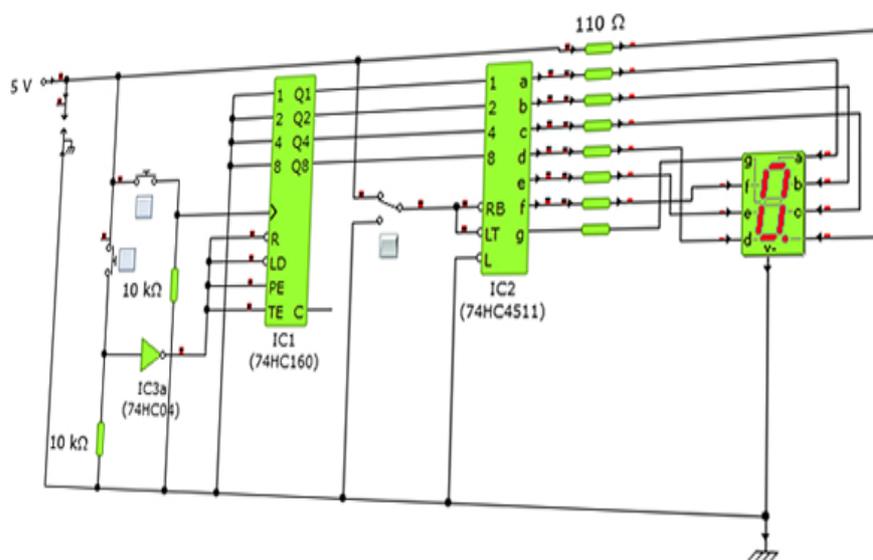


UNIDAD TEMÁTICA 3

Electrónica Digital



4º DE ESO
(REFUERZO)

ELABORADO POR: *Pedro Landín*

CPR COLEXIO SAGRADO CORAZÓN DE XESÚS
(Placeres). Pontevedra



I. 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.

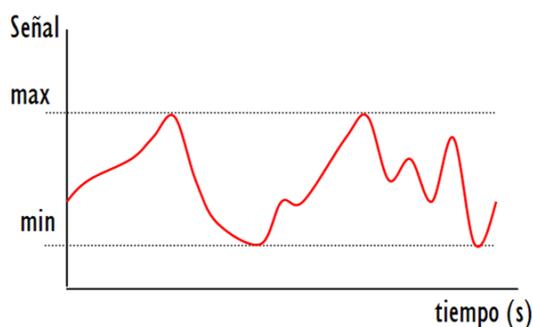
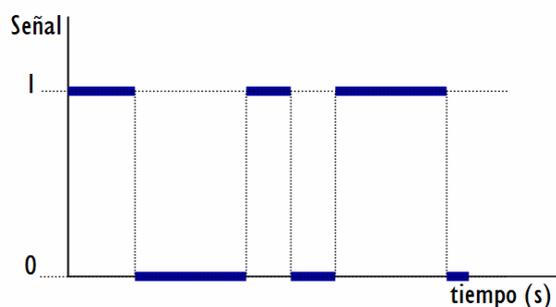


Fig 1: Ejemplo de señal analógica.

- **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, 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. 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 valor de cada cifra depende de su posición dentro de la cantidad que representa.

$$2165 = 2 \cdot 10^3 + 1 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0 = 1000 + 60 + 5$$

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.

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.

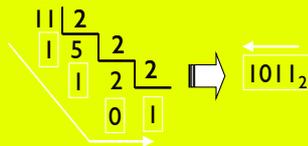
EJERCICIO RESUELTO:

Transformar los números 11 y 28 en sistema decimal a código binario (el subíndice indica la base del sistema de numeración):

Primero transformamos el número 11

$$\begin{array}{l} 11 : 2 = 5 \quad \text{Resto: } 1 \\ 5 : 2 = 2 \quad \text{Resto: } 1 \\ 2 : 2 = 1 \quad \text{Resto: } 0 \end{array} \quad \left. \vphantom{\begin{array}{l} 11 : 2 = 5 \\ 5 : 2 = 2 \\ 2 : 2 = 1 \end{array}} \right\} 11_{10} = 1011_2$$

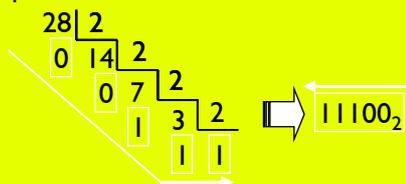
Otra manera de expresarlo será:



