Electrónica Digital

(4ºESO)

1. INTRODUCCIÓN

1. SEÑALES Y TIPOS

Como vimos en el tema anterior, la electrónica es la rama de la ciencia que se ocupa del estudio de los circuitos y de sus componentes que permiten modificar la corriente eléctrica amplificándola, atenuándola, rectificándola y filtrándola y que aplica la electricidad al tratamiento de la información. Por otro lado el término digital deriva de la forma en que las computadoras realizan las operaciones; i.e. contando dígitos o números.

Una **señal** es la variación de una magnitud que permite transmitir información. Las señales pueden ser de dos tipos:

- **Señales analógicas:** aquellas donde la señal puede adquirir infinitos valores entre dos extremos cualesquiera. La variación de la señal forma una gráfica continua. La mayoría de las magnitudes en la naturaleza toman valores continuos, por ejemplo la temperatura. Para pasar de 20 a 25°C, la temperatura irá tomando los infinitos valores entre 20 y 25°C.

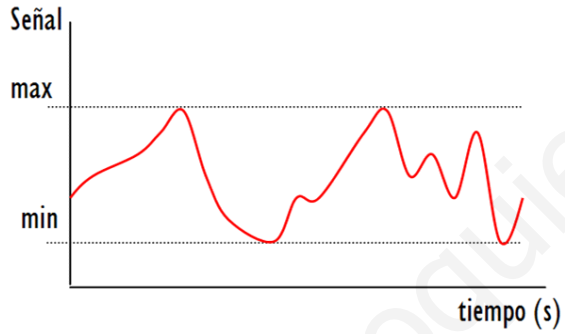
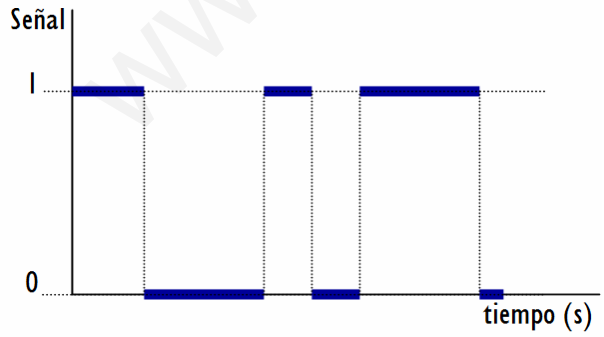


Fig 1: Ejemplo de señal analógica.

- **Señales digitales:** las cuales pueden adquirir únicamente valores concretos; i.e. no varían de manera continua.

Fig 2: Ejemplo de señal digital.



Para nosotros los sistemas digitales que tienen mayor interés, por ser los que se pueden implementar electrónicamente, son los sistemas binarios. Un **sistema binario** es aquel en el que las señales sólo pueden tomar dos valores, que representaremos de ahora en adelante con los símbolos 0 y 1. Por ejemplo , el estado de una bombilla sólo puede tener dos valores (0 apagada, 1 encendida). A cada valor de una señal digital se le llama **bit** y es la unidad mínima de información.

2. VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS DIGITALES

El mejor argumento a favor de la mayor flexibilidad de los sistemas digitales se encuentra en los actuales ordenadores o computadoras digitales, basados íntegramente en diseños y circuitos digitales. Las principales ventajas de los sistemas digitales respecto a los analógicos son:

- Mayor facilidad de diseño, púes las técnicas están bien establecidas.
- El ruido (fluctuaciones de tensión no deseadas) afecta menos a los datos digitales que a los analógicos), ya que en sistemas digitales sólo hay que distinguir entre valor alto y valor bajo.
- Las operaciones digitales son mucho más precisas y la transmisión de señales es más fiable porque utilizan un conjunto discreto de valores, fácil de diferenciar entre sí, lo que reduce la probabilidad de cometer errores de interpretación.
- Almacenamiento de la información menos costoso

Los sistemas digitales presentan 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).

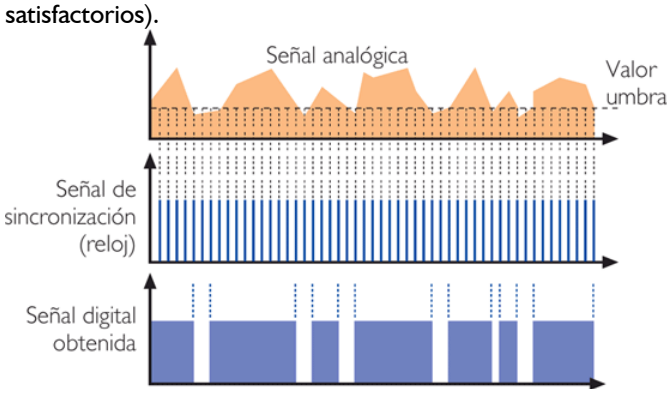


Fig 3: Conversión de señal analógica a señal digital. Si el valor de la señal en ese instante está por debajo de un determinado umbral, la señal digital toma un valor mínimo (0). Cuando la señal analógica se encuentra por encima del valor umbral, la señal digital toma un valor máximo (1).

3. TIPO DE LÓGICA

En los circuitos electrónicos digitales se emplean niveles de tensión distintos para representar los dos bits. Las tensiones

que se utilizan para representar los unos y los ceros se les denominan niveles lógicos. Existen distintos tipos de lógica

- **Lógica positiva:** al nivel alto se le da el valor de 1 y al nivel bajo un valor de 0 ($V_H = 1$ y $V_L = 0$)
- **Lógica negativa:** al nivel alto se le da el valor 0 y al nivel bajo un valor de 1 ($V_H = 1$ y $V_L = 0$).
- **Lógica mixta:** se mezclan ambos criterios en el mismo sistema, eligiendo uno u otro según convenga.

Nosotros trabajaremos con la lógica positiva.

