ALGEBRA BOOLEANA

Son las matemáticas de los sistemas digitales.

Historia

Su nombre se debe a George Boole, matemático inglés que en 1854 presentó un sistema algebraico de dos valores titulado "investigación de las leyes del pensamiento sobre las que se basan las teorías matemáticas de la lógica y la probabilidad", ahora se conoce como algebra booleana o algebra de conmutación.
En 1938 Claude E. Shannon investigador de laboratorios Bell, presentó un trabajo para graduarse en postgrado en el MIT en el cual describía como el álgebra booleana se adaptaba a la representación y al diseño de circuitos de conmutación basados en relés e interruptores al cual le llamó "Análisis simbólico de los circuitos de conmutación y relés".
Más tarde los dispositivos para realizar esos trabajos los fabricaron con materiales semiconductores y se llaman compuertas lógicas, están constituidas por resistencias, diodos y transistores (bipolares o unipolares).

Constantes y variables booleanas o lógicas

En algebra booleana las entradas y salidas de un circuito digital se representan mediante caracteres alfabéticos en mayúsculas llamados variables booleanas o lógicas.

Las variables y constantes booleanas son binarias, o sea que solo tienen valores 0 o 1.
En electrónica digital, una variable booleana representa el **nivel de voltaje** presente en un punto de un circuito, el uno designa un nivel alto y el cero un nivel bajo.
En la lógica digital se usan también otros términos como sinónimos de 0 y 1 como se muestra en la siguiente tabla.

Tabla 1.

0 lógico	1 lógico
Falso	Verdadero
Apagado	Encendido
Bajo	Alto
No	Sí
Interruptor abierto	Interruptor cerrado

Las variables booleanas se combinan para formar ecuaciones booleanas o lógicas.
Una ecuación booleana es una **expresión matemática que sintetiza el funcionamiento** de un circuito digital.
La ecuación booleana Q = (A.B.C.D)' + B'.C consta de tres elementos: variables de entrada (A a D), variable de salida (Q) y operadores lógicos.

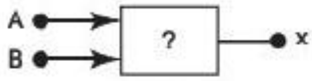
Tabla de verdad

Una tabla de verdad es una herramienta para describir completamente la forma en que la salida de un circuito lógico depende de los niveles lógicos presentes en las entradas del circuito.

Salida

Entradas

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0



(a)

A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

Figura 1

Operadores booleanos o lógicos

Son signos que relacionan entre si las variables de entrada y establecen su relación con la variable de salida, esos operadores son: OR (+), AND (.) y NOT (-).
En la ecuación booleana anterior se ven dichos operadores, el signo = establece la equivalencia entre el estado lógico de las salidas y el de las entradas.

Operaciones booleanas o lógicas

Son de 2 tipos: básicas y derivadas.

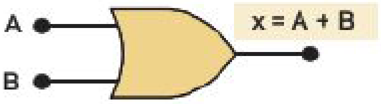
Operaciones básicas

Son la OR, AND y NOT.

Operación OR o suma lógica

Esta operación produce como resultado un 1 lógico cuando una **o** más variables implicadas en la operación sea 1 lógico. Esta operación es similar a la suma decimal excepto en aquellos casos donde más de una variable sea 1 lógico, por eso también se conoce como suma lógica.
X = A + B es la ecuación que representa a esa operación y se lee: **X es igual a A OR B.**
En circuitos digitales esta operación la realiza un dispositivo digital conocido como **compuerta OR** la cual puede tener 2 o más entradas.
Funciona de modo que cuando en una o más entradas se aplica un voltaje alto (1 lógico), en la salida se presenta un voltaje alto (1 lógico).
El análisis de la operación y la compuerta OR se presenta mediante: a) símbolo lógico, b) ecuación booleana, c) tabla de verdad y d) circuito eléctrico equivalente.

Símbolo lógico   Ecuación booleana   Tabla de verdad   Circuito eléctrico equivalente



X = A + B

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Operación AND o producto lógico y operación NOT

Estudiar uds. estos temas.

