ELECTRONICA DIGITAL
[REPASO]

Kevin Jaramillo

UNIVERSIDAD POLITÉCNICA SALESIANA
CUENCA - ECUADOR

kjaramillog@est.ups.edu.ec

Abstract.- The present gathers all the information about The Digital Electronics, of the importance that this he has, also the applications that he could have and small circuits that could be very useful in the daily life, it has been described in a very detailed way the basic concepts of this matter or area, presently we focus ourselves with much more rigor in the area he practices, making brief explanations of each one of the exposed circuits, attempting is easier to understand this way the summary of this.

I. INTRODUCCIÓN.

La electrónica digital es una rama de la electrónica sumamente importante ya que es la base para entender todo lo concerniente a el almacenamiento de datos, ya que empezaremos con el almacenamiento de un bit con un flip – flop hasta muchos de estos con una memoria RAM, también con el conocimiento de esta materia y un poco de imaginación podemos hacer circuitos muy sencillos pero con una gran aplicación, como el mando de motores, ayudar en las instalaciones civiles, hacer un sencillo mando de un semáforo, hacer la automatización de instalaciones industriales, haciendo así el aporte con tecnología mucho mas nueva y económica, a continuación haremos la descripción de forma resumida de los conocimientos básicos que debemos tener para empezar a entender y explorar el mundo digital.

II. DESARROLLO.

1. SISTEMAS, CÓDIGOS Y OPERACIONES LÓGICAS.

Dentro de los sistemas numéricos digitales, podemos decir que son una manera diversa de numeración a manera de ceros y unos, dentro de lo que trataremos, también se podrá usar la terminología de niveles bajos o altos, entendiendo a los niveles bajos como ceros (0) y a los altos como unos (1), los mismos que también se les llamara bits.

La definición más básica o elemental de sistemas numéricos digitales es que es una forma más de representar a un número, en seguida describiremos los sistemas más comunes, para criterio del autor todos estos sistemas tienen la misma importancia así que el orden en el que se describan no representa el nivel de importancia.

a) SISTEMA DECIMAL.

Es el sistema normal que utilizamos para contar y el que conocemos desde la escuela, no tiene ninguna variante.

b) SISTEMA BINARIO.

El sistema binario es una manera de representar los números con ceros y uso a manera de códigos, ejemplo 9 1001.

c) SISTEMA HEXADECIMAL.

En este sistema no se diferencia en nada del sistema decimal hasta 9, después de eso se sigue contando con las letras del abecedario hasta la F que representa el 15.

d) CÓDIGO BCD (Binary Direct Code).

Este código se representa muy similar al sistema binario pero con 8bits, ejemplo: para representar el 33 se emplea el 0011 0011, que en binario se separa los números y se busca la equivalencia en binario.

Estos códigos son los más utilizados y los que se utilizara más adelante para realizar las prácticas o circuitos, a continuación en la tabla 1 exponemos las equivalencias más básicas de los diferentes sistemas.

DECIMAL	BINARIO	HEXADECIMAL	BCD
0	**0000**	**0**	**0000**
1	**0001**	**1**	**0001**
2	**0010**	**2**	**0010**
3	**0011**	**3**	**0011**
4	**0100**	**4**	**0100**
5	**0101**	**5**	**0101**
6	**0110**	**6**	**0110**
7	**0111**	**7**	**0111**
8	**1000**	**8**	**1000**
9	**1001**	**9**	**1001**
10	**1010**	**A**	**0001 0000**
11	**1011**	**B**	**0001 0001**
12	**1100**	**C**	**0001 0010**
13	**1101**	**D**	**0001 0011**
14	**1110**	**E**	**0001 0100**
15	**1111**	**F**	**0001 0101**

TABLA 1. EQUIVALENCIAS DE LOS CUATROS SISTEMAS QUE UTILIZAREMOS.

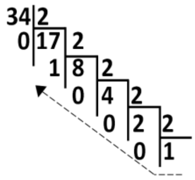
2. CONVERSIONES.

En muchos casos es necesario cuando conocemos un numero pasarlo a su equivalente en otro tipo de sistema, por ello explicaremos las conversiones que podemos realizar.

a. CONVERSIÓN DE DECIMAL A BINARIO.

Para realizar estas conversiones tenemos que realizar múltiples divisiones para 2, ejemplo:

Convertir 34 a binario.



Como se puede apreciaren se realizó la múltiple división para 2, después de esto escribimos los números en el sentido que muestra la flecha.

100010 ➔ 34

b. CONVERSIÓN DE BINARIO A DECIMAL.

Para realizar esta conversión se separa los números y multiplicamos cada número por 2 y después empezamos a elevar a cada número elevando a la potencia de forma creciente, luego se hace la suma de cada número, ejemplo:

1 0 0 0 1 0

X

2

2 0 0 0 2 0

↓

POTENCIA CRECIENTE

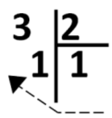
$2^5 \quad 0^4 \quad 0^3 \quad 0^2 \quad 2^1 \quad 0^0$

$32 + 0 + 0 + 0 + 2 + 0 = 34$

c. CONVERSIÓN DE DECIMAL A BCD.

