

# Facultad de Ingeniería Eléctrica

Laboratorio de Electrónica

"Ing. Luís García Reyes"

Materia: "Laboratorio de Electrónica Digital I"

## Practica Número 3

"Implementación de compuertas y circuitos lógicos con transistores"

### Objetivo:

En esta practica, se construirán circuitos lógicos basados en componentes discretos utilizando resistencias y transistores, se evaluar su funcionamiento así como la comprobación de las compuertas lógicas básicas.

### Introducción:

Los sistemas digitales emplean señales que tienen dos valores distintos conocidos como "1" y "0" lógicos. Existe una analogía directa entre señales binarias, elementos binarios de circuito y dígitos binarios.

Los sistemas digitales representan y manipulan además de números muchos otros elementos discretos de información.

Los sistemas digitales pueden representar cualquier elemento discreto de información utilizando el código binario adecuado

Los códigos deben de estar en Binario porque los circuitos digitales solo pueden interpretar y almacenar "unos" y "ceros".

Es importante entender que **los códigos binarios simplemente cambian los símbolos, no el significado de los elementos de información que representan**, por esto, si examinamos la información almacenada por ejemplo en una computadora, veremos que la mayor parte de la información representa algún tipo de información codificada no números binarios.

### Lógica Binaria

La lógica binaria se ocupa de variables que adoptan dos valores discretos y de operaciones que asumen un significado lógico. Los dos valores que pueden adoptar las variables reciben diferentes nombres.

- Verdadero, Falso
- Si, No
- Alto, Bajo
- Uno, Cero

- Etc.

En este caso, es conveniente pensar en ellos en términos de bits y asignarles los valores de "1" y de "0", a este tipo de lógica binaria que se implementa en esta práctica se le conoce como álgebra booleana. En esta práctica se dará una pequeña introducción al funcionamiento y uso de las compuertas "básicas" de los circuitos digitales y relacionarlas con las señales binarias.

La lógica binaria consiste en variables binarias y operaciones lógicas. Las variables se designan con letras del alfabeto como A,B,C,x,y,z etc. y cada variable tiene dos posibles valores 1 ó 0

En lógica binaria existen 3 operaciones básicas AND, OR, NOT.

## AND

Esta operación se representa con un punto (·) o bien omitiendo el operador por ejemplo:

$$x \cdot y = z$$

$$xy = z$$

A estas operaciones se leen como "x AND y es igual a z"

La operación lógica AND significa que  $z = 1$  solo si  $x = 1$  e  $y = 1$ , en cualquier otro caso  $z = 0$  (No se olvide que x y y son variables binarias y solo pueden tener un valor de 1 ó 0).

De estas condiciones se puede obtener una tabla que represente la operación AND.

Operación AND		
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

## OR

Esta operación se representa con un signo mas (+) por ejemplo:

$$x + y = z$$

Esta operación se lee como "x OR y es igual a Z"

En este caso la operación que realiza se puede describir como:  $z = 0$  si  $x = 0$  y  $y = 0$ , en cualquier otro caso  $z = 1$

De la misma manera es posible construir una tabla que represente la operación OR.

Operación OR		
x	y	z
0	0	0
0	1	1
1	0	1

1

1

1

## NOT

La representación de esta operación se realiza por un apóstrofe ( ' ) y en ocasiones por una barra (  $\bar{\phantom{x}}$  ) por ejemplo:

$$x' = z \text{ o como } \bar{x} = z$$

Se puede observar que esta operación solo tiene una entrada y una salida. De manera que la operación se lee como "x negada igual a z" y significa que z es lo contrario de x, Así si  $x = 1$ , entonces  $z = 0$ , y si  $x = 0$ , entonces  $z = 1$ . A esta operación se también se llama operación de complemento, ya que cambia "1" por "0" y "0" por "1". La tabla que representa la operación NOT es muy simple.

Operación NOT	
X	z
0	1
1	0

La lógica binaria se parece a la aritmética binaria y las operaciones AND y OR tienen similitudes con la multiplicación y la suma respectivamente, De hecho, los símbolos que se utilizan para la AND y OR son los mismos que los empleados para la multiplicación y la suma. No obstante, la lógica binaria no debe confundirse con la aritmética binaria.

