TÍTULO DEL DOCUMENTO: Electrónica Digital (Parte 1)

Dr. C. Evaristo González Milanés[1], Ing. Carlos Molina[2]

1. Universidad de Matanzas "Camilo Cienfuegos", Vía Blanca Km.3, Matanzas, Cuba.

2. Profesor Adjunto Universidad de Matanzas "Camilo Cienfuegos", Vía Blanca Km.3, Matanzas, Cuba.

Resumen.

El trabajo tiene como objetivo describir los principales elementos que caracterizan a la Electrónica Digital, para lo cual se realiza una introducción a los Sistemas digitales y se explica brevemente las características del sistema de numeración binario, se exponen los símbolos y las tablas de la verdad de las Compuertas lógicas fundamentales y se enuncian los teoremas y postulados del álgebra de Boole que permite llevara acabo la Simplificación de funciones de conmutación. Seguidamente es explicado el procedimiento para la Construcción de circuitos con compuertas lógicas, así como las Familias lógicas que las implementan y finalmente se explican un grupo de circuitos básicos que cuyo conocimiento resulta esencial para la comprensión de la temática tratada, dentro de los cuales se encuentran los Circuitos aritméticos y los Multivibradores en general, particularizándose en los diferentes tipos de biestables que son de uso generalizado en la construcción de circuitos digitales.

***Palabras claves**: Electrónica Digital; Algebra de Boole; Dispositivos electrónicos; Compuertas lógicas*

1.- Sistemas digitales. Numeración binaria.

Los circuitos digitales operan en el sistema numérico binario, que implica que todas las variables de circuito asumen dos niveles discretos de voltaje, a uno de los cuales se le asigna el símbolo 1, mientras que al otro se le representa por 0. Las operaciones básicas que debe ejecutar el sistema son pocas y relativamente sencillas, pero pueden repetirse muchas veces.

El álgebra utilizada para resolver problemas y procesar la información en los sistemas digitales se denomina álgebra de Boole, basada sobre la lógica más que sobre el cálculo de valores numéricos reales. El álgebra booleana considera que las proposiciones lógicas son verdaderas o falsas, según el tipo de operación que describen y si las variables son verdaderas o falsas. Verdadero corresponde al valor digital 1, mientras que falso corresponde a 0.

El sistema numérico decimal no es práctico para los sistemas de conmutación y los sistemas digitales. Por ello es necesario utilizar un sistema de numeración que se ajuste a las necesidades reales de la electrónica digital.

En general cualquier sistema numérico se puede representar por un polinomio.

$$\mathbf{N_b = \sum_{i=-p}^{q-1} a_i b^i}$$

Donde b es la base del sistema.
Los dígitos a_{q-1} se le denominan más significativos.
Los dígitos a_{-p} se le denominan menos significativos.

La base es el número de símbolos diferentes, o guarismos, necesarios para representar un número cualquiera, de los infinitos posibles, en el sistema. A lo largo de la historia se han usado multitud de sistemas numéricos. En realidad, cualquier número mayor que 1 puede ser utilizado como base. Algunas civilizaciones usaban sistemas basados en los números 3, 4 ó 5. Los babilonios utilizaron el sistema sexagesimal, basado en el número 60, y los romanos (en ciertas aplicaciones) el sistema duodecimal, con el número 12 como base. Los mayas utilizaban el sistema vigesimal, basado en el número 20. El sistema decimal es utilizado hoy de forma universal (con la excepción de los ordenadores o computadoras), y el mismo necesita diez símbolos diferentes o dígitos para representar un número y es, por tanto, un sistema numérico en base 10.
Por tanto, la base es fundamental en el sistema numérico, y por ello tenemos que:

b=2 sistema binario cuyos dígitos son (0,1).
b=8 sistema octal cuyos dígitos son (0,1,2,3,4,5,6,7).
b=10 sistema decimal cuyos dígitos son (0,1,2,3,4,5,6,7,8,9).
b=16 sistema Hexadecimal cuyos dígitos son : (0,1,2,3,4,5,6,7,8, 9, A, B, C, D, E, F).

El número 126.35_{10} puede ser representado por el polinomio.

$$\sum_{i=-p}^{q-1} = 1*10^2+2*10^1+6*10^0+3*10^{-1}+5*10^{-2} = 100+20+6+0.3+0.05 = 126.35_{10}$$

A fin de crear un sistema digital de procesamiento, lo primero que hay que hacer es escoger el sistema de numeración o "base" que será utilizada. Se selecciona b= 2 debido a que resulta más fácil construir circuitos electrónicos que tengan dos estados bien diferenciados entre sí y que pueden ser identificados con los números 0 y 1, únicos existentes en la numeración binaria. A cada uno de estos números se le denomina bit, donde el cero lógico, es un rango de voltaje cercano a cero, y el uno lógico es un rango de voltaje alejado de cero Volt como se muestra en la figura. Cuando el sistema es representado así se está trabajando con lógica positiva, y si invertimos los rangos de voltajes donde el uno lógico esta próximo a cero Volt, estamos en presencia de la lógica negativa.

El sistema binario desempeña un importante papel en la tecnología de los ordenadores. Los primeros 20 números en el sistema en base 2 son 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010, 10011 y 10100. Cualquier número se puede representar en el sistema binario, como suma de varias potencias de dos. Por ejemplo, el número 10101101 representa, empezando por la derecha, $(1 \times 2^0) + (0 \times 2^1) + (1 \times 2^2) + (1 \times 2^3) + (0 \times 2^4) + (1 \times 2^5) + (0 \times 2^6) + (1 \times 2^7) = 173_{10}$.

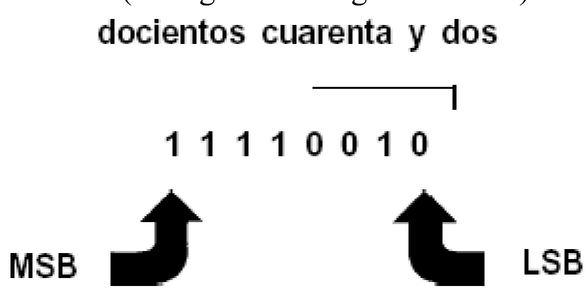
<u>Conversión de un sistema decimal a binario</u>. Este procedimiento puede llevarse a efecto de una manera muy rápida con el empleo de una calculadora electrónica científica.
Así puede encontrarse que $\mathbf{7_{10}}$ puede escribirse en binario como:

$$\sum_{i=-p}^{q-1} = \mathbf{0}*2^3+\mathbf{1}*2^2+\mathbf{1}*2^1+\mathbf{1}*2^0 = \mathbf{0111_2}$$

De igual forma $\mathbf{44_{10}}$ será:

$$\sum_{i=-p}^{q-1} = \mathbf{1}*2^5+\mathbf{0}*2^4+\mathbf{1}*2^3+\mathbf{1}*2^2+\mathbf{0}*2^1+\mathbf{0}*2^0 = \mathbf{101100_2}$$

Así como en el sistema decimal se hablaba de las posiciones más y menos significativas, en el sistema binario se habla de bit más significativo *MSB* (del ingles "most significant bit") y del bit menos significativo *LSB* (del ingles "less significant bit").



Para simplificar la descripción de procesos en donde se manejan grupos de bits, es común agruparlos.

