ALGEBRA BOOLEANA

FUNDAMENTOS DE DISEÑO DIGITAL
SISTEMAS DIGITALES

En 1849, George Boole publico un esquema de la descripción algebraica de los procesos relativos al pensamiento y al razonamiento lógico, basado en dos elementos “verdadero” y “falso”. Luego ese esquema y sus posteriores refinamientos recibieron el nombre de ***álgebra Boolena***. Fue casi 100 años después que esta álgebra halló aplicación en la ingeniería. A fines de la decada de 1930, Claude Shannon demostró que el álgebra Boolena constituye un medio eficaz para describir circuitos basados en la conmutación de dos valores a través de interruptores.

OPERACIONES BOOLEANAS BASICAS

NOT, INVERSOR, COMPLEMENTO (¯ , ´)

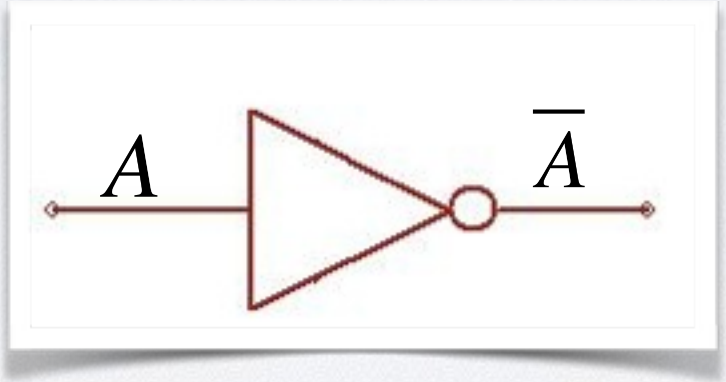
Sólo una entrada

Función Booleana o Lógica: $f(A) = \overline{A} = A´$

Tabla de Verdad:

A	F
0	1
1	0

Compuerta lógica:



VHDL

F<= NOT A;

En el álgebra de Booleana cualquier variable sólo puede tener dos valores: 0 y 1 log.

Nota: La Tabla de verdad describe la información relacionada con todas las combinaciones de entrada a la función lógica y el valor de la salida para cada una de éstas.

OPERACIONES BOOLEANAS BASICAS

OR (+)

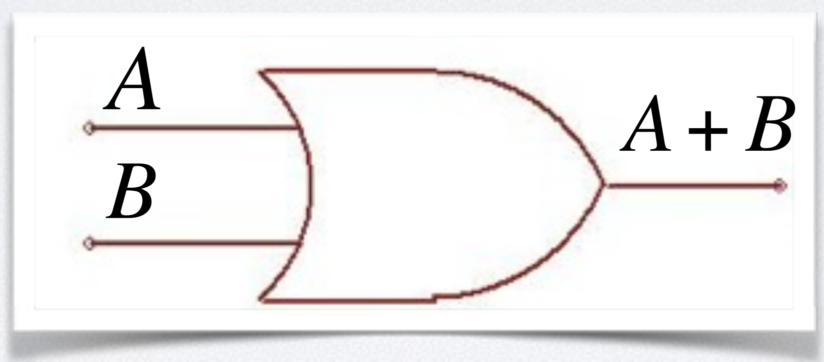
Minimo dos entradas

Función Booleana o Lógica: $f(AB) = A + B$

Tabla de Verdad:

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

Compuerta lógica:



VHDL

F<= A OR B;

En el álgebra de Booleana cualquier variable sólo puede tener dos valores: 0 y 1 log.

OPERACIONES BOOLEANAS BASICAS

AND ($\cdot$,la ausencia de signo)

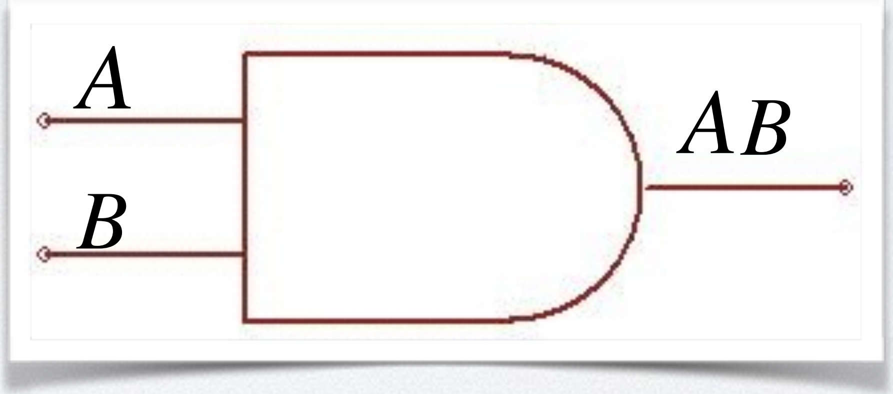
Minimo dos entradas

Función Booleana o Lógica: $f(AB) = A \cdot B = AB$

Tabla de Verdad:

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

Compuerta lógica:



VHDL

F<= A AND B;

OPERACIONES BOOLEANAS BASICAS

NOR (NOT & OR)

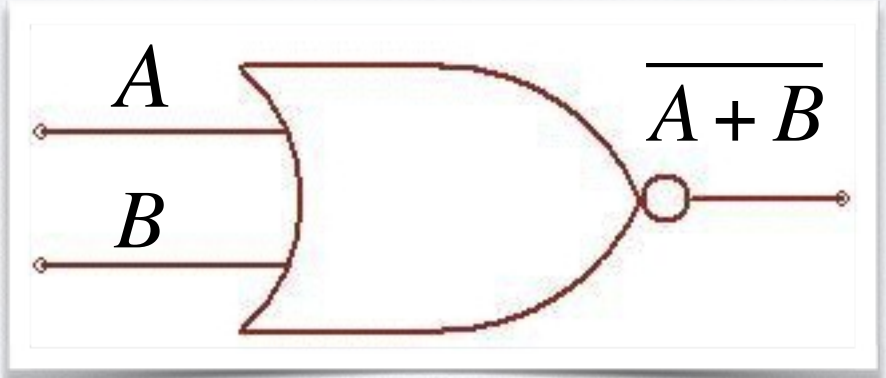
Minimo dos entradas

Función Booleana o Lógica: $f(AB) = \overline{A+B} = (A+B)'$

Tabla de Verdad:

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

Compuerta lógica:



VHDL

F<= A NOR B;

OPERACIONES BOOLEANAS BASICAS

NAND (NOT & AND)

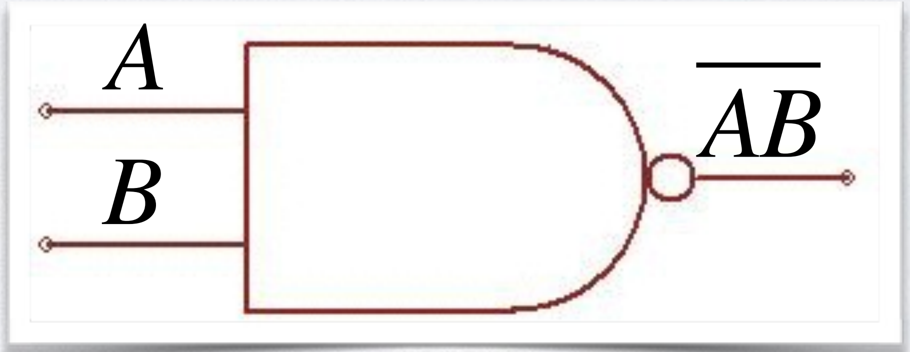
Minimo dos entradas

Función Booleana o Lógica: $f(AB) = \overline{AB} = (AB)'$

Tabla de Verdad:

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

Compuerta lógica:



VHDL

F<= A NAND B;

OPERACIONES BOOLEANAS BASICAS

XOR o OR-EXCLUSIVA

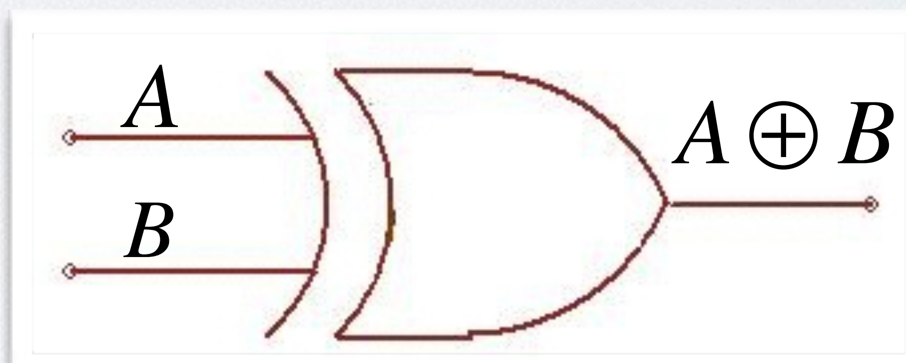
Sólo dos entradas

Función Booleana o Lógica: $f(AB) = \overline{A}B + A\overline{B} = A \oplus B$

Tabla de Verdad:

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

Compuerta lógica:



VHDL

```
F<= A  XOR B;
```

XNOR o NOR-EXCLUSIVA

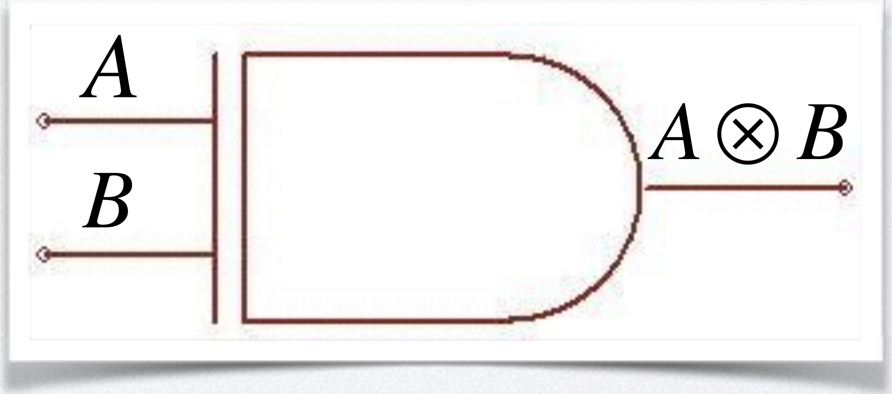
Sólo dos entradas

Función Booleana o Lógica: $f(AB) = \overline{A}\,\overline{B} + AB = A \otimes B$

Tabla de Verdad:

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

Compuerta lógica:



VHDL

F<= A XNOR B;

Teoremas del álgebra Boolena

$1.-\ x \cdot 0 = 0$

$2.-\ x + 1 = 1$

$3.-\ x \cdot 1 = x$

$4.-\ x + 0 = x$

$5.-\ x \cdot x = x$

$6.-\ x + x = x$

$7.-\ x \cdot \overline{x} = 0$

$8.-\ x + \overline{x} = 1$

$9.-\ \overline{\overline{x}} = x$

Propiedad conmutativa

$10.-\ x \cdot y = y \cdot x$

$11.-\ x + y = y + x$

Propiedad asociativa

$12.-\ x \cdot (y \cdot z) = (x \cdot y) \cdot z = xyz$

$13.-\ x + (y + z) = (x + y) + z = x + y + z$

Teoremas del álgebra Boolena

Propiedad distributiva

$$14.-\ x\cdot(y+z)=xy+xz$$

$$15.-\ x+yz=(x+y)\cdot(x+z)$$

Propiedad de combinación

$$18.-\ xy+x\overline{y}=x$$

$$19.-\ (x+y)\cdot(x+\overline{y})=x$$

Propiedad de absorción

$$16.-\ x+xy=x$$

$$17.-\ x(x+y)=x$$

Teorema de DeMorgan

$$20.-\ \overline{xy}=\overline{x}+\overline{y}$$

$$21.-\ \overline{x+y}=\overline{x}\cdot\overline{y}$$

Teoremas del álgebra Boolena

Propiedad de consenso

$$22.-\ x+\overline{x}y=x+y$$

$$23.-\ x(\overline{x}+y)=xy$$

$$24.-\ xy+yz+\overline{x}z=xy+\overline{x}z$$

$$25.-\ (x+y)\cdot(x+z)\cdot(\overline{x}+z)=(x+y)\cdot(\overline{x}+z)$$

Ejemplo.- de la siguiente función obtenga:

1. el diagrama con compuertas
2. la tabla de verdad
3. la expresión mínima
4. las formas canónicas
5. el diagrama usando sólo compuertas NAND de 2 entradas

Formas Canónicas de las funciones Booleanas

- Una variable boolenan o lógica puede tomar dos valores diferentes que pueden representarse como ***0*** y ***1***.
- Una función booleana de una o más variables representa una relación lógica entre dichas variables.
- Un termino consta de una o más variables en su forma normal o complementada unidas por una operación ***AND***.
- Un minitérmino (o término producto) es un termino que contiene todas las variables de la función unidas por la operación ***AND***.
- Se le denomina minitérmino por que solamente para una de las 2^n posibles combinaciones de los valores de las variables toma el valor de 1.
- Un maxitérmino (o término suma), es realmente, una operación ***OR*** de de n términos, cada uno de los cuales contiene solamente una de las variables de la función.
- Se le llama maxitérmino por que solamente para cada una de las 2^n combinaciones de los valores de las variables toman el valor de 0.
- Existen dos formas canónicas únicas con las que se puede representar cualquier función boolana.
- Función canónica en minitérminos en la cual se expresan los unos de la función (minitérminos) en forma de una suma de productos.
- Función canónica en maxitérminos en la cual se expresan los ceros de la función (maxitérminos) en forma de un producto de sumas.