Para esta separamos cada número y lo convertimos a su equivalente binario, pero recordando que es de 4 bits, ejemplo:

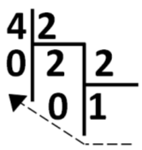
Convertir 34 a BCD.



En el resultado nos sale 11 pero como tiene que ser de 4 bits, agregamos dos ceros, quedándonos:

3 → 0011,

Lo mismo hacemos con el 4,



0100

34 → 0011 0100

d. CONVERSIÓN DE HEXADECIMAL A DECIMAL.

Es muy similar a la conversión de binario a decimal, a diferencia que no se multiplica por 2 sino por 16, ejemplo.

Convertir A34C a decimal.

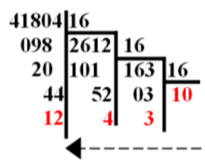
Para esto seguiremos el mismo proceso de la conversión binario a decimal, separaremos los números, basándonos en la tabla 1, reemplazaremos las letras por su equivalente en número decimal para poder multiplicar y eleva a su potencia.

1. A34C → 10 3 4 12
2. 10X16 3X16 4X16 12X16
3. $10X16^3$ $3X16^2$ $4X16^1$ $12X16^0$
4. 40960 + 768 + 64 + 12
5. 41804

e. CONVERSIÓN DE DECIMAL A HEXADECIMAL.

Realizaremos la conversión siguiendo los mismos pasos que realizamos para convertir de decimal binario pero con la variante de que ahora dividiremos para 16, ejemplo:

Convertir 41804 a hexadecimal.



El resultado es 10 3 4 12

Ahora simplemente buscamos el equivalente del 10 y 12 en la tabla 1.

A34C

f. CONVERSIÓN DE HEXADECIMAL A BINARIO.

Para realizar la conversión de hexadecimal a binario tenemos que separar cada uno de los números y buscar su equivalente en la tabla 1, ejemplo:

Convertir A34C a binario.

A34C ➔ A 3 4 C

A=1010
3=0011
4=0100
C=1100,
Entonces:

A34C ➔ 1010 0011 0100 1100

Como podemos observar es muy fácil la transformación, solo tenemos que fijarnos muy bien en la tabla.

g. CONVERSIÓN DE BINARIO A HEXADECIMAL.

Cuando nos den un número en binario debemos separar en grupos de 4 desde la derecha hacia la izquierda y luego buscar su equivalente en la tabla 1, si los últimos números no son de 4 debemos completar el grupo de 4 con ceros para no afectar en nada al número.

3. OPERACIONES CON LOS NÚMEROS DIGITALES.

Una vez conociendo las conversiones entre un sistema y otro, procederemos a aprender las operaciones que podemos realizar con los mismos, tales como suma, resta, multiplicación y división.

En esta sección no explicaremos las operaciones con números decimales, ya que son las mismas que ya estudiamos en niveles bajos o en educación básica.

a) EN SISTEMA BINARIO

- **SUMA.**

Antes de proceder con una ligera explicación de cómo se realiza la suma debemos tener en cuenta que **1+1=10** y que **1+1+1=11,** la suma se realiza igual que con los números decimales, se pone un numero debajo de otro y se realiza la suma, a continuación expondremos un ejemplo para una mejor comprensión del tema.

Sumar 1111 y 110.

Para empezar con la suma podemos hacer que ambos términos sean de 4 bits agregando un cero a la izquierda del numero que solo consta de tres bits.

```
  1111 (15)
+ 0110 (6)
 10101 (21)
```

- **MULTIPLICACIÓN.**

Se realiza igual que con la multiplicación convencional de números en sistema decimal, ejemplo:

Multiplicar 1111 por 110.

```
      1111 (15)
    x 0110 (6)
      0000
     1111
    1111
   0000
   1011010 (90)
```

- **RESTA.**

Antes de hablar de resta entenderemos como se realiza el complemento A1 y A2.

- ✓ **COMPLEMENTO A1.**

El complemento A1 consiste en tener un número y negarlo por completo, ejemplo.

Escriba el complemento A1 de 1010.

Al negar por completo el número 1010 nos queda: **0101**.

- ✓ **COMPLEMENTO A2.**

Para poder realizar el complemento A2 debemos conocer primero el complemento A1 de un número y luego sumarle 1, ejemplo:

Escriba el complemento A2 de 1010.
De la sección anterior ya conocemos el complemento A1 de dicho numero procederemos a sumarle 1 para tener nuestro complemento A2.

$$\begin{array}{r} 0101 \\ +\quad\;\; 1 \\ \hline \mathbf{0110} \end{array}$$

Entonces nuestro complemento A2 del número 1010 es 0110.