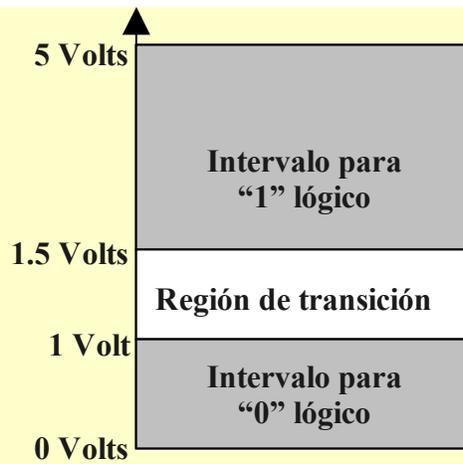
Debemos tener presente que una variable aritmética puede representar un número que puede constar de varios dígitos, mientras que una variable lógica solo se puede almacenar un solo dígito, es decir un "1" o un "0". Por ejemplo:

**En Aritmética Binaria tenemos  $1 + 1 = 10$ , (uno mas uno igual a dos)**

En lógica binaria sería  $1 + 1 = 1$  (uno AND uno igual a uno)

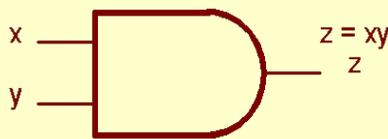
## Compuertas Lógicas

Las compuertas lógicas son circuitos electrónicos que operan con una o mas señales de entrada para producir una señal de salida. En los sistemas digitales que se implementan, responden a dos niveles distintos que representan una variable binaria cuyo valor lógico es uno lógico o cero lógico. Podemos citar por ejemplo a un sistema digital donde 5 volts es 1 lógico y 0 volts es cero lógico. En la práctica cada nivel de voltaje tiene un intervalo aceptable de acuerdo a la siguiente figura.

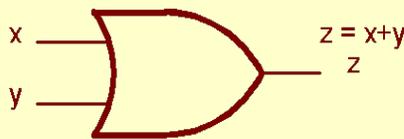


Las terminales de entrada de los circuitos digitales aceptan señales binarias dentro de un intervalo permisible y responden en las terminales de salida con señales binarias que están dentro del intervalo especificados. La región intermedia entre las regiones permitidas solo se cruza durante las transiciones de estado.

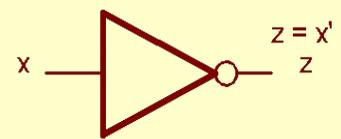
Las operaciones lógicas AND, OR y NOT descritas anteriormente también se representan de manera gráfica utilizando diferentes símbolos los cuales se muestran en la siguiente figura:



Compuerta AND de dos entradas



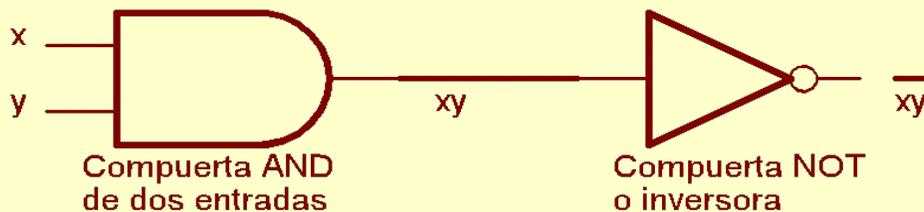
Compuerta OR de dos entradas



Compuerta NOT o inversora

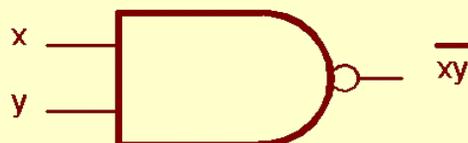
De estas compuertas básicas se desprenden mas compuertas adicionales, las cuales tiene indudablemente una gran aplicación en la implementación de circuitos. Entre las compuertas que se desprenden de las anteriores encontramos a:

La compuerta NAND, la cual la podemos definir como una compuerta AND seguida de una compuerta NOT, basándonos en la tabla de la operación AND y de la compuerta NOT, podemos obtener la siguiente tabla y su símbolo correspondiente.



Compuerta AND de dos entradas

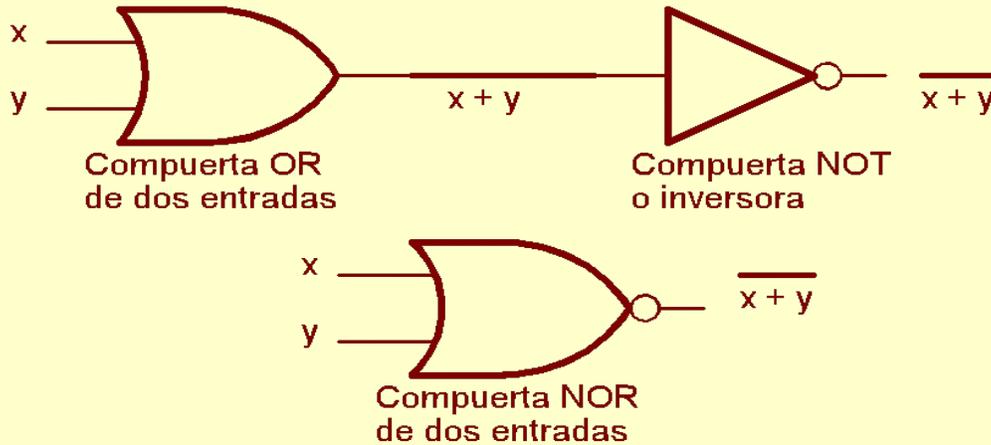
Compuerta NOT o inversora



Compuerta NAND de dos entradas

Operación NAND		
x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

De manera similar a la compuerta NAND, la compuerta NOR se puede definir como una compuerta OR seguida de una compuerta NOT, la misma tabla de operación de la compuerta e invirtiendo la señal de salida de la compuerta. Así en la siguiente gráfica se observa "la construcción" de la compuerta NOR así como su tabla de funcionamiento.



