

Espacio curricular: "Educación Tecnológica" Profesor: GIMENEZ DARIO ALEJANDRO Cursos: 4° C.O. División 2ª

ELECTRÓNICA DIGITAL.

4.1. INTRODUCCIÓN. SEÑALES ANALÓGICAS Y DIGITALES.

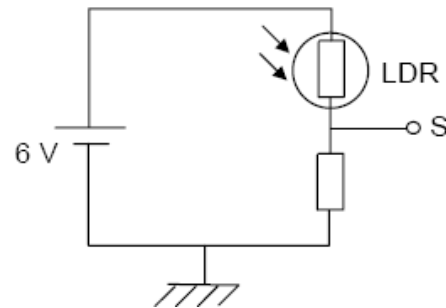
Podemos dividir la electrónica en dos grandes campos: la electrónica analógica y la electrónica digital, según el tipo de señales que utilice.

Llamamos **señal**, a la variación de una magnitud que permite transmitir información. Las señales pueden ser de dos tipos:

Señales analógicas: son las señales que varían de forma continua en el tiempo entre dos valores

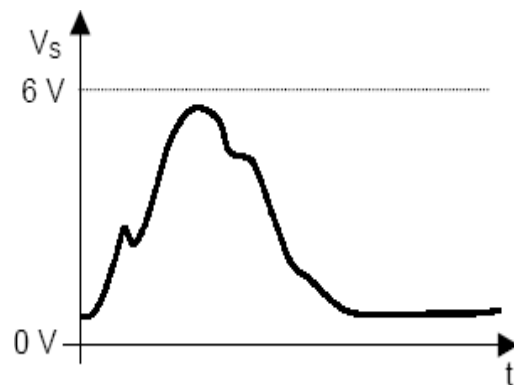
Señales digitales: son señales que pueden adoptar sólo algunos valores concretos.

Ejemplo: Supongamos un circuito formado por una LDR, como el de la figura. Consideramos como señal de salida del circuito la tensión en el punto S.



Vamos a exponer la LDR a dos situaciones diferentes:

a) Colocamos la LDR al aire libre, expuesta a luz natural. Esta luz irá variando a lo largo del día, y tendrá variaciones debido, por ejemplo, a la ocultación temporal del sol por el paso de alguna nube. Si representamos en un gráfico la variación de la tensión en el punto S (con respecto a masa) a lo largo del tiempo, obtendremos una curva similar a la de la figura: Se observa que la tensión varía de forma continua y toma todos los valores intermedios entre los valores máximo y mínimo. Se trata de una señal analógica.



b) Colocamos la LDR en un habitáculo cerrado (sin luz natural) junto a un foco luminoso. A continuación encendemos y apagamos el foco varias veces según nos parezca. La variación de la tensión en el punto S adoptará ahora una forma bien distinta:

concretos, un valor bajo cuando el foco está apagado y un valor alto cuando el foco está encendido. Se trata de una señal digital.

Hoy en día, con la creciente complejidad de los procesos industriales y de los elementos necesarios para su control, se observa que la tensión varía de forma discontinua, adoptando únicamente dos valores, los grandes volúmenes de información que es necesario tratar, la revolución de las comunicaciones, etc, se hacen imprescindibles métodos de control electrónico cada vez más sofisticados. En este contexto, **las señales digitales presentan importantes ventajas frente a las analógicas**, como son su mayor inmunidad a las interferencias, mayor simplicidad de tratamiento, economía de circuitos, etc.

En electrónica digital se utilizan señales que pueden adoptar únicamente dos valores bien diferenciados. Por ello, estas señales se denominan **señales binarias**.

Los circuitos digitales estarán compuestos por dispositivos capaces de distinguir y de generar señales binarias; como veremos, los dispositivos electrónicos digitales más básicos, y a partir de los cuales están constituidos todos los demás, se denominan **puertas lógicas**.

4.2. SISTEMA DE NUMERACIÓN BINARIO.

El sistema de numeración de la vida cotidiana es el **sistema decimal**, que utiliza diez signos (de 0 a 9). Codificando adecuadamente estos diez signos podemos representar cualquier número, realizar operaciones con ellos y, en definitiva, representar y transmitir cualquier tipo de información. Los circuitos digitales utilizan para su trabajo el sistema de numeración binario, que utiliza únicamente dos signos, el 0 y el 1. A cada uno de estos símbolos se le denomina **bit**.

El sistema binario es de base 2, solamente utiliza dos valores el 0 y el 1. Por ejemplo:

$$110101 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 53$$

4.2.1. PASO DE SISTEMA BINARIO A DECIMAL Y VICEVERSA.

Para pasar un número en sistema binario a su equivalente en sistema decimal se expresa el número binario por su polinomio equivalente de potencias de dos y se suman sus términos.

