iadeArgentina

escuelas iade 50

ELECTRONICA
DIGITAL
7

TÉCNICAS DIGITALES

Paralelamente a la aparición de las computadoras nació una técnica llamada “SISTEMAS DIGITALES”, que sirvió de fundamentos a estas. El avance de estos sistemas se ha desarrollado tan ampliamente que en la actualidad son aplicadas no tan solo en dispositivos, digitales, sino también en dispositivos analógicos para funciones específicas, por ejemplo, en televisores, sistemas de audio, video games, lavadoras, cocinas, hornos, computadoras, calculadoras, relojes, control remoto, etc., en fin pocos son los artefactos donde aún no se aplica esta técnica.

El término analógico deriva del griego “análogo”, que significa: Algo que es como alguna otra cosa.

El término digital deriva del latín “digitus”, que significa: Dedo.

Los sistemas analógicos son proporcionales y continuos tan uniformemente como el ruido lo permita; en cambio, los sistemas digitales desarrollan procesos que son inherentemente discontinuos. Estos procesos evolucionan por dos estados lógicos definidos como (0 – 1, alto, abierto o cerrado, etc.), que pueden ser representado por pulsos de voltaje, corriente, magnetismo, etc. Cuando estos pulsos son debidamente contabilizados se pueden realizar operaciones tales como, suma, resta, multiplicación, división, almacenamiento de datos, etc.

Los sistemas lógicos de dos estados se llaman sistemas binarios y el término binario significa: compuesto de dos partes diferentes.

Cuando se elige uno de los dos estados lógicos, se tienen un dígito binario al cual se le denomina bit (BINARY DIGIT), que es la unidad de información más pequeño dentro de los sistemas digitales.

- Con 8 bits se tiene 1 BYTE.
- 1 KILO BYTE (Kb) = 1024 BYTE (128 x 8 = 1024).
- 1 MEGA BYTE (Mb) = 1048576 BYTE (1024 x 1024).
- 1 GYGA BYTE (Gb) = 1048576 x 1024 BYTE.

De este breve análisis se deduce que los sistemas digitales tienen un rol de primera importancia en la electrónica moderna, y por tal motivo resulta imprescindible comprender el sistema binario y conocer su aritmética de los dispositivos de tipo lógico, como además el conocimiento de otros sistemas numéricos de base diferente que nos permitan comprender el lenguaje de los sistemas digitales de máquinas microprosadoras, y microcomputadores.

FORMA DE REPRESENTAR UN NÚMERO

La base o raíz de un sistema numérico define la cantidad de dígitos que usa dicho sistema, por ejemplo, el sistema decimal opera con 10 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

En cualquier sistema podemos asignar un peso a cada dígito, y así representar un sistema pesado. Ejemplo 345_{10} (el valor 10 como subíndice indica en este caso la base del sistema que es 10).

Este número puede ser representado también del siguiente modo:

3 x 100 + 4 x 10 + 5 x 1 = 345
o también como potencia de 10.

$3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 = 345$

observará que cada dígito se le asigna un peso que corresponde a 10^0 para las unidades, ya que $10^0 = 1$; 10^1 para las decenas ya que $10^1 = 10$; 10^2 para las centenas ya que $10^2 = 100$, etc.

Otro sistema numérico es el binario el cuál tiene base 2, ya que dispone de dos dígitos (0 y 1) para representar las cantidades.

CONVERSIÓN DECIMAL A BINARIA

Para convertir un valor decimal en binario se divide el decimal sucesivamente por 2 para obtener cocientes enteros solamente, división que se prolonga hasta obtener un resto 0 ó 1. El primer resto encontrado lo llamaremos el bits menos significativo (bms), y el último resto que siempre será 1 lo llamaremos el bits más significativo (BMS).

Ejemplo:

	12/2 = 6
bms	0
	6/2 =3
	0
	3/2 = 1
	1
	1/2 = 0
BMS	1

12_{10} = 1100 2

que se lee uno uno cero cero 2 y no mil cien.

EJERCICIOS: Hallar el equivalente binario de los siguientes números decimales.

133 =	______________	68 =	______________
126 =	______________	85 =	______________
400 =	______________	18 =	______________
106 =	______________	34 =	______________

CONVERSIÓN DE BINARIO A DECIMAL.

Cada dígito o bits tiene asignado un peso en base 2 de orden creciente a partir del bit menos significativo. Para efectuar la transformación a decimal sumaremos todos los pesos para los que el bits tome el valor 1.

Ejemplo:

16	8	4	2	1
2^4	2^3	2^2	2^1	2^0
1	0	1	1	0_2

$$\begin{array}{r} 16 \\ +\,4 \\ +\,2 \\ \hline 22_{10} \end{array}$$

$10110_2 = 22_{10}$

EJERCICIOS: Transformar a decimal los siguientes números binarios.

10101_2 =

1111_2 =

11100_2 =

1001_2 =

Binarios con signo: Normalmente en el lenguaje de microprocesadores se opera con datos de 8 bits, lo que permite representar números sin signos o resta hasta 255. Pero cabe la necesidad de representar números negativos o positivos convencionalmente en un sistema numérico de 8 bits, el octavo bits se usa para la representación del signo siendo ésta positivo cuando el bits 7 es 0, y negativo cuando el bits 7 es 1.

Ejemplo:

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	..... representa + 1_{10}
1	0	0	0	0	0	0	1	..... representa - 1_{10}
0	1	1	1	1	1	1	1	..... representa + 127_{10}
1	1	1	1	1	1	1	1	..... representa – 127_{10}

bits signo

Como se observa usando representaciones con signos, el número mayor que se puede representar es 127_{10}, por lo que representar cantidades mayores será necesario aumentar la cantidad de bits a utilizar en la representación. Por ejemplo, usando 16 bits se puede representar números con signos hasta 32768_{10}.

SISTEMAS NUMÉRICOS UTILIZADOS EN ELECTRÓNICA DIGITAL

DECIMAL	BINARIOS NATURAL	OCTAL	HEXADECIMAL
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F

NÚMEROS HEXADECIMALES

El sistema hexadecimal de números es el sistema de números de base 16, que utiliza los símbolos 0-9, A, B, C, D, E, F.

La ventaja del sistema hexadecimal es su facilidad de conversión directa a número binario, de cuatro bit, en la tabla numérica se puede observar que cada número binario de cuatro bit, o sea, del 0000 al 1111, puede representarse por un solo dígito hexadecimal.

Al fijarse en la columna hexadecimal de la tabla numérica se puede ver que el equivalente de 16 en el sistema hexadecimal es 10 lo que demuestra que el sistema hexadecimal también emplea el concepto de valor de posición. El 1 en 10_{16} significa 16 unidades, mientras que el 0 representa cero unidades.

CONVERSIÓN DE HEXADECIMAL A DECIMAL

Para tal efecto, se confecciona un sistema pesado, en el que cada dígito o bits tiene asignado un peso en base 16 de orden creciente a partir del bits menos significativo el que se multiplica por cada símbolo del sistema hexadecimal, luego los valores parciales que se obtengan se suman para, obtener la conversión a decimal.

Ejemplo:

	16_2	16_1	16_0
BMS	256	16	1
	X	X	X
	2	B	6
	512 +	176 +	6 = 694_{10}

$2B6_{16} = 694_{10}$

EJERCICIOS: Convertir a decimal los siguientes números hexadecimales.

$43C_{16}$ =

FF_{16} =

248_{16} =

CONVERSIÓN DE DECIMAL A HEXADECIMAL

Se divide el número decimal por 16 en forma sucesiva hasta obtener un resto final menor que 16.

Ejemplo:

584_{10} / 16 = 36
104
bms **08**

36 / 16 = 2
4

2 / 16 = 0
BMS **2**

EJERCICIOS:

1084_{10} =

255_{10} =

CONVERSIÓN DE HEXADECIMAL A BINARIO

En este caso para la conversión diremos que cada dígito hexadecimal será representado en su equivalente binario, pero empleando en cada caso 4 bits.

Ejemplo:

F	F_{16}	4	3	C_{16}
↓	↓	↓	↓	↓
1111	1111_2	0100	0011	1100_2

2	4	8_{16}
↓	↓	↓
0010	0100	1000_2

CONVERSIÓN DE BINARIO A HEXADECIMAL

Para la conversión debemos separar el número binario en grupos de 4 bits empleando desde el bits menos significativo, y luego representando cada grupo por su equivalente en hexadecimal.

Ejemplo:

1100	1010	1001_2
C	A	9_{16}

$110010101001_2 = CA9_{16}$

110	1110	1110_2
6	E	E_{16}

$11011101110_2 = 6EE_{16}$

ARITMÉTICA BINARIA

El sistema binario de numeración admite las mismas operaciones aritméticas que el sistema decimal, es decir que podrán realizar es la suma, resta, multiplicación y división de números binarios. Para nuestro propósito tienen sumo interés las operaciones aritméticas.

1- SUMA DE NÚMEROS BINARIOS

Para efectuar la suma binaria basta tener presente las siguientes reglas.

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 con reserva de 1.
1 + 1 + 1 = 1 con reserva de 1.

Ejemplo:

```
   1  1(reserva)
      1  1  0  1 ....... 13₁₀
      1  1  1  0 ....... 14₁₀
+  ______________       ____
   1  1  0  1  1₂        27₁₀
```

```
   1  1  1  1(reserva)
      1  1  1  1 ............   15₁₀
         1  1  1 ............    7₁₀
+  ______________            + ____
   1  0  1  1  0₂               22₁₀
```

Ejercicios:

```
   1  0  0  1   ........     9
+     1  0  1   ........  +  5
   ___________              ___
```

```
   1  0  0  1  1 ...........  19
+  1  1  0  0  1 ........... +25
   _____________              ___
```

```
   1  0  1  1  1 ...........  23
+  1  1  0  1  1 ........... +27
   _____________              ___
```

Cuando la cantidad de sumandos es mayor que dos, se procede a seccionar la suma en sumas parciales de dos sumandos y luego a hacer lo mismo con los resultados parciales, hasta llegar al resultado final.

```
  1 1 0 1 1  (1)
  1 0 1 1 0  (2)
  1 0 1 1 1  (3)
+ 1 1 1 1 0  (4)
 ____________
```

COMPUERTAS LÓGICAS BÁSICAS Y TABLAS DE VERDAD. Todo sistema digital, sea combinacional o secuencial, se encuentra implementado a partir de tres circuitos u operadores básicos, denominados de acuerdo a la función AND, OR y NOT. Esto significa que si empezamos a descomponer un sistema en partes cada vez más elementales, tendremos finalmente una cuya arquitectura corresponderá a un conexionado de operadores lógicos básicos.

Estos operadores básicos se definen como sistemas electrónicos que basándose en la numeración binaria permiten el desarrollo de múltiples operaciones matemáticas, almacenamiento de datos, control exacto del tiempo, etc.

Las compuertas se pueden clasificar en:

a) Compuerta u operador AND. ....

b) Compuerta u operador OR. ....

c) Compuerta u operador NOT. ....

Cada compuerta está diseñada para una función determinada y esa función se puede comparar mediante una tabla que indica todas las posibles combinaciones de entrada y su salida. Esta tabla recibe el nombre de "Tabla de verdad". La cantidad de combinaciones que es capaz de manejar una determinada compuerta lógica dependerá del número de entradas; la base del sistema que es 2 se eleva a un exponente igual a su número de entrada:

*Para una compuerta de dos entradas:

2 = 2. 2 = 4 combinaciones de entrada

Combinaciones de entrada.

A	B	A . B
0	0	
0	1	
1	0	
1	1	

*Para una compuerta de tres entradas:

2 = 2 . 2 . 2 = 8 combinaciones de entrada.

Combinaciones de Entrada.

A	B	C	A . B . C
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Para comprender el funcionamiento de las compuertas lógicas, también se hace necesario entender que:

a) Un pulso negativo es igual a "0"
b) Un pulso más negativo que otro es "0"
c) Un pulso positivo es igual a "1"
d) Un pulso más positivo que otro es iguala a "1"

Lógica positiva

a) Un pulso negativo es igual a "1".
b) Un pulso más negativo que otro es igual a "1".
c) Un pulso es igual a "0".
d) Un pulso más positivo que otro es igual a "0".

Lógica negativa

DEFINICIONES FUNDAMENTALES:

VARIABLE BINARIA. Es aquella que puede tomar solo dos valores distintos. Estos valores pueden designarse en diferentes formas:

a) "1" ó "0".
b) "alto" o "bajo".
c) "si" o "no".
d) "cierto" o "falso".

Ejemplos de estados binarios:

a) El estado de un interruptor puede ser abierto o cerrado.
b) El estado de un lámpara puede ser encendida o apagada.
c) El estado de un transistor puede ser saturado o al corte.

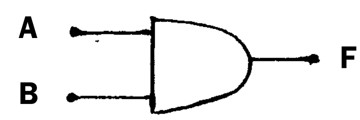
FUNCIÓN LÓGICA: Una función lógica puede ser dependiente de una o más variables, pero no puede aceptar más de dos valores: "1" ó "0".
La función lógica no queda definida si no se precisa que valor adopta para cada combinación posible de los valores de las variables.

TABLA DE VERDAD._ Otra forma de expresar una función es mediante una tabla de verdad. La tabla de verdad es una forma cómoda de resumir los valores de una función para todas las combinaciones posibles de las variables de entrada. Si la función depende de n cantidad de variables de entrada, habrá 2^n combinaciones posibles para los valores de las variables. La función se define por una tabla de verdad de varias columnas (una por cada variable y una para la función) y varias líneas 8una para cada combinación de valores).

E1	E2	F
0	0	0
0	1	0
1	0	0
1	1	1

COMPUERTA Y (AND). La característica principal de esta compuerta es que para tener una salida "1", todas sus variables de entrada deben valer "1".
Esta compuerta puede tener dos ó más entradas, pero una salida.

