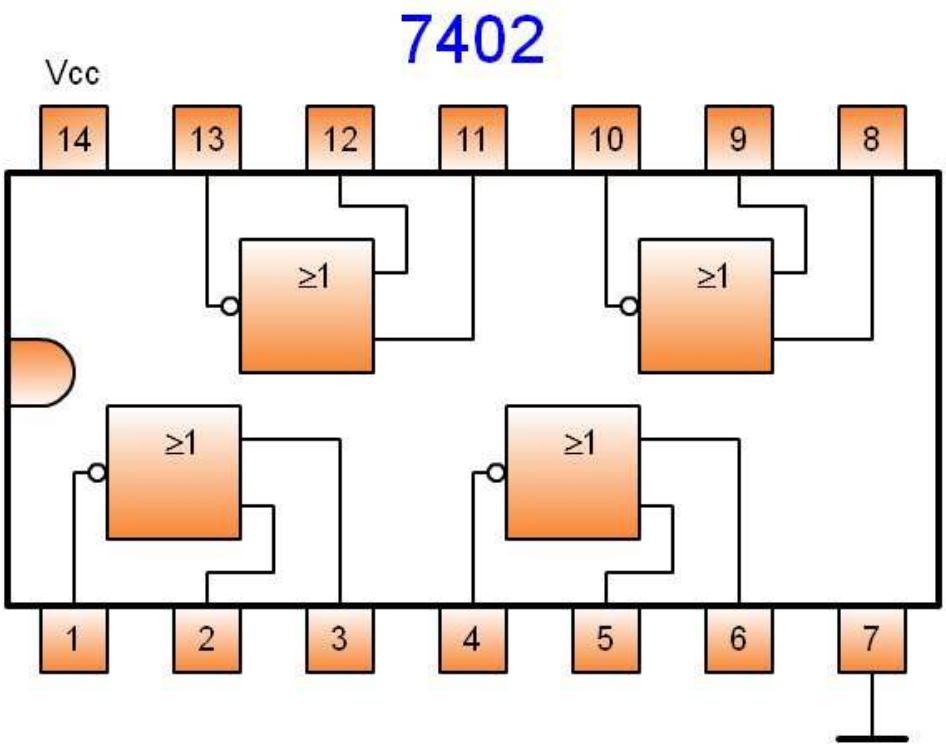
0	1	1
1	0	1
1	1	0

Su símbolo normalizado sería el siguiente, también se muestra el símbolo antiguo en desuso que no debe utilizarse:

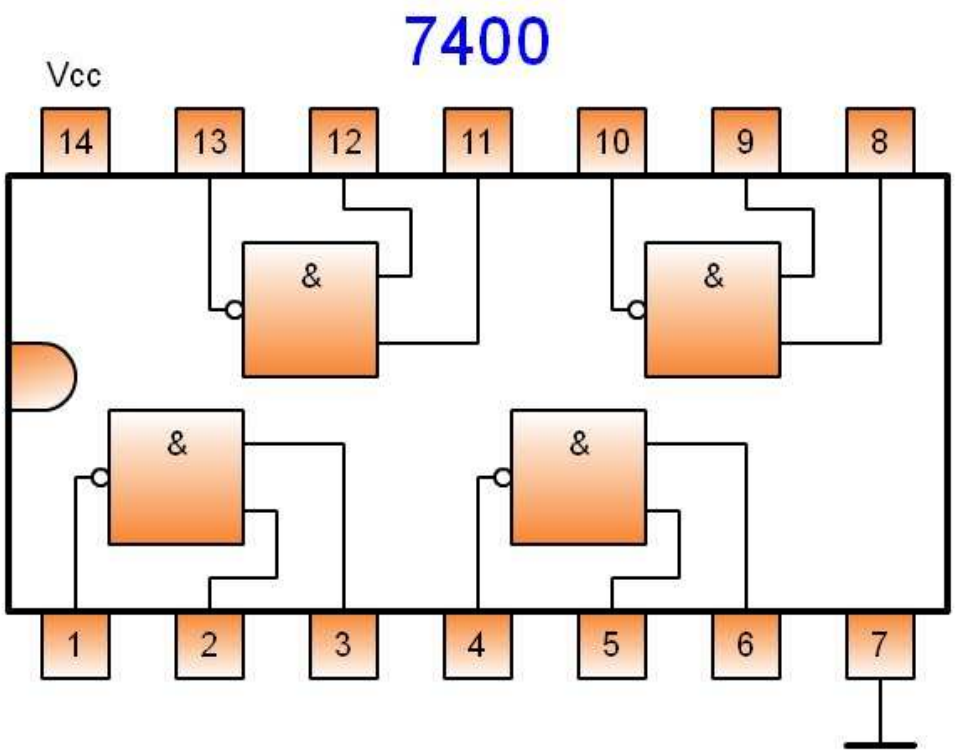


Símbolo de las puertas NAND y NOR, actual y antiguo en desuso

Este es el aspecto de los circuitos que las contienen:



Circuito integrado 7402, NOR



Circuito integrado 7400, NAND

Regresar al índice

3.3- Propiedades del álgebra de Boole.

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

1) Propiedad **conmutativa**:

- a+b = b+a
- a·b = b·a

2) Propiedad **asociativa**:

- a+b+c = a+(b+c)
- a·b·c = a·(b·c)

3) Propiedad **distributiva**:

- a·(b+c) = a·b + a·c
- a+(b·c) = (a+b)·(a+c) ¡ojo!

4) Elementos **neutros**: son el "0" para la suma y el "1" para el producto.

- a + 0 = a
- a ·1 = a

5) Elementos **absorbentes**: son el "1" para la suma y el "0" para el producto.

- a + 1 = 1
- a ·0 = 0

6) Ley del **complementario**:

- $a + \bar{a} = 1$
- $a \cdot \bar{a} = 0$

7) **Idempotente**:

- $a + a = a$
- $a \cdot a = a$

8) **Simplificativa**:

- $a + a \cdot b = a$
- $a \cdot (a+b) = a$

9) **Teoremas de Demorgan**:

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

Regresar al índice

4.- Funciones lógicas, tabla de verdad.

La función lógica S, es una expresión algebraica en la que se relacionan las variables independientes (a,b,c...) mediante las operaciones lógicas.

Por ejemplo:

$$S = a \cdot b + \bar{a} \cdot c + (a+b) \cdot \bar{c}$$

La forma más simple de definir una función lógica es mediante su tabla de verdad. Consiste en establecer todas las posibles combinaciones de las variables independientes en forma de tabla, e indicar el valor de S para cada una de ellas. El número total de combinaciones es 2n, siendo n el número de ellas.

El primer paso en resolución de circuitos lógicos es la obtención de la tabla de verdad y posteriormente obtener la función lógica a partir de esta. A continuación se muestra como obtener la función a partir de la tabla de verdad.

Por ejemplo:, una función lógica de tres variables puede ser:

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

Cuando a=0, b=0, c=0 la función S= 0,
Cuando a=0, b=0, c=1 la función S = 1,
Y así con el resto de combinaciones.

Se puede obtener de dos formas, como suma de productos (Minterms) o como producto de sumas (Maxterms).

Por ejemplo:

$S_1 = a \cdot b + \bar{a} \cdot b + a \cdot \bar{b}$ Minterms

$S_2 = (a+b) \cdot (\bar{a}+b) \cdot (a+\bar{b})$ Maxterms

Las funciones S1 y S2 son distintas.

Para obtener la función en suma de productos (Minterms) se opera de la forma siguiente:

Se deben tomar todas las combinaciones posibles de las variables donde la función tiene como valor "1", asignado el nombre de la variable cuando vale "1" y en nombre negado cuando vale "0", multiplicando las variables de una combinación. Y se suman todos los términos obtenidos de esta manera.

Por ejemplo,en la tabla de verdad anterior tenemos:

$S = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c$ por Minterms

Para obtener la función en productos de sumas (Maxterms) se opera de la forma siguiente:

Se deben tomar todas las combinaciones posibles de las variables donde la función tiene como valor 0, asignado el nombre de la variable cuando vale 0 y en nombre negado cuando vale 1, sumando las variables de una combinación. Y se multiplican todos los términos obtenidos de esta manera.

Por ejemplo,en la tabla de verdad anterior tenemos:

$S = (a+b+c) \cdot (a+\bar{b}+c) \cdot (\bar{a}+b+\bar{c}) \cdot (\bar{a}+\bar{b}+c)$ por Maxterms

Con el único objeto de no complicar demasiado el tema sólo se va a tratar la obtención de funciones y su simplificación por Minterms (suma de productos).

Regresar al índice

5.- Simplificación de funciones.

Tal como obtenemos una función a partir de la tabla de verdad, no se trata de la expresión más reducida de la misma. Por lo que se hace necesario simplificarla.

Cuanto menor es el tamaño de la función, es más rápida su resolución y el coste económico de implementación también es menor.

Regresar al índice

5.1- Simplificación mediante propiedades.

Se trata de aplicar las propiedades y teoremas del álgebra de Boole para obtener una función más reducida.

