

Circuitos y Sistemas Digitales

Funciones Lógicas y Tablas de verdad

Clase 9



1. Introducción

La teoría de conjuntos y su aplicación en la programación es una rama fundamental del álgebra de Boole, que juega un papel crucial en el diseño digital de circuitos electrónicos. Esta teoría permite representar relaciones entre elementos mediante lenguaje formal, lo cual es esencial para el análisis y síntesis de sistemas digitales. En este orden, las funciones lógicas son fundamentales en la programación de computadoras y el diseño de circuitos digitales. Estas funciones definen una salida dependiendo de las entradas proporcionadas. Para comprender mejor cómo funcionan, es útil visualizarlas mediante tablas de verdad, que muestran todos los posibles resultados de las combinaciones de entrada.

El estudio de las expresiones minterm y maxterm es otro aspecto importante en la simplificación de funciones lógicas. Las expresiones minterm representan el estado "1" para cada fila de una tabla de verdad, mientras que las maxterm representan el estado "0". La simplificación de estas expresiones utilizando técnicas booleanas, en esta primer parte del curso, es esencial para reducir la complejidad y costos del diseño de circuitos digitales.

clase 9: Funciones Lógicas y Tablas de verdad.

Resultado o resultados de aprendizaje que será abordado con el contenido de la clase. **(No agregue los criterios)**

Diseñar circuitos digitales y sistemas digitales complejos, empleando las herramientas adecuadas.

3.5. Funciones Lógicas y Tablas de verdad

Las funciones lógicas son relaciones matemáticas entre dos o más variables booleanas (es decir, variables que pueden tomar solo dos valores posibles: verdadero [true] o falso [false]). Estas funciones definen una salida como resultado de las combinaciones posibles de las entradas. La representación gráfica de cómo estas funciones se comportan en función de todas sus entradas es conocida como tabla de verdad.

En la tabla de verdad, cada fila representa un conjunto diferente de valores de entrada y la columna correspondiente a la salida muestra el resultado de la función lógica para esos valores de entrada. Las tablas de verdad son una herramienta valiosa en el diseño digital ya que permiten visualizar y analizar los posibles comportamientos de una función lógica. También son útiles para determinar las funciones equivalentes, simplificar expresiones y diseñar circuitos digitales.

En este punto y antes de desarrollar tablas de verdad con lógica booleana, es importante recordar la equivalencia entre los sistemas de numeración que se relacionan con la computación, misma que se muestra en la Tabla 1

Tabla 1

Equivalencia entre sistemas de numeración decimal, binario, octal y hexadecimal.

Decimal	Binario	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	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

Nota: Elaborado por Escobar C. (2025)

3.5.1. BCD (Binary-Coded Decimal)

El Código Binario Codificado por Dígito Decimal es un método de codificación en el que cada dígito decimal (0-9) se representa como un grupo de 4 bits. Es decir, cada dígito del sistema decimal se convierte a su equivalente binario. BCD es especialmente útil en la entrada y salida de números decimales, como cuando se registran datos de teclado o muestran información en displays. Aunque no es tan eficiente en términos de espacio que el código binario estándar para representar los mismos números, su simplicidad hace que sea una opción popular en sistemas donde la facilidad de lectura humana es más importante que el espacio de almacenamiento o la velocidad de procesamiento. La Tabla 2 muestra su equivalencia.

Tabla 2

Tabla BCD (0-9)

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

Nota: Elaborado por Escobar C. (2025)

El Código Binario Codificado por Dígito Decimal es un método en el que cada dígito decimal (0-9) se representa como un grupo de 4 bits. Es decir, cada dígito del sistema decimal se convierte a su equivalente binario.

Ejemplo:

- El número decimal 5 se representa en BCD como 0101.
- El número decimal 23 se representaría como 0010 0011.

