

ELECTRÓNICA DIGITAL

Conceptos y Métodos Básicos

Oscar Ignacio Botero Henao

iCartesiLibri

ELECTRÓNICA DIGITAL

Conceptos y Métodos Básicos

Oscar Ignacio Botero Henao
Institución Universitaria Pascual Bravo



Fondo Editorial RED Descartes



Córdoba (España)
2022

Título de la obra:
ELECTRÓNICA DIGITAL
Conceptos y Métodos Básicos

Autor:
Oscar Ignacio Botero Henao

Revisión:
Juan Guillermo Rivera Berrío

Código JavaScript para el libro: [Joel Espinosa Longi](#), [IMATE](#), UNAM.
Recursos interactivos: [DescartesJS](#)
Fuentes: [Lato](#) y [UbuntuMono](#)

Red Educativa Digital Descartes
Córdoba (España)
descartes@proyectodescartes.org
<https://proyectodescartes.org>

Proyecto iCartesiLibri
<https://proyectodescartes.org/iCartesiLibri/index.htm>

ISBN: 978-84-18834-46-2



Esta obra está bajo una licencia Creative Commons 4.0 internacional: Reconocimiento-No Comercial-Compartir Igual.

Tabla de contenido

Agradecimientos	7
Prefacio	9
1. GLOSARIO	15
1.1 Glosario Básico	17
1.2 Objeto Interactivo de Aprendizaje - OIA	22
2. FAMILIAS LÓGICAS DE CIRCUITOS INTEGRADOS	25
2.1 Familias Lógicas	27
2.2 Familias lógicas TTL y CMOS	27
3. CLASIFICACIÓN DE LA INTEGRACIÓN DE LOS CIRCUITOS INTEGRADOS	33
3.1 Clasificación	35
3.2 Personajes Históricos en la Electrónica	36
3.3 Objeto Interactivo de Aprendizaje - OIA	39
4. SISTEMAS NUMÉRICOS ANTIGUOS Y MODERNOS.	41
Conversiones con números enteros y fraccionarios	
4.1 Sistemas de Numeración	43
4.2 Sistemas Numéricos Antiguos	44
4.3 Objetos Interactivos de Aprendizaje - OIA	51
4.4 Sistemas Numéricos Modernos en la Informática	52
4.5 Conversiones entre Sistemas Numéricos	56
4.5.1 Conversión de Decimal a Binario	56
4.5.2 Conversión de Decimal a Octal	59
4.5.3 Conversión de Decimal a Hexadecimal	61
4.5.4 Conversión de Binario a Decimal	63

4.5.5 Conversión de Octal a Decimal	65
4.5.6 Conversión de Hexadecimal a Decimal	66
4.5.7 Conversión de Binario a Octal	67
4.5.8 Conversión de Octal a Binario	69
4.5.9 Conversión de Binario a Hexadecimal	70
4.5.10 Conversión de Hexadecimal a Binario	71
5. CREACIÓN DE TABLAS DE VERDAD	75
5.1 Variables y Combinaciones para crear una tabla de verdad	77
5.1.1 Tabla con una variable	78
5.1.2 Tabla con dos variables	78
5.1.3 Tabla con tres variables	79
5.1.4 Tabla con cuatro variables	80
6. COMPUERTAS LÓGICAS Y FUNCIONES BOOLEANAS	83
6.1 George Boole	85
6.2 Compuertas Lógicas	85
6.2.1 Compuertas YES y su complemento NOT	86
6.2.2 Compuertas AND y su complemento NAND	87
6.2.3 Compuertas OR y su complemento NOR	88
6.2.4 Compuertas XOR y su complemento XNOR	89
6.3 Referencias Comerciales de las Compuertas Lógicas	91
6.4 Estructura interna de las compuertas lógicas básicas	91
6.4.1 Estructura interna de las compuertas lógicas básicas en la familia TTL	92
6.4.2 Estructura interna de las compuertas lógicas básicas en la familia CMOS (1 y 2 entradas)	93
6.5 Funciones Booleanas	94

7. MINTERM Y MAXTERM (Lógica AOI)	103
7.1 Introducción	105
7.2 Minterm o Minitérminos (Σ)	106
7.3 Maxterm o Maxitérminos (Π)	107
7.4 Ejemplos de Minterm y Maxterm	109
8. MAPAS DE KARNAUGH	115
8.1 Maurice Karnaugh y John Venn	117
8.2 Procedimiento	118
8.3 Distribución de las celdas	119
8.3.1 Mapa de Karnaugh de dos (2) variables	120
8.3.2 Mapa de Karnaugh de tres (3) variables	120
8.3.3 Mapa de Karnaugh de cuatro (4) variables	122
8.4 Ejercicios con Mapas de Karnaugh	124
8.4.1 Agrupación de datos adyacentes	124
8.5 Objetos Interactivos de Aprendizaje (OIA)	130
9. EPÍLOGO	133

Agradecimientos

Ofrezco mis más sinceros agradecimientos a las siguientes personas, debido a su apoyo y colaboración en la elaboración de este Libro Interactivo de Aprendizaje (LIA).

- Doctor Juan Guillermo Rivera Berrio
- Magister y Profesor Ramiro Antonio Lopera Sánchez
- Magister y Profesor Sergio Hernando Ruiz Obando
- Profesor José María Fernández
- Profesional Gustavo Adolfo Tobón Pereira
- Mónica María Díaz Moreno (esposa)

Se incluyen biografías simples de personajes fundamentales para esta área, notas informativas, fórmulas, ejemplos, ejercicios propuestos con respuesta y objetos interactivos de aprendizaje (OIA); que sirven de apoyo al aprendizaje del área de **Electrónica Digital**.

ESTILOS PARA PERSONAJES:



Nombre (Fechas de vida) Profesión y Nacionalidad.
Biografía simple.

NOTAS COMPLEMENTARIAS:

NOTA: Información complementaria.

FÓRMULAS:

$2^n =$ No. de combinaciones posibles para un sistema digital

ETIQUETAS PARA LOS EJEMPLOS:

► **Ejemplo:** La etiqueta en color púrpura indica que es uno o varios ejemplos solucionados como demostración.

ETIQUETAS PARA LOS EJERCICIOS:

► **Ejercicio:** La etiqueta en color verde indica ejercicios con respuestas para que resuelva y autoevalúe el aprendizaje adquirido.

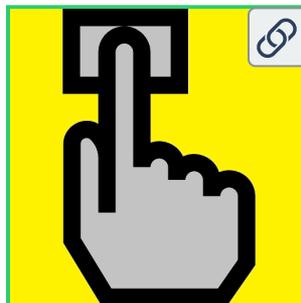
OBJETOS INTERACTIVOS DE APRENDIZAJE - OIA:

Dando clic en la esquina superior derecha se expande el OIA.



Ñ	L	Q	O	L	J	R	L	S	L	Q	L	R	V	M
S	Q	C	K	W	T	C	A	W	J	Ñ	J	J	Y	Z
O	Y	M	P	Ñ	A	M	G	P	I	L	X	A	S	P
L	Q	H	L	O	U	K	R	X	N	B	T	S	C	O
P	P	E	R	S	O	N	A	J	E	S	B	O	Ñ	O
M	S	O	I	C	I	C	R	E	J	E	G	V	D	F
E	P	R	E	N	D	I	Z	A	J	E	I	S	B	B
J	A	L	U	S	M	X	R	U	E	G	M	O	X	T
E	V	F	Y	S	A	T	O	N	Ñ	K	A	E	L	E
K	M	T	P	G	C	T	A	F	Z	V	G	D	Q	F
I	N	F	O	R	M	A	C	I	Ó	N	E	I	B	C
U	J	N	W	S	A	Z	S	X	C	P	N	V	A	M
Ñ	B	O	D	Z	G	Q	M	X	A	U	E	V	F	W
W	B	A	Z	A	I	O	J	I	J	V	S	Q	F	G
N	F	P	S	L	I	A	P	A	V	Y	W	F	M	N

Este **L**ibro **I**nteractivo de **A**prendizaje (**LIA**) titulado "**ELECTRÓNICA DIGITAL, Conceptos Básicos**" contiene figuras a las cuales usted como usuario, le puede dar "CLIC" para que ejecute un "zoom" (ampliación de la imagen), con el objetivo de visualizarla en una ventana emergente de forma expandida y poder visualizarla plenamente. En las figuras expandibles observará el siguiente ícono dentro de ella.



!Ensayo! Dando clic sobre la imagen, observa si la figura es expandible.

Función lógica

$F = \{(A \times B') \text{ XOR } C'\} + (B+D)'$

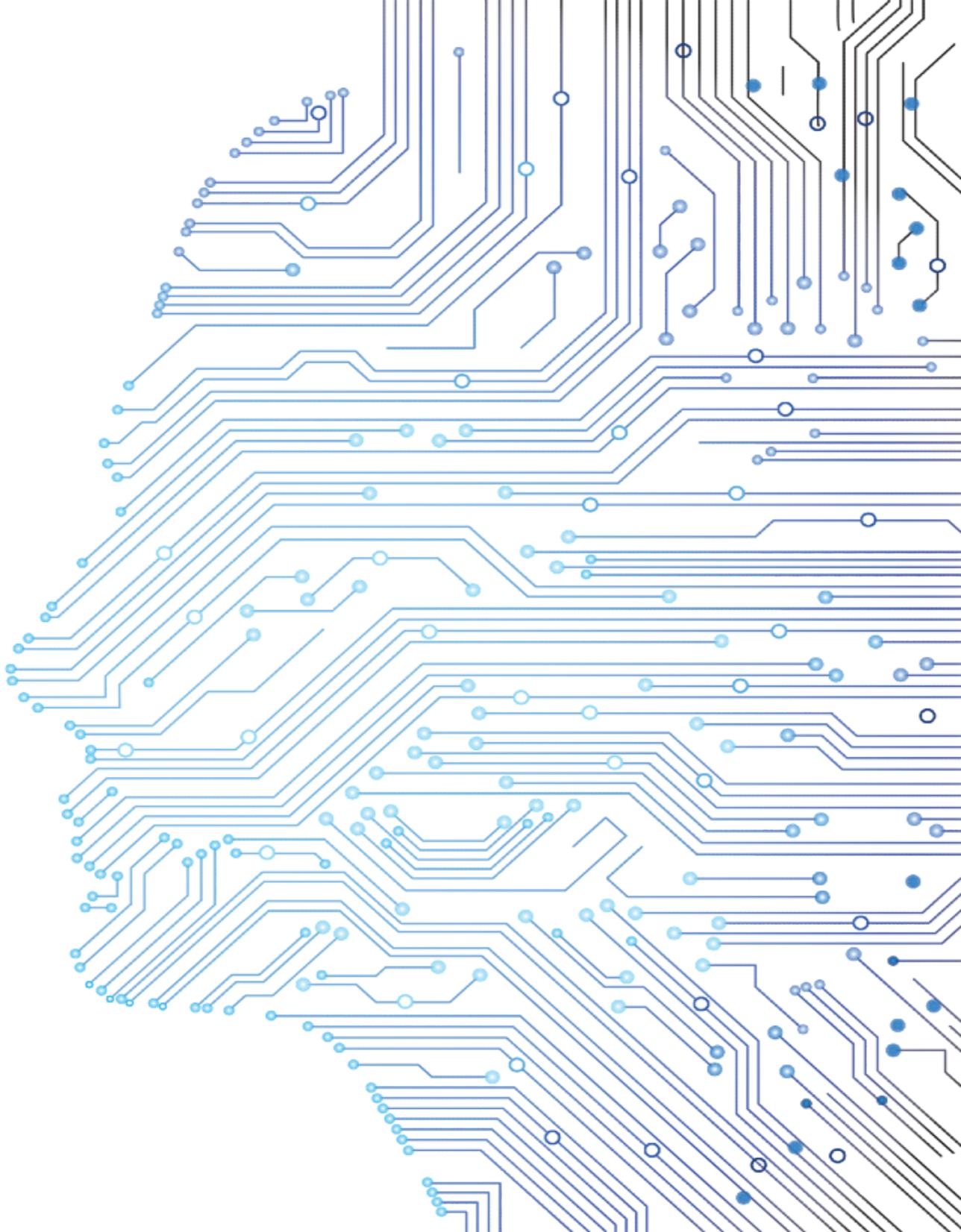
No. DEC	D	C	B	A	B'	1 (A x B')	3 C'	4 (2 xor 3)	5 (B nor D)	6 = F 4 + 5
0	0	0	0	0	1	0	1	1	1	1
1	0	0	0	1	1	1	1	0	1	1
2	0	0	1	0	0	0	1	1	0	1
3	0	0	1	1	0	0	1	1	0	1
4	0	1	0	0	1	0	0	0	1	1
5	0	1	0	1	1	1	0	1	1	1
6	0	1	1	0	0	0	0	0	0	0
7	0	1	1	1	0	0	0	0	0	0
8	1	0	0	0	1	0	1	1	0	1
9	1	0	0	1	1	1	1	0	0	0
10	1	0	1	0	0	0	1	1	0	1
11	1	0	1	1	0	0	1	1	0	1
12	1	1	0	0	1	0	0	0	0	0
13	1	1	0	1	1	1	0	1	0	1
14	1	1	1	0	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0

Compuertas NOT AND NOT XOR NOR OR

TITULO: **FUNCION BOOLEANA 4**

PADE: **1/1**

Haz clic sobre la imagen... y verás el resultado.



BIT
Binarío
Decimal
High
Hexadecimal
Nibble
Flancos
LOW

Capítulo I

GLOSARIO



1.1 Glosario Básico

A continuación se muestra la definición de la terminología básica a nivel del área de la Electrónica Digital, el cual se trabajará en este capítulo del libro.

▶ **BIT:** es la mínima unidad de almacenamiento y se representa por un "0" o un "1" lógico.

"0" es ausencia de señal

"1" es presencia de señal

▶ **NIBBLE:** es un conjunto conformado por 4 bits.

1011 = 1 nibble

▶ **BYTE:** es un conjunto conformado por 8 bits o 2 conjuntos conformados por 2 nibbles.

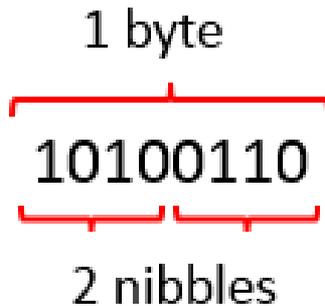


Figura 1. Un byte

▶ **PONDERACIÓN:** es determinar el peso de un elemento perteneciente a un conjunto.

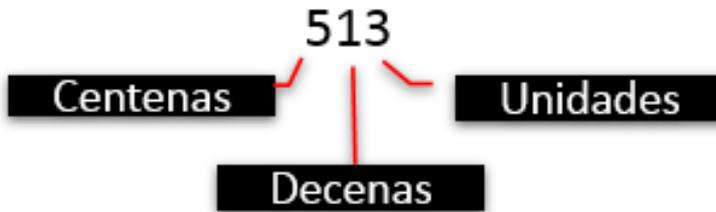


Figura 2. Ponderación

- ▶ **LSB - Bit Menos Significativo - Least Significant Bit:** es la posición del bit en un número binario que tiene la menor ponderación o valor (está situado en el extremo derecho).
- ▶ **MSB - Bit Más Significativo - Most Significant Bit:** es la posición del bit en un número binario que tiene la mayor ponderación o valor (está situado en el extremo izquierdo).

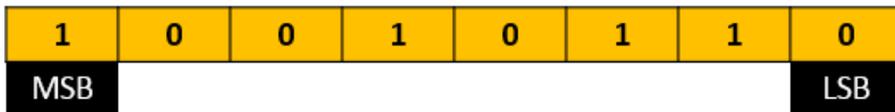


Figura 3. MSB - LSB de un byte

- ▶ **NIVEL LÓGICO BAJO:** nivel de voltaje mínimo, se toma como un "0" lógico, o sea ausencia de señal ("L" Low).

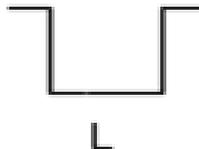


Figura 4. Nivel lógico bajo

- ▶ **NIVEL LÓGICO ALTO:** nivel de voltaje máximo, se toma como un "1" lógico, o sea presencia de señal ("H" High).

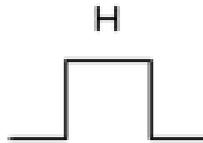


Figura 5. Nivel lógico alto

► **FLANCO POSITIVO:** es la transición entre el estado BAJO y el ALTO, también es llamado flanco de subida.

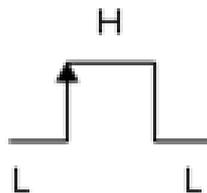


Figura 6. Flanco positivo

► **FLANCO NEGATIVO:** es la transición entre el estado ALTO y el BAJO, también es llamado flanco de caída.

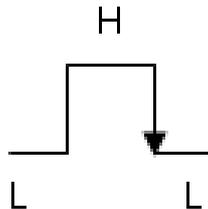


Figura 7. Flanco negativo

► **BCD Decimal Codificado en Binario – Binary Coded Decimal:** es la representación de los números decimales (0 al 9) escritos en sistema numérico binario.

► **FAN OUT o Abanico de Salida:** es el factor de salida cuando a una de ellas se le conectan varias entradas de otras compuertas; en otras palabras, es la cantidad de compuertas lógicas que puede soportar una sola salida.

► **FAN IN o Abanico de Entrada:** es el factor de entrada para compuertas que tienen varias de ellas; en otras palabras, es la cantidad de entradas que puede tener una compuerta lógica simultáneamente.

► **Señal Análoga:** es una señal que puede tener infinitos valores, positivos y/o negativos

► **Señal Digital:** es una señal que sólo puede tener dos valores "1" o "0"

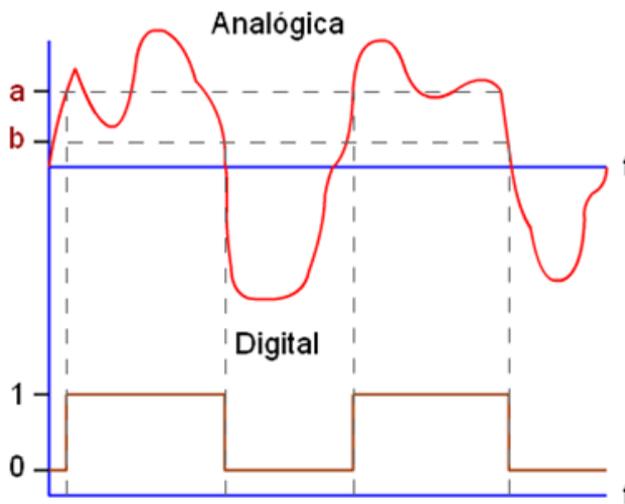


Figura 8. Señales análoga y digital

En el ejemplo de la figura 8, la señal digital toma el valor "1" cuando supera al valor **a**, y toma valor "0" cuando desciende por debajo del valor **b**. Cuando la señal permanece entre los valores **a** y **b**, se mantiene con el valor anterior.

► **RETARDO DE PROPAGACIÓN (t_p):** es el tiempo que tarda la salida de una compuerta en cambiar de estado después de que haya cambiado el estado de la entrada, estos tiempos se miden en ns (nano segundos, 10^{-9} s).

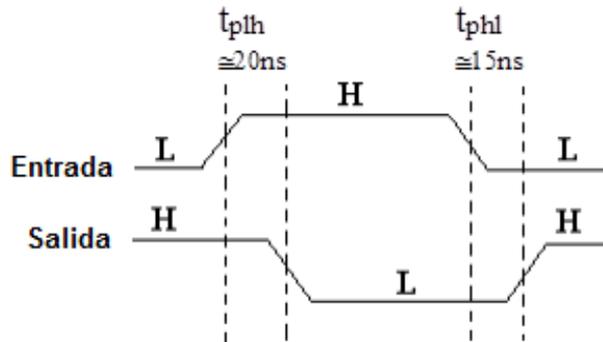
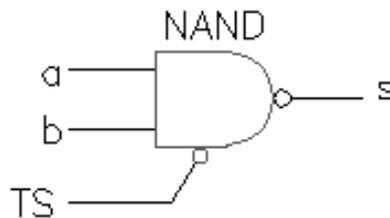


Figura 9. Señales de retardo de propagación

► **TTL:** Transistor Transistor Logic – Lógica Transistor Transistor.

► **CMOS:** Complementary Metal Oxide Semiconductor – Semiconductor Complementario de Óxido Metálico.

► **Lógica TRI-STATE:** es una lógica que posee una terminal adicional en las compuertas para poder habilitar o deshabilitar la utilización de dicha compuerta. La terminal TRI-STATE se une en paralelo con la salida de la compuerta y de esta forma seleccionar cual compuerta utilizar o habilitar. Las compuertas no habilitadas poseen en su salida una alta impedancia (Hi-Z).



TS activo en BAJO

Figura 10. Terminal TRI-STATE activa en bajo

La compuerta se habilita cuando TS está en un nivel bajo "0", observar que la entrada TS tiene un círculo, el resto del comportamiento cuando está habilitada es el de una compuerta NAND común.

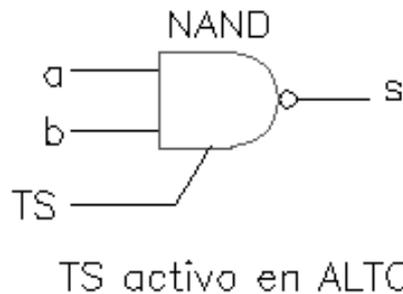


Figura 11. Terminal TRI-STATE activa en alto

La compuerta se habilita cuando TS está en un nivel alto "1", observar que la entrada TS no tiene el círculo, el resto del comportamiento cuando está habilitada es el de una compuerta NAND común.

1.2 Objeto Interactivo de Aprendizaje - OIA

En el siguiente **O**bjeto **I**nteractivo de **A**prendizaje ("**OIA**"), podrá autoevaluar el aprendizaje adquirido con respecto al glosario básico del área de la Electrónica Digital.

!Animate! a participar en esta actividad de refuerzo formativo.

El objetivo de esta actividad, es encontrar 10 palabras pertenecientes al glosario básico del área de la Electrónica Digital.

!Vamos!

Dando clic en la esquina superior derecha se expande el OIA.