Ejemplos:
Tocci: 3.2 a 3.5 B
Floyd: 3.1 a 3.8

Ejercicios: dar solución a alguno de los siguientes problemas.
Tocci: preguntas de repaso (páginas 90, 93 y 94).
problemas al final del capítulo: 3-1 a 3-6, 3-8 a 3-11.
Floyd: problemas relacionados (páginas 126, 129, 131, 132 y 137), revisión de la sección 3.1, 3.2 y 3.3, problemas al final del capítulo: 1 a 10.

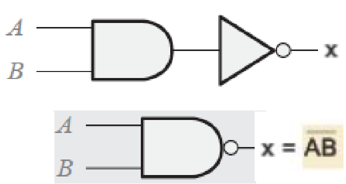
Operaciones derivadas

Se obtienen a partir de las operaciones básicas y son: NOR, NAND, XOR y XNOR.

Operación NAND

Esta operación complementa a la operación AND, produce como resultado un 1 lógico cuando una o más variables implicadas en la operación sea 0 lógico.
X = (A . B)' es la ecuación que representa a esa operación y se lee: **X es igual a A AND B todo negado o X es igual al inverso de A AND B o X es igual al complemento de A AND B.**
En circuitos digitales esta operación la realiza un dispositivo digital conocido como **compuerta NAND** la cual puede tener 2 o más entradas (hasta 13).
Funciona de modo que cuando en una o más entradas se aplica un voltaje bajo (0 lógico), en la salida se presenta un voltaje alto (1 lógico).
El análisis de la operación y la compuerta NAND se presenta mediante: a) símbolo lógico, b) ecuación booleana, c) tabla de verdad y d) circuito eléctrico equivalente.

Símbolo lógico | Ecu. booleana | Tabla de verdad | Circuito eléctrico equivalente



$$x = \overline{AB}$$

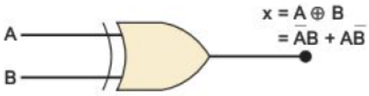
A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

Operación XOR (exclusive OR)

Esta operación se realiza entre 2 variables y excluye algo de la operación OR, produce como resultado un 1 lógico cuando las 2 variables implicadas en la operación tengan valores lógicos diferentes.

$x = \overline{A}B + A\overline{B} = A \oplus B$ es la ecuación que representa a esa operación y se lee: **X es igual a A OR exclusive B.**
En circuitos digitales esta operación la realiza un dispositivo digital conocido como **compuerta XOR** la cual tiene solo 2 entradas.
Funciona de modo que cuando en sus entradas se aplican voltajes diferentes, en la salida se presenta un voltaje alto (1 lógico).
El análisis de la operación y la compuerta XOR se presenta mediante: a) símbolo lógico, b) ecuación booleana, c) tabla de verdad y d) circuito eléctrico equivalente.

Símbolo lógico | Ecu. Booleana | Tabla de verdad | Circuito lógico equivalente



$$x = A \oplus B$$
$$x = \overline{A}B + A\overline{B}$$

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

Operaciones NOR y XNOR.

Estudiar uds. estos temas.

Les recomiendo los videos sobre compuerta XOR “Aplicaciones interesantes Puerta XOR O-exclusiva (clase 10.1) y “Utilidad que no sabes de la Puerta XOR (O exclusiva). Clase 10.2

Ejemplos:
Tocci: 3-8, 3-10 y 4-16.
Floyd: 3.9, 3.10 3.13 a 3.15 y 3.18 a 3.20.

Ejercicios: dar solución a alguno de los siguientes problemas
Tocci: problemas al final del capítulo: 3-17 y 3-18.
Floyd: problemas relacionados (página 141, 142, 147, 150, 152 y 154), revisión de las secciones 3.4, 3.5 y 3.6, problemas al final del capítulo 11 a 13, 15 a 17 y 19 a 22.

Descripción algebraica de circuitos lógicos.
Evaluación de las salidas de los circuitos lógicos.
Implementación de circuitos a partir de expresiones boleanas.

Estudiar uds. estos temas.

Ejemplos
Tocci: páginas 95, 96, 98 y 100, ejemplos: 3-6, 7 y 11.
Floyd: páginas 213 y 213.