Dependiendo del número de bits que se escojan para formar la agrupación, estos grupos se denominan con algunos de los siguientes nombres:

Nibble 4 bits
Byte 8 bits
Word 16 bits

Un número binario que posea muchos bits es difícil de recordar y teclear por lo que se hizo necesario utilizar otros sistemas que simplificarán esta operación.

Conversión del sistema binario al octal. Para convertir al sistema octal (base 8), se agrupan en 3 los dígitos binarios, simplificando notablemente la notación binaria. Como ya fue expresado anteriormente, se usa los dígitos (0, 1, 2, 3, 4, 5, 6,7). Así tenemos:

011 111 000 100_2 Binario

3 7 0 4_8 Octal

$\mathbf{3}*8^3+\mathbf{7}*8^2+\mathbf{0}*8^1+\mathbf{4}*8^0 = \mathbf{3}*512+\mathbf{7}*64+\mathbf{0}*8+\mathbf{4}*1 = \mathbf{1988_{10}}$

Conversión del sistema binario al sistema hexadecimal. Si se posee un número binario equivalente a un número decimal de cifras altas, el proceso de conversión es demasiado largo, y por ello, la mayoría de las microcomputadoras, usan el sistema hexadecimal. Este sistema según se expresó anteriormente, tiene 16 dígitos, los cuales son (0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F). El número 2EF del sistema hexadecimal es el número $(2 \times 16^2) + (14 \times 16^1) + (15 \times 16^0) = 751$ en el sistema decimal.

También puede verse que el número $1234_{16} = 4660_{10}$ pues:

$$1 * 16^3 \quad + \quad 2 * 16^2 \quad + \quad 3 * 16^1 \quad + \quad 4 * 16^0$$

$$4096 + 512 + 48 + 4 = 4660_{10}.$$

De igual forma el $2C6E_{16} = 11374_{10}$ ya que:

$\mathbf{2}*16^3+\mathbf{C}*16^2+\mathbf{6}*16^1+\mathbf{E}*16^0 = \mathbf{2}*4096+\mathbf{C}*256+\mathbf{6}*16+\mathbf{E}*1$
$\mathbf{2}*4096+\mathbf{12}*256+\mathbf{6}*16+\mathbf{14}*1 = 11374_{10}$

Entonces la equivalencia entre el binario y el hexadecimal puede verse seguidamente. El 1001 1110 en binario será el 9E hexadecimal pues:
1001 1110_2
9 E_{16} y el 2C6E = 0010 1100 0110 1110,

2 C 6 E_{16}
0010 1100 0110 1110_2

Números BCD.

La conversión manual binaria a decimal es bastante difícil y en calculadoras, juegos electrónicos y los instrumentos digitales en lo que es común la entrada y salida de números en notación decimal, se emplea un código especial para presentar esta notación. A este código se le denomina BCD. Veamos algunos ejemplos y la tabla que permite la conversión entre ellos:

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

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

2.- COMPUERTAS LÓGICAS Y OPERACIONES LOGICAS FUNDAMENTALES.

Los ordenadores o computadoras utilizan la lógica digital para efectuar operaciones. La lógica digital implica tomar sucesivas decisiones de verdadero o falso, que se representan como 1 y 0, respectivamente. Los circuitos lógicos, que están en el corazón de los chips de la computadora, están diseñados para tomar series de decisiones de este tipo a través de circuitos denominados compuertas. Éstas están diseñadas y organizadas de tal forma que pueden tomar diferentes tipos de decisiones acerca de las entradas que reciben. Los valores individuales de entrada y de salida son siempre verdaderos o falsos, y se transmiten a través del circuito en forma de diferentes tensiones.

Los circuitos digitales se clasifican a su vez en dos grupos, de acuerdo al tratamiento que hagan de la información. Cuando el esquema solo es capaz de transmitir una información mediante la combinación lógica de varias entradas, el circuito es combinacional, en tanto que si es posible almacenar la información o una parte de ella, el circuito es de tipo secuencial.

Los esquemas combinacionales se realizan utilizando cualquiera de las compuertas lógicas fundamentales o combinaciones de ellas Las compuertas lógicas básicas son las siguientes: NOT, AND y OR

<u>Compuerta lógica NOT o Inversora (negación).</u>

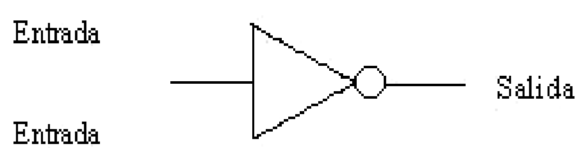
Presenta una entrada y una salida y tiene como característica de que la salida se encuentra siempre en el estado lógico contrario (inverso) al de la entrada.

Entrada (X)	Salida (T)
0	1
1	0

La tabla anterior, nos muestra las posibles combinaciones que pueden existir en la entrada y la salida de la compuerta y recibe el nombre de tabla de la verdad de dicha compuerta.

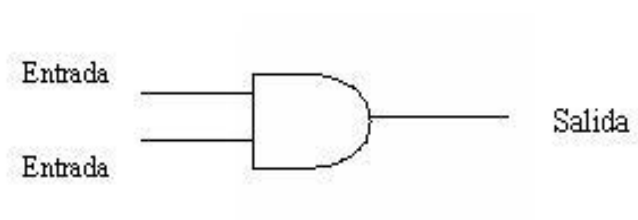
La simbología de la compuerta se muestra a continuación:

COMPUERTA LÓGICA NOT (SIMBOLOGÍA)



Compuerta LÓGICA AND:

Puede presentar varias entradas y una única salida. Realiza la multiplicación lógica de las señales de entrada. En el caso más simple posee dos entradas A y B por lo que la salida será el producto lógico de estas dos entradas (A*B) y podemos definirla que tiene un “1” a la salida sí y solo sí, sus dos entradas son “1”. Entonces:



T = X * Y

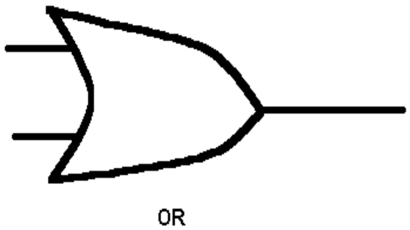
X	Y	T
0	0	0
0	1	0
1	0	0
1	1	1

Compuerta LÓGICA OR:

Puede presentar varias entradas y una única salida. Realiza la suma lógica de las señales de entrada. En el caso más simple posee dos entradas A y B por lo que la salida será la suma lógica de estas dos entradas (A + B) y podemos definirla que tiene un “0” a su salida si y solo si sus dos entradas son “0” (A + B). Entonces:

T = X + Y

Compuertas lógicas combinadas.



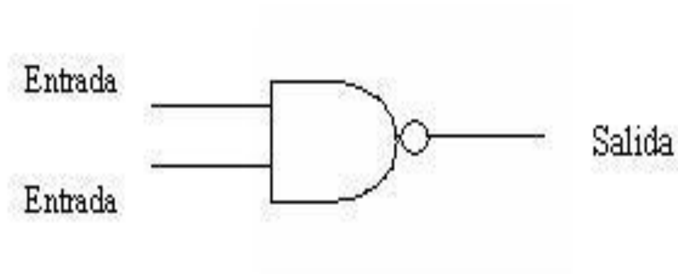
X	Y	T
0	0	0
0	1	1
1	0	1
1	1	1

Se obtienen a partir de la combinación de las compuertas lógicas básicas.

