

# CURSO: ELECTRÓNICA DIGITAL

## UNIDAD 1: COMPUERTAS LÓGICAS - TEORÍA

### PROFESOR: JORGE ANTONIO POLANÍA

## 1. NÚMEROS BINARIOS

Los números binarios son la base para el estudio de la Electrónica Digital. Así como los números decimales tienen como base el número 10, pues,

$$345 = 5 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 = 5 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2$$

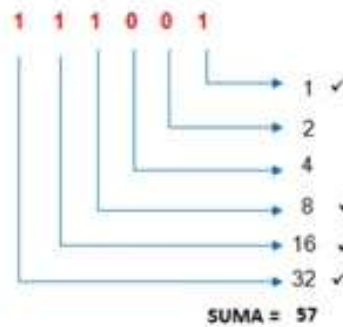
0, 1, 2, 3, ..., 10

Los números binarios tienen como base el número 2 y sólo tienen los números 0 y 1. Por eso se llaman binarios.

El número binario 111001 tiene su representación decimal de la siguiente forma:

$$111001_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5$$

$$= 1 + 0 + 0 + 8 + 16 + 32 = 57_{10}$$



	8	4	2	1	
HEX	DEC	A	B	C	D
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
A	10	1	0	1	0
B	11	1	0	1	1
C	12	1	1	0	0
D	13	1	1	0	1
E	14	1	1	1	0
F	15	1	1	1	1

## EJEMPLO

En el cuadro anterior, está la representación de los números binarios en formato Hexadecimal que son del 0 al 9 y A=10, B=11, C=12, D=13, E=14 y F=15.

El número decimal 57, en formato binario es igual a:

$$57=32+16+8+1=1\ 1\ 1\ 0\ 0\ 1$$

Note que la ponderación binaria de derecha a izquierda es: 1,2,4,8,16,32,64,...(potencias de 2)

En formato hexadecimal:

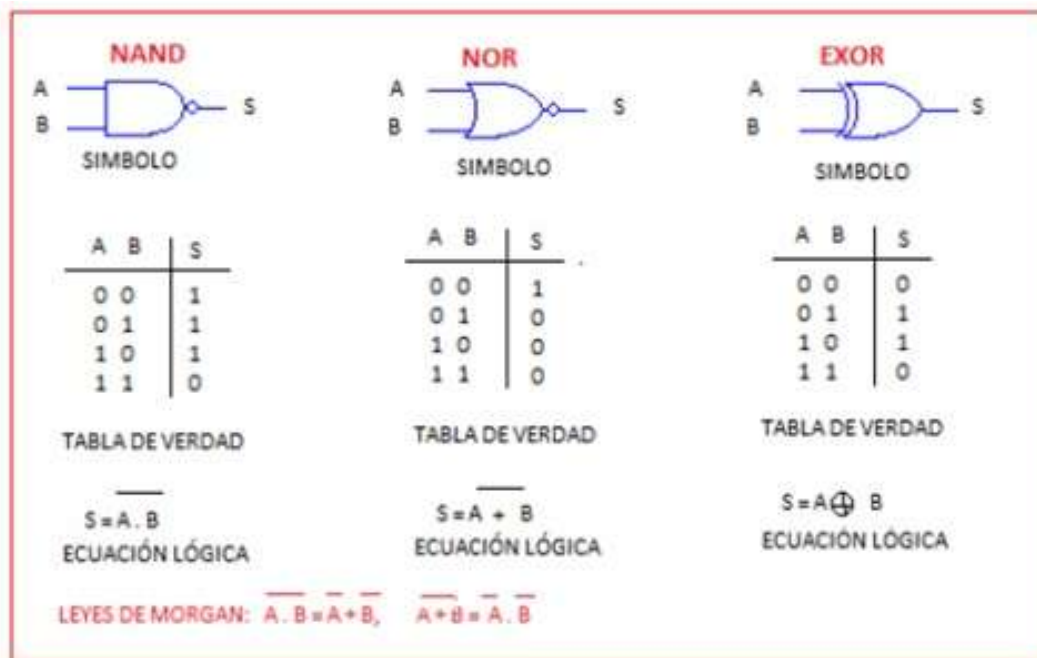
$$57=0101(5)\ 0111(7)=\ 01010111$$

## 2. PUERTAS LÓGICAS BÁSICAS

Las puertas o compuertas lógicas básicas son: La puerta AND, la puerta OR y la puerta NOT. A continuación se presenta su símbolo, la tabla de verdad que nos dice la salida dependiendo de la combinación de las entradas y su ecuación lógica. Observe que la puerta AND (Puerta Y) solo tiene una salida =1 o nivel alto si fuera un voltaje si ambas entradas son 1. La puerta OR (puerta O) tiene una salida =1 si cualquiera o ambas de las entradas es 1. La puerta NOT o Inversor niega la entrada, esto es, si la entrada es 0 la salida es 1 y si es 1 la salida es 0.



Si a las compuertas anteriores se les niega la salida tenemos la configuración de otras compuertas como la NAND, la NOR. En el cuadro aparece su tabla de verdad con su ecuación lógica. La puerta EXOR es una puerta que se denomina OR EXCLUSIVO y su salida es 1 siempre que las entradas sean diferentes, o sea, entradas, 01 y 10 la salida es 1. Se puede decir que es un comparador. Si las entradas son diferentes la salida es 1 y si son iguales la salida es 0.



