

TEMA 5. ELECTRÓNICA DIGITAL

1. INTRODUCCIÓN

Los ordenadores están compuestos de elementos electrónicos cuyas señales, en principio, son analógicas. Pero las señales que entiende el ordenador son digitales. Por eso hemos de transformar dichas señales en 0 y 1.

El sistema que sólo usa estos dos números se llama binario. El que nosotros usamos habitualmente se llama decimal porque tiene diez números.

Cada uno de los números del sistema binario, 0 y 1, recibe el nombre de **bit** y constituye la mínima cantidad de información que puede transmitirse. La siguiente cantidad es:

$$1 \text{ byte} = 8 \text{ bits}$$

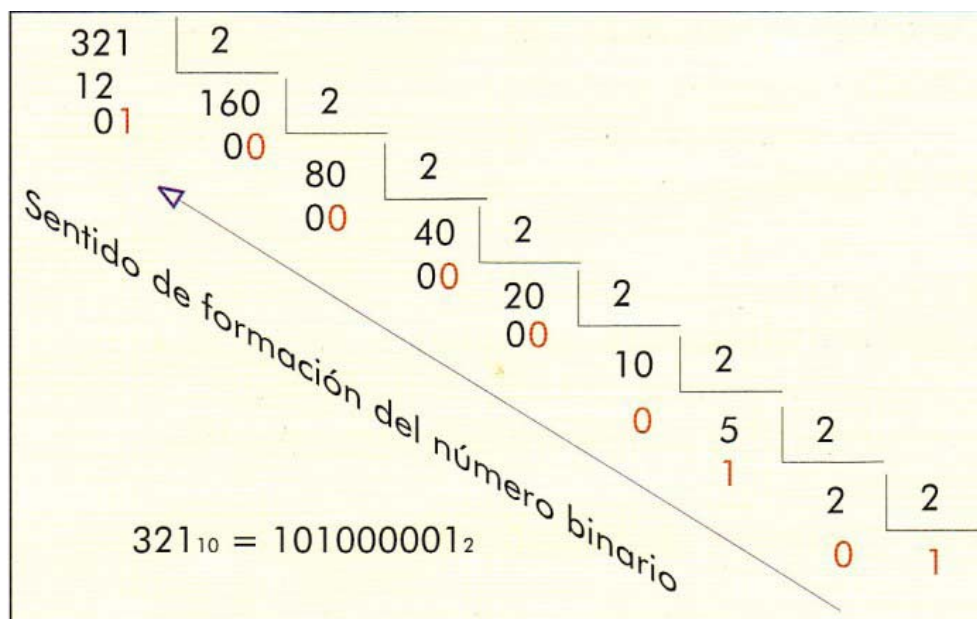
2. PASAR DEL SISTEMA BINARIO AL DECIMAL Y VICEVERSA

Vamos a pasar del sistema binario al decimal:

$$(1011001)_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 64 + 16 + 8 + 1 = (89)_{10}$$

$$(110'01)_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 4 + 2 + 1/4 = 6 + 0'25 = (6'25)_{10}$$

Vamos a pasar del sistema decimal al binario:



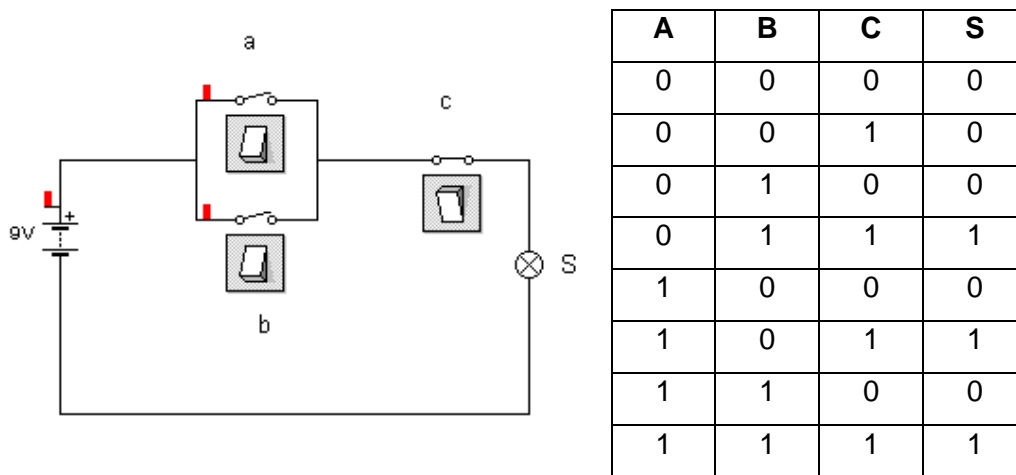
Dígitos decimales en binario:

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

3. TABLA DE VERDAD

La tabla de verdad de un circuito digital es una tabla en la que se representan todos los estados en que pueden encontrarse las entradas así como las salidas. En electrónica digital trabajamos sólo con señales binarias.