II. Sistemas de Numeración

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.

1. SISTEMA DECIMAL

Su origen lo encontramos en la India y fue introducido en España por los árabes. Es un sistema de base 10; i.e. emplea 10 caracteres o dígitos diferentes para indicar una determinada cantidad: 0, 1, 2, 3, 4, 5, 6, 7, 8, y 9. Es un sistema posicional, de manera que el e valor de cada cifra depende de su posición dentro de la cantidad que representa.

$$2165_{10} = 2 \cdot 10^3 + 1 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0 =$$

$$= 2000 + 100 + 60 + 5$$

2. SISTEMA BINARIO

Los ordenadores y en general todos los sistemas que utilizan electrónica digital utilizan el sistema binario. En la electrónica digital sólo existen dos estados posibles (1 o 0) por lo que interesa utilizar un sistema de numeración en base 2, el sistema binario. Dicho sistema emplea únicamente dos caracteres, 0 y 1. Estos valores reciben el nombre de ***bits*** (dígitos binarios). Así, podemos decir que la cantidad 10011 está formada por 5 bits.

Al igual que en el sistema decimal, la información transportada en un mensaje binario depende de la posición de las cifras. Por ejemplo, en la notación decimal, sabemos que hay una gran diferencia entre los números 126 y 621. ¿Cómo sabemos esto? Porque los dígitos (es decir, el 6, el 2 y el 1) se encuentran en posiciones diferentes.

Los grupos de bits (combinaciones de ceros y unos) se llaman **códigos** y se emplean para representar números, letras, instrucciones, símbolos. Cada bit dentro de una secuencia ocupa un intervalo de tiempo definido llamado **periodo del bit**. En los sistemas digitales todas las señales han de estar sincronizadas con una señal básica periódica llamada **reloj** (ver figura 3)

3. TRANSFORMACIÓN DE BINARIO A DECIMAL

Para pasar de binario a decimal se multiplica cada una de las cifras del número en binario en potencias sucesivas de 2.

EJERCICIO RESUELTO:

Transformar los números 1010 y 10011 en código binario a sistema decimal (el subíndice indica la base del sistema de numeración):

$$1010_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8 + 2 = 10_{10}$$

$$11001_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 24 + 23 + 1 = 25_{10}$$

4. TRANSFORMACIÓN DE DECIMAL A BINARIO

El convertir un número decimal al sistema binario es muy sencillo: basta con realizar divisiones sucesivas por 2 hasta que el último cociente sea inferior a 2 y escribir los restos obtenidos en cada división en orden inverso al que han sido obtenidos.

EJERCICIO RESUELTO

Transformar elnúmero 28 en sistema decimal a código binario (el subíndice indica la base del sistema de numeración):

28 : 2 = 14 Resto: **0**
14: 2 = 7 Resto: **0**
7: 2 = 3 Resto: **1**
3 : 2 = 1 Resto: **1**

$28_{10} = 11100_2$

O, al igual que antes:

28	2
0 14	2
0 7	2
1 3	2
1 1	

$\Rightarrow 11100_2$

VÍDEOS:
Código Binario e Interconversiones entres sistemas de numeración

5. CANTIDAD DE BITS NECESARIOS PARA REPRESENTAR UN NÚMERO

La cantidad de dígitos necesarios para representar un número en el sistema binario es mayor que en el sistema decimal. Así, en el ejemplo anterior, para representar el número **11**, han hecho falta 4 dígitos en binario. Para representar números grandes harán falta muchos más dígitos. Por ejemplo, para representar números mayores de 255 se necesitarán más de 8 dígitos, porque $2^8 = 256$ y podemos afirmar, por tanto, que 255 es el número más grande que puede representarse con ocho dígitos.

Como regla general, con ***n*** dígitos binarios pueden representarse un máximo de 2^n códigos diferentes. El número más grande que puede escribirse con ***n*** dígitos es una unidad menos, es decir, $2^n - 1$.

EJERCICIO RESUELTO

Calcular cuantas combinaciones posibles de bits puede hacerse con 4 bits:

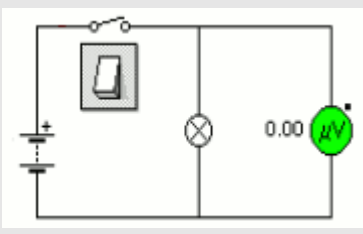
$2^4 = 16$ ➢ Pueden representarse un total de 16 combinaciones diferentes,

2^4-1 = 15 ➢ el mayor de los números en sistema decimal que podemos representar con 4 bits es el 15.

III. ÁLGEBRA DE BOOLE

En 1854, en su obra *An Investigation of the Laws of Thought* el matemático inglés George Boole desarrolló un álgebra que afecta a conjuntos de dos tipos: 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, ... Boole nunca conoció las tremendas repercusiones de su álgebra, pues no fue hasta 1939, en que Claude. E. Shannon publicó su obra *A Symbolic Analysis of Relay and Switching Circuits*, cuando se estableció la relación existente entre el álgebra de Boole y el estudio de los circuitos electrónicos.

Imaginemos el circuito de la figura. Si el interruptor está abierto, no pasa la corriente, la lámpara está apagada y el voltímetro marcará una tensión de 0 voltios.



En electrónica digital, cuando no tenemos tensión (i.e. cuando la tensión es de 0 voltios) decimos que la lámpara está en **OFF** o que tenemos un **bit "0"**..

Si ahora tenemos el interruptor cerrado, el voltímetro indica 9 V, la corriente está pasando por la bombilla (se enciende). En electrónica digital diremos que la lámpara está en **ON** o que tenemos un **bit "1"**.

Las tres operaciones o funciones lógicas del álgebra de Boole fueron la suma, a multiplicación y la negación, tal y como muestra la tabla.