Operación NOR		
x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

### Construcción de compuertas lógicas

La construcción de las compuertas lógicas "básicas" presentadas en esta sección, se pueden realizar utilizando diferentes componentes como relevadores, diodos, transistores, etc. Sin embargo en esta prácticas se implementarán utilizando transistores, a la lógica basada en transistores se le denomina TTL y es una de las familias mas utilizadas por los diseñadores de circuitos lógicos discretos debido a las mejoras sobre las otras familias.

En nuestro caso se utilizara al transistor BC547 como transistor discreto.

### Requisitos:

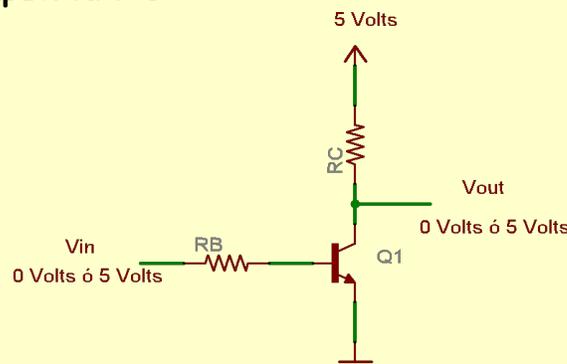
Para el desarrollo de esta práctica se requiere de la siguiente información:

- Investigar las principales características de la familia TTL
- Ganancia ( $\beta$  o  $h_{fe}$ ) del transistor utilizado
- Configuración de pines del transistor BC547 y BC337
- Configuración de los transistores para un circuito emisor común en la región de corte y de saturación, para obtener la resistencia de base y la resistencia de colector considerar una corriente de colector de 5 mA. y una fuente de 5 volts.
- Resumen de las tablas de operación de TODAS (5) las operaciones lógicas.

## Desarrollo:

La implementación de las compuertas lógicas basadas en transistor, se basa en la operación de los transistores en la región de corte y de saturación, estas regiones de operación permiten al transistor permite en la región de corte eliminar el flujo de corriente a través del transistor y en la región de saturación que fluya la corriente por el transistor, de manera que el transistor se comporta como un interruptor abierto o cerrado.

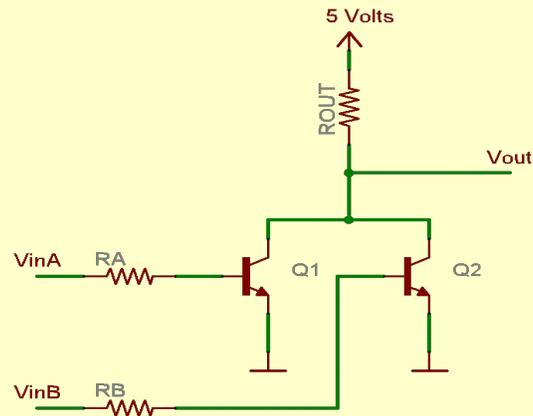
### Implementación de la compuerta NOT



De este circuito se puede apreciar que en caso de aplicar a la entrada 5 volts (1 lógico) el transistor se cierra, teniendo a la salida 0 volts (cero lógico), en caso de aplicar en la entrada un 0 volts (0 lógico), el transistor entra en la región de corte. Este funcionamiento corresponde al funcionamiento de la compuerta NOT. Siendo la más fácil de implementar.