Ejemplo: Pasar 110101 a decimal

$$110101 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 32 + 16 + 0 + 4 + 0 + 1 = 53$$

Para pasar un número en sistema decimal a su equivalente binario se realizan sucesivas divisiones por dos hasta que el último cociente sea 1. El número binario estará formado por un 1 seguido de los restos ordenados de las sucesivas divisiones. El orden de colocación viene determinado por la siguiente regla: "el resto de la primera división corresponde al bit menos significativo (el situado más a la derecha)".

Ejemplo: Pasar 26 a binario.

División	Cociente	Resto	
26 : 2	13	0	
13 : 2	6	1	
6 : 2	3	0	
3 : 2	1	1	
			1 1 0 1 0

4.2.2. OTROS CÓDIGOS BINARIOS.

El código que hemos visto se denomina **código binario natural**, pero existen otros códigos binarios.

Uno de los más utilizados es el **código BCD** (Decimal Codificado en Binario). Para representar un número decimal en BCD, se representa por separado cada una de sus cifras en código binario natural. El número de bits necesarios para representar cada cifra es de cuatro.

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Ejemplo: Representar 348 en BCD

348 = 0011 0100 1000

El código BCD que hemos descrito se denomina BCD natural, existen otros códigos BCD pero que no veremos.

4.3. EL ÁLGEBRA DE BOOLE

Como hemos dicho, los circuitos digitales operan con señales binarias, de forma que sólo distinguen entre dos valores de tensión: nivel alto y nivel bajo. Los niveles de tensión dependerán de la tecnología utilizada. Por ejemplo, con los dispositivos de tecnología TTL, el nivel alto es 5 V y el nivel bajo 0 V. Para la codificación binaria de las señales, al nivel alto se le asigna el 1 y al nivel bajo el 0 (aunque puede ser al contrario). Ahora bien, los circuitos digitales deben realizar a menudo operaciones de gran complejidad, de forma que el diseño del circuito no es simple. Es necesaria una herramienta matemática útil para abordar el diseño de estos circuitos. Dicha herramienta es el álgebra de Boole.

El **álgebra de Boole** es aplicable a variables que sólo admiten dos valores posibles, que se designan por 0 y 1. Estos símbolos no representan números, sino dos estados diferentes de un dispositivo. Por ejemplo, una lámpara puede estar encendida (1) o apagada (0), un interruptor o un pulsador pueden estar cerrados (1) o abiertos (0).

4.3.1. FUNCIÓN LÓGICA Y TABLA DE VERDAD.

Llamamos **función lógica** a toda variable binaria cuyo valor depende de una expresión matemática formada por otras variables binarias relacionadas entre sí por las operaciones $+$ (más) y $\cdot$ (por). A la función lógica se le denomina variable dependiente y a las variables que forman la expresión matemática se les denomina variables independientes.

Ejemplo: la función $S = a + b \cdot c$

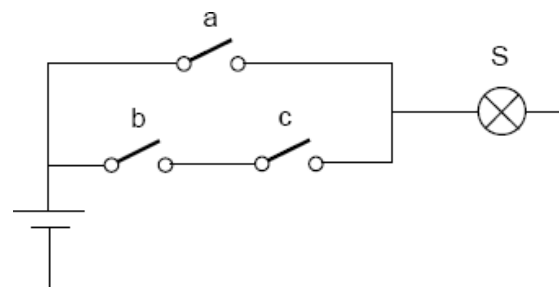
Esta expresión se interpreta como *"la variable **S** vale 1 cuando la variable **a** vale 1 o las variables **b** y **c** valen 1"*. S es la variable dependiente y a, b y c son las variables independientes.

Podemos verlo más fácilmente con una analogía eléctrica. Supongamos el siguiente circuito:

Definimos la función S como el estado de la lámpara: encendido (1) o apagado (0).

La variable "a" es el estado del interruptor "a": abierto (0) o cerrado (1).

Las variables "b" y "c" se definen igual que la "a".



En efecto, podemos observar que la lámpara estará encendida ($S = 1$) cuando "a" esté cerrado ($a = 1$) o bien "b" y "c" estén cerrados simultáneamente ($b = 1$ y $c = 1$).

Las funciones lógicas se representan mediante las llamadas **tablas de verdad**, en las cuales se indican los valores que adopta la función lógica ante todas y cada una de las combinaciones de valores de las variables independientes. Si tenemos n variables independientes, tendremos 2^n combinaciones posibles.