Multiplicación (·)	$0 \cdot 0 = 0$	$0 \cdot 1 = 0$	$1 \cdot 0 = 0$	$1 \cdot 1 = 1$
Suma (+)	$0+0 = 0$	$0+1 = 1$	$1+0 = 1$	$1+1 = 1$
Negación (¯)	$\overline{0} = 1$		$\overline{1} = 0$	

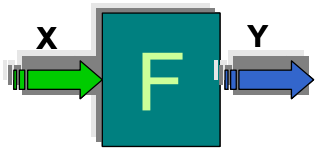
La prioridad de estos operadores es: primero la negación, después la multiplicación y por último la suma.

El álgebra de Boole son las matemáticas de los circuitos digitales. Para todas variables a,b y c que pertenecen al conjunto de álgebra de Boole se cumplen, entre otras propiedades:

PROPIEDADES DEL ÁLGEBRA DE BOOLE		
Elemento neutro	$0+A = A$	$1 \cdot A = A$
Teoremas de idempotencia	$A + A = A$	$A \cdot A = A$
Teoremas de identidad	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
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	$(A+B)+C = A+(B+C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
Teorema de involución	$\overline{\overline{A}} = A$	
Teoremas de absorción	$A + A \cdot B = A$	$A \cdot (A+B) = A$
	$A + \overline{A} \cdot B = A + B$	$A \cdot (\overline{A}+B) = A \cdot B$
Teoremas de Morgan	$\overline{A+B} = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$

IV. Funciones Lógicas y Tablas de Verdad

Dentro de los sistemas digitales nos centraremos en el estudio de los llamados ***sistemas digitales combinacionales***, que se definen, como aquel los **sistemas en el que las salidas son solamente función de las entradas actuales**, es decir, dependen únicamente de las combinaciones de las entradas, de ahí su nombre. Estos sistemas se pueden representar a través de una función digital del tipo F(X) = Y, donde X representa todas las entradas posibles e Y el conjunto de todas las salidas posibles.



Un ejemplo sencillo de sistema combinacional es un portaminas. En este sistema sólo son posibles dos acciones o entradas (pulsar o no pulsar), y sólo son posibles dos salidas (salir la mina o no hacer nada). El sistema es combinacional porque, siempre que se aplique una entrada, la respuesta del sistema sólo depende de esa entrada.

Las relaciones entre variables de entrada y salida se pueden representar en una tabla de verdad. Una **tabla de verdad** es una tabla que indica qué salida va a presentar un circuito para cada una de las posibles combinaciones de sus entradas. Debemos recordar que el número total de combinaciones es 2^n, siendo n el número de las entradas.

Imaginemos una circuito con una única salida y tres entradas (a, b, y c), donde la salida (S) toma el valor de 1 para 3 de estas combinaciones. Una posible tabla de verdad sería:

Variable de Entrada			SALIDA
a	b	c	S
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

En la parte izquierda de la tabla figuran las variables de entrada y todas las posibles combinaciones de las entradas, donde según una lógica positiva el 0 representa el nivel bajo, y un 1 representa un valor alto de la tensión. En la parte derecha figurarán las salidas en función de las entradas. Los valores de la salida son función de las entradas, cuya función, como veremos en otro apartado pueden obtenerse operando *booleanamente* con los valores de las entradas.

Así, toda función lógica puede quedar definida de tres maneras: por su expresión matemática, por su tabla de verdad o por su símbolo.

V. Puertas Lógicas

Las operaciones matemáticas habituales, en el mundo de las matemáticas binarias, son operaciones "complicadas". Existen operaciones más sencillas llamadas operaciones lógicas. Las operaciones lógicas pueden hacerlas algunos circuitos construidos con transistores. Este tipo de circuitos se llaman puertas lógicas. Por consiguiente, una **puerta lógica** no es ni mas ni menos que un circuito electrónico especializado en realizar operaciones booleanas.

Las puertas lógicas fundamentales son tres AND, OR y NOR): Combinando algunas de las puertas anteriores podemos obtener otras nuevas (NAND, NOR, XOR, XNOR.....).

VÍDEO PUERTAS LÓGICAS

1. PUERTA LÓGICA AND ("Y")

Realiza la función lógica de multiplicación. La que la señal de salida (S) será un **1** solamente en el caso de que todas (dos o más) señales de entrada sean **1**. Las demás combinaciones posibles de entrada darán una señal de salida de **0**.

SÍMBOLO	SÍMBOLO NORMALIZADO

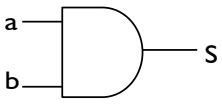
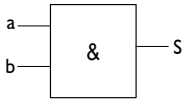
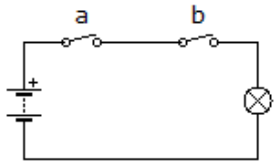

	

TABLA DE VERDAD	**FUNCIÓN**
2 entradas = $2^2 = 4$ combinaciones de las entradas	$S = a \cdot b$

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

CIRCUITO EQUIVALENTE



2. PUERTA LÓGICA OR ("O")

Realiza la función lógica de la suma lógica. Por consiguiente, la señal de salida será un 1 siempre que alguna de las señales de entrada sea un 1.

SÍMBOLO	SÍMBOLO NORMALIZADO

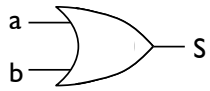
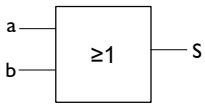
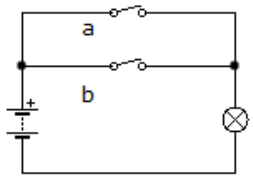

	

TABLA DE VERDAD	**FUNCIÓN**
2 entradas $= 2^2 = 4$ combinaciones de las entradas	$S = a + b$

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

CIRCUITO EQUIVALENTE



3. PUERTAS LÓGICAS NOT ("NO")

Realiza la operación lógica de inversión o complementación i.e. cambia un nivel lógico al nivel opuesto. En este caso la puerta sólo tiene una entrada.

