



Tema 3.1

Introducción a los circuitos combinacionales. Algebra de Boole



Índice

- Algebra de Boole. Definición.
- Operaciones lógicas: OR, AND, XOR y NOT
- Puertas lógicas
- Algebra de Boole
 - Postulados
 - Teoremas
 - Funciones lógicas: formas canónicas.
- Tablas de verdad
- Realización de funciones en puertas NAND y NOR

Bibliografía

- Fundamentos de sistemas digitales.
Thomas Floyd.
Prentice-Hall.
- Fundamentos de diseño lógico y computadoras.
M. Morris Mano.
Prentice-Hall

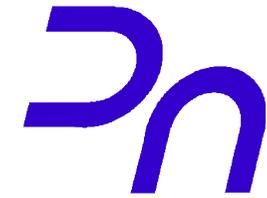
Algebra de Boole.

Definición



- Algebra de Boole es todo conjunto de elementos capaz de adoptar dos valores (**0** y **1**).
- Cada uno de dichos elementos recibe el nombre de **Variable lógica**
- Están definidas dos operaciones: suma lógica y producto lógico (**+** y *****).





Operaciones lógicas

▪ OR

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

▪ NOT

a	NOT a
0	1
1	0

▪ AND

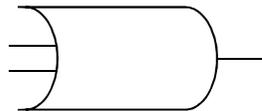
a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1

▪ XOR

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

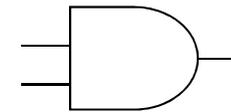


Puertas lógicas (I)



a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

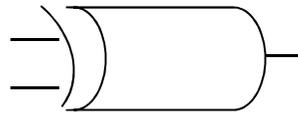
Se representa como: **$a + b$**



a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1

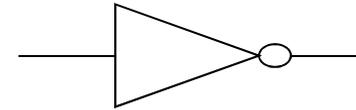
Se representa como: **$a \cdot b$**

Puertas lógicas (II)



a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Se representa como: $a \oplus b$



a	NOT a
0	1
1	0

Se representa como: \bar{a}

Algebra de Boole. Postulados



- El Algebra de Boole cumple los siguientes postulados:
- 1. Propiedad conmutativa.
 - $a + b = b + a$ $a * b = b * a$
- 2. Propiedad distributiva
 - $a*(b+c) = a*b + a*c$
 - $a+(b*c) = (a+b) * (a+c)$
- 3. Elemento neutro
 - $a + 0 = a$ $a * 1 = a$
- 4. Elemento Simétrico (complementario):
 - $a + \bar{a} = 1$ $a * \bar{a} = 0$



Algebra de Boole. Teoremas



- Se demuestran a partir de los 4 postulados anteriores.
- 1. Ley de idempotencia:
 - $a+a = a$ $a*a = a$
- 2. Ley de complemento:
 - $a + \bar{a} = 1$ $a * \bar{a} = 0$
- 3. Ley conmutativa:
 - $a + b = b + a$ $a * b = b * a$
- 4. Ley asociativa:
 - $a+(b+c) = (a+b)+c$ $a*(b*c) = (a*b)*c$
- 5. Ley distributiva:
 - $a*(b+c) = a*b + a*c$ $a+(b*c) = (a+b) * (a+c)$



Algebra de Boole. Teoremas



- 6. Ley de cancelación:
 - $(a*b)+a = a$ $(a+b)*a = a$
- 7. Ley de identidad:
 - $a+0 = a$ $a*1 = a$
- 8. Ley de dominación:
 - $a+1 = 1$ $a*0 = 0$
- 9. Ley de doble complemento
 - $\overline{\overline{a}} = a$
- 10. Leyes de Morgan:
 - $\overline{a+b} = \overline{a} * \overline{b}$ $\overline{a*b} = \overline{a} + \overline{b}$
 - Las leyes de morgan junto con la doble negación nos permiten pasar de expresiones en sumas lógicas a expresiones equivalentes en productos lógicos y viceversa

Algebra de Boole. Funciones Lógicas. Formas Canónicas



- **Función lógica:** Expresión de variables booleanas o binarias unidas por las operaciones lógicas suma, producto y complementación. Ejemplo:

$$f_1(c,b,a) = a + c \cdot b + c \cdot b \cdot a$$

- **Término canónico:** Producto o suma en el que aparecen todas las variables (o sus complementos) de que depende una función.
- **Función canónica:** formada exclusivamente por términos canónicos
- **Minterm:** término canónico en forma de producto de variables (ej.: $c \cdot b \cdot a$).
 - **Conversión:** Multiplicar cada término no canónico por la suma de las variables que le falten, en su forma normal y complementada.
- **Maxterm:** término canónico en forma de suma de variables (ej.: $c+b+a$).
 - **Conversión:** Sumar a cada término no canónico productos formados por cada variable que falte y su complementada.