SOPA DE LETRAS

T	E	Z	I	S	N	X	M	E	Q	S	L	D	A	E
V	T	E	B	I	V	S	M	I	Ñ	C	P	B	Y	O
T	W	R	Y	B	G	K	D	F	E	R	S	O	F	V
N	O	I	C	A	R	E	D	N	O	P	B	F	P	I
V	R	K	V	W	V	R	O	Q	M	E	I	O	K	T
F	T	R	K	I	U	D	E	Y	Ñ	W	Y	M	N	A
V	D	S	M	B	Y	T	E	G	I	G	B	L	I	G
U	O	F	Ñ	B	S	L	K	F	L	O	B	X	B	E
C	S	R	C	B	A	J	O	A	F	R	S	J	B	N
U	R	G	P	P	P	G	W	Y	Y	S	M	P	L	O
Y	N	H	C	A	L	T	O	U	G	W	M	I	E	C
F	L	A	N	C	O	P	O	S	I	T	I	V	O	N
V	Z	D	J	L	V	E	R	M	R	I	C	Ñ	G	A
Z	P	N	C	M	X	X	U	I	S	K	A	Y	P	L
S	I	U	C	T	I	B	Q	Q	X	B	X	H	Y	F

Halla 10 GLOSARIOS BÁSICOS
Pueden estar en dirección horizontal
o vertical, al derecho o al revés.

Muestra palabras

Haz clic en la primera letra de la palabra,
luego dirige el ratón a la última letra y
vuelve a hacer clic. Palabra roja es
incorrecta, palabra verde es correcta.

Familia lógica
CMOS

Familia lógica
TTL

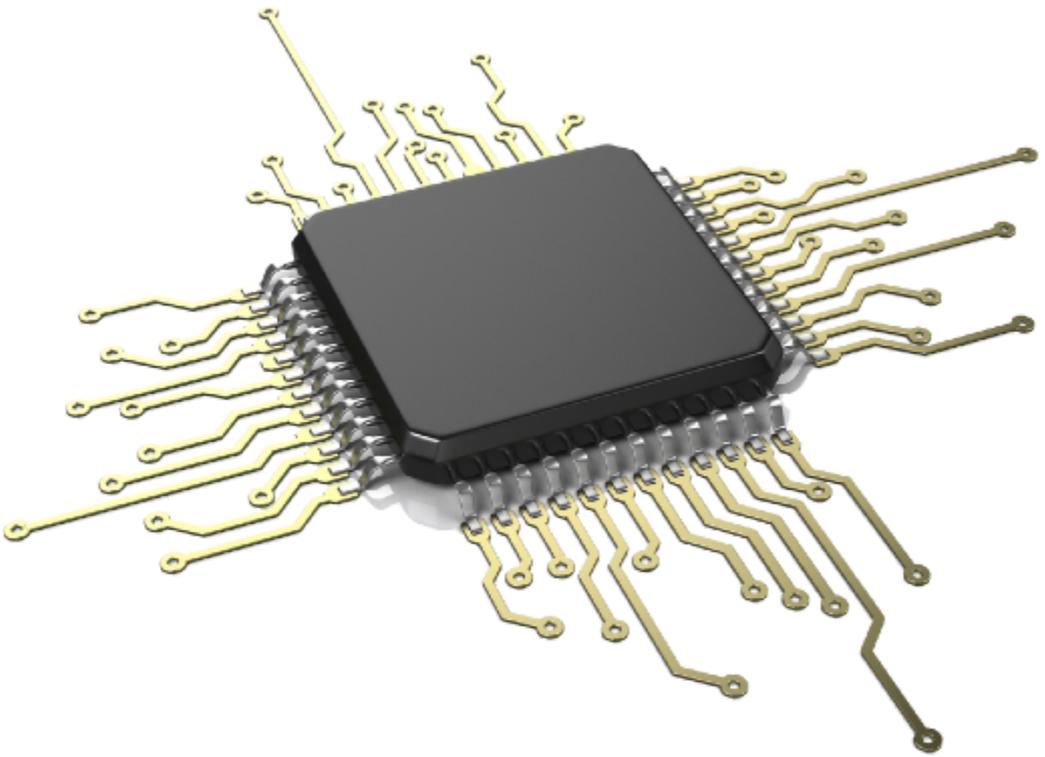
Características

¿Que es?

Familias lógicas
de circuitos
integrados

Capítulo II

FAMILIAS LÓGICAS DE CIRCUITOS INTEGRADOS



2.1 Familias Lógicas

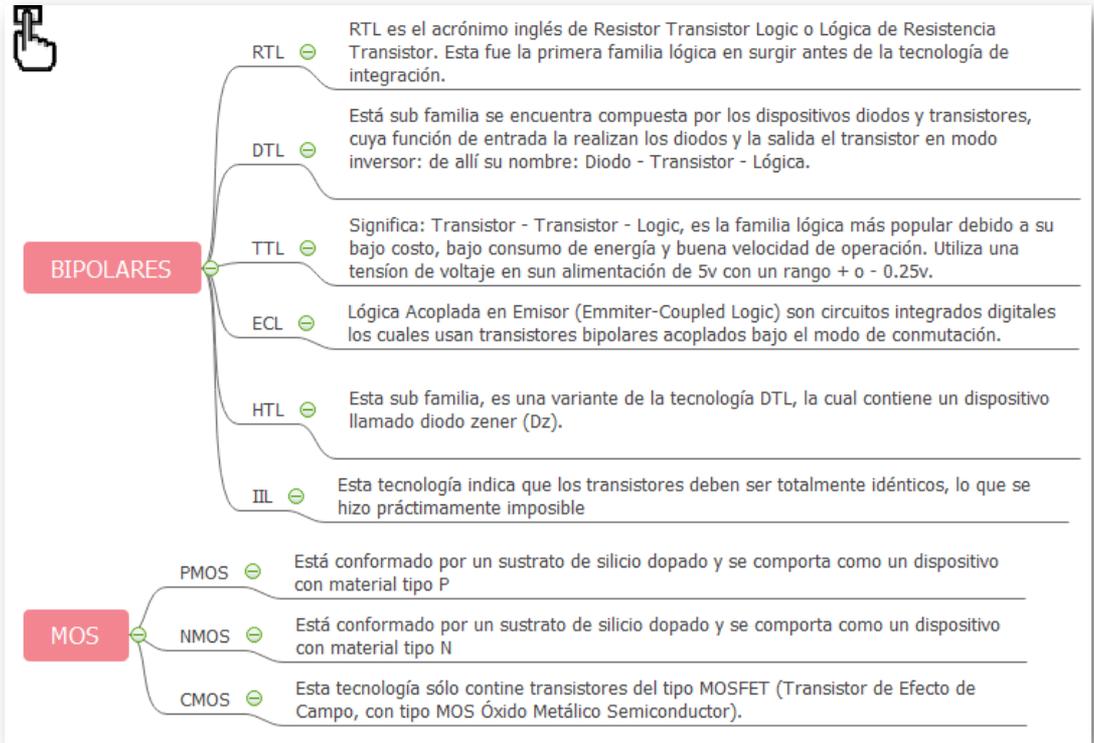


Figura 12. Familias Lógicas

2.2 Familias lógicas TTL y CMOS

En la siguiente figura se procede a realizar una comparación de las características principales entre las dos familias lógicas más importantes, ellas son: TTL y CMOS:



	TTL	CMOS
Nombre de la Familia	Lógica Transistor Transistor	Semiconductor de Óxido Metálico Complementario
Voltaje de Alimentación	4,75v y 5,25v	3v y 16v
Niveles lógicos de Entrada	Estado Bajo = 0v a 0,8v Estado Alto = 2v a 5v	Estado Bajo = 0v a 1,5v Estado Alto = 3,5v a 5v
Niveles lógicos de Salida	Estado Bajo = 0v a 0,4v Estado Alto = 2,4v a 5v	Estado Bajo = 0v a 0,01v Estado Alto = 4,9v a 5v
Series de Referencias Comerciales	74xxxx excepto 74C y 74HC	40xxxx, 45xxxx, 47xxxx, 74Cxxxx, 74HCxxxx
Series de Referencias Militares	54xxxx excepto 54C y 54HC	74C y 74HC
Asume las terminales de entrada desconectadas como	Alto = High (H) = "1"	Bajo = Low (L) = "0"
Susceptibles a la Electricidad Estática	No	Si, pueden dañarse cuando se manipulan debido a la electricidad estática generada por el cuerpo humano
Fan Out (típico)	10	50
Velocidad de Transmisión	Es mayor, hasta 35Mhz	Es menor, hasta 10Mhz
Consumo de Potencia	Mayor debido a la mayor velocidad de transmisión	Menor debido a menor velocidad de transmisión
Degradación aceptable de la señal de Salida	Hasta 1 metro	Varios metros
Costo	Menor	Mayor

Figura 13. Características de las familias lógicas TTL y CMOS

Comparación de los estados o niveles entre estas dos familias operando con 5v de alimentación, la zona blanca que se observa en la parte media, se le llama zona de incertidumbre, debido a que el diapositivo no detecta si es un "1" o un "0" lógico.

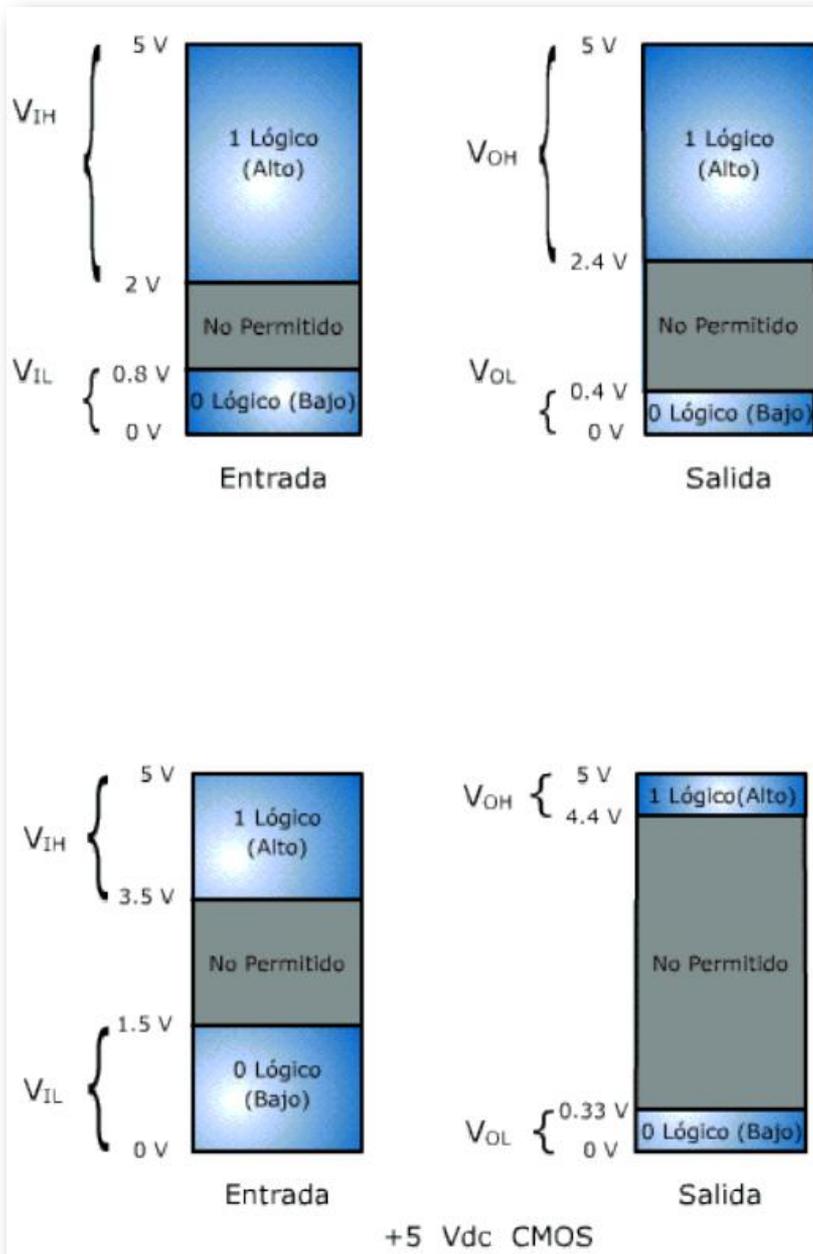
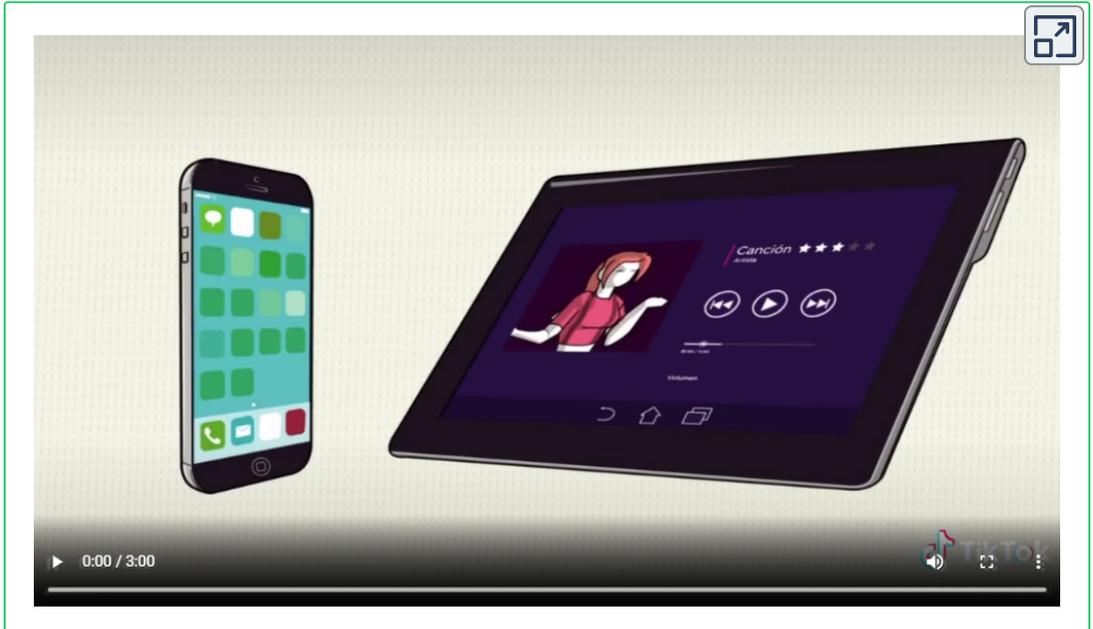


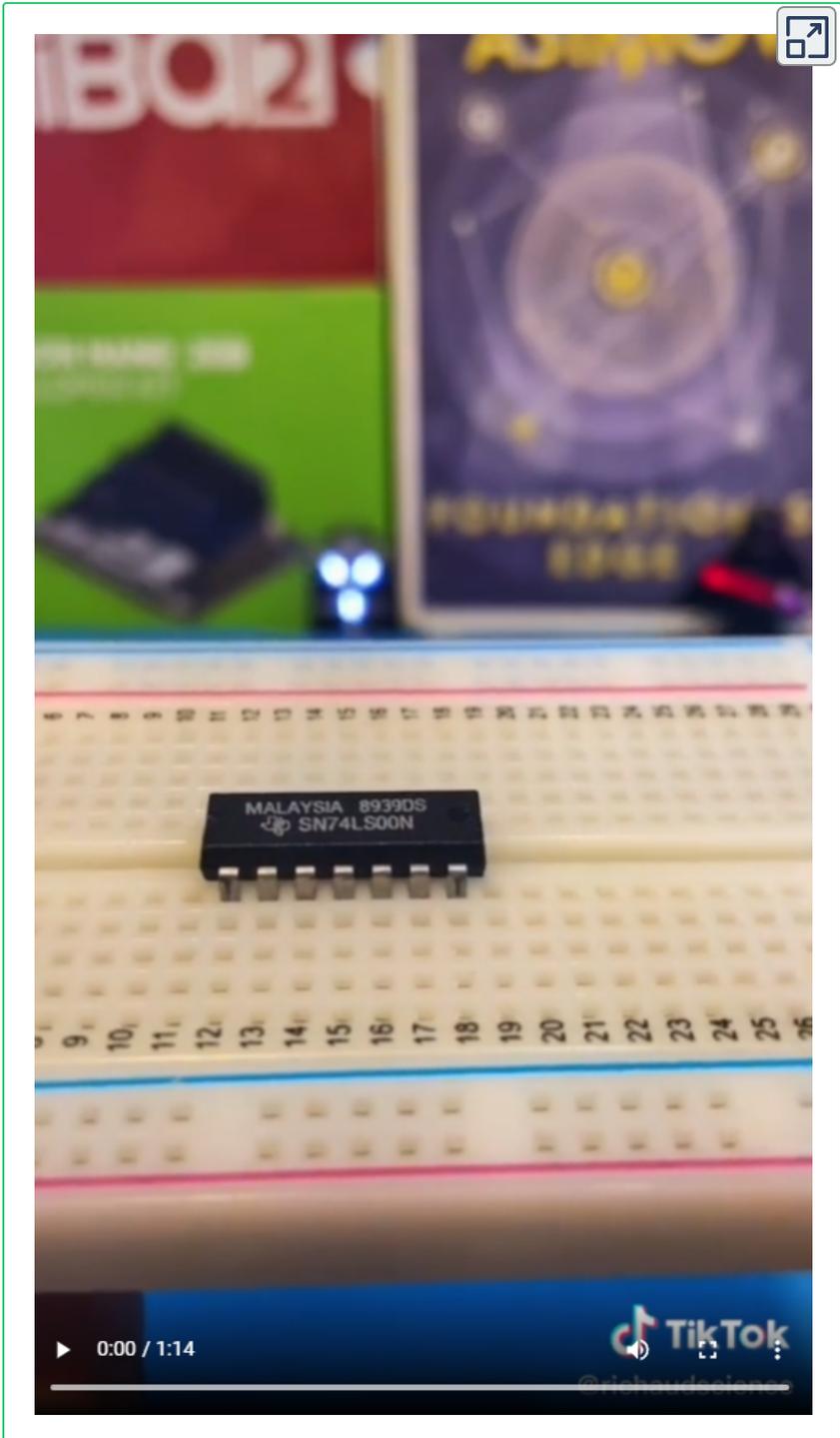
Figura 14. Comparación de los niveles de voltaje entre las familias TTL y CMOS

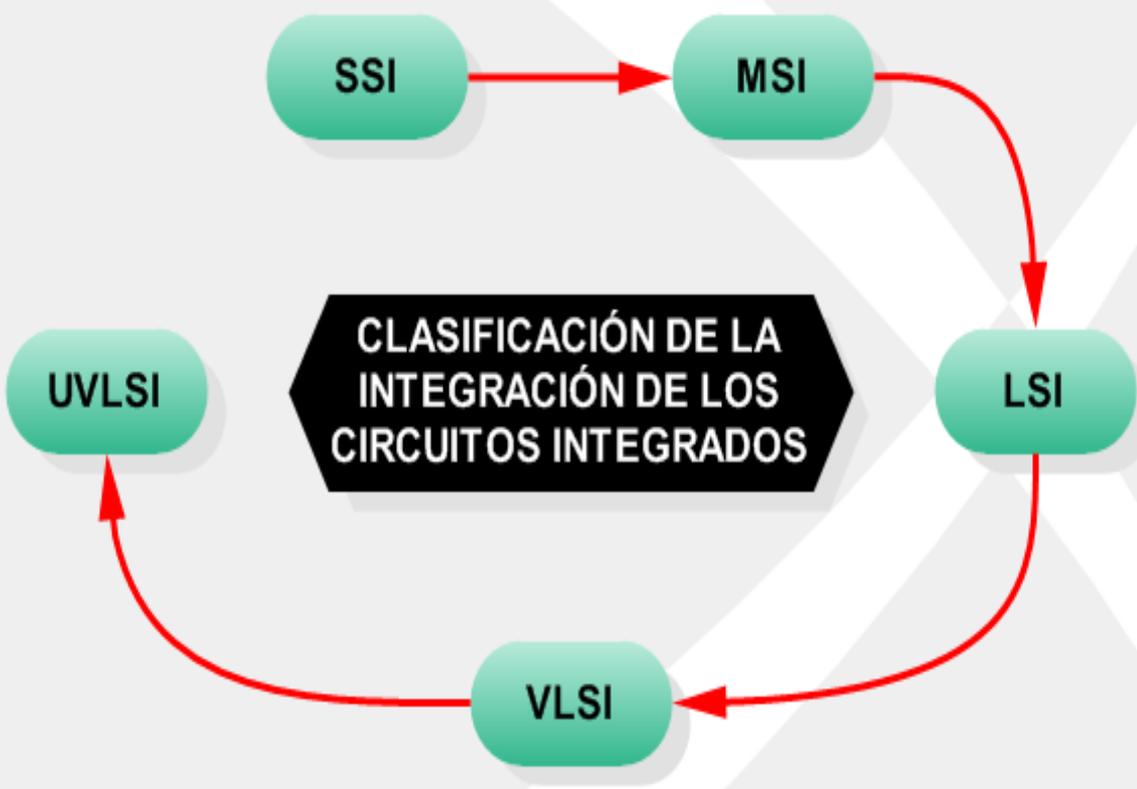
El siguiente video publicado en [TikTok](#), en el canal [CURSOS.GANT](#), se describen familias de circuitos digitales:

Vídeo:



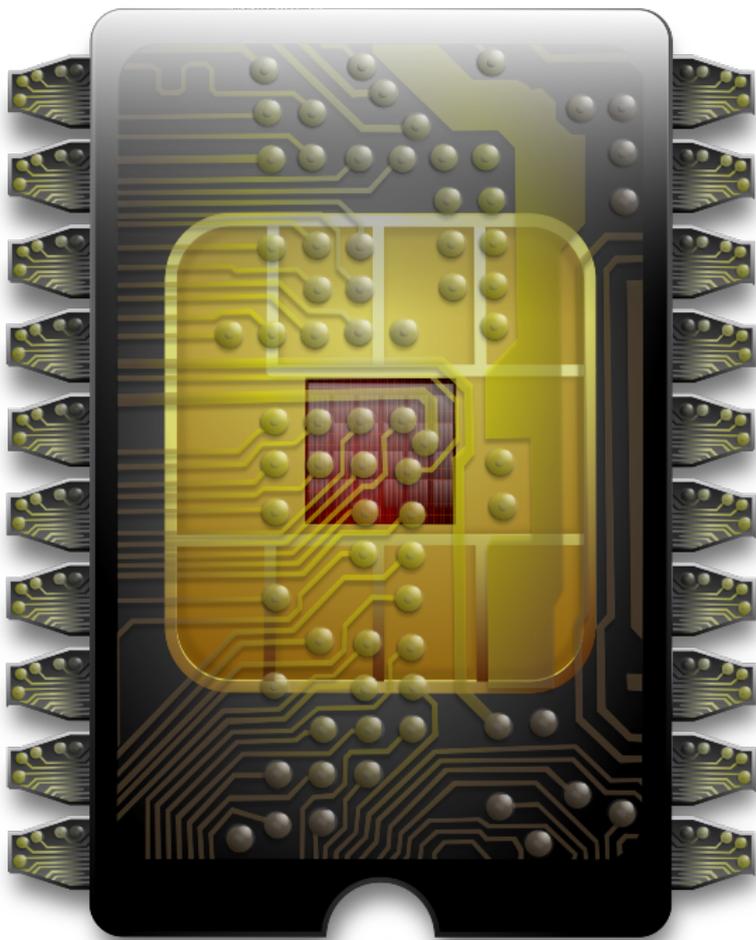
En la siguiente página, hay otro video publicado en [TikTok](#), en el canal [LoFi\(860862\) - skollbeats](#) que, también, describe familias de circuitos digitales.