Si observamos el siguiente circuito eléctrico, podemos ver que su tabla de verdad es la que aparece al lado. Es decir, dependiendo de qué interruptor y cuantos se cierren (entradas), se encenderá la bombilla (salida) o no.



4. FUNCIÓN LÓGICA DE UN CIRCUITO

Es una expresión matemática que nos relaciona las salidas con las entradas y, a partir de ellas, podemos deducir como montar el circuito. Esta función se puede obtener de la tabla de verdad (**función canónica**). Por ejemplo, una función lógica puede ser la siguiente:

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

Por muy complicada que sea una función lógica siempre podrá expresarse como una suma de productos de sus variables, negadas o no (las negadas llevan una rayita encima y corresponden a los ceros). Sólo se eligen los términos en que **S** corresponde a un 1. Por ejemplo en la tabla de verdad del circuito anterior la función lógica es:

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

5. ÁLGEBRA DE BOOLE

Este algebra está pensada para trabajar con sistemas binarios como es el caso de los sistemas informáticos. El éxito de este álgebra es que:

1 – Muchos problemas tecnológicos pueden traducirse del sistema decimal al lenguaje binario.

2 – Podemos identificar 0 y 1 con dos estados físicos diferentes. Por ejemplo, un interruptor abierto (0) y un interruptor cerrado (1), una bombilla apagada (0) y una encendida (1), etc.

3 – Las operaciones booleanas de suma, multiplicación y negación se pueden hacer físicamente con circuitos eléctricos, neumáticos, hidráulicos, etc.

EL ÁLGEBRA DE BOOLE

El álgebra de Boole está formada por:

- **2 elementos:** los números 0 y 1.
- **Tres operaciones definidas así:**

– Multiplicación (\cdot):

$0 \cdot 0 = 0$	$0 \cdot 1 = 0$
$1 \cdot 0 = 0$	$1 \cdot 1 = 1$

– Suma (+):

$0 + 0 = 0$	$0 + 1 = 1$
$1 + 0 = 1$	$1 + 1 = 1$

– Negación ($\bar{}$):

$\bar{0} = 1$	$\bar{1} = 0$
---------------	---------------

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

- **Las siguientes propiedades:**

– Las operaciones suma y multiplicación son conmutativas:

$$x \cdot y = y \cdot x \quad x + y = y + x$$

– Las operaciones suma y multiplicación son asociativas:

$$(x + y) + z = x + (y + z)$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

– Las operaciones suma y multiplicación son distributivas una respecto a otra:

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

– El elemento negado satisface:

$$x + \bar{x} = 1 \quad x \cdot \bar{x} = 0$$

Operación	Forma de representarla	Postulados básicos
Suma	$Y = A + B$	$A + 0 = A$ $A + 1 = 1$ $A + A = A$ $A + \bar{A} = 1$
Multiplicación	$Y = A \cdot B$	$A \cdot 0 = 0$ $A \cdot 1 = A$ $A \cdot A = A$ $A \cdot \bar{A} = 0$
Complementación	$Y = \bar{A}$	$\bar{\bar{A}} = A$

Propiedades:

- Propiedad conmutativa
 $A + B = B + A$
 $A \cdot B = B \cdot A$
- Propiedad asociativa
 $A + B + C = A + (B + C)$
 $A \cdot B \cdot C = A \cdot (B \cdot C)$
- Propiedad distributiva
 $A(B + C) = A \cdot B + A \cdot C$
 $A + B \cdot C = (A + B) \cdot (A + C)$

Teoremas:

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

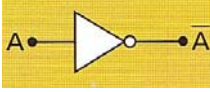
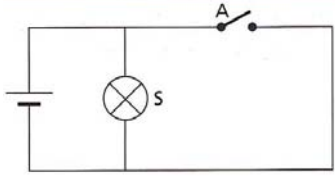
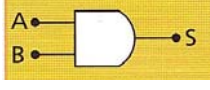
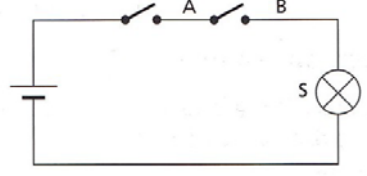

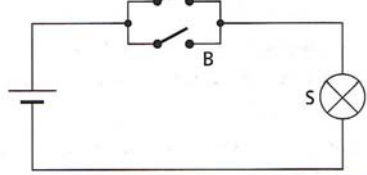
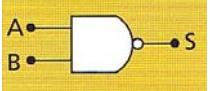
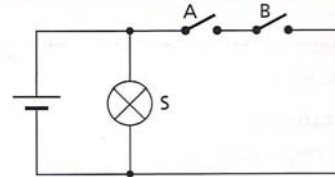
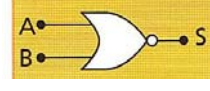
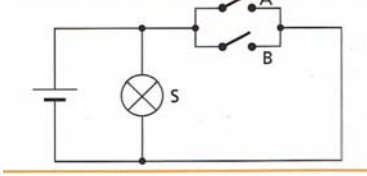
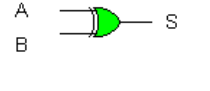
$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

6. PUERTAS LÓGICAS

Una puerta lógica es un circuito electrónico que proporciona unas señales digitales en su salida cuando a sus entradas se aplican también señales digitales. Las señales en la salida dependen de las señales de la entrada, es decir, pueden haber múltiples entradas pero a la salida sólo pueden asumirse los valores 0 y 1.

Las puertas lógicas básicas corresponden a las operaciones definidas en el álgebra de Boole: AND (multiplicación), OR (suma) y NOT (negación). Además de éstas, existen otras puertas como la NAND, que es la negación de la AND; la NOR, que es la negación de la OR y la XOR, que es la OR exclusiva.

Vamos a ver estas puertas, sus símbolos, sus tablas de verdad, sus circuitos eléctricos equivalentes y sus funciones lógicas:

Puerta lógica	Símbolo	Función lógica	Tabla de verdad	Circuito eléctrico equivalente															
NOT		$S = \bar{A}$	<table border="1"> <thead> <tr> <th>A</th> <th>$S = \bar{A}$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	$S = \bar{A}$	0	1	1	0										
A	$S = \bar{A}$																		
0	1																		
1	0																		
AND		$S = A \cdot B$	<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>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	S	0	0	0	1	0	0	0	1	0	1	1	1	
A	B	S																	
0	0	0																	
1	0	0																	
0	1	0																	
1	1	1																	
OR		$S = A+B$	<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>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	S	0	0	0	1	0	1	0	1	1	1	1	1	
A	B	S																	
0	0	0																	
1	0	1																	
0	1	1																	
1	1	1																	
NAND		$S = \overline{A \cdot B}$	<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>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	S	0	0	1	1	0	1	0	1	1	1	1	0	
A	B	S																	
0	0	1																	
1	0	1																	
0	1	1																	
1	1	0																	
NOR		$S = \overline{A + B}$	<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>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	S	0	0	1	1	0	0	0	1	0	1	1	0	
A	B	S																	
0	0	1																	
1	0	0																	
0	1	0																	
1	1	0																	
XOR		$S = A \oplus B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>F</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>0</td> </tr> </tbody> </table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0	
A	B	F																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	

7. CONSTRUCCIÓN DE CIRCUITOS DIGITALES CON PUERTAS LÓGICAS

Aunque la tabla de verdad puede hacerse de cualquier circuito ya conocido, lo habitual cuando diseñamos sistemas electrónicos es crear una tabla a partir de unas condiciones que queremos cumplir y, a partir de ella, determinar cómo ha de montarse el circuito correspondiente.

Emplearemos un ejemplo sencillo para entender cómo se construyen circuitos digitales que resuelven problemas concretos.