Ejercicios: dar solución a algunos de los siguientes problemas.
Tocci: preguntas de repaso (páginas 96 ,99, 101 y 104, problemas al final del capítulo: 3-12 a 16, 19 a 21, 38 y 41.
Floyd: revisión de la sección 4.4 (página 213), problemas al final del capítulo: 12 a 16.
Wakerly: problemas al final del capítulo: 4.7 y 8.

TEOREMAS BOOLEANOS

También son conocidos como **reglas, relaciones** o **leyes**, muy sencillos e importantes, de modo que cuando se entienden y aplican correctamente, contribuyen a simplificar las ecuaciones booleanas y con ello a minimizar el número de compuertas y por lo tanto de C.I. requeridos para implementar los circuitos digitales.

Teoremas de una variable

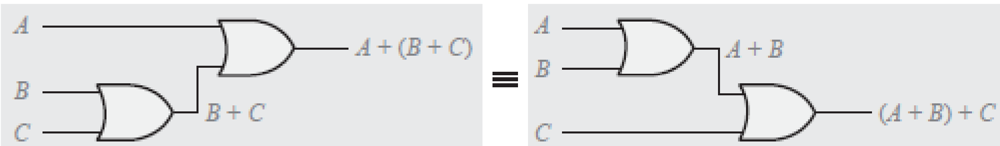
Para la operación OR	Para la operación AND	Para la operación NOT
1) A+0 = A	5) A0 = 0	9) (A’)’ = A
2) A+1 = 1	6) A1 = A	
3) A+A = A	7) AA = A	
4) A+A’ = 1	8) AA’ = 0	

Los teoremas 1 y 6 se conocen como leyes de identidad.
Los teoremas 2 y 5 se conocen como leyes de los elementos dominantes.
Los teoremas 3 y 7 se conocen como leyes de tautología o de idempotencia.
Los teoremas 4 y 8 se conocen como leyes de los complementos.
El teorema 9 se conoce como ley de doble negación o ley involutiva.

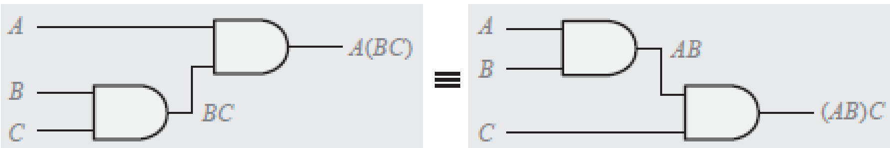
Ejemplo: X = AB’(AB’)’; si AB’ = C entonces X = AB’(AB’)’ = C (C’) = 0

Teorema de dos o más variables

10) A+B = B+A; 11) AB = BA, son conocidas como leyes conmutativas.
12) A+B+C = (A+B)+C = A+(B+C) = (A+C)+B

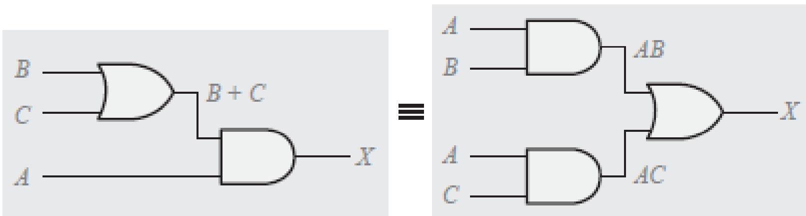


13) ABC = (AB)C = A(BC) = (AC)B



Son conocidas como leyes asociativas

14) AB + AC = A(B+C); 15) (A+B)(C+D) = AC + AD + BC + BD.

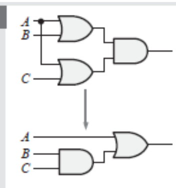


$X = A(B + C)$ $X = AB + AC$

16) A + BC = (A+B)(A+C).

A	B	C	$A+B$	$A+C$	$(A+B)(A+C)$	BC	$A+BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

igual



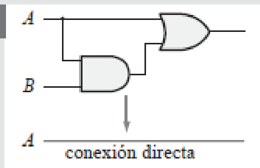
Son conocidas como leyes distributivas y operan de la misma forma que la factorización del algebra decimal.