Para explicar este método lo mejor es emplear una función como ejemplo:

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

a) Primero agrupamos términos en parejas que tengan el mayor número de variables iguales. Se puede utilizar el mismo término varias veces si es necesario. Propiedad distributiva.

$$S = a \cdot b \cdot (c + \bar{c}) + \bar{a} \cdot c \cdot (b + \bar{b})$$

b) Las parejas $(c + \bar{c}) = 1$ y $(b + \bar{b}) = 1$ Ley del complementario.

$$S = a \cdot b \cdot 1 + \bar{a} \cdot c \cdot 1$$

c) Quitamos el 1. Elemento neutro para la multiplicación.

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

Esta ya es la expresión simplificada de la función inicial. Generalmente es necesario aplicar más propiedades hasta llegar a ella.

Regresar al índice

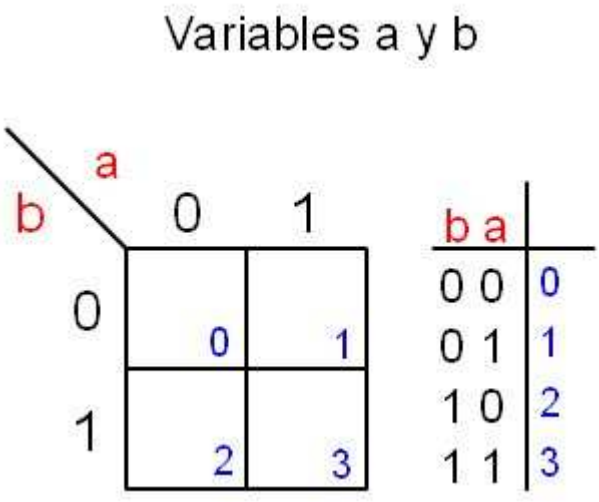
tr>

5.2- Simplificación mediante mapas de Karnaugh.

Es un método gráfico de simplificación que se usa cuando se utilizan pocas variables.

Se trata de una tabla donde se colocan las variables de manera que la intersección de las variables obtiene el valor que toma la función para esas variables. Además la distribución es tal que siempre las combinaciones adyacentes (que se diferencian en un bit) quedan juntas.

El mapa de dos variables es:



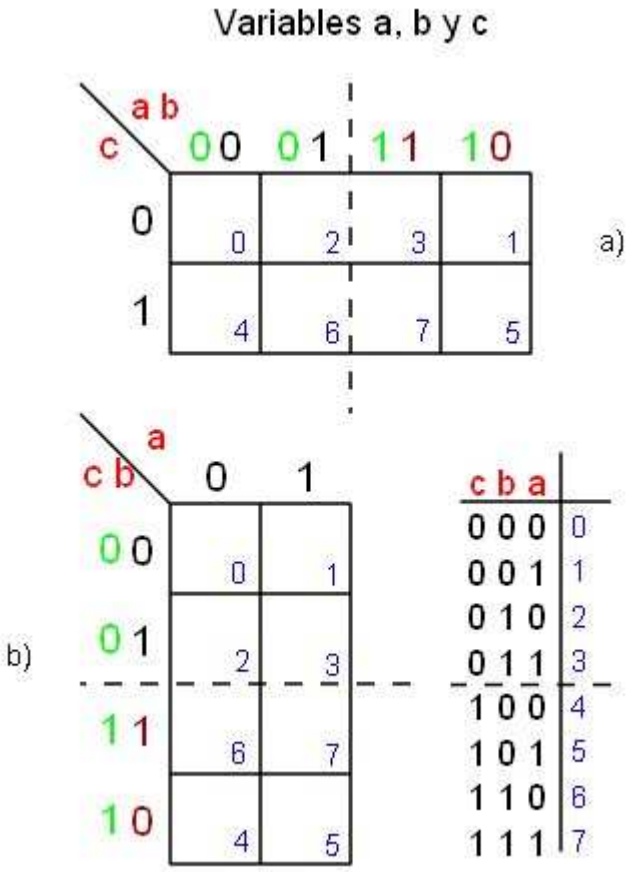
Mapa de Karnaugh de dos variables

Los valores internos 0, 1, 2 y 3 indican la combinación natural de las variables a y b, que tomaran el valor "0" o "1" según corresponda.

Para obtener un mapa de tres variables se crea el simétrico del de dos variables y se añade una variable nueva de valor "0" para el mapa antiguo y de valor "1" para el nuevo. Esto puede hacerse

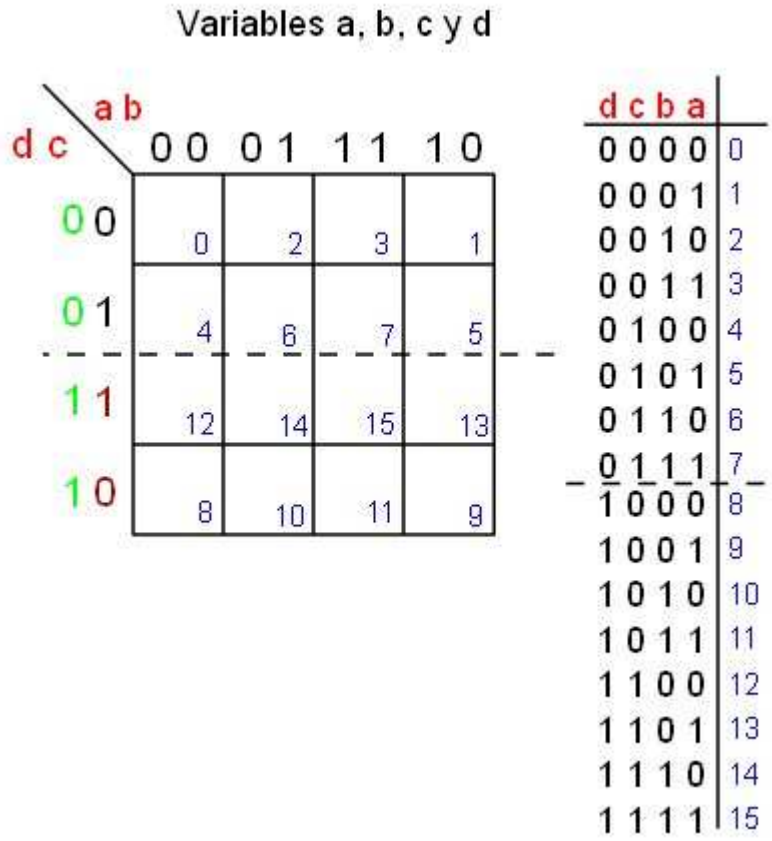
horizontalmente o verticalmente.

Ahora el valor de cada combinación debe colocarse en la celda correspondiente.



Mapa de Karnaugh de tres variables a) horizontal, b) vertical

Para obtener el mapa de cuatro variables, se parte del mapa de tres y creamos el simétrico horizontal o vertical del anterior. Ponemos la nueva variable y le añadimos "0" a los valores del mapa antiguo y "1" a los del mapa nuevo.



Mapa de Karnaugh de cuatro variables

Se opera de la misma forma para crear el resto de mapas.

Para obtener la expresión simplificada de una función con este sistema de precede de la forma siguiente:

- Una vez seleccionado el mapa según sea el número de variables, a partir de la tabla de verdad se sitúan los "1" o "0" en la celda correspondiente. En el caso de que existan términos indefinidos (X) se toman como "1" o "0" en cada celda como más interese.
- Formar grupos de unos, con el siguiente criterio:
  - a) Se toman todos los unos que no se puedan agrupar con ningún otro.
  - b) Se forman grupos de dos unos que no puedan formar grupos de cuatro.
  - c) Se forman grupos de cuatro unos que no puedan formar grupos de ocho.
  - d) Etc.
  - e) Cuando se cubran todos los unos se detiene el proceso.
  - f) Cada grupo de unos debe formar una figura de cuatro lados teniendo en cuenta que el mapa se cierra por los extremos laterales, superior e inferior.
- Una vez establecidos los grupos se obtiene la expresión de S. Esta será una suma de tantos términos como grupos distintos de unos haya. Para cada uno de los grupos, si una variable toma el valor "0" en la mitad de las casillas y "1" en la otra mitad, no aparecerá el término; si toma el valor "1" en todas las casillas del grupo aparecerá de forma directa; y si toma el valor "0", de forma inversa.:

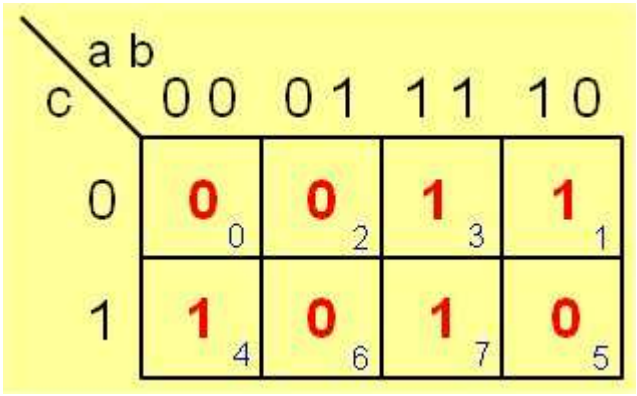
Veamos un ejemplo:

Por ejemplo, obtener simplificada por el método de Karnaugh la función lógica siguiente:

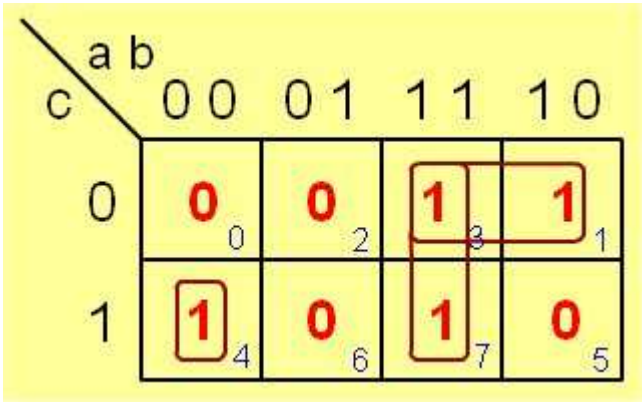
a	b	c	S
0	0	0	0
0	0	1	1

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

Creamos el mapa de Karnaugh y colocamos el valor de la función en cada celda.