Formas Canónicas de las funciones Booleanas

A	B	C	Miniterinos				F
0	0	0	$\overline{A}$	$\overline{B}$	$\overline{C}$	m	1
0	0	1	$\overline{A}$	$\overline{B}$	C	m	0
0	1	0	$\overline{A}$	B	$\overline{C}$	m	0
0	1	1	$\overline{A}$	B	C	m	1
1	0	0	A	$\overline{B}$	$\overline{C}$	m	0
1	0	1	A	$\overline{B}$	C	m	1
1	1	0	A	B	$\overline{C}$	m	0
1	1	1	A	B	C	m	1

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

Ejemplo

$$F(ABC) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}BC + A\overline{B}C + ABC$$

$$F(ABC) = m_0 + m_3 + m_5 + m_7$$

$$F(ABC) = \sum m(0,3,5,7)$$

Formas Canónicas de las funciones Booleanas

A	B	C	Maxiterminos		F
0	0	0	$A + B + C$	M	1
0	0	1	$A + B + \overline{C}$	M	0
0	1	0	$A + \overline{B} + C$	M	0
0	1	1	$A + \overline{B} + \overline{C}$	M	1
1	0	0	$\overline{A} + B + C$	M	0
1	0	1	$\overline{A} + B + \overline{C}$	M	1
1	1	0	$\overline{A} + \overline{B} + C$	M	0
1	1	1	$\overline{A} + \overline{B} + \overline{C}$	M	1

$\overline{A}=1$

$A=0$

Ejemplo

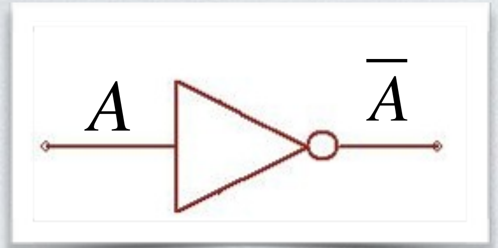
$$F(ABC) = (A+B+\overline{C})\cdot(A+\overline{B}+C)\cdot(\overline{A}+B+C)\cdot(\overline{A}+\overline{B}+C)$$

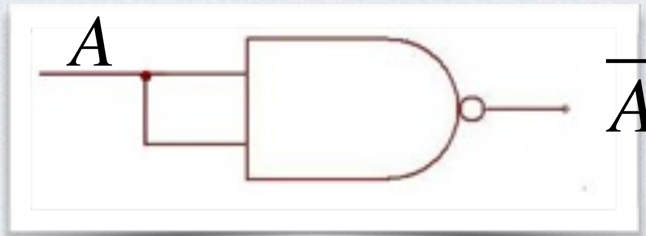
$$F(ABC) = M_1 \cdot M_2 \cdot M_4 \cdot M_6$$