Símbolo Eléctrico



Circuito equivalente

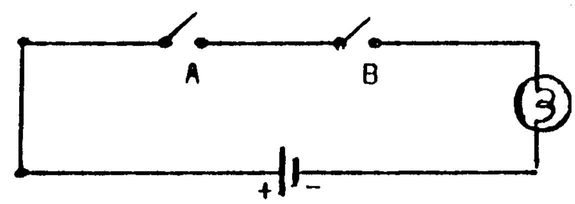


Tabla de verdad

Entradas		Salida
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Ecuación de salida $F = A \,.\, B$

Esta compuerta es equivalente a dos o más interruptores conectados en serie, de modo de que para que circule la señal o corriente es necesario que todos los interruptores estén cerrados, pues basta que esté abierto uno solo para que la corriente se interrumpa.
Si designamos por uno al estado encendido y por cero el de apagado, y por uno al estado de conexión y

por cero al estado de desconexión de los interruptores, tenderemos la tabla de verdad mostrada en la figura.

En la práctica el cero binario se define como un voltaje bajo o masa u el uno binario se define como un voltaje alto, el cual puede ser de 3 – 5 ó más volts.

El álgebra booleana es una forma simbólica que muestra como operan los circuitos lógicos, mediante un método taquigrafico.

La expresión booleana para la compuerta AND indicada anteriormente es:

$$F = A \cdot B \qquad \text{ó} \qquad F = AB$$

Esto se lee, A y B igual F. El punto entre las dos letras no indica multiplicación.

Las leyes formales del álgebra booleana para la compuerta AND son:

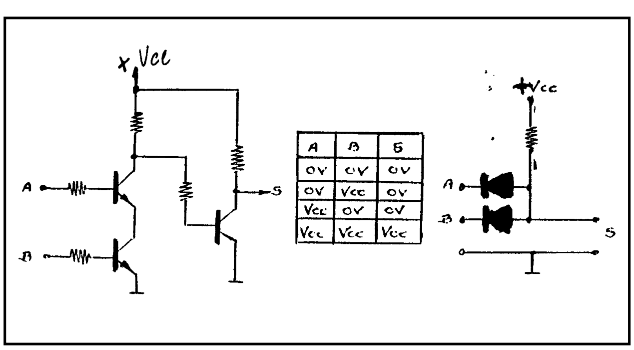
$$A \cdot \overline{A} = 0$$
$$A \cdot 1 = A$$
$$A \cdot 0 = 0$$
$$A \cdot A = A$$

La barra sobre la variable A representa el complemento de A(opuesto) y se lee "A negado". En álgebra de BLOOE el complemento de cero es uno y el de uno es cero. Por lo tanto:

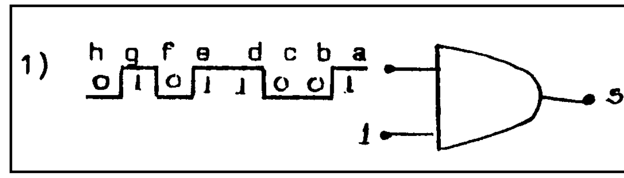
$$A = \overline{\overline{A}}$$
$$\overline{1} = 0$$
$$\overline{0} = 1$$

En la figura se muestra la tabla de la verdad de la función AND.

El operador que realiza la función AND se puede implementar eléctricamente de muchas formas diferentes, tal como lo muestra la figura siguiente, donde los estados eléctricos considerados en la operación de los circuitos corresponden a niveles de voltaje.

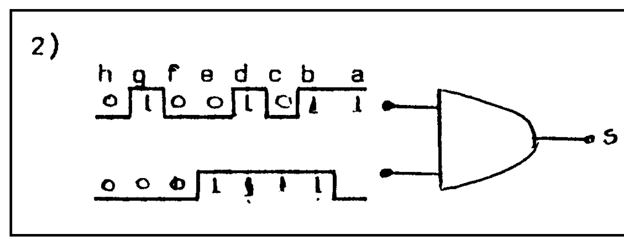


EJERCIIOS: Indique cuales serán los pulsos de salida en las siguientes compuertas AND:



Solución

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =

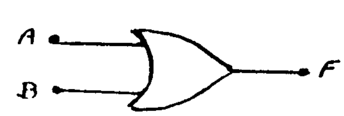


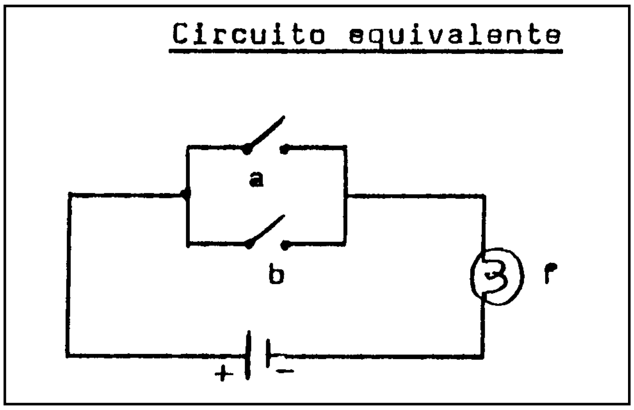
Solución

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =

COMPUERTA 0 (OR). Esta compuerta se caracteriza por tener salida 1 cuando una ó todas sus entradas valen 1.

Símbilo eléctrico





ENTRADAS		SALIDA
B	A	F
0	0	0
0	1	1
1	0	1
1	1	1

Ecuación de salida.

F = A + B

Esta compuerta es equivalente a dos interruptores conectados en paralelo, de modo que para que circule la corriente basta que uno cualquiera de ellos se encuentre cerrado.

La expresión booleana para la compuerta o (OR) indicada anteriormente es:

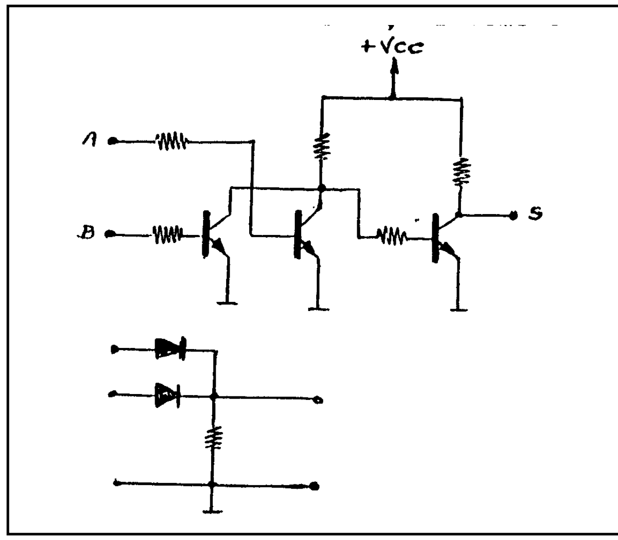
F = A + B

Esto se lee como A o B es igual a F. El signo más en este caso no indica suma.

La compuerta O (OR), al igual que la Y (AND), puede tener varias entradas pero una salida.

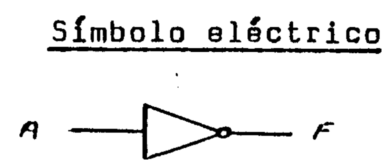
En la figura se muestra un circuito formados por dos interruptores conectados en paralelo y en serie con una lámpara indicadora. De acuerdo con la configuración del circuito, la lámpara encenderá si se cierra el interruptor a ó el b o ambos interruptores. La tabla de verdad muestra la situación del circuito, para cada una de las posibles combinaciones de entrada.

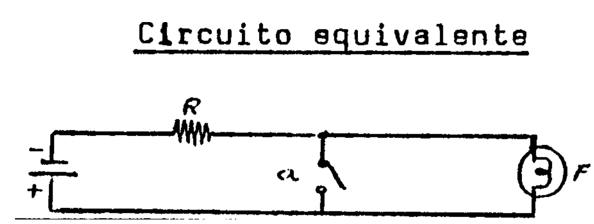
Al igual que la compuerta y (AND), la función o (OR) se puede obtener por distintos medios electrónicos, tal como se muestra en la figura siguiente:



A	B	S
OV	OV	OV
OV	Vcc	Vcc
Vcc	Ov	Vcc
Vcc	Vcc	Vcc

COMPUERTA NO (NOT). A esta compuerta también se le conoce con el nombre de “INVERSOR” O “NEGADOR” y se caracteriza por tener siempre en su salida un estado lógico opuesto al de su entrada.





Ecuación de salida.

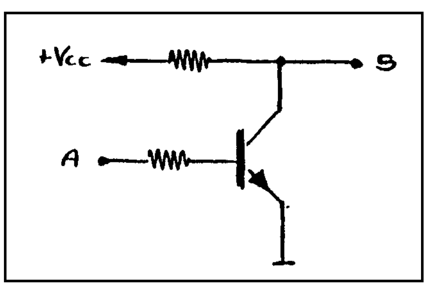
$F = \overline{A}$

Tabla de verdad

Entrada	Salida
A	F
0	1
1	0

La configuración circuital de este operador lógico se muestra en la figura siguiente, donde tenemos un circuito formado por un transistor en montaje emisor común. Supongamos que la base del transistor no posea polarización, el transistor se mantendrá en la condición de corte, por lo cual el voltaje de salida se mantendrá en un valor cercano al de Vcc; por el contrario, cuando la base del transistor posea polarización, el voltaje de salida, que corresponde al existente entre el colector y emisor se mantendrá en un valor cercano a cero volts. Resumiendo, podemos apreciar que al situación de la salida de este circuito corresponde siempre a la condición opuesta a la existente en la entrada. Esto queda de manifiesto en la tabla de verdad se que se muestra junto a la configuración circuital.

A	S
0V	Vcc
Vcc	0V



Como ya dijimos en un principio, todo sistema digital, cualquiera sea su envergadura, se encuentra implementando a partir de estos res operadores lógicos básicos, los cuales al ser conectados en distintas configuraciones, darán forma a nuevos sistemas capaces de realizar otras funciones lógicas.

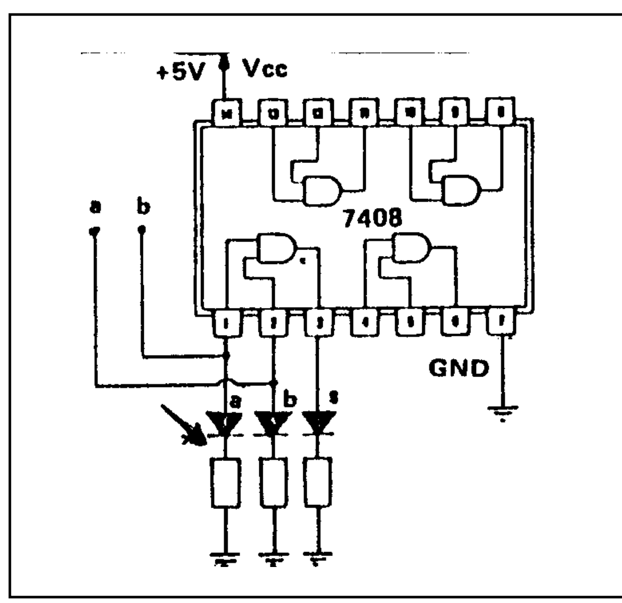
Cuando se implementan sistemas digitales, no es necesario considerar los aspectos electrónicos de la operación del sistema, esto debido a que la tecnología actual de alta integración permite que en un solo circuito integrado (CHIP) pueden "colocarse" todos los componentes electrónicos que constituyen a un sistema digital que realice una función específica; así, por ejemplo, existen circuitos integrados que contienen los operadores lógicos básicos que hemos estudiado.

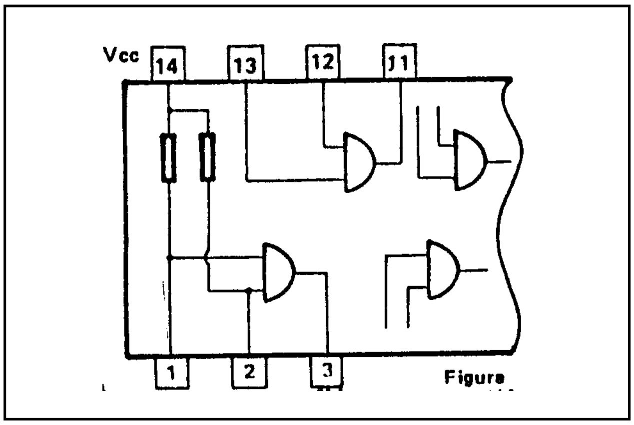
En la figura siguiente se muestran los circuitos integrados TTL 7408, 7432 y 7404, que contienen los operadores AND, OR y NOT respectivamente.

OBTENCIÓN DE TABLAS DE VERDAD. En el diagrama siguiente se muestra la configuración circuital en que se ha colocado el chip 7408, con el objeto de obtener la tabla de verdad de uno de sus operadores lógicos AND. Este chip contiene cuatro operadores lógicos AND de dos entradas cada uno, de los cuales en el diagrama de la figura se hace uso del que se encuentra conectado entre los terminales 1, 2 y 3 del circuito integrado 7408. En cada uno de los terminales de entrada se encuentra un diodo led, cuya finalidad es indicar visualmente el estado lógico de cada entrada. Otro led se encuentra conectado a la salida del operador a objeto de mostrar en todo momento la situación de salida.