3.5.2. Función lógica.

Las funciones lógicas son fundamentales en la teoría del álgebra de Boole, que es una rama de la matemática utilizada ampliamente en el diseño y análisis de sistemas digitales. Estas funciones permiten representar decisiones y operaciones mediante valores booleanos (verdadero o falso), lo cual es esencial para el funcionamiento de los circuitos electrónicos y las computadoras. Las funciones lógicas son una parte integral del álgebra de Boole y se utilizan para expresar decisiones y operaciones en términos de entradas (variables) y salidas booleanas. Una función lógica puede ser definida como una relación entre variables de entrada que produce un resultado de salida, donde los valores posibles de las variables son generalmente verdadero o falso.

Una función lógica es una función matemática que toma como entrada variables booleanas (0 o 1) y produce una salida también booleana. Estas funciones se utilizan para modelar el comportamiento de circuitos digitales, donde los estados lógicos (0 y 1) representan dos posibles estados del sistema, como ejemplo sencillo considere un interruptor simple que puede estar encendido o apagado. Podemos representar el estado del interruptor con una variable booleana: 1 si está encendido y 0 si está apagado.

A. Operadores Lógicos

Los operadores lógicos son los símbolos que se utilizan para combinar las variables en funciones lógicas. Los tres primeros operadores lógicos básicos con sus respectivas tablas de verdad se revisaron previamente en la clase 5:

- **AND:** (Conjunción): Representado por el símbolo $\wedge$ o un punto ($\cdot$). Produce 1 solo si todas las entradas son 1.
- **OR:** (Disyunción): Representado por el símbolo $\vee$ o un signo más (+). Produce 1 si al menos una de las entradas es 1.
- **NOT:** (Negación): Representado por el símbolo $\neg$, Invierte el valor de la entrada.

De manera complementaria a los operadores básicos previos incorporaremos, con sus respectivas tablas de verdad, los operadores:

- **NAND (No AND):** Representado por $\bar{\wedge}$. Produce 0 solo si todas las entradas son 1.
- **NOR (No OR):** Representado por $\bar{\vee}$. Produce 1 solo si todas las entradas son 0.
- **XOR (OR Exclusivo):** Representado por $\oplus$. Produce 1 solo si las entradas son diferentes.

- **XNOR (No OR Exclusivo):** Representado por $\odot$. Produce 1 solo si las entradas son iguales.

B. Tabla de verdad NAND:

La Tabla 3 muestra la tabla de verdad del operador NAND, dónde se puede apreciar que la salida es 0 solo si todas las entradas son 1.

La figura 1 muestra el empaquetamiento de un circuito integrado de 4 compuertas NAND. En este documental puede ver como se fabrican los circuitos integrados.

Tabla 3

Tabla de verdad del operador NAND

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

Nota: Elaborado por Escobar C. (2025)

Figura 1

Circuito integrado NTE74LS00.



Nota: Recuperado de (NTE Electronics, 2025a)

Figura 2

DM74LS00 Quad 2-Input NAND Gate



August 1986
Revised March 2000

DM74LS00 Quad 2-Input NAND Gate

General Description

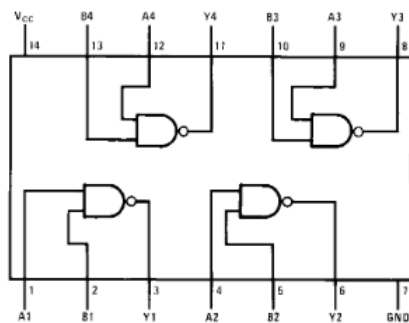
This device contains four independent gates each of which performs the logic NAND function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS00M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS00SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS00N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = \overline{AB}$$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

Nota: Recuperado de (Fairchild Semiconductor Datasheet, 2000b).

C. Tabla de verdad NOR:

La Tabla 4 muestra la tabla de verdad del operador NOR, donde se puede apreciar que la salida es 0 solo si todas las entradas son 1.

