TEMA 3 - ELECTRÓNICA DIGITAL

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 0s y 1s.

El sistema que sólo usa estos dos números se llama binario. El que nosotros usamos habitualmente se llama decimal porque tiene diez caracteres, del 0 al 9.

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 byte = 8 bits

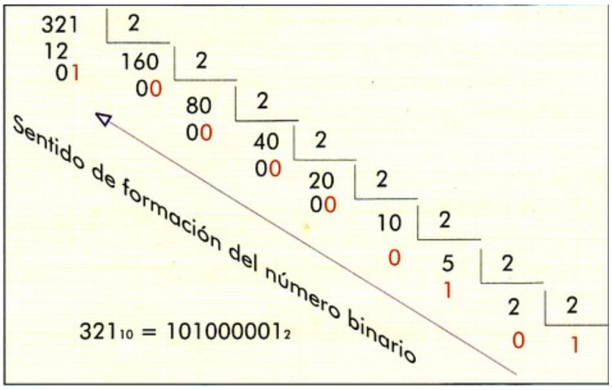
1.- PASAR DEL SISTEMA BINARIO AL DECIMAL Y VISEVERSA

Vamos a pasar del sistema binario al decimal:

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

$110'01 = 1{\cdot}2^2 + 1{\cdot}2^1 + 0{\cdot}2^0 + 0{\cdot}2^{-1} + 1{\cdot}2^{-2} = 4 + 2 + 1/4 = 6 + 0'25 = 6'25$

Vamos a **pasar del sistema decimal al binario**:



Vamos a **pasar del sistema binario al decimal:**

$\mathbf{(1010.1100.0101)_2} = 1x2^{11}+0x2^{10}+1x2^9+0x2^8+1x2^7+1x2^6+0x2^5+0x2^4+0x2^3+1x2^2+0x2^1+1x2^0=$

$= 2048+0+512+0+128+56+0+0+0+4+0+1 = \mathbf{(2749)_{10}}$

1.- Pasa a binario los siguientes valores decimales:

406, 321, 128, 34, 11

2.- Pasa a decimal los siguientes valores binarios:

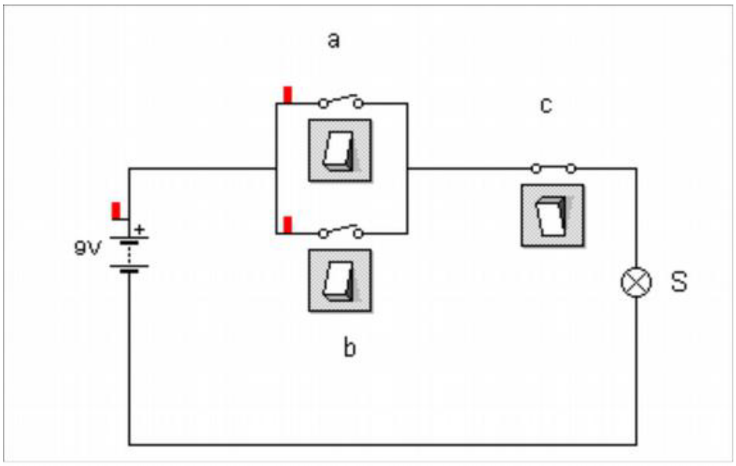
1101.001, 0011.0000.0110, 0101.1110, 0100.1111.1011

2.- 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. Es decir, se ponen todas las combinaciones posibles que pueden hacerse con el número de variables que se tengan de modo que siempre se tendrán $\mathbf{2^n}$ combinaciones posibles, siendo n el número de variables. Por ejemplo si tenemos 2 variables tendremos 2^2, es decir, 4 combinaciones posibles, con 3 variables tendremos 2^3, es decir, 8 combinaciones posibles y con 4 variables tendremos 2^4, es decir 16, combinaciones posibles etc…

En electrónica digital trabajamos sólo con señales binarias.