Ahora transformamos el nº 28

$$\begin{array}{l} 28 : 2 = 14 \quad \text{Resto: } 0 \\ 14 : 2 = 7 \quad \text{Resto: } 0 \\ 7 : 2 = 3 \quad \text{Resto: } 1 \\ 3 : 2 = 1 \quad \text{Resto: } 1 \end{array} \quad \left. \vphantom{\begin{array}{l} 28 : 2 = 14 \\ 14 : 2 = 7 \\ 7 : 2 = 3 \\ 3 : 2 = 1 \end{array}} \right\} 28_{10} = 11100_2$$

0, al igual que antes:



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:

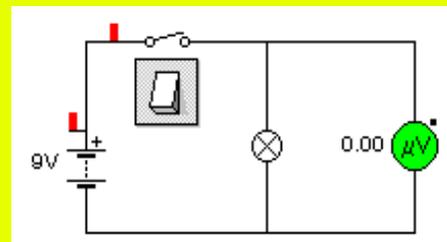
$2^4 = 16 \Rightarrow$ Pueden representarse un total de 16 combinaciones diferentes,

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

III. ÁLGEBRA DE BOOLE

En 1854, 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, ...

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



En electrónica digital, cuando no tenemos tensión (i.e. cuando la tensión es de cero 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.

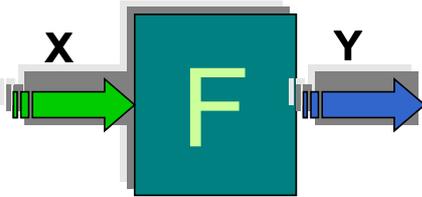
Multiplicación (·)	0 · 0 = 0	0 · 1 = 0	1 · 0 = 0	1 · 1 = 1
Suma (+)	0 + 0 = 0	0 + 1 = 1	1 + 0 = 1	1 + 1 = 1
Negación (¬)	$\bar{0} = 1$		$\bar{1} = 0$	

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.

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 aquellos sistemas en el que las salidas son solamente función de las entradas actuales, es decir, dependen únicamente de las combinaciones de las entradas. 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. (El número total de combinaciones es 2^n , siendo n el número de las entradas).

Imaginemos una circuito con una única salida y 2 entradas (a y b), donde la salida (S) toma el valor de 1 para 2 de estas combinaciones. Una posible tabla de verdad sería:

VARIABLE DE ENTRADA		SALIDA
a	b	S
0	0	1
0	1	0
1	0	1
1	1	0

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.**

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

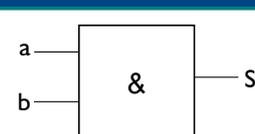
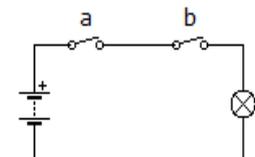
V. PUERTAS LÓGICAS

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 más 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.....).

1. PUERTA LÓGICA AND ("Y")

Aquella en 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. Dicho de otra manera, **realiza la función lógica de multiplicación.**

SÍMBOLO	SÍMBOLO NORMALIZADO
	
TABLA DE VERDAD	FUNCIÓN
2 entradas = $2^2 = 4$ combinaciones de las entradas	$S = a \cdot b$
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 VERDAD	FUNCIÓN															
2 entradas = 2 ² = 4 combinaciones de las entradas	$S = a + b$															
CIRCUITO EQUIVALENTE																
<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	S	0	0	0	0	1	1	1	0	1	1	1	1	
a	b	S														
0	0	0														
0	1	1														
1	0	1														
1	1	1														

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	$S = \bar{a}$						
CIRCUITO EQUIVALENTE							
<table border="1"> <thead> <tr> <th>a</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	a	S	0	1	1	0	
a	S						
0	1						
1	0						

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 VERDAD	FUNCIÓN															
2 entradas = 2 ² = 4 combinaciones de las entradas	$S = \overline{a \cdot b} = \bar{a} + \bar{b}$															
<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	a	b	S	0	0	1	0	1	1	1	0	1	1	1	0	
a	b	S														
0	0	1														
0	1	1														
1	0	1														
1	1	0														

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	$S = \overline{a + b} = \bar{a} \cdot \bar{b}$															
<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>SS</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	SS	0	0	0	0	1	1	1	0	1	1	1	1	
a	b	SS														
0	0	0														
0	1	1														
1	0	1														
1	1	1														

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 n° 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 (2³-1=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 la de los mapas de Karnaugh (no estudiaremos este curso los métodos de simplificación)

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.