Una vez que conocemos de los complementos A1 y A2 podemos hablar de resta binaria, entonces cuando nos digan realizar una resta en binario debemos tener presente que al sustraendo se le encuentra el complemento A2 y esto se suma al minuendo, ejemplo.

Restar 3 de 10 en binario.

Para esto conocemos que:

3 ➔ 0011
10 ➔ 1010

Encontrando el complemento A2 de 10:

0011 ➔ complemento A1 ➔ 1100
Complemento A2:

$$\begin{array}{r} 1100 \\ +\quad\; 1 \\ \hline \mathbf{1101} \end{array}$$

Ahora procedemos a sumar este resultado a 1010:

$$\begin{array}{r} 1101 \\ +\,1010 \\ \hline \mathbf{10111} \end{array}$$

Ahora bien ya que conocemos el resultado procederemos a despreciar el bits sobrante y a escribir el resultado.

$$\begin{array}{r} 1101 \\ +\,1010 \\ \hline \mathbf{10111} \end{array}$$

Nuestro resultado de la resta de 10-3=7 y eso fue lo que hayamos **0111.**

b) EN SISTEMA BCD.

- **SUMA.**

La suma se realiza de manera similar a la suma binara con la única diferencia que si el resultado de la suma excede a el numero 9 debemos corregir sumando el 6 (0110 a nuestro resultado), a continuación se presentan dos ejemplos para que quede clara la explicación y el lector sepa cuando tiene que sumar un seis y cuando no.

Sumar 4 y 3 en BCD.

$$\begin{array}{r} 0100 \\ +\,0011 \\ \hline \mathbf{0111} \end{array}$$

Como la suma NO excede al 9 queda ahí el resultado.

Sumar 6 y 8.

$$\begin{array}{r} 0110 \\ +\,1000 \\ \hline \mathbf{1110} \end{array}$$

Como esta vez la respuesta si excede al 9 tenemos que sumar 6 al resultado.

$$\begin{array}{r} 1110 \\ +\,0110 \\ \hline \mathbf{10100} \end{array}$$

Entonces se separa en grupos de 4 el resultado empezando de derecha a izquierda quedándonos 1 0100, finalmente para expresar el resultado se le agrega al 1 tres ceros para poder expresar el resultado en BCD.

0001	**0100**
1	**4**

La información expuesta nos servirá para las secciones posteriores, y nos ayudará para comprender los temas que abordaremos en las secciones siguientes.

4. VARIABLES, CONSTANTES Y SUS NIVELES LÓGICOS

Una variable o constante lógica puede diferir su valor entre ceros o unos, en diferentes tiempos, en los circuitos lógicos podemos hablar de un cero cuando los valores a medirse varían en un rango de 0 a 0.7V y de un uno cuando los valores estén entre 2.1 a 5 V, esto se da cuando trabajemos con circuitos TTL, y serian los valores ideales para trabajar ya que en el rango de 0.8 a 2 V se considera como zona prohibida y las variables o constantes no pueden tomar los valores dentro de esta zona, porque provocaría que el circuito sea erróneo.

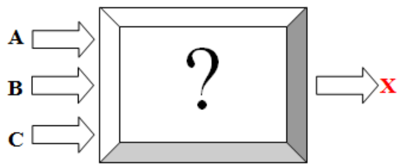
5. TABLA DE VERDAD.

Una tabla de verdad nos ayuda para saber cómo va a reaccionar un circuito por medio de su salida ante diferentes tipos de entrada. A continuación se presenta una tabla de verdad de tres entradas y una salida.

ENTRADAS			SALIDA
C	B	A	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Con la tabla de verdad anterior podemos conocer el comportamiento del este circuito, ya que conocemos por ejemplo que cuando en la entrada **C** halla un 1 y en la **B** y **A** sean ceros tendremos a la salida un uno.

Representación del circuito:



6. COMPUERTAS LÓGICAS.

Tener conocimientos sobre compuertas lógicas, así como su funcionamiento, la tabla de verdad de cada una de ellas, y el cómo y por que utilizarlas nos ayudará a diseñar cualquier circuito que deseemos, además que también entenderemos con mayor facilidad el funcionamiento de elemento que se basan en estas para poder funcionar.

a. COMPUERTA OR.

Es una compuerta denominada sumadora, su tabla de verdad se expondrá a continuación, así como su simbología y su equivalente eléctrico.

SIMBOLOGÍA.

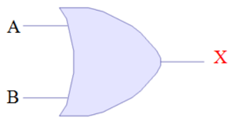
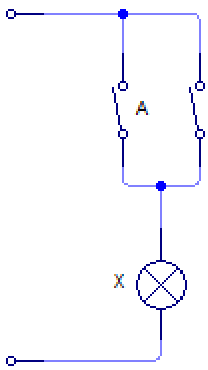


TABLA DE VERDAD

A	B	X =A+B
0	0	0
0	1	1
1	0	1
1	1	1

EQUIVALENTE ELECTRICO.



b. COMPUERTA AND

Se conoce como compuerta multiplicadora, expondremos su simbología, tabla de verdad y equivalente eléctrico.