### 3. CIRCUITOS INTEGRADOS

Las compuertas son fabricadas en circuitos integrados con base en transistores bipolares que se conoce con tecnología TTL (Transistor Transistor Logic), en donde en su interior pueden estar varias compuertas. Por ejemplo, el CI (circuito integrado) 7408 tiene en su interior 4 compuertas AND de 2 entradas, el 7411 tiene 3 puertas AND de 3 entradas, el 7421 tiene 2 puertas AND de 4 entradas, el 7404

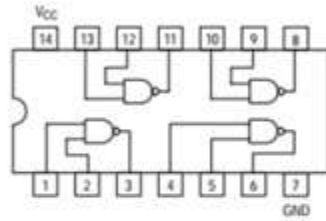
tiene 6 puertas NOT obviamente de 1 entrada, y así como se muestra en la siguiente tabla.

PUERTA	DESCRIPCIÓN	TTL
AND	4x2 ENTRADAS	7408
	3x3 ENTRADAS	7411
	2x4 ENTRADAS	7421
OR	4x2 ENTRADAS	7432
NAND	4x2 ENTRADAS	7400
	3x3 ENTRADAS	7410
	2x4 ENTRADAS	7420
NOT	6x1 ENTRADA	7404
NOR	4x2 ENTRADAS	7402
	3x3 ENTRADAS	7427
	2x4 ENTRADAS	7425
EX OR	4x2 ENTRADAS	7486

En el siguiente cuadro se presenta la distribución de las compuertas NAND en un circuito integrado. Observe que tiene 14 patas o pines, en donde la tierra (GND) es el pin 7 y la fuente (VCC) debe conectarse en el pin 14. La primera compuerta tiene como entradas los pines 1 y 2 y como salida el pin 3. Se puede apreciar la conexión de las demás compuertas en el integrado.

# 7400

## Quad 2-Input NAND Gate



LOW  
POWER  
SCHOTTKY

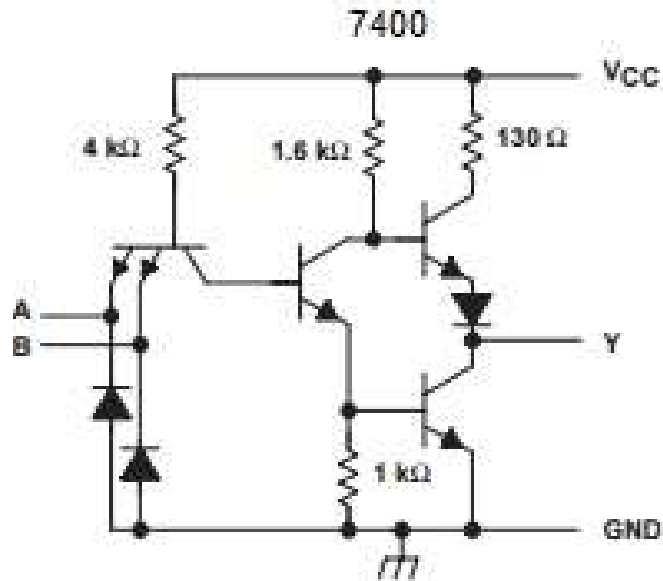


### GUARANTEED OPERATING RANGES

Symbol	Parameter	Min	Typ	Max	Unit
$V_{CC}$	Supply Voltage	4.75	5.0	5.25	V
$T_A$	Operating Ambient Temperature Range	0	25	70	$^{\circ}\text{C}$
$I_{OH}$	Output Current - High			-0.4	mA
$I_{OL}$	Output Current - Low			8.0	mA

<http://www.alldatasheet.com/datasheet-pdf/pdf/12608/ONSEMI/7400.html>

## ESQUEMA ELÉCTRICO DE LA COMPUERTA 7400



## CARACTERÍSTICAS ELÉCTRICAS

Bien importante es conocer las características eléctricas de los circuitos integrados. Me refiero a sus niveles de voltaje y corrientes. En el cuadro se observa,

El nivel alto de voltaje de entrada ( $V_{IH}$ ) es mínimo de 2V

Nivel bajo de voltaje de entrada ( $V_{IL}$ ) es máximo 0.8V

El nivel alto de voltaje de salida ( $V_{OH}$ ) es mínimo de 2.7V

Nivel bajo de voltaje de salida ( $V_{OL}$ ) es máximo 0.4V

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$V_{IH}$	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
$V_{IL}$	Input LOW Voltage			0.8	V	Guaranteed Input LOW Voltage for All Inputs
$V_{IK}$	Input Clamp Diode Voltage		-0.65	-1.5	V	$V_{CC} = \text{MIN}$ , $I_{BI} = -18 \text{ mA}$
$V_{OH}$	Output HIGH Voltage	2.7	3.5		V	$V_{CC} = \text{MIN}$ , $I_{OH} = \text{MAX}$ , $V_{BI} = V_{BI}$ or $V_{IL}$ per Truth Table
$V_{OL}$	Output LOW Voltage		0.25	0.4	V	$I_{OL} = 4.0 \text{ mA}$
			0.35	0.5	V	$I_{OL} = 8.0 \text{ mA}$
$I_{IH}$	Input HIGH Current			20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{BI} = 2.7 \text{ V}$
				0.1	mA	$V_{CC} = \text{MAX}$ , $V_{BI} = 7.0 \text{ V}$
$I_{IL}$	Input LOW Current			-0.4	mA	$V_{CC} = \text{MAX}$ , $V_{BI} = 0.4 \text{ V}$
$I_{OS}$	Short Circuit Current (Note 1)	-20		-100	mA	$V_{CC} = \text{MAX}$
$I_{CC}$	Power Supply Current Total, Output HIGH			1.6	mA	$V_{CC} = \text{MAX}$
	Total, Output LOW			4.4		