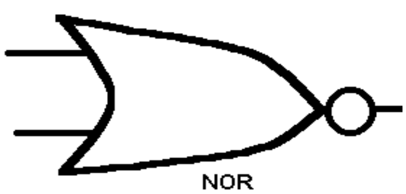
Compuerta LÓGICA NAND: No es más que la negación de la operación lógica AND y se define un “0” a su salida si y solo si todas sus entradas son “1”

T = $\overline{\mathbf{X * Y}}$

X	Y	T
0	0	1
0	1	1
1	0	1
1	1	0



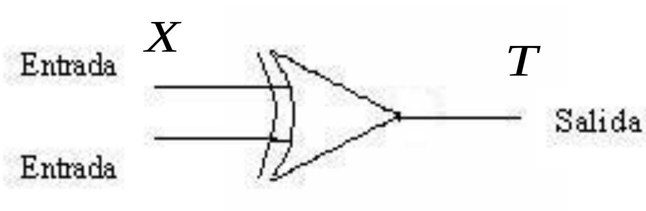
Compuerta LÓGICA NOR: Se define porque tiene un “1” lógico a su salida si y solo si sus dos entradas son “0”.



X	Y	T
0	0	1
0	1	0
1	0	0
1	1	0

T = $\overline{\mathbf{X + Y}}$

Compuerta LÓGICA OR EXCLUSIVO: Se define porque tiene un “1” a su salida si y solo si una de sus entradas tiene valor “1” $(\bar{A}B + AB)$



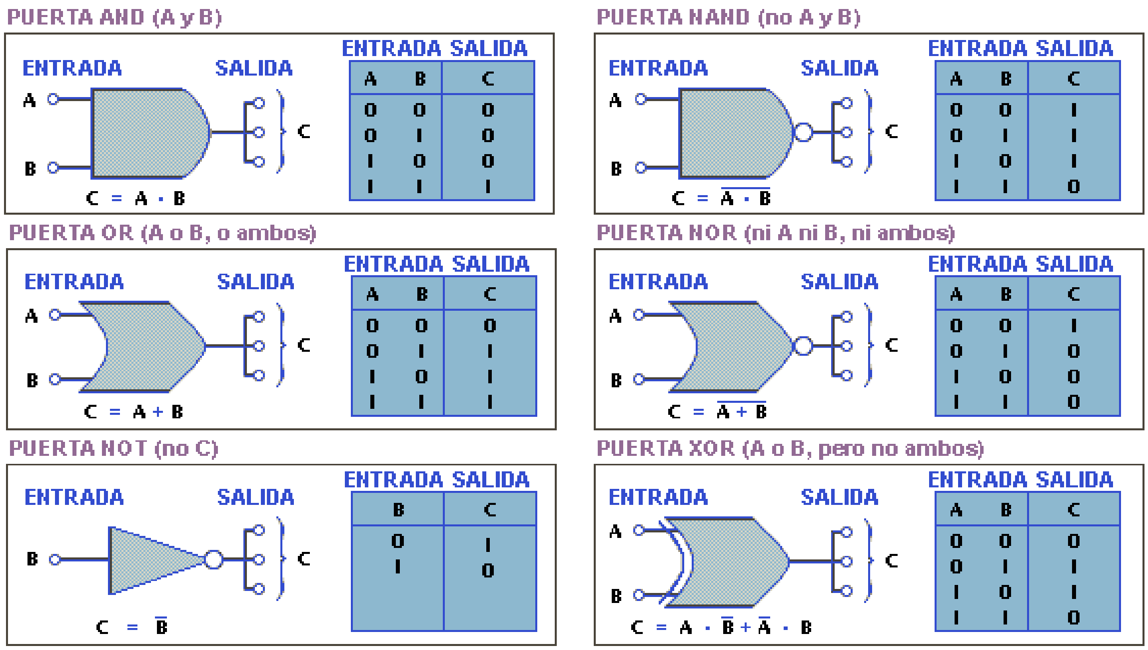
T = $X\bar{Y} + \bar{X}Y$

T = $X \oplus Y$

X	Y	T
0	0	0
0	1	1
1	0	1

1 1	0

A continuación se presenta un resumen de los símbolos, operaciones lógicas y las tablas de la verdad de las compuertas AND, OR, NAND, NOR, NOT Y OR EXCLUSIVO (XOR).



3.- Algebra de conmutación.
La acción de los circuitos lógicos se puede comprender mediante la lógica booleana. Los fundamentos del álgebra de conmutación es la base matemática del álgebra de Boole.

El álgebra booleana es sencilla una vez que uno se acostumbra a ella, pero cuesta un poco conseguirlo. El conjunto de teoremas e identidades que se presenta debajo resulta fácil de comprender.

Teoremas

1-) $X \cdot 0 = 0$ $\quad X + 0 = X$

2-) $\overline{X} \cdot X = 0$ $\quad X + \overline{X} = 1$

3-) $X \cdot X = X$ $\quad X + X = X$

4-) $X \cdot 1 = X$ $\quad X + 1 = 1$

5-) $X \cdot Y \cdot Z = X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$ Ley Conmutativa.

6-) $X + Y + Z = X + (Y + Z) = (X + Y) + Z$ Ley Conmutativa.

7-) $X \cdot Y + X \cdot \overline{Y} = X$

8-) $(X + Y) \cdot (X + \overline{Y}) = X$

9-) $X + X \cdot Y = X$ Teorema de la absorción. Término redundante

10-) $X \cdot (X + Y) = X$ Teorema de la absorción. Término redundante

11-) $\overline{X \cdot Y \cdot Z} = \overline{X} + \overline{Y} + \overline{Z}$ Teorema de Morgan.

12-) $\overline{X + Y + Z} = \overline{X} \cdot \overline{Y} \cdot \overline{Z}$ Teorema de Morgan.

13-) $X + \overline{X} \cdot Y = X + Y$

14-) $X \cdot (\overline{X} + Y) = X \cdot Y$

15-) $Z \cdot X + Z \cdot \overline{X} \cdot Y = Z \cdot X + Z \cdot Y$ si un término o factor menor aparece incluido en un término o factor mayor excepto que una variable aparece complementada, la correspondiente variable del término mayor es redundante

16-) $(Z + X)(Z + \overline{X} + Y) = (Z + X)(Z + Y)$ si un término o factor menor aparece incluido en un término o factor mayor excepto que una variable aparece complementada, la correspondiente variable del término mayor es redundante.

17-) $X \cdot Y + \overline{X} \cdot Z + Y \cdot Z = X \cdot Y + \overline{X} \cdot Z$ si existen dos términos o factores en los cuales aparecen en uno, una variable y en el otro complementada, el término o factor formado por las variables restantes esta contenida en dicha expresión, por lo cual si aparece es redundante.

18-) $(X + Y) \cdot (\overline{X} + Z) \cdot (Y + Z) = (X + Y)(\overline{X} + Z)$ si existen dos términos o factores en los cuales aparecen en uno, una variable y en el otro complementada, el término o factor formado por las variables restantes esta contenida en dicha expresión, por lo cual si aparece es redundante.

19-) $(X \cdot Y) + (\overline{X} \cdot Z) = (X + Z)(\overline{X} + Y)$ Pasar de suma a producto.