Capítulo III

CLASIFICACIÓN DE LA INTEGRACIÓN DE LOS CIRCUITOS INTEGRADOS



3.1 Clasificación

La clasificación del nivel de integración de los circuitos integrados depende de la cantidad de compuertas lógicas que contenga.

- ▶ **SSI (SMALL SCALE INTEGRATION, INTEGRACIÓN A PEQUEÑA ESCALA):** contiene hasta 10 compuertas lógicas.
- ▶ **SMSI (MEDIUM SCALE INTEGRATION, INTEGRACIÓN A MEDIANA ESCALA):** contiene entre 10 y 100 compuertas lógicas.
- ▶ **LSI (LARGE SCALE INTEGRATION, INTEGRACIÓN A GRAN ESCALA):** contiene entre 100 y 10.000 compuertas lógicas.
- ▶ **VLSI (VERY LARGE SCALE INTEGRATION, INTEGRACIÓN A MUY ALTA ESCALA):** contiene 10.000 y 100.000 compuertas lógicas.
- ▶ **UVLSI (ULTRA VERY LARGE SCALE INTEGRATION, ULTRA ALTA ESCALA DE INTEGRACIÓN):** contiene más de 100.000 compuertas lógicas.

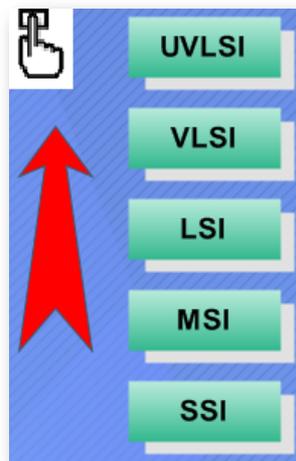


Figura 15. Clasificación de los circuitos integrados (IC)

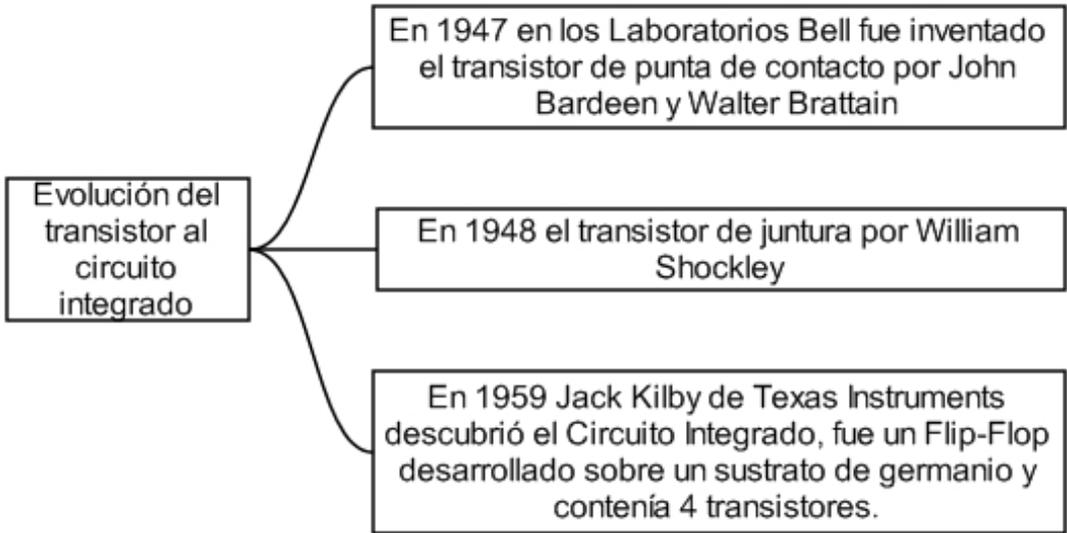


Figura 16. Evolución del transistor al circuito integrado

3.2 Personajes Históricos en la Electrónica

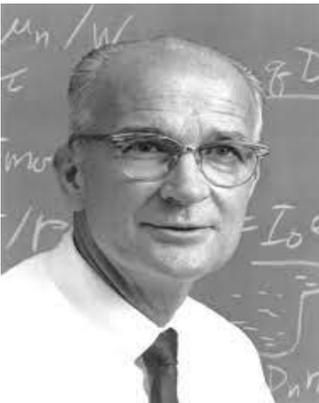


John Bardeen (1908-1991) Ingeniero Eléctrico y Físico estadounidense. Licenciado en física y matemáticas. En 1945 empezó a trabajar en los Laboratorios Bell de Nueva Jersey realizando investigaciones científicas alrededor de los semiconductores. En 1956 fue galardonado con el Premio Nobel de Física por su investigación en semiconductores y por el descubrimiento del efecto transistor. La primera aplicación se realizó en los audífonos.



Walter Houser Brattain (1902-1987) Físico chino. Terminada la guerra volvió a los laboratorios Bell y, trabajando en colaboración con otros eminentes físicos, prosiguió investigando en la línea anterior, centrandose principalmente su estudio en el silicio y el germanio. En 1956 compartió el Premio Nobel de Física con los también

físicos norteamericanos John Bardeen y William Shockley por el diseño del transistor de germanio, ingenio cuyo posterior desarrollo sería la base de los modernos microprocesadores y del formidable despegue de la informática. Autor de numerosos artículos sobre física del estado sólido, Walter Houser Brattain registró diversas patentes a lo largo de su vida.



William Bradford Shockley (1910-1989) Físico norteamericano. Doctorado por la Universidad de California, en 1953 fue nombrado director del departamento de transistores de la empresa Bell Telephone Company. Shockley colaboró con John Bardeen y Walter Houser Brattain en la construcción de dispositivos

semiconductores que reemplazarán a los tubos de vacío.

En la figura a continuación se encuentran ubicados de izquierda a derecha los siguientes personajes: John Bardeen, William Bradford Shockley y Walter Houser Brattain.



Figura 17. Personajes históricos en la invención de la electrónica

Fuente: <https://www.biografiasyvidas.com/biografia/b/fotos/brattain.jpg>

NOTAS COMPLEMENTARIAS:

El transistor fue inventado en los Laboratorios Bell de Estados Unidos el 16 diciembre de 1947 por John Bardeen, Walter Houser Brattain y William Bradford Shockley, recibiendo el nobel de física en 1956 por esta investigación..



Jack St. Clair Kilby (1923-2005) ingeniero eléctrico y físico estadounidense. Inventor del circuito integrado. En su carrera, patentó más de 60 inventos. A partir de 1958 fue empleado de la compañía de semiconductores estadounidense Texas Instruments, donde desarrolló el microchip en este mismo año. En el 2000 recibió el

Premio Nobel de Física.



William Bradford Shockley



John Bardeen



Walter Houser Brattain

Figura 18. Premios Nobel de Física en 1956

Fuente: <https://co.pinterest.com/pin/317011261244004399/>

3.3 Objeto Interactivo de Aprendizaje - OIA

En este objeto interactivo de aprendizaje debe depositar en cada contenedor el personaje correspondiente.

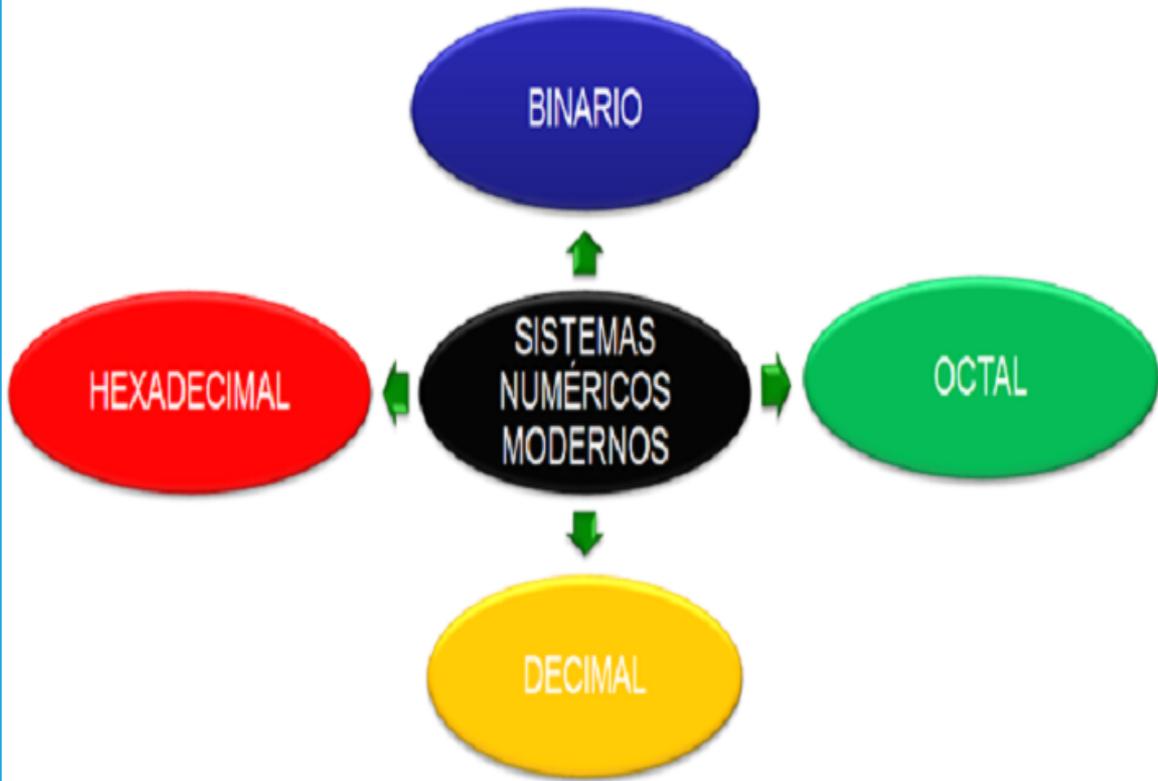
Dando clic en la esquina superior derecha se expande el OIA.

Coloca las imágenes en el contenedor correspondiente 

Bardeen	Shockley	Brattain	Kilby 
----------------	-----------------	-----------------	---







Capítulo IV

**SISTEMAS NUMÉRICOS
ANTIGUOS Y MODERNOS.
Conversiones con números
enteros y fraccionarios**

4.1 Sistemas de Numeración

A lo largo de la historia de la humanidad, el ser humano ha buscado diferentes maneras de representar cantidades. Si nos remontamos hacia más de dos mil años, los pueblos de aquella época no utilizaban números para contar objetos, sino que hacían uso de cualquier elemento que pudiera servirles para contar, ya sea utilizando sus propios dedos, dibujando símbolos, marcando bastones (ramas) o haciendo nudos en una cuerda, entre otros.

Ahora bien, el primer uso que se le dio a los números, se relaciona con la necesidad de ordenar elementos y no con la de contar o medir objetos.



Figura 19. Característica de los sistemas numéricos

A continuación, la explicación de los 6 sistemas de numeración más característicos de la historia, reconociendo sus elementos principales y los símbolos que utilizaron para representar las cantidades establecidas.

4.2 Sistemas Numéricos Antiguos



Figura 20. Sistemas numéricos antiguos

▪ El sistema de numeración Egipcio

Los Egipcios han dejado un gran legado cultural, por sus monumentales construcciones, conocimientos y descubrimientos en agricultura, arte y matemáticas.

- El sistema numérico es no posicional.
- Utilizado en medición de construcciones.
- Contaban de 10 en 10.
- No tenían representación para el número cero (0).
- Utilizaban jeroglíficos para agrupar las unidades.

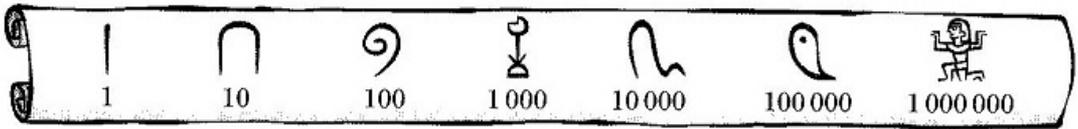


Figura 21. Caracteres de la numeración Egipcia

► Ejemplo:



Figura 22. Valor 13745

$$1 \times 10000 + 3 \times 1000 + 7 \times 100 + 4 \times 10 + 5$$

$$10000 + 3000 + 700 + 40 + 5 = 13745$$

▪ El sistema de numeración Chino

La cultura china es una de las más completas y antiguas de la humanidad. Su legado perdura hasta la actualidad, ya que han sido gestores de grandes descubrimientos, realizando aportes importantes para la humanidad.

- El sistema numérico es posicional.
- Utiliza 9 símbolos y las potencias de 10.
- No tenían representación para el número cero (0).



Figura 23. Caracteres de la numeración China

► Ejemplo:

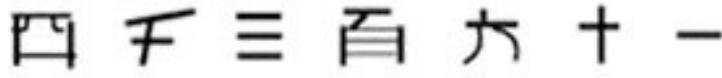

$$4 \times 1000 + 3 \times 100 + 6 \times 10 + 1 = 4361$$

Figura 24. Valor 4361

■ Los sistemas de numeraciones Griegos (Ático y Jónico)

- El sistema numérico ático o acrofónico es el más antiguo y es no posicional.
- Los números 50, 500 y 5000, se obtenían agregando el signo de 10, 100 o 1000 a la representación del número 5.



Figura 25. Caracteres de la numeración Griega más antigua (ático o acrofónico)

► Ejemplo:


$$3 \times 1000 + 1 \times 500 + 2 \times 100 + 3 \times 10 + 1 \times 5 + 2 \times 1$$
$$3000 + 500 + 200 + 30 + 5 + 2 = 3737$$

Figura 26. Valor 3737

- El sistema numérico Jónico es más nuevo y es posicional.

- Utilizaron letras del alfabeto griego para representar cantidades y fue llamado Jónico.

Este sistema utilizaba las 24 letras del alfabeto griego, junto con algunos otros símbolos, tal como muestra la siguiente figura:

α	β	γ	δ	ϵ	ζ	η	θ	
1	2	3	4	5	6	7	8	9
ι	κ	λ	μ	ν	ξ	\omicron	π	ρ
10	20	30	40	50	60	70	80	90
σ	τ	υ	ϕ	χ	ψ	ω		
100	200	300	400	500	600	700	800	

Figura 27. Letras del alfabeto Griego para la numeración de los Jónicos

► **Ejemplo:**

$$200 + 40 + 2 = \sigma + \mu + \beta = 242$$

▪ **El sistema de numeración Romano**

Es el sistema de numeración que ha perdurado en el tiempo. Actualmente se utiliza para numerar capítulos o escenas de una obra de teatro, para designar el nombre de algunas autoridades (como emperadores, reyes y papas), para ordenar los contenidos de un índice y los tomos de una enciclopedia, entre otros.

- El sistema numérico es semiposicional.
- Se utilizan letras mayúsculas y reglas.

I	II	III	IV	V	
1	2	3	4	5	
VI	VII	VIII	IX		
6	7	8	9		
X	L	C	D	M	
10	50	100	500	1000	

Figura 28. Caracteres de la numeración Romana

► Ejemplos:

$$LXV = 50 + 10 + 5 = 65$$

$$DCLX = 500 + 100 + 50 + 10 = 660$$

▪ El sistema de numeración Maya

- El sistema numérico es posicional.
- Sistema en base 20.
- Utilizaban rayas con valor de 5 y puntos con valor de 1.
- Simbolizaron el cero con un caracol.
- Las cifras se escriben verticalmente, con las unidades en la parte inferior.
- Se agrupan símbolos hasta el número 19.

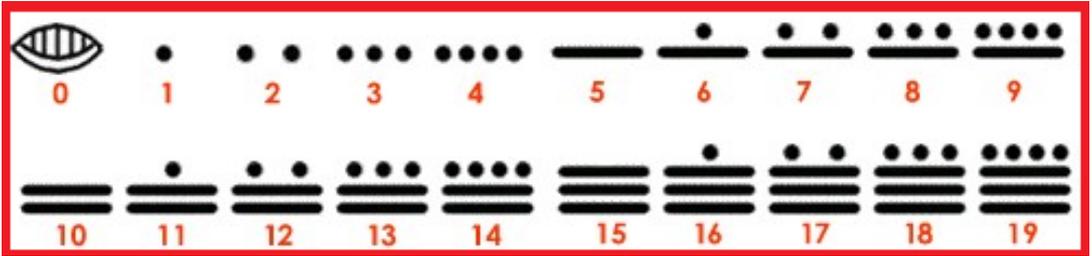


Figura 29. Caracteres de la numeración Maya

Analizando los símbolos que se presentan, se puede observar que el número 14 está formado por 2 rayas y 4 puntos. Como las rayas representan al 5 y los puntos al 1, se multiplica 2×5 y 4×1 , obteniendo un total de $10 + 4 = 14$.

► **Ejemplo:**

Para escribir números iguales o superiores al número 20, las cifras adquirirían un valor que dependía de la posición en donde se encontraban, disponiéndose en columnas y asignándole un valor de abajo hacia arriba, en el que hay que multiplicar el valor de cada cifra por 1, 20, 20×20 , $20 \times 20 \times 20 \dots$ según el lugar que ocupe, como se observa a continuación:

Tercer Orden	$20 \times 20 = 400$		$7 \times 400 = 2800$
Segundo Orden	20		$7 \times 20 = 140$
Primer Orden	1		<u>2 +</u>
			2942

Figura 30. Valor 2942

■ **El sistema de numeración Inca**

- Sistema de numeración decimal de carácter posicional.

- Utilizaron el "Quipu" que poseía cuerdas y mediante la realización de nudos de variados colores y tamaños, les permitió registrar la información numérica

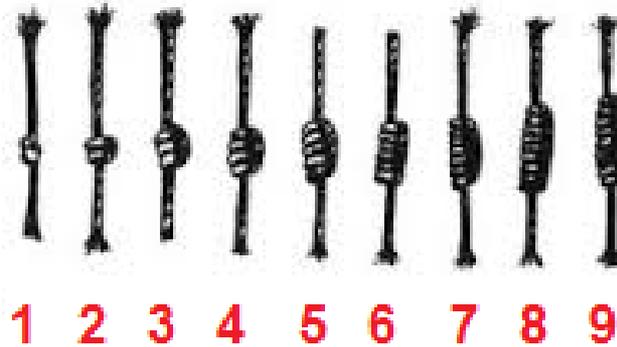


Figura 31. Quipu

► Ejemplo:

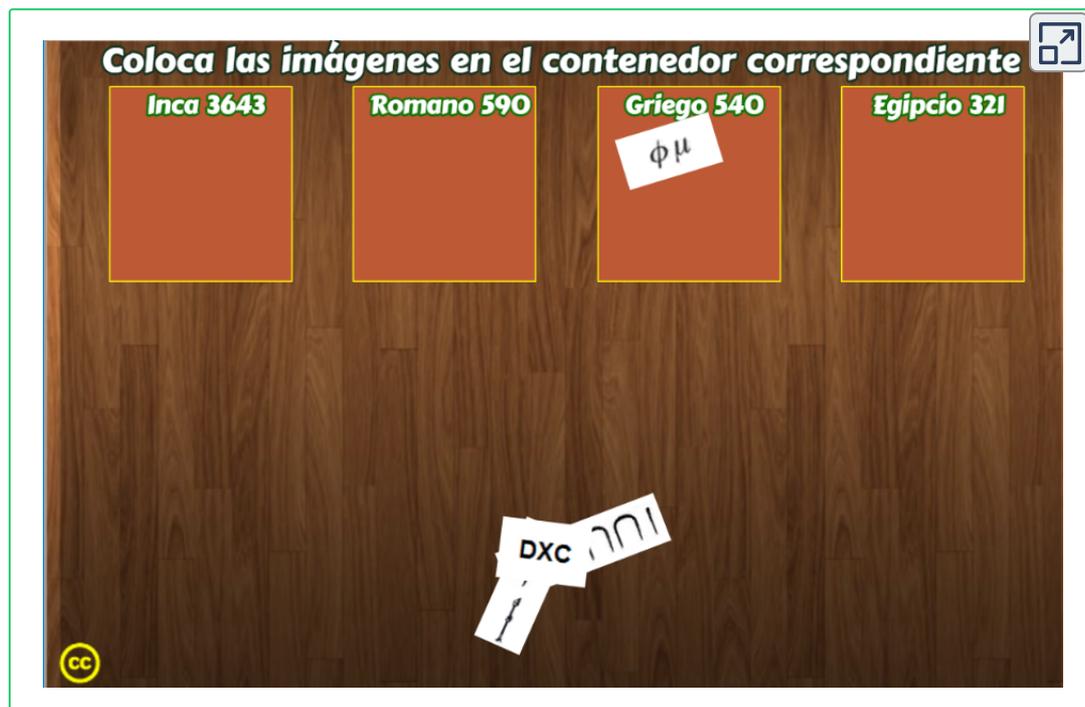


Figura 32. Valor 4643

4.3 Objetos Interactivos de Aprendizaje - OIA

En este OIA, arrastre cada Sistema Numérico Antiguo al contenedor correspondiente.

Dando clic en la esquina superior derecha se expande el OIA.



En la siguiente página, se presenta una unidad interactiva del proyecto [@prende.mx](https://prende.mx)¹, en la cual se puede explorar y aprender sobre las similitudes y diferencias entre el sistema decimal de numeración y el sistema maya.

¹ El proyecto [@prende.mx](https://prende.mx) es una iniciativa de la Secretaría de Educación Pública del gobierno mejicano a través de la cual busca la introducción de las TIC en la Educación.



4.4 Sistemas Numéricos Modernos en la Informática

Un sistema numérico es un conjunto de símbolos y reglas que se utilizan para representar cantidades.

Un número está constituido por una sucesión de dígitos situados ordenadamente a izquierda y a derecha de un punto de referencia (en los países anglosajones se utiliza un punto, mientras que en los latinos se utiliza una coma).