Agrupamos los unos.



Obtenemos la expresión de S a partir de los grupos

Grupo (1,3) = $a \cdot \bar{c}$; b varía su valor y no aparece.

Grupo (3,7) = $a \cdot b$; c varía su valor y no aparece.

Grupo (4) = $\bar{a} \cdot \bar{b} \cdot c$

Luego la función será:

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

Observar que todavía se puede simplificar un poco más la función aplicando la propiedad distributiva:

$$S = a \cdot (\bar{c} + b) + \bar{a} \cdot \bar{b} \cdot c$$

Regresar al índice

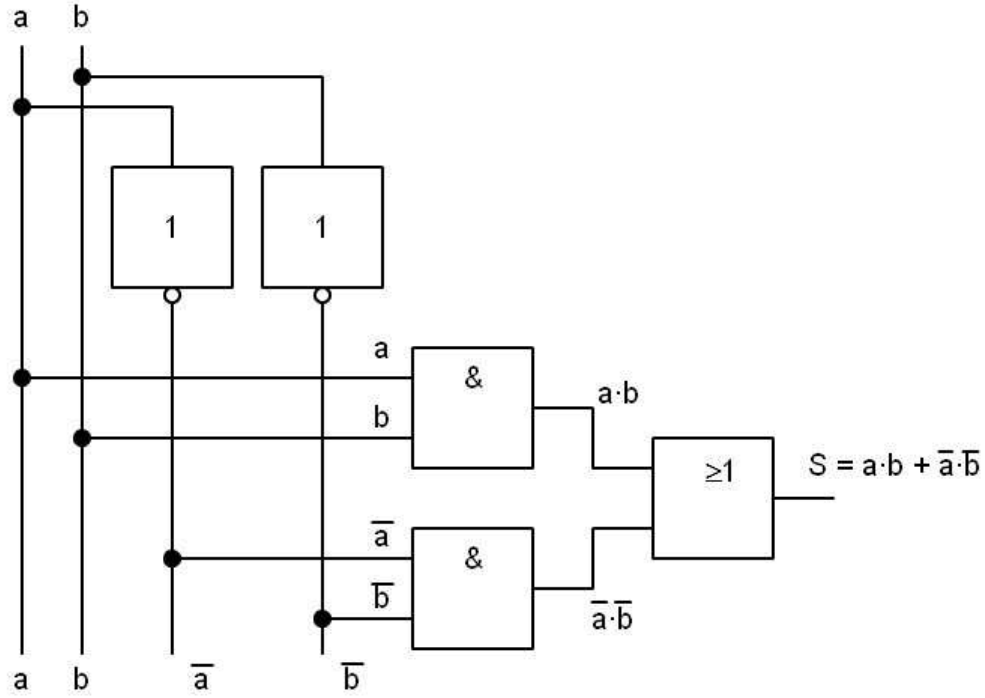
6.- Implementación de funciones con puertas de todo tipo.

Una vez obtenida la función simplificada, podemos implementarla con puertas lógicas que la resolverán.

Si en la función aparecen todos los términos negados en primer lugar realizamos la negación de todas

las variables y luego las operaciones.

Dada esta función $S = a \cdot \bar{b} + \bar{a} \cdot b$ su implementación será:



Implementación de una función lógica con puertas de todo tipo

Para implementarla necesitaríamos un circuito 7404 (2 puertas inversoras), un circuito 7408 (2 puertas AND) y un circuito 7432 (1 puerta OR). Total 5 puertas en 3 CI.

La función anterior también se encuentra integrada en un circuito electrónico y se la conoce con el nombre de or-exclusiva (EXOR).

Función or-exclusiva o (EXOR):

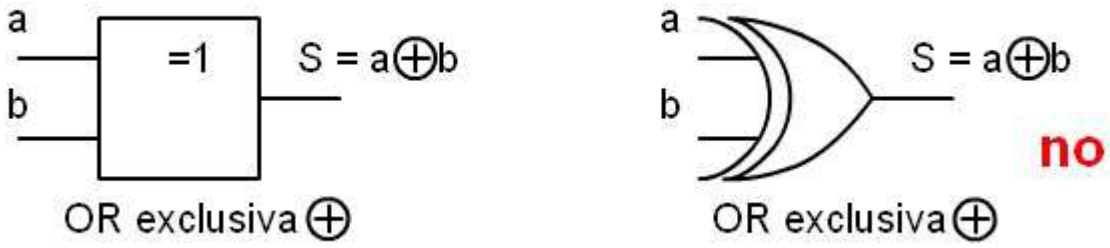
$$S = a \oplus b$$

La función toma valor lógico "1" cuando **a** o **b** valen "1" y toma el valor lógico "0" cuando **a** y **b** son iguales.

Su tabla de verdad es:

a	b	$S = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	1

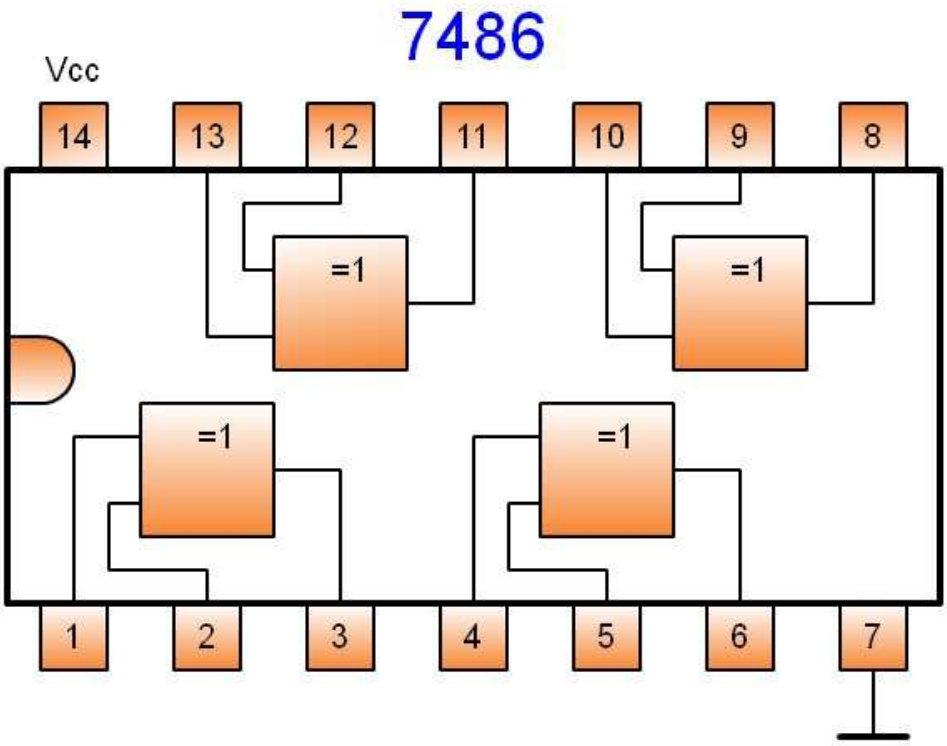
Su símbolo actual y el símbolo antiguo en desuso es:



Símbolo de la puerta OR-exclusiva y símbolo antiguo, no usar.

El circuito 7486 integra cuatro puertas EXOR de dos entradas en un encapsulado de 14 patillas, dos de las cuales son la de alimentación +5V(14) y masa (7).