20-) $(X + Y)(\overline{X} + Z) = (X \cdot Z) + (\overline{X} \cdot Y)$ Pasar de producto a suma.

4.- Simplificación de expresiones booleanas

Una expresión booleana determinada puede expresarse con diferentes funciones que son equivalentes. Sin embargo en la construcción del circuito, el proyectista debe obtener la función que minimice el costo y maximice la fiabilidad del circuito y para ello debe optar por la función más adecuada, simplificada o minimizada lo más posible para obtener circuitos que tengan la menor cantidad de compuertas o niveles. Uno de los posibles métodos para lograr la minimización de estas funciones es aplicar los teoremas o postulados convenientemente. Veamos los siguientes ejemplos

Ejercicio: Llevar la función a su mínima expresión aplicando los teoremas o postulados del álgebra de Boole.

1. $AB + ABZ + A\overline{X} + BX + \overline{A}\,\overline{X}$

   Término redundante

$AB + A\overline{X} + BX + \overline{A}\,\overline{X}$ Como $A\overline{X} + \overline{A}\,\overline{X} = \overline{X}$ Teorema 7

$AB + \overline{X} + BX$ Como $\overline{X} + BX = \overline{X} + B$ Teorema 13

$AB + \overline{X} + B$

Término menor contenido en uno mayor, es redundante

$\overline{X} + B$ Resultado.

2. $AB + ABZ + \overline{A}X + BX + \overline{A}\,\overline{X}$

   Término redundante

$AB + \overline{A}X + BX + \overline{A}\,\overline{X}$

BX Término redundante BX por teorema 17

$AB + \overline{A}X + \overline{A}\,\overline{X}$ **pero** $\overline{A}X + \overline{A}\,\overline{X} = \overline{A}$ Por teorema 7

$AB + \overline{A}$ donde $\overline{A} + AB = \overline{A} + B$ Por teorema 13

$\longrightarrow$ $\overline{A} + B$ es el Resultado

3. $\left(\overline{Y} + \overline{Z}\right)\left(\overline{Y} + Z\right)\left(Y + X\right)\left(\overline{Y} + X\right)\left(Z + X\right)$

Tenemos que $\left(\overline{Y} + \overline{Z}\right)\left(\overline{Y} + Z\right) = \overline{Y}$ por lo que entonces

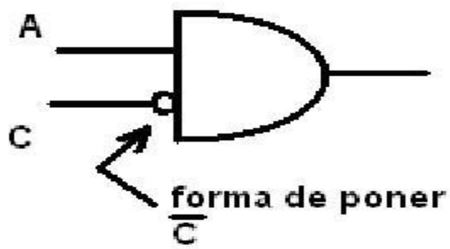
$\left(\overline{Y}\right)\left(Y + X\right)\left(\overline{Y} + X\right)\left(Z + X\right)$

$(\overline{Y})(Y+X)(\overline{Y}+X)(Z+X)$ Como $(Y+X)(\overline{Y}+X)=X$ Por Teorema 8

$(\overline{Y})(X)(Z+X)$

Término redundante

$\boxed{(\overline{Y})(X)}$ Resultado



4. $A\overline{B}\overline{C}+A\overline{B}\overline{C}D+\overline{C}A$

$A\overline{B}\overline{C}+\overline{C}A$

$\boxed{\overline{C}A}$ $\longrightarrow$ Resultado

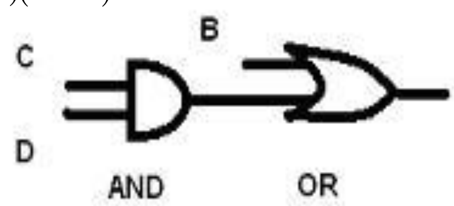
5. $(A+B+CD)(\overline{A}+B)(\overline{A}+B+E)$

Redundante

$(A+B+CD)(\overline{A}+B)$ $(A+B)(\overline{A}+B)=B$

$B+CD\overline{A}$ + CDB

$\boxed{B+CD\overline{A}}$ $\longrightarrow$ Resultado

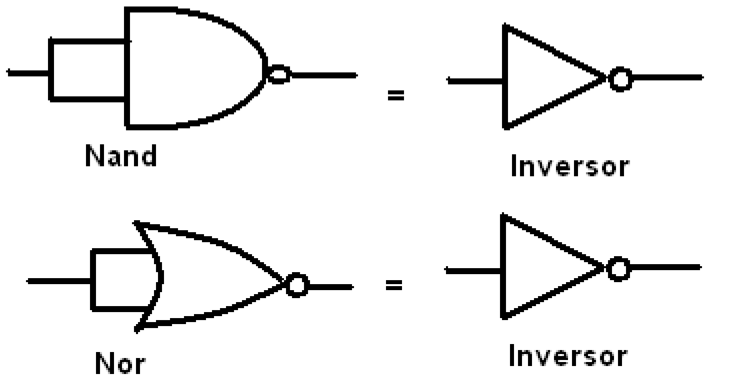


6. $(A+B)(\overline{A}+C)(\overline{B}+C)$

$\boxed{(A+B)C}$ $\longrightarrow$ Resultado

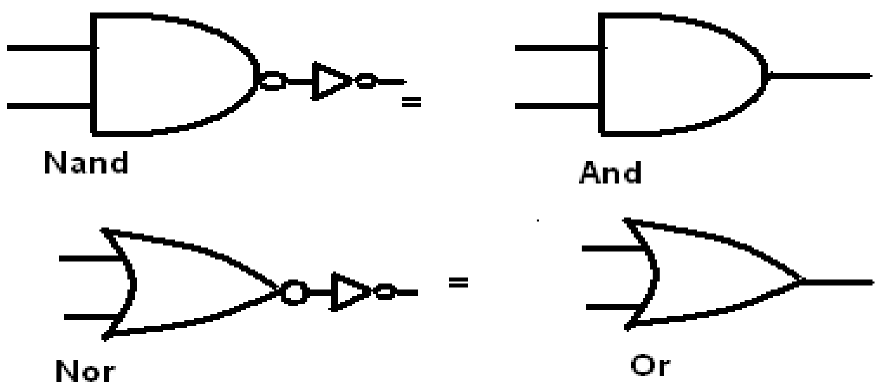
Las compuertas NAND y NOR son denominadas compuertas universales, ya que con la combinación de ellas se pueden lograr las otras compuertas.

Así por ejemplo, si a las compuertas NAND y NOR de dos entradas se les cortocircuitan las entradas se convierten en una compuerta inversora.



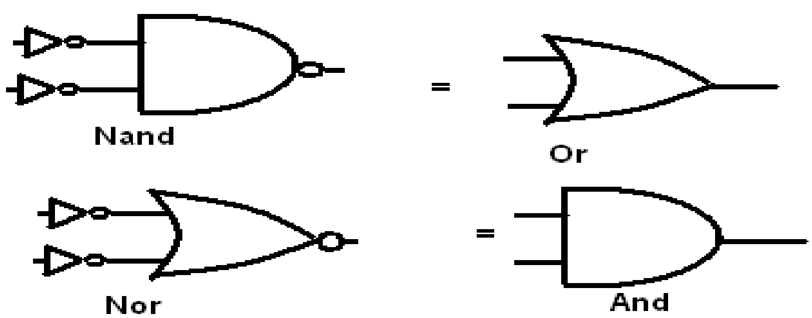
Se cortocircuitan las entradas y se convierten en un inversor

Si a las compuertas NAND y NOR se le pone un inversor a su salida, se niega la misma y se convierte en otra compuerta AND y OR



Se le invierten la salida y son And y Or

Si a las compuertas NAND y NOR se les pone un inversor en sus entradas, se convierten en compuertas OR o AND.