La tabla de verdad de la función $S = a + b \cdot c$ es:

La tabla tiene dos partes, las columnas de la izquierda corresponden a las variables independientes o variables de entrada. La columna de la derecha corresponde a la variable dependiente o variable de salida.

Cada fila de la tabla representa una combinación posible de las variables de entrada, y el correspondiente valor que adopta la variable de salida.

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

Con "n" variables de entrada pueden darse 2^n combinaciones diferentes.

4.3.2. OPERACIONES BÁSICAS DEL ÁLGEBRA DE BOOLE.

Se definen tres operaciones básicas: la suma lógica, el producto lógico y la complementación (o negación).

- **SUMA LÓGICA**

Se representa por el signo $+$. Si tenemos dos variables de entrada **a** y **b**, su suma lógica se representa por:

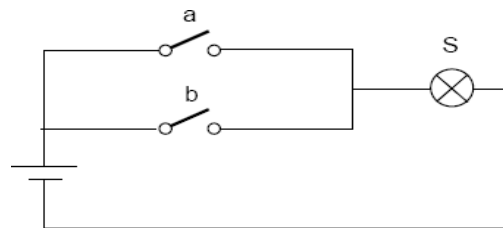
$$S = a + b$$

"la suma lógica vale 1 cuando alguna de las variables de entrada vale 1".

Para dos variables, su tabla de verdad es:

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

El circuito eléctrico equivalente es:



Los circuitos electrónicos que realizan esta operación lógica se denominan **puertas lógicas OR**. El símbolo que se emplea puede ser de dos tipos dependiendo de las normas que se empleen.



- **PRODUCTO LÓGICO**

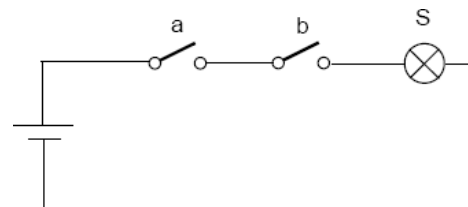
Se representa por el signo $\cdot$. Si tenemos dos variables de entrada **a** y **b**, su producto lógico se representa por: **$S = a \cdot b$**

"el producto lógico vale 1 cuando todas las variables de entrada valen 1". Para dos variables, su tabla de verdad es:

Para dos variables, su tabla de verdad es:

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

El circuito eléctrico equivalente es:



Los circuitos electrónicos que realizan esta operación lógica se denominan **puertas lógicas AND**. El símbolo que se emplea depende de la norma empleada:



• COMPLEMENTACIÓN O NEGACIÓN.

Se aplica a una sola variable de entrada. Se representa colocando un guiñon encima del nombre de la variable. Si ésta es "a" por ejemplo, su complementación se representa por " $\bar{a}$ " (se lee a negada).

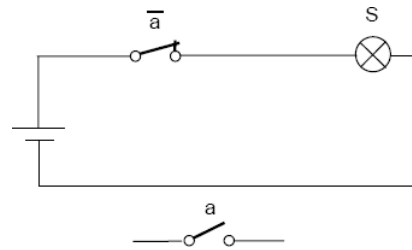
$$S = \bar{a}$$

Si a = 0 entonces S = 1, si a = 1 entonces S = 0.

Su tabla de verdad es:

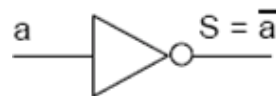
a	S
0	1
1	0

El circuito eléctrico equivalente es:



El contacto " $\bar{a}$ " es complementario del "a" de forma que cuando éste último está abierto el primero está cerrado y viceversa.

El circuito electrónico que realiza la operación lógica de complementación se denomina **inversor** o **puerta NOT**. Los símbolos empleados son:



4.3.3. PROPIEDADES DEL ÁLGEBRA DE BOOLE.

Estas propiedades y teoremas son muy importantes para simplificar las funciones lógicas.

$$a + 1 = 1$$

$$a \cdot 1 = a$$

$$a + 0 = a$$

$$a \cdot 0 = 0$$

$$a + a = a$$

$$a \cdot a = a$$

$$a + \bar{a} = 1$$

$$a \cdot \bar{a} = 0$$

$$\bar{\bar{a}} = a$$

Propiedad conmutativa:

$$a + b = b + a$$

$$a \cdot b = b \cdot a$$

Propiedad distributiva:

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

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

Propiedad asociativa:

$$(a + b) + c = a + (b + c) \quad (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

Teoremas de absorción

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

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

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

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

Teoremas de Morgan

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

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

4.3.4. PROPIEDADES DEL ÁLGEBRA DE BOOLE.

Aparte de las puertas anteriores, que realizan las operaciones básicas del álgebra de Boole, existen otras puertas que realizan **funciones lógicas especiales** porque resultan de la combinación de dos o más funciones simples. Estas puertas son las siguientes:

- **Puerta NOR**

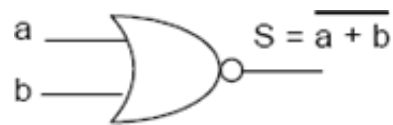
Realiza la suma lógica negada (Función NO OR, o abreviadamente función NOR). La expresión matemática para dos variables es:

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

La tabla de verdad de la función NOR es:

Su símbolo, depende de la norma:

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



- **Puerta NAND**

Realiza el producto lógico negado (Función NO AND, o abreviadamente función NAND).