Figura 33. Sistemas Numéricos Modernos en la Informática

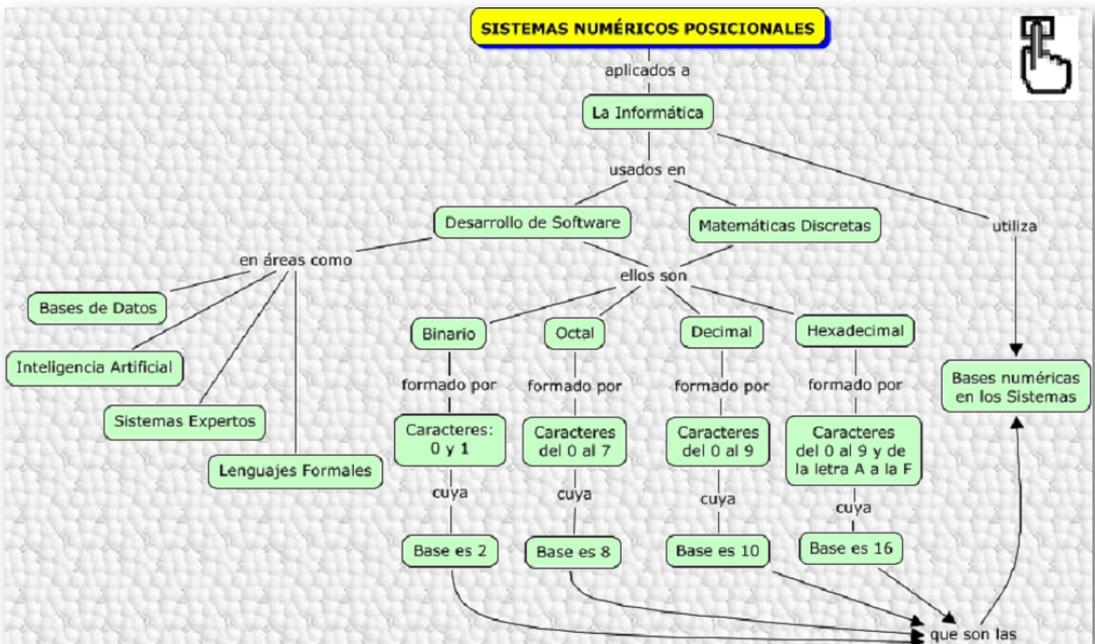


Figura 34. Sistemas Numéricos Posicionales aplicados en la informática

- Los posicionales: muestran que el valor del dígito depende de la posición que ocupe y del símbolo que se utilice (binario, octal, decimal y hexadecimal).
- Los semi posicionales: son sistemas numéricos que poseen símbolos con el mismo valor independiente en la posición que se encuentre (romano).
- Los no posicionales: son sistemas numéricos donde los dígitos solo tienen el valor del símbolo utilizado (egipcio, griego).

Los sistemas numéricos más utilizados son: decimal, octal y hexadecimal y la principal característica de estos sistemas es que utilizan una base de dígitos, es decir un sistema básico. Así:

Base	Sistema	Dígitos
2	Binario	0,1
8	Octal	0,1,2,3,4,5,6,7
10	Decimal	0,1,2,3,4,5,6,7,8,9
16	Hexadecimal	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Figura 35. Sistemas numéricos y sus caracteres

Para definir los números y sus bases, se escribirá el numero y se pondrá la base como subíndice.

- $1101_2 = 1101_b = 1101$ base binaria.
- $67_8 = 67_o = 67$ base octal.
- $42_{10} = 42_d = 42$ base decimal.
- $5FA_{16} = 5FA_h = 5FA$ base hexadecimal.

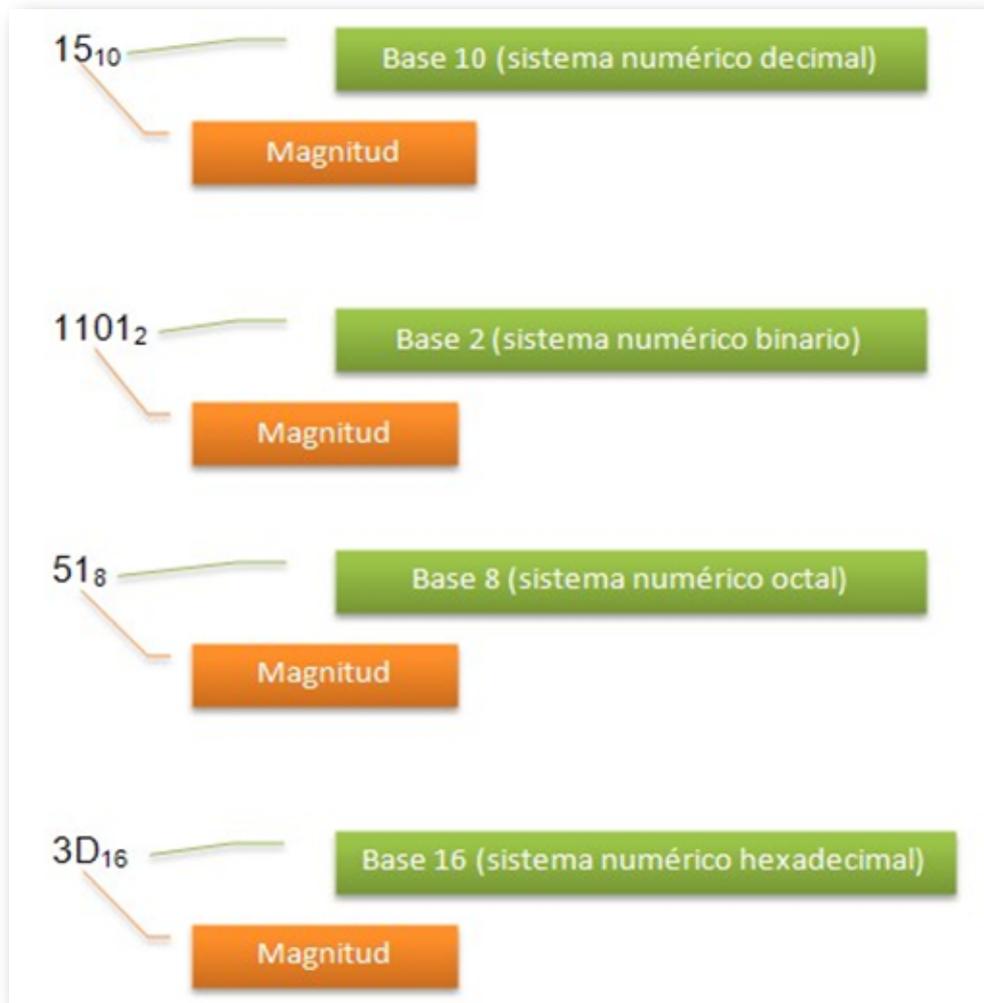


Figura 36. Sistemas numéricos y sus bases

Al hablar de sistemas numéricos es necesario tratar el tema de las conversiones existentes entre unos y otros sistemas de numeración, dichos procesos son necesarios en las aplicaciones tecnológicas de electrónica y computación. La equivalencia entre un sistema y otro se logra realizando operaciones aritméticas básicas.

4.5 Conversiones entre Sistemas Numéricos

4.5.1 Conversión de Decimal a Binario

Para la conversión de la parte entera, se va dividiendo la cantidad decimal por 2 (divisiones sucesivas), tantas veces como sea posible, hasta que llegue a un cociente de 1. Luego, se toma el último "Cociente" y todos los "Residuos" comenzando con el último (bit más significativo - MSB) hasta llegar al primero (bit menos significativo - LSB).

► Ejemplo:

• 27,375_d a binario

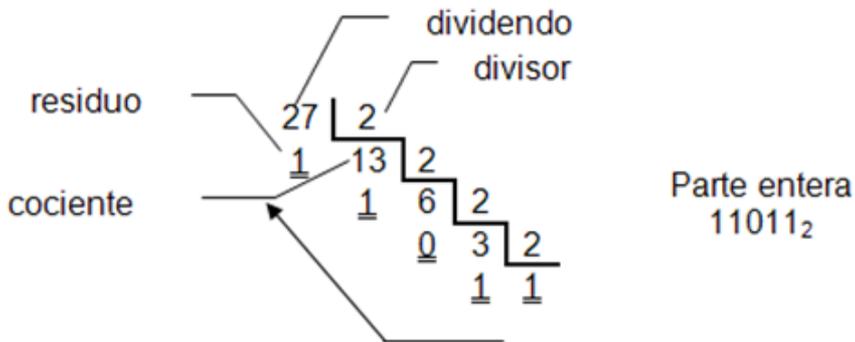


Figura 37. Conversión de Decimal a Binario de la parte entera

O también, así:

$$27 \div 2 = 13 \text{ y sobra } \mathbf{1}$$

$$13 \div 2 = 6 \text{ y sobra } \mathbf{1}$$

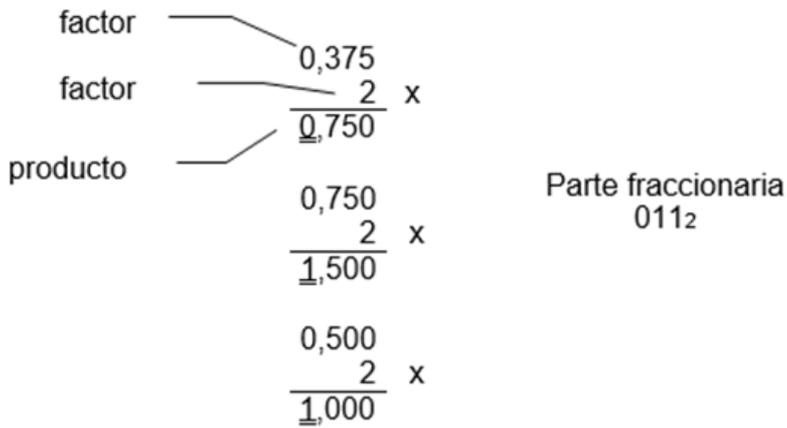
$$6 \div 2 = 3 \text{ y sobra } \mathbf{0}$$

$$3 \div 2 = 1 \text{ y sobra } \mathbf{1}$$

$\mathbf{1}$

$$\text{Parte entera} = \mathbf{11011}$$

Para la conversión de la parte fraccionaria se multiplica por la base (2) sucesivamente hasta la cantidad de dígitos que se requieran, se toman para la respuesta los números enteros de cada producto y lo que sobra lo sigue multiplicando por 2.



$$27,375_{10} \text{ a binario} = 11011,011_2$$

Figura 38. Conversión de Decimal a Binario de la parte fraccionaria

$0,375 \times 2 = 0$ y sobra 0,75

$0,75 \times 2 = 1$ y sobra 0,5

$0,5 \times 2 = 1$ y sobra 0

Parte fraccionaria = **011**

R/. $27,375_d = 11011,011_b$

► **Ejercicios:**

Los ejercicios con respuesta tiene como finalidad practicar la conversión de Decimal a Binario.

• $91,12_d$ a binario = R/. $1011011,000111_b$

• $146,56_d$ a binario = R/. $10010010,10001111_b$

El siguiente video complementa su proceso de formación:

Vídeo:



Video realizado por Rafael Nuila, titulado "Convertir decimal a binario", licencia de atribución de Creative Commons.

En el siguiente Objeto Interactivo se aprecia el procedimiento de conversión de decimal a binario:

An interactive object with a green border. It shows the conversion of the decimal number 24 to binary using a ladder method. The steps are: 24 divided by 2 gives 12 with remainder 0; 12 divided by 2 gives 6 with remainder 0; 6 divided by 2 gives 3 with remainder 0; 3 divided by 2 gives 1 with remainder 1; 1 divided by 2 gives 0 with remainder 1. A red arrow points from the final remainders (1, 1, 0, 0, 0) up to the initial number (24). Below the ladder, the result is shown as $(24)_{10} = (11000)_2$. At the bottom, there are two buttons: "Con parte fraccionaria" (highlighted in blue) and "Otro ejemplo" (greyed out). A share icon is in the top right corner.

$$\begin{array}{r} 0,625 \\ \underline{8} \times \\ \hline \underline{5},000 \\ 0,000 \end{array}$$

Parte fraccionaria
5₈

$$310,625_{10} \text{ a octal} = 466,5_8$$

Figura 40. Conversión de Decimal a Octal de la parte fraccionaria

0,625 x 8 = 5 y sobra 0

0

Parte fraccionaria = 5

R/. 310,625_d = 466,5_o

► Ejercicios:

Los ejercicios con respuesta tiene como finalidad practicar la conversión de Decimal a Binario.

- 84,15_d a octal = R/. 124,114_o
- 125,3_d a octal = R/. 175,231463_o

Vídeo:



Video realizado por Rafael Nuila, titulado "Convertir decimal a octal", licencia de atribución de Creative Commons.

4.5.3 Conversión de Decimal a Hexadecimal

Se denomina **Hexadecimal**, porque "**Hexa**" hace alusión a las 6 letras que utiliza este sistema numérico y "**Decimal**" porque usa los caracteres del 0 al 9 (base decimal).

Para la conversión de la parte entera, se va dividiendo la cantidad decimal por 16 (divisiones sucesivas), tantas veces como sea posible, hasta que llegue a un cociente menor de 16. Luego, se toma el último "Cociente" y todos los "Residuos" comenzando con el último hasta llegar al primero.

► **Ejemplo:**

• **158,25_d a hexadecimal**

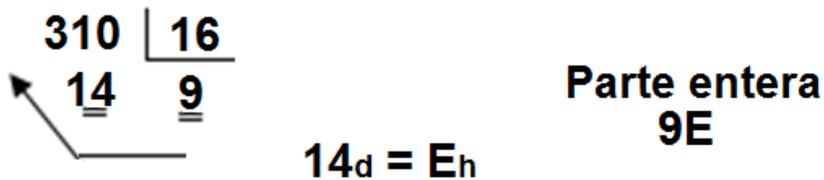


Figura 41. Conversión de Decimal a Hexadecimal de la parte entera

También se puede colocar de la siguiente forma:

$$158 \div 16 = 9 \text{ y sobra } 14_{\text{d}} = E_{\text{h}}$$

9

Parte entera = 9E

Para la conversión de la parte fraccionaria se multiplica por la base (16) sucesivamente hasta la cantidad de dígitos que se requieran, se toman para la respuesta los números enteros de cada producto y lo que sobra lo sigue multiplicando por 16.

$$\begin{array}{r}
 0,25 \\
 \underline{16} \times \\
 4,0 \\
 \\
 0,000
 \end{array}$$

Parte fraccionaria
48

$$158,25_{10} \text{ a hexadecimal} = 9E,4_{16}$$

Figura 42. Conversión de Decimal a Hexadecimal de la parte fraccionaria

$$0,25 \times 16 = 4 \text{ y sobra } 0$$

0

$$\text{Parte fraccionaria} = 4$$

$$\text{R/. } 158,25_d = 9E,4_h$$

► Ejercicios:

Los ejercicios con respuesta tiene como finalidad practicar la conversión de Decimal a Hexadecimal.

- $92,15_d$ a hexadecimal = **R/. 5C,266_h**

- $101,8_d$ a hexadecimal = **R/. 65,CCC_h**

Vídeo:



Video realizado por Pasos por ingeniería, titulado "CONVERTIR de DECIMAL a HEXADECIMAL con PARTE FRACCIONARIA - Ejercicio #2", licencia de atribución de Creative Commons.

4.5.4 Conversión de Binario a Decimal

Para las conversiones de Binario, Octal y Hexadecimal al sistema numérico Decimal, se utilizará el "método de la suma de pesos".

Para realizar la conversión de binario a decimal, tome los dígitos de la parte entera desde el lado derecho y multiplique cada bit por 2 elevado a la potencia consecutiva (comenzando con 2^0), a esto se le denomina técnicamente "**ponderar**"; después de ello realice la suma de todos los resultados y el número resultante será el equivalente al sistema decimal.

Para la parte fraccionaria, tome los dígitos de izquierda a derecha con exponentes negativos comenzando con 2^{-1} y multiplique por cada bit que está en esa posición y finalmente sume los resultados de la parte fraccionaria.

$$\begin{aligned}
101,011_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\
&= 1 \times 4 + 0 \times 2 + 1 \times 1 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\
&= 4 + 1 + 1/4 + 1/8 \\
&= 4 + 1 + 0,25 + 0,125 \\
101,011_2 &= 5,375_{10}
\end{aligned}$$

Figura 43. Conversión de Binario a Decimal

$$\begin{aligned}
1101,101_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
&= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
&= 8 + 4 + 1 + 1/2 + 1/8 \\
&= 8 + 4 + 1 + 0,5 + 0,125 \\
1101,101_2 &= 13,625_{10}
\end{aligned}$$

Figura 44. Conversión de Binario a Decimal

Observe el siguiente video para que complemente su proceso de formación:

Vídeo:



Video realizado por Prof. Alejandro Martínez González, titulado "Conversión binario a decimal", licencia de atribución de Creative Commons.

4.5.5 Conversión de Octal a Decimal

El sistema octal recibe ese nombre porque trabaja en base 8 y tiene un total de ocho símbolos que van desde el 0 hasta el 7.

Para convertir de octal a decimal simplemente tienes que coger el número en octal de derecha a izquierda y asignar a cada uno la potencia en base ocho que le corresponde, siendo la primera de todas $8^0 = 1$.

► Ejemplos:

• 374,25_o a decimal

$$\text{Parte entera: } 374_o = 3 \times 8^2 + 7 \times 8^1 + 4 \times 8^0 = 192 + 56 + 4 = 252_d$$

$$\text{Parte fraccionaria: } 0,25_o = 2 \times 8^{-1} + 5 \times 8^{-2} = 0,25 + 0,078125 = 0,328125_d$$

$$\text{R/. } 374,25_o = 252,328125_d$$

• 406,57_o a decimal

$$\text{Parte entera: } 406_o = 4 \times 8^2 + 0 \times 8^1 + 6 \times 8^0 = 256 + 0 + 6 = 262_d$$

$$\text{Parte fraccionaria: } 0,57_o = 5 \times 8^{-1} + 7 \times 8^{-2} = 0,625 + 0,109375 = 0,734375_d$$

$$\text{R/. } 406,57_o = 262,734375_d$$

Vídeo:



Vídeo realizado por Profe Apolinar, titulado "SISTEMAS DIGITALES _ CONVERSIÓN DE NUMERO OCTAL A NUMERO DECIMAL", licencia de atribución de Creative Commons.

4.5.6 Conversión de Hexadecimal a Decimal

Multiplicamos el valor de posición de cada dígito por el hexadecimal correspondiente.

El resultado del número decimal equivalente se obtiene, sumando todos los productos obtenidos en el paso anterior.

► Ejemplos:

• $3AF, D_h$ a decimal

$$\text{Parte entera: } 3AF_h = 3 \times 16^2 + A \times 16^1 + F \times 16^0 = 768 + 160 + 15 = 943_d$$

$$\text{Parte fraccionaria: } 0, D_h = D \times 16^{-1} = 0,8125_d$$

$$\text{R/. } 3AF, DB_h = 943,8125_d$$

• ABC,37_h a decimal

Parte entera: $ABC_h = A \times 16^2 + B \times 16^1 + C \times 16^0 = 2560 + 176 + 12 = 2748_d$

Parte fraccionaria: $0,37_h = 3 \times 16^{-1} + 7 \times 16^{-2} = 0,1875 + 0,02734375 = 0,21484375_d$

R/: 406,57_h = 262,734375_d

Vídeo:



Video realizado por Profe Apolinar, titulado "SISTEMAS DIGITALES _ CONVERSIÓN DE NUMERO HEXADECIMAL A NUMERO DECIMAL", licencia de atribución de Creative Commons.

4.5.7 Conversión de Binario a Octal

Este método consiste en agrupar de a tres (3) bits. Para la parte entera se comienza de derecha a izquierda, o sea del bit menos significativo (LSB) hasta el más significativo (MSB), para la parte fraccionaria se procede a agrupar al contrario, o sea de izquierda a derecha. Cuando las cifras no están completas, o sea de a tres (3) bits se completan con ceros (0) a la izquierda para la parte entera y con ceros a la derecha para la parte fraccionaria.

Hay que recordar que el sistema numérico octal va desde el número 0 hasta el número 7, lo que abarca un total de 8 caracteres.

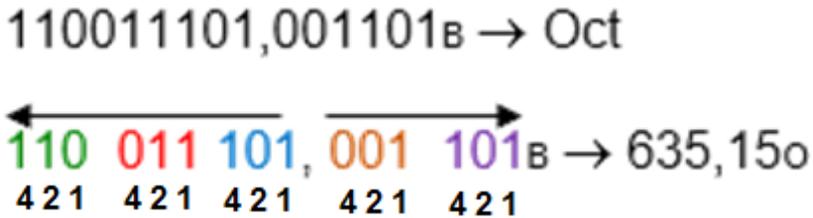


Figura 45. Conversión de Binario a Octal

► Ejemplos:

• 1001011011,1010_b a Octal

Parte entera: 001 001 011 011_b, agrupados de a 3 bits = 1 1 3 3_o

Parte fraccionaria: 101 000_b, agrupados de a 3 bits = 5 0_o

R/: 1001011011,1010_b = 1133,50_o

• 111010011,101011_b a Octal

Parte entera: 111 010 011_b, agrupados de a 3 bits = 7 2 3_o

Parte fraccionaria: 101 011_b, agrupados de a 3 bits = 5 3_o

R/: 111010011,101011_b = 723,53_o

4.5.8 Conversión de Octal a Binario

Cada caracter octal se representa en 3 bits, tanto para la parte entera como la fraccionaria.

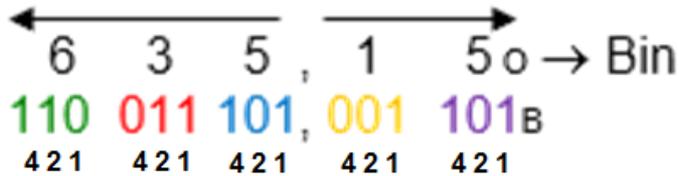


Figura 45. Conversión de Octal a Binario

► Ejemplos:

• 37,61_o a Binario

Parte entera: 37_o, representados en 3 bits cada caracter = 011 111_b

Parte fraccionaria: 61_o, representados en 3 bits cada caracter = 110 001_b

R/: 37,61_o = 011111,110001_b

• 152,731_o a Binario

Parte entera: 152_o, representados en 3 bits cada caracter = 001 101 010_b

Parte fraccionaria: 731_o, representados en 3 bits cada caracter = 111 011 001_b

R/: 152,731_o = 001101010,111011001_b

4.5.9 Conversión de Binario a Hexadecimal

Este método es similar a la conversión de Binario a Octal, pero agrupando de a cuatro (4) bits. Para la parte entera se comienza de derecha a izquierda, o sea del bit menos significativo (LSB) hasta el más significativo (MSB), para la parte fraccionaria se procede a agrupar al contrario, o sea de izquierda a derecha. Cuando las cifras no están completas, o sea de a cuatro (4) bits se completan con ceros (0) a la izquierda para la parte entera y con ceros a la derecha para la parte fraccionaria.