Tablas de verdad

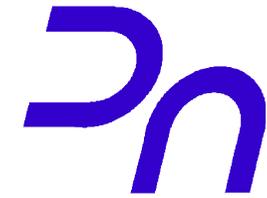
- Es otra forma de representar un función lógica y sirve para obtener el desarrollo en forma canónica de la misma.
- Ejemplo:

$$f(c,b,a) = c \cdot b + \bar{c} \cdot a$$

Tabla de verdad:

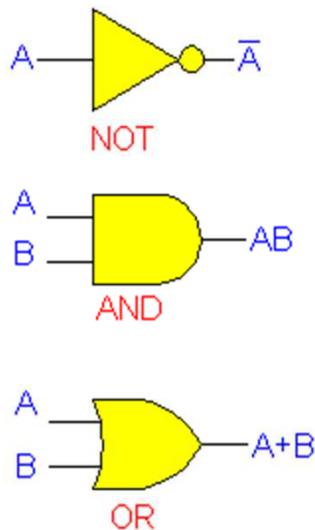
	c	b	a	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1



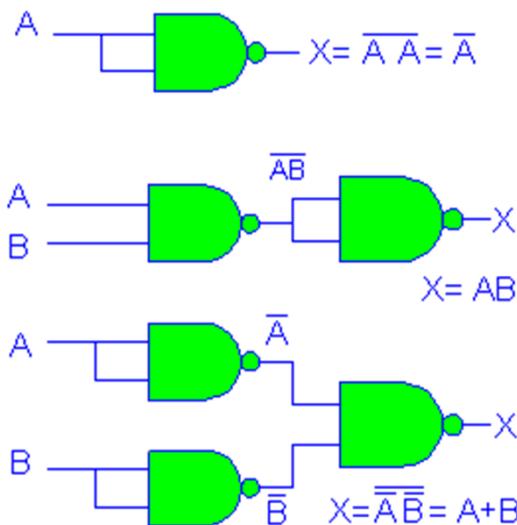


Realización de funciones en puertas NAND y NOR (I)

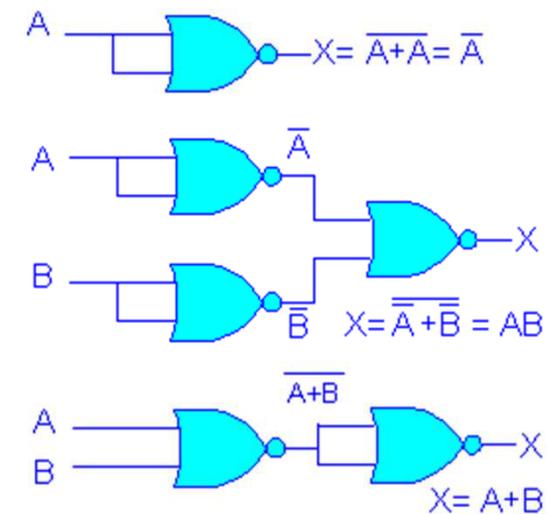
- Las puertas NAND y NOR son puertas Universales, es decir, cualquier función lógica se puede expresar utilizando solo puertas NAND o solo puertas NOR.
- Además son las mas fáciles de construir.



Con NAND



Con NOR



Realización de funciones en puertas NAND y NOR (II)

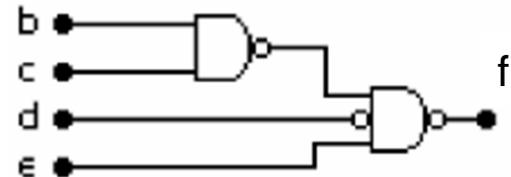
- Para pasar cualquier circuito a puertas NAND o NOR se emplean la ley de la doble negación y las leyes de Morgan.

- Ejemplo: Sea la función $f_1 = bc + d + \bar{e}$

A NAND

- Pasamos sumas a productos

$$f_1 = \overline{\overline{bc + d + \bar{e}}} = \overline{\overline{bc} \cdot \overline{d} \cdot \overline{\bar{e}}} = \overline{\overline{bc} \cdot \overline{d} \cdot e}$$



A NOR

- Pasamos productos a sumas

$$f_1 = \overline{\overline{bc + d} - \bar{e}} = \overline{\overline{bc + d} + e} = \overline{\overline{bc} + \overline{d} + e}$$