Se le invierten las entradas y se comvierten en Or y And

5.- Familias lógicas

Los circuitos digitales utilizan componentes encapsulados, los cuales pueden alojar compuertas lógicas o circuitos lógicos más complejos.

Estos componentes están estandarizados, para que haya una compatibilidad entre fabricantes, de forma que las características más importantes sean comunes. De forma global los componentes lógicos se abarcan dentro de una de las dos familias siguientes:

TTL: diseñada para una alta velocidad.

CMOS: diseñada para un bajo consumo.

Actualmente a partir de estas dos familias se han creado otras, que intentan obtener lo mejor de ambas: un bajo consumo y una alta velocidad.

No se hace reseña a la familia lógica ECL, la cual se halla entre la TTL y la CMOS. Esta familia nació como un intento de alcanzar la rapidez de TTL y el bajo consumo de CMOS, pero se emplea en pocas ocasiones

Comparación de las familias

PARÁMETRO	**TTL estándar**	**TTL 74L**	**TTL Schottky de baja potencia (LS)**	**Fairchild 4000B CMOS (con Vcc=5V)**	**Fairchild 4000B CMOS (con Vcc=10V)**
Tiempo de propagación de puerta	10 ns	33 ns	5 ns	40 ns	20 ns
Frecuencia máxima de funcionamiento	35 MHz	3 MHz	45 MHz	8 MHz	16 MHz
Potencia disipada por puerta	10 mW	1 mW	2 mW	10 nW	10 nW
Margen de ruido admisible	1 V	1 V	0'8 V	2 V	4 V
Fan out	10	10	20	50 (*)	50 (*)

(*) O lo que permita el tiempo de propagación admisible

Dentro de la familia TTL encontramos las siguiente sub-familias:

- L: Low power = disipación de potencia muy baja
- LS: Low power Schottky = disipación y tiempo de propagación pequeños

- S: Schottky = disipación normal y tiempo de propagación pequeño.
- AS: Advanced Schottky = disipación normal y tiempo de propagación extremadamente pequeño

6. CIRCUÍTOS ARITMÉTICOS

Las compuertas hasta ahora estudiadas son las encargadas de realizar las operaciones lógicas, pero también en esta técnica digital se necesita de la realización de operaciones aritméticas.

La operación suma de dos números en binario es muy simple. Para ello solo ha de cumplirse la siguiente tabla:

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

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

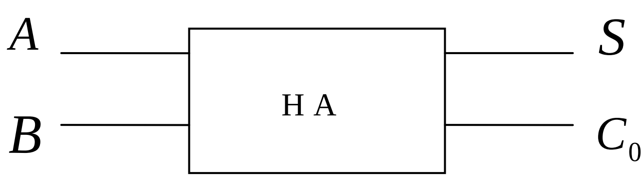
En ella A y B representan los sumandos, S es el resultado de la suma y Co lo que llevo o arrastro para sumarse al dígito que se encuentra a la izquierda. El procedimiento es similar al que se realiza con la suma de dos números en el sistema decimal.

6.1 Semisumador. El circuito capaz de sumar dos números en el sistema binario, recibe el nombre de semisumador (half adder). El mismo tiene dos entradas A y B, y dos salidas S y Co (suma y acarreo)

De la tabla de verdad anteriormente mostrada pueden obtenerse fácilmente las expresiones booleanas para S y Co., la representación de este circuito pudiera tomar la siguiente forma.

.

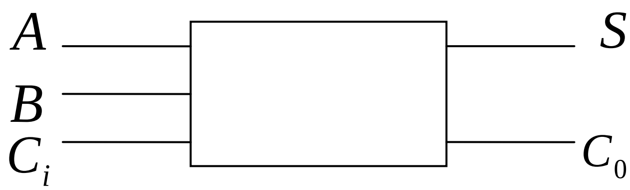
$$S = \overline{A}B + A\overline{B}$$
$$C_O = AB$$



Este circuito tiene por objetivo efectuar la suma de los bits. Es por tanto el sumador básico elemental

6.2 Sumador Completo Binario.

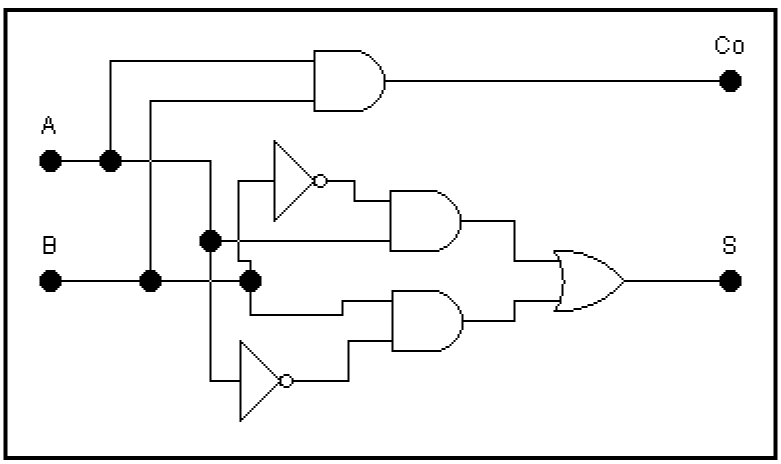
Surge ante la necesidad de disponer de un circuito que efectúe la suma con tres entradas: A, B y el acarreo, y que presente dos salidas: acarreo y suma. Este sumador completo puede construirse con dos semisumadores.



Las ecuaciones son en este caso: $S = A \oplus B \oplus C_i$

$$C_0 = AC_i + BC_i + AB$$

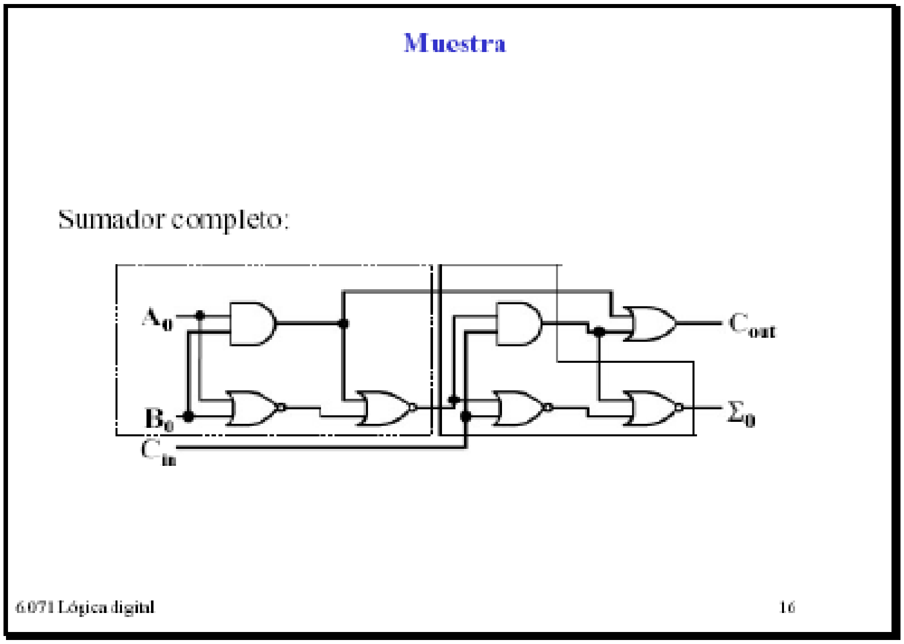
Por lo que el circuito lógico puede quedar como sigue:



Su tabla de la verdad será la siguiente:

A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Otra variante de cómo construir un sumador completo en base a compuertas lógicas sería la siguiente:



Para diseñar un circuito lógico capaz de sumar dos cantidades binarias hay que tener en cuenta de que es necesario disponer de un sistema capaz de incorporar el acarreo de la posición anterior a la posterior, por lo que hay que saber si la información (cantidad a sumar) se transmite en forma serie o paralelo.