Hay que recordar que el sistema numérico hexadecimal consta de los números decimales del cero al nueve (0-9) hexa porque tiene 6 letras (A-F), lo que abarca un total de 16 caracteres.

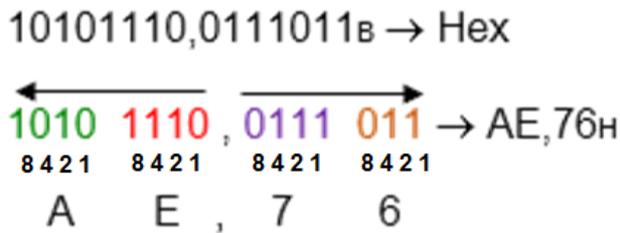


Figura 46. Conversión de Binario a Hexadecimal

► Ejemplos:

• 1101101111,101011_b a Hexadecimal

Parte entera: 0011 0110 1111_b, agrupados de a 4 bits = 36F_h

Parte fraccionaria: 1010 1100_b, agrupados de a 4 bits = AC_h

R/: 1011011011,101011_b = 36F,AC_h

• 111101011,1000101_b a Hexadecimal

Parte entera: 0001 1110 1011_b, agrupados de a 4 bits = 1 E B_h

Parte fraccionaria: 1000 1010_b, agrupados de a 4 bits = 8 A_h

R/: 111101011,1000101_b = 1EB,8A_h

4.5.10 Conversión de Hexadecimal a Binario

Cada caracter hexadecimal se representa en 4 bits, tanto para la parte entera como la fraccionaria; similar a la conversión de octal a binario.

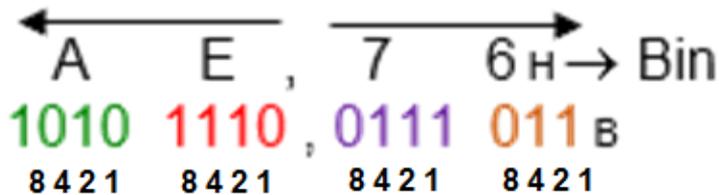


Figura 47. Conversión de Hexadecimal a Binario

► Ejemplos:

• F4A,1B_h a Binario

Parte entera: F 4 A_h, representados en 4 bits cada caracter = 1111 0100 1010_b

Parte fraccionaria: 1 B_h, representados en 4 bits cada caracter = 0001 1011_b

R/: F4A,1B_h = 111101001010,00011011_b

• 15B,731_h a Binario

Parte entera: 15 B_h, representados en 4 bits cada caracter = 0001 0101 1011_b

Parte fraccionaria: 73 1_h, representados en 4 bits cada caracter = 0111 0011 0001_b

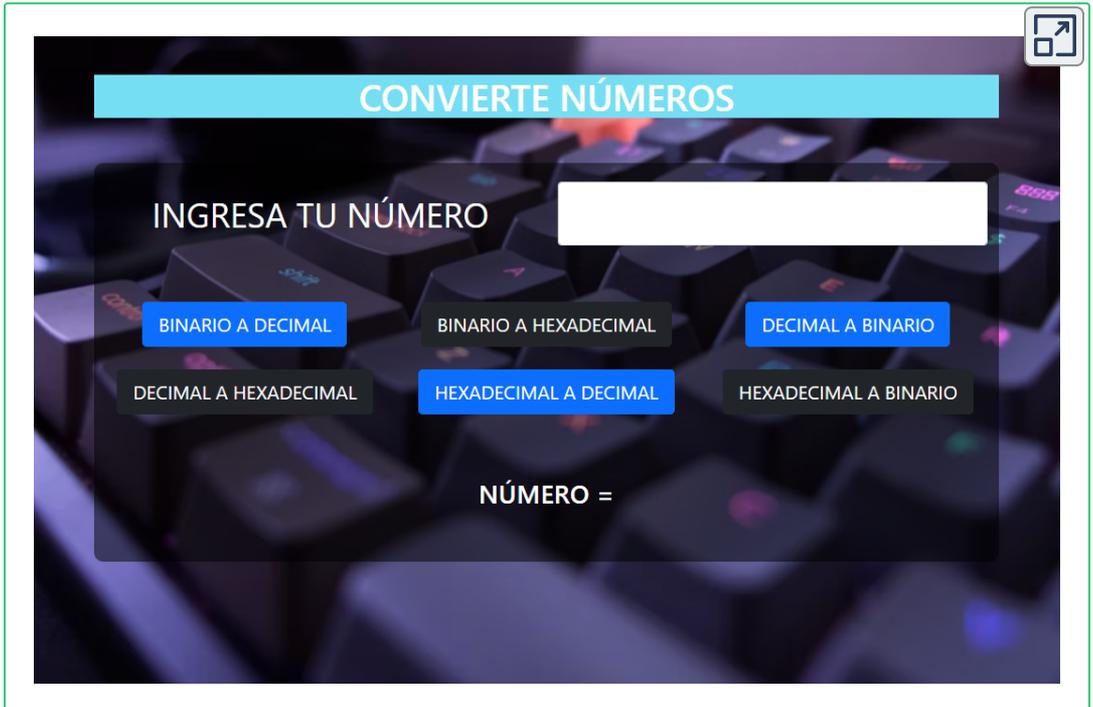
R/: 15B,731_h = 000101011011,011100110001_b

► **Ejercicios propuestos con respuestas para practicar las diferentes conversiones y así verificar el nivel de aprendizaje::**

- ¿11111111,10001_b = hexadecimal? ► **R/. 1FF,88_h**
- ¿1010111,11101_b = octal? ► **R/. 127,72_o**
- ¿142,75_d = octal ► **R/. 216, 6_o**
- ¿109,E_h = decimal? ► **R/. 265,875_d**
- ¿375,25_o = hexadecimal? ► **R/. FD,54_h**
- ¿357,15_o = binario? ► **R/. 11101111,001101_b**
- ¿A35,B7_h = binario? ► **R/. 101000110101,10110111_b**
- ¿197,8125_d = hexadecimal? ► **R/. C5,D_h**
- ¿381,0A_h = octal? ► **R/. 1601,024_o**
- ¿11010011,01101_b = decimal? ► **R/. 211,40625_d**

- ¿ $301,75_d = \text{binario?}$ ► R/. $100101101,11_b$
- ¿ $164,15_o = \text{decimal?}$ ► R/. $116,203125_d$

En el siguiente OIA, diseñado por [Alfredo Sierra](#), puede convertir números de un sistema a otro.



Ingrese, en primer lugar, el número a convertir. Luego haga clic sobre uno de los seis botones de conversión. Los números hexadecimales se deben escribir en mayúscula.



No.	$2^1 = 2$	$2^0 = 1$
DEC	B	A
0	0	0
1	0	1
2	1	0
3	1	1

2 variables



No.	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
DEC	C	B	A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

3 variables



No.	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
DEC	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

4 variables

Capítulo V

CREACIÓN DE TABLAS DE VERDAD



5.1 Variables y Combinaciones para crear una tabla de verdad

Para crear una tabla de verdad se utiliza el sistema numérico binario, se le asigna a cada variable de la función una columna y por medio de la siguiente fórmula se obtiene el número de combinaciones que tendrá la tabla.

FÓRMULAS:

$2^n =$ Número de combinaciones posibles

$n =$ Cantidad de variables de la función

► Ejemplo:

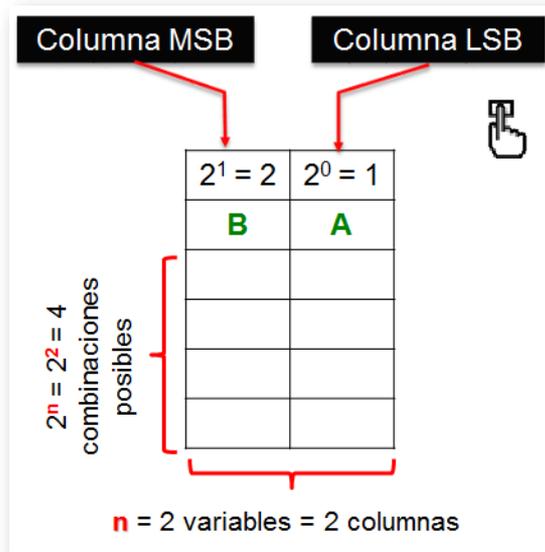


Figura 48. Variables y combinaciones para crear una tabla de verdad

Las variables se asignan en orden de derecha a izquierda o sea del bit menos significativo (LSB) hacia el más significativo (MSB) y se hace en sistema numérico binario.

5.1.1 Tabla con una variable

La columna A tiene una ponderación de 1, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a uno, comenzando con el '0'.



No.	$2^0 = 1$
DEC	A
0	0
1	1

1 variable

Figura 49. Creación de la tabla con una variable

5.1.2 Tabla con dos variables

La columna A tiene una ponderación de 1, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a uno, comenzando con '0'.

La columna B tiene una ponderación de 2, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a dos, comenzando con '0'.



No.	$2^1 = 2$	$2^0 = 1$
DEC	B	A
0	0	0
1	0	1
2	1	0
3	1	1

2 variables

Figura 50. Creación de la tabla con dos variables

5.1.3 Tabla con tres variables



No.	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
DEC	C	B	A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

3 variables

Figura 51. Creación de la tabla con tres variables

La columna A tiene una ponderación de 1, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a uno, comenzando con '0'.

La columna B tiene una ponderación de 2, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a dos, comenzando con '0'.

La columna C tiene una ponderación de 4, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a cuatro, comenzando con '0'.

5.1.4 Tabla con cuatro variables



No.	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
DEC	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

4 variables

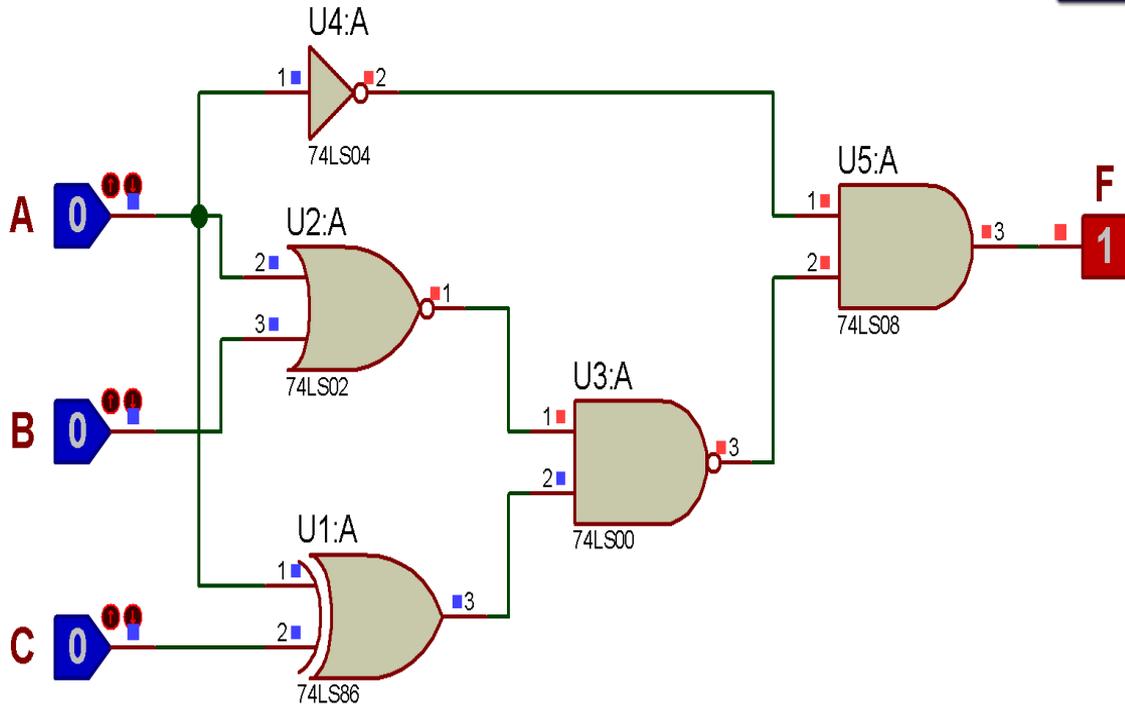
Figura 52. Creación de la tabla con cuatro variables

La columna A tiene una ponderación de 1, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a uno, comenzando con '0'.

La columna B tiene una ponderación de 2, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a dos, comenzando con '0'.

La columna C tiene una ponderación de 4, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a cuatro, comenzando con '0'.

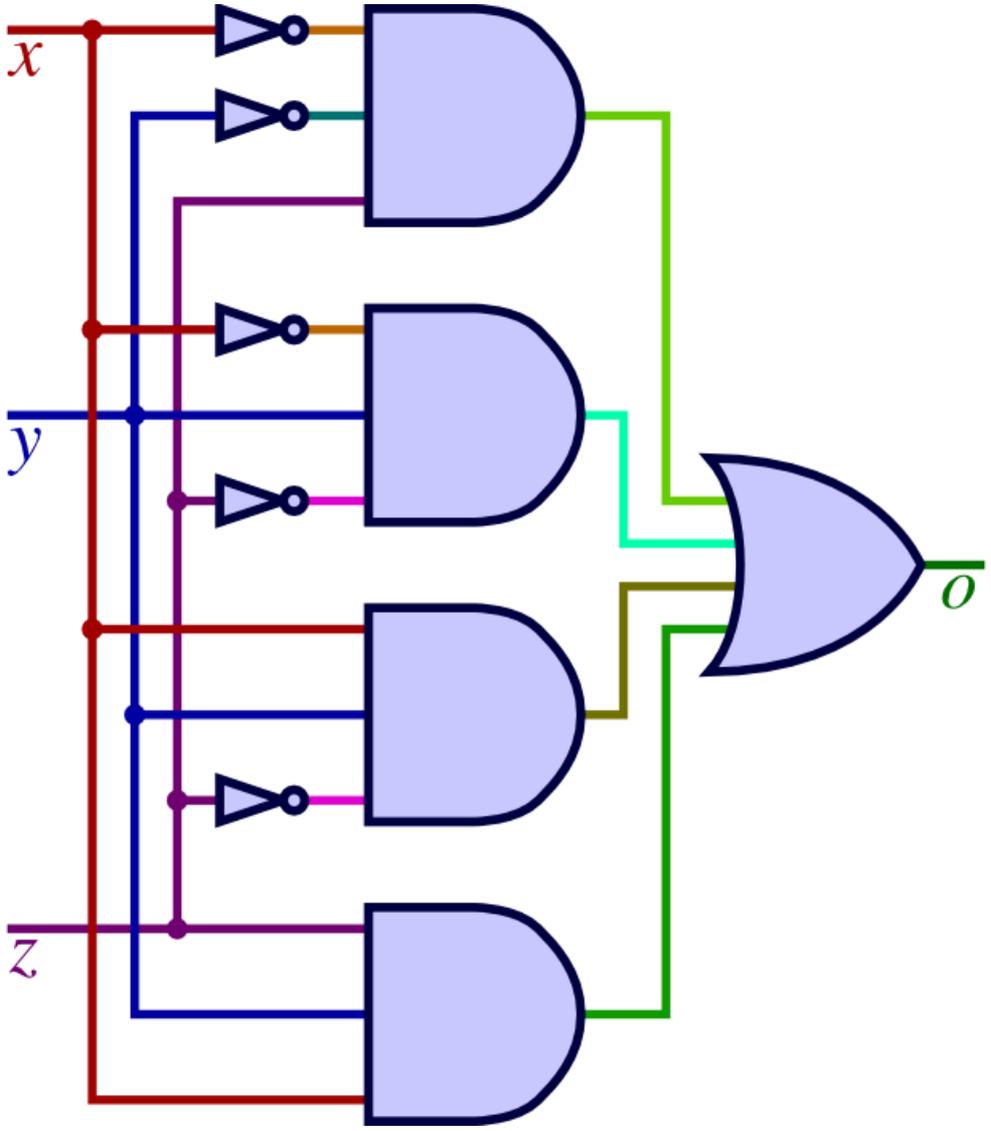
La columna D tiene una ponderación de 8, lo que quiere decir que en esa columna se colocarán '0' y '1' alternados de a ocho, comenzando con '0'. Y así sucesivamente, hasta completar todas las variables del sistema.



$$F = \{(A+B)' \times (A \text{ xor } C)\}' \times A'$$

Capítulo VI

COMPUERTAS LÓGICAS Y FUNCIONES BOOLEANAS



6.1 George Boole



George Boole (1815-1864) Matemático Británico. Es el inventor del álgebra de Boole, que marca los fundamentos de la aritmética computacional moderna; en este sistema, las proposiciones se reducen a símbolos que se operan matemáticamente, lo que llevó a un avance fundamental de la lógica. A los 16 años enseñaba matemáticas

en un colegio privado y más tarde fundó uno propio. En 1849 fue nombrado profesor de matemáticas del Queen's College, en Cork, donde permaneció el resto de su vida. Se destaca su obra *Análisis Matemático de la Lógica* (1847), donde plasmó sus primeras observaciones sobre los vínculos entre la lógica y las matemáticas que dio pie al nacimiento de la lógica matemática.

6.2 Compuertas Lógicas

Una compuerta lógica es un dispositivo electrónico con una función booleana propia, que realizan operaciones lógicas como: suma, multiplicación, niegan, afirman, incluyen o excluyen.

Las compuerta lógicas trabajan con niveles lógicos y no con los flancos de la señal.

El sistema numérico binario es el que se utiliza para los datos de entrada a la compuerta lógica, igualmente el que entrega en su salida.

Cada compuerta lógica tiene su complemento y son 8 en total.

6.2.1 Compuertas YES y su complemento NOT

YES	NOT												
$S = A$	$S = \bar{A}$												
													
													
La salida es "1" cuando la entrada es "1"	La salida es "0" cuando la entrada es "1"												
<table border="1"> <thead> <tr> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	S	0	0	1	1	<table border="1"> <thead> <tr> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	S	0	1	1	0
A	S												
0	0												
1	1												
A	S												
0	1												
1	0												
Proteus: 4050	Proteus: NOT 74LS04												

Figura 53. Compuertas YES y NOT

Diagrama de tiempos compuerta YES

In	A								
Out	F								



Diagrama de tiempos compuerta NOT

In	A								
Out	F								



Figura 54. Diagrama de tiempos y circuito en Proteus

6.2.2 Compuertas AND y su complemento NAND

AND	NAND																														
$S = A \times B$	$S = \overline{A \times B}$																														
La salida es "1" cuando TODAS las entradas son "1"	La salida es "0" cuando TODAS las entradas son "1"																														
<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	B	A	S	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	S	0	0	1	0	1	1	1	0	1	1	1	0
B	A	S																													
0	0	0																													
0	1	0																													
1	0	0																													
1	1	1																													
B	A	S																													
0	0	1																													
0	1	1																													
1	0	1																													
1	1	0																													
Proteus: AND / AND_2 74LS08	Proteus: NAND / NAND_2 74LS00																														

Figura 55. Compuertas AND y NAND

Diagrama de tiempos compuerta AND

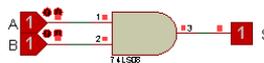
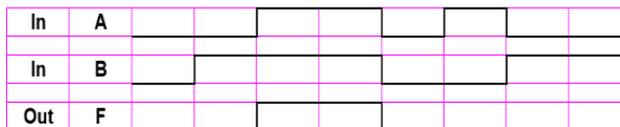


Diagrama de tiempos compuerta NAND

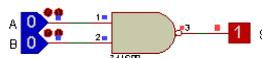
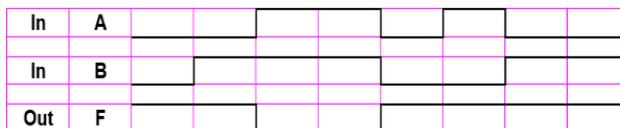


Figura 56. Diagrama de tiempos y circuito en Proteus

6.2.3 Compuertas OR y su complemento NOR

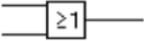
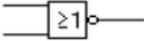
OR	NOR																														
$S = A + B$	$S = \overline{A + B}$																														
																															
																															
La salida es "1" cuando ALGUNA entrada es "1"	La salida es "0" cuando ALGUNA entrada es "1"																														
<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	B	A	S	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	S	0	0	1	0	1	0	1	0	0	1	1	0
B	A	S																													
0	0	0																													
0	1	1																													
1	0	1																													
1	1	1																													
B	A	S																													
0	0	1																													
0	1	0																													
1	0	0																													
1	1	0																													
Proteus: OR 74LS32	Proteus: NOR / NOR_2 74LS02																														

Figura 57. Compuertas OR y NOR

Diagrama de tiempos compuerta OR

In	A								
In	B								
Out	F								

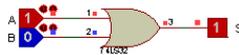


Diagrama de tiempos compuerta NOR

In	A								
In	B								
Out	F								

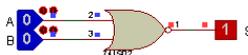


Figura 58. Diagrama de tiempos y circuito en Proteus

6.2.4 Compuertas XOR y su complemento XNOR

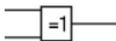
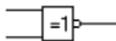
XOR	XNOR																														
$S = A \oplus B$ $S = (A \times \bar{B}) + (\bar{A} \times B)$	$S = \bar{A} \oplus \bar{B}$ $S = (A \times B) + (\bar{A} \times \bar{B})$																														
																															
																															
La salida es "1" cuando son CONTRARIAS las entradas	La salida es "0" cuando son CONTRARIAS las entradas																														
<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	B	A	S	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	B	A	S	0	0	1	0	1	0	1	0	0	1	1	1
B	A	S																													
0	0	0																													
0	1	1																													
1	0	1																													
1	1	0																													
B	A	S																													
0	0	1																													
0	1	0																													
1	0	0																													
1	1	1																													
Proteus: XOR / XOR_2 74LS86	Proteus: 4077																														

Figura 59. Compuertas XOR y XNOR

Diagrama de tiempos compuerta XOR

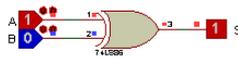
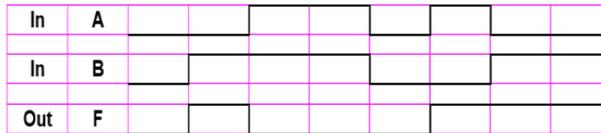


Diagrama de tiempos compuerta XNOR

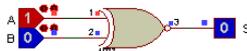
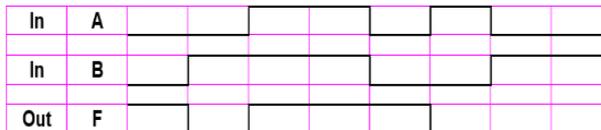


Figura 60. Diagrama de tiempos y circuito en Proteus

En el siguiente objeto interactivo de aprendizaje, diseñado por Juan Guillermo Rivera Berrío², se describen las compuertas lógicas y su funcionamiento.