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

La tabla de verdad de la función NAND es:

Su símbolo, como antes, depende de la norma:

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



- **Puerta OR EXCLUSIVA :**

También llamada **puerta EXOR**. Sólo existe para dos entradas. Presenta a su salida el valor lógico 1 cuando las variables de entrada presentan valores diferentes, y presenta el valor lógico 0 cuando los valores de las variables de entrada coinciden. Se representa por:

$$S = a \oplus b$$

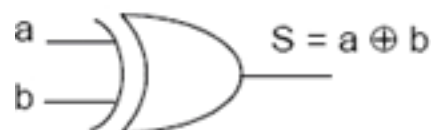
y equivale a:

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

La tabla de verdad de la función EXOR es:

Sus símbolos son:

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



- **Puerta NOR EXCLUSIVA**

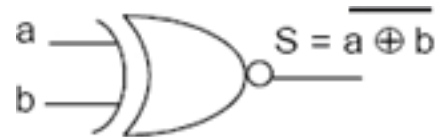
También llamada **puerta EXNOR**. Sólo existe para dos variables. Presenta a su salida el valor lógico 1 cuando los valores de las dos variables de entrada coinciden, y presenta el valor lógico 0 cuando los valores de las variables de entrada son diferentes. Se representa por:

$$S = a \oplus b$$

y equivale a: $S = a \cdot b + \bar{a} \cdot \bar{b}$

La tabla de verdad de la función EXNOR es: Sus símbolos son:

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



4.4. DISEÑO DE CIRCUITOS DE PUERTAS LÓGICAS.

El método más simple, cuando el número de variables de entrada no es grande, consiste en obtener la tabla de verdad de la función lógica a partir de las condiciones físicas de funcionamiento del circuito que quiero diseñar.

Después obtendremos la función lógica a partir de dicha tabla de verdad y por último se simplifica esta función lógica.

Ejemplo 1: Disponemos de tres finales de carrera, "a" "b" y "c" para el gobierno de tres motores, M1, M2 y M3, según las siguientes condiciones:

- No estando accionado ningún final de carrera, permanecerán parados los tres motores.
- Estando pulsado sólo "a" debe girar M1.
- Estando pulsado sólo "b" debe girar M2.
- Estando pulsado sólo "c" debe girar M3.
- Accionando dos finales de carrera cualesquiera, girarán los tres motores.
- Mientras se encuentren accionados los tres finales de carrera, no deberá girar ningún motor.

La tabla de verdad del circuito de control del sistema es:

a	b	c	M1	M2	M3
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	0	0	0

Términos de indiferencia

Hasta ahora hemos supuesto que cada combinación de entradas a un circuito lógico ha de dar una salida o bien 0 o bien 1. Sin embargo, a veces sucede que algunas de dichas combinaciones de entrada no podrán darse físicamente debido a las características del sistema que se pretende controlar con el circuito lógico.

Pensemos, por ejemplo, en el circuito para controlar el movimiento de un ascensor, y que algunas de las variables de entrada son finales de carrera que detectan la planta el edificio en la que se encuentra el ascensor. Resulta evidente que no podrán estar activados al mismo tiempo el final de carrera de la 1ª planta y el de la 3ª.

A estos términos se les llama términos de indiferencia, y da lo mismo que la salida del circuito lógico sea 0 ó 1, ya que de hecho no se va a dar este caso (evidentemente salvo averías). Estos términos se representan mediante una "x" o un guión "-" en la tabla de verdad, y, como veremos luego, pueden ser bastante interesantes de cara a simplificar el circuito lógico.

4.4.1. OBTENCIÓN DE LA FUNCIÓN LÓGICA A PARTIR DE LA TABLA DE VERDAD.

Para obtener la función lógica se suman todos los productos lógicos correspondientes a las combinaciones que dan salida 1, asignando al valor 1 la variable en estado normal y al valor 0 la variable en estado complementada.