SÍMBOLO	SÍMBOLO NORMALIZADO

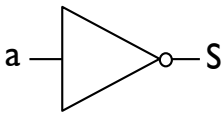
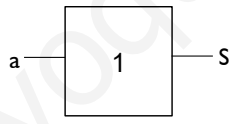
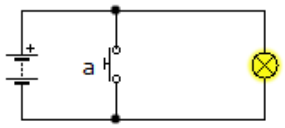

	

TABLA DE VERDAD	**FUNCIÓN**
1 entrada $= 2^1 = 2$ combinaciones de entradas	$S = \overline{a}$

a	S
0	1
1	0

CIRCUITO EQUIVALENTE



4. PUERTAS LÓGICAS NAND

La función toma valor lógico 1 cuando las entradas valen 0. Es la negación de la AND, de manera que combinando una puerta AND y una NOT obtendríamos la nueva puerta NAND.

SÍMBOLO	SÍMBOLO NORMALIZADO

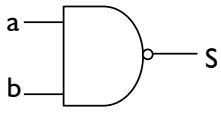
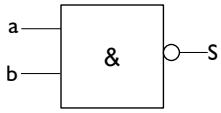
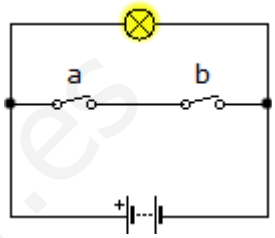

	

TABLA DE VERDAD	**FUNCIÓN**
2 entradas $= 2^2 = 4$ combinaciones de las entradas	$S = \overline{a \cdot b} = \overline{a} + \overline{b}$

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

CIRCUITO EQUIVALENTE



5. PUERTAS LÓGICAS NOR

La función toma valor lógico 1 cuando las entradas valen 0. Es la negación de la OR, de modo que combinando una puerta OR y una NOT obtendríamos la nueva puerta NOR.

SÍMBOLO	SÍMBOLO NORMALIZADO

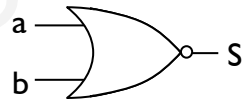
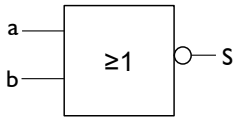
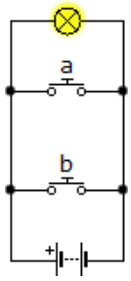

	

TABLA DE VERDAD	**FUNCIÓN**
2 entradas $= 2^2 = 4$ combinaciones de las entradas	$S = \overline{a + b} = \overline{a} \cdot \overline{b}$

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

CIRCUITO EQUIVALENTE



LABORATORIO VIRTUAL DE ELECTRÓNICA DIGITAL

SIMULADORES DE ELECTRÓNICA DIGITAL

VI. Resolución de Problemas

Para llevar a buen término la resolución de problemas deberemos seguir un orden determinado. Para poderlo explicar emplearemos el siguiente enunciado.

Implementar con puertas lógicas un sistema para determinar si un nº entre 0 y 7 es numero primo.

1. **Identificar las entradas y salidas:** en los enunciados se dan las condiciones a partir de las cuales identificaremos las entradas y salidas. En el ejemplo, como debemos obtener números entre 0 y 7 debemos emplear 3 entradas (2^3-1 =7) con una única salida.

2. **Crear la tabla de verdad a partir de del enunciado**: en nuestro caso pondremos como salida un 1 en todos los casos donde las combinaciones binarias corresponden a un número primo (2,3,5 y 7).

Nº representado	a	b	c	S
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Normalmente las tablas de verdad deben simplificarse empleando técnicas como, por ejemplo, los mapas de Karnaugh.

3. **Obtener la función lógica a partir de la tabla de verdad:** podemos elegir por dos opciones, implementación por 1s o por 0s.

   - **Implementación por 1s** para obtener la ***primera forma canónica*** de una función lógica. Se obtiene directamente a partir de la tabla de verdad **sumando todos los productos lógicos correspondientes a las salidas que dan una salida igual a 1** (despreciamos los que corresponden a una salida igual a 0). **Las entradas con 0 se consideran negadas, y las entradas con 1 no negadas.**

a	b	c	S	
0	0	0	0	
0	0	1	0	
0	1	0	1	$f = \bar{a}\cdot b\cdot \bar{c}$
0	1	1	1	$f = \bar{a}\cdot b\cdot c$
1	0	0	0	
1	0	1	1	$f = a\cdot \bar{b}\cdot c$
1	1	0	0	
1	1	1	1	$f = a\cdot b\cdot c$

La 1ª forma canónica (F1) en su forma estandar es una suma de productos lógicos, que en nuestro ejemplo será:

$$F_1 = \bar{a}\cdot b\cdot \bar{c} + \bar{a}\cdot b\cdot c + a\cdot \bar{b}\cdot c + a\cdot b\cdot c$$

   - **Implementación por 0s** para obtener la ***segunda forma canónica*** de una función lógica. Se obtiene a partir de la tabla de verdad **multiplicando todos los sumandos lógicos cuya salida sea 0** (despreciamos los aquellos cuya salida es 1). **Las entradas con 1 se consideran negadas, y las entradas con 0 no negadas.**

a	b	c	S	
0	0	0	0	$f = a+b+c$
0	0	1	0	$f = a+b+\bar{c}$
0	1	0	1	
0	1	1	1	
1	0	0	0	$f = \bar{a}+b+c$
1	0	1	1	
1	1	0	0	$f = \bar{a}+\bar{b}+c$
1	1	1	1	

La 2ª forma canónica (F2) en su forma estandar es un producto de sumas lógicas, que en nuestro ejemplo será:

$$F_2 = (a+b+c)\cdot(a+b+\bar{c})\cdot(\bar{a}+b+c)\cdot(\bar{a}+\bar{b}+c)$$

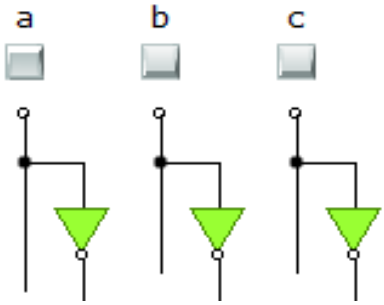
4. **Simplificar la función lógica:**

Con el objetivo de reducir el coste, ocupar menos espacio y aumentar la fiabilidad de los circuitos, las funciones lógicas derivadas de las tablas de verdad deberían ser lo más simples posibles.