Selecciona la compuerta, haciendo clic en el botón correspondiente

OR NOR AND NAND XOR NXOR NOT

La compuerta OR implementa la disyunción lógica: se comporta de acuerdo con la tabla de verdad de la derecha. Se produce una salida ALTA si una o ambas entradas a la puerta son ALTAS. Si ninguna de las entradas es alta, se obtiene como resultado una salida BAJA.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



² El OIA se diseñó tomando como referencia la escena de [MushroomFX](#)

6.3 Referencias Comerciales de las Compuertas Lógicas

TIPO DE COMPUERTA	FAMILIA LÓGICA TTL	FAMILIA LÓGICA CMOS	DESCRIPCIÓN
AND	7408	4081	Quad 2 – input (4 compuertas de 2 entradas cada una)
	7411	4073	Triple 3 – input (3 compuertas de 3 entradas cada una)
	7421	4082	Dual 4 – input (2 compuertas de 4 entradas cada una)
NAND	7400	4011	Quad 2 – input (4 compuertas de 2 entradas cada una)
	7410	4023	Triple 3 – input (3 compuertas de 3 entradas cada una)
	7420 – 7440	4012	Dual 4 – input (2 compuertas de 4 entradas cada una)
	7430	4068	8 – input (1 compuerta de 8 entradas)
	74133	–	13 – input (1 compuerta de 13 entradas)
OR	7432	4071	Quad 2 – input (4 compuertas de 2 entradas cada una)
	–	4075	Triple 3 – input (3 compuertas de 3 entradas cada una)
	–	4072	Dual 4 – input (2 compuertas de 4 entradas cada una)
NOR	7402	4001	Quad 2 – input (4 compuertas de 2 entradas cada una)
	7427	4025	Triple 3 – input (3 compuertas de 3 entradas cada una)
	7425	4002	Dual 4 – input (2 compuertas de 4 entradas cada una)
	–	4078	8 – input (1 compuerta de 8 entradas)
NOT ó INV	7404	4069	Hex (6 compuertas de 1 entrada cada una)
YES ó BUFFER	7407	4050	Hex (6 compuertas de 1 entrada cada una)
XOR	7486 – 74136	4030 – 4070	Quad 2 – input (4 compuertas de 2 entradas cada una)
XNOR	–	4077	Quad 2 – input (4 compuertas de 2 entradas cada una)

Figura 61. Referencias comerciales de las compuertas lógicas

6.4 Estructura interna de las compuertas lógicas básicas

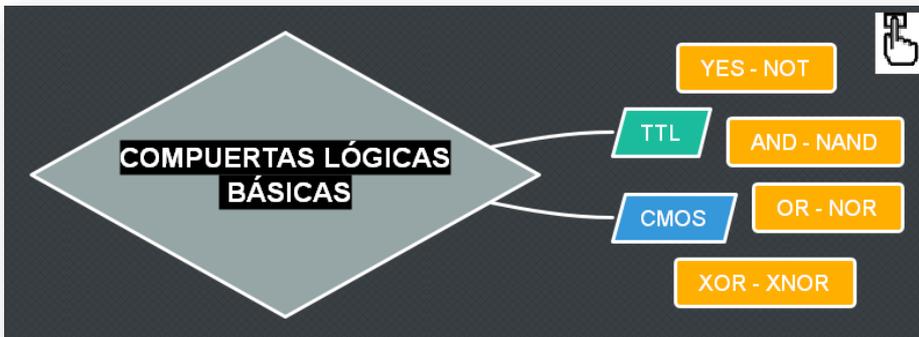
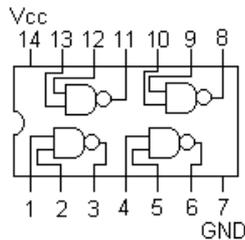
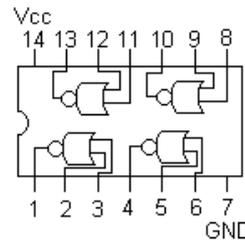


Figura 62. Compuertas básicas en cada familia lógica

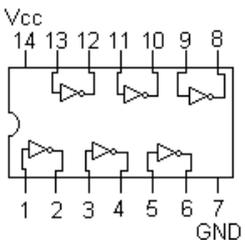
6.4.1 Estructura interna de las compuertas lógicas básicas en la familia TTL



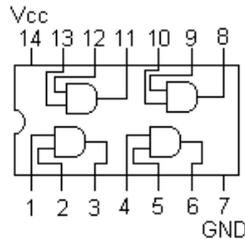
7400 NAND
QUAD 2 INPUT
 $S = \overline{A \times B}$



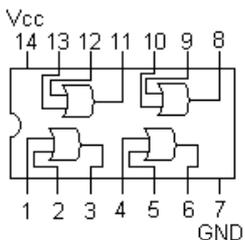
7402 NOR
QUAD 2 INPUT
 $S = \overline{A + B}$



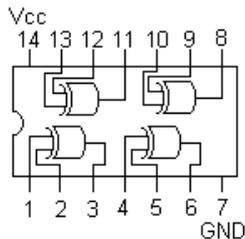
7404
HEX INVERTER
 $S = \overline{A}$



7408 AND
QUAD 2 INPUT
 $S = A \times B$



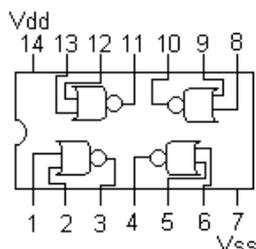
7432 OR
QUAD 2 INPUT
 $S = A + B$



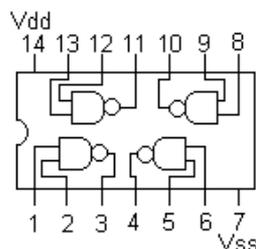
7486 XOR
QUAD 2 INPUT
 $S = A \oplus B$

Figura 63. Estructura interna de las compuertas lógicas básicas en la familia TTL (1 y 2 entradas)

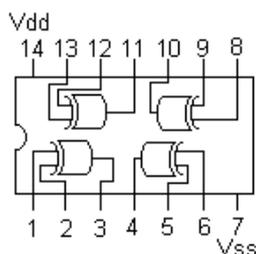
6.4.2 Estructura interna de las compuertas lógicas básicas en la familia CMOS (1 y 2 entradas)



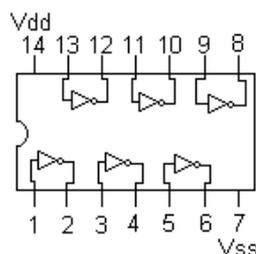
4001 NOR
QUAD 2 INPUT
 $S = \overline{A + B}$



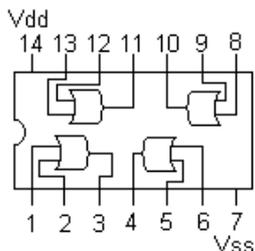
4011 NAND
QUAD 2 INPUT
 $S = \overline{A \times B}$



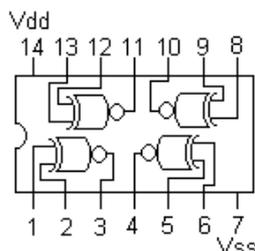
4030 XOR
QUAD 2 INPUT
 $S = A \oplus B$



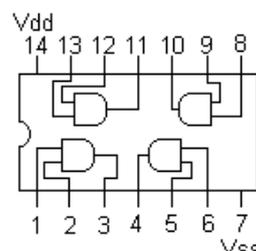
4069
HEX INVERTER
 $S = \overline{A}$



4071 OR
QUAD 2 INPUT
 $S = A + B$



4077 XNOR
QUAD 2 INPUT
 $S = \overline{A \oplus B}$



4081 AND
QUAD 2 INPUT
 $S = A \times B$

Figura 64. Estructura interna de las compuertas lógicas básicas en la familia CMOS

6.5 Funciones Booleanas

► Ejemplos:

- Halle la tabla de verdad paso a paso y dibuje el circuito (plano esquemático - SCH), para la siguiente función Booleana:

$$F = \overline{(A \times B) + (A + B)}$$

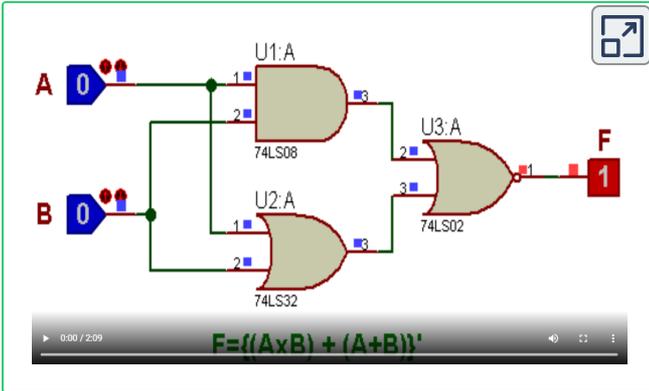
n = 2 variables (A y B)

2ⁿ = 2² = 4 combinaciones posibles

$$F = \overline{\underbrace{(A \times B)}_1 + \underbrace{(A + B)}_2}$$

No.	2 ¹	2 ⁰	1	2	F
Dec	B	A	AxB	A+B	$\overline{1+2}$
0	0	0	0	0	1
1	0	1	0	1	0
2	1	0	0	1	0
3	1	1	1	1	0
			AND	OR	NOR

Vídeo:



Video realizado por Oscar Ignacio Botero Henao, titulado "Ejemplo 1"

- Halle la tabla de verdad paso a paso y dibuje el circuito (plano esquemático - SCH), para la siguiente función Booleana:

$$F = \{(\overline{A + B}) \times (B \times C)\} \oplus (\overline{A \times C})$$

$n = 3$ variables (A, B y C)

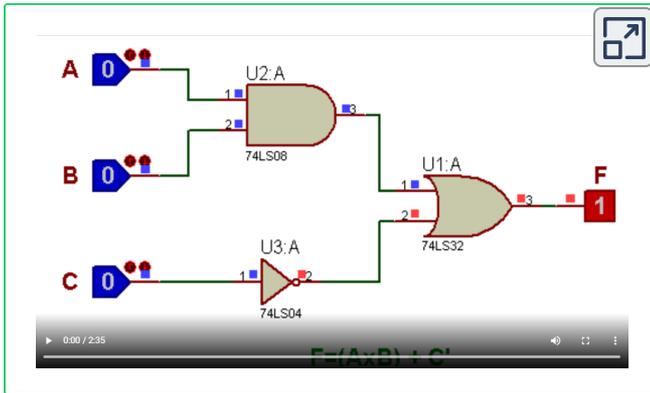
$2^n = 2^3 = 8$ combinaciones posibles

$$F = \underbrace{\{(\overline{A + B}) \times (B \times C)\}}_4 \oplus \underbrace{(\overline{A \times C})}_3$$

1
2
3

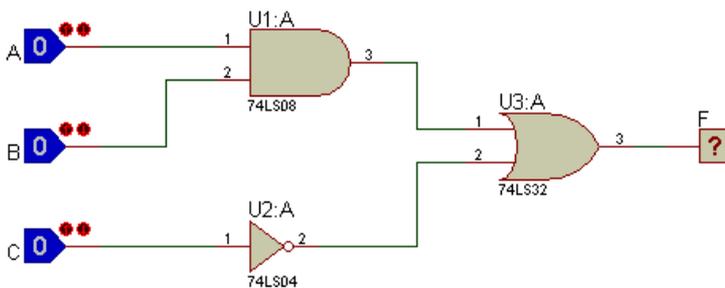
No.	2^2	2^1	2^0	1	2	3	4	F
Dec	C	B	A	$\overline{A+B}$	$B \times C$	$\overline{A \times C}$	1×2	$4 \oplus 3$
0	0	0	0	1	0	1	0	1
1	0	0	1	0	0	1	0	1
2	0	1	0	0	0	1	0	1
3	0	1	1	0	0	1	0	1
4	1	0	0	1	0	1	0	1
5	1	0	1	0	0	0	0	0
6	1	1	0	0	1	1	0	1
7	1	1	1	0	1	0	0	0
				NOR	AND	NAND	AND	XOR

Vídeo:

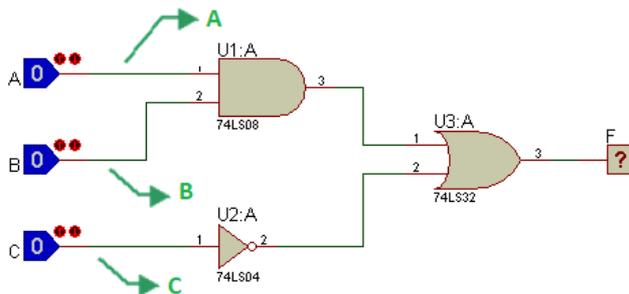


Video realizado por Oscar Ignacio Botero Henao, titulado "Ejemplo 2".

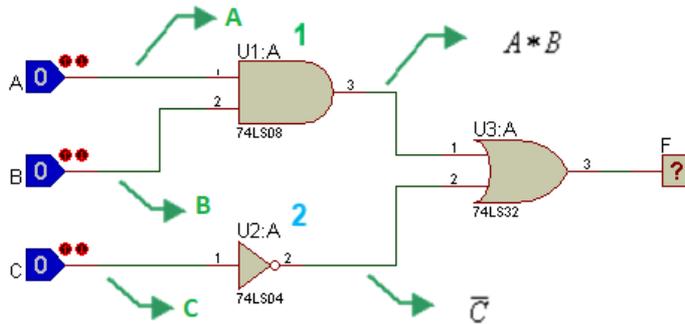
- Halle la tabla de verdad paso a paso y su función lógica, con base en siguiente circuito (pano SCH):



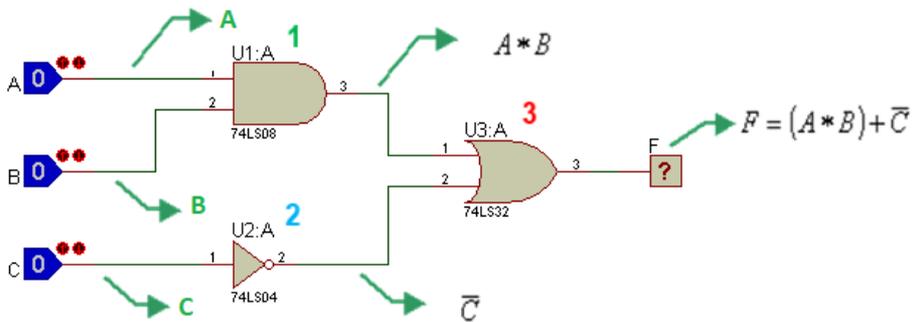
Se procede a etiquetar cada alambre comenzando por las entradas hasta llegar a la salida y así hallar la Función Booleana de la respuesta.



Ahora, se etiquetan los alambres de las salidas de las compuertas AND y NOT.



Finalmente, se etiqueta el alambre de la salida de la compuerta OR, que es la función Booleana final.



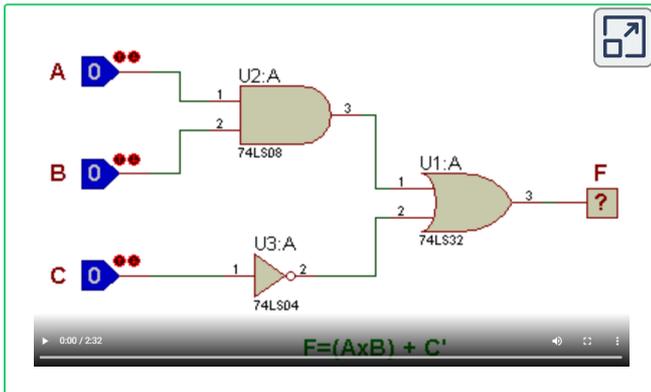
Se observa en la función Booleana obtenida en la salida que contiene 3 variables (A, B y C), lo que conlleva a una tabla de verdad de 8 combinaciones y se procede a elaborarla paso a paso

$n = 3$ variables (A, B y C)

$2^n = 2^3 = 8$ combinaciones posibles

No.	2^2	2^1	2^0	1	2	3
Dec	C	B	A	$A * B$	\overline{C}	$F = (A \times B) + \overline{C}$
0	0	0	0	0	1	1
1	0	0	1	0	1	1
2	0	1	0	0	1	1
3	0	1	1	1	1	1
4	1	0	0	0	0	0
5	1	0	1	0	0	0
6	1	1	0	0	0	0
7	1	1	1	1	0	1
				AND	INV	OR

Vídeo:



Video realizado por Oscar Ignacio Botero Henao, titulado "Ejemplo 3".

- Halle la tabla de verdad paso a paso y dibuje el circuito (plano esquemático - SCH), para la siguiente función Booleana:

$$F = \overline{[(\overline{A + B}) * (A \oplus C)] * \overline{A}}$$

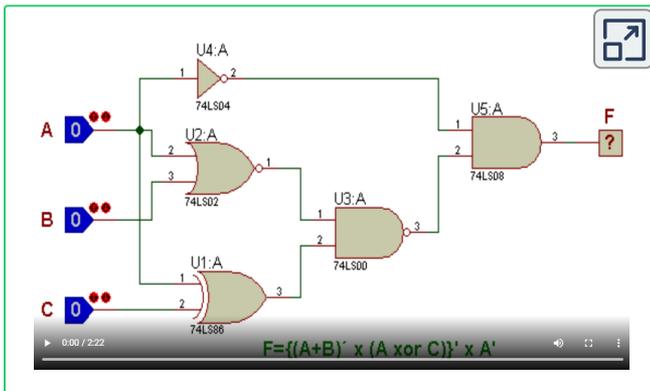
$n = 3$ variables (A, B y C)

$2^n = 2^3 = 8$ combinaciones posibles

$$F = \overline{\underbrace{[\underbrace{(A+B)}_1 * \underbrace{(A \oplus C)}_2]}_{3}} * \underbrace{\bar{A}}_4$$

No.	2 ²	2 ¹	2 ⁰	1	2	3	4	F
Dec	C	B	A	(A+B)	(A ⊕ C)	1+2	Ā	3 x 4
0	0	0	0	1	0	1	1	1
1	0	0	1	0	1	1	0	0
2	0	1	0	0	0	1	1	1
3	0	1	1	0	1	1	0	0
4	1	0	0	1	1	0	1	0
5	1	0	1	0	0	1	0	0
6	1	1	0	0	1	1	1	1
7	1	1	1	0	0	1	0	0
				NOR	XOR	NAND	NOT	AND

Vídeo:



Video realizado por Oscar Ignacio Botero Henao, titulado "Ejemplo 4"

► Ejercicios propuestos con respuestas para practicar Funciones Booleanas, con el fin de verificar el aprendizaje:

$$\blacktriangleright F = \overline{(A \times \bar{B}) + C}$$

R/. en orden descendente en la tabla 1-0-1-1-0-0-0-0

$$\blacktriangleright F = (A \oplus \bar{B}) \times (A + \bar{C})$$

R/. en orden descendente en la tabla 1-0-0-1-0-0-0-1

$$\blacktriangleright F = \overline{(\bar{A} \times C) \oplus (B + C)}$$

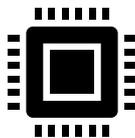
R/. en orden descendente en la tabla 1-1-0-0-1-0-1-0

$$\blacktriangleright F = \{(A \times \bar{B}) \oplus \bar{C}\} + \overline{(B + D)}$$

R/. en orden descendente en la tabla 1-1-1-1-1-1-0-0-1-0-1-1-0-1-0-0

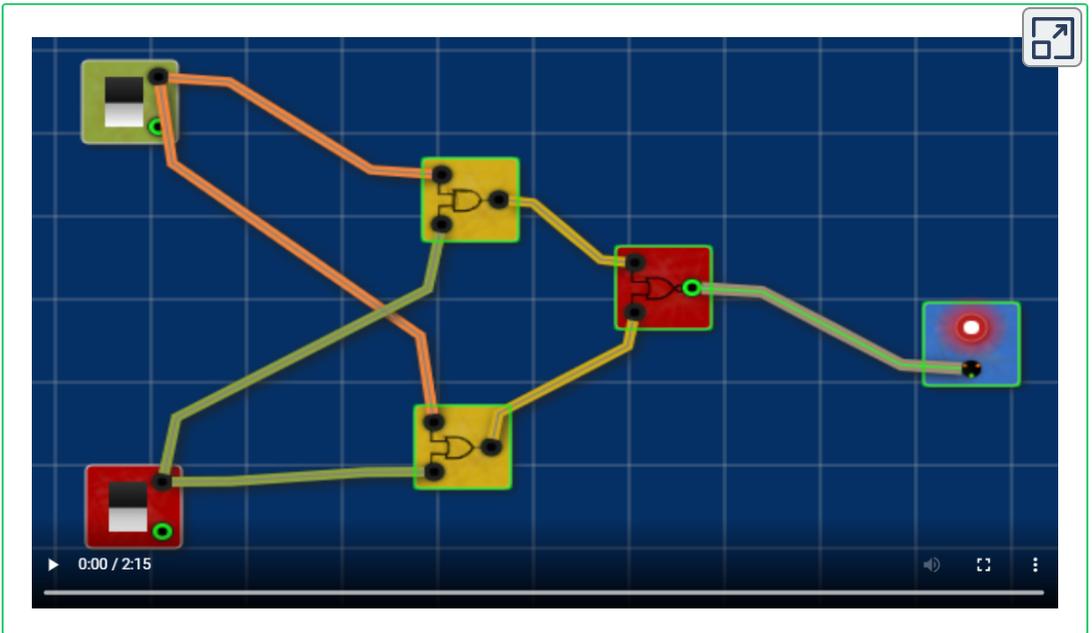
Objeto Interactivo de compuertas lógicas

Una opción a Proteus, es usar el objeto interactivo diseñado por [Buys de Barbanson](#), que se incluye en la barra de herramientas de este libro con el icono:



En el siguiente video se muestra cómo se puede simular el [Ejemplo 1](#).