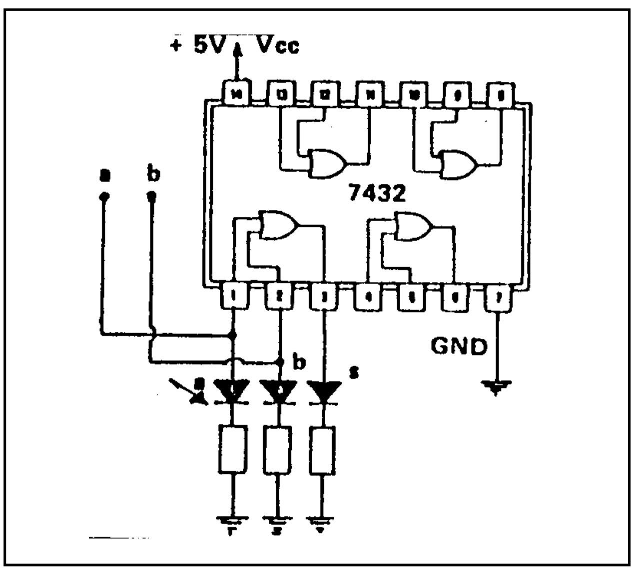
La tabla de la verdad se obtiene realizando con las entradas todas las posibles combinaciones, conectando a Vcc para obtener un "uno" y a masa para obtener un "cero".

Nótese que cuando todas las entradas quedan desconectadas (flotantes), el led de salida indica la existencia de un uno lógico en la salida, esto se debe a que cuando una entrada se encuentra flotante, esta queda conectada internamente a través de una resistencia a Vcc, tal como lo muestra la figura siguiente:



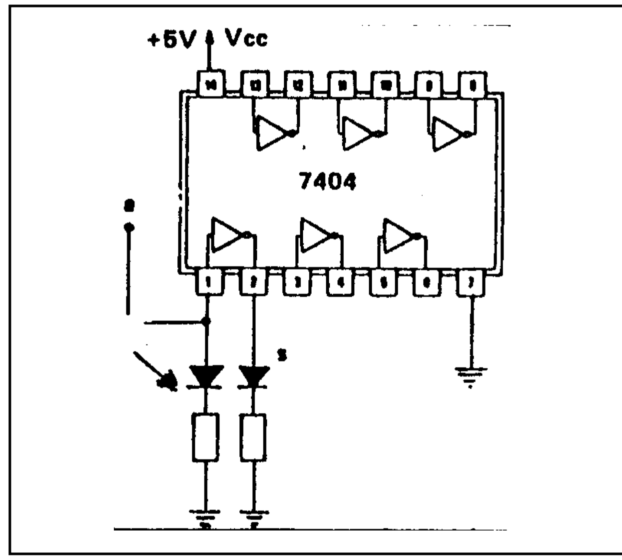


En la figura siguiente se muestra la disposición circuital para obtener la tabla de verdad del operador OR, para lo cual se utiliza ahora el chip 7432 el que contiene cuatro operadores OR de dos entradas cada uno.



Al igual que en el caso anterior, un led encendido indica un estado lógico "uno", mientras que un led apagado indica un estado lógico "cero".

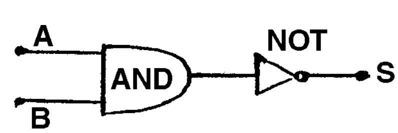
Finalmente, en la próxima figura se muestra el circuito de prueba para obtener la tabla de verdad de un operador NOT, para lo cual se utiliza el chip 7404, el que contiene seis operadores NOT o inversores. En este caso al igual que en los anteriores, el resto de los operadores contenidos en el chip realizan exactamente la misma función.



Hasta aquí hemos analizado la operación de los denominados circuitos básicos de lógica combinacional, como se ha podido observar, estos trabajan en función de solo dos estados eléctricos, los cuales se encuentran claramente diferenciados entre sí. El comportamiento de los circuitos lógicos estudiados, se ha representado a través de tablas de verdad, en las cuales se indican todos los posibles estados que puede tomar la salida, en función de las combinaciones que se pueden realizar en los estados de entrada.

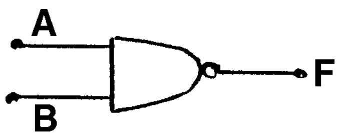
Combinando la acción de los circuitos lógicos estudiados, se obtienen los circuitos compuestos, cuya operación se deberá también considerar la operatoria de los circuitos lógicos elementales, ya que básicamente las salidas de estos serán funciones obtenidas de funciones básicas; esto por último será analizado en mayor detalle más adelante.

COMPUERTA NAND. Como se muestra en la figura siguiente, este circuito se obtiene conectando un inversor a la salida de un circuito AND, en consecuencia, tal como se puede apreciar en la misma figura, la salida de este circuito corresponde al de la compuerta AND negada, ya que entrega un cero a su salida solamente cuando todas sus entradas se encuentran en estado lógico "uno", siendo la salida "uno" si una o más entradas se encuentran en estado cero.

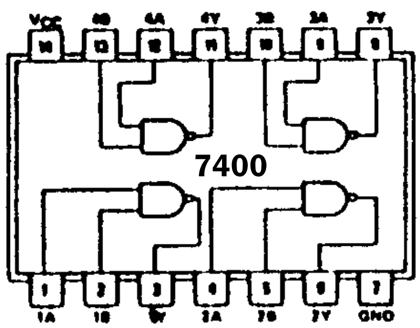


A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

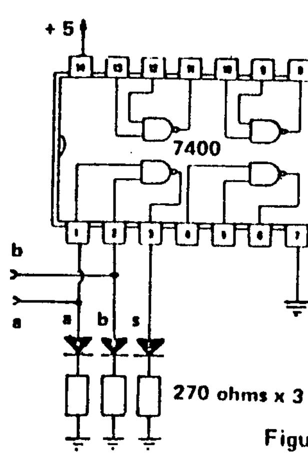
Normalmente el símbolo utilizado para representar esta compuerta es el mostrado en la figura siguiente:



El circuito integrado que se muestra a continuación (IC 7400) contiene cuatro compuertas NAND, las cuales quedarán en condiciones de operar al aplicar polarización entre los terminales 7 y 14 del chip.

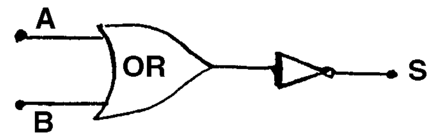


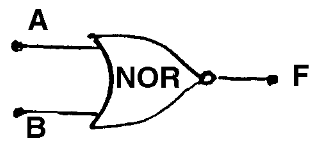
Para obtener la tabla de verdad del circuito en cuestión será necesario conectar una de las compuertas contenidas en el chip tal como lo muestra la figura siguiente. Luego mediante los terminales que han quedado libres (a y b), se realizarán las distintas combinaciones de entrada o objeto de obtener la correspondiente tabla de verdad.



Figura

COMPUERTA NOR. Este circuito se obtiene conectando a la salida de una compuerta OR un circuito inversor, lo cual se muestra en la figura siguiente:





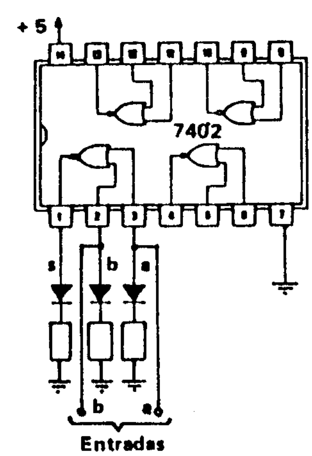
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

En esta figura se muestra además la función de la compuerta mediante la correspondiente tabla de verdad. Básicamente la compuerta NOR entrega un función, la cual corresponde a la negación de la función realizada por la compuerta OR.

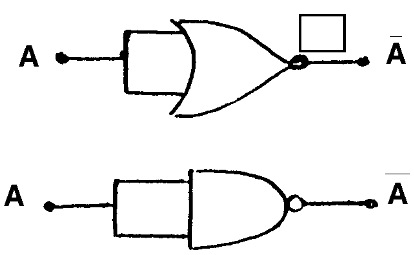
En la figura también se muestra el símbolo utilizado para representar a esta compuerta.

El circuito que permite obtener la tabla de verdad de este circuito se muestra en la próxima figura, donde se utiliza una compuerta que se encuentra contenida en el integrado 7402; al igual que en el caso anterior, las combinaciones se realizarán mediante los terminales indicados como “a” y “b”, correspondientes a las entradas del circuito.

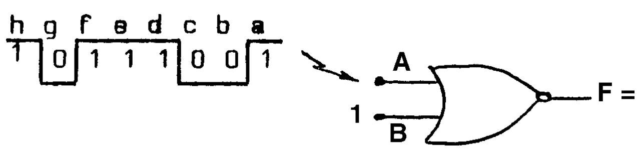
Es muy probable que en algunos casos, los estados de salidas correspondientes a “uno lógico”, no resulten muy claros a través de los diodos led que se utilizan como indicadores, esto último se debe a que algunos circuitos integrados no poseen la suficiente capacidad de corriente en sus salidas en estado lógico “uno”, por esta razón se recomienda utilizar algún tipo de indicador que produzca una mínima carga sobre el circuito bajo prueba.



Nota: Cuando se puentean las entradas de una compuerta NAND o NOR, estas pasan a comportarse como una compuerta NOT o inversora.



EJERCICIOS: Indique cual el tren de pulsos de salida en la compuerta NOR si la entrada en b se encuentra en 1 lógico:



Solución

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =

COMPUERTA OR EXCLUSIVA (XOR). Esta compuerta se caracteriza por entregar en su salida un bit "1" solo cuando en sus entradas hay un número impar de "1", por esta razón se le considera como un detector de número impar de "1".

El circuito lógico XOR puede ser formado por compuertas AND – OR e inversores.

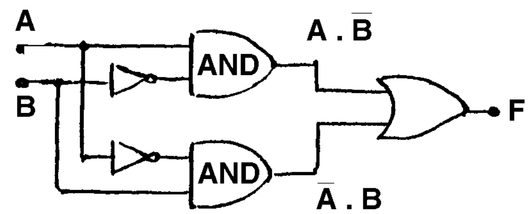
Símbolo XOR

Ecuación de salida

$F = A \oplus B$

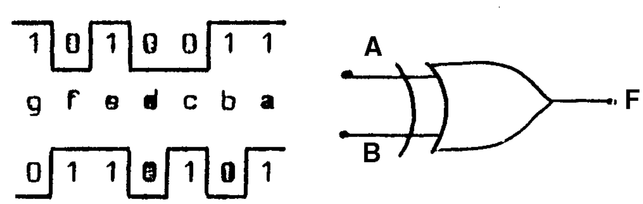
XOR

Circuito lógico que realiza la función XOR



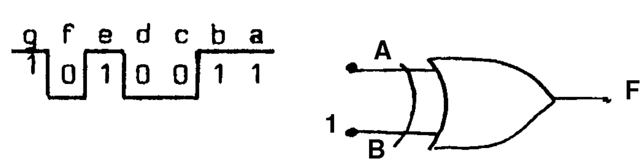
Entradas		Salida
B	A	F
0	0	0
0	1	1
1	0	1
1	1	0

EJERCICIOS: Indique cual será el tren de pulsos de salida de las siguientes compuertas:



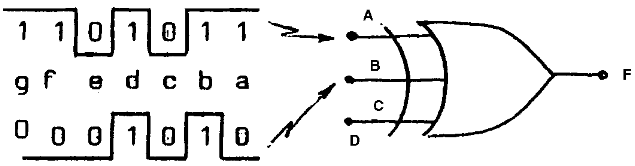
Solución

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =



Solución

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =



Solución

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =

COMPUERTA NOR EXCLUSIVA (XNOR). Esta compuerta se caracteriza por negar las funciones XOR y por este motivo se considera a esta compuerta como un detector de número par de "1".

Símbolo XNOR

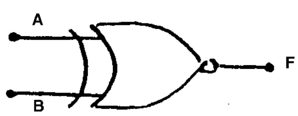
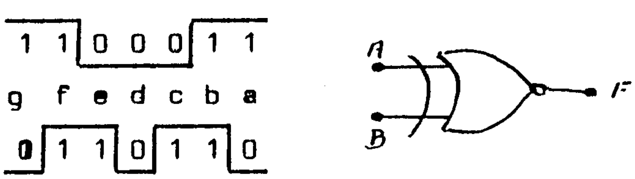


Tabla de verdad para las compuertas XOR y XNOR

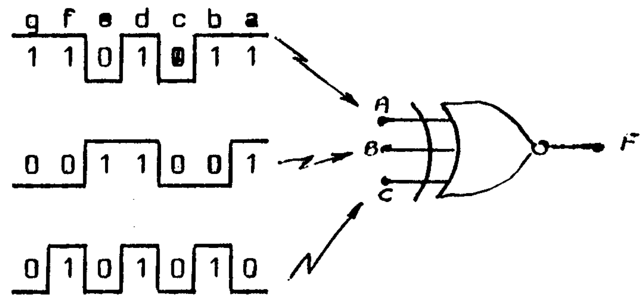
Entradas		Salidas	
B	**A**	**XOR**	**XNOR**
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

EJERCICIOS: Indique cual será el tren de pulsos de salida de las siguientes compuertas:



Solución:

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =

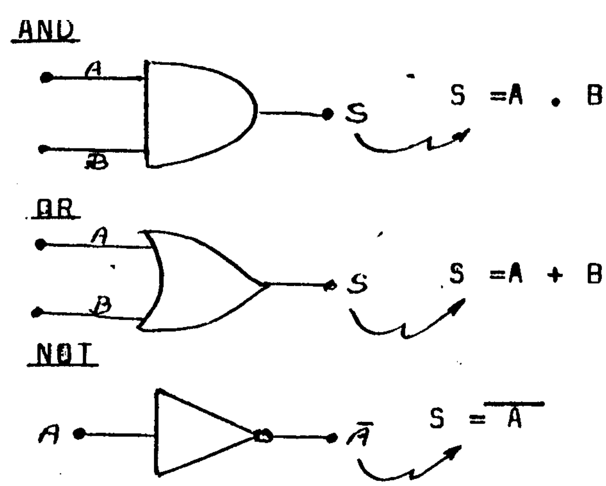


Solución

Pulso a =	Pulso e =
Pulso b =	Pulso f =
Pulso c =	Pulso g =
Pulso d =	Pulso h =

EXPRESIÓN LITERAL DE LAS FUNCIONES LÓGICAS

Con el objeto de simplificar tanto la representación como el procesamiento de las funciones lógicas, es que estas se representan en forma literal de acuerdo a lo indicado en la figura siguiente:



De acuerdo a lo indicado, la función AND se representa relacionando las variables de entrada con el símbolo "." o bien, " ", tal como se muestra a continuación:

Y = F (a.b.c.d) o bien Y = abcd.