Implementar con puertas lógicas un sistema para determinar si un n° entre 0 y 7 es número primo

1. Identificar las entradas y salidas: en los enunciados se dan las condiciones a partir de las cuáles identificaremos las entradas y salidas. En el ejemplo, como debemos obtener números entre 0 y 7 debemos emplear 3 entradas ($2^3 > 7$) 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

3. Obtener la función lógica a partir de la tabla de verdad.

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	⇒ $f = a \cdot b \cdot c$

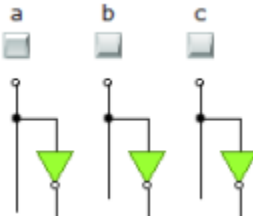
4. Se obtiene directamente a partir de la tabla de verdad sumando todos los productos lógicos correspondientes a las salidas que dan una salida igual a 1 (despreciamos los que corresponden a una salida igual a 0). Las entradas con 0 se consideran negadas, y las entradas con 1 no negadas.

5. La función lógica de nuestro ejemplo será:

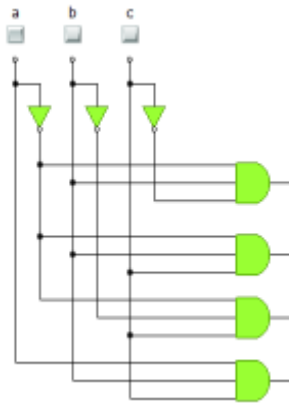
$$S = \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot c$$

6. Diseñar el circuito empleando puertas lógicas a partir de la función:

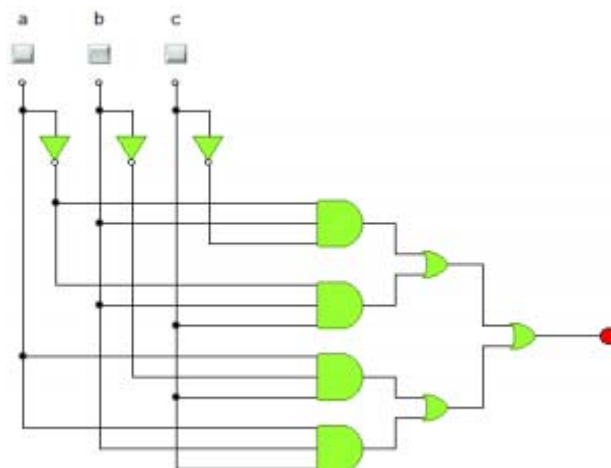
a. Para ello se dibujarán tantos terminales lógicos de entrada (inputs) como variables de las que dependa la función (tres en nuestro ejemplo). Estos terminales deberían incluir, en caso necesario) sus valores negados utilizando puertas NOT.



b. 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.



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



EJEMPLO DE EJERCICIO PROPUESTO DE UN PROBLEMA PRÁCTICO

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

⇒ $f = \bar{a} \cdot b \cdot c$

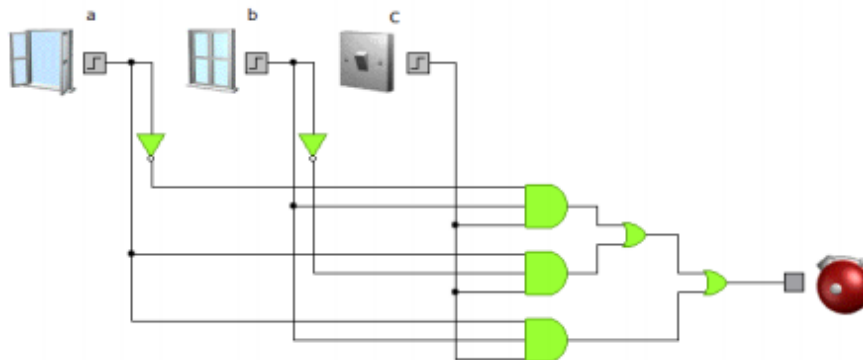
⇒ $f = a \cdot \bar{b} \cdot c$

⇒ $f = a \cdot b \cdot c$

}

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

→ Finalmente implementamos el circuito:



8. CIRCUITOS INTEGRADOS

Los Circuitos Integrados (I.C. Integrated Circuits) son circuitos que están formados por componentes electrónicos (transistores, diodos, resistencias, condensadores....) fabricados en una placa 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 caber millones de transistores además de resistencias, condensadores, diodos, etc. Un ejemplo muy bueno sería el microprocesador de un ordenador. Así, el microprocesador Intel Pentium Core i7 tiene más de 700 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 y CMOS

Ejemplos de Circuitos integrados