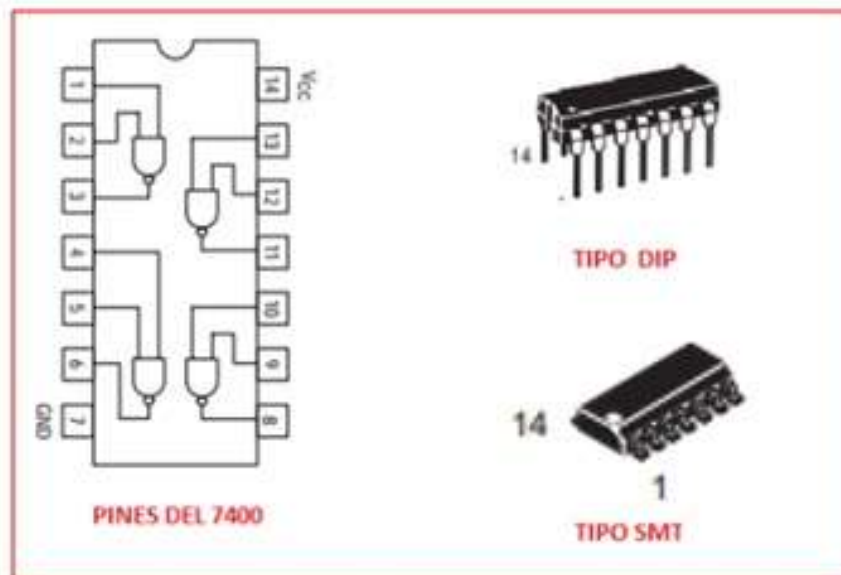
Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

### AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ )

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$t_{PLH}$	Turn-Off Delay, Input to Output		9.0	15	ns	$V_{CC} = 5.0 \text{ V}$ $C_L = 15 \text{ pF}$
$t_{PHL}$	Turn-On Delay, Input to Output		10	15	ns	

## TIPOS DE CIRCUITOS INTEGRADOS

Las puertas lógicas de tecnología bipolar TTL (Transistor Transistor Logic) o de tecnología MOSFET (Metal Oxido Semiconductor FET) vienen encapsuladas en circuitos integrados tipo DIP (paquete de doble línea) o tipo montaje superficial (SMT).



## FAMILIAS DE CIRCUITOS INTEGRADOS

A continuación se presentan las diferentes familias de circuitos integrados que se fabrican según el tipo de transistores utilizados ya sean BJT (bipolares) o del tipo MOS. Las puertas NAND pueden venir según sea su aplicación como: 7400, 74L00, 74S00, 74F00, .....



Se identifican con las letras interiores de la referencia. Ejemplo, 7400 tipo estándar, 74LS00 Schottky de baja potencia.

L: TTL de baja potencia

LS: TTL Schottky de baja potencia

S: TTL Schottky

H: TTL de alta velocidad

AS: TTL Schottky avanzada

F: TTL Schottky avanzada de Fairchild

ALS: TTL Schottky avanzada de baja potencia

HC: CMOS de alta velocidad

HCT: CMOS de alta velocidad con entradas TTL

ACT: CMOS avanzada FACT con entrada TTL

AC: CMOS avanzada tecnología FACT

FCT: CMOS Tecnología FACT avanzada compatible TTL

## **NIVELES LÓGICOS DE CIRCUITOS INTEGRADOS**

Los niveles lógicos de voltaje (nivel del cero y del 1), varían de acuerdo a la familia como se indica a continuación.



TTL		
FAMILIA	ENTRADA	SALIDA
TODAS	$V_{IH} = 2.0$ a $5.0V$ $V_{IL} = 0.0$ a $0.8V$	$V_{OH} = 2.4$ a $5.0V$ $V_{OL} = 0.0$ a $0.4V$
ESTANDAR	$I_{IH} = 40\mu A$ $I_{IL} = 1.6mA$	$I_{OH} = 400\mu A$ $I_{OL} = 16mA$
BAJA POTENCIA (LS)	$I_{IH} = 20\mu A$ $I_{IL} = 400\mu A$	$I_{OH} = 400\mu A$ $I_{OL} = 8mA$
AVANZADO BAJA POTENCIA (ALS)	$I_{IH} = 20\mu A$ $I_{IL} = 100\mu A$	$I_{OH} = 400\mu A$ $I_{OL} = 8mA$
AVANZADO FAST (AH)	$I_{IH} = 20\mu A$ $I_{IL} = 0.6mA$	$I_{OH} = 1mA$ $I_{OL} = 20mA$
CMOS		
ALTA VELOCIDAD (HC)	$I_{IH} = 1\mu A$ $I_{IL} = 1\mu A$	$I_{OH} = 4mA$ $I_{OL} = 4mA$
AVANZADA FACT (AC/ACT)	$I_{IH} = 1\mu A$ $I_{IL} = 1\mu A$	$I_{OH} = 24mA$ $I_{OL} = 24mA$

## RESISTENCIA PULL UP DE ENTRADA

La resistencia que se debe conectar a la entrada de un circuito integrado depende del tipo de familia por sus niveles de voltaje como se indica en la siguiente tabla.