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



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

3.- FUNCIÓN LOGICA 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 la obtendremos de la tabla de verdad . Por ejemplo, una función lógica puede ser la siguiente:

$$\mathbf{S = a\cdot b + a\cdot b□}$$

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 = a·b·c + a·b·□ c + a·b·c

4.- ÁLGEBRA DE BOOLE

Este algebra está pensado 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 – La operaciones booleanas de suma, multiplicación y negación se pueden hacer físicamente con circuitos eléctricos, neumáticos, hidráulicos, etc.

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

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

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

El Álgebra de Boole

El álgebra de Boole está formada por:

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

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

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:

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

  - Las operaciones suma y multiplicación son asociativas:

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

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

  - Las operación 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:

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

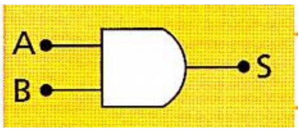
5.- PUERTAS LÓGICAS

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 señalen de la entrada, es decir, puede haber múltiples entradas pero a la salida sólo pueden asumirse los valores 0 y 1.

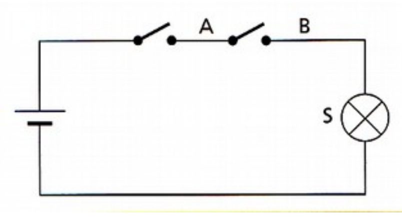
Las puertas lógicas básicas corresponden a las operaciones definidas en el álgebra de Boole AND (multiplicación), OR (suma) y NOT (negación). Además de éstas, existen otras puertas como la NAND; que es la negación de la AND, la NOR, que es la negación de la OR, la XOR, que es la OR exclusiva y la XNOR,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

Vamos a ver estas puertas, sus símbolos, sus tablas de verdad, sus circuito equivalente y sus funciones lógicas:

5.1.- Puerta **AND(Producto lógico):** La salida de esta puerta es el producto de las entradas. Será "1", sólo cuando todas sus entradas estén a "1". En el caso de dos entradas, A y B, sólo cuando A **y(AND)** B estén a "1".

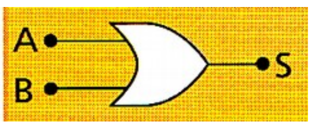


A	B	S
0	0	0
1	0	0
0	1	0
1	1	1

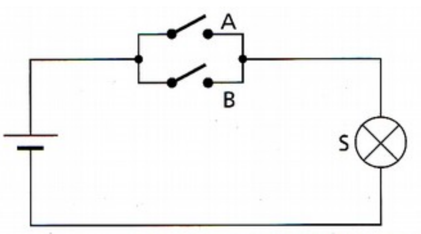


$$S = A \cdot B$$

5.2.- Puerta **OR(Suma lógica):** Devuelve la suma de las señales de entrada: Será "1", siempre que alguna de sus entradas esté a "1". En el caso de dos entradas, A y B, sólo cuando A **o (OR)** B estén a "1".

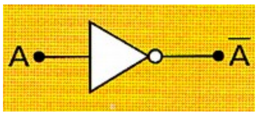


A	B	S
0	0	0
1	0	1
0	1	1
1	1	1

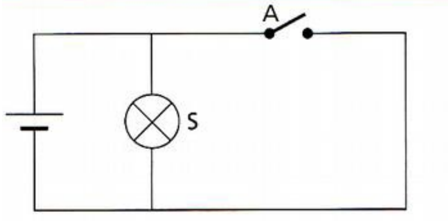


$$S = A + B$$

<u>5.3.- Puerta NOT:</u> esta puerta devuelve a la salida la señal inversa de la de entrada.