Esto se lee “la salida Y es función de las variables de entrada a, b, c y d”, lo cual significa que la salida del circuito AND depende en todo momento del estado en que se encuentran las variables de entrada o identificadas como a, b, c y d.

La función OR se representa relacionando las variables de entrada con el símbolo “+”, en consecuencia, toda vez que aparezca el signo MAS entre las variables de un sistema, significa que estas se encuentran conectadas a la entrada de una compuerta OR.

Así entonces, la función OR se representara literalmente tal como se indica a continuación:

Y = F (a+ b +c +d) o bien Y = a+b+c+d

Tanto en la función AND como en la función OR, el número de variables de entrada puede ser cualquiera, lo cual significa que podremos tablas de verdad con muchas combinaciones.

Finalmente la función NOT, se representa mediante una línea sobre la variable negada o también mediante una comilla, tal como se muestra a continuación:

Y = a = á o bien Y = a =‘a

Todo circuito de lógica combinacional se puede representar a través de una expresión literal, así como también es posible transformar una expresión literal a u circuito de lógica combinacional.

Para transformar una expresión literal a un circuito de lógica, es necesario seguir el siguiente orden de prioridad de las operaciones lógicas.

1) Resolver paréntesis.
2) Realizar operaciones AND.
3) Realizar operaciones OR.

Los complementos se resolverán en el orden que corresponda para ejecutar las operaciones antes indicadas.

En la figura siguiente se muestra como se transformo a un circuito lógico la expresión literal:

F = a . (ā + b)

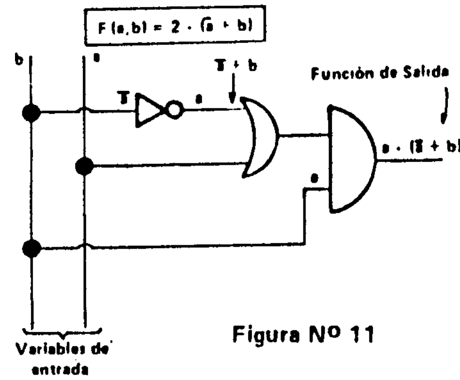


Figura Nº 11

Nótese que en la implementación de este circuito se ha utilizado una compuerta OR de dos entradas, una compuerta AND también de dos entradas y un inversor NOT.

En este diagrama las dos líneas verticales son los conductores que poseen el estado de las variables a y b. Luego, siguiendo el orden de prioridad de las operaciones lógicas se resuelve el paréntesis, de lo cual resulta una compuerta OR de dos entradas a las cuales deben llegar las variables a y b. Nótese que la variable a debe legar complementada, razón por la cual se utiliza un inversor. En estas condiciones a la salida de la compuerta OR se tendrá una función lógica que tendrá la siguiente expresión literal:

ā + b

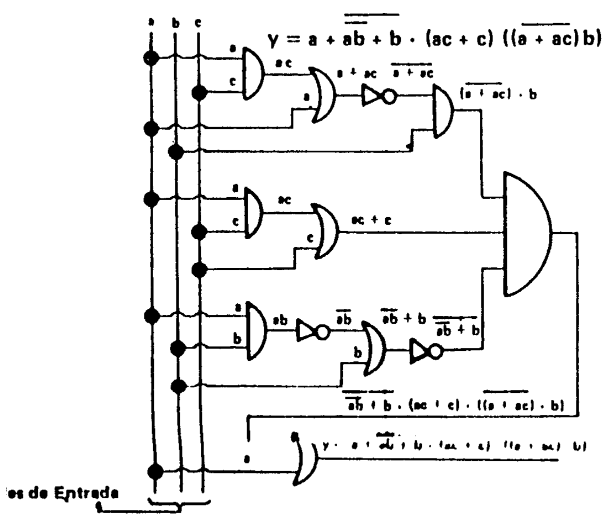
De acuerdo con la expresión literal de la función, la salida de la compuerta OR debe ingresar a la compuerta AND de dos entradas, debiendo ingresar por la segunda entrada la misma variable “a”, pero ahora sin complementar. De acuerdo con lo anteriormente señalado, a la salida de la compuerta AND se obtiene la operatoria correspondiente entre lo que ingresó por la entrada superior (salida de compuerta OR) y lo que ingresó por la entrada inferior (variable “a”). En la figura se muestra además, en cada parte del circuito, la expresión literal que corresponde a la función presente en ese punto.

A través de este ejemplo propuesto queda de manifiesto que para implementar el circuito a través de una expresión literal, es necesario descomponer a esta última en subfunciones que se puedan implementar con los circuitos y compuertas conocidos.

Normalmente la cantidad de compuertas utilizadas en la implementación de un circuito de lógica, así como el número de entradas que posean las mismas, es proporcional a la extensión de la expresión literal que le corresponde, aunque en algunos casos como veremos más adelante, es posible encontrar una expresión literal más reducida a objeto de implementar un circuito también más reducido y que realice la misma función.

La figura siguiente muestra la transformación a circuito de la función cuya expresión literal es:

F = a+‘ab + b . (ac + c) ((a + ac)b)



Nótese que en esta expresión literal existen operaciones lógicas entre paréntesis que a su vez se encuentran entre paréntesis mas externos; en estos casos se comenzará primero por resolver aquellos más externos para continuar hacia fuera con los siguientes. En el diagrama mostrado en la figura se muestra en los distintos puntos del circuito la expresión literal parcial que le corresponde, por lo tanto al avanzar hacia la salida del circuito se obtiene la expresión literal original.

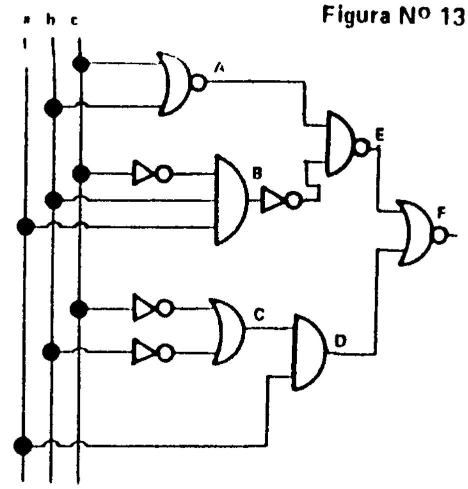
<u>EJERCICIOS:</u> Transformar en circuito las siguientes expresiones literales:

F = a + b + ca
F = (ab + c)ac + b
F = abc + bd +bc
F = a(cd + ab) (((a-b)c + d)a + ad)
F = bc + bd + ab +ac

Así como se puede transformar una expresión literal a un circuito de lógica combinacional, también es posible transformar un circuito a una expresión literal. Para obtener la expresión literal del circuito de lógica combinacional, se coloca en la salida de cada compuerta la operación lógica que le corresponde con las variables que posee como entradas, así, al avanzar hacia la salida del circuito se va construyendo la expresión literal de la función que le corresponde.

En la figura siguiente se muestra como obtener la expresión literal de una función que se encuentra implementada en un circuito. Los puntos indicados entre A hasta E corresponden a subfunciones que actúan como entradas de los circuitos siguientes.

Figura Nº 13



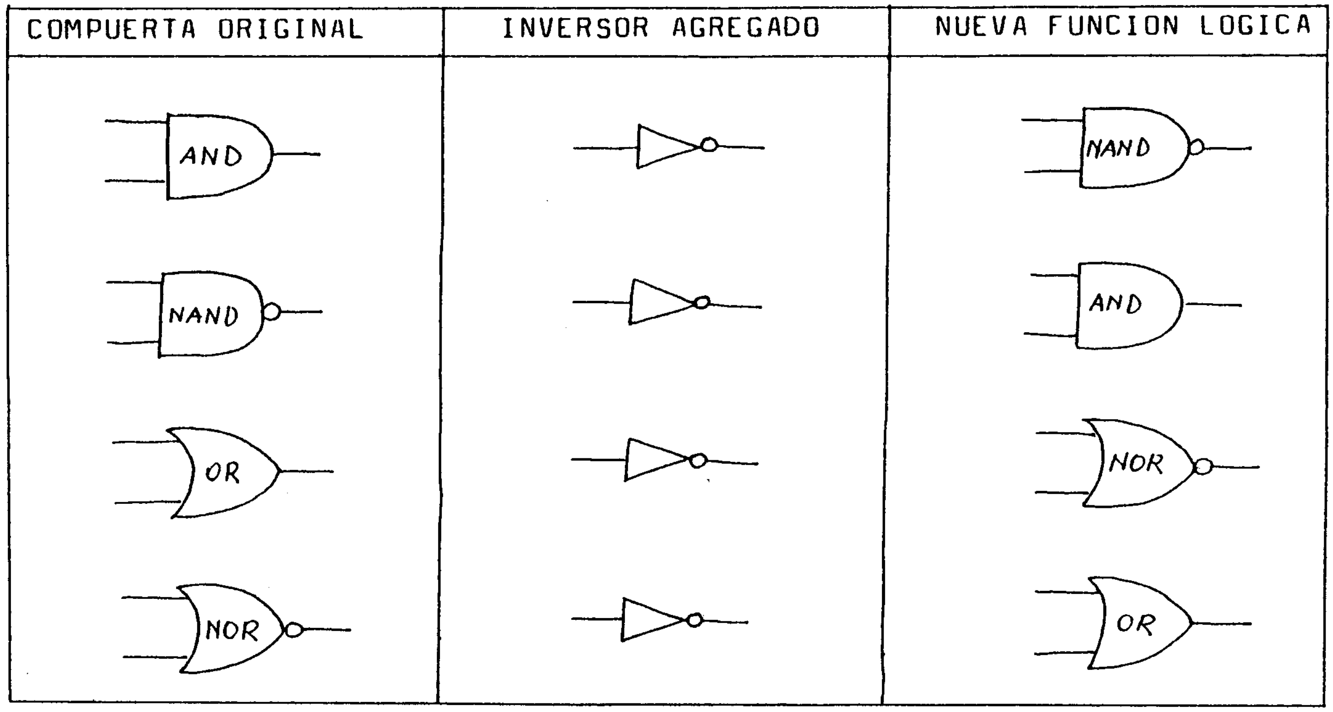
$A = \overline{b+c}$
$B = a.\ b.\ \bar{c}$
$C = \bar{b} + \bar{c}$
$D = a\ .\ C = a\ .\ (\bar{b} + \bar{c})$
$E = A + \bar{B} = \overline{\overline{b+c}\ .a\ .b\bar{c}}$
$F = E + D = \overline{A + B + D}$

$F = \overline{\overline{\overline{b + c}.\ ab\bar{c}} + (\bar{b} + \bar{c}).a}$

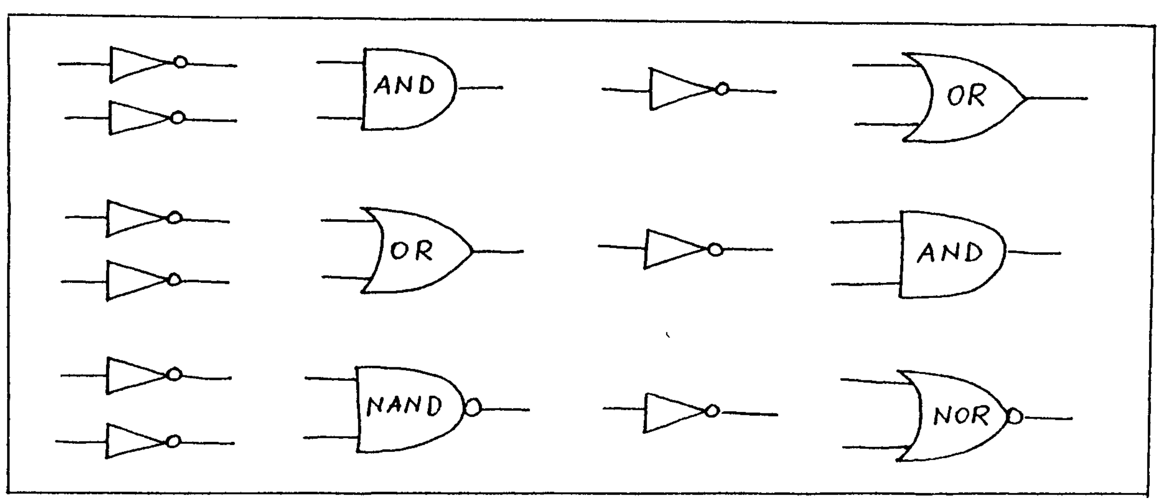
Expresión Resultante

CONVERSIÓN DE COMPUERTAS USANDO INVERSORES

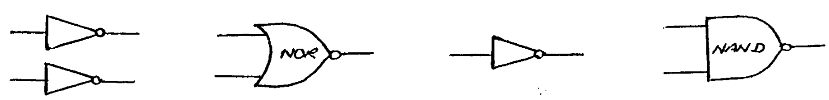
Cuando se usan compuertas lógicas, surge la necesidad de convertir a otras funciones lógicas. Un método de conversión sencillo es el de colocar inversores en las entradas y / o salidas de las compuertas.