Ejemplo 1: veamos la función lógica correspondiente a cada uno de los motores del ejemplo 1 anterior:

$$\begin{aligned}M1 &= \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c} \\M2 &= \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c} \\M3 &= \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}\end{aligned}$$

4.4.3. SIMPLIFICACIÓN DE FUNCIONES LÓGICAS

El diseñador debe intentar simplificar lo más posible la función lógica obtenida a partir de la tabla de verdad, con objeto de reducir el coste, ocupar menos espacio y aumentar la fiabilidad del circuito.

Normalmente, lo que se hace es intentar obtener una función lógica equivalente a la anterior, con el menor número de términos posible y cada término con el menor número de variables posible.

Existen diversos métodos. Veamos dos de ellos:

SIMPLIFICACIÓN POR EL MÉTODO ALGEBRAICO

Consiste en utilizar las propiedades y teoremas del álgebra de Boole que hemos visto para agrupar y simplificar los términos de la función lógica. No es un método sistemático y no resulta muy útil cuando la función es compleja. Además, tampoco tenemos garantía de que el resultado obtenido sea la expresión mínima. Requiere, además cierta soltura con las matemáticas.

MÉTODO GRÁFICO DE KARNAUGH

A diferencia del método anterior, el método de Karnaugh asegura obtener la expresión irreducible mínima de una función lógica.

El fundamento del método de Karnaugh consiste en reducir a un solo término grupos de 2, 4, 8,términos adyacentes.

Para aplicar el método, a partir de la tabla de verdad se construye otra tabla llamada **tabla de karnaugh**, cuyo número de casillas es el mismo que tiene la tabla de verdad, que como sabemos depende del número de variables de entrada que tenga la función que se quiere simplificar.

Así, para n variables tendrá 2^n casillas.

La forma de las tablas para 2, 3 y 4 variables es:

a \ b	0	1
0		
1		

a \ b	00	01	11	10
0				
1				

a \ b	00	01	11	10
00				
01				
11				
10				

Es **importante** establecer correctamente el orden de numeración de las casillas. Obsérvese que están numeradas de forma que dos casillas contiguas corresponden a términos adyacentes, es decir, entre dos casillas contiguas, sólo una de las variables cambia de valor.

Las **relaciones de adyacencia** en las tablas de Karnaugh son las siguientes:

- En la tabla de **dos variables** son adyacentes las casillas contiguas (un lado común).
- En la tabla de **tres variables** son adyacentes tanto las casillas contiguas como las casillas de la primera y última columna (es como si la tabla fuera el desarrollo de un cilindro).
- En la tabla de **cuatro variables** son adyacentes, además de las anteriores, las de la fila superior con las de la fila inferior (siendo de la misma columna).

Veamos el **procedimiento del método de Karnaugh**:

1.- Desde la tabla de verdad, se trasladan a la tabla de Karnaugh los valores que adopta la variable de salida cuya función lógica se quiere simplificar.

2.- Agrupamientos de "1". Para que la función lógica quede lo más reducida posible nos conviene realizar el mínimo de agrupamientos de "1" y con el mayor número de casillas posible. Procedemos de la siguiente forma:

- Se toman todos los "1" que no se pueden agrupar con ningún otro.
- Se forman los grupos de dos "1" que no pueden formar un grupo de cuatro.
- Se forman los grupos de cuatro "1" que no pueden formar un grupo de ocho.

Al hacer agrupamientos no hay ningún problema en que una casilla pertenezca a más de un agrupamiento simultáneamente.

Los agrupamientos conseguidos y los "1" aislados serán los términos que expresarán la función lógica en forma irreducible.

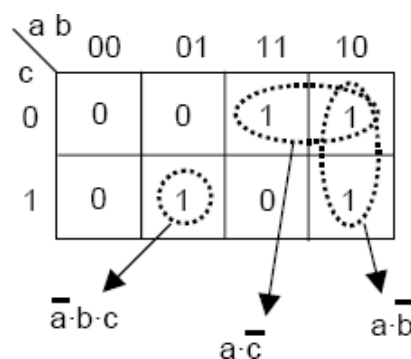
Podemos observar que agrupando 2^n "1" adyacentes, eliminamos n variables en el término que representa al agrupamiento. En los "1" aislados no se elimina ninguna variable.