SERIE TTL	MAX $I_{IH}$	MAX $I_{IL}$	RESISTENCIA
54/74	40 $\mu A$	-1.6mA	4K $\Omega$
54ALS/74ALS	20 $\mu A$	-0.1mA	40K $\Omega$
54AS/74AS	20 $\mu A$	-0.5mA	8K $\Omega$
54H/74H	50 $\mu A$	-2.0mA	2.8K $\Omega$
54LS/74LS	20 $\mu A$	-0.4mA	18K $\Omega$
54S/74S	50 $\mu A$	-2.0mA	2.8K $\Omega$

## EJEMPLO

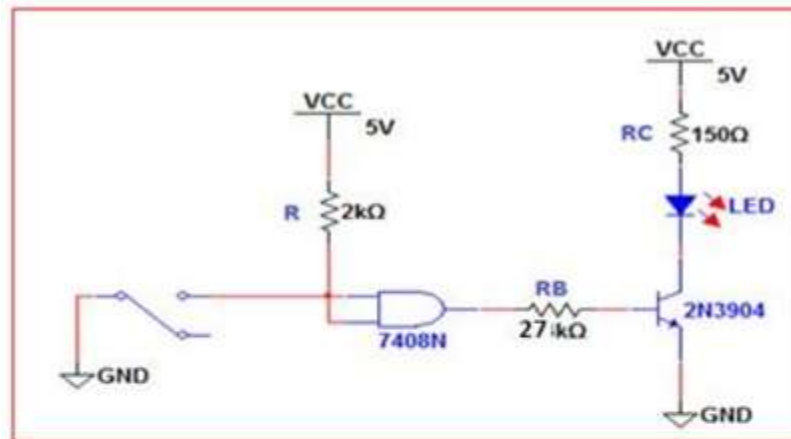
En el circuito de la figura, hallar la  $R_B$  que se debe colocar en la base del transistor.

### Solución

Según la tabla anterior para un CI estándar la resistencia de entrada es de  $4K\Omega$ , pero como son dos entradas se coloca una de  $2K\Omega$ .

Para el diodo:  $V_D=2V$ ,  $I_D=20mA$

para el transistor:  $V_{CESAT}=0.2V$ ,  $\beta \approx 200$ ,  $V_{BESAT}=0.7V$



$$V_D = 2V, I_D = 20mA, V_{CESAT} = 0.2V, \beta \approx 200, V_{BESAT} = 0.7V$$

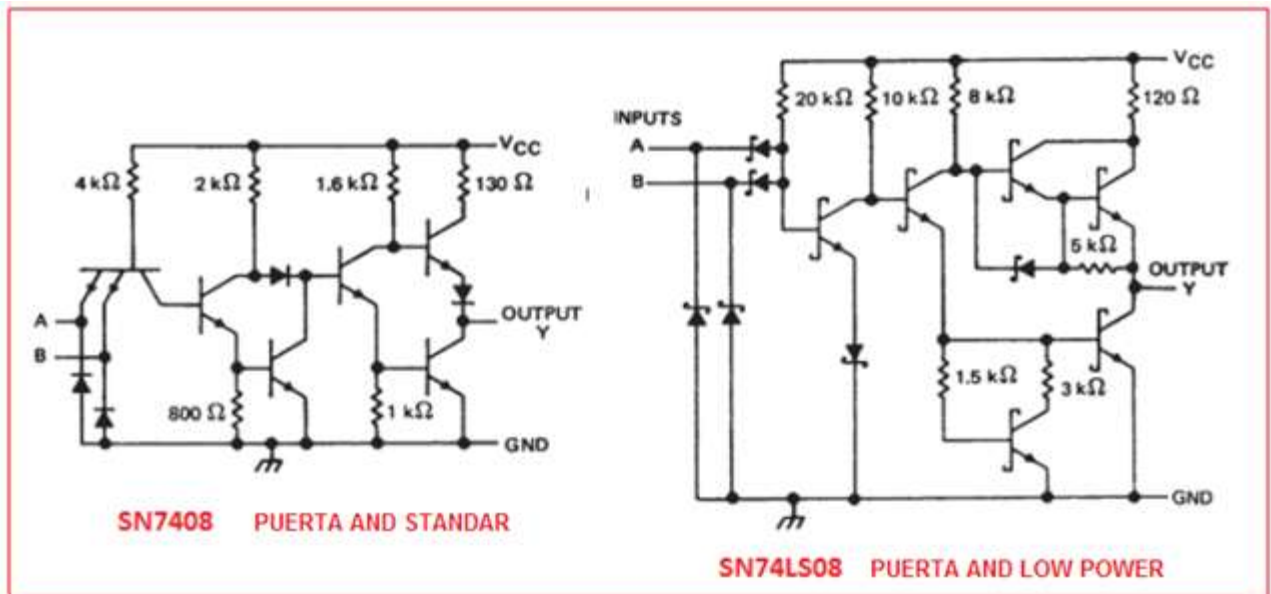
$$R_C \approx \frac{5V - 2V}{20mA} = 0.15K = 150\Omega \quad \frac{I_C}{\beta} = \frac{20mA}{200} = 0.1mA$$

Para que haya saturación del transistor es necesario que:

$$I_B > \frac{I_C}{\beta}, \text{ entonces, } R_B < \frac{5 - 0.7}{0.1} = 30K, \text{ se escoge } 27K$$

Respuesta:  $R_B=27K\Omega$

## DIAGRAMA DE LAS PUERTAS LÓGICAS



## 4. ALGEBRA DE BOOLE

George Boole británico nació en 1815 y murió en 1864. Como inventor del álgebra de Boole, que marca los fundamentos de la aritmética computacional moderna, Boole es considerado como uno de los fundadores del campo de las Ciencias de la computación. Se podría decir que es el padre de las operaciones lógicas y gracias a su álgebra hoy en día es posible manipular operaciones lógicas. Su teoría se base en postulados y teoremas lógicos.