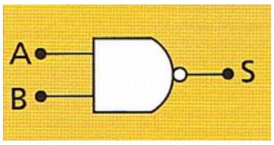


A	$S = \overline{A}$
0	1
1	0

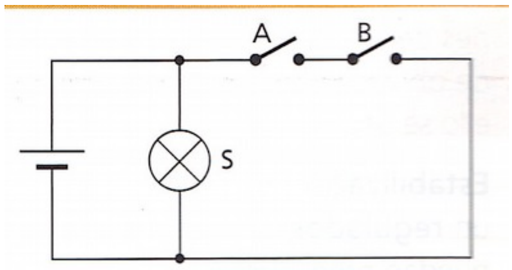


A = A

<u>5.4.- Puerta NAND:</u> la salida es la inversa del producto de las entradas:

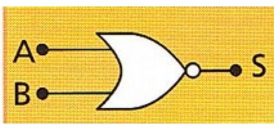


A	B	S
0	0	1
1	0	1
0	1	1
1	1	0

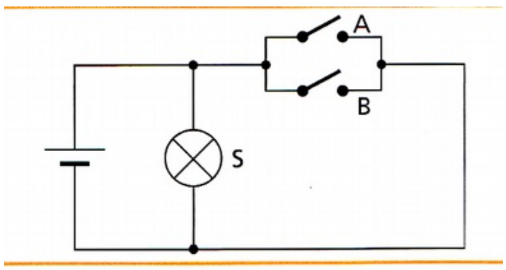


$S = \overline{A \cdot B}$

<u>5.5.- Puerta NOR:</u> nos da a la salida la inversa de la puerta OR, es decir, de la suma.



A	B	S
0	0	1
1	0	0
0	1	0
1	1	0

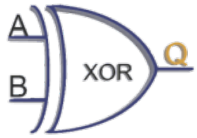


$S = \overline{A + B}$

Estas próximas puertas son únicamente de dos entradas, las anteriores pueden ser de dos o más entradas, salvo la NOT que es de una entrada.

<u>5.6.- Puerta XOR (OR EXCLUSIVA):</u> nos da a la salida un "1" si las entradas son distintas.

SÍMBOLO



EXPRESIÓN

$$Q = A \oplus B$$
$$= A \cdot \bar{B} + \bar{A} \cdot B$$

TABLA DE VERDAD

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

<u>5.7.-</u> <u>Puerta **XNOR (NOR EXCLUSIVA)**:</u> nos da a la salida un "1" si las entradas son iguales.

SÍMBOLO

EXPRESIÓN

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

TABLA DE VERDAD

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

6.- 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 nº entre 0 y 7 es número 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 ($2^3 > 7$) 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).

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	$f = a\cdot b\cdot c$

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.**

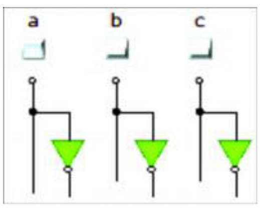
   Se obtiene directamente a partir de la tabla de verdad sumando todos los productos lógicos correspondientes a las salidas que dan una salida igual a 1 (despreciamos los que corresponden a una salida igual a 0). Las entradas con 0 se consideran negadas, y las entradas con 1 no negadas.

   La función lógica de nuestro ejemplo será:

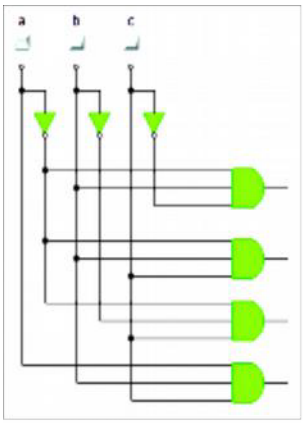
$$F_1 = \bar{a}\cdot b\cdot \bar{c} + \bar{a}\cdot b\cdot c + a\cdot \bar{b}\cdot c + a\cdot b\cdot c$$