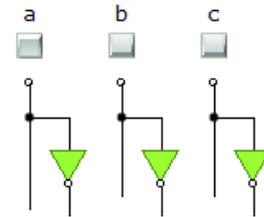
a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

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

La **1ª forma canónica (F1)** en nuestro ejemplo será:
 $F_1 = \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot c$

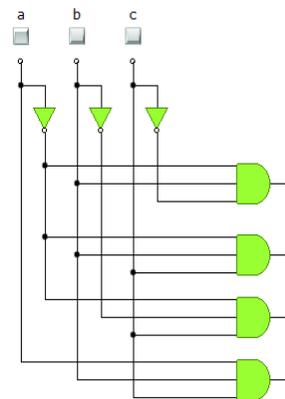
4. Implementar el circuito empleando puercas lógicas a partir de las funciones obtenidas:

4.1 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.



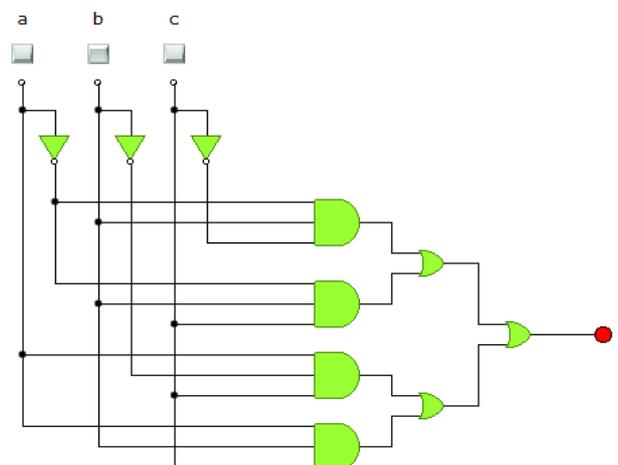
4.2 A continuación conectamos las variables de cada término con puertas AND. Si sólo hay dos entradas se usará una sola puerta, si hay tres o más se irán añadiendo puertas.

Para F1:



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

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



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:

x Ventanas: cerradas (0), abiertas (1) x Interruptor: abierto (0), cerrado (1) x 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

⇒ $f = \bar{a} \cdot b \cdot c$

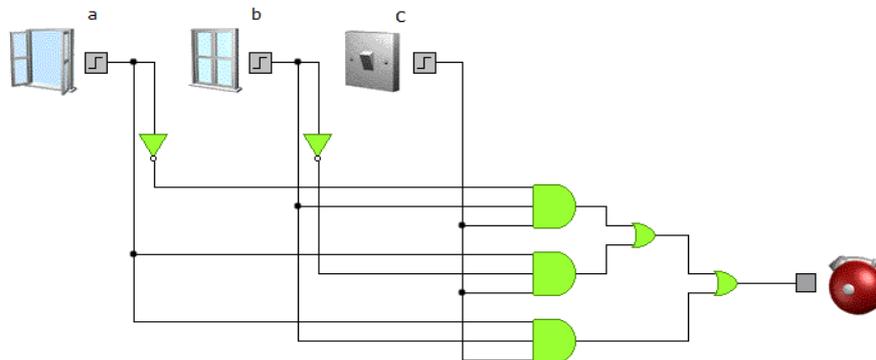
⇒ $f = a \cdot \bar{b} \cdot c$

⇒ $f = a \cdot b \cdot c$

}

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

→ 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*) son circuitos que están formados por componentes electrónicos (transistores, diodos, resistencias, condensadores....) fabricados en una oblea de silicio (miniaturizados), Utilizan pequeños chips de silicio protegidos por

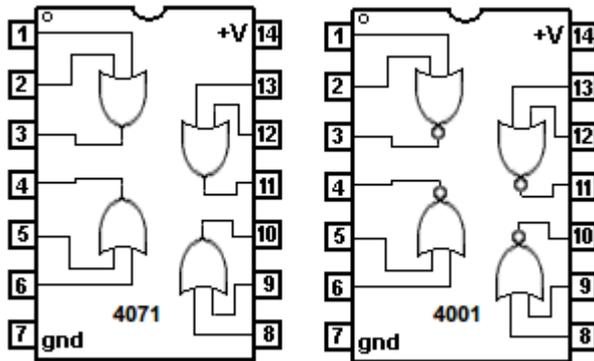
una funda o carcasa de plástico y con unas patillas para realizar las conexiones. También se les llama chip o **microchip**.