En la lógica proposicional, las **leyes de De Morgan** (Augustus De Morgan, británico, 1806-1871) son un par de reglas de transformación que permiten la expresión de las conjunciones y disyunciones en términos de negación.

Las reglas se pueden expresar en español como:

1. La negación de la conjunción es la disyunción de las negaciones.
2. La negación de la disyunción es la conjunción de las negaciones.

En los siguiente cuadros el apóstrofe significa entrada negada.

#### POSTULADOS

$$A+0=A, \quad A.1=A,$$

$$0.A=0, \quad 1.A=A$$

$$A.A=A, \quad A.A'=0,$$

$$(A')'=A, \quad 1+A=1$$

$$0+A=A, \quad A+A=A,$$

$$A+A'=1$$

#### TEOREMAS

$$A+B=B+A, \text{ conmutativas de la suma}$$

$$A.B=B.A, \text{ conmutativa de multiplicación}$$

$$A+A.B=A.(1+B)=A.1=A$$

$$A+A'.B=A+B$$

$$A.B+A.B'=A.(B+B')=A.1=A$$

#### LEYES DE MORGAN

$$(A.B.C)'=A'+B'+C' \quad (A+B+C)'=A'.B'.C'$$

### EJEMPLO

Simplificar la siguiente expresión:

$$S=(A+AB).(B'+AB).[C'+(A.(A'+B'))+(B.(A'+B'))]+C.[C'+(A.(A'+B'))+(B.(A'+B'))]$$

### Solución

$$S=(A.B'+A.AB+AB.B'+AB.AB).[C'+(AA'+A.B'+B.A'+BB')]+C.[C'+(AA'+A.B'+BA'+BB')]$$

$$S=(A.B'+A.B+AB).[C'+(A.B'+B.A')]+C.[C'+(A.B'+B.A')]$$

$$S=(A.B'+A.B).[C'+(A.B'+B.A')]+C.[C'+(A.B'+B.A')]$$

$$S=(A.B'+A.B)C'+(AB'+AB).(AB'+BA')+C.C'+C'.(AB'+BA')$$

$$S=(A.B'+AB)C'+C(A.B'+A'.B)$$

$$S=AB'C'+ABC'+AB'CA'+A'BC$$

## TABLAS DE VERDAD

Las Tablas de verdad es un recurso lógico para plasmar las ecuaciones lógicas en la tabla, siguiendo la metodología que una variable negada en la tabla es un 0 y una no negada es un 1. Por ejemplo,

Es la forma de dar una salida verdadera del circuito lógico según la función lógica o booleana que represente al circuito. Por ejemplo para el ejercicio anterior donde,

$$S = AB'C + ABC' + AB'C + A'BC$$

S=1 para las entradas lógicas dadas en la tabla de verdad con color rojo.



ENTRADAS			SALIDA
A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

A'BC  
AB'C'  
AB'C  
ABC'

## 5. CIRCUITOS LÓGICOS

A partir de una expresión Booleana o ecuación lógica se puede implementar los circuitos con las puertas lógicas estudiadas. Por ejemplo,

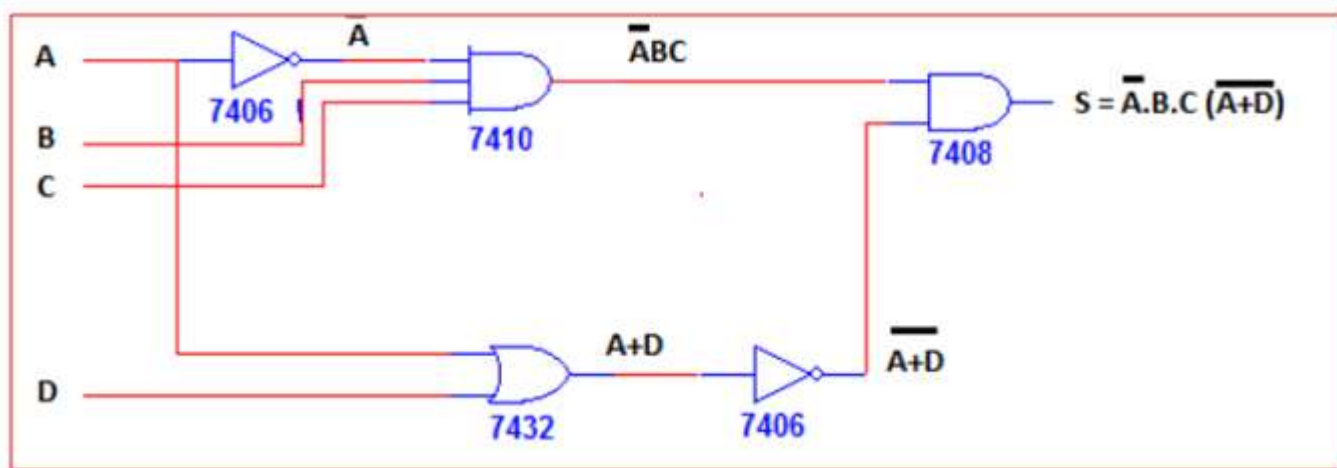
Implementar la ecuación Booleana:  $S = A'BC(A+D)'$