SIMBOLOGÍA.

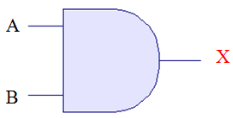
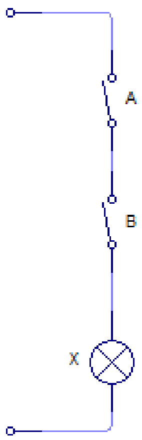


TABLA DE VERDAD

A	B	X =A*B
0	0	0
0	1	0
1	0	0
1	1	1

EQUIVALENTE ELECTRICO.



c. COMPUERTA NOT.

Si tradujéramos su nombre del inglés al español tendremos que se llama compuerta NO, entonces nos basamos en esto para decir que se trata de una negación y que invierte dodo lo que tiene a la entrada.

SIMBOLOGÍA.

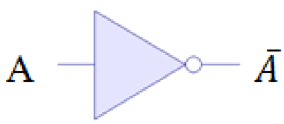
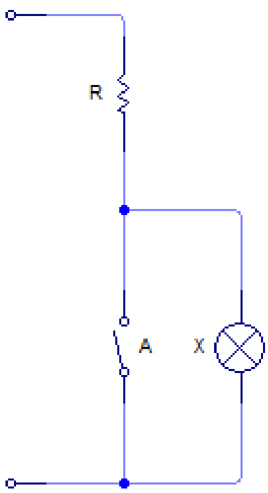


TABLA DE VERDAD

A	$\bar{A}$
0	1
1	0

EQUIVALENTE ELECTRICO.



d. COMPUERTA YES.

Si tradujéramos su nombre del inglés al español tendremos que se llama compuerta SI, entonces nos basamos en esto para decir que se trata de una compuerta que simplemente pasa lo que tiene a la entrada a la salida sin variantes.

SIMBOLOGÍA.

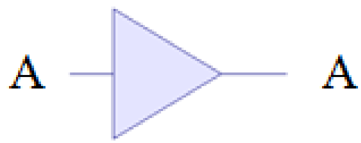
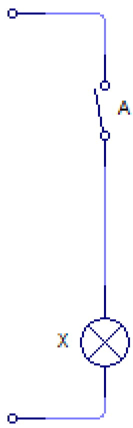


TABLA DE VERDAD

A	X=A
0	0
1	1

EQUIVALENTE ELECTRICO.



e. COMPUERTA NOR

Básicamente para criterio del autor y mayor facilidad para el lector es la negación de la compuerta OR, se puede confirmar esta afirmación al visualizar la siguiente información.

SIMBOLOGÍA.

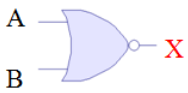
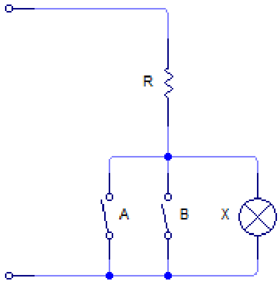


TABLA DE VERDAD

A	B	$X = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

EQUIVALENTE ELECTRICO.



f. COMPUERTA NAND

Básicamente para criterio del autor y mayor facilidad para el lector es la negación de la compuerta AND, se puede confirmar esta afirmación al visualizar la siguiente información.

SIMBOLOGÍA.

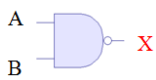
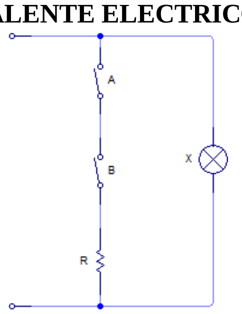


TABLA DE VERDAD

A	B	$X = \overline{A*B}$
0	0	1
0	1	1
1	0	1
1	1	0

EQUIVALENTE ELECTRICO.



g. COMPUERTA XOR

Esta es una de las compuertas que se las denomina exclusividad, la comprensión de la misma se explica con mucha claridad en la información siguiente.

SIMBOLOGÍA.

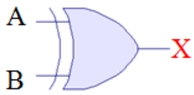


TABLA DE VERDAD

A	B	$X = A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

EQUIVALENTE ELECTRICO.

h. COMPUERTA XNOR

Es lo contrario que la compuerta XOR, su funcionamiento también se describe a continuación.

SIMBOLOGÍA.

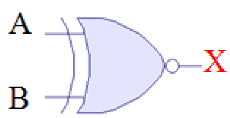


TABLA DE VERDAD

A	B	$X = \overline{A\oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

EQUIVALENTE ELECTRICO.

Conociendo el funcionamiento de cada una de las compuertas procedemos a listar los códigos de las mismas, para su adquisición en el mercado.