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 haber millones de transistores además de resistencias, condensadores, diodos, etc. Un ejemplo muy bueno sería el microprocesador de un ordenador. El pentium IV de Intel, sacado al mercado en el 2001, integraba unos 42 millones de transistores.

Los IC se pueden implementar con diferentes técnicas o *tecnologías*, según sean los métodos de fabricación de los componentes. 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.

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



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 interconexión de componentes.
- Disminución del nº de averías debido al contacto entre cables, malas soldaduras, errores en la fabricación...

Existen miles de circuitos integrados 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 (por ejemplo los microprocesadores de los ordenadores) o con corrientes tan elevadas que podrían llegar a fundir.

EJERCICIOS. TEMA 4: ELECTRÓNICA DIGITAL

1. Relaciona los términos de ambas columnas:

- | | |
|--|-------------------|
| ● Señales que pueden adquirir únicamente valores concretos. | ● Señal analógica |
| ● Señales que varían de manera continua. | ● Señal digital |
| ● Señales que no varían de manera continua. | |
| ● Señal que puede adquirir infinitos valores entre dos extremos cualesquiera | |

2. Indica si las siguientes magnitudes son analógicas (A) o digitales (D):

- | | |
|---------------------------|---|
| ● Tu peso | ● Tu nota final de Tecnología |
| ● Los números de teléfono | ● La numeración de los edificios de una calle |
| ● Los códigos postales | ● El dinero que llevas en el bolsillo |
| ● Tu altura | ● El número de alumnos de una clase |
| ● La temperatura ambiente | ● El número de preguntas acertadas en un examen |
| ● La presión atmosférica | ● Número de electrones de un átomo |
| ● Cantidad de carga | ● La numeración de las hojas de un libro |

3. Indica cada caso el valor lógico que tomarían los niveles (0 o 1) según una lógica positiva:

MAGNITUD	CARACTERÍSTICA	Valor lógico
Depósito de agua	Lleno	
	Vacío	
Resultado de un examen	Aprobado	
	Suspenso	
Estado de un ordenador	Apagado	
	Encendido	
Estado de una bombilla	Encendido	
	Apagada	
Resultado partido de baloncesto	Ganado	
	Perdido	

4. Transforma los siguientes números en código binario a sistema decimal:

✓ 110

✓ 101

✓ 1011

✓ 1010

✓ 10111

✓ 11101

✓ 101101

5. Transforma los siguientes números en sistema decimal a código binario:

✓ 32=

✓ 31=

✓ 47 =

✓ 54=

✓ 68

✓ 134

6. ¿Cuál es el mayor número en sistema decimal que se puede representar con los siguientes números de bits?

✓ 2 bits

✓ 3 bits

✓ 4 bits

✓ 8 bits

✓ 10 bits

7. ¿Cuántas combinaciones se pueden obtener con los siguientes números de bits? Escribe las posibles combinaciones. En forma de tabla

✓ 1 bits

✓ 2 bits

✓ 3 bits

✓ 4 bits

✓ 5 bits

8. Escribe los siguientes números en decimal en sistema binario, ordenándolos de mayor a menor en una tabla:

✓ del 0 al 3

Sistema binario		Sistema decimal
		0
		1
		2
		3

✓ del 0 al 7

Sistema binario		Sistema decimal
		0
		1
		2
		3
		4
		5
		6
		7

✓ del 0 al 15

Sistema binario		Sistema decimal
		0
		1
		2
		3
		4
		5
		6
		7
		8
		9
		10
		11
		12
		13
		14
		15