Tabla 4

Tabla de verdad del operador NOR

A	B	$A \nabla B$
0	0	1
0	1	0

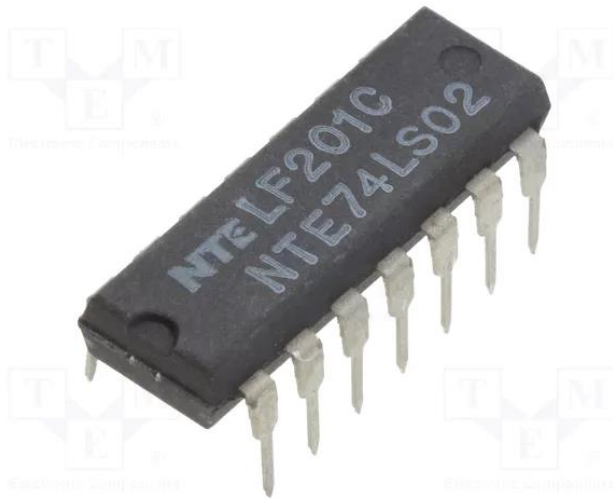
A	B	$A \nabla B$
1	0	0
1	1	0

Nota: Elaborado por Escobar C. (2025)

Las Figura 3 y Figura 4 muestran respectivamente el empaquetamiento comercial de este circuito integrado y su diagrama esquemático de conexiones

Figura 3

Circuito integrado NTE74LS02



Nota: Recuperado de (NTE Electronics, 2025b).

Figura 4

DM74LS02 Quad 2-Input NOR Gate.



May 1986
Revised March 2000

DM74LS02 Quad 2-Input NOR Gate

General Description

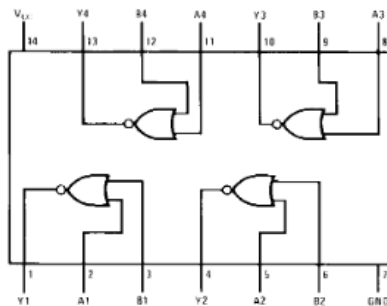
This device contains four independent gates each of which performs the logic NOR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS02M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS02SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS02N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = \overline{A + B}$$

Inputs		Output
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

Nota: Recuperado de (Fairchild Semiconductor Datasheet, 2000c).

D. Tabla de verdad XOR:

La Tabla 5 muestra la tabla de verdad del operador XOR, donde se puede apreciar que la salida es 1 solo si las entradas son diferentes.

Tabla 5

Tabla de verdad del operador XOR.

A	B	$A \oplus B$
0	0	0
0	1	1

A	B	$A \oplus B$
1	0	1
1	1	0

Nota: Elaborado por Escobar C. (2025)

Las Figura 5 y Figura 6 muestran respectivamente el empaquetamiento comercial de este circuito integrado y su diagrama esquemático de conexiones

Figura 5

Circuito integrado NTE74LS86.



Nota: Recuperado de (NTE Electronics, 2025d).

Figura 6

74LS86 Quad 2-Input Exclusive-OR Gate.



August 1986
Revised March 2000

DM74LS86

Quad 2-Input Exclusive-OR Gate

General Description

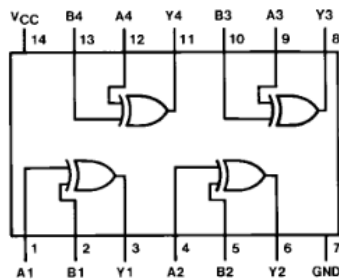
This device contains four independent gates each of which performs the logic exclusive-OR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS86M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS86SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS86N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

Nota: Recuperado de (Fairchild Semiconductor Datasheet, 2000a).

E. Tabla de verdad XNOR:

La Tabla 6 muestra la tabla de verdad del operador XNOR, donde se puede apreciar que la salida es 0 solo si las entradas son diferentes.

Tabla 6

Tabla de verdad del operador XNOR.

A	B	$A \odot B$
0	0	1
0	1	0

A	B	A B	⊙
1	0	0	
1	1	1	

Nota: Elaborado por Escobar C. (2025)

Las Figura 7 y Figura 8 muestran respectivamente el empaquetamiento comercial de este circuito integrado y su diagrama esquemático de conexiones

Figura 7

Circuito integrado NTE74LS266.



Nota: Recuperado de (NTE Electronics, 2025c).

Figura 8

DM74LS266 Quad 2-Input Exclusive-NOR Gate.