Métodos de simplificación tales como métodos algebraicos, mapas de Karnaugh, de Quine-McCluskey...etc intentan obtener funciones lógicas equivalentes a las canónicas pero más sencillas; i.e. funciones que con las mismas entradas, proporcionan las mismas salidas minimizando el número de variables en cada término y el número de términos. Veremos algunoas formas sde simplificación en la **Ficha nº1** y en el apéndice de este tema.

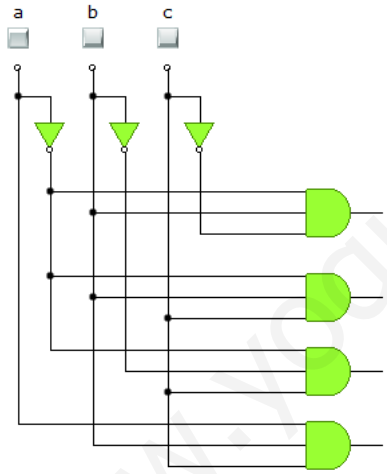
5. Implementar el circuito empleando puertas lógicas a partir de las funciones obtenidas:

5.1 Para ello se **dibujarán tantos terminales lógicos de entrada (*inputs*) como variables** de las que dependa la función (tres en nuestro ejemplo). Estos terminales deberían incluir, en caso necesario) sus valores negados utilizando puertas NOT.

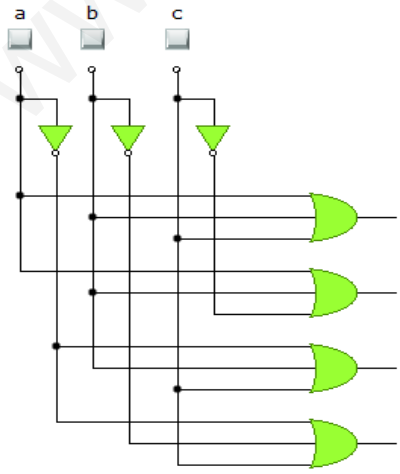


5.2 A continuación **conectamos las variables de cada término con puertas AND** (si empleamos la 1ª forma canónica) **u OR** (si usamos la 2ª forma canónica). Si sólo hay dos entradas se usará una sola puerta, si hay tres o más se irán añadiendo puertas.

Para F1:

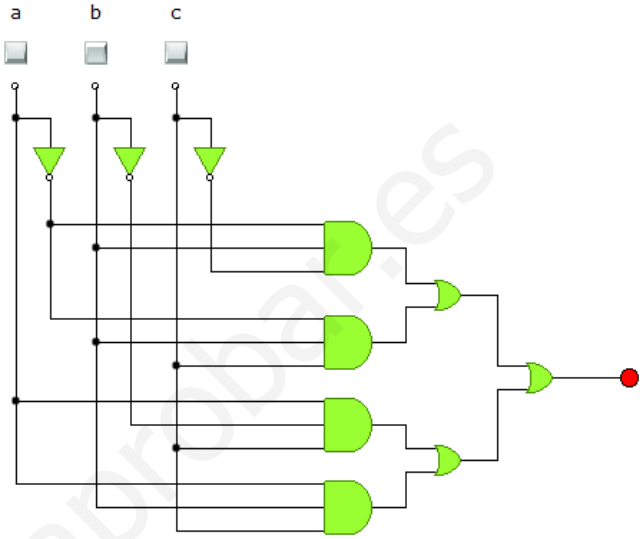


Para F2:

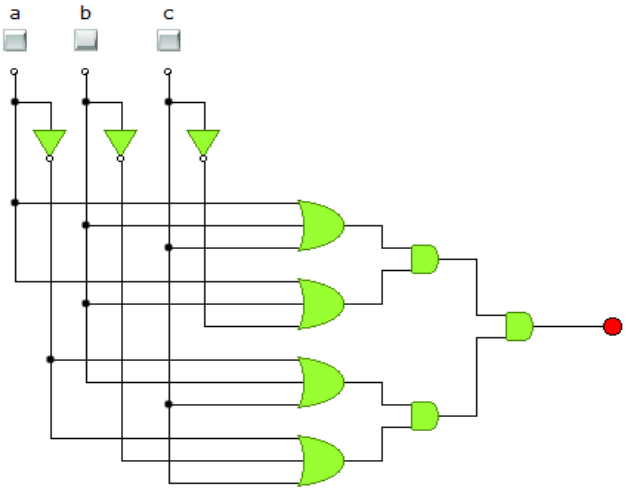


5.3 Seguidamente, **conectaremos las salidas de las últimas puertas AND** (de cada sumando) **u OR** (de cada producto) utilizando puertas OR (suma) o AND (producto), respectivamente. De esa manera conseguiremos implementar las operaciones correspondientes.