<u>Forma serie</u>. Es cuando la información es transmitida por un solo hilo (una sola vía) un bit a continuación de otro.

<u>Forma paralelo</u>. Es cuando la información es transmitida por varios hilos simultáneamente, transmitiéndose un bit por cada hilo.

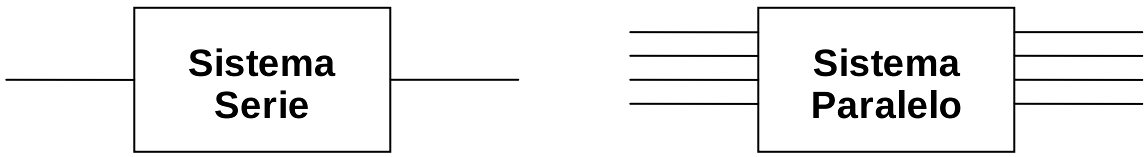
Es evidente que la comunicación paralelo es mucho mas rápida, pero son necesarios tantos hilos entre el transmisor y el receptor como bits tenga el número que voy a transmitir (como bits tenga la palabra). Cuando las distancias son muy grandes, por ejemplo, enlaces de computadoras a través de un sistema de comunicación, lograr más de un canal de comunicación encarece notablemente el sistema y por tanto en este caso se prefiere la transmisión en serie de la información. Esto nos quiere decir que en un sistema digital podremos encontrarnos la información que deseamos sumar en serie o paralelo indistintamente. De aquí que se puedan presentar las siguientes variantes:

<u>Sumador Binario Serie</u>:

La información viene en la misma línea, o sea los diferentes bits del número binario se suceden unos a otros en el tiempo.

Sumador en Forma Paralelo.

La información viene por varias líneas en el mismo instante, o sea cada bit se trasmite por una línea diferente. Es la variante más rápida, pero se necesitan circuitos más complejos. Puede presentarse con dos tipos de acarreo:



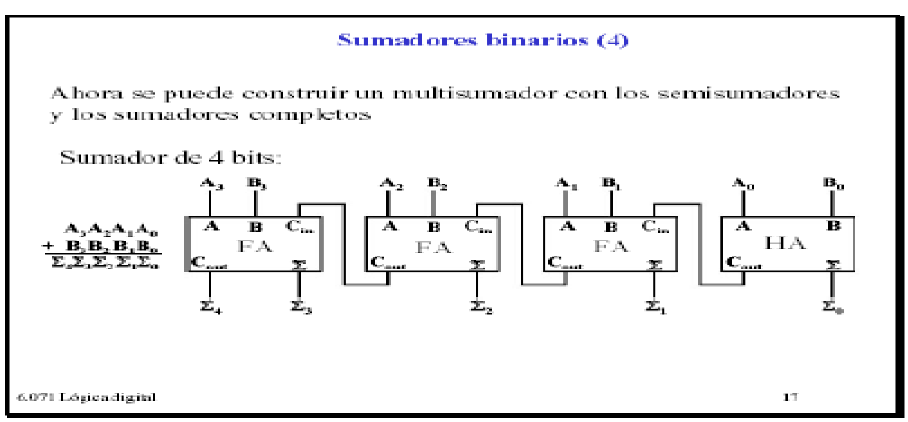
Sumador en Paralelo con Acarreo Serie.

Todos los bits de ambos números a sumar se representan simultáneamente en paralelo en las entradas del sumador, y por tanto se necesitan tantos sumadores como bits tengan los números a sumar. El acarreo producido por cada pareja de bit del mismo peso, es introducido en el sumador del bit de orden inmediato superior. Esto repercute en la velocidad de la suma.

Sumador Rápido: Sumador en Paralelo con acarreo en Paralelo.

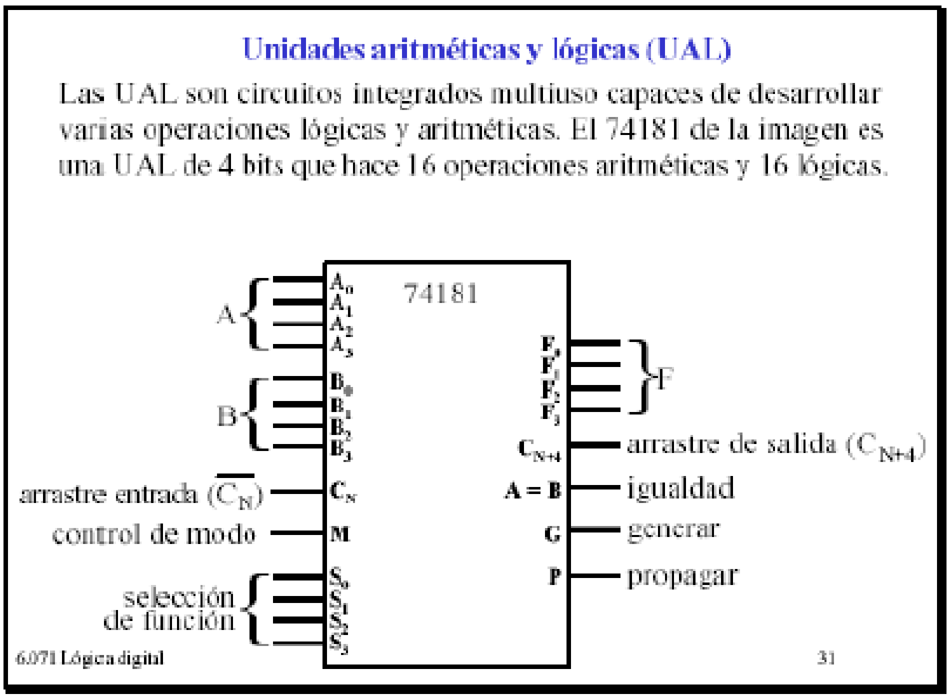
La pérdida de velocidad en el sumador rápido con acarreo serie necesita ser disminuida en algunos sistemas y esto ha dado lugar a buscar vías, que permitan reducir el tiempo que se tarda en conocer la suma y el acarreo final. Para ello se han ido reduciendo el número de puentes a través de los cuales debe de propagarse el acarreo.