March 1989
Revised March 2000

DM74LS266

Quad 2-Input Exclusive-NOR Gate with Open-Collector Outputs

General Description

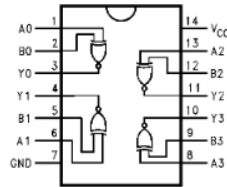
This device contains four independent gates each of which performs the logic exclusive-NOR function. Outputs are open collector.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS266M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS266N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Truth Table

Inputs		Outputs
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	H

H = HIGH Voltage Level
L = LOW Voltage Level

Nota: Recuperado de (Fairchild Semiconductor Datasheet, 2000a).

[El mercado ofrece una amplia variedad de compuertas lógicas que son usadas para diseñar los circuitos digitales, en este enlace puede profundizar acerca de tipos, diseño y aplicaciones de compuertas lógicas \(Isaac et al., 2025\).](#)

3.3.3. Tablas de Verdad

Aunque las hemos venido usando informalmente para describir las tablas de verdad de los operadores lógicos, de manera formal una tabla de verdad es una representación matemática que muestra todos los posibles estados de entrada y sus correspondientes salidas para una función lógica específica. Cada fila en la tabla representa una combinación única de valores de entrada, y la columna final muestra el valor de salida resultante.

Así por ejemplo consideremos la tabla de verdad para la función $F(A,B)=A \wedge (B \vee \neg A)$.

A	B	$\neg A$	$B \vee \neg A$	$A \wedge (B \vee \neg A)$
0	0	1	1	0
0	1	1	1	0
1	0	0	0	0
1	1	0	1	1

Aquí se tienen dos variables binarias de entrada; A y B, por tanto para construir la tabla de verdad se deben considerar todas las posibles combinaciones de A y B, en este caso como son dos variables las combinaciones posibles son 2^2 , es decir 4 y estas son 00, 01, 10 y 11, nótese que si se lee como números binarios corresponden a 0, 1, 2 y 3 en decimal, dato que nos servirá mas adelante para la simplificación gráfica de expresiones booleanas.

Las dos primeras columnas de la tabla de verdad muestran todas las combinaciones posibles de A y B, luego conforme a la función de ejemplo $F(A,B)=A \wedge (B \vee \neg A)$, se debe obtener el valor de $\neg A$, por lo que la tercera columna muestra los valores de $\neg A$. La siguiente operación para respñver esta función es “ $B \vee \neg A$ ”, por lo que la cuarta columna muestra este contenido, y finalmente en la última columna se muestra el resultado de la operación de la función “ $A \wedge (B \vee \neg A)$ ”.

3.3.3.1. Simplificación de Funciones Lógicas

La simplificación de funciones lógicas es un proceso que busca reducir la complejidad de una función, utilizando leyes y teoremas booleanos, con el objetivo de obtener una expresión más eficiente. Esto puede resultar en circuitos digitales más sencillos y eficientes.

Para ello se deben tener presente las leyes y teoremas del algebra booleana revisados previamente en la clase previa.

Como ejemplo; si se tiene la función $F(A,B,C)=A'B'C' + AB'C' + ABC' + ABC$.

Para facilitar la lectura de las variables negadas se está usando la sintaxis de incorporar un apóstrofe junto a las variables negadas, así; A' corresponde a la negación de A o A negada.

Primero, agrupamos los dos primeros términos y los dos últimos términos:

$$F = B'C'(A' + A) + AB(C' + C)$$

Dónde

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

$$(C' + C) = 1$$

Reemplazando

$$F = B'C' + AB$$

La cual es una versión simplificada de la función original.

Para esta función simplificada elaboraremos la tabla de verdad, dado que tenemos tres variables de entrada A, B y C, entonces son posibles 2^3 combinaciones, es decir 8; del 0 al 7 en decimal. A continuación construimos la tabla de verdad llenando la tabla de verdad de acuerdo a las operaciones booleanas que contemplan la función.