Ejemplo 1: Sea el caso ya visto en un ejemplo anterior de los tres motores gobernados por tres finales de carrera, cuya tabla de verdad era:

a	b	c	M1	M2	M3
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	0	0	0

Como tenemos tres variables de entrada, usamos la tabla de Karnaugh de tres variables:

Empezamos con el **motor M1**: $M1 = \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$



El "1" aislado no permite reducir variables. Se observa que corresponde a los valores $a = 0$, $b = 1$ y $c = 1$. Para expresar este término de forma algebraica se asigna estado normal a las variables que valen 1 y estado complementario a las variables que valen 0. Por ello es: $\bar{a} \cdot b \cdot c$

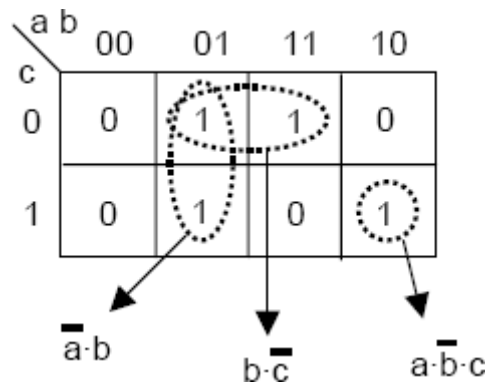
Las casillas del agrupamiento de dos "1" de la fila superior tienen en común que $a = 1$ y $c = 0$; sin embargo, b no coincide. Esto indica que b es la variable que se puede eliminar. Queda: $a \cdot \bar{c}$

Las casillas del agrupamiento de dos "1" de la última columna tienen en común que $a = 1$ y $b = 0$; ahora es c la que no coincide, lo que indica que se elimina. Queda: $a \cdot \bar{b}$

En definitiva:

$$M1 = \bar{a} \cdot b \cdot c + a \cdot \bar{b} + a \cdot \bar{c}$$

La simplificación de la función del motor M2 es: $M2 = \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$



Queda:

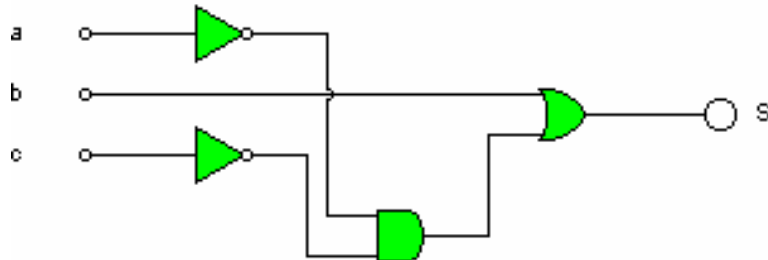
$$M2 = a \cdot \bar{b} \cdot c + \bar{a} \cdot b + b \cdot \bar{c}$$

4.4.4. ESQUEMA DEL CIRCUITO A PARTIR DE LA FUNCIÓN LÓGICA.

Una vez que tenemos la función lógica ya simplificada, procedemos a implementarla con puertas lógicas. Veamos algunos ejemplos:

Ejemplo A

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

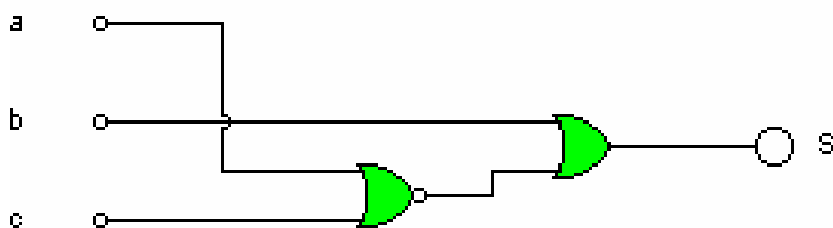


Sin embargo, podemos tener en cuenta que según uno de los teoremas de Morgan:

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

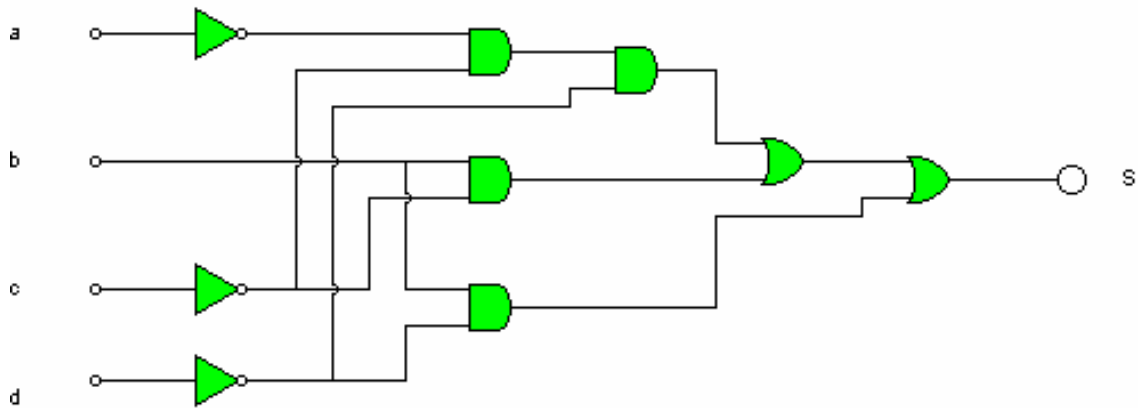
con lo que queda mucho más simple usando una puerta NOR.