$A'BC$  se puede implementar con una puerta AND de 3 entradas  $A'$ , B y C

$A'$  requiere una NOT o inversor

$(A+D)'$  requiere una OR con entradas A, D y luego un inversor o puerta NOT

Estas dos salidas son las entradas de una AND con entradas  $A'BC$  y  $(A+D)'$



## CONVERSIÓN DE PUERTAS LÓGICAS

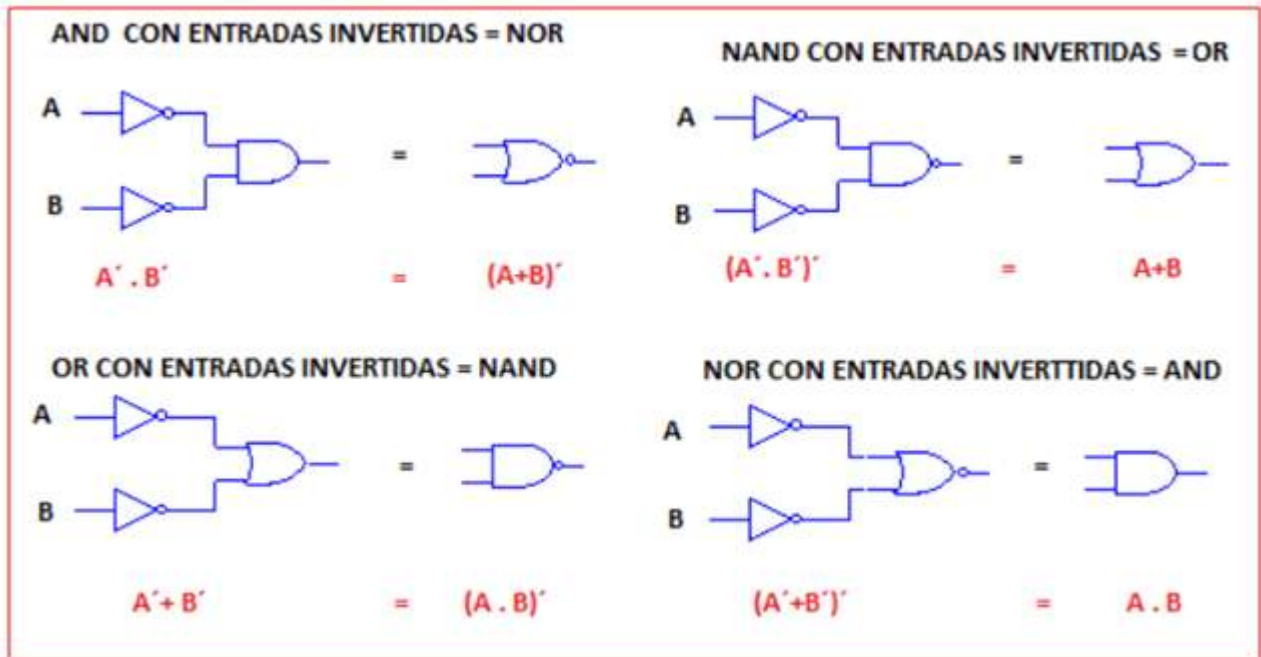
Para que el diseño de un circuito lógico sea más eficiente (utilizar menos circuitos integrados) se requiere con frecuencia convertir una puerta lógica en otra. Aquí se presentan algunas de ellas:

Una NOR es una AND con entradas invertidas

Una OR es una NAND con entradas invertidas

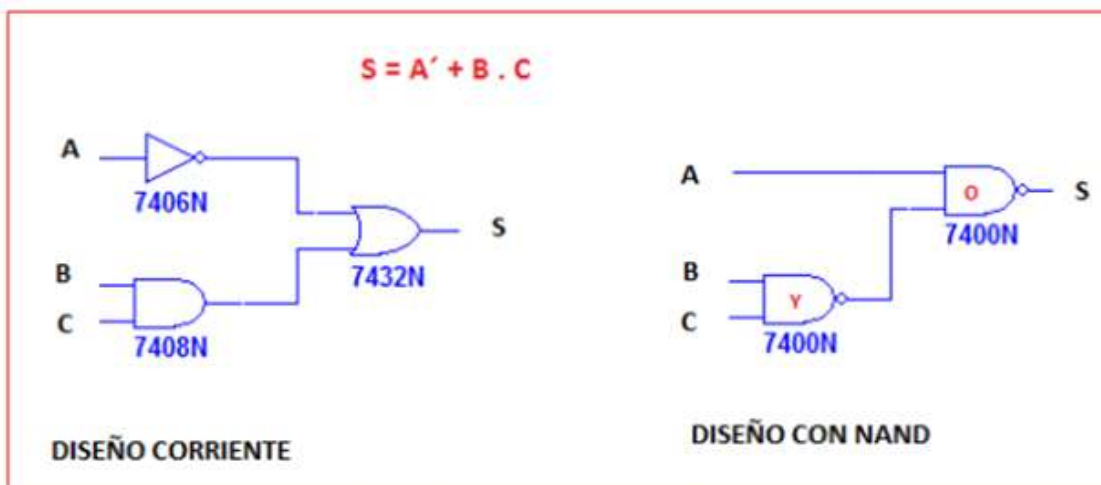
Una NAND es una OR con entradas invertidas

Una AND es una NOR con entradas invertidas



## DISEÑO CON PUERTAS NAND

Es muy frecuente diseñar circuitos lógicos utilizando solamente puertas NAND. Por ejemplo, implementar la ecuación lógica  $S=A'+B.C$  con puertas NAND. Observe que en el diseño corriente se necesitan trece CI, mientras que en el segundo diseño se necesita solamente un CI.