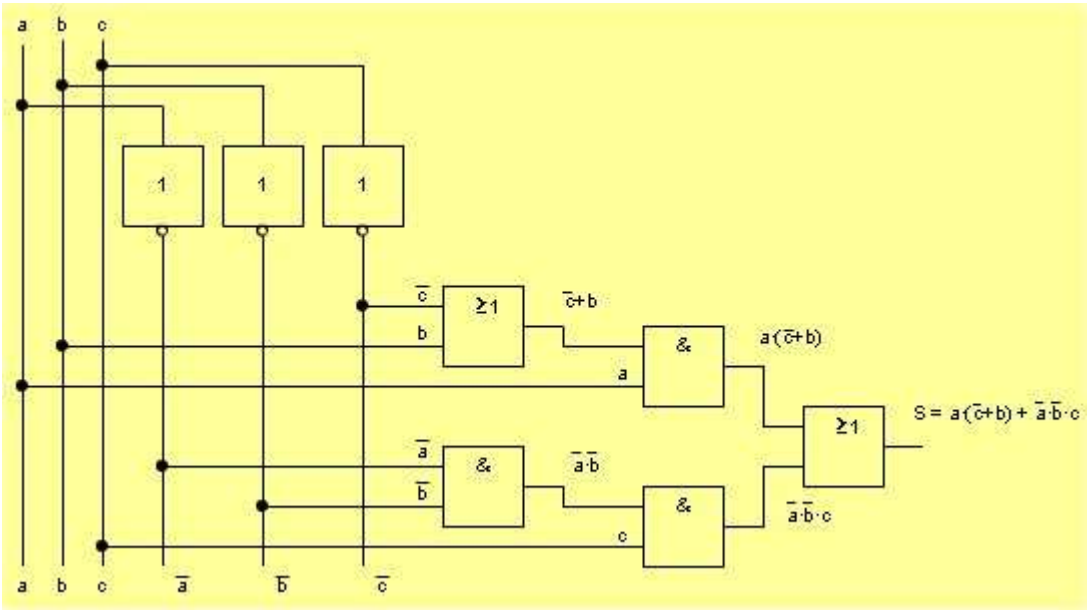
El aspecto de dicho integrado puede verse a continuación:



Circuito integrado 7486, EXOR

A continuación puede verse otro ejemplo.

Por ejemplo la función $S = a \cdot (\bar{c} + b) + \bar{a} \cdot \bar{b} \cdot c$ su implementación en puertas de todo tipo es:



Para implementarla necesitaríamos un circuito 7404 (3 puertas inversoras), un circuito 7408 (2 puertas AND) y un circuito 7432 (3 puertas OR). Total 8 puertas en 3 CI.

Nos interesa buscar la forma de implementar la función que ocupe el menor número posible de puertas y de circuitos integrados.

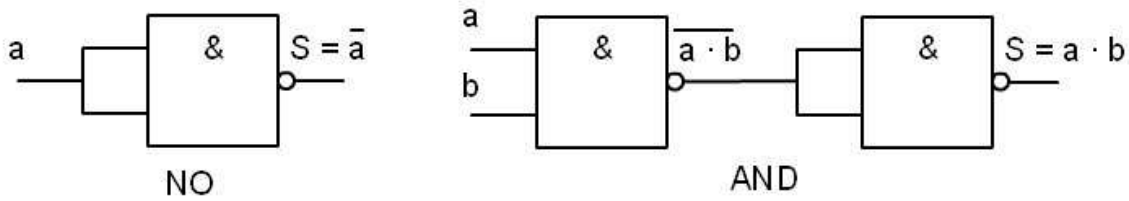
En este mundo tan competitivo, la menor cantidad de circuitos implica menor coste y menor cantidad de puertas implica mayor rapidez a la hora de resolver la función. Por todo ello vamos a estudiar como se implementaría con sólo puertas NAND o NOR.

Regresar al índice

7.- Implementación de funciones con puertas NAND o NOR.

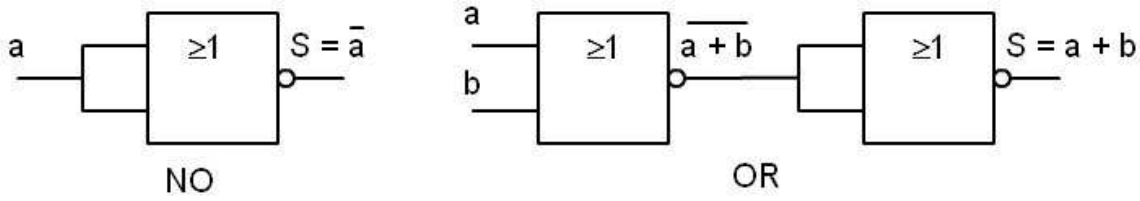
Toda función puede expresarse en función de multiplicaciones y negaciones o de sumas y negaciones.

A partir de puertas NAND puede obtenerse puertas Inversoras, y AND.



Puertas Inversora y AND a partir de NAND.

A partir de puertas NOR puede obtenerse puertas Inversoras, y OR.



Puertas Inversora y OR a partir de puertas NOR.

Para implementar una función con puertas NAND debemos convertirla en multiplicaciones y negaciones. Para ello utilizaremos los teoremas de Demorgan.

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

Veamos el proceso con un ejemplo:

Dada la función $S = a \cdot \overline{b} + \overline{a} \cdot b$ para cambiar la suma por una multiplicación seguimos los pasos:

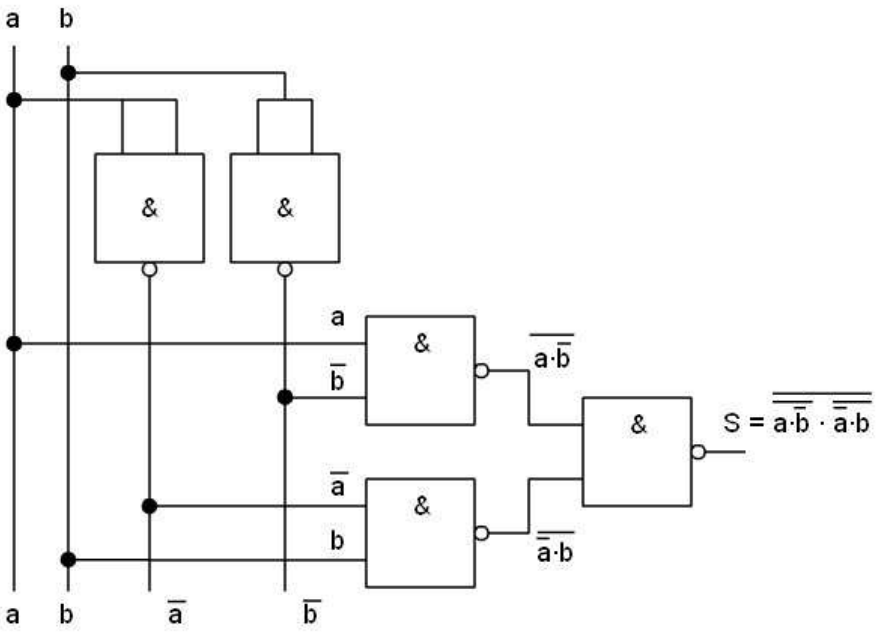
1.- Hacer una doble inversión en toda la función.

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

2.- Aplicar el teorema de Demorgan sobre la inversión de bajo y convertir la negación de términos sumados en la multiplicación de términos negados.

$$S = \overline{(\overline{a \cdot \overline{b}}) \cdot (\overline{\overline{a} \cdot b})}$$

3.- Con esto ya tenemos toda la función convertida en multiplicaciones y negaciones y se puede implementar con puertas NAND.



Función implementada con puertas NAND.

Para implementarla necesitaríamos 2 circuitos 7400 (4 + 1 puertas NAND). Esto supone un ahorro respecto la utilización de puertas de todo tipo.

Regresar al índice

Para implementar una función con puertas NOR debemos convertirla en sumas y negaciones. También utilizamos los teoremas de Demorgan.

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

Veamos el proceso con un ejemplo:

Dada la función $S = a \cdot \bar{b} + \bar{a} \cdot b$ para cambiar las multiplicaciones por sumas seguimos los pasos:

1.- Hacer una doble inversión sobre cada una de las multiplicaciones.

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

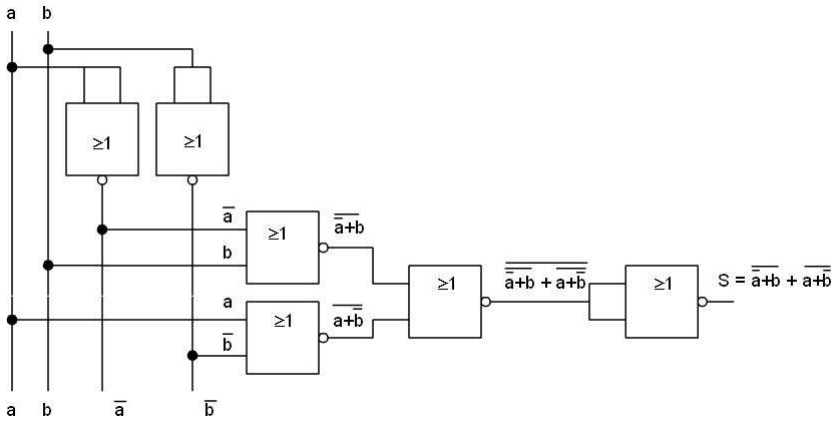
2.- Aplicar el teorema de Demorgan sobre la inversión de bajo y convertir la negación de términos multiplicados en la suma de términos negados.

$$S = \overline{(\bar{a} + \bar{\bar{b}})} + \overline{(\bar{\bar{a}} + \bar{b})}$$

3.- Ahora quitamos la doble inversión de las variables que la tienen.

$$S = \overline{(\bar{a} + b)} + \overline{(a + \bar{b})}$$

4.- Con esto ya tenemos toda la función convertida en sumas y negaciones y se puede implementar con puertas NOR.



Función implementada con puertas NOR.

Para implementarla necesitaríamos 2 circuitos 7402 (4 + 2 puertas NOR). Esto también supone ahorro respecto la utilización de puertas de todo tipo, sin embargo aparece una puerta más, haciendo la resolución de la función más lenta que la de puertas de todo tipo.

A continuación pueden verse más ejemplos.