Lo visto hasta el momento nos da la posibilidad de decir que en un sistema digital podemos encontrarnos la información que deseamos sumar en serie o paralelo indistintamente.
Veamos como ejemplo un circuito donde se refleja la forma de construir un sumador binario de 4 bits en forma paralelo, utilizando tres sumadores completos y un semisumador:



Existen circuitos integrados que pueden desarrollar, mediante un selector de funciones programable, cualquier operación lógica y muchas aritméticas con palabras de 24 bits. Como ejemplo tenemos el 74181. En realidad se trata de un dispositivo que es ya un poco anticuado para los estándares actuales, pero que tiene algunas buenas capacidades de chips más complejos. Hoy se utilizaría más bien un microprocesador que una UAL.

Para las operaciones aritméticas, el chip dispone de una entrada (desde una serie de UAL de menor significación) y un arrastre de salida (a una serie de UAL de mayor significación). El chip funciona en complemento de 2 y puede sumar, restar, doblar, etc. Una descripción de las entradas y salidas que posee se muestra en el esquema que aparece a continuación:



7.- Multivibradores.

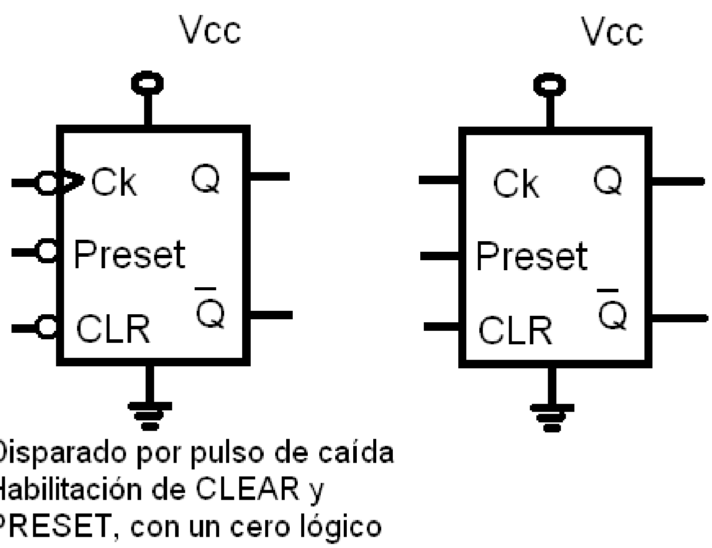
Los multivibradores son circuitos de mucha utilización dentro de la electrónica digital. De acuerdo con sus características, se dividen en biestables, monoestables y astables.

Multivibrador Astable es aquel circuito que posee dos estados inestables y se encuentra oscilando entre ellos, de forma continua sin necesidad de un estimulo externo, con una frecuencia que depende de los parámetros del circuito.

Los **Multivibradores Monoestables**, son aquellos circuitos que poseen un solo estado estable y pueden pasar a su estado inestable a través de un impulso externo, permaneciendo en este estado un determinado tiempo, regresando por sí solo a su estado estable. Los mismos pueden ser redisparables o no redisparables, utilizándose como generador de pulsos de compuerta (pulsos anchos) de duración controlable o como circuito de demora,

Los **Multivibradores biestables** también llamados flip flop, son circuitos que poseen dos estados estables bien definidos, el nivel alto (uno lógico) y el nivel bajo (cero lógico) y pueden permanecer en uno u en otro indistintamente y solo se pueden cambiar a voluntad cuando las condiciones así lo requieran. Pueden estar elaborados en base a transistores bipolares, como el caso de la lógica de transistor a transistor TTL, en base a los MOS-FETs o transistores de efecto de campo de óxido metálico de silicio o empleando compuertas lógicas.

Su representación más sencilla, consiste en un rectángulo con sus terminales de fuente, el V_{CC} o polo positivo y masa o polo negativo (GND). Tiene además dos terminales Q y /Q (Q negada). También dispone de terminales de entrada, uno para los pulsos de reloj (Ck o Cp) y otros opcionales para limpieza o Clear (CLR) y Preset.



Los terminales de salida Q y Q^i (Q invertido) presentan dos niveles opuestos, el uno y el cero, es decir sí Q esta con cero, Q^i con uno y viceversa.

El Terminal de entrada de reloj (Ck), conmuta o cambia el estado de las salidas Q y Q^i. El pulso del reloj, que propicia el cambio, puede realizarse por cambio de nivel, o sea, por un "1" lógico o "0" lógico o activarse por frente de subida o bajada del pulso del reloj.

El Terminal Clear, cuando está activado, fuerza a la salida Q a ponerse en un nivel de cero lógico. El Terminal Preset cuando está activado fuerza a la salida Q a ponerse en un nivel de uno lógico. Estos terminales de entradas se habilitan con nivel bajo o alto. Para habilitarse con nivel bajo el esquema eléctrico debe tener en estas entradas un pequeño círculo asociado y se activa con nivel alto cuando el esquema eléctrico no posee el círculo asociado.

Los biestables pueden clasificarse según su entrada de reloj.

Sin entrada de reloj, llamado también **asincrónico.** El biestable sin entrada de reloj es aquel que no necesita de una señal de reloj para hacer válidos los datos presentes en sus entradas, razón por la cual sus salidas Q y Q^{i} (Q invertido) van a reflejar en todo momento, los cambios en la medida que estos van ocurriendo en sus entradas de datos.

Con entrada de reloj o **sincrónico.** El pulso de un reloj en los procesos sincrónicos es semejante al compás marcado por el director de la orquesta; si no existiera director de la orquesta, los músicos ejecutantes podrían perder la noción de los momentos en que debe hacer sonar su instrumento. En el biestable con señal de reloj, la salida Q y Q^{i} (Q invertido) no depende solamente de las entradas, sino también de la presencia del pulso del reloj, para que pueda existir un cambio en las salidas.

La importancia de los biestables se debe a que los sistemas digitales necesitan de elementos de memorias, donde la información pueda ser almacenada temporal o definitivamente, o sea, un circuito cuya salida dependa de la señal de entrada en un instante específico y que además recuerde el estado de la entrada en ese instante, a pesar de que haya cambiado ya. El circuito puede permanecer indefinidamente en cualquiera de sus estados estables.

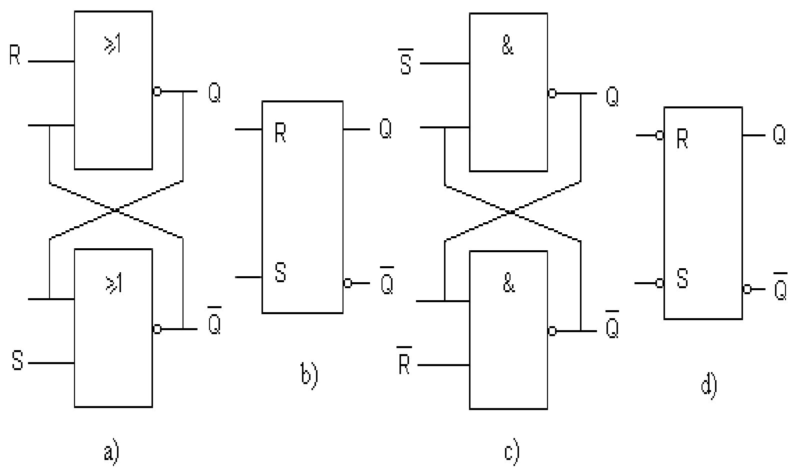
7.1 Tipos de biestables.