INVERSORES EN LA ENTRADA | COMPUERTA ORIGINAL | INVERSOR EN LA SALIDA | NUEVA FUNCION LOGICA

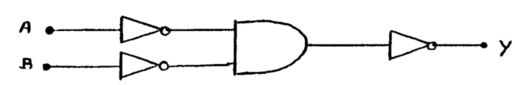


INVERSORES EN LA ENTRADA | COMPUERTA ORIGINAL | INVERSOR EN LA SALIDA | NUEVA FUNCION LOGICA



EJERCICIOS: Desarrolle la tabla de verdad, ecuación de salida y compuerta equivalente de los siguientes circuitos lógicos:

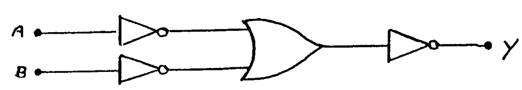
1.-



Circuito equivalente a compuerta..............

Entradas		Salida
B	A	S
0	0	
0	1	
1	0	
1	1	

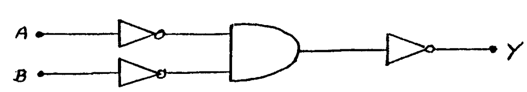
2.-



Circuito equivalente a compuerta..............

Entrada		Salida
B	A	S
0	0	
0	1	
1	0	
1	1	

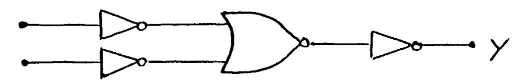
3.-



Circuito equivalente a compuerta..............

Entradas		Salida
B	A	
0	0	
0	1	
1	0	
1	1	

4.-



Circuito equivalente a compuerta..............

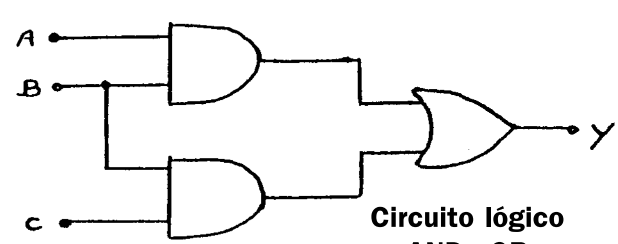
Entradas		Salida
A	B	
0	0	
0	1	
1	0	
1	1	

Señale cual sería la compuerta resultante si en los cuatro ejemplos señalados se eliminará el inversor de salida.

1) Compuerta ....
2) Compuerta..
3) Compuerta..
4) Compuerta..

COMBINACIÓN DE COMPUERTAS LÓGICAS

Muchos problemas cotidianos de lógica digital emplean diversas compuertas lógicas. El patrón más común de compuertas es el AND – OR.



Circuito lógico AND - OR

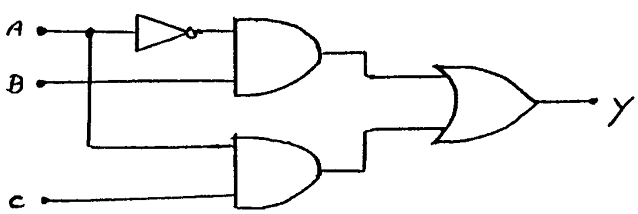
Entradas			**Salida**
C	B	A	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

EXPRESIÓN BOOLEANA

Y = A . B + B . C o bien Y = AB + BC

Los circuitos lógicos AND – OR pueden presentar diversas variantes cuando funcionan en combinación con compuertas inversoras. A continuación se presentan diversas combinaciones a las cuales se deberá desarrolla r su respectiva tabla de verdad, basándose en los ejemplos antes mencionados, agregando además la correspondiente expresión booleana:

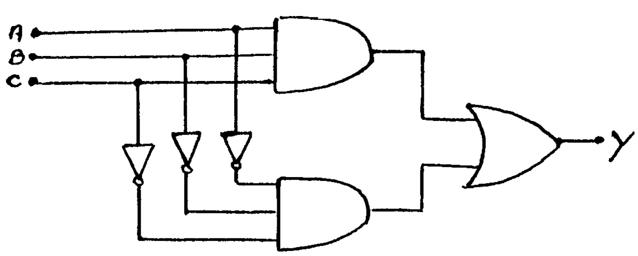
1)



Expresión booleana
Y = ...

Entradas			Salida
C	B	A	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	

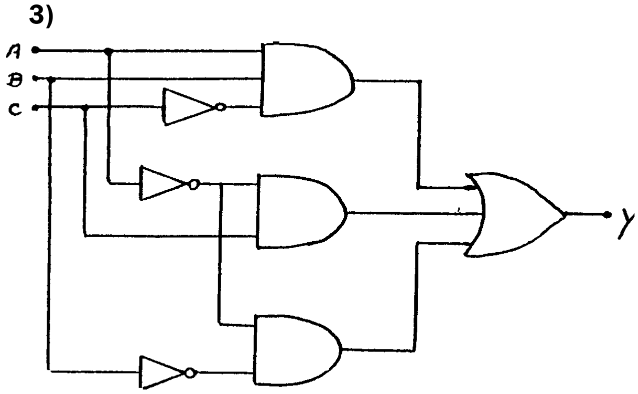
2)



Expresión booleana
Y = ...

Entradas			Salida
C	B	A	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	

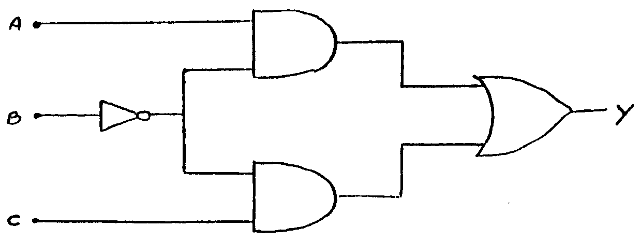
3)



Expresión booleana
Y = ...

Entradas			Salida
C	B	A	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	

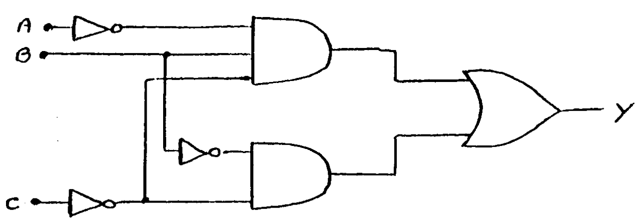
4)



Expresión booleana
Y = ...

Entradas			Salida
C	B	A	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	

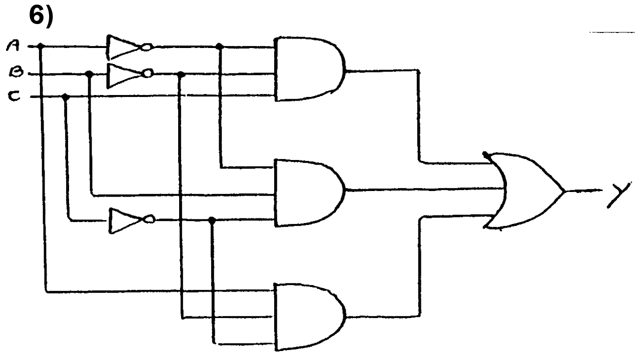
5)



Expresión booleana
Y = ...

Entradas			Salida
C	B	A	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	

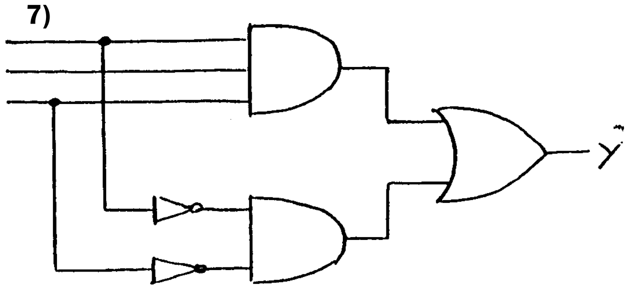
6)



Expresión booleana
Y = ...

Entradas			Salida
A	B	C	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	

7)

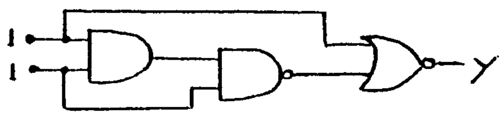


Expresión booleana
Y = ...

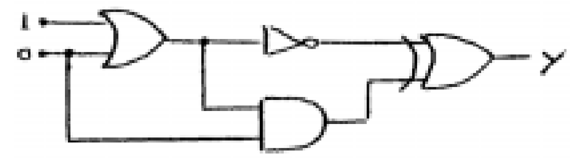
Entradas			Salida
B	C	A	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Señale el nivel de salida de los siguientes circuitos lógicos (0 ó 1)

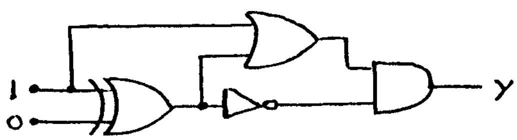
1)



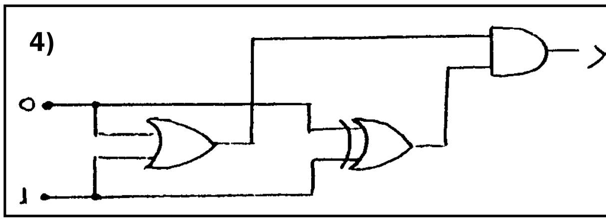
2)



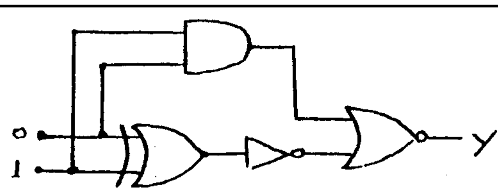
3)



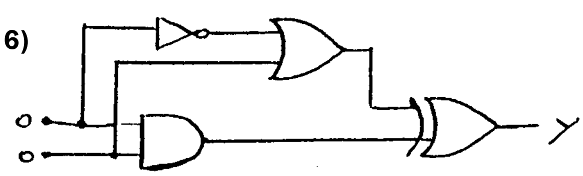
4)



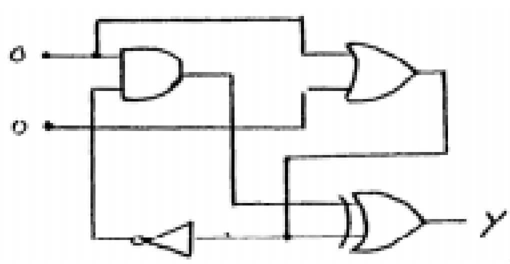
5)



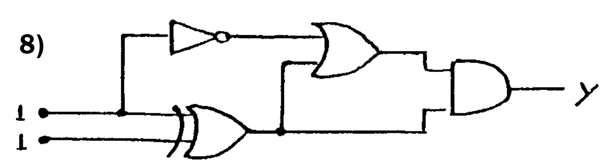
6)



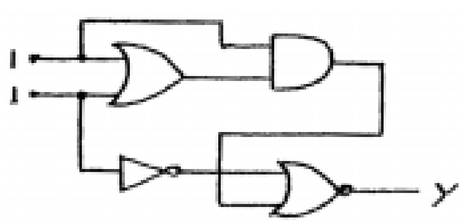
7)



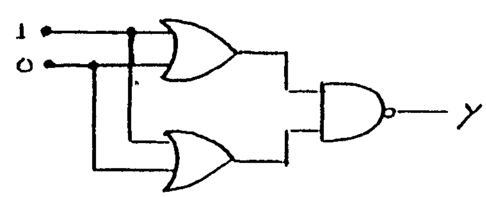
8)



9)



10)



<u>Salida de los circuitos lógicos:</u>

1) Y = ...
2) Y = ...
3) Y = ...
4) Y = ...
5) Y = ...

6) Y = ...
7) Y = ...
8) Y = ...
9) Y = ...
10) Y = ...

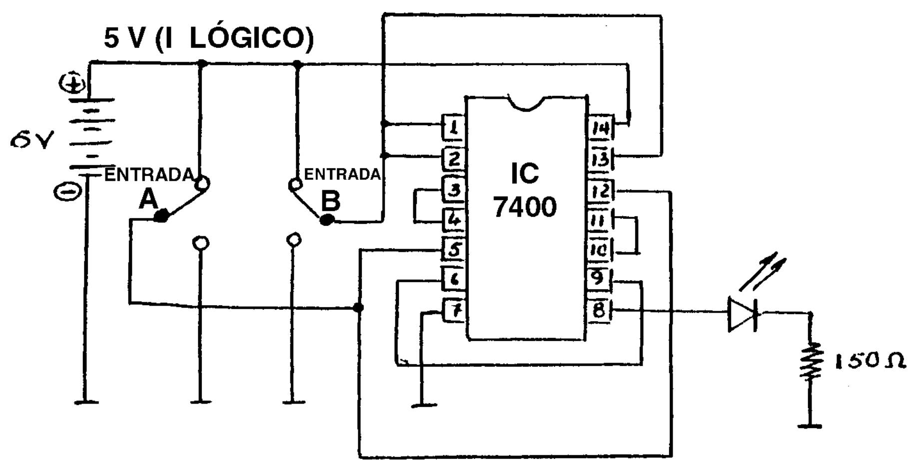
EJERCICIO: Sobre el circuito representado a continuación (IC – 7400), realice el siguiente trabajo:

a) Dibuje el conexionado de las compuertas NAND, partiendo de las entradas A y B hasta finalizar el diodo led.

b) Dibuje la tabla de verdad del circuito.

c) Señale si el diodo quedará encendido o apagado cuando las entradas A y B se encuentren en nivel 1.