TIPO	NUMERACIÓN	FAMILIA
0R	74LS32	TTL
AND	74LS08	TTL
NOT	74LS04	TTL
YES	74LS07	TTL
NOR	74LS02	TTL
NAND	74LS00	TTL
XOR	74LS86	TTL
XNOR	74LS266	TTL

0R	4061	CMOS
AND	4081	CMOS
NOT	4069	CMOS
YES	4050	CMOS
NOR	4001	CMOS
NAND	4011	CMOS
XOR	4070	CMOS
XNOR	4077	CMOS

En el anexo 1 se puede visualizar los circuitos sobre todas las compuertas para afirmar y comprobar su funcionamiento.

7. ALGEBRA DE BOOLE

El utilizar el algebra de boole nos sirve o ayuda a reducir las expresiones que obtengamos de las tablas de verdad, las mismas que pueden ser des dos entradas y una salida hasta el número de entradas y salidas que nosotros deseemos, cabe decir que para números más elevados de de variables de entrada se emplea otras estrategias más avanzadas como Los Mapas de Karnaugh, que veremos en secciones siguientes.

A continuación expondremos algunas de los teoremas booleanos que podremos utilizar para realizar una simplificación más optimizada.

En seguida se expone un ejemplo donde detallaremos paso a paso como se realiza uno de estos circuitos desde la elaboración de la tabla de verdad, pasando por el proceso de reducción de la expresión hasta llegar al circuito final.

Ejemplo:

TABLA DE VERDAD

Para ejemplo y demostración de los pasos a seguir para realizar este tipo de circuitos haremos un circuito de un ingreso de tres bits y de dos salidas, en donde llamaremos **C** a nuestro bit más significativo **(MSB)**, **B** a nuestro bit medio y **A** al bit menos significativo **(B)** **(LSB),** las salidas serán “X” y “Y”.

Como son de tres bits y para poder saber que dimensiones tendrá nuestra tabla realizamos lo siguiente:

2^n, siendo n el número de bits en la o las entradas.

$2^3 = 8$, nuestra tabla tendrá una dimensión de 8.

ENTRADA				SALIDAS	
C	B	A		**X**	**Y**
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Una vez que tenemos nuestra tabla procedemos a ubicar los unos en las posiciones que deseemos o que se nos pida, para así conocer cómo deben reaccionar las salidas.

ENTRADA				SALIDAS	
C	B	A		**X**	**Y**
0	0	0			
0	0	1		**1**	
0	1	0			**1**
0	1	1		**1**	**1**
1	0	0			**1**
1	0	1		**1**	
1	1	0			
1	1	1			

Ahora que tenemos las salidas procedemos a obtener las expresiones booleanas.

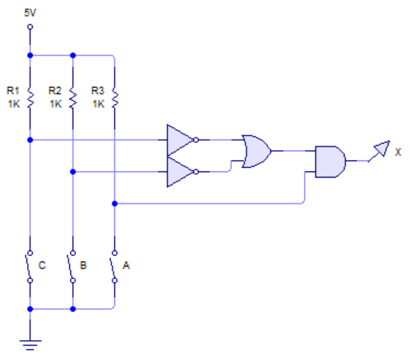
$x = \bar{C}\bar{B}A + \bar{C}BA + C\bar{B}A$
$y = \bar{C}B\bar{A} + \bar{C}BA + C\bar{B}\bar{A}$

Ya que conocemos las expresiones de ambas salidas, aplicaremos los teoremas antes expuestos para hacer las simplificaciones.

SALIDA X

$x = \bar{C}\bar{B}A + \bar{C}BA + C\bar{B}A$
Utilizaremos el teorema 8.
$x = \bar{C}A(\bar{B} + B) + C\bar{B}A$
$x = \bar{C}A + C\bar{B}A$
Ahora utilizaremos el teorema 17
$x = A(\bar{C} + C\bar{B})$
$x = A(\bar{C} + \bar{B})$

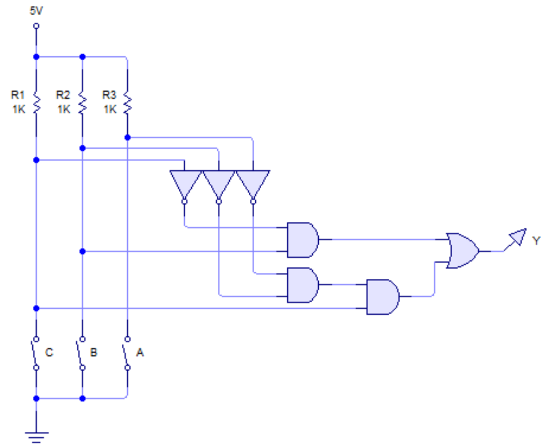
Ya que tenemos simplificada nuestra expresión a lo máximo procedemos a dibujar el circuito.