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

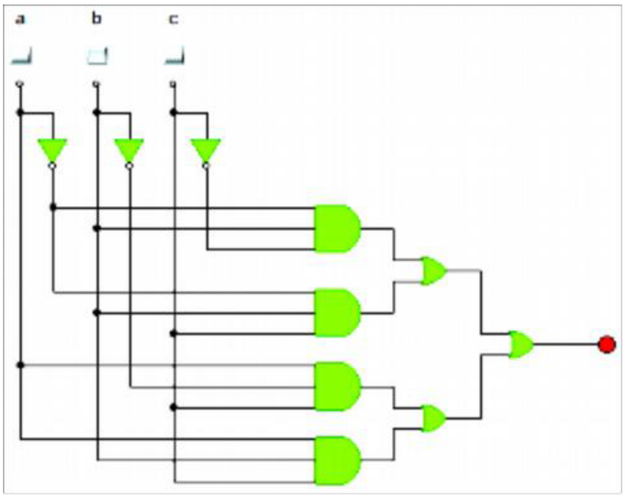
   Para ello se dibujarán tantos terminales lógicos de entrada (inputs) como variables de las que dependa la función (tres en nuestro ejemplo). Estos terminales deberían incluir, en caso necesario) sus valores negados utilizando puertas NOT.



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.



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:

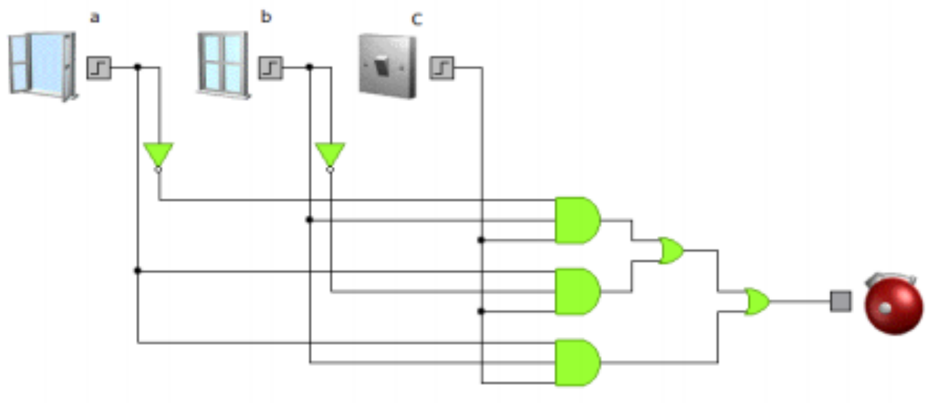
- ✗ Ventanas: cerradas (0), abiertas (1)
- ✗ Interruptor: abierto (0), cerrado (1)
- ✗ Alarma sonora : inactiva (0), activa (1)

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

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

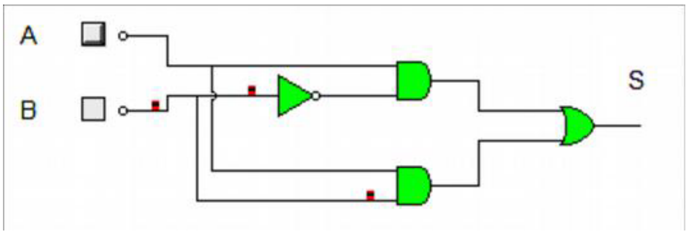
➔ Finalmente implementamos el circuito:



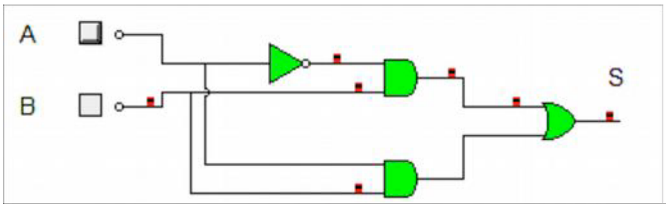
Ejercicios

1 – Elaborar la tabla de verdad y la función lógica de los siguientes circuitos:

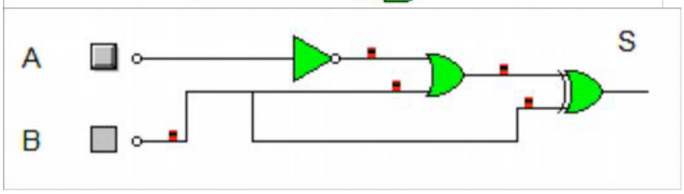
a)