Ejemplos:
1) ABC + ABD = AB(C+D)
2) AB’C + A’B’C = B’C(A+A’) = B’C(1) = B’C
3) (A’+B)(A+B) = A’A + B = 0+B = B. También se puede usar el teorema 15).

17) A + AB = A

A	B	AB	$A+AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

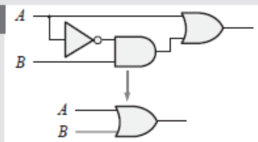
igual



18) A + A’B = A+B

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

igual



19) A’ + AB = A’+B

Son conocidas como leyes de absorción.

20) AB + A’C + BC = AB + A’C es conocido como teorema del consenso generalizado

Ejemplos:
1) X = ACD + A’BCD = CD(A+A’B) = CD(A+B)
2) Z = ABC + ABC’ + AB’C = AB(C+C’) + AB’C = AB(1)+AB’C = AB + AB’C = A(B+B’C) = A(B+C)
3) X = A’B’C’ + A’BC + ABC + AB’C’ + AB’C. Para evitar el tener que usar los teoremas de absorción se procede como en el ejemplo 4-2 método 2 del libro de Tocci, se usa dos veces el término ABC (ABC = ABC + ABC).
X = (A’B’C’ + AB’C’) + (A’BC + ABC) + (AB’C + ABC)
X = B’C’(A’ + A) + BC(A’ + A) + AC(B’ + B) = B’C’(1) + BC(1) + AC(1) = B’C’ + BC + AC.

Ejemplos de libros
Tocci: página 106, ejemplos 3-13, 15, 4-2, 4 a 6.
Floyd: 4.8 a 10.

Ejercicios: dar solución a algunos de los siguientes problemas.
Tocci: preguntas de repaso (página 108), problemas al final del capítulo: 3-24, 4-1 a) a c) y e).
Floyd: problemas relacionados (páginas 214 a 216), revisión de la sección 4.5 (página 217)
problemas al final del capítulo: 17, 18 sin c) y 19 sin b) y d).

Wakerly: problemas al final del capítulo: 4.6.

TEOREMAS DE DeMORGAN

Son dos de los teoremas más importantes del algebra booleana para las operaciones NAND y NOR, fueron propuestos por el matemático inglés Augustus DeMorgan contemporáneo de Boole nacido en la India.

Para la operación NAND

21) (A.B)’ = A’ + B’

Ejemplos:
1) Y = (A+B + C’D)’ = A’(B’)’(C’D)’ = A’B[(C’)’ + D’] = A’B[C + (D)’] = A’BC + A’BD’.
2) [A(B+C)’D] = A’ + [(B+C)’]’+D’ = A’ + (B+C)+D’ = A’ + B + C + D’.

Para la operación NOR

22) (A+B)’ = A’.B’

Ejemplos:

1) Z = [(A+B)C’]’ = (A+B)’ + (C’)’ = A’B’ + C.
2) [(M+N’)(M’+N)]’ = (M+N’)’ + (M’+N)’ = M(N’)’ + (M’)’N’ = M’N + MN’.

FORMAS DE REPRESENTAR LAS EXPRESIONES O ECUACIONES BOOLEANAS

Se pueden representar de dos formas distintas: 1) Suma de productos (SOP) y
2) Producto de sumas (POS).

Suma de productos (SOP)

Son expresiones que consisten en dos o más términos AND (producto lógico) a los que se les aplica la operación OR (suma lógica). Las variables de los términos AND pueden ser no complementadas (normales) o complementadas, pero el operador NOT no puede cubrir más de una variable. Es la representación más usada.

1. $ABC + \overline{A}B\overline{C}$
2. $AB + \overline{A}B\overline{C} + \overline{C}\,\overline{D} + D$
3. $\overline{A}B + C\overline{D} + EF + GK + H\overline{L}$

Producto de sumas (POS)

Son expresiones que consisten en dos o más términos OR (suma lógica) a los que se les aplica la operación AND (producto lógico). Las variables de los términos OR pueden ser no complementadas (normales) o complementadas, pero el operador NOT no puede cubrir más de una variable.