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



Ejemplo B

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

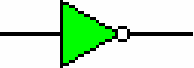
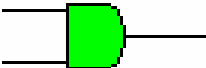
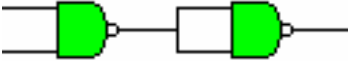
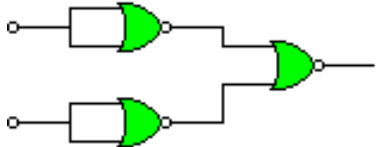
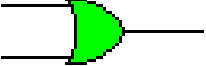
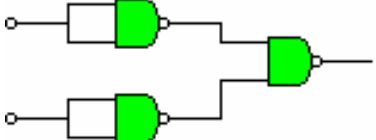
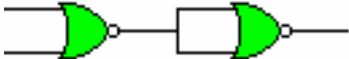


4.4.5.IMPLEMENTACIÓN DE PUERTAS LÓGICAS CON PUERTAS NAND Y NOR.

De cara a la realización física del circuito electrónico con puertas lógicas, puede resultar interesante tener en cuenta que cualquier puerta lógica se puede construir con puertas NAND o con puertas NOR. Por ello a estas puertas, se les llama **puertas universales**.

Esto es interesante, primero porque el coste de los circuitos con puertas NAND es más bajo que con otras puertas, y segundo, porque si necesitamos para completar el diseño una sola puerta de cualquier tipo, no merece la pena colocar un nuevo circuito integrado, desperdiciando el resto de puertas que contenga, cuando puede que nos sobren puertas NAND o NOR en otro integrado.

En la tabla se muestra la forma de realizar las funciones básicas con puertas NAND y NOR:

FUNCIÓN	Con puertas NAND	Con puertas NOR
		
		
		

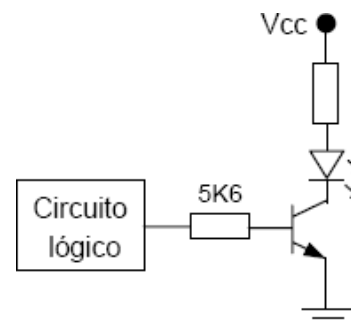
4.5. CONEXIÓN DE LA SALIDA DEL CIRCUITO LÓGICO A OTROS CIRCUITOS.

Por los circuitos constituidos por componentes electrónicos digitales circulan intensidades de corriente muy pequeñas. De hecho, aunque depende del tipo de tecnología, la salida de una puerta lógica no puede dar más de allá de unos pocos mA de corriente. Concretamente, con la tecnología LS TTL, que es una de las más habituales, la corriente de salida es de 8 mA, y en tecnología CMOS, también bastante utilizada, es aún menor, de unos 2 mA.

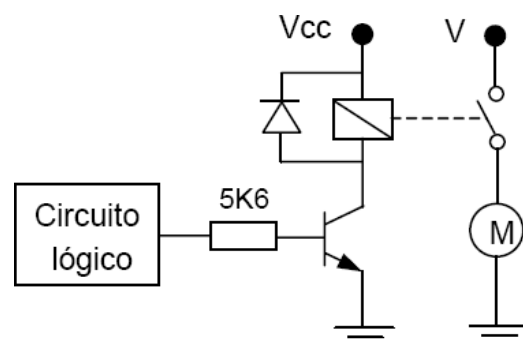
Todo lo anterior nos indica que nosotros, en ningún caso podemos conectar a la salida de un circuito lógico sin más, el receptor que queramos controlar, como puede ser un motor, una lámpara o un relé, ya que todos estos elementos consumen una corriente muy superior a la que el circuito lógico puede dar.

La forma más sencilla de resolver este problema es que la salida del circuito lógico se conecte a la base de un transistor o de un par Darlington, interponiendo una resistencia adecuada para limitar la salida de corriente. Para la conexión del receptor que queramos controlar tenemos dos posibilidades:

a) Si el receptor requiere una pequeña tensión continua y su consumo de corriente es bajo, se puede conectar directamente al colector del transistor (por ejemplo, un led o un zumbador).



b) Si el receptor requiere una tensión elevada o tiene mayor consumo, como pueden ser lámparas de incandescencia, motores, etc, es conveniente conectar la bobina de excitación de un relé al colector del transistor y que sean los contactos del relé los que activen el receptor.