Regresar al índice

Ejemplo, dada la función $S = a \cdot (\bar{c} + b) + \bar{a} \cdot \bar{b} \cdot c$ cambia su expresión para ser implementada en puertas NAND:

1.- Hacemos una doble inversión en toda la función.

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

2.- Aplicamos el teorema de Demorgan sobre la inversión de bajo y convertir la negación de términos sumados en la multiplicación de términos negados.

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

3.- Para eliminar la suma del interior del paréntesis realizamos la doble inversión del paréntesis.

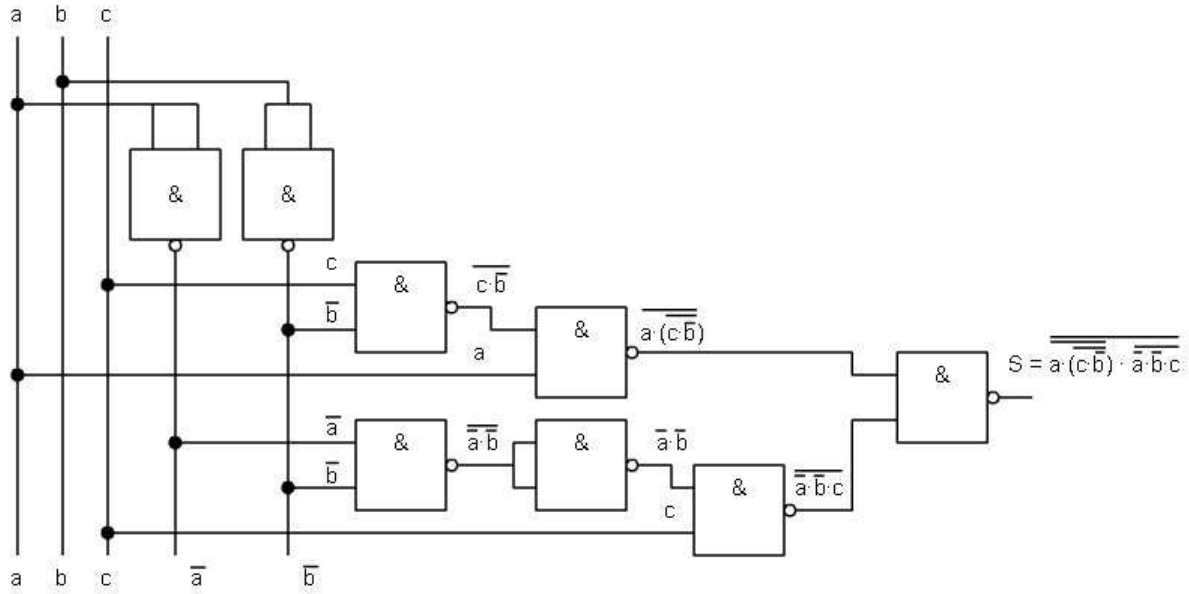
$$S = \overline{\overline{a \cdot (\overline{c} + b)} \cdot \overline{\overline{a} \cdot \overline{b} \cdot c}}$$

4.- Aplicamos el teorema de Demorgan sobre la inversión inferior del paréntesis y con ello se cambia la suma por una multiplicación.

$$S = \overline{\overline{a \cdot (\overline{c} \cdot \overline{b})} \cdot \overline{\overline{a} \cdot \overline{b} \cdot c}}$$

5.- Ahora eliminamos la doble inversión de la variable c y ya está.

$$S = \overline{\overline{a \cdot (\overline{c} \cdot \overline{b})} \cdot \overline{\overline{a} \cdot \overline{b} \cdot c}}$$



Función implementada con puertas NAND.

Regresar al índice

Otro Ejemplo, dada la función $S = a \cdot (\overline{c} + b) + \overline{a} \cdot \overline{b} \cdot c$ cambia su expresión para ser implementada en puertas NOR:

1.- Hacemos una doble inversión en una parte y otra de la suma.

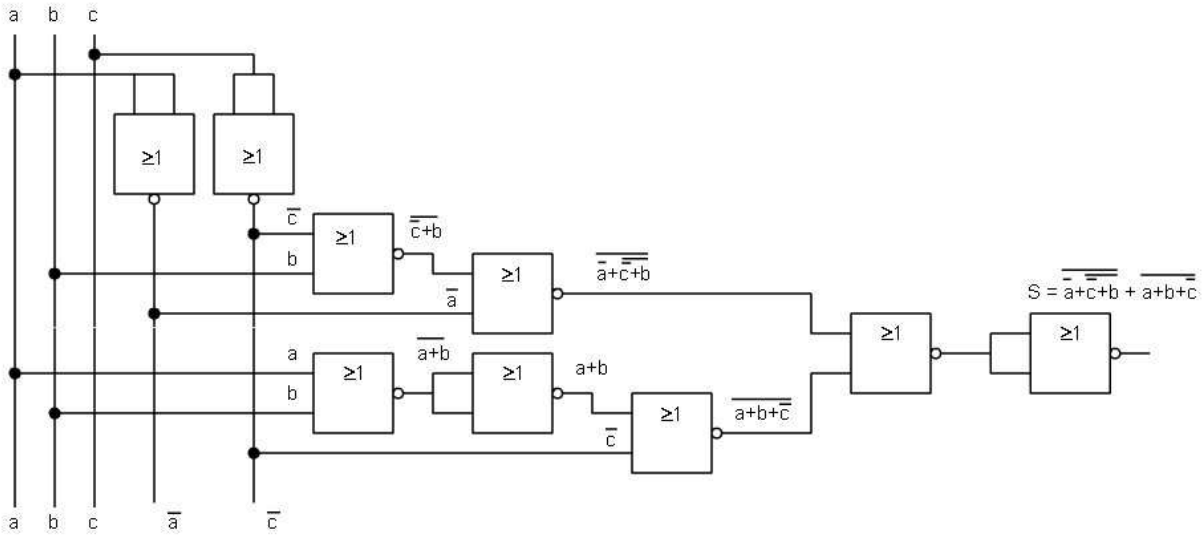
$$S = \overline{\overline{a \cdot (\overline{c} + b)}} + \overline{\overline{\overline{a} \cdot \overline{b} \cdot c}}$$

2.- Aplicamos el teorema de Demorgan sobre la inversión de bajo y convertir la negación de términos multiplicados en la suma de términos negados.

$$S = \overline{\overline{a} + \overline{(\overline{c} + b)}} + \overline{\overline{a} + \overline{b} + \overline{c}}$$

3.- Ahora eliminamos la doble inversión de las variables a y b, y ya está.

$$S = \overline{\overline{a} + \overline{(\overline{c} + b)}} + \overline{a + b + \overline{c}}$$



Función implementada con puertas NOR.

Regresar al índice

8.- Resolución de problemas lógicos.

Para resolver un problema real se deben seguir los siguientes pasos:

1.- Identificar las entradas y salidas del sistema. Las entradas serán las variables que tomarán el valor "0" o "1" en cada caso. Las salidas valdrán "1" cuando deban activarse.

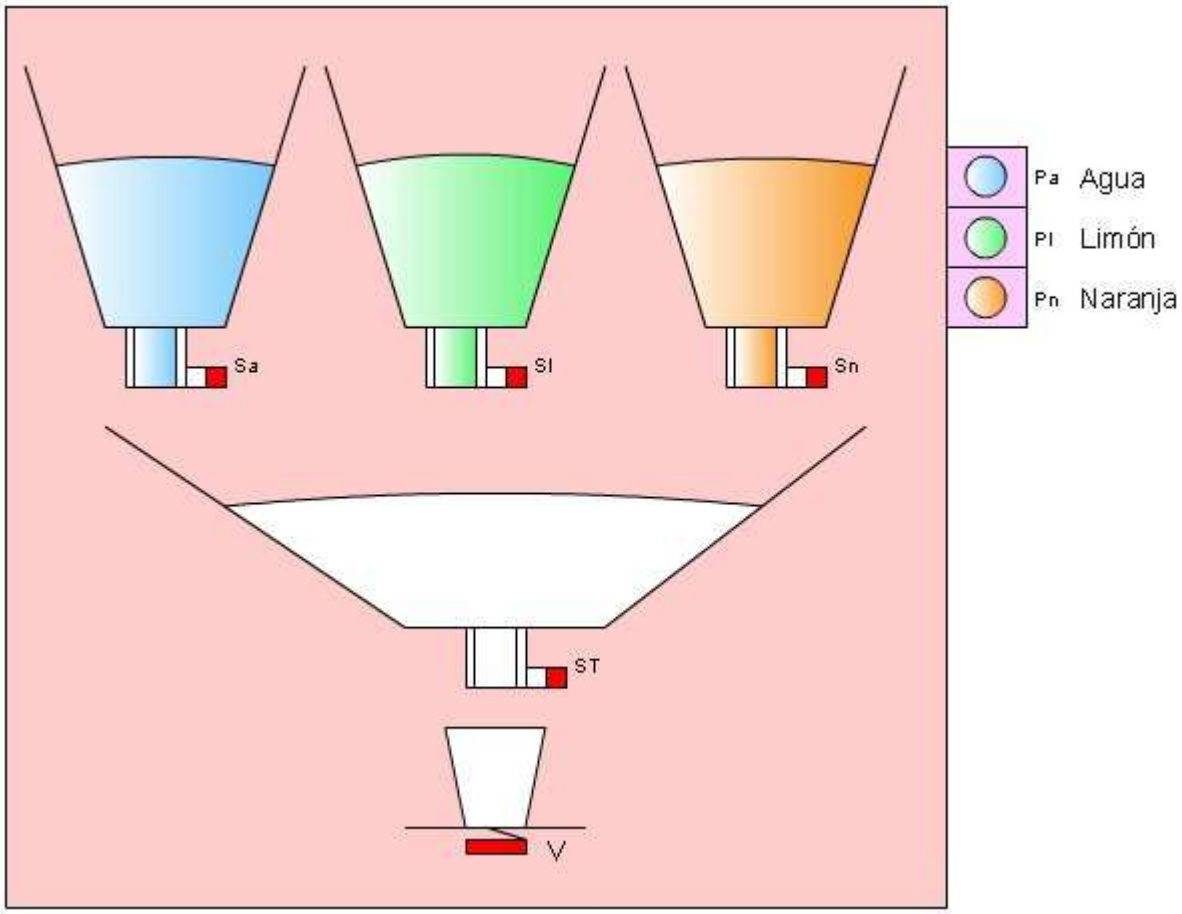
2.- Crear la tabla de verdad con todas las variables de entrada para cada salida.