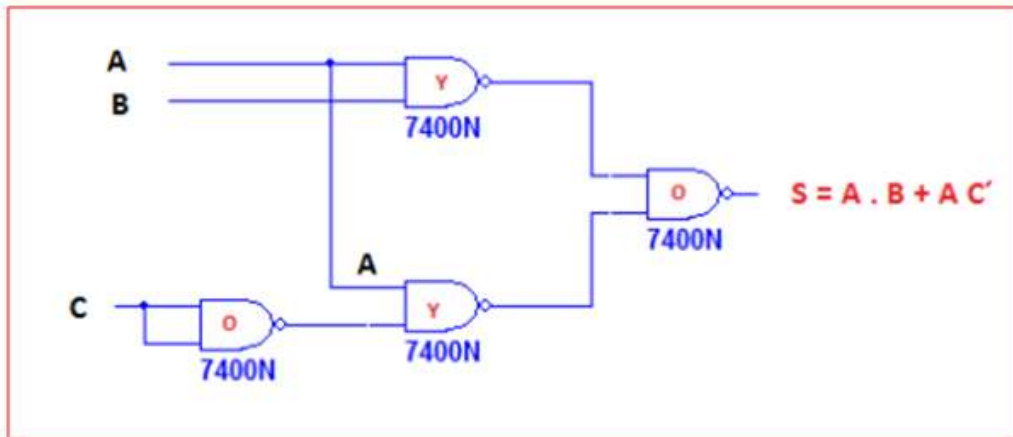




## EJEMPLO

Implementar con puertas NAND la ecuación booleana:  $S = A \cdot B + A \cdot C'$

De salida hacia la entrada el primer nivel de puertas es OR, el segundo nivel es NAND, el tercer nivel es nuevamente OR, etc. Observe para el último nivel de OR se debe invertir las entradas.



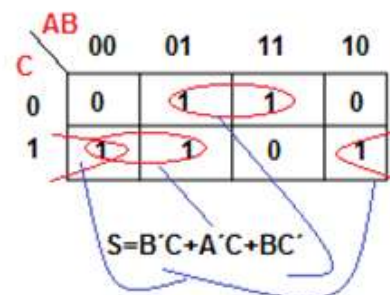
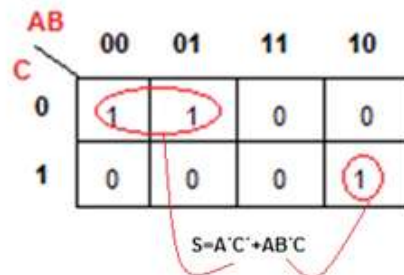
## MAPAS DE KARNAUGH

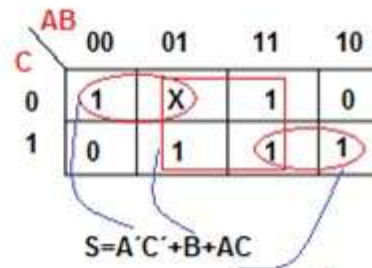
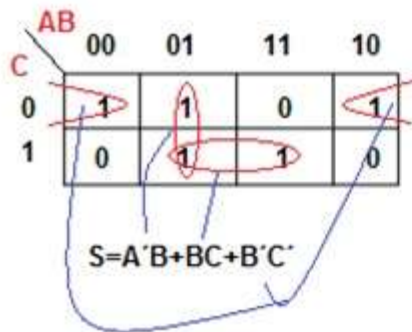
Son mapas que se utilizan para simplificar circuitos, de una forma más sencilla que si se utiliza directamente el álgebra de Boole. Cada término de la función lógica es colocada en la casilla correspondiente: 0 corresponde a variable negada y 1 a variable no negada.  $A'BC' = 010$

Las reglas de simplificación son las siguientes:

- Enlaces de dos elimina una variable
- Enlaces de cuatro elimina dos variables
- Enlaces de ocho elimina tres variables, etc...