La comprobación de esta salida queda como tarea del lector.

SALIDA Y

$y = \bar{C}B\bar{A} + \bar{C}BA + C\bar{B}\bar{A}$
Utilizaremos el teorema 8,
$y = \bar{C}B(\bar{A} + A) + C\bar{B}\bar{A}$
$y = \bar{C}B + C\bar{B}\bar{A}$
Al ya no poderse reducir más la expresión dibujamos el circuito.



Ahora comprobaremos si el circuito cumple con los requerimientos de nuestra tabla de verdad, para ello realizaremos las mediciones en el mismo orden de la tabla de verdad. (la comprobación se la puede visualizar en la tabla que se encuentra en el anexo 2).

8. TEOREMAS DE DEMORGAN

[UNIVERSALIDAD DE LAS COMPUERTAS].

Los teoremas de De Morgan son una estrategia más para la simplificación de expresiones booleanas, pero con el gran beneficio que en el circuito solo utilizamos un tipo de compuertas ya sea NOR o NAND, en muchas ocasiones no será más fácil y económico utilizar esta estrategia para el diseño de nuestros circuitos, o al menos tenerla presente para hacer una muy buena elección en diseños.

Para realizar esto utilizaremos los siguientes teoremas:

8.1. $(\overline{A + B} = \bar{A}\bar{B})$ [NOR]

8.2. $\overline{AB} = \bar{A} + \bar{B}$ [NAND]

Como ejemplo tomaremos las expresiones simplificadas de los dos ejemplos anteriores y luego aremos un pequeño análisis para saber cuál de los circuitos es más económico y pequeño, recordando que deben cumplir estrictamente lo que dice la tabla de verdad.

$$x = A(\bar{C} + \bar{B})$$

$$y = \bar{C}B + C\bar{B}\bar{A}$$

Ya que tenemos las dos salidas convertiremos a la salida X en NOR y a la Y en NAND.

SALIDA X.

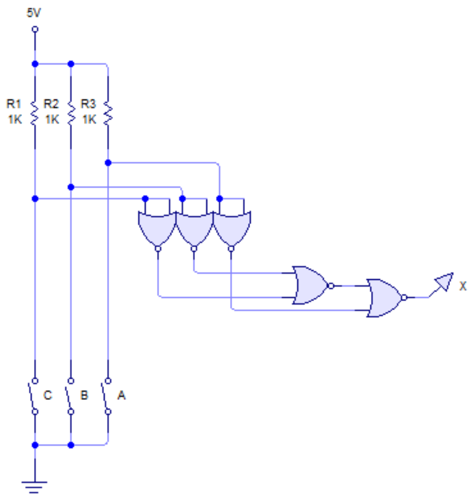
$$x = A(\bar{C} + \bar{B})$$

Primero negamos toda la expresión dos veces para no afectarla.

$$x = \overline{\overline{A(\bar{C} + \bar{B})}}$$

$$x = \overline{\bar{A} + \overline{(\bar{C} + \bar{B})}}$$

Ya tenemos nuestro circuito, el cual solo utiliza compuertas NOR, ahora dibujaremos el circuito.



SALIDA Y.

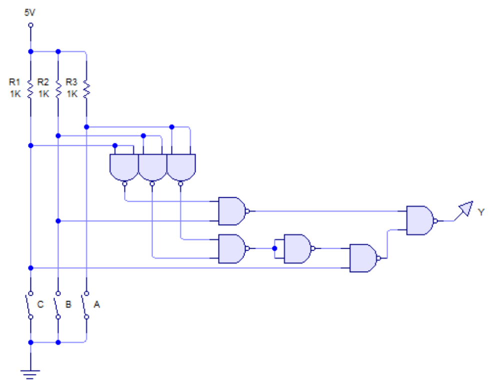
$$y = \bar{C}B + C\bar{B}\bar{A}$$

Primero negamos toda la expresión dos veces para no afectarla.

$$y = \overline{\overline{\bar{C}B + C\bar{B}\bar{A}}}$$

$$y = \overline{\overline{\bar{C}B}\ \overline{C\bar{B}\bar{A}}}$$

Visualizar el circuito a continuación.



La comprobación de los circuitos queda como tarea del lector.

Una vez que conocemos ambas salidas podemos realizar un análisis con respecto a las salidas que utilizamos solo por algebra de Boole, así es que en la salida X normal utilizamos un circuito integrado NOT, uno OR y uno AND, como cada circuito integrado tiene un costo de $0.50 en el mercado, si implementáramos el circuito gastaríamos $1.50, si implementamos el equivalente en NOR utilizaremos 2 circuitos integrados NOR y solo gastamos $1.00, aparentemente no es tanta la diferencia pero por que estos circuitos son pequeños y de ejemplo, pero en la vida diaria aremos circuitos muchos más grandes y complejos donde abra mucha más diferencia entre el costo de uno y de otro. En la salida Y con el circuito convencional también utilizamos un circuito integrado NOT, un OR y un AND, mientras que en el equivalente NAND utilizamos dos circuitos integrados, otra vez la diferencia es de tan solo $0.50.