3.- Obtener la función simplificada, bien utilizando las propiedades del álgebra de Boole o bien mediante el mapa de Karnaugh.

4.- Implementar la función con puertas de todo tipo, puertas NAND y puertas NOR.

Se elegirá la implementación que utilice el menor número de circuitos integrados y de puertas. Un menor número de puertas implica mayor velocidad en la obtención de la salida. Un menor número de circuitos implica menor costo del circuito.

Para ilustrar el método planteamos el siguiente ejercicio.



Máquina expendedora de agua-limón-naranja

Una máquina expendedora de refrescos puede suministrar agua fresca, agua con limón y agua con naranja. Pero no puede suministrar nunca limón solo, naranja sola, ni limón con naranja solos o con agua.

Los refrescos se encuentran en el interior de unos depósitos. La cantidad adecuada de cada líquido sale cuando se activa la electroválvula correspondiente, Sa (agua), Sl (limón), Sn (naranja). Y una vez caído el líquido sale hasta el vaso si está activada la salida general (ST), y se encuentra el vaso en su sitio (V).

Para seleccionar el líquido que queremos tenemos tres pulsadores Pa (agua), Pl (limón) y Pn (naranja). Deben pulsarse uno o dos según lo que deseemos, pero recordar que si se pulsan los que no corresponde no debe salir nada.

Diseñar el circuito digital capaz de resolver el problema y elegir aquel capaz de resolver el problema con mayor prontitud y menor coste.

Regresar al índice

1.- Identificar entradas y salidas:

Entradas, serán los pulsadores **Pa, Pl, Pn** y el sensor que detecta la presencia del vaso **V**.

Puesto que el problema no especifica nada entendemos que un pulsador pulsado será "1" y no pulsado será "0". Cuando hay vaso V será "1" y cuando no hay vaso V será "0".

Salidas, serán todas las electroválvulas sobre las que hay que actuar, **Sa, Sl, Sn** y **ST**.

Como tampoco se dice nada al respecto cuando la electroválvula en cuestión valga "1" permitirá que salga la cantidad de líquido necesario.

2.- Crear la tabla de verdad.

Como existen cuatro entradas y cuatro salidas deberíamos crear cuatro tablas de verdad una para cada salida. Pero para simplificar y dar una visión más general, sobre una misma tabla de verdad vamos a colocar las cuatro salidas, que se deben resolver de forma independiente cada una de ellas.

Luego la tabla debe tener 2^4 combinaciones = 16.

Si elegimos la variable de entrada de existencia de vaso la de mayor peso, luego la de agua y luego las otras dos tendremos una visión más fácil del problema.

El orden de situación de las salidas no importa puesto que son independientes.

Entradas				**Salidas**			
V	Pa	Pl	Pn	ST	Sa	Sl	Sn
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	0	0	0	0

En la tabla observamos que solamente se permite que salga el refresco cuando hay vaso.

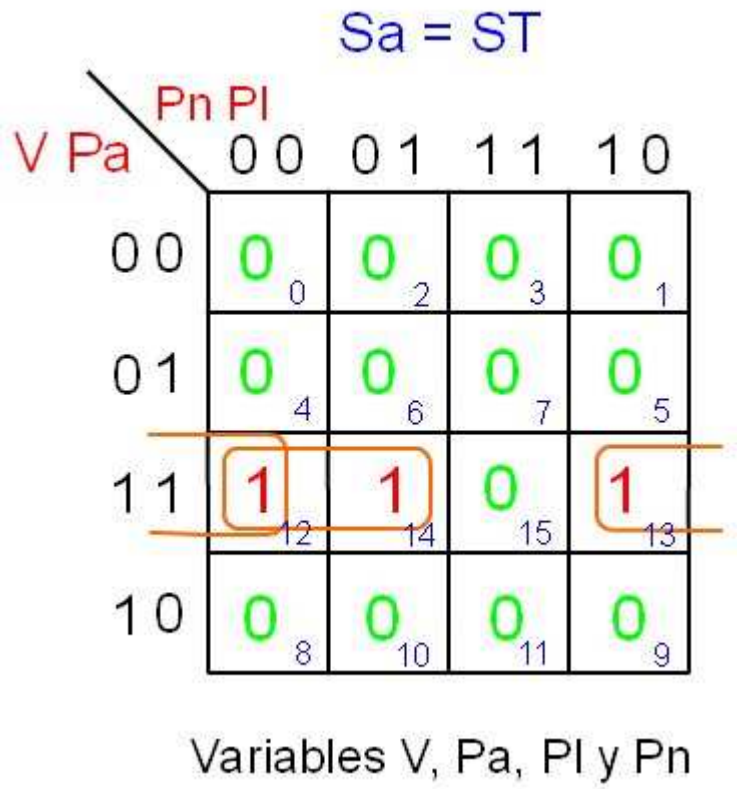
3.- Obtener la función simplificada.

En este caso debemos obtener cuatro funciones.

La función de la electroválvula ST y Sa es la misma.

$$Sa = V \cdot Pa \cdot \overline{Pl} \cdot \overline{Pn} + V \cdot Pa \cdot \overline{Pl} \cdot Pn + V \cdot Pa \cdot Pl \cdot \overline{Pn}$$

Si la simplificamos por medio del mapa de Karnaugh, tendremos dos grupos (12,14) y (13,12), en el primero Pl varía y no se tiene en cuenta y en el segundo Pn varía y no se tiene en cuenta.



$$ST = Sa = V \cdot Pa \cdot \overline{Pn} + V \cdot Pa \cdot \overline{Pl} = V \cdot Pa \cdot (\overline{Pl} + \overline{Pn})$$

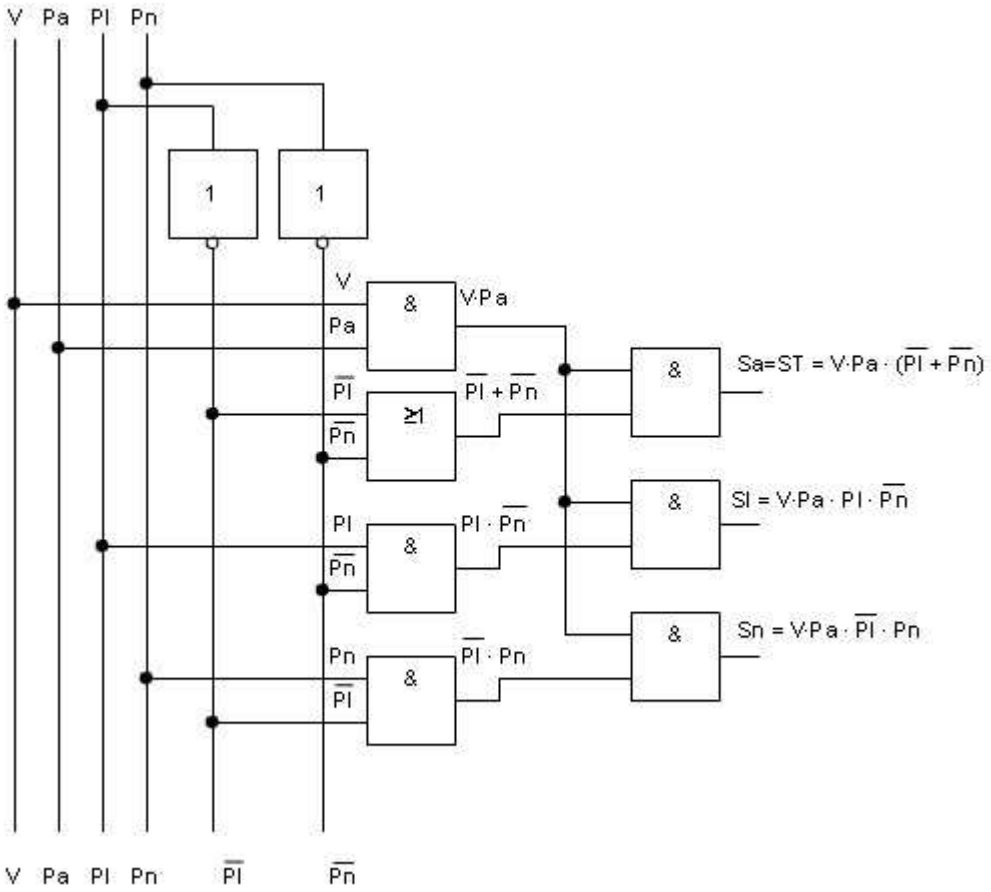
El resto de variables no se pueden simplificar puesto que sólo tienen un término en el que vale "1".

$$Sl = V \cdot Pa \cdot Pl \cdot \overline{Pn}$$

$$Sn = V \cdot Pa \cdot \overline{Pl} \cdot Pn$$

4.- Implementar la función.