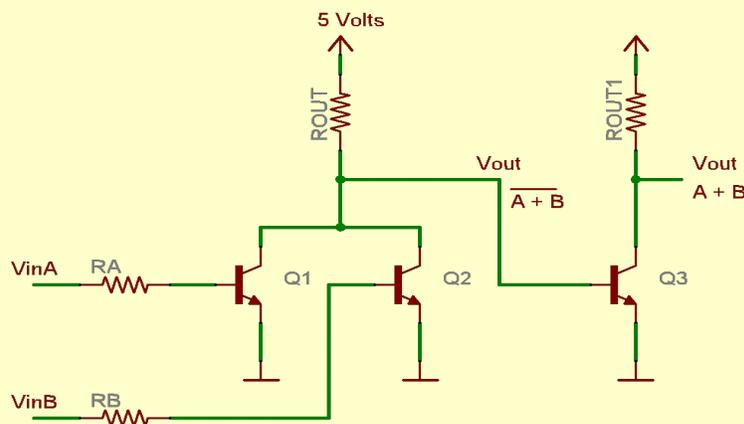
### Implementación de la compuerta NOR

La implementación de la compuerta NOR se realiza utilizando el circuito de la figura siguiente, del análisis del mismo, se desprende que en caso de conectar cualquiera de las entradas a 5 volts (nivel lógico 1), la salida se va a 0 volts, debido a la saturación de alguno de los transistores. Y la única posibilidad de que la salida este en 5 volts (1 lógico), es que ambas entradas sean de 0 volts (0 lógico).



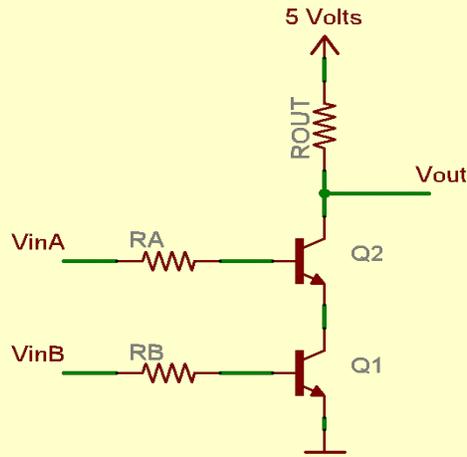
### Implementación de la compuerta OR

La compuerta OR se realiza utilizando el circuito de la compuerta NOR y agregando el circuito inversor, este arreglo permite invertir la señal de salida de la compuerta NOR y dar paso a la compuerta OR, como se puede apreciar la implementación de una compuerta NOR es mas simple que una compuerta OR al utilizar transistores tipo NPN, sin embargo en el diseño de una compuerta lógica se utilizan los transistores mas adecuados para la aplicación



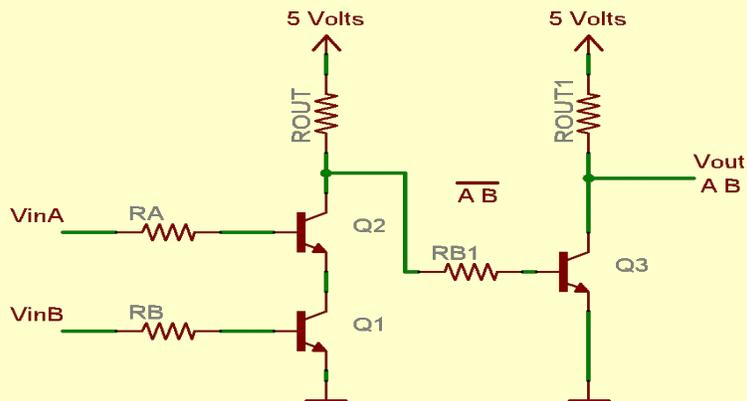
### Implementación de la compuerta NAND

La implementación de la compuerta NAND es la base de la lógica digital e indudablemente es la mas utilizada en la implementación de circuitos lógicos de la familia TTL, para el análisis de funcionamiento de esta compuerta lógica, se puede realizar por inspección y se puede observar que la única forma para que a la salida exista un cero lógico es que ambas entradas se encuentren activadas en un mismo tiempo, en caso de que solo una o incluso ninguna este activada, la salida de la compuerta lógica permanecerá en 5 volts (1 lógico)



### Implementación de la compuerta AND

La implementación de la compuerta AND se realiza de manera similar a la implementación de la compuerta OR que parte de la NOR, en esta caso se coloca una compuerta inversora a la salida de la compuerta NAND y se obtiene la compuerta AND. En este caso el número de transistores se incrementa debido a que se están utilizando transistores del tipo NPN, sin embargo en la construcción de circuitos lógicos es posible utilizar el tipo de transistor necesario para minimizar el número de componentes.



Implementar cada una de estas compuertas, y obtener la tabla de funcionamiento o tabla de verdad de cada una, llenando las siguientes tablas:

NOT	
InA	Out

NOR		
InA	InB	Out

OR		
InA	InB	Out

NAND		
InA	InB	Out

AND		
InA	InB	Out

Reportar:

Las tablas de verdad obtenidas en la práctica

Describir el funcionamiento de la compuerta AND implementada llenando la siguiente tabla:

Compuerta AND					
In A	In B	Q1	Q2	Q3	Out
0	0				
0	1				
1	0				
1	1				

En las columnas Q1, Q2 y Q3, se presentará el estado de los transistores, (corte o saturación) para cada una de las diferentes entradas de la compuerta