IC – 7400 IC – 7410



ELECTRÓNICA DIGITAL
PRACTICAS

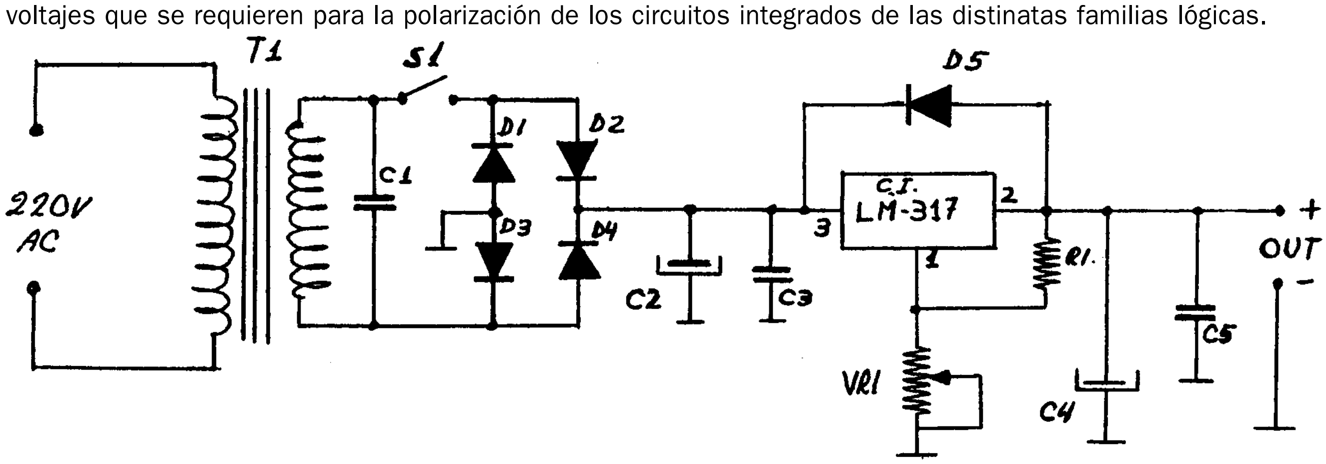
CONSIDERACIONES INICIALES.

Para lograr un adecuado trabajo práctico en circuitos digitales, como por ejemplo el armado de los mismos, características de los circuitos integrados utilizados, alimentación del circuito y comprobación del funcionamiento de acuerdo a tablas de verdad, se hacen necesarios los siguientes materiales:

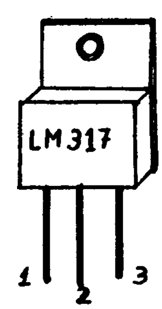
1) Fuente de poder regulada de 0 a 37Vcc.
2) Protoboard.
3) Punta de prueba lógica.
4) Tester digital.
5) Manual de características.

1. FUENTE DE PODER REGULADORA DE 1.2 A 37VCC.

A través de una fuente regulada, como la que se muestra a continuación, se pueden obtener los distintos voltajes que se requieren para la polarización de los circuitos integrados de las distinatas familias lógicas.



Materiales necesarios para el armado.



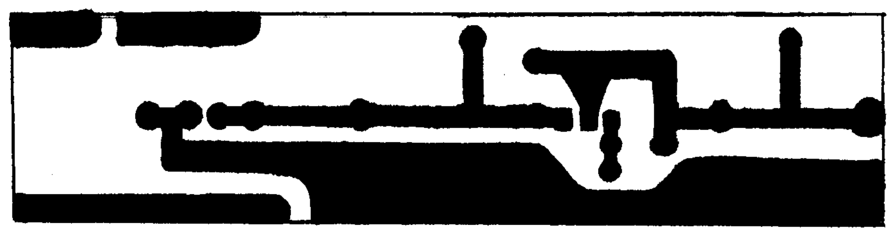
T1 = Transformador de 30v / 2 A.
D1/D5 = 4 Diodos 1N 4007.
C1/C3/C5 = 3 condensadores de 0.33 UF x 400V.
C2 = Condensador de 3.300 UF x 63V.
C4 = Condensador de 2.2 UF x 50V.
VR1 = Potenciómetro de 10K con interruptor (eje corto y plástico).
R1 = Resistencia de 270 / 1/2W.
IC = Circuito integrado LM317.

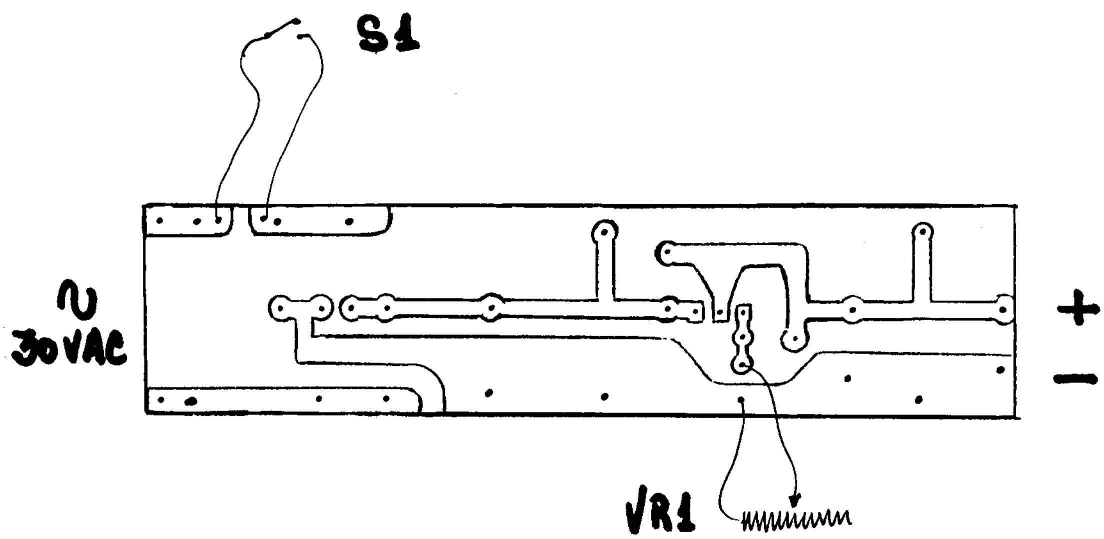
Es importante destacar que a este diseño se le pueden agregar los accesorios que el interesado estime convenientes (Voltímetro, amperímetro, diodo leds, etc.).

Identificación de los terminales del CI LM317.

1 = Adj. (Potenc.)
2 = Out.
3 = In. (40 Vcc max.)

Circuito Impreso en negativo y positivo





Precauciones en el uso.

Antes de encender la fuente, verificar que los terminales de salida de la misma no se encuentran cortocircuitados, puesto que de ser así, se quema la fuente.

2. PROTOBOARD.

Las propiedades y el uso del Protoboard ya fueron explicadas en videos anteriores.

3. PUNTA DE PRUEBA LÓGICA.

Hoy en día es muy común que en los aparatos electrónicos encontremos circuitos integrados digitales. En las mediciones que realicemos en ellos solo nos interesa conocer los niveles lógicos presentes en las salidas; es decir, saber si estamos frente a la presencia de un nivel lógico alto (1), bajo (0) o frente a un tren de pulsos.

Para poder efectuar este tipo de mediciones, lo más lógico y útil es emplear una punta de prueba lógica. Esta punta de prueba lógica es muy sencilla de construir y de bajo costo.

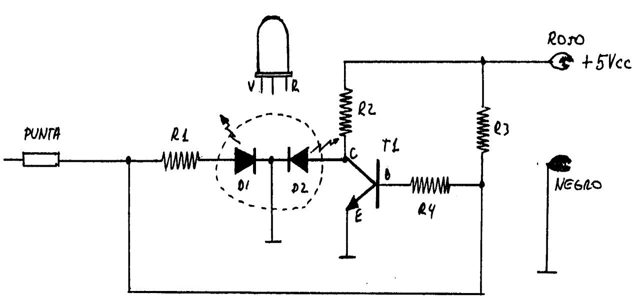
En nuestra práctica nos será de mucha utilidad y su metodología de operación se explica de la siguiente manera:

a) Cuando la punta está en el aire o conectada a un punto muerto del circuito bajo prueba, D1 estará apagado y D2 también estará apagado.
b) Si la punta de prueba está en alto, D1 estará encendido y D2 continuará apagado.
c) Si la punta de prueba está en bajo, D1 estará apagado y D2 encendido.
d) Si la punta está presente en tren de pulsos de baja frecuencia, D1 y D2 estarán encendidos en forma intermitente.
e) Si se presenta en la punta de prueba un tren de pulsos de alta frecuencia, D1 y D2 estarán encendidos, adquiriendo un color anaranjado.

Este circuito puede ser montado dentro de un lápiz en deshuso, otorgándole así mayor movilidad.
Con lo anteriormente expuesto podemos resumir la operación del circuito de la siguiente manera:

- Led verde encendido .... "0"
- Led rojo encendido .... "1"
- Leds apagados .... "No conectada"
- Leds intermitentes .... "Tren de pulsos de baja frecuencia"
- Leds anaranjado .... "Tren de pulsos de altas frecuencia"

Circuito eléctrico de la punta de prueba.



4. TESTER DIGITAL.

Permite verificar que el voltaje de alimentación que polariza a los CI es acorde a su familia lógica. Se caracteriza en la gran precisión que se obtiene de la medición.

5. MANUAL DE CARACTERÍSTICAS.

Permite averiguar las principales características del circuito integrado y su reemplazo en el caso de avería.

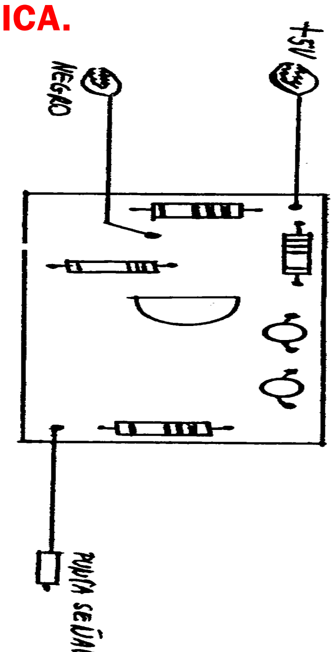
ARMADO DE LA PUNTA DE PRUEBA LÓGICA.

a) De acuerdo al circuito eléctrico, confeccionar el pictograma.
b) Transferir el pictograma a una placa de circuito impreso.
c) Montar los elementos de acuerdo al circuito.

LISTA DE MATERIALES.

R1 = Resistencia de 220 ohms-1/4 W.
R2 = resistencia de 220 ohms-1/4 W.
R3 = Resistencia de 1.2K. ohms 1/4 W
R4 = Resistencia de 100 ohms 1/4 W.
T1 = Transistor BC548.
D1/D2 = Leds de dos colores (V/R), con cátodo común.

PICTOGRAMA DE LA PUNTA DE PRUEBA LÓGICA.



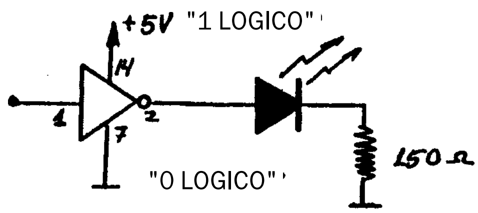
COMPROBACIÓN DE COMPUERTAS LÓGICAS EN INTEGRADOS TTL.

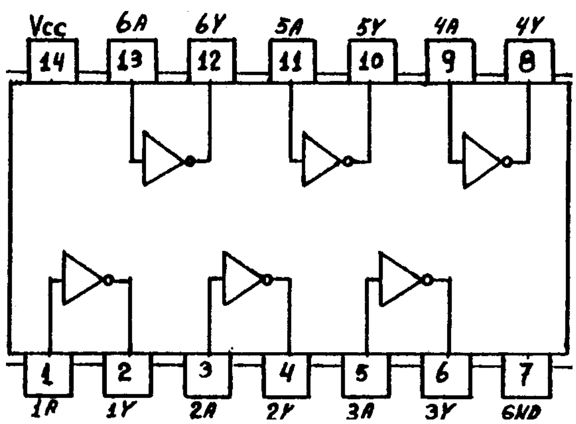
1. Verificar tabla de la verdad del CI 7404, que actúa como sextuple compuerta inversora (Hex inverter).

Tabla de verdad

In	Out
A	S
1	
0	

IC 7404





Para efectuar esta prueba, realice el siguiente conexionado:

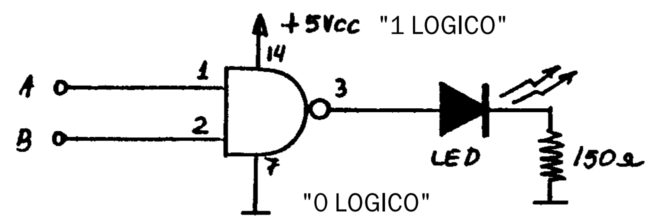
a) Por tratarse de un integrado de la familia TTL, regule la fuente a 5Vcc.
b) Alimente el pin 14 del integrado con +5Vcc y el pin 7 con el negativo de la fuente.
c) El positivo de la fuente será considerado como "1 lógico" y el negativo como "0 lógico".
d) Conecte la entrada de la primera compuerta a "1 lógico" y a través de la punta de prueba lógica verifique el novel de salida. Anote este nivel en la tabla de verdad.
e) Conecte la entrada de la primera compuerta a "0 lógico" y a través de la punta de prueba lógica verifique el nivel de la salida. Anote este nivel en la tabla de verdad.
f) Repita estas pruebas con cada una de las restantes compuertas inversoras.

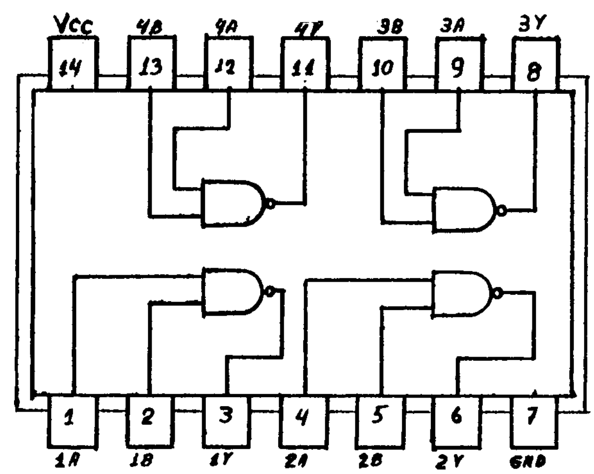
2. Verificar tabla de verdad del CI 7400, que actúa como cuádruple compuerta NAND de dos entradas cada una.