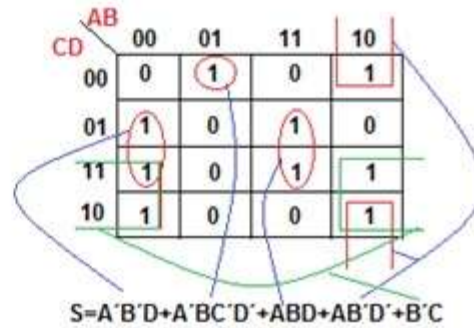
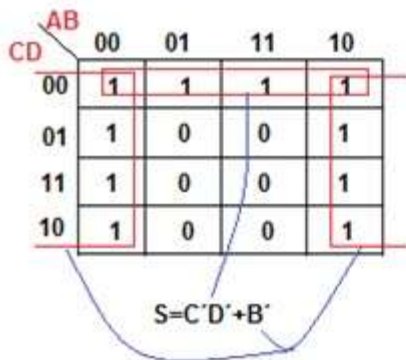
### MAPAS DE TRES VARIABLES





X: Don't care; indiferente

**MAPAS DE CUATRO VARIABLES**



## EJEMPLO DE APLICACIÓN

Diseñar un circuito lógico que controle la puerta de un ascensor en un edificio de tres pisos. El circuito tiene cuatro entradas y una salida. Las entradas P1, P2, P3 indican el piso en que se está (P=1) y M el movimiento del ascensor (M=1 cuando se mueve). La salida S indica cuando la puerta se abre (S=1).

M=0 ascensor detenido

M=1 ascensor en movimiento

S=0 la puerta está cerrada

S=1 la puerta se abre

Son tres pasos:

1. Tabla de verdad
2. Mapa de Karnaugh
3. Circuito lógico

**M P1 P2 P3 S**

0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	X
0	1	0	0	1
0	1	0	1	X
0	1	1	0	X
0	1	1	1	X
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	X
1	1	0	0	0
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

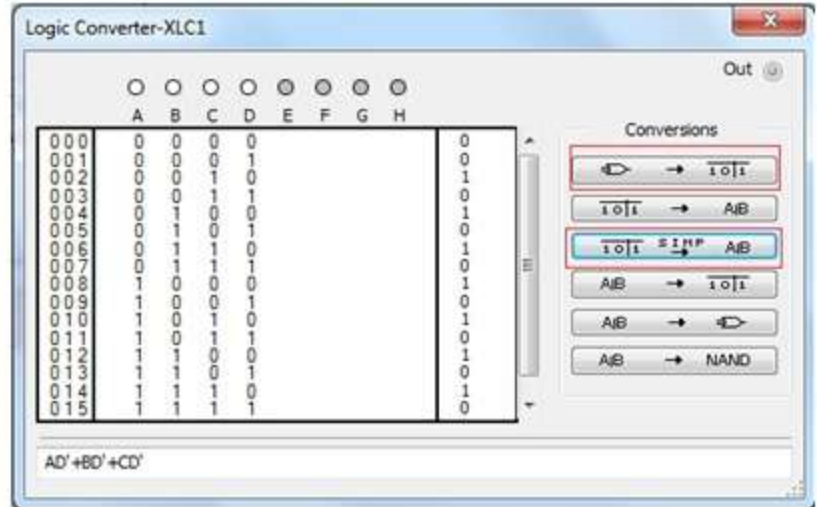
**TABLA DE VERDAD**

# MAPA DE KARNAUGH

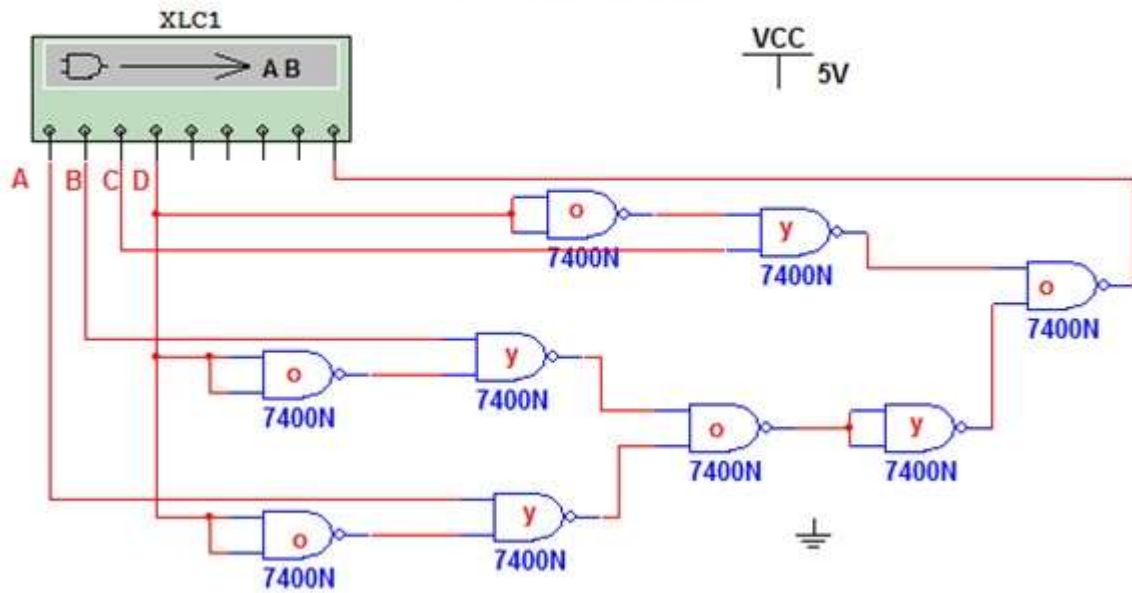
		P2P3			
		00	01	11	10
MP1	00	0	1	x	1
	01	1	x	x	x
	11	0	x	x	x
	10	0	0	x	0

$$S = M'P1 + M'P3 + M'P2 = M'(P1 + P2 + P3)$$

# SIMULACIÓN CON MULTISIM



# CIRCUITO LÓGICO



A=P1, B=P2, C=P3, D=M