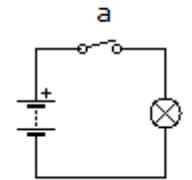
✓ Del 0 al 5

Sistema binario		Sistema decimal
		0
		1
		2
		3
		4
		5

9. Completa las tablas de verdad para los siguientes circuitos (considerar un 1 cuando la bombilla se encienda).

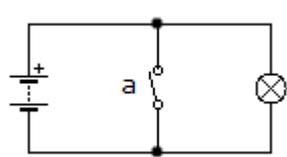
A

a	S



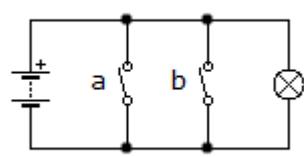
B

a	S



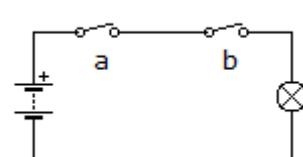
C

a	b	S



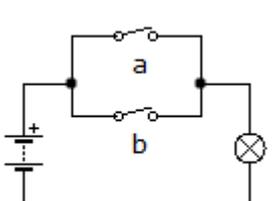
D

a	b	S



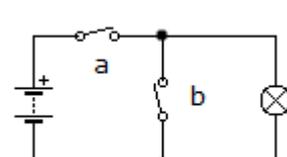
E

a	b	S



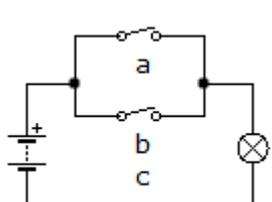
F

a	b	S



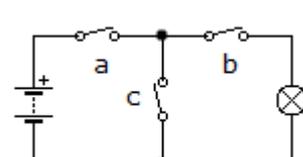
G

a	b	c	S



H

a	b	c	S



I			
a	b	S ₁	S ₂

10. Identifica las siguientes puertas lógicas con la columna de la derecha

- | | |
|---------|--|
| 1. NAND | a) La señal de salida será un 1 sólo si todas las señales de entrada son 1. |
| 2. NOT | b) La señal de salida será un 1 cuando las entradas valen 0. |
| 3. OR | c) La señal de salida será un 1 siempre que alguna de las señales de entrada sea un 1. |
| 4. AND | d) La señal de salida será el nivel lógico opuesto al de entrada. |
| 5. NOR | e) La señal de salida será una 1 cuando todas las entradas valen 0. |

11. Relaciona las siguientes puertas lógicas con su símbolo

- | | |
|--------|----|
| 1. NOT | a) |
| 2. OR | b) |
| 3. AND | c) |
| 4. AND | d) |
| 5. NOR | e) |

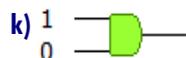
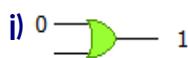
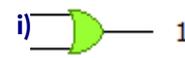
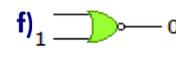
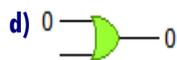
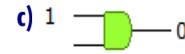
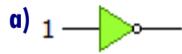
12. Rellena los huecos con los valores lógicos (0 y 1). Nota: puede haber varias posibilidades.

- En una puerta NAND cuando las señales de entrada son 0 y 1 la señal de salida será un ____
- En una puerta OR cuando las señales de entrada son 0 y 1 la señal de salida será un ____
- En una puerta AND cuando las señales de entrada son 0 y 1 la señal de salida será un ____
- En una puerta NOT cuando la señal de entrada es un 1 la señal de salida será un ____
- En una puerta NOT cuando la señal de salida es un 1 la señal de entrada será un ____
- En una puerta NOR cuando las señales de entrada son 0 y 0 la señal de salida será un ____
- En una puerta NAND cuando las señales de entrada son 1 y 0 la señal de salida será un ____
- En una puerta OR cuando la señal de salida es un 0, las señales de entrada serán ____

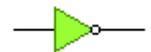
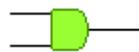
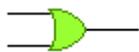
13. Nombra cada puerta e indica la operación lógica que representa:



14. Nombra los tipos de puertas lógicas y coloca el valor del bit que falta, bien en la entrada o bien en la salida, según corresponda.



15. Identifica cada una de las puertas lógicas mostradas, y tras mostrárselas al profesor cubre las tablas de verdad correspondientes (cúbrelas a lápiz):



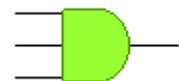
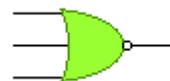
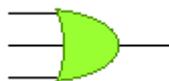
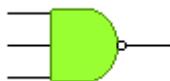
ENTRADAS		SALIDA
A	B	S

ENTRADAS		SALIDA
A	B	S

ENTRADAS		SALIDA
A	B	S

ENTRADAS		SALIDA
A	B	S

ENTRADA	SALIDA
A	S



ENTRADAS			Salida
A	B	C	S

ENTRADAS			Salida
A	B	C	S

ENTRADAS			Salida
A	B	C	S

ENTRADAS			Salida
A	B	C	S

16. Obtén la función lógica de salida a partir de las siguientes tablas de verdad:

a)