En el tema hablamos de universalidad de las compuertas, con esto nos referimos a que es mucho mejor en un circuito electrónico tener un solo tipo de compuertas, se nos facilita la implementación, la revisión de fallas, es mucho más barato, en fin un sin número de facilidades.

9. MAPAS DE KARNAUGH.

Debido a que es muy complejo manejara expresiones booleanas con más de cuatro variables de entrada, también cabe decir que estos mapas denominados mapas K son muy utilices para tablas que tengan 4, 5 y 6 variables de entrada, ya que para un mayor numero se utiliza software de computadoras avanzados para facilitar la reducción de las expresiones.

Como ejemplo haremos un mapa K de seis variables de entrada y una de salida e implementaremos su circuito.

Ejemplo:

Por problemas de espacio no se dibujara la tabla debido a que es de seis variables de entrada entonces la dimensión de la tabla es de 64. Solo nos limitaremos a dibujar el mapa K.

DEF \ ABC	000	001	011	010	110	111	101	100
000	0	0	0	0	0	0	0	0
001	1	0	0	0	0	0	0	0
011	1	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
101	1	0	0	0	0	0	0	0
100	1	0	0	0	0	0	0	0

FIG 1. MAPA K SIN LAZOS.

DEF \ ABC	000	001	011	010	110	111	101	100
000	0	0	0	0	0	0	0	0
001	1	0	0	0	0	0	0	0
011	1	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
101	1	0	0	0	0	0	0	0
100	1	0	0	0	0	0	0	0

FIG 2. MAPA K CON EL PRIMER LAZO.

DEF \ ABC	000	001	011	010	110	111	101	100
000	0	0	0	0	0	0	0	0
001	1	0	0	0	0	0	0	0
011	1	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
101	1	0	0	0	0	0	0	0
100	1	0	0	0	0	0	0	0

FIG 3. MAPA K CON EL SEGUNDO LAZO.

DEF \ ABC	000	001	011	010	110	111	101	100
000	0	0	0	0	0	0	0	0
001	1	0	0	0	0	0	0	0
011	1	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
101	1	0	0	0	0	0	0	0
100	1	0	0	0	0	0	0	0

FIG 4. MAPA K CON EL TERCER LAZO.

DEF \ ABC	000	001	011	010	110	111	101	100
000	0	0	0	0	0	0	0	0
001	1	0	0	0	0	0	0	0
011	1	0	0	0	0	0	0	0
010	1	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
101	1	0	0	0	0	0	0	0
100	1	0	0	0	0	0	0	0

FIG 4. MAPA K CON TODOS LOS LAZOS.

Ahora ya que tenemos los lazos procedemos a obtener las expresiones directamente del mapa,

PRIMER LAZO

$$\bar{A}\bar{B}\bar{C}\bar{D}F$$

SEGUNDO LAZO

$$\bar{A}\bar{B}\bar{C}\bar{D}E$$

TERCER LAZO

$$\bar{A}\bar{B}\bar{C}D\bar{E}$$

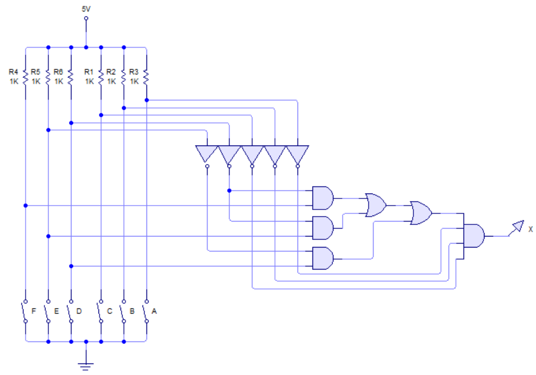
EXPRESIÓN

$$\boldsymbol{X = \bar{A}\bar{B}\bar{C}\bar{D}F + \bar{A}\bar{B}\bar{C}\bar{D}E + \bar{A}\bar{B}\bar{C}D\bar{E}}$$

Como se puede apreciar del mapa K ya obtenemos las expresiones lo mas simplificadas posibles, así que después podemos hacer pequeñas reducciones más fáciles en caso de que allá como.

$$\boldsymbol{X = \bar{A}\bar{B}\bar{C}(\bar{D}F + \bar{D}E + D\bar{E})}$$

CIRCUITO:



10. MSI (Media Escala de Integración).

Dentro de estos sistemas tenemos decodificadores, codificadores, pantallas de cristal líquido, multiplexores (MUX), demultiplexores (DEMUX). Nosotros aremos una pequeña introducción a cada uno de estos, explicaremos su funcionamiento muy ligeramente y expondremos unos ejemplos con su respectiva explicación para así dejar en claro la idea de que se trata cada funcionamiento.

a) DECODIFICADORES.