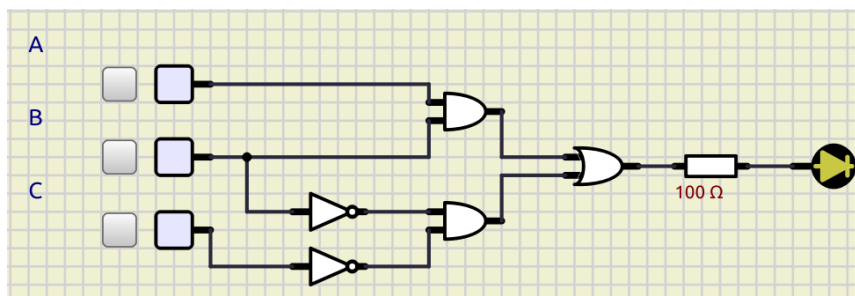
	A	B	C	B'	C'	B'C'	AB	F = B'C' + AB
0	0	0	0	1	1	1	0	1
1	0	0	1	1	0	0	0	0
2	0	1	0	0	1	0	0	0
3	0	1	1	0	0	0	0	0
4	1	0	0	1	1	1	0	1
5	1	0	1	1	0	0	0	0
6	1	1	0	0	1	0	1	1
7	1	1	1	0	0	0	1	1

De manera complementaria, esta función la podemos implementar en el simulador Simulide (González et al., 2025) y comprobar el funcionamiento con todas las combinaciones de las variables de entrada.

Así, para la combinación 000 de las entradas ABC la salida debe ser 1, lo cual se valida en la Figura 9.

Figura 9

Combinación de entrada 000 para la función $F = B'C' + AB$

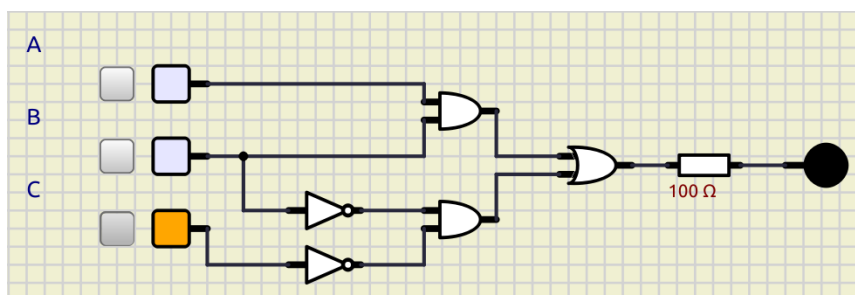


Nota: Elaborado por Escobar C. (2025)

Para la combinación 001 de las entradas ABC, conforme nuestra tabla de verdad desarrollada previamente, la salida debe ser 1, lo cual se valida en la Figura 10.

Figura 10

Combinación de entrada 001 para la función $F = B'C' + AB$



Nota: Elaborado por Escobar C. (2025)

Para profundizar sus conocimientos revise el documento de profundización de esta sesión donde se muestran todas las combinaciones en la simulación “función1.sim1” para Simulide y verifique que la salida de la función es 1, para las combinaciones de Entrada ABC.

3.5.3. Minterm y Maxterm

En lógica booleana, los másterms y miniterms son conceptos fundamentales que se utilizan para representar funciones booleanas en términos de productos y sumas canónicas. Estos conceptos son esenciales para la simplificación y análisis de circuitos digitales.

A. Minterm

Es el producto lógico (AND) de todas las variables de una función, donde cada variable aparece una vez, ya sea en forma directa o complementada. Representa una combinación única de entradas que produce una salida 1. Para n variables booleanas, existen 2^n minterms posibles.

Por ejemplo, consideremos una función de dos variables booleanas, A y B:

Los minterms posibles son: $A'B'$, $A'B$, AB' , AB .

un minterm se puede representar como m_i , donde i es el número en binario que corresponde a la combinación de valores de las variables.

Por ejemplo para una función de tres variables A,B,C:

Los minterms posibles son:

$$m_0 = A'B'C'$$

$$m_1 = A'B'C$$

$$m_2 = A'BC'$$

$$m_3 = A'BC$$

$$m_4 = AB'C'$$

$$m_5 = AB'C$$

$$m_6 = ABC'$$

$$m_7 = ABC$$

Debe tomarse en cuenta que cuando se expresa una función en base a los minterm, esta corresponden a la suma de minterms en cuya salida existe un uno lógico "1".