Cuando la implementamos podemos aprovechar una parte de la función si se puede para las otras. Por ejemplo V·Pa es común a todas.



Implementación con puertas de todo tipo.

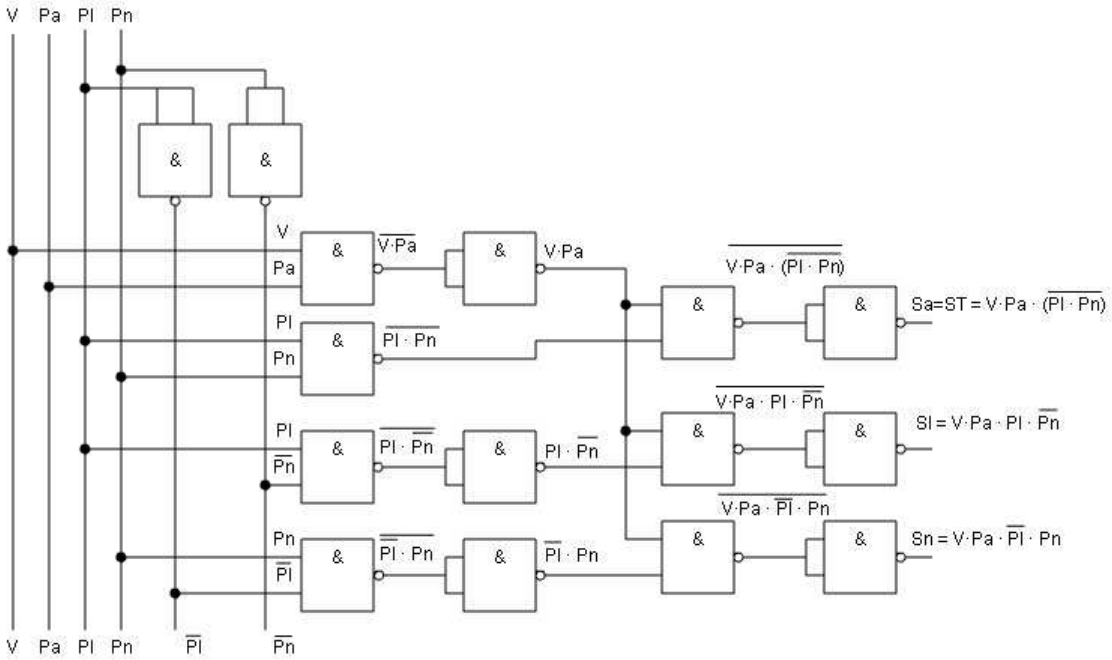
Para implementarla necesitaríamos un circuito 7404 (2 puertas inversoras), dos circuitos 7408 (6 puertas AND) y un circuito 7432 (1 puerta OR). Total 9 puertas en 4 CI.

Ahora con puertas NAND, las funciones quedarán:

$$ST = Sa = V \cdot Pa \cdot (\overline{Pl \cdot Pn})$$

$$Sl = V \cdot Pa \cdot Pl \cdot \overline{Pn}$$

$$Sn = V \cdot Pa \cdot \overline{Pl} \cdot Pn$$



Implementación con puertas NAND.

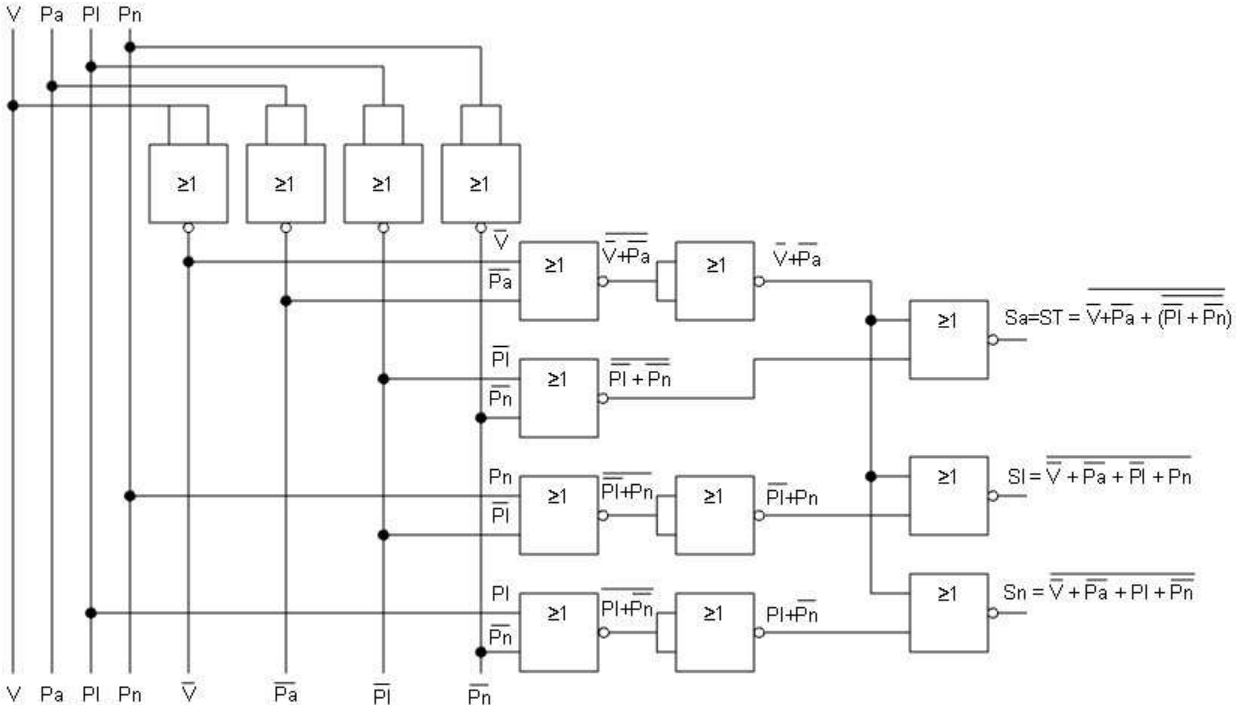
Para implementarla necesitaríamos cuatro circuitos 7400 (15 puertas NAND). No se observa ahorro ni se mejora la velocidad respecto del de todo tipo de puertas.

Ahora con puertas NOR, las funciones quedarán:

$$ST = Sa = \overline{\overline{V} + \overline{Pa} + \overline{(\overline{Pl} + \overline{Pn})}}$$

$$Sl = \overline{\overline{V} + \overline{Pa} + \overline{Pl} + Pn}$$

$$Sn = \overline{\overline{V} + \overline{Pa} + Pl + \overline{Pn}}$$



Implementación con puertas NOR.

Para implementarla necesitaríamos cuatro circuitos 7402 (14 puertas NOR). No se observa ahorro ni se

mejora la velocidad respecto del de todo tipo de puertas.

Se montaría la implementación con puertas de todo tipo, por ser la más rápida. Por utilizar sólo 9 puertas frente a las 14 de NOR o 15 de NAND, su consumo en funcionamiento también será menor.

Regresar al índice

9.- Actividades.

1.- Pasa los siguientes números hexadecimales a binario y a decimal.

FF23, 9A0, 451, CCC

2.- Dados los números de la tabla en la base que indica arriba, pásalos al resto de bases.

Decimal	Binario	Hexadecimal
6243		
	1001101101	
		3F2
763		

3.- Dibuja los símbolos de las siguientes puertas:

AND, OR, Inversora, NAND, NOR, OR-exclusiva.

4.- ¿Cuál será el símbolo de la puerta NOR-exclusiva?

5.- Demuestra mediante la tabla de verdad el siguiente teorema de Demorgan, $\overline{a+b} = \bar{a} \cdot \bar{b}$, para ello haz una tabla de verdad con la función de una parte del signo igual y otra con la de la otra parte y observa que es lo mismo.

6.- Implementa mediante interruptores las siguientes funciones lógicas: $S = a \cdot b + c$, $S = (a + b + c) \cdot d$

7.- Simplifica la siguiente función utilizando las propiedades del álgebra de Boole.