ENTRADAS		SALIDA
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

b)

ENTRADAS		SALIDA
A	B	S
0	0	0
0	1	1
1	0	0
1	1	0

c)

ENTRADAS		SALIDA
A	B	S
0	0	0
0	1	0
1	0	1
1	1	0

d)

ENTRADAS		SALIDA
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

e)

ENTRADA	SALIDA
A	S
0	1
1	0

a)

d)

b)

e)

c)

f)

ENTRADAS		SALIDA
A	B	S
0	0	1
0	1	1
1	0	0
1	1	0

g)

ENTRADAS		SALIDA
A	B	S
0	0	1
0	1	0
1	0	1
1	1	0

h)

ENTRADAS		SALIDA
A	B	S
0	0	1
0	1	1
1	0	0
1	1	1

i)

ENTRADAS		SALIDA
A	B	S
0	0	0
0	1	0
1	0	1
1	1	1

j)

ENTRADA	SALIDA
A	S
0	1
1	1

f)

i)

g)

j)

h)

k)

ENTRADAS		SALIDA
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

l)

ENTRADAS		SALIDA
A	B	S
0	0	0
0	1	1
1	0	0
1	1	1

m)

ENTRADAS		SALIDA
A	B	S
0	0	1
0	1	1
1	0	1
1	1	1

n)

ENTRADAS		SALIDA
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

o)

ENTRADA	SALIDA
A	S
0	0
1	1

k)

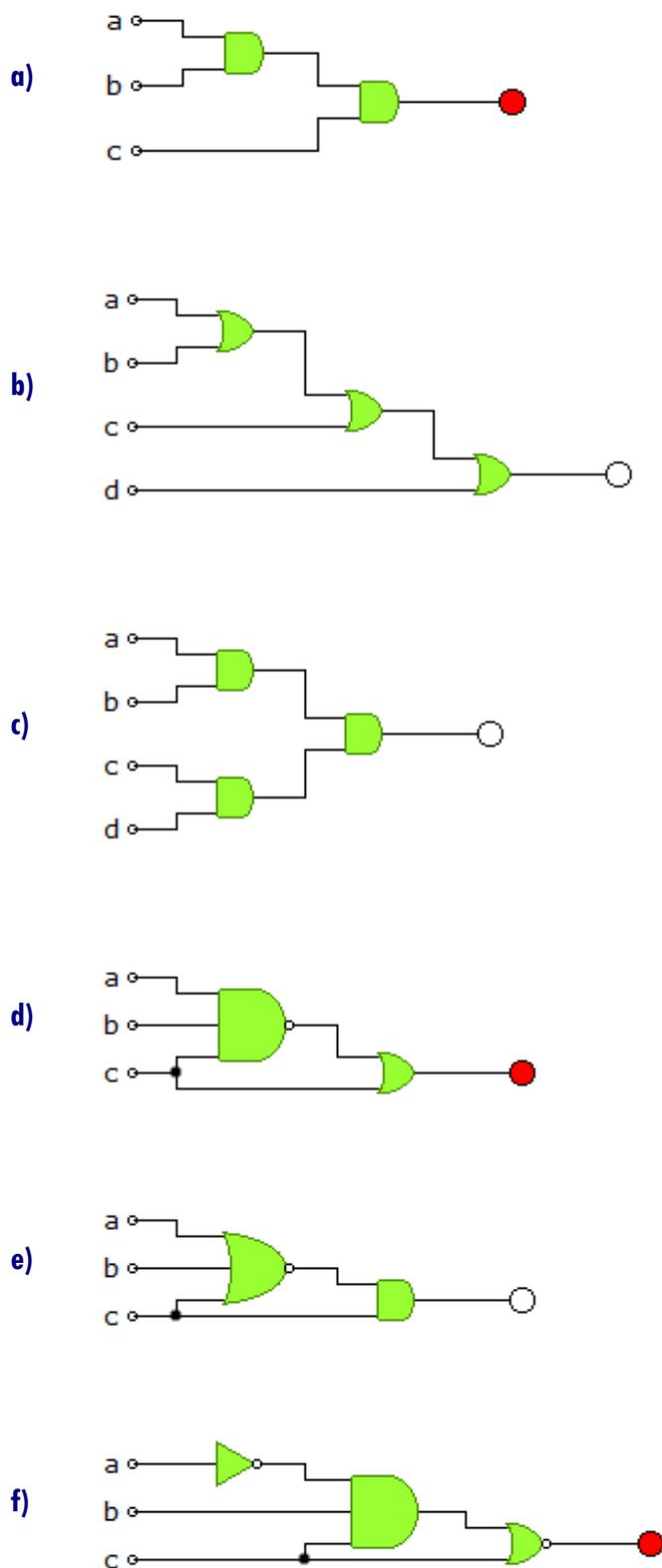
n)