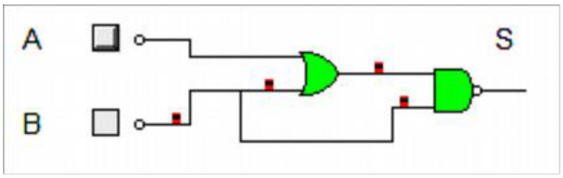
b)



c)

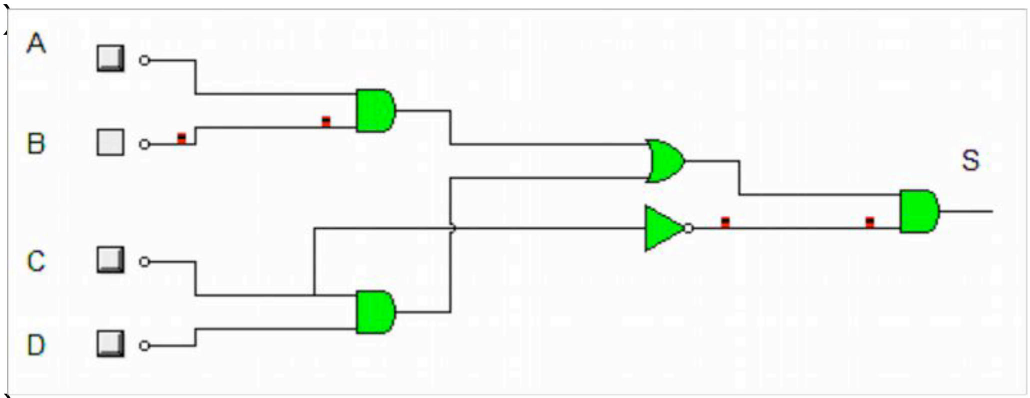


d)

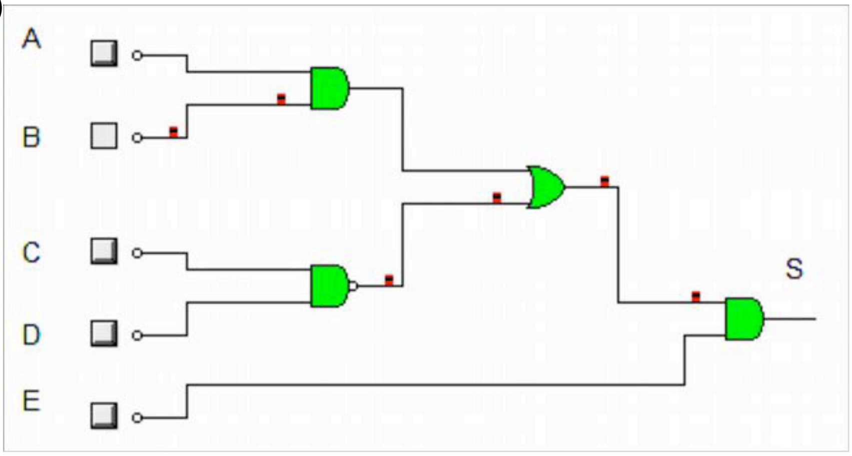


2 – Halla la función lógica de los siguientes circuitos y el valor final de S si A, C, D, F = 1; B, E, G, H = 0:

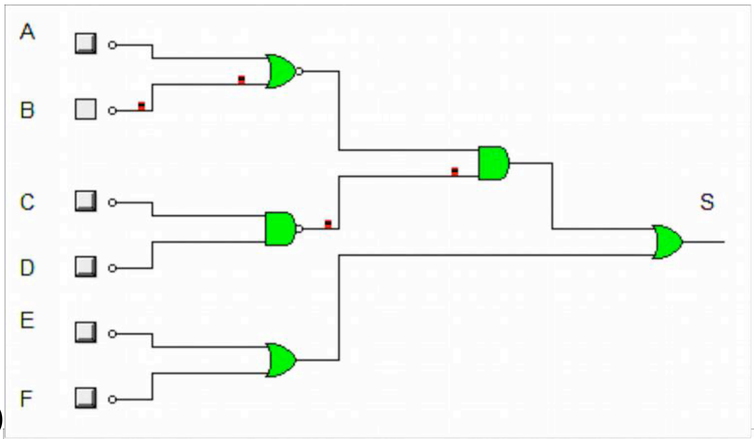
a)