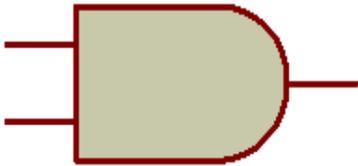
Vídeo:



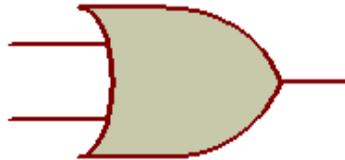
Σ



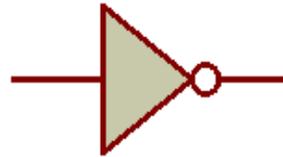
Π



AND



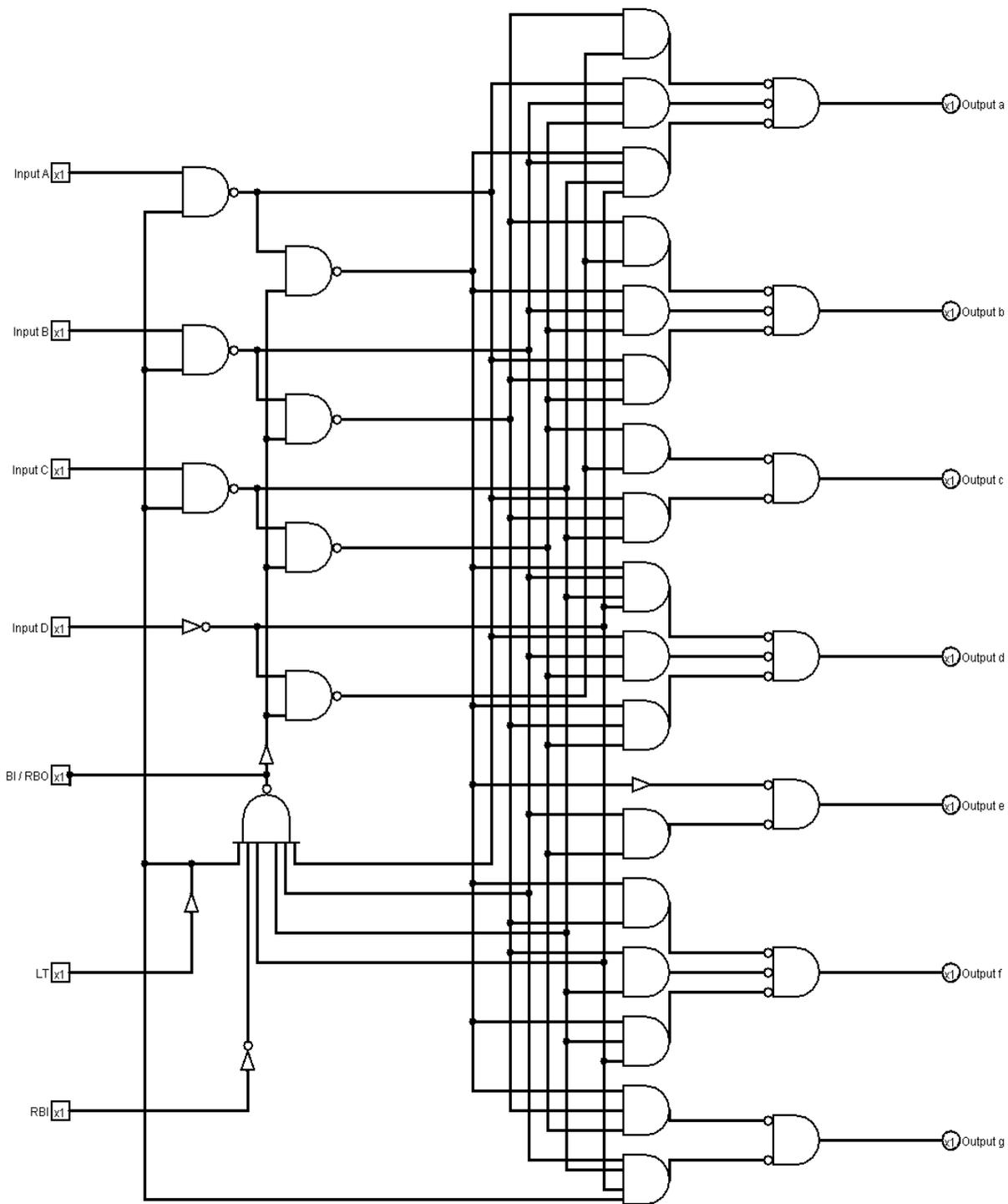
OR



INV

Capítulo VII

MINTERM Y MAXTERM (Lógica AOI)



7.1 Introducción

En muchas ocasiones este método simplifica las funciones lógicas, pero su objetivo no es ese, es convertir una función lógica booleana con múltiples de tipos de compuertas lógicas, en sólo tres (3) tipos de compuertas (AND - OR - INV), por ello ha este sistema se le llama "Lógica AOI", debido a las iniciales de las compuertas lógicas.

Término Canónico: es un término producto o suma en el cual aparecen declaradas todas las variables de la función lógica (negadas o no negadas).

Término Canónico Producto:

$$(\bar{A} \times \bar{B} \times C)$$

Término Canónico Suma:

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

Formatos POS o SOP: es un término en una función donde NO se presentan todas las variables de la función.

$$F(\Sigma) = \underbrace{(A \times B \times C)}_{\text{Término Canónico}} + \underbrace{(A \times \bar{C})}_{\text{Término SOP}}$$

$$F(\pi) = \underbrace{(A + B + C)}_{\text{Término Canónico}} \times \underbrace{(A + \bar{B})}_{\text{Término POS}}$$

7.2 Minterm o Minitérminos (Σ)

Toda tabla de verdad tiene una función lógica y ésta se puede expresar como una **SUMA DE PRODUCTOS CANÓNICOS**, donde cada término **PRODUCTO** será un **MINTERM**. Cada MINTERM se asocia con una combinación de entradas que producirá un '1' en la función de salida.

Si una variable vale '0' se niega y si vale '1' no se niega.

► Ejemplo:

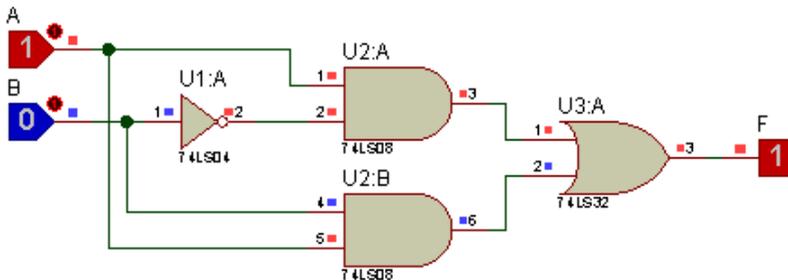
• Se tiene la siguiente tabla de verdad y se requiere hallar la función por Minterm e implementar el circuito:

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

Como es por Minterm, se tiene en cuenta solo las salidas que valen '1'

B	A	F	
0	0	0	
0	1	1	$(A \times \bar{B})$
1	0	0	
1	1	1	$(A \times B)$

$$F(\Sigma) = (A \times \bar{B}) + (A \times B)$$



7.3 Maxterm o Maxitérminos (Π)

Toda tabla de verdad tiene una función lógica y ésta se puede expresar como una **PRODUCTOS DE SUMAS CANÓNICAS**, donde cada término **SUMA** será un **MAXTERM**. Cada MAXTERM se asocia con una combinación de entradas que producirá un '0' en la función de salida.

Si una variable vale '0' no se niega y si vale '1' se niega.

► Ejemplo:

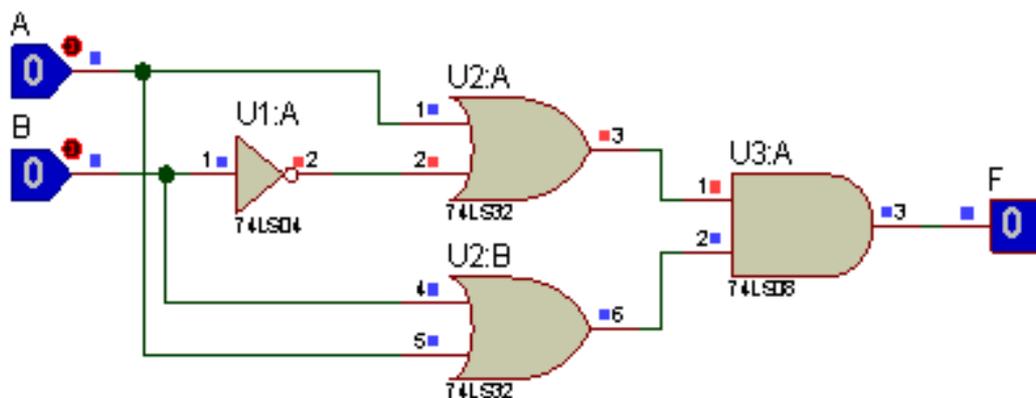
- Se tiene la siguiente tabla de verdad y se requiere hallar la función por Minterm e implementar el circuito:

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

Como es por Maxterm, se tiene en cuenta solo las salidas que valen '0'

B	A	F	
0	0	0	$(A + B)$
0	1	1	
1	0	0	$(A + \bar{B})$
1	1	1	

$$F(\pi) = (A + B) \times (A + \bar{B})$$



7.4 Ejemplos de Minterm y Maxterm

► Ejemplos:

- Se tiene la siguiente función booleana, donde se involucran términos canónicos y un formato SOP (Suma de Productos):

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

Nótese que el primer término es canónico y el segundo término es formato SOP (Suma de Productos) ya que le falta la variables C o C'.

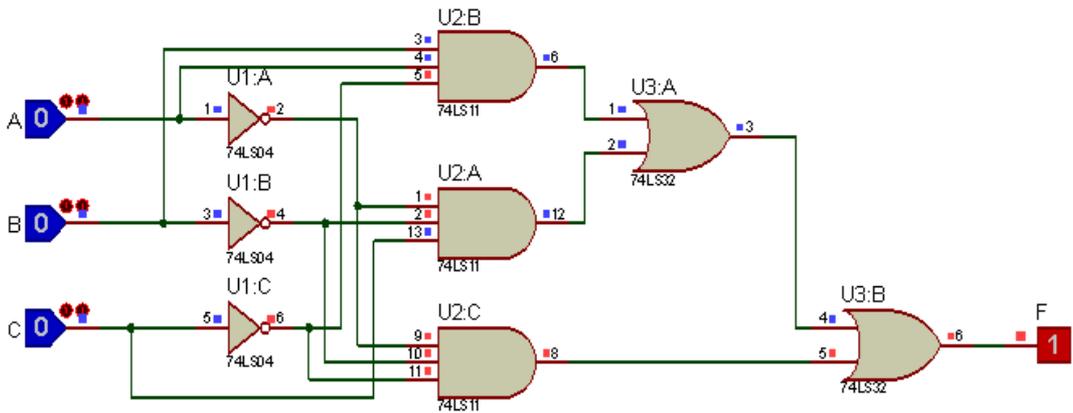
Para la solución se debe convertir en término canónico; para ello se debe completar el término SOP teniendo en cuenta las dos (2) posibles opciones de la variable C.

$$F(\Sigma) = (A \times B \times \bar{C}) + (\bar{A} \times \bar{B} \times C) + (\bar{A} \times \bar{B} \times \bar{C})$$

$$F(\Sigma) = (1 \times 1 \times 0) + (0 \times 0 \times 1) + (0 \times 0 \times 0)$$

Como es por Minterm, se tiene en cuenta solo las salidas que valen '1'

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

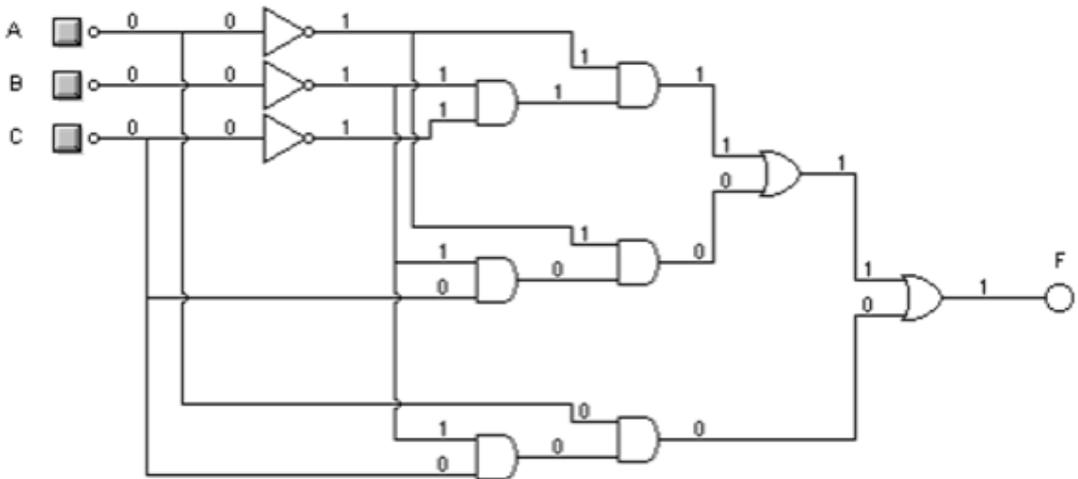


• Obtenga la tabla de verdad, la función por "minterm" y diseñe el circuito combinacional que active una alarma cuando cumpla las siguientes dos condiciones de forma individual:

- 1 - La variable C no esté en '0', la variable B esté en '0'.
- 2 - Las tres variables no estén en '1'.

C	B	A	F	
0	0	0	1	$(\bar{A} \times \bar{B} \times \bar{C})$
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	1	$(\bar{A} \times \bar{B} \times C)$
1	0	1	1	$(A \times \bar{B} \times C)$
1	1	0	0	
1	1	1	0	

Minterm ('1') $F(\Sigma) = (\bar{A} \times \bar{B} \times \bar{C}) + (\bar{A} \times \bar{B} \times C) + (A \times \bar{B} \times C)$



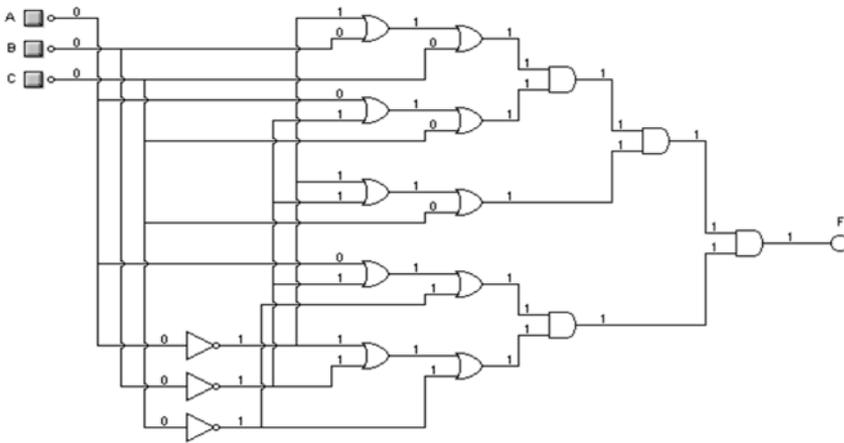
• Obtenga la tabla de verdad, la función por "maxterm" y diseñe el circuito combinacional que active una alarma cuando cumpla las siguientes dos condiciones de forma individual:

1 - La variable C no esté en '0', la variable B esté en '0'.

2 - Las tres variables no estén en '1'.

C	B	A	F	
0	0	0	1	
0	0	1	0	$(\bar{A} + B + C)$
0	1	0	0	$(A + \bar{B} + C)$
0	1	1	0	$(\bar{A} + \bar{B} + C)$
1	0	0	1	
1	0	1	1	
1	1	0	0	$(A + \bar{B} + \bar{C})$
1	1	1	0	$(\bar{A} + \bar{B} + \bar{C})$

Maxterm ('0') $F(\pi) = (\bar{A} + B + C) \times (A + \bar{B} + C) \times (\bar{A} + \bar{B} + C) \times (A + \bar{B} + \bar{C}) \times (\bar{A} + \bar{B} + \bar{C})$



► **Ejercicios propuestos con respuestas para practicar Lógica AOI, con el fin de verificar lo aprendido:**

► ¿La siguiente función está expresada de forma Minterm o Maxterm? y halle la tabla de la función:

$$(A \times B \times \bar{C}) + (A \times B \times C) + (\bar{A} \times C)$$

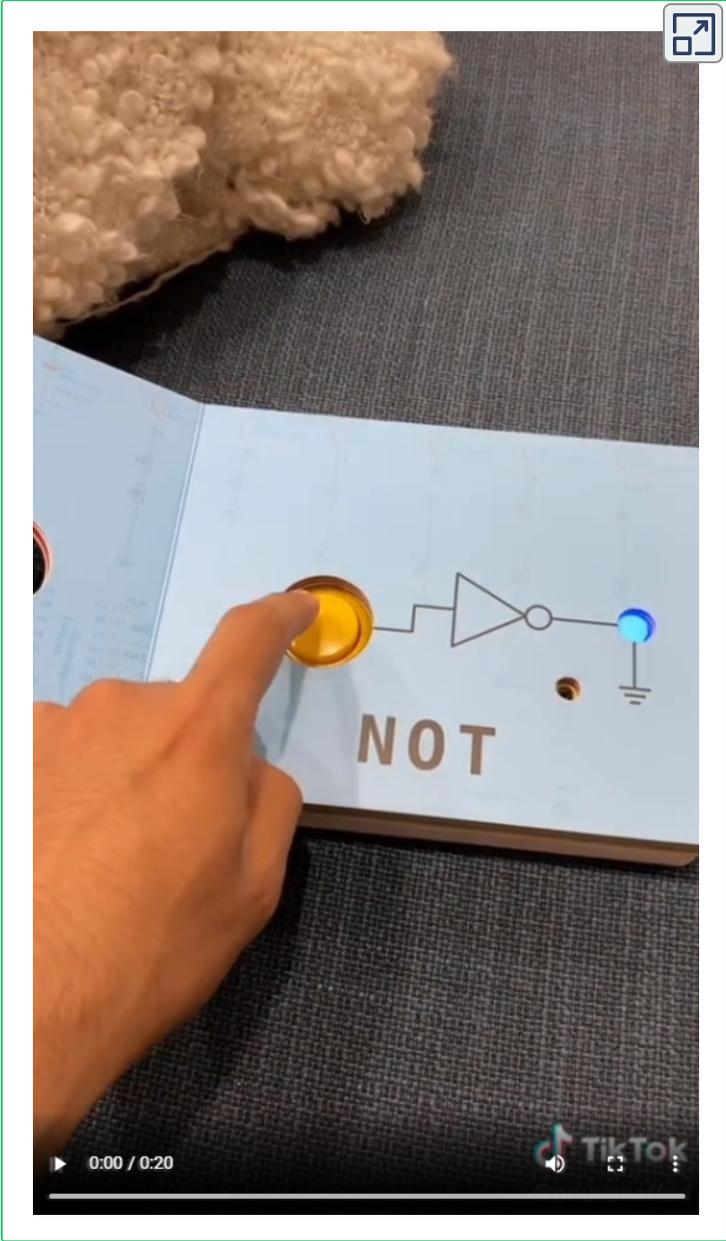
R/. La función está expresada de forma Minterm con dos términos canónicos y uno en formato SOP (suma de productos) y la tabla en orden descendente es: 0-0-0-1-1-0-1-1

► Se tiene la respuesta de una tabla de verdad en el siguiente orden descendente: 1-0-1-0-1-1-1-0 y se requiere hallar la función por Maxterm.

La función expresada por Maxterm es:

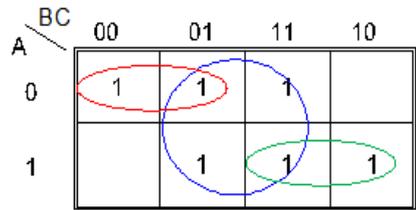
$$(\bar{A} + B + C) \times (\bar{A} + \bar{B} + C) \times (\bar{A} + \bar{B} + \bar{C})$$

A continuación, un curioso video sobre cómo explicar las compuertas lógicas:

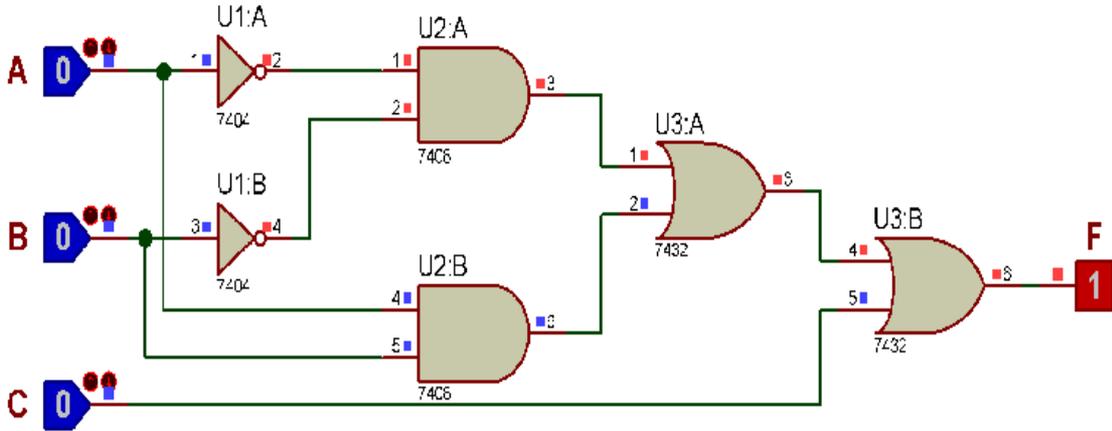


Video publicado en [TikTok](#), en el canal [@wiltonpalma_9425](#)

C	B	A	F
0	0	0	1
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	1



$$F(\pi) = (\bar{A} \times \bar{B}) + (A \times B) + C$$



Capítulo VIII

MAPAS DE KARNAUGH

A

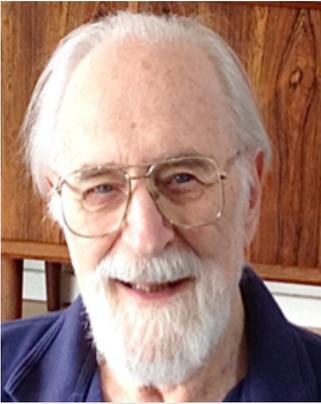
1 $\bar{A}\bar{B}\bar{C}\bar{D}$ 0000	3 $\bar{A}\bar{B}C\bar{D}$ 0010	11 $\bar{A}B\bar{C}\bar{D}$ 1010	9 $\bar{A}B\bar{C}D$ 1000
2 $\bar{A}B\bar{C}\bar{D}$ 0001	4 $\bar{A}B\bar{C}D$ 0011	12 $\bar{A}BC\bar{D}$ 1011	10 $\bar{A}BCD$ 1001
6 $\bar{A}B\bar{C}D$ 0101	8 $\bar{A}BCD$ 0111	16 $ABC\bar{D}$ 1111	14 $ABC\bar{D}$ 1101
5 $\bar{A}BC\bar{D}$ 0100	7 $\bar{A}BCD$ 0110	15 $ABC\bar{D}$ 1110	13 $ABC\bar{D}$ 1100

D

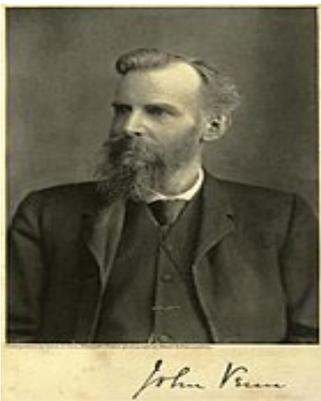
B

C

8.1 Maurice Karnaugh y John Venn



Maurice Karnaugh (1924-) Matemático y Físico Norte Americano. Estudió matemáticas y física en el City College de Nueva York (1944-1948), luego en la Universidad de Yale donde hizo su licenciatura (1949), M.Sc. (1950) y Ph.D. en Física con una tesis sobre la teoría de la resonancia magnética y grecas duplicación de Óxido Nítrico (1952). Trabajo como: Investigador en los Laboratorios Bell desde 1952 hasta 1966. En el centro de investigación de IBM de 1966 a 1993. Profesor de informática en el Politécnico de Nueva York de 1980 a 1999. Miembro del IEEE desde 1975 (Instituto de Ingenieros Eléctricos y Electrónicos – Institute of Electrical and Electronics Engineers). Actualmente gobernador emérito del ICCA (Consejo Internacional para las Comunicaciones Computacionales – International Council for Computer Communication). En 1950 creó el método llamado mapa de Karnaugh, cuya función es minimizar o simplificar las funciones algebraicas booleanas.