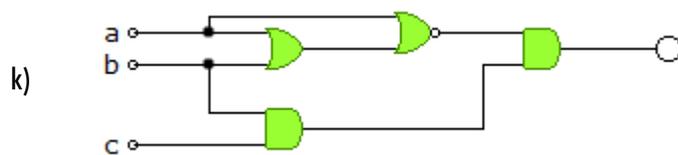
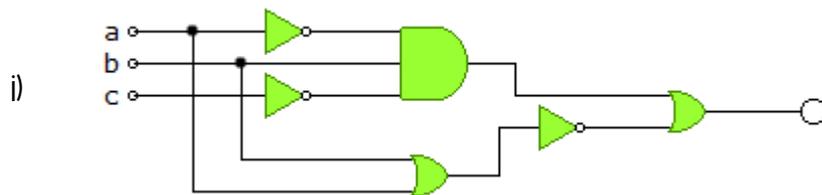
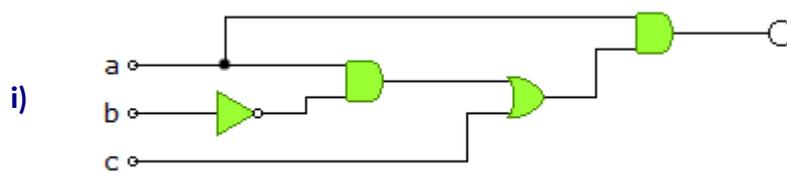
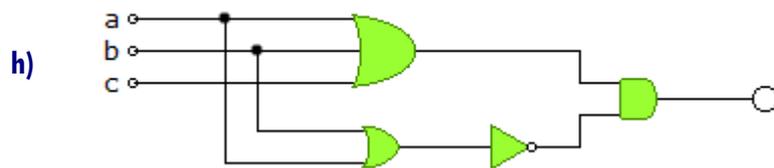
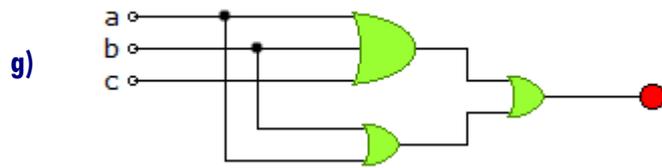
l)

o)

m)

17. Nombra las puertas lógicas y obtén la función lógica de salida de los siguientes circuitos.





18. Diseña el circuito lógico combinacional para abrir automáticamente la puerta de un comercio (Sensor interior A e exterior B).

19. Diseña el circuito lógico combinacional que active una alarma sonora cuando la temperatura y la humedad de una habitación sean demasiado elevadas.

20. Diseña el circuito lógico combinacional que active una alarma sonora cuando la temperatura y/o la humedad de una habitación sean demasiado elevadas.

21. Obtén la función lógica que permita decidir si se ve o no la televisión en una casa sabiendo que en el caso de que los dos padres estén de acuerdo esa será la decisión a tomar. Sólo en el caso de que los padres no estén de acuerdo, la decisión la tomará el hijo (A: madre; B: padre; C: hijo). Cuando la salida S sea 1 se verá la tele.

22. Diseña un circuito constituido por tres pulsadores (a, b y c) y una lámpara que se encenderá cuando se pulsen los tres pulsadores a la vez o sólo uno de ellos.

23. Diseña un circuito que conste de 3 variables de entrada y una de salida que toma el valor de 1 cuando el número representado a la salida sea par y mayor o igual a 6.

24. En un coche al abrir cualquiera de las cuatro puertas se activa un LED en señal de alarma. Obtén la función lógica para controlar el funcionamiento de la alarma.

25. Una habitación dispone de un sistema de alumbrado con 4 interruptores. El sistema se encenderá cuando el número de interruptores accionados sea impar. Obtén la tabla de verdad y la función lógica.