Biestables RS

Su tabla de verdad es la mostrada (**Q** representa el estado actual de la salida y **q** el estado anterior a la última activación):

Tabla de verdad biestable RS			
R	**S**	**Q (NO-O)**	$\overline{Q}$ **(NO-Y)**
0	0	q	N. D.
0	1	1	0
1	0	0	1
1	1	N. D.	q
N. D.= Estado no determinado			

El flip-flop más sencillo es el elaborado en base a dos compuertas NAND. Para poder analizar su funcionamiento se debe tener en cuenta en que estados se encontraba la salida.

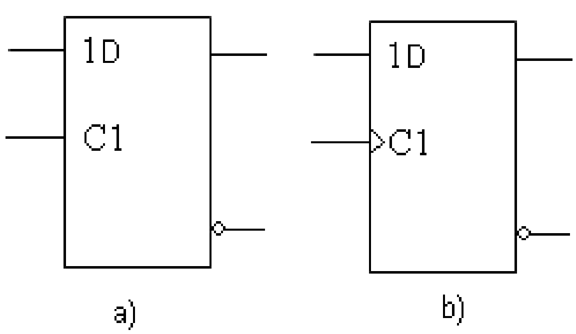


Circuito Biestable RS sincrónico. Además de las entradas R y S, posee una entrada C de sincronismo también llamada de reloj (clock) cuya misión es la de permitir o no el cambio de estado del biestable.

Cuando en este biestable el Terminal S (Set) y R (Reset) se colocan en niveles altos, no cambian las salidas Q y /Q, actuando como una memoria básica.

Existe una combinación de las entradas en que la salida es indeterminada, ya que depende de las demoras de las compuertas y no se puede predecir el resultado, por lo que esta combinación se trata de evitar y ocurre cuando ambas entradas están en nivel bajo.

Biestables Tipo D



D	**Q**	**$Q_{siguiente}$**
0	X	**0**
1	X	**1**
X=no importa		

Este biestable es un dispositivo de almacenamiento temporal de dos estados (alto y bajo), cuya salida adquiere el valor de la entrada D cuando se activa la entrada de sincronismo, C. Es ampliamente usado como elemento de demora. En función del modo de activación de dicha entrada de sincronismo, existen dos tipos de biestables D:

- Activo por nivel (alto o bajo), también denominado registro o cerrojo (*latch* en inglés).

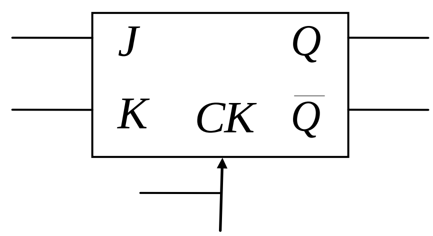
Activo por flanco (de subida o de bajada). La ecuación característica del biestable D que describe su comportamiento es:

$$Q_{siguiente} = D$$

Biestable J-K. La ecuación característica del biestable JK que describe su comportamiento es: $Q_{siguiente} = J\overline{Q} + \overline{K}Q$

Y su tabla de verdad es:

J	K	Q	$Q_{siguiente}$
0	0	0	**0**
0	0	1	**1**
0	1	X	**0**
1	0	X	**1**
1	1	0	**1**
1	1	1	**0**
X=no importa			



El flip-flop **J-K** es uno de los más elaborados multivibradores biestables, donde las entradas J y K son entradas programables.

Actualmente vienen varios chips o CI que contienen dos flip flop de este tipo.

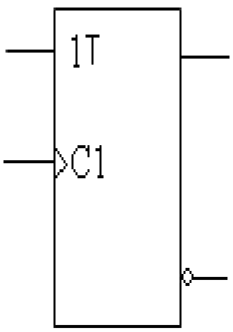
Mientras que los terminales J y K tengan niveles altos, el flip flop cambia o conmuta las salidas Q y Q^{i} con cada pulso del reloj. De lo contrario si las entradas presentan niveles bajos la salida continua manteniendo el estado que poseía antes de cada pulso del reloj.

Cuando las entradas J y K poseen niveles diferentes, la salida Q toma el nivel que tenga la J, por lo que el flip flop memoriza o lo que es lo mismo, transfiere a su salida el dato que tiene la entrada J.

También existe el biestable J – K denominado amo-esclavo (master-slave), el que utiliza dos flip flop conectados en serie, y teniendo como característica que las entradas nunca se conectan a la salida directamente.

Para explicar su funcionamiento se precisa comprender que con el frente del pulso la información de la entrada es almacenada en el flip flop amo, y con el flanco posterior del pulso, la información pasa a la salida del esclavo. Para ello las entradas J y K se deben mantener estables hasta que el pulso del reloj caiga.

Biestable T



T	Q	Qsiguiente
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

Este biestable es un dispositivo de almacenamiento temporal de dos estados (alto y bajo). El biestable T cambia de estado ("toggle" en inglés) cada vez que se activa la entrada de sincronismo o de reloj si la entrada T está en un nivel alto. Si la entrada T está a nivel bajo, la salida retiene el nivel previo.

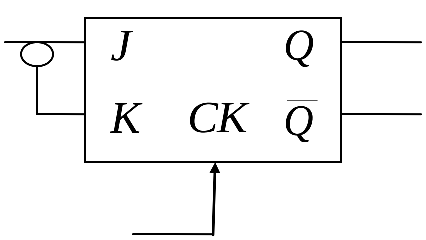
La ecuación característica del biestable T que describe su comportamiento es:

$$Q_{siguiente} = T \oplus Q$$

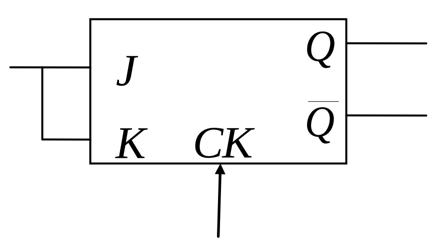
Este biestable sirve para ser un divisor de dos en la frecuencia de entrada. Para ello se cambian los pulsos del biestable con solo los pulsos del reloj.

Para concluir el presente estudio de los multivibradores biestables, puede verse que el biestable J-K se puede convertir en D y T uniendo los terminales J Y K (Biestable tipo **T),** o poniendo un inversor entre los terminales J y K (Biestable tipo **D),** tal como aparece en los esquemas que se muestran a continuación:

BIESTABLE D



BIESTABLE T



Lo que se ha expuesto hasta aquí, constituye la base para que se pueda abordar con éxito un estudio más profundo de la Electrónica Digital. En la segunda parte de este trabajo se abordará la descripción de dispositivos de una mayor complejidad, pero igualmente necesarios para la comprensión del funcionamiento de los sistemas digitales contemporáneos.

Bibliografía.

Writte, Robert A.. 1993. Electronic test Instrument: Theory and applications. Prentice Hall International. USA. ISBN 0-13-253147-X., 274 P.

Muñoz Merino, Elías y otros. 1990. Circuitos electrónicos digitales II. Editora Revolucionaria. Cuba. 527 P.

Luis, Ernesto. *¿Qué es la Electrónica?* [on-line], descargado: julio del 2009. disponible en http: //www.monografias.com/trabajo 5

Rodríguez, Carlos. *Curso Básico de Electrónica*. [on-line], consultado: julio del 2009. disponible en http: //www.monografias.com/trabajo 38 electrónica-básic**a**