Así, si usamos la 1ª forma canónica tendremos el siguiente circuito:



Si partimos de la 2ª forma canónica tendremos el siguiente circuito:



VÍDEOS DE REPASO

EJERCICIO RESUELTO: SISTEMA DE SEGURIDAD DE UNA VIVIENDA

Se desea instalar un sistema de alarma en una vivienda compuesto por dos sensores (a y b) en sendas ventanas, y un interruptor de la alarma (c). Cuando el sistema está activado (se cerrará el interruptor), un timbre deberá sonar al abrir alguna o las dos ventanas. Si el sistema no está activado, el timbre no sonará aunque se abra alguna de las ventanas. Implementar un circuito electrónico digital empleando puertas NOT, OR y AND para el control del sistema

➔ Identificamos 3 entradas (a,b y c) y la salida (S), asignando los siguientes valores lógicos 0 y 1 a los estados físicos: entradas y salidas:

- ✗ Ventanas: cerradas (0), abiertas (1)
- ✗ Interruptor: abierto (0), cerrado (1)
- ✗ Alarma sonora : inactiva (0), activa (1)

➔ Elaboramos la tabla de verdad y obtenemos la 1ª forma canónica (en la salida hay más 1s que 0s).

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

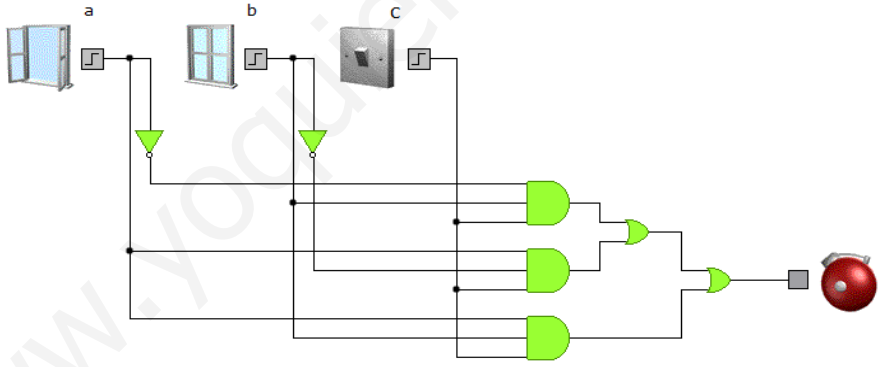
$f = \bar{a}\cdot b\cdot c$

$f = a\cdot \bar{b}\cdot c$

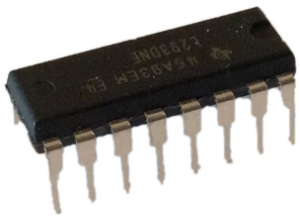
$f = a\cdot b\cdot c$

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

➔ Finalmente implementamos el circuito:



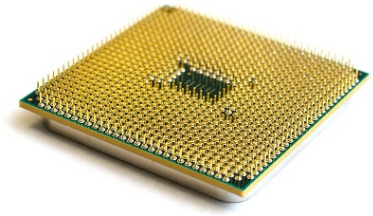
VII. Circuitos Integrados



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

Los **Circuitos Integrados** (I.C. *Integrated Circuits*), también llamados **chips**, son circuitos formados por componentes electrónicos miniaturizados (transistores, diodos, resistencias, condensadores....) y montados sobre una oblea de silico. Utilizan pequeños chips de silicio protegidos por una funda o carcasa de plástico y con unas patillas para realizar las conexiones.

Los **microprocesadores** que se emplean en ordenadores y en múltiples aplicaciones industriales son un ejemplo de circuito integrado. Estos dispositivos son los encargados de ejecutar los programas. A diferencia de otros IC; los microprocesadores, sólo ejecutan instrucciones programadas; i.e. requieren de un programa específico o **software**, realizando operaciones aritméticas y lógicas tales como suma, resta, multiplicación y división, lógicas binarias y accesos a la memoria.



En un chip, los elementos del circuito son tan pequeños que se necesita un buen microscopio para verlo. En un microchip de un par de centímetros de largo por un par de centímetros de ancho pueden caber millones de transistores (además de resistencias, condensadores, diodos, etc. Así por ejemplo, el Core i7 de Intelintegra entre 731 y 1170 millones de transistores, según el modelo.

Los IC se pueden implementar con diferentes técnicas o *tecnologías*, según sean los métodos de fabricación de los componentes. Las tecnologías más conocidas y usadas son las **TTL** (***Transistor-Transistor Logic***) y **CMOS** (***Complementary Metal Oxide Semiconductor***), aunque existen otras tales como la ECL, DTL, Bipolar, NMOS, PMOS. Sin embargo, no es el objetivo de esta unidad el profundizar en su conocimiento.

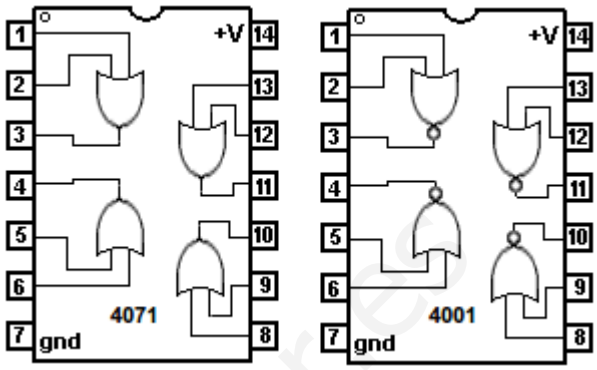
El empleo de los IC desde la década de 1950 ha permitido, entre otros:

- Minimizar el cableado de los equipos electrónicos
- Minimizar el tamaño y el peso de éstos
- Facilitar el ensamblaje y montaje de los equipos electrónicos.