John Venn (1834-1923). Matemático y Lógico Británico. Se conoce por su método de representar de forma gráfica 1880 las proposiciones, teniendo en cuenta sus características de cualidad y cantidad; además, de silogismos que posteriormente fueron utilizados en la teoría de conjuntos.

Un mapa de Karnaugh consiste de una serie de cuadrados y cada uno de ellos representa una línea o combinación de la tabla de verdad. Cada cuadrado del mapa alberga un '0', un '1' o una 'X' (no importa, don't care); depende si la solución es por minterm (Σ) o por maxterm (Π).

Recuerden que la siguiente fórmula es para saber cuantas combinaciones va a tener la tabla.

$$n = \text{No. de variables}$$

$$2^n = \text{No. de combinaciones}$$

Los mapas de Karnaugh se utilizan en funciones hasta de 8 variables.

8.2 Procedimiento

Los pasos a seguir son:

- Obtener una expresión booleana en forma de minterm o maxterm (lógica AOI).
- Colocar '**1**' o '**0**' en el mapa de Karnaugh de acuerdo a la expresión.
- Se agrupan conjuntos adyacentes de **2, 4, 8, 16, 32 o 64**, unos o ceros.
- Si un '**1**' o '**0**' no tiene datos adyacentes, es una **isla** y no tiene simplificación.
- Los conjuntos solo se conforman de forma **horizontal** y **vertical**, nunca en diagonal.

- Se encierran los '1' o '0' que formen grupos de dos (pareja) pero que no formen grupos de cuatro.
- Se encierran los '1' o '0' que formen grupos de cuatro (cuarteto) pero que no formen grupos de ocho.
- Así sucesivamente hasta cuando todos los '1' o '0' del mapa sean cubiertos.
- Se eliminan las variables que aparezcan con sus complementos y se guardan las restantes (se tienen en cuenta las variables que **NO** cambian).
- Se enlazan con operadores **OR** los grupos obtenidos para formar la expresión simplificada en forma de minterm (Σ) y con operadores **AND** en forma de maxterm (Π).

8.3 Distribución de las celdas

Las variables de la expresión son ordenadas en del código Gray, nombrado así en honor del investigador Frank Gray, es un sistema de numeración binario en el que dos números consecutivos difieren solamente en uno de sus dígitos.



Frank Gray (1887-1969). Físico Norteamericano. Fue un investigador que trabajo en los laboratorios Bell. Realizó muchas innovaciones tanto mecánicas como electrónicas para la televisión. Diseñó el código Gray que es un sistema binario en el que dos números consecutivos difieren solamente en uno de sus dígitos.

8.3.1 Mapa de Karnaugh de dos (2) variables

$2 = \text{variables}$

$2^2 = \text{tabla con } 4 \text{ combinaciones}$

No. DECIMAL	B	A	F
0	0	0	
1	0	1	
2	1	0	
3	1	1	

Minterm

	B	\bar{B}	B
A		0	1
\bar{A}	0	0	2
A	1	1	3

Maxterm

	B	B	\bar{B}
A		0	1
A	0	0	2
\bar{A}	1	1	3

8.3.2 Mapa de Karnaugh de tres (3) variables

$3 = \text{variables}$

$2^3 = \text{tabla con } 8 \text{ combinaciones}$

No. DECIMAL	C	B	A	F
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

Minterm

		BC	$\overline{B}\overline{C}$	$\overline{B}C$	$B\overline{C}$	BC	$B\overline{C}$
A			00	01	11	10	
\overline{A}	0		0	4	6	2	
A	1		1	5	7	3	

Maxterm

		B+C	$B + \overline{C}$	$\overline{B} + \overline{C}$	$\overline{B} + C$	
A			0+0	0+1	1+1	1+0
\overline{A}	0		0	4	6	2
A	1		1	5	7	3

8.3.3 Mapa de Karnaugh de cuatro (4) variables

$4 = \text{variables}$

$2^4 = \text{tabla con } 16 \text{ combinaciones binarias}$

No. DECIMAL	D	C	B	A	F
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

		Minterm				
		CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
AB	$\bar{A}\bar{B}$	00	0	8	12	4
	$\bar{A}B$	01	2	10	14	6
	AB	11	3	11	15	7
	$A\bar{B}$	10	1	9	13	5

Maxterm

	C+D	$C + D$	$C + \bar{D}$	$\bar{C} + \bar{D}$	$\bar{C} + D$
A+B	0+0	0+0	0+1	1+1	1+0
$A + B$	0+0	0	8	12	4
$A + \bar{B}$	0+1	2	10	14	6
$\bar{A} + \bar{B}$	1+1	3	11	15	7
$\bar{A} + B$	1+0	1	9	13	5

Objeto interactivo de aprendizaje (OIA)

En el siguiente OIA, se explican las relaciones existentes entre las tablas de verdad y los mapas de Karnaugh:



Mapas de Karnaugh

En esta última escena de inicio, puedes inferir las relaciones entre las tablas de verdad y el mapa de Karnaugh.

Dos Variables
 Tres Variables
 Cuatro Variables
 Limpiar

Haz clic sobre los botones de la tabla de verdad o en el mapa de Karnaugh para cambiar valores.

Pasa el puntero del ratón sobre los minitérminos de la función **F** para ver su representación en el Mapa de Karnaugh.

Tabla de verdad

A	B	C	F(ABC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Mapa de Karnaugh

		AB			
		00	01	11	10
C	0	0	0	0	0
	1	0	0	0	0

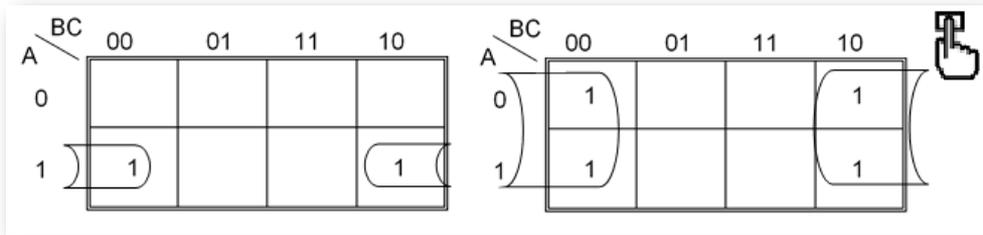
F(ABC)= 0

Atrás

8.4 Ejercicios con Mapas de Karnaugh

8.4.1 Agrupación de datos adyacentes

Los mapas de Karnaugh de 3 variables se pueden doblar horizontalmente, uniéndose el extremo izquierdo con el derecho, para formar ya sea una pareja o un cuarteto, como se observa en la siguiente figura.



Recuerde, que se tienen en cuenta las variables que **NO** cambian.

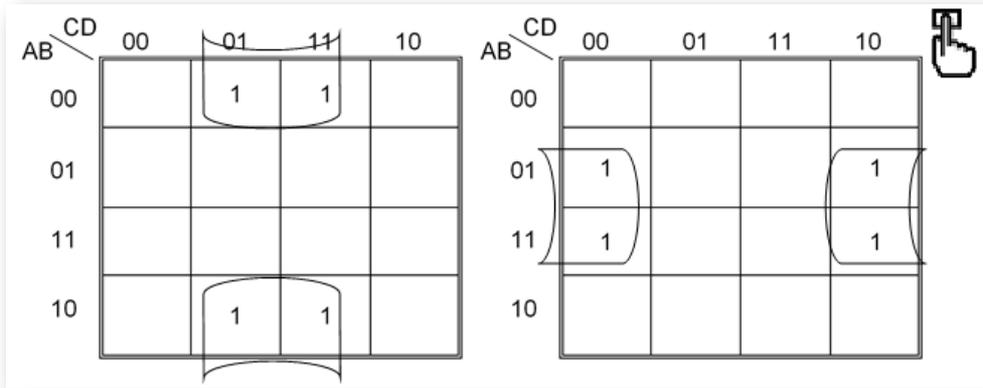
Siempre se lee filas y columnas. Para el mapa del lado izquierdo, observe que los dos (2) '1' se encuentran localizados en la fila 'A=1', lo cual no cambia de estado y con respecto a las dos columnas que ocupan no cambia la variable 'C=0'. El mapa está desarrollado por minterm ya que los datos que descargó al mapa son '1'.

$$F(\Sigma) = (A \times \bar{C})$$

Para el mapa del lado derecho, observe que los cuatro (4) '1' cambian de fila y con respecto a las dos columnas que ocupan no cambia la variable 'C=0'. El mapa está desarrollado por minterm ya que los datos que descargó al mapa son '1'.

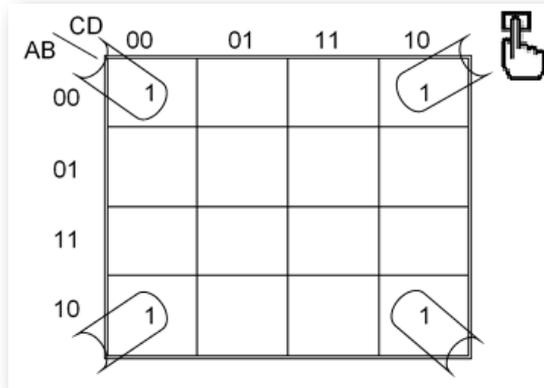
$$F(\Sigma) = (\bar{C})$$

Los mapas de Karnaugh de 4 variables se pueden doblar horizontal, vertical o en ambos sentidos simultáneamente, uniéndose el extremo izquierdo con el derecho, el superior con el inferior o las cuatro esquinas, para formar ya sea una pareja, cuarteto u octeto, como se observa en la siguiente figura.



Función simplificada mapa izquierdo: $F(\Sigma) = (\bar{B} \times D)$

Función simplificada mapa derecho: $F(\Sigma) = (B \times \bar{D})$



Función simplificada: $F(\Sigma) = (\bar{B} \times \bar{D})$

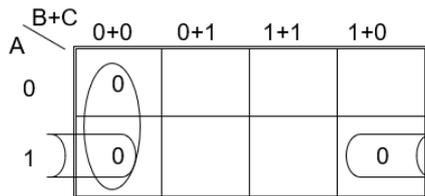
Implementación por medio de maxterm con tres variables.

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

$$(A+B+C)$$

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

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



Función simplificada: $\mathbf{F}(\pi) = (\mathbf{B} + \mathbf{C}) \times (\bar{\mathbf{A}} + \mathbf{C})$

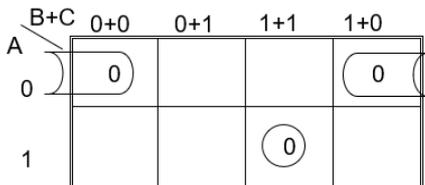
Implementación por medio de maxterm con tres variables:

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

$$(A+B+C)$$

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

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



Función simplificada: $\mathbf{F}(\pi) = (\mathbf{A} + \mathbf{C}) \times (\bar{\mathbf{A}} + \bar{\mathbf{B}} + \bar{\mathbf{C}})$

Implementación por medio de maxterm con cuatro variables:

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

	0+0 C + D	0+1 C + \bar{D}	1+1 \bar{C} + \bar{D}	1+0 \bar{C} + D
0+0 A + B	0			0
0+1 A + \bar{B}	0			
1+1 \bar{A} + \bar{B}	0			0
1+0 \bar{A} + B	0			0

Función simplificada: $F(\pi) = (C + D) \times (B + D) \times (\bar{A} + D)$

Implementación por medio de maxterm con cuatro variables:

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

	0+0 $C+D$	0+1 $C+\bar{D}$	1+1 $\bar{C}+\bar{D}$	1+0 $\bar{C}+D$
0+0 $A+B$	0	0		0
0+1 $A+\bar{B}$				
1+1 $\bar{A}+\bar{B}$				
1+0 $\bar{A}+B$	0			0

Función Booleana simplificada: $\mathbf{F(\pi) = (A + B + C) \times (B + D)}$

► Ejercicios propuestos con respuestas para practicar Mapas de Karnaugh, con el fin de verificar lo aprendido:

► Para la tabla de la izquierda utilizar mapas K por maxterm y para la de la derecha utilizar mapas K por minterm. Baje los datos de la tabla al mapa y luego simplifique, para hallar la nueva función booleana:

C	B	A	F
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	1

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

Funciones Booleanas simplificadas:

$$F(\pi) = (B + C) \times (A + C)$$

$$F(\Sigma) = (A \times B) + (\bar{A} \times \bar{B} \times C)$$

8.5 Objetos Interactivos de Aprendizaje (OIA)

Dando clic en la esquina superior derecha se expande el OIA.

Para este OIA, arrastra cada "Interrogante" al contenedor correspondiente.

SÍ	FRASES	NO
	¿Un mapa de Karnaugh de 4 variables tiene 16 celdas?	
	¿Un mapa de Karnaugh de 4 variables tiene 15 celdas?	
	¿Se puede agrupar 8 bits adyacentes?	
	¿El método de mapas Karnaugh simplifica las funciones booleanas?	

Los siguientes OIA fueron diseñados por Juan Guillermo Rivera Berrío y Hernán Dario Ortíz.



Mapa de Karnaugh para cuatro variables

En esta escena, observarás diferentes combinaciones entre cuatro variables.

Observa los valores que toma la función, y también la relación lógica con los diagramas de Carroll.

$$f(ABCD) \rightarrow \bar{B}$$

CD \ AB	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

Otro ejemplo



Primer ejercicio de cierre

Observa la función y haz clic sobre las celdas, hasta obtener las combinaciones definidas por la función.

$$f(ABCD) \rightarrow BD$$

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

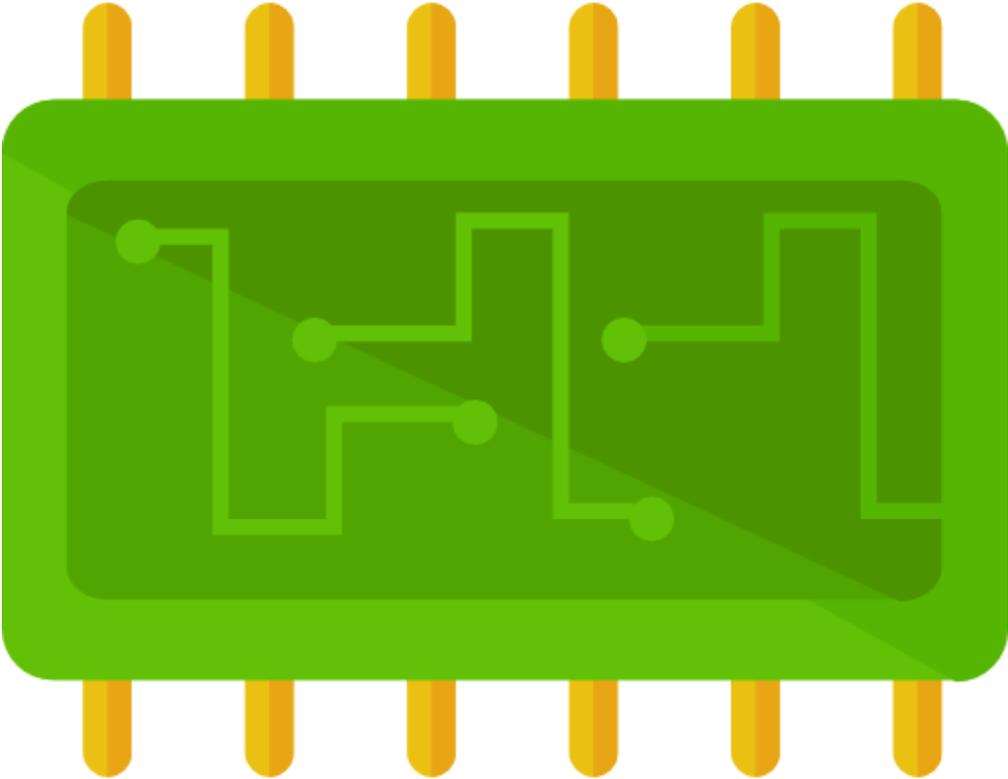
Verificar

Otro ejercicio

Ver respuesta



EPÍLOGO



En la mayoría de sistemas digitales, el número de estados discretos es tan solo de dos y se les denomina niveles lógicos. Estos niveles se representan por un par de valores de voltaje, uno cercano al valor de referencia del circuito (normalmente 0 voltios, tierra o "GND"), y otro cercano al valor dado por la fuente de alimentación del circuito. Estos dos estados discretos reciben muchas parejas de nombres en libros de electrónica y otros textos especializados, siendo los más comunes "0" y "1", "false" y "true", "off" y "on" o "bajo" y "alto" entre otros. Tener solo estos dos valores nos permite usar el álgebra booleana y códigos binarios, los que nos proporciona herramientas muy potentes para realizar cálculo sobre las señales de entrada.

Los más complejos sistemas digitales, aplicados y útiles hoy en día son posibles gracias a la integración de los componentes, herramientas, equipos y subsistemas electrónicos, informáticos y mecánicos. En tiempos modernos es tan fácil tocar una pantalla con nuestras manos (pantalla táctil), ejecutar un comando de voz y cambiar un canal o abrir una ventana, apagar y encender una bombilla; todo gracias a la electrónica digital. Como su nombre lo indica ella se sustenta en su propio lenguaje, el lenguaje de código binario "1" y "0", se crean ciclos de palabras, password, secuencias de bit y byte y se hace realidad lo que nunca se pensó poder monitorear en tiempo real un proceso a miles de kilómetros de distancia. Todas las demás ciencias hoy en día se deben a la invención de los sistemas digitales, es difícil pensar en cocinar algo, llamar a un pariente lejano o ir al cine sin dejar a un lado la electrónica digital (<https://es.wikipedia.org/>).

Este libro es un pequeño aporte a esos sistemas digitales, con el propósito de acercar al estudiante al lenguaje digital. Para ello, se presentaron conceptos básicos como las tablas de verdad, la lógica booleana, las compuertas lógicas y, finalmente, los mapas de Karnaugh.

Como cierre final, se presentan tres objetos interactivos de aprendizaje, que resumen, en parte, los temas tratados en este libro.

SOPA DE LETRAS

Dando clic en la esquina superior derecha se expande el OIA.

Este OIA consta de una sopa de letras con términos tratados en este LIA (Libro Interactivo de Aprendizaje).

P	N	I	A	N	A	E	L	O	O	B	Ñ	Y	H	U
H	O	Ñ	K	A	R	N	A	U	G	H	Y	Q	C	C
U	H	Ñ	I	R	O	N	X	B	Ñ	L	N	J	S	O
W	M	E	P	Y	Ñ	D	J	A	C	S	Z	A	K	Z
O	U	H	L	U	E	V	V	U	W	R	N	U	J	L
T	A	I	J	Z	W	I	E	O	D	O	M	I	E	A
V	K	B	F	H	I	G	H	T	Ñ	Y	I	L	L	T
W	V	C	B	L	M	Q	Y	J	Y	O	O	O	J	C
U	O	F	X	U	R	A	F	I	R	J	I	W	Ñ	O
E	H	F	M	I	N	T	E	R	M	F	R	V	Z	Ñ
Ñ	J	Y	U	B	J	Q	K	Z	V	O	A	U	A	B
G	C	F	S	O	C	J	F	L	Q	V	N	E	W	N
O	R	S	B	A	U	K	T	R	G	S	I	E	Z	I
C	I	P	M	R	E	T	X	A	M	L	B	Z	B	W
W	C	F	V	N	I	B	B	L	E	T	O	Q	Q	L

SOPA DE LETRAS

Sopa de letras general del LIA .
Pueden estar en dirección horizontal o vertical, al derecho o al revés.

Muestra palabras

Haz clic en la primera letra de la palabra, luego dirige el ratón a la última letra y vuelve a hacer clic. Palabra colorada es incorrecta, palabra verde es un acierto.

PREGUNTAS SÍ/NO

Dando clic en la esquina superior derecha se expande el OIA.

En este OIA, responde SÍ o NO a cada "Interrogante" que se realiza.



Haz clic sobre la respuesta, SI/NO, en cada caso

La compuerta lógica que realiza sumas lógicas es la OR.	Sí	No	0.00
BCD significa Binario Codificado en Decimal.	Sí	No	
La familia lógica que se alimenta exclusivamente con +5V es la CMOS.	Sí	No	
Un mapa K de 3 variables NO se puede doblar horizontalmente.	Sí	No	

Verificar

PUZLE GIRATORIO

Dando clic en la esquina superior derecha se expande el OIA.

Actividad lúdica, que permite usar imágenes enlazadas a Flickr.



Haz clic sobre las piezas del puzle, hasta armar la imagen
Las imágenes de Flickr demoran unos segundos más en cargar

Imágenes desde Flickr

Electrónica Digital
Integrado Decimal
Logic gate Lógica

Otra imagen local