$$F(ABC) = \Pi M(1,2,4,6)$$

EQUIVALENCIAS DE LAS COMPUERTAS BASICAS A NAND DE DOS ENTRADAS

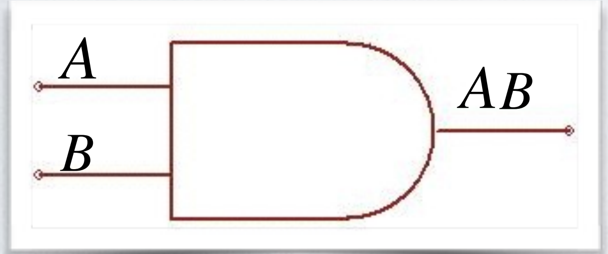
NOT

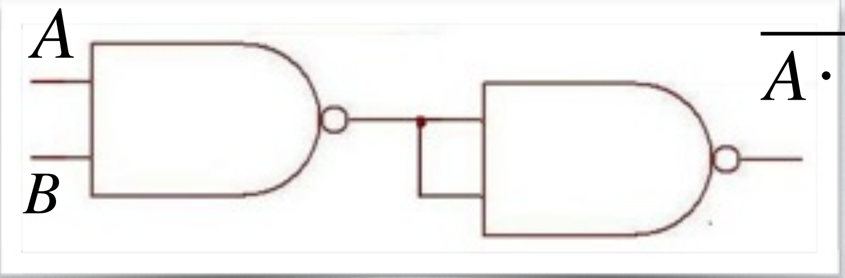




$$\overline{A\cdot A} = \overline{A}$$

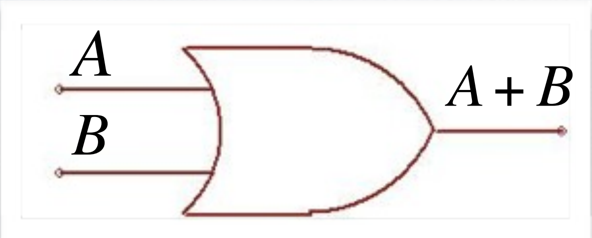
AND

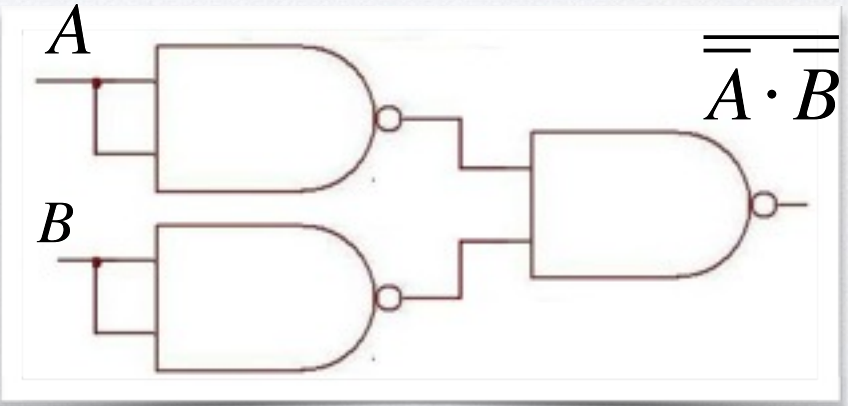




$$\overline{A\cdot B} = \overline{\overline{A\cdot B}} = A\cdot B$$

OR





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

ESCALA DE INTEGRACIÓN

Los CI digitales se clasifican de acuerdo a la complejidad del circuito que implementa, que se estima por el numero de compuertas lógicas equivalentes NAND de dos entradas en el sustrato. De tal manera que cualquier circuito lógico puede ser convertido a un equivalente que sólo utilice NAND de dos entradas, el resultado de la suma de las compuertas lógicas equivalentes se puede catalogar en alguna de las escalas de Integración.

ESCALAS DE INTEGRACIÓN	**NUMERO DE COMPUERTAS EQUIVALENTES**
Small Scale Integration **(SSI)** Pequeña Escala de Integración	1-9
Medium Scale Integration **(MSI)** Mediana Escala de Integración	10-99
Large Scale Integration **(LSI)** Gran Escala de Integración	100 a 9999
Very Large Scale Integration **(VLSI)** Muy Grande Escala de Integración	10 000 a 99 999
Application Specific Integrated Circuits **(ASIC)** Circuitos Integrados de Aplicación Especifica	más de 100 000

MAPAS DE KARNAUGH

Un mapa de Karnaugh es una cuadricula con 2^n divisiones donde cada división representa una de las 2^n combinaciones de la tabla de verdad para n variables con la única condición de que cada división sea ***adyacente*** con todas las que tenga a los lados.

El mapa se llenará con los minitérminos o maxitérminos correspondientes a la función a minimizar.

Se formarán conjuntos de mini o maxi términos, cuya cantidad de términos sea potencia de dos (1,2,4,8,16, etc.). La finalidad de los conjuntos es encontrar las variables que se complementan y poderlas eliminar de la función minina a través de la relación: $x + \overline{x} = 1$

MAPAS DE DOS VARIABLES

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

A \ B	0	1
0	C X_0	C X_1
1	C X_2	C X_3

X → minitérminos o maxitérminos

ESCOM
Escuela Superior de Cómputo

MAPAS DE TRES VARIABLES

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

A \ BC	00	01	11	10
0	C X_0	C X_1	C X_3	C X_2
1	C X_4	C X_5	C X_7	C X_6

AB \ C	0	1
00	C X_0	C X_1
01	C X_2	C X_3
11	C X_6	C X_7
10	C X_4	C X_5

X → minitérminos o maxitérminos

MAPAS DE CUATRO VARIABLES

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

AB \ CD	00	01	11	10
00	X_0 C	X_1 C	X_3 C	X_2 C
01	X_4 C	X_5 C	X_7 C	X_6 C
11	X_{12} C	X_{13} C	X_{15} C	X_{14} C
10	X_8 C	X_9 C	X_{11} C	X_{10} C

X → minitérminos o maxitérminos