Algunas de las ventajas del empleo de IC frente a una implementación tradicional basada en transistores discretos, son:

- Alto grado de integración, llegándose a implementar millones de componentes en un chip de reducidas dimensiones.
- Reducción de coste, debido al alto grado de automatización existente en la fabricación de los CI y la producción en masa.
- La fiabilidad. Un IC posee mayor fiabilidad en cuanto a funcionamiento y duración que los transistores discretos.
- La velocidad de funcionamiento es mayor ya que el paso de la corriente depende de las longitudes de las interconexiones, muy pequeñas dentro del CI.
- Reducción de los posibles errores de montaje e interconexionado de componentes.
- Disminución del nº de averías debido al contacto entre cables, malas soldaduras, errores en la fabricación...

Fig 4: Diagrama de las conexiones de un circuito integrado CMOS



Existen miles de CI diferentes. Cada fabricante especifica las funciones y condiciones de funcionamiento de cada uno de ellos. Uno de los factores más importantes a considerar es la temperatura, ya que algunos trabajan a tales velocidades o con corrientes tan elevadas que podrían llegarse a fundir.

A continuación presentamos algunos de los más habituales:

- **Amplificadores Operacionales (*μ*741):** Este circuito integrado de 8 patas sirve para aumentar una señal de entrada; por ejemplo, la señal de voltaje que tiene un micrófono para que salga por un altavoz, o para amplificar la señal de antena de una televisión (no el 741 sino otro modelo).
- **Comparador (LM741, LM311...)** se emplea para comparar el nivel de dos señales; por ejemplo activar un ventilador si se supera determinada temperatura...
- **Regulador de tensión (7805,7806,7809...):** Se emplea cuando es necesario una tensión continua a partir de la tensión alterna de la red eléctrica.
- **Temporizador (N555):** El temporizador NE555 es otro circuito integrado de 8 patas. Genera señales temporales con mucha estabilidad y precisión, lo cual lo convierte en el circuito base de muchas aplicaciones que necesite un control del tiempo: temporizadores, generadores de señales, relojes, retardadores, etc.

DISPLAY DE 7 SEGMENTOS

Muchos equipos electrónicos emplean un display de 7 segmentos (Fig: 5) formado por 7 LEDs y un circuito decodificador BCD (de 4 bits) para formar los caracteres decimales de 0 a 9 (y algunas veces los caracteres hexadecimales de A a F). Para tratar de explicar su funcionamiento, vamos a considerar un depósito de agua en el cual hemos colocado 3 sensores de humedad (S1, S2 y S3) para poder conocer su nivel de llenado. Cada sensor entregará un 1 cuando el agua haya alcanzado o superado el nivel del sensor, y un valor 0 cuando no le alcance el agua. Los tres sensores irán conectados al circuito de control que vamos a diseñar.

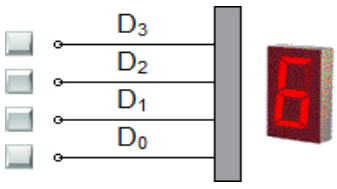


Fig 5: Display de 7 segmentos

Tabla 1: Visualización del display en función de las entradas

D3	D2	D1	D0	Mensaje
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Los displays de 7 segmentos suelen comercializarse con un codificador BCD incorporado, que dispone de 4 terminales de entrada (D1, D2, D3 y D4). La tabla 1 muestra la visualización que ofrece el display en función de las entradas del codificador BCD. Estos 4 terminales serán las salidas del circuito lógico, por lo que a cada uno le asignaremos una función lógica. Un valor lógico 0 indicará que no llega corriente y un valor 1 indicará que llega corriente. Emplearemos la tabla de verdad 2 para diseñar el circuito de control. En nuestro sistema cuando se presente una situación absurda (por ejemplo que el sensor S3 detecte agua, no detectándola el sensor S1) el mensaje que debe mostrar el display será de E de error.

Tabla 2: Tabla de verdad según las variables de entrada

Variables			Mensaje del Display	Funciones			
S3	S2	S1		D3	D2	D1	D0
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	E	1	1	1	0
0	1	1	2	0	0	1	0
1	0	0	E	1	1	1	0
1	0	1	E	1	1	1	0
1	1	0	E	1	1	1	0
1	1	1	3	0	0	1	1

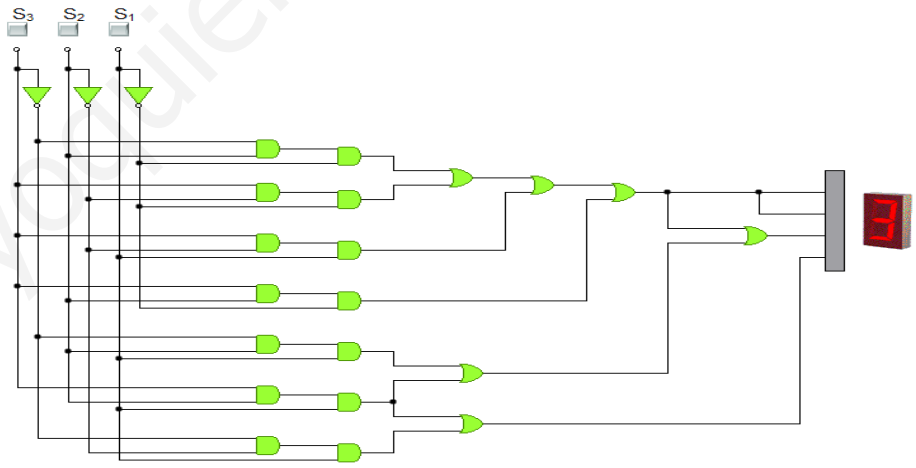
A partir de la tabla de verdad 2 podemos deducir las primeras formas canónicas (Fijaros que en este ejemplo, las funciones D2 y D3 son iguales). Estas formas canónicas serán:

$$D0 = \overline{S3}\cdot\overline{S2}\cdot S1 + S3\cdot S2\cdot S1$$

$$D1 = \overline{S3}\cdot S2\cdot\overline{S1}\cdot + \overline{S3}\cdot S2\cdot S1 + S3\cdot\overline{S2}\cdot\overline{S1} + S3\cdot\overline{S2}\cdot S1 + S3\cdot S2\cdot\overline{S1} + S3\cdot S2\cdot S1$$

$$D3 = D2 = \overline{S3}\cdot S2\cdot\overline{S1}\cdot + S3\cdot\overline{S2}\cdot\overline{S1} + S3\cdot\overline{S2}\cdot S1 + S3\cdot S2\cdot\overline{S1}$$