Tabla de verdad.

In		Out
B	A	S
0	0	
0	1	
1	0	
1	1	

IC 7400





a) Inserte el CI 7400 entre dos columnas anchas del protoboard.
b) Regule la fuente a 5Vcc.
c) Alimente el pin 14 del integrado con +5Vcc y el pin 7 con el negativo de la fuente.
d) Conecte la entrada A de la primera compuerta a "0 lógico" y la entrada B a "0 lógico". Luego, a través de la punta de prueba lógica verifique el nivel de salida y anótelo en la tabla de verdad.
e) Conecte la entrada A de la primera compuerta a "1 lógico" y la entrada B a "0 lógico". Luego, a través

de la punta de prueba lógica verifique el nivel de salida y anótelo en la tabla de verdad.

f) Conecte la entrada A de la primera compuerta a "0 lógico" y la entrada B a "1 lógico". Luego, verifique el nivel de salida y anótelo en la tabla de verdad.

g) Conecte las entradas (A y B) a "1 lógico" y luego, a través de la punta de prueba lógica verifique el nivel de salida. Anótelo en la tabla de verdad.

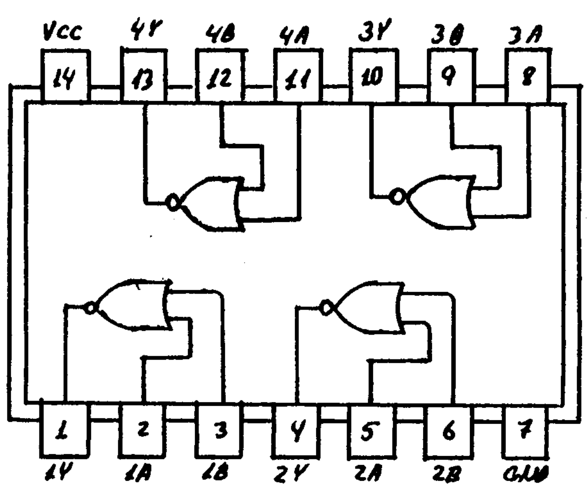
h) Repita estas pruebas en cada una de las restantes compuertas que conforman el integrado 7400.

3. Verificar tabla de la verdad del CI 7402, que actúa como cuádruple compuerta NOR de dos entradas cada una.

Tabla de verdad

In		Out
B	A	S
0	0	
0	1	
1	0	
1	1	

IC 7402



a) Inserte el CI 7402 entre dos columnas anchas del protoboard.

b) Regula la fuente 5Vcc.

c) Alimente el pin 14 del integrado con +5Vcc y el pin 7 con el negativo de la fuente.

d) Conecte la entrada A de la primera compuerta a "0 lógico" y lo mismo haga con la entrada B. Verifique posteriormente el nivel de salida y anótelo en la tabla de verdad.

e) Conecte la entrada A de la primera compuerta a "1 lógico" y la entrada B a "0 lógico". Compruebe el nivel de salida y anótelo en la tabla de verdad.

f) Conecte la entrada A de la primera compuerta "0 lógico y la entrada B "1 lógico". Averigüe el nivel de la salida y remítalo a la tabla de verdad.

g) Conecte ambas entradas a "1 lógico" y, posteriormente, verifique el nivel de la salida. Anótelo en la tabla de verdad.

Conclusiones: ....
....
....
....
....

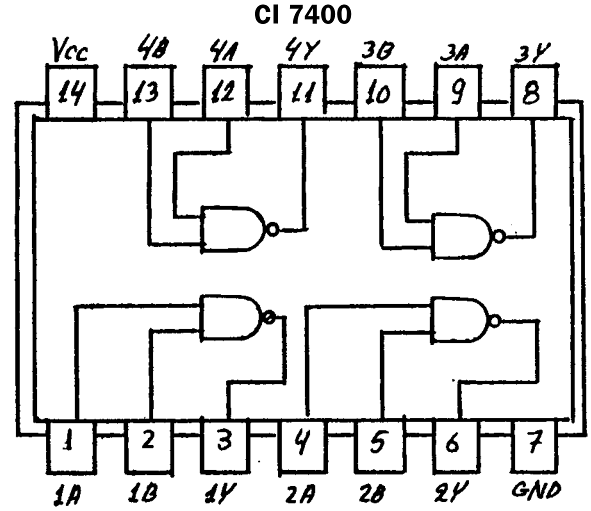
PRÁCTICA DE LÓGICA COMBINACIONAL

1) Utilice un integrado 7400 y un integrado 7404, para conformar una compuerta AND de dos entradas.

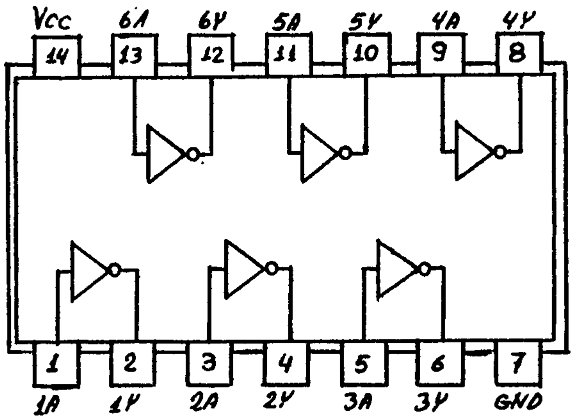
Tabla de verdad

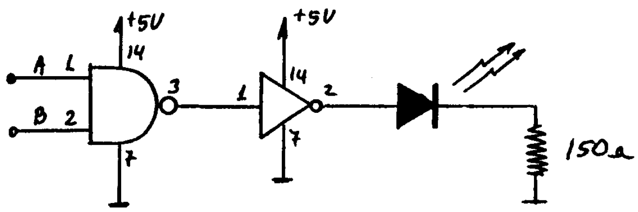
B	A	$\overline{A.B}$	A.B
0	0		
0	1		
1	0		
1	1		

CI 7400



CI 7404





a) Inserte el CI7400 y el CI 7404 entre dos columnas anchas del protoboard.

b) Conecte la salida de la primera compuerta NAND (pin 3) a la entrada de la primera compuerta inversora (pin1).Ver esquema de conexiones.

c) Regule la fuente a 5Vcc.

d) Alimente el pin 14 de ambos integrados con +5Vcc y el pin 7 de ambos integrados al negativo de la fuente.

e) Conecte la entrada A de la primera compuerta NAND (pin 1 del CI 7400) a "0 lógico" y la entrada B del mismo integrado (pin 2), también a "0 lógico". Compruebe el nivel de salida de la compuerta NAND (pin 3 del CI 7400) y anótelo en la tabla de verdad. Compruebe luego el nivel de salida a la salida de la compuerta inversora (pin 2 del CI 7404) y anótelo en la tabla de verdad.

f) Conecte la entrada A de la primera compuerta NAND (pin 1 del CI 7400) a "1 lógico" y la entrada B del mismo integrado (pin 2) a "0 lógico".

Compruebe luego el nivel de la salida de la compuerta NAND y señálelo en la tabla de la verdad. Verifique posteriormente el nivel de salida de la compuerta inversora (pin 2 del CI 7404) y márquelo en la tabla de verdad.

g) Conecte la entrada A de la primera compuerta NAND a "0 lógico" y la entrada B del mismo integrado a "1 lógico". Verifique luego el nivel de salida de la primera compuerta NAND (pin 3) y destáquelo en la tabla de verdad. Posteriormente señale el nivel de salida de la primera compuerta inversora (pin 2) y destáquelo en la tabla de verdad.

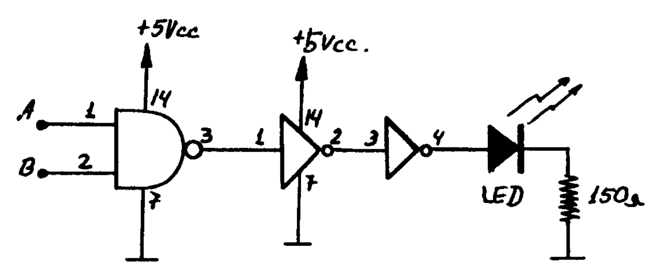
h) Conecte ambas entradas de la primera compuerta NAND a "1 lógico" y verifique el nivel de salida de la misma (pin 3). Verifique luego, el nivel de salida de la primera compuerta inversora y destáquelo en la tabla de la verdad.

Conclusión: ....
....
....
....

2) Realice una tabla de verdad en la cual se señale el nivel de salida resultante, si al mismo circuito anterior se le agrega una nueva compuerta inversora a la salida.

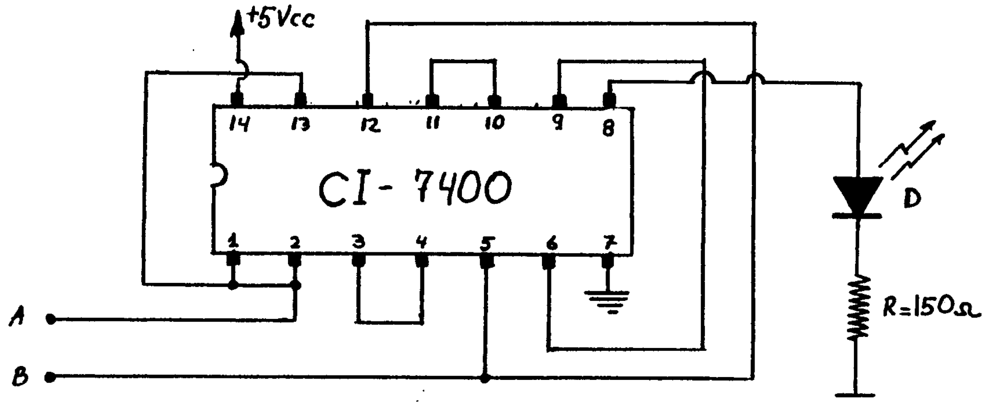
Tabla de verdad

B	A	$\overline{A.B}$	$\overline{A.B}$	$\overline{\overline{A.B}}$
0	0			
0	1			
1	0			
1	1			



Conclusión: El circuito señalado es equivalente a una compuerta:..
....
....
....

3) Realice el siguiente montaje utilizando un CI 7400, un diodo Led y una resistencia de 150 ohms.



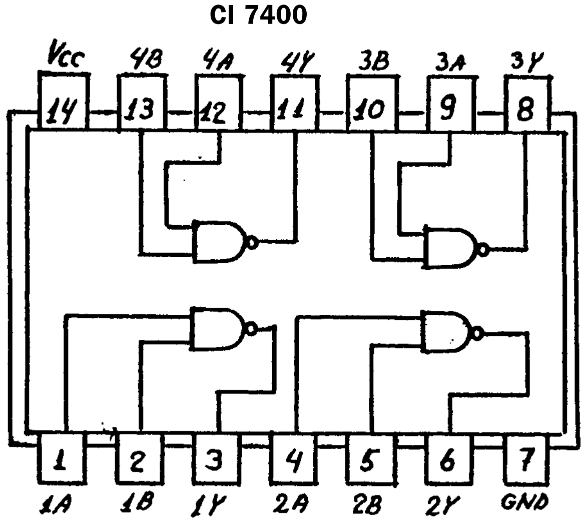
a) Inserte el CI 7400 entre dos columnas anchas del protoboard.
b) Efectúe el conexionado según lo indica el circuito.
c) Conecte el diodo Led y el resistor de 150 ohms.
d) Asegúrese de que el conexionado se encuentra correcto.
e) Desarrolle la tabla de la verdad del circuito.

B	A	$\overline{A}$	A.B	$\overline{A}.B$	$\overline{\overline{A}.B}$	$\overline{A.B}$	$\overline{(\overline{\overline{A}.B})\ (\overline{A.B})}$
0	0						
0	1						
1	0						
1	1						

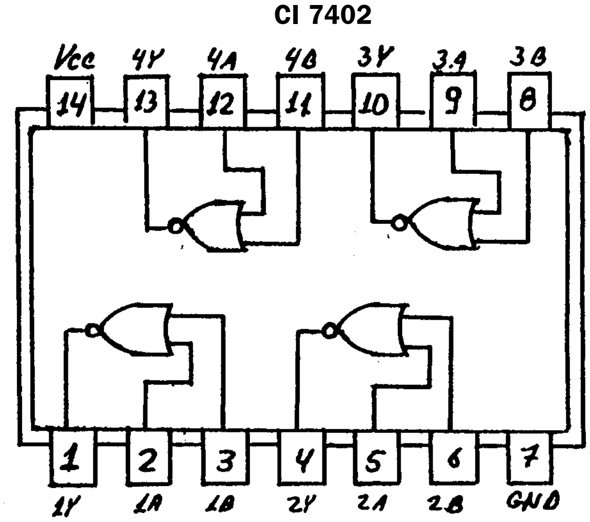
Comprobación del mismo a través de la punta lógica.
Verifique cada una de las salidas indicadas en la tabla de la verdad esquematizada más arriba y compárelas con dicha tabla.

4) Utilice un CI 7400 y 7402, para realizar el siguiente montaje.