Ejemplo 2: Sea un sencillo montacargas que se mueve entre dos plantas, que llamaremos "baja" y "alta". Dispone de dos interruptores, "s" y "b" para ordenarle que suba o baje respectivamente, que ofrecen un nivel lógico 1 cuando se accionan. Además dispone de dos finales de carrera, uno en la planta baja, "Fcb" y otro en la planta alta "Fca" que se activan, dando lugar a un nivel lógico 1, cuando el montacargas se posiciona justamente en su planta respectiva. El circuito ofrecerá dos salidas, una, llamada "Ms" que al activarse con un valor lógico 1 hará que se ponga en marcha un motor que hará que el montacargas suba, y otra, llamada "Mb" que al activarse con un valor lógico 1 hará que el motor gire en sentido contrario y el montacargas baje.

Las condiciones de funcionamiento son:

- Si se activa el interruptor "s" y el montacargas no está en la planta alta, el montacargas sube.
- Si se activa el interruptor "b" y el montacargas no está en la planta baja, el montacargas baja.
- El montacargas estará parado tanto si no están activos ni "s" ni "b" como si lo están ambos simultáneamente.

Tenemos un sistema con cuatro variables de entrada ("s", "b", "Fcb", "Fca") y dos variables de salida ("Ms" y "Mb"), cada una de las cuales tendrá su función lógica.

Con cuatro variables de entrada pueden darse $2^4 = 16$ combinaciones diferentes, pero tendremos en cuenta que, salvo averías, las señales "Fcb" y "Fca" no pueden estar activas simultáneamente, por lo que la salida en estos casos es indiferente. Completa la tabla de verdad y completa el diseño.

Fcb	Fca	s	b	Ms	Mb
0	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	0	0	0		
1	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		

FCb	FCa	s	b	Ms	Mb
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	x	x
1	1	0	1	x	x
1	1	1	0	x	x
1	1	1	1	x	x

Obtención de las funciones: veamos ahora las funciones lógicas correspondientes a las salidas Ms y Mb del ejemplo 2 anterior:

$$M_s = \overline{FCb} \cdot \overline{FCa} \cdot s \cdot \overline{b} + \overline{FCb} \cdot \overline{FCa} \cdot s \cdot b$$

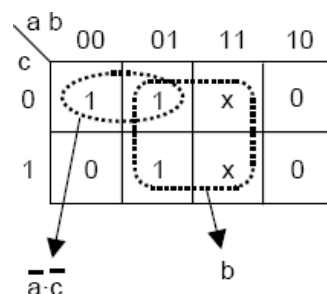
$$M_b = \overline{FCb} \cdot \overline{FCa} \cdot \overline{s} \cdot b + \overline{FCb} \cdot \overline{FCa} \cdot s \cdot b$$

Ejemplo 3 simplificación de la función por Karnaugh: Sea un sistema cuya tabla de verdad es la siguiente:

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

Observamos que hay dos combinaciones de entradas cuya salida es indiferente. Esto es debido a que, por las características físicas del sistema que se quiere controlar, las variables a y b no pueden estar activas simultáneamente (recordar el ejemplo del montacargas que no puede estar en dos plantas al mismo tiempo).

Vamos a simplificar la función lógica por el método de Karnaugh.



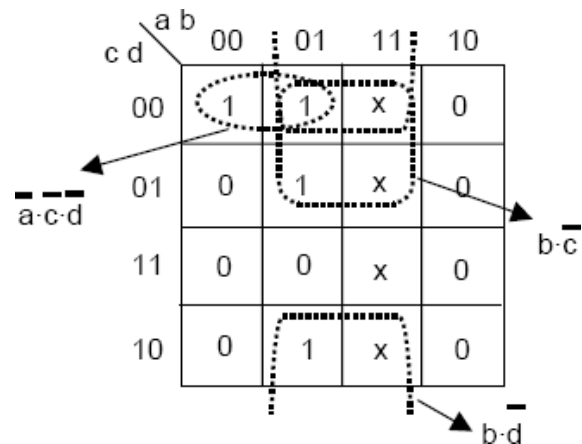
Hemos tomado las dos casillas de términos indiferentes como "1" ya que de esta forma puedo formar un agrupamiento de cuatro casillas, que es más conveniente que uno de dos casillas.

Me queda, por tanto:

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

Ejemplo 4 simplificación de la función por Karnaugh: Sea el sistema cuya tabla de verdad se da a continuación:

a	b	c	d	S
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x



Vamos a simplificar por el método de Karnaugh: Tras realizar los agrupamientos que se indican en la tabla de karnaugh siguiente, nos queda:

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