Por ejemplo la función de minterm: $\{m_0, m_6, m_7\}$, corresponderían a la tabla de verdad:

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

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

$$F = \{m_0, m_6, m_7\}$$

Lo cual es equivalente a:

$$F = A'B'C' + ABC' + ABC$$

B. Maxterm

Por otro lado los maxterm son el resultado de la suma lógica (OR) de todas las variables de una función, donde cada variable aparece una vez, ya sea en forma directa o complementada. Representa una combinación única de entradas que produce una salida 0. Para n variables booleanas, existen 2^n maxterm posibles.

Siguiendo el ejemplo anterior con dos variables booleanas A y B:

Los maxterm posibles son: $(A' + B')$, $(A' + B)$, $(A + B')$, $(A + B)$

Un maxterm se puede representar como M_i , donde i es el número en binario que corresponde a la combinación de valores de las variables.

Por ejemplo para una función de tres variables A,B,C los maxterm posibles son:

$$M_0 = A + B + C$$

$$M_1 = A + B + C'$$

$$M_2 = A + B' + C$$

$$M_3 = A + B' + C'$$

$$M_4 = A' + B + C$$

$$M_5 = A' + B + C'$$

$$M_6 = A' + B' + C$$

$$M_7 = A' + B' + C'$$

En base al ejemplo previo de la función expresada en minterm, su equivalente en maxterm sería:

$$F = \{M_1, M_2, M_3, M_4, M_5\}$$

Nótese que los maxterm de una función corresponden al producto de las sumas cuyas salidas sean cero lógico "0", es decir en este caso:

$$F = (A+B+C')(A+B'+C)(A+B'+C')(A'+B+C)(A'+B+C')$$

Referencias citadas en la clase 9.

Fairchild Semiconductor Datasheet. (2000a). Quad 2-Input Exclusive-OR Gate. Quad 2-Input Exclusive-OR Gate. <https://www.alldatasheet.com/datasheet-pdf/view/51095/FAIRCHILD/74LS86.html>

Fairchild Semiconductor Datasheet. (2000b). Quad 2-Input NAND Gate. Quad 2-Input NAND Gate. <https://www.alldatasheet.com/datasheet-pdf/view/51021/FAIRCHILD/74LS00.html>

Fairchild Semiconductor Datasheet. (2000c). Quad 2-Input NOR Gate. Quad 2-Input NOR Gate. <https://www.alldatasheet.com/datasheet-pdf/view/51020/FAIRCHILD/74LS02.html>

González, S., De Francisco, P., Padrah, Z., Vrana, J., Kellner, K., Salishhchev, A., & Otros. (2025). Simulador de circuitos SimulIDE. Simulador de Circuitos SimulIDE. <https://github.com/KDE/ktechlab>

Isaac, P. E., Arboledas, J., García, J., & De Frutos, T. (2025). Hardware Libre. Hardware Libre. <https://www.hwlibre.com/>

NTE Electronics. (2025a). Compuerta NAND NTE74LS00. Compuerta NAND NTE74LS00. <https://www.tme.com/pe/es/details/nte74ls00/barreras-inversores/nte-electronics/>

NTE Electronics. (2025b). Compuerta NAND NTE74LS02. Compuerta NAND NTE74LS02. <https://www.tme.com/pe/es/details/nte74ls02/barreras-inversores/nte-electronics/>

NTE Electronics. (2025c). Compuerta XNOR NTE74LS266. Compuerta XNOR NTE74LS266. <https://www.tme.com/pe/es/details/nte74ls266/barreras-inversores/nte-electronics/>

NTE Electronics. (2025d). Sumador de 4 bits 74LS86. Sumador de 4 Bits 74LS86. <https://www.tme.com/pe/es/details/nte74ls86/barreras-inversores/nte-electronics/>

Definición de los términos citados en la clase 9.

Función lógica: Una función lógica puede ser definida como una relación entre variables de entrada que produce un resultado de salida, donde los valores posibles de las variables son generalmente verdadero o falso.

Tabla de verdad: una tabla de verdad es una representación matemática que muestra todos los posibles estados de entrada y sus correspondientes salidas para una función lógica específica.



La excelencia no se improvisa

síguenos