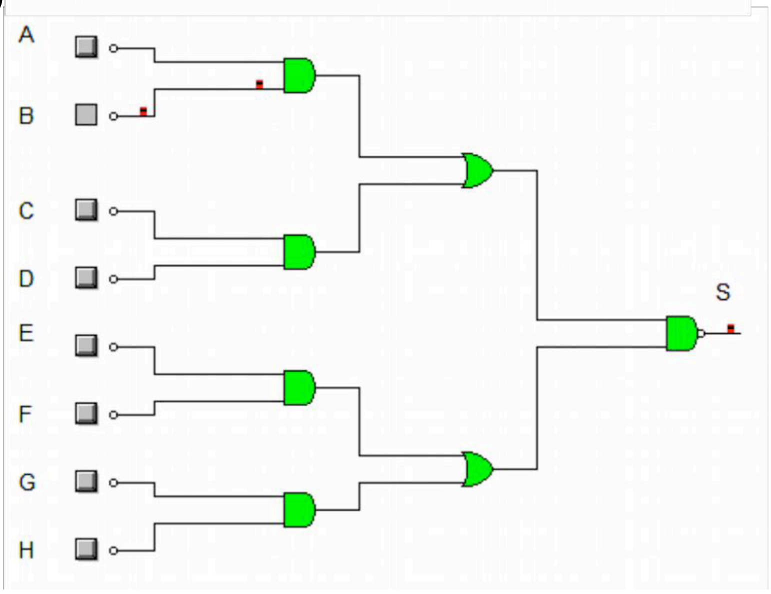
b)



c)

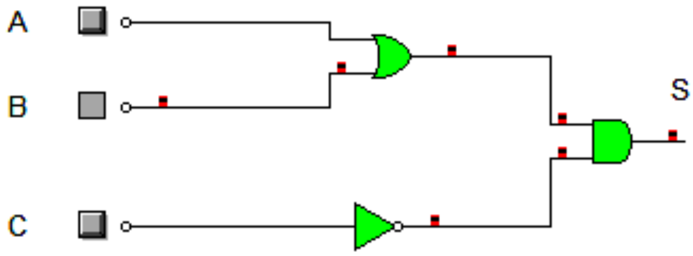


d)

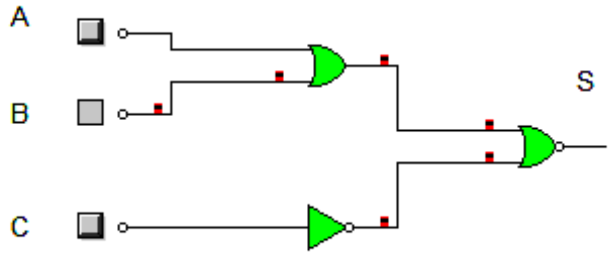


3 – Elaborar la tabla de verdad y la función lógica de los siguientes circuitos:

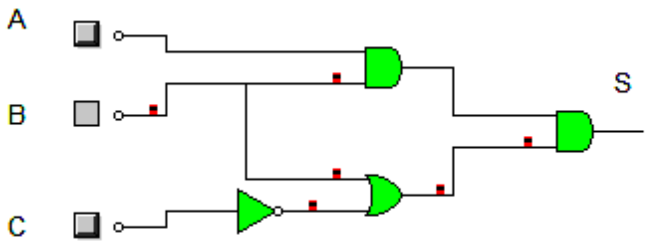
a)



b)



c)



4 – Saca la función lógica y el circuito a partir de la tabla de verdad:

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