Según estas funciones una posible implementación del circuito con puertas NOT, OR y NAD con dos entradas puede ser como el mostrado. Sin embargo empleando el álgebra de Boole u otro cualquier método de simplificación (no estudiado) dicho circuito se podrá simplificar, con el propósito de obtener un circuito con menos puertas y por lo tanto más barato, y que tarde menos tiempo en procesar la información.



APENDICE: Simplificación Mediante Álgebra de Boole

En el desarrollo de la unidad hemos visto como a partir de un enunciado, podemos obtener, una tabla de verdad, y a partir de ésta obtener las funciones canónicas ya sea en forma de suma de productos (implementando por 1) o en forma de producto de sumas (implementando por 0. Sin embargo, en la mayoría de los casos el empleo de la formas canónicas para diseñar un circuito conlleva el empleo de numerosas puertas lógicas.

En un circuito, para reducir su coste, tamaño y aumentar su fiabilidad, es conveniente simplificar dichas formas. Hay varios métodos para llevar la simplificación. En esta unidad veremos únicamente 2: la simplificación empleando el álgebra de Boole y el método gráfico mediante mapas de Karnaugh. Éste último se trabajará en una **ficha** aparte.

El primero de estos métodos consiste en simplificar las funciones lógicas con el empleo de las propiedades del álgebra de Boole. Recordamos algunas de ellas, las cuales, en su mayoría os sonarán por haberlas estudiado en matemáticas

PROPIEDADES DEL ÁLGEBRA DE BOOLE		
Elemento neutro	$0 + A = A$	$1 \cdot A = A$
Teoremas de idempotencia	$A + A = a$	$A \cdot A = A$
Teoremas de identidad	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
Propiedades conmutativas	$A + B = B + A$	$A \cdot B = B \cdot A$
Propiedades asociativas	$(A+B)+C = A+(B+C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
Propiedades distributivas	$A \cdot (B+C) = (A \cdot B)+(A \cdot C)$	$A+(B \cdot C) = =(A+B) \cdot (A+C)$
Teoremas de absorción	$A + A \cdot B = A$	$A \cdot (A+B) = A$
	$A + \overline{A} \cdot B = A + B$	$A \cdot (\overline{A}+B) = A \cdot B$
Teoremas de Morgan	$\overline{A+B} = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$

Veamos algunos ejemplos:

Ejemplo 1

- $\overline{A}\,\overline{B}\,C\,D + \overline{A}\,\overline{B}\,C\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}\,D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} = \overline{A}\,\overline{B}\,C\,(D+\overline{D}) + \overline{A}\,\overline{B}\,\overline{C}\,(D+\overline{D}) =$ Podemos aplicar la **Propiedad distributiva** para sacar factor común.
- $\overline{A}\,\overline{B}\,C\,(D+\overline{D}) + \overline{A}\,\overline{B}\,\overline{C}\,(D+\overline{D}) = \overline{A}\,\overline{B}\,C + \overline{A}\,\overline{B}\,\overline{C} =$ Empleamos el **Teorema de la identidad** $(D + \overline{D}) = 1$
- $\overline{A}\,\overline{B}\,C + \overline{A}\,\overline{B}\,\overline{C} = \overline{A}\,\overline{B}\,(C+\overline{C}) =$ Volvemos a aplicar la **propiedad distributiva para sacar factor común**
- $\overline{A}\,\overline{B}\,(C+\overline{C}) = \overline{A}\,\overline{B}$ Y finalmente usamos otra vez el **Teorema de la identidad** $(C + \overline{C} = 1$

Ejemplo 2:

- $\overline{A}\,B\,C + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} = \overline{A}\,C\,(B+\overline{B}) + \overline{A}\,\overline{C}\,(B+\overline{B})$ Aplicamos la **Propiedad distributiva** para sacar factor común.
- $\overline{A}\,C\,(B+\overline{B}) + \overline{A}\,\overline{C}\,(B+\overline{B}) = \overline{A}\,C + \overline{A}\,\overline{C}$ Usamos el **Teorema de la identidad** para simplificar la función $B + \overline{B} = 1$
- $\overline{A}\,C + \overline{A}\,\overline{C} = \overline{A}\,(C+\overline{C})$ Volvemos a aplicar la **propiedad distributiva para sacar factor común**
- $\overline{A}\,(C+\overline{C})\,0 = \overline{A}$ Aplicando otra vez el **Teorema de la identidad,** obtenemos la función simplificada. $C + \overline{C} = 1$

Ejemplo 3:

- $AB + A(B+C) + B(B+C) = AB + AB + AC + BB + BC$ Empleamos la **propiedad distributiva** para obtener únicamente una suma de productos
- $AB + AB + AC + BB + BC = AB + AC + B + BC$ Aplicamos la **Propiedad de la idempotencia** para eliminar términos iguales $AB=BA$ y $BB=B$
- $AB + AC + B + BC = AB + AC + B(1+C) = AB + AC + B$ **Sacamos factor común** $B+BC = (1+C)$ y sabiendo que $(1+C)=1$, podemos seguir simplificando
- $AB + AC + B = AC + B(A+1) = AC+B$ Volvemos a **sacar factor común** B y sabiendo que $A+1 = 1$ podemos simplificar $AB+B = B$