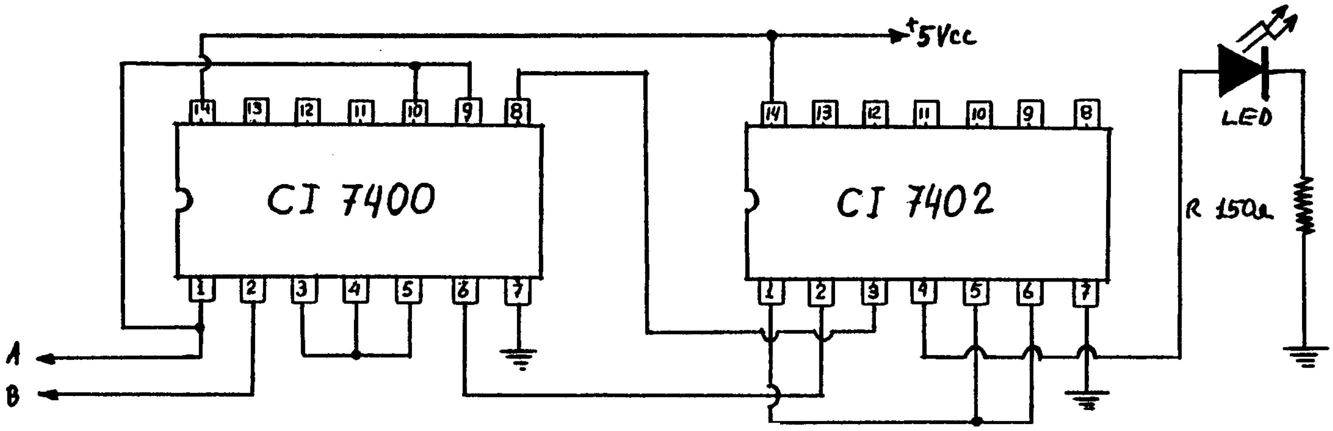
CI 7400



CI 7402



Circuito a realizar:



Circuito desarrollado:

Tabla de verdad:

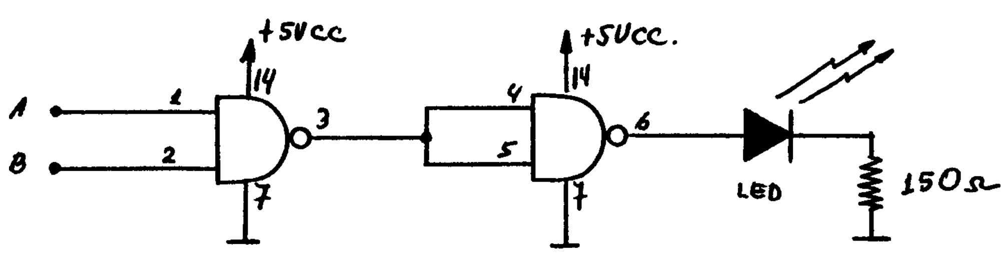
B	A	$\bar{A}$	A.B	$\overline{A.B}$	$\overline{\overline{A.B}}$	$\overline{\overline{A.B}}+\bar{A}$	$\overline{\overline{\overline{A.B}}+\bar{A}}$	$\overline{\overline{\overline{\overline{A.B}}+\bar{A}}}$
0	0							
0	1							
1	0							
1	1							

Montaje del circuito:

a) Inserte los circuitos integrados entre dos columnas anchas del protoboard, a corta distancia entre sí.
b) Efectúe el conexionado que señala en circuito superior.
c) Agregue el diodo Led y el resistor de 150 ohms.
d) Asegúrese de que el conexionado es el correcto.
e) Verifique cada una de las salidas indicadas en la tabla de la verdad mostrada más arriba y compárelas con dicha tabla.

5) Utilizando un CI 7400, de la familia TTL, realice los siguientes colexionados:

a)

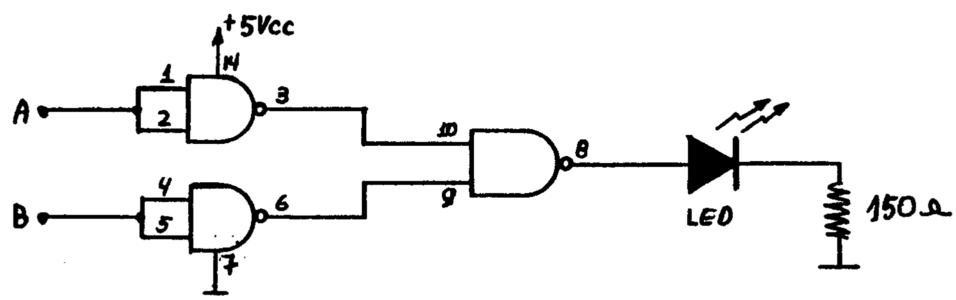


- Desarrolle la tabla de verdad del circuito:

- De acuerdo a la tabla de verdad el circuito final es equivalente a una compuerta.................... de...................entradas.
- Dibuje la compuerta equivalente:

- A través de una punta de prueba lógica verifique si los niveles de salida del circuito coinciden con la tabla de verdad.

b)



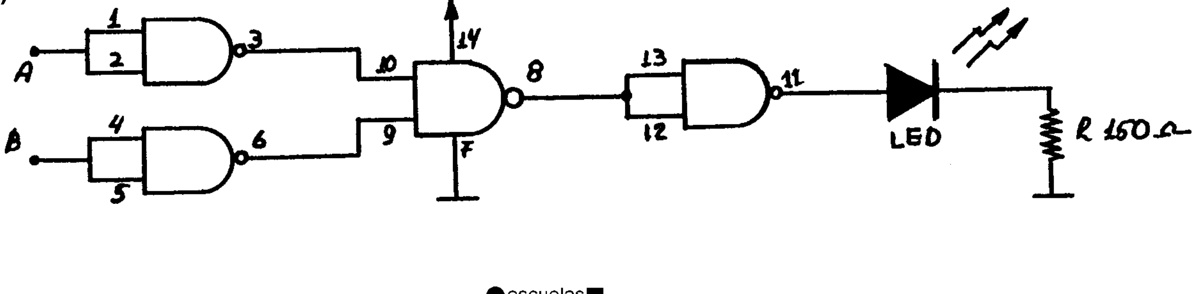
- Desarrolle la tabla de la verdad del circuito:

- De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta......................de........................entradas.

- Dibuje la compuerta equivalente:

- Utilizando la puerta de prueba lógica compruebe si los niveles de salida del circuito concuerdan con la tabla de la vardad.

c)



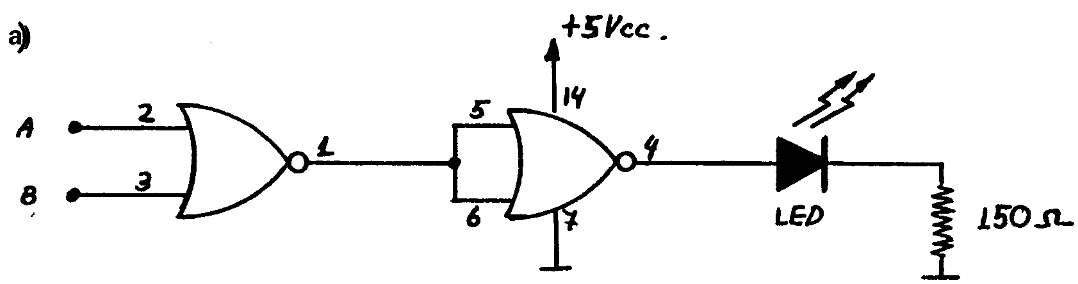
- Desarrolle la tabla de la verdad del circuito:

- En conformidad a la tabla de la verdad el circuito representado es equivalente a la compuerta............de................entradas.
- Dibuje la compuerta equivalente:

- A través de la punta de prueba lógica verifique si los niveles de salida del circuito concuerdan con la tabla de verdad.

6) Utilizando un CI 7402, de la familia TTL, realice los siguientes conexionados:

a)



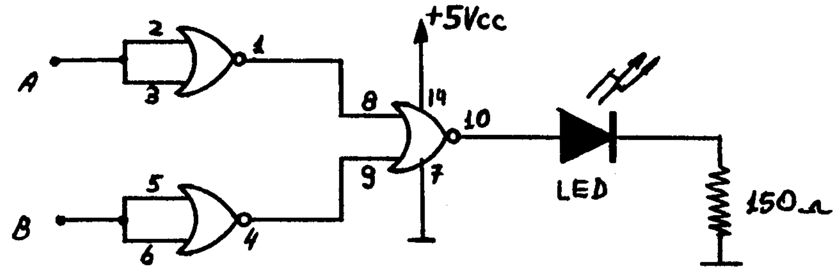
- Desarrolle la tabla de la verdad del circuito:

- De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta.....................de.................entradas.

- Dibuje la compuerta equivalente:

- A través de la punta de prueba lógica certifique si los niveles de salida del circuito coinicen con la tabla de verdad.

b)



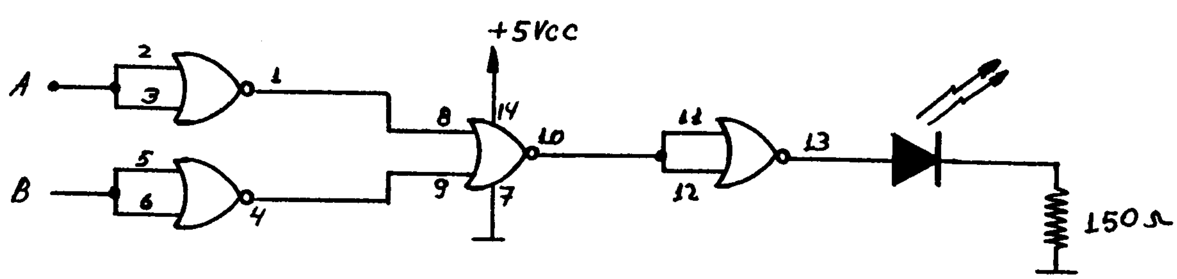
- Desarrolle la tabla del circuito:

- De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta..................de............entradas.

- Dibuje la compuerta equivalente:

- Por medio de la punta de prueba lógica determine si los niveles de salida del circuito son coincidentes con la tabla de verdad.

c)

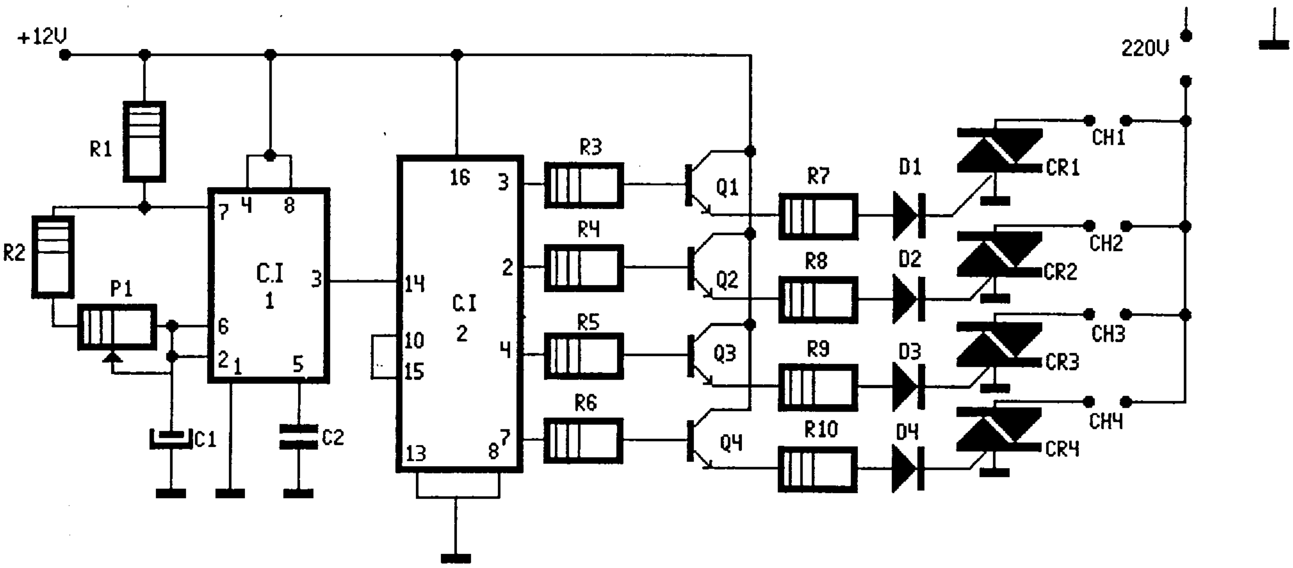


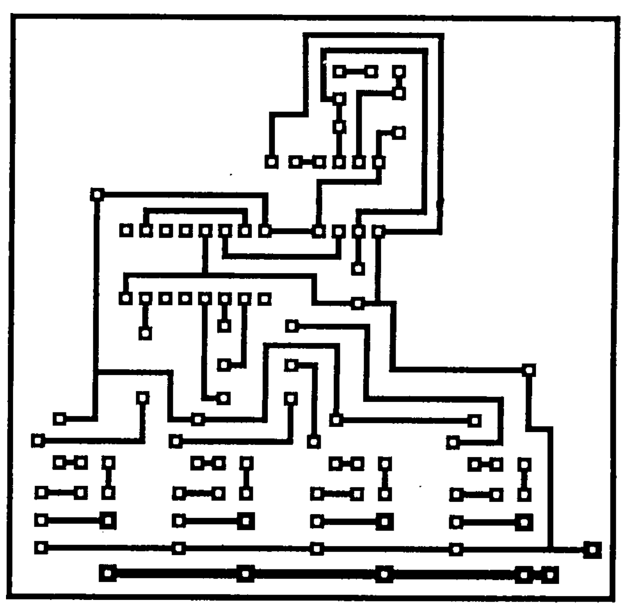
- Desarrolle la tabla de la verdad del circuito:

- De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta...................de...........entradas.

- Dibuje la compuerta equivalente:

- A través de la punta de prueba lógica determine si los niveles de salida del circuito coinciden con la tabla de la verdad.





LISTA DE MATERIALES

CI 1 - CA 555
CI 2 - CD 4017
CR1 a CR4 - TIC226D con disipador
R1 - 1k2Ω
R2 - 1kΩ
P1 - 100kΩ
R3 a R6 1kΩ
R7 a R10 - 2k2Ω
D1 a D4 - 1N4004
Q1 a Q4 - BC548
C1 - 1µF x 12V
C2 - .01nF