1. $(A + \overline{B} + C)(A + C)$
2. $(A + \overline{B})(\overline{C} + D)F$
3. $(A + C)(B + \overline{D})(\overline{B} + C)(A + \overline{D} + \overline{E})$

UNIVERSALIDAD DE LAS COMPUERTAS NAND Y NOR

Todas las expresiones booleanas consisten de combinaciones de las operaciones básicas OR, AND y NOT. Por lo tanto, dichas expresiones se pueden implementar con combinaciones de compuertas OR, AND y NOT. Sin embargo, también es posible obtener las operaciones básicas con operaciones NAND o NOR y con ello implementar las expresiones sólo con compuertas NAND o NOR.

Universalidad de la compuerta NAND

Con combinaciones de esta compuerta inversora se pueden implementar las operaciones OR, AND y NOT.

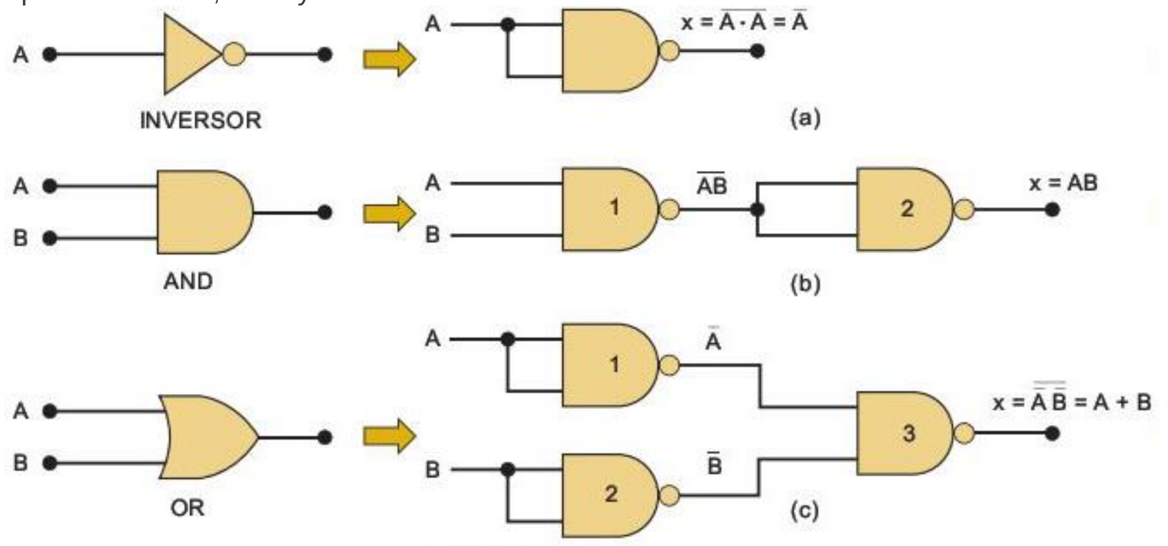


FIGURA2 Las compuertas NAND pueden usarse para implementar cualquier función booleana.

Universalidad de la compuerta NOR

REPRESENTACIONES ESTÁNDAR DE LAS FUNCIONES LÓGICAS

Estudiar uds. estos temas.

Ejemplos:
Tocci: página 108, ejemplo 3-16, página 109, ejemplo 3-17 y 3-18, 4-1 y 4-3.
Floyd: ejemplo 4.3 y 4, página 209, ejemplo 4.5 a 4.7 y 4.11, 5.7 a 5.9.

Ejercicios:
Tocci: preguntas de repaso (páginas 111, 114 y 149), problemas al final del capítulo: 3-26,
28 a 31, 4-1 d) y f) a h).
Floyd: problemas relacionados (páginas 208 a 211, 217, 289 y 291), revisión de la sección
4.3 y 5, 5.3 y 4, problemas al final del capítulo: 18 a 25.