Los decodificadores son circuitos lógicos que cuya misión es recibir los datos de las entradas y activar ciertas salidas para que sean visualizadas en displays, otros tipos de decodificadores realizan el trabajo de recibir datos a las entradas de alguno de los sistemas de numeración lógica y entregar a la salida en otro sistema, ejemplo recibe a la entrada un sistema binario y entrega a la salida decimal. Los más usados son los circuitos integrados 74LS47, 74LS48.

b) CODIFICADORES.

Estos circuitos lógicos básicamente hacen el trabajo opuesto al de los decodificadores, esto quiere decir que en una de su entrada recibe una señal y entrega a sus salidas codificadas en código binario o BCD, etc.

c) MULTIPLEXORES.

Como su nombre lo indica multiplexa las entradas o selecciona datos, es una infinidad de aplicaciones que tienen estos, pero básicamente su misión es seleccionar datos.

Ejemplo:

Tenemos una sola entrada y podemos seleccionar la salida para que nuestra entrada este presente en esta durante un tiempo y luego podemos seleccionar otra salida.

d) DEMULTIPLEXORES.

Estos hacen el trabajo inverso que los multiplexores, tienen varias entradas pero una sola salida y podemos seleccionar cual de sus entradas queremos que esté presente a la salida.

e) DISPLAYS.

Los displays so elementos electrónicos que nos permiten visualizar dato como números (7 segmentos) o letras (alfanuméricos), mediante la ayuda de decodificadores (ánodo común o cátodo común).

f) COMPARADORES DE MAGNITUD.

Son circuitos que comparan entre un dato y otro para saber cual es mayor, menor o igual.

En el anexo 3 podemos visualizar un ejemplo básico para entender como funcionan cada uno de estos circuitos lógicos y las diferentes aplicaciones que podemos hacer con ellos. Además de esto al documento se adjunta un archivo digital con las simulaciones y los programas en los que se realizó cada una de estas para un mejor entendimiento de la información y un mejor aprendizaje.

Ubicación de las simulaciones:

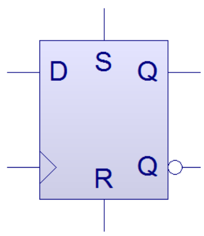
En la carpeta **"SIMULACIONES", anexo 3.**

11. FLIP FLOP.

En este tipo de elemento existen dos tipos, los tipos: SET RESET, D y los JK, explicaremos la simbología de cada uno, una breve descripción y la tabla de verdad.

- **FLIP – FLOP TIPO SET RESET.**

SIMBOLOGIA



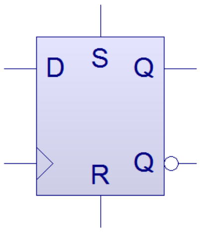
Como se puede apreciar en la simbología este tipo de flip flop son actives en flancos ascendentes.

TABLA DE VERDAD.

S	R	CLK		Q
0	0	↑		Q_0
1	0	↑		1
0	1	↑		0
1	1	↑		PROHIBIDO.

- **FLIP – FLOP TIPO D.**

SIMBOLOGIA



Como se puede apreciar en la simbología este tipo de flip flop son actives en flancos ascendentes y además de esto la característica principal es que lo que está en D tendremos a la salida.

TABLA DE VERDAD.

D	CLK		Q
0	↑		0
1	↑		1

- **FLIP FLOP TIPO JK**

SIMBOLOGIA

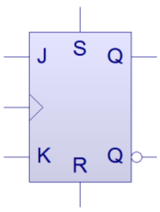


TABLA DE VERDAD.

J	K	CLK		Q
0	0	↑		Q_0
1	0	↑		1
0	1	↑		0
1	1	↑		CRUCE.

En el anexo 4 exponemos un circuito en donde se aplica los flip flops para elaborar un circuito en donde se encienda una secuencia de leds.

12. SUMADORES.

Este tipo de circuitos digitales cuentan con dos entradas cada una de cuatro bits, en donde suma la entrada A y la B, mostraremos un ejemplo en donde ingresamos números por medio del sistema BCD y resta y suma ambos números, ver anexo 5.

13. CONTADORES.

Es son circuitos digitales que cuentan el número de pulsos que reciban en la entrada del reloj, por lo general estos cuentan en el sistema binario, hay algunas clases, entre ellos podemos encontrar algunos que solo cuenten de forma ascendente, otros en donde podemos elegir si el conteo es de forma descendente o ascendente.

Con un pequeño y sistema en donde combinemos una LDR y un poco de imaginación podremos hacer un contador de eventos, este circuito se muestra en el anexo 6.

También en el anexo 7 podremos ver el circuito de un cronometro hecho solo a base de contadores un par de compuertas y unos pulsantes, muy práctico y fácil de implementar.

III. CONCLUSIONES.

IV. RECOMENDACIONES.