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

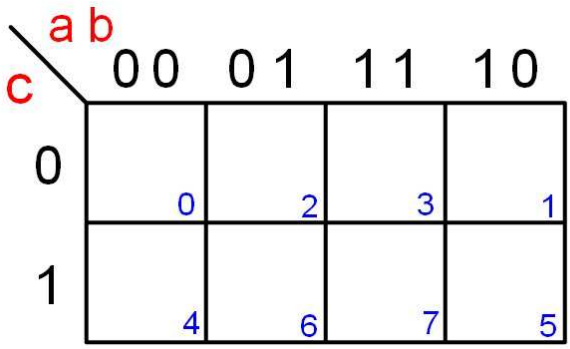
8.- Obtener la función que hay en la siguiente tabla de verdad.

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

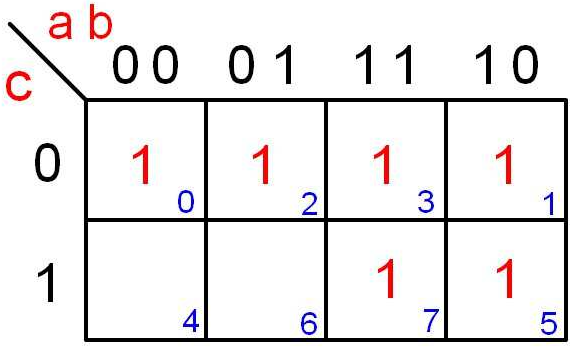
9.- Simplifica la función anterior utilizando las propiedades del álgebra de Boole.

10.- Dibuja un mapa de Karnaugh de cuatro variables.

11.- Simplifica la función de la actividad 8 con ayuda del mapa de Karnaugh.



12.- Obtén la función simplificada que aparece en el siguiente mapa de Karnaugh.



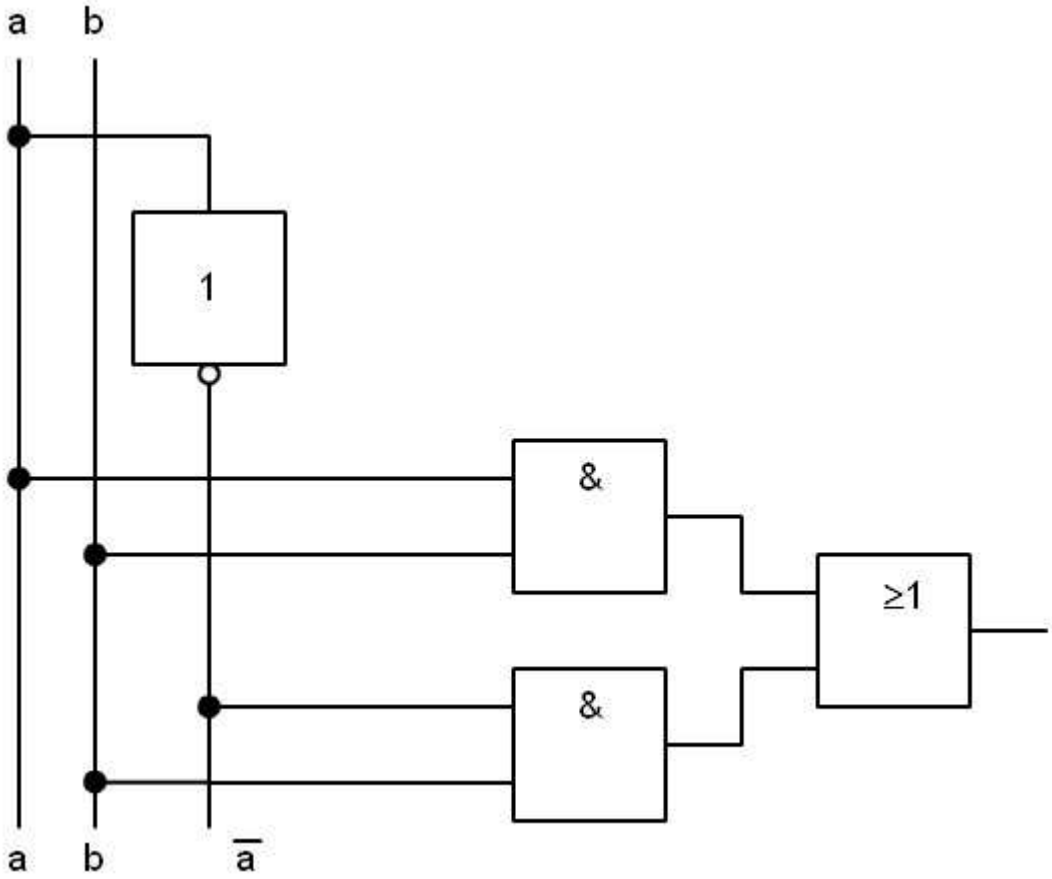
13.- Haz la tabla de verdad del mapa de Karnaugh anterior.

14.- La siguiente función está preparada para ser implementada con un tipo concreto de puertas. ¿Cuál?

$S = \overline{\overline{a \cdot b} \cdot \overline{\overline{c} \cdot d} \cdot \overline{c \cdot d} \cdot \overline{\overline{a} \cdot b}}$

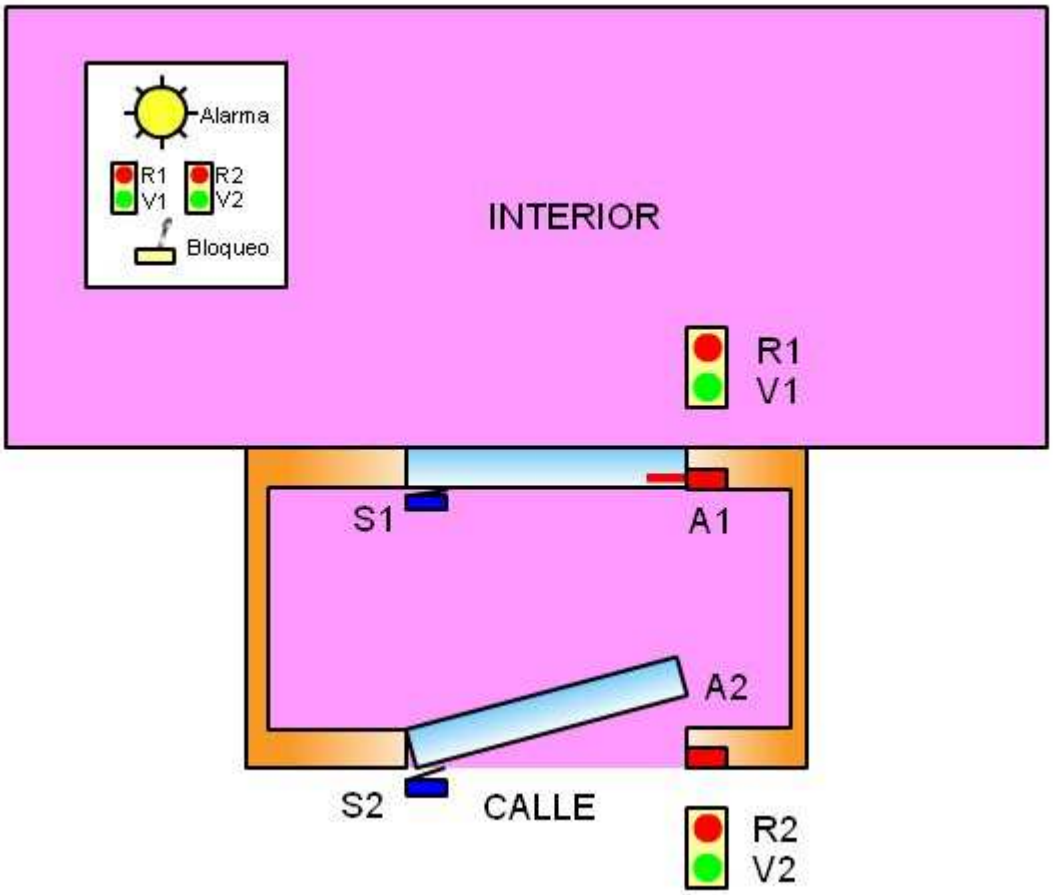
15.- Dada la siguiente función exprésala sólo en sumas (NOR) y después sólo en productos (NAND). $S = a \cdot \overline{b} \cdot c + \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot \overline{c} + a \cdot b \cdot c$. Si lo consideras necesario simplifícala.

16.- Escribe la función que es implementada por el esquema siguiente:



17.- Dada la siguiente función impleméntala con puertas de todo tipo, sólo con puertas NAND y sólo con puertas NOR. $S = a \cdot \overline{b} + \overline{a} \cdot c + a \cdot b \cdot c$ Indica cuál es que se debe montar de todos.

18.- El gráfico siguiente muestra las puertas de entrada de un banco.



Las puertas están provistas de anclajes de seguridad (A1, A2) y de sensores (S1,S2) que indican si están abiertas o cerradas. Así como de un semáforo que indica si se permite o no el paso (R1,V1) y (R2,V2).

Cuando se abre una de las puertas se debe cerrar el anclaje de la otra, y encender las luces de los semáforos de manera que impida el paso a las personas que intentan entrar por la otra puerta.

Si se produce el caso indeseado de que se abran las dos puertas a la vez se debe indicar con una luz de alarma al cajero. Y no deben activarse los anclajes.

El cajero tiene un mando donde se visualiza el estado de las puertas y un interruptor que las bloquea cuando están cerradas.

Diseña el sistema que resuelve el problema con puertas de todo tipo, NAND y NOR, he indica cuál es el que debemos montar.

Regresar al índice