MÓDULO DE MATEMÁTICA

Les damos la bienvenida a este primer paso en la universidad y queremos contarles que este material de matemática ha sido pensado para ayudarles a recuperar y consolidar temas que seguramente adquirieron en el nivel medio y que son la base de muchos temas más complejos que estudiaremos durante el curso de matemática en la carrera.

Juntos y con el acompañamiento de docentes y tutores de la comunidad educativa de la carrera, recordaremos conceptos, desarrollaremos ejercicios, ejemplos, etc.

Es importante comprender que esta nueva etapa que emprenden conlleva una gran responsabilidad y compromiso, sabiendo que sin esfuerzo y dedicación nada es posible, pero que nada es inalcanzable o tan difícil como parece es sólo cuestión de dedicación, paciencia y tiempo de estudio.

Estaremos a cada paso, acompañando siempre en el marco del respeto porque sin respeto no hay comunicación ni convivencia, en definitiva, no hay educación.

Las matemáticas puras son, en su forma, la poesía de las ideas lógicas.

Albert Einstein.

Contenido

Capítulo I. Álgebra de Boole 3

1.1OPERACIONES LÓGICAS Y TEOREMAS 4

1.1.1La complementación lógica 4

1.1.2La suma lógica 5

1.1.3El producto lógico 5

1.2Teoremas 6

1.3PUERTAS LÓGICAS 8

1.3.1Las puertas OR 8

1.3.2Las puertas AND 9

1.3.3La puerta NOT 9

1.3.3Las puertas NOR 10

1.3.4Las puertas NAND 10

1.3.5Las puertas OR EXCLUSIVAS (EOR o XOR) 10

1.3.6Las puertas NOR EXCLUSIVAS (XNOR) 11

1.4 Algunas consideraciones 11

1.4 Ejercitación propuesta del capítulo II 12

1.5 Resolución a la ejercitación propuesta del capítulo II 12

Capítulo I. Álgebra de Boole

Las computadoras tienen su propio sistema de representación. Debido a su construcción basada fundamentalmente en circuitos electrónicos digitales, utiliza un sistema binario. Esto obliga a transformar la representación de nuestra información, tanto numérica como alfanumérica, a una representación binaria para que la máquina sea capaz de procesarlos.

Como se estudiará más adelante, tanto el sistema decimal como el binario están basados en los mismos principios. En ambos, la representación de un número se efectúa por medio de cadenas de símbolos, los cuales representan una determinada cantidad dependiendo de cada símbolo y la posición que ocupa dentro de la cadena con respecto al denominado punto (o coma) decimal.

Por cuestiones de índole técnica, los circuitos electrónicos que conforman una computadora suelen estar capacitados para reconocer señales eléctricas de tipo digital; por lo tanto, se hace necesario que los métodos de codificación internos tengan su origen en el sistema binario, y con ellos se pueda representar todo tipo de informaciones y órdenes que sean manejadas por la computadora.

En los circuitos electrónicos suele representarse la presencia de tensión (electricidad) en un punto de un circuito por medio de un 1, en tanto que 0 representa la ausencia de dicha tensión.

La electrónica digital está fundamentada en la base matemática formada por el álgebra de Boole (George Boole, matemático inglés, 1815-1864). Este método de análisis considera que todos los elementos poseen únicamente dos estados (biestables) o dos valores, verdadero o falso (1 o 0) que son opuestos entre sí, no permitiéndose nunca la adopción de estados intermedios. Estudiando las distintas asociaciones entre ellos se obtienen las leyes generales sobre los procesos lógicos.

Fue Claude Shannon (matemático e ingeniero norteamericano, 1916-2001) quien aplicó estas técnicas de estudio a los circuitos compuestos de elementos que solo pueden adoptar dos estados estables posibles, apareciendo entonces los llamados circuitos lógicos.

Puede decirse entonces que el álgebra de Boole es el sistema matemático empleado en el diseño de circuitos lógicos, que nos permite identificar mediante símbolos el objeto de un circuito lógico de modo que su estado sea equivalente a un circuito real.

2.1OPERACIONES LÓGICAS Y TEOREMAS

Se definen básicamente tres tipos de operaciones sobre las variables del álgebra de Boole o variables booleanas que son: La complementación lógica, la suma lógica y el producto lógico.

2.1.1La complementación lógica

Sea una variable booleana A, que por el hecho de serlo solamente podrá poseer dos estados. Si en un instante determinado posee el estado lógico 1, diremos que su estado inverso o complementado será el 0. Si por el contrario la variable A posee el estado lógico 0, su complemento será el 1. El complemento de una variable A se representa simbólicamente por: $\bar{A}$ (con una barra encima de la variable). Otra forma de indicación puede utilizar el símbolo "⌐" adelante de la variable (⌐A) .

La tabla de verdad de los estados lógicos correspondientes a una variable y a su complementaria o inversa es la siguiente:

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

Tabla de verdad del complemento

2.1.2La suma lógica

La operación lógica suma entre dos o más conjuntos (o variables booleanas) se representa mediante el signo "+". Este resultado puede generalizarse para "n" variables de entrada.
RECUERDE QUE UNA SUMA LÓGICA NO ES UNA SUMA DE NÚMEROS.

La función suma lógicas e define mediante la siguiente tabla de verdad:

+	0	1
0	0	1
1	1	1

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

Tabla de verdad suma lógica

2.1.3El producto lógico

La operación producto entre dos conjuntos se representa mediante el símbolo.●(o * dependiendo de la bibliografía)

La operación producto lógico se define mediante la siguiente tabla de verdad:

•	0	1
0	0	0
1	0	1

A	B	D
0	0	0
0	1	0
1	0	0
1	1	1

Tabla de verdad producto lógico

2.2Teoremas

Teorema 1. El resultado de aplicar cualquiera de las tres operaciones antes definidas, a variables booleanas, es otra variable booleana y además el resultado es único.

Teorema 2. Ley de idempotencia. Tanto la suma como el producto de una variable booleana consigo misma da como resultado la misma variable:

A + A = A

A * A = A

Teorema 3. Ley de involución. Una variable booleana negada dos veces, da como resultado la misma variable:

$$\bar{\bar{A}} = A$$

Teorema 4. Ley conmutativa. Se define respecto a la suma (y al producto) y nos dice que el orden de los sumandos (factores) no altera el resultado:

A + B = B + A

A * B = B * A

Teorema 5. Ley asociativa. Se define respecto a las operaciones suma y producto de la siguiente forma:

Respecto de la suma: A+(B+C) = (A+B)+C = A+B+C

Respecto del producto: A(BC) = (AB)C = ABC

Teorema 6. Ley distributiva.

Respecto de la suma: A+BC = (A+B)(A+C)

Respecto del producto: A(B+C) = AB+AC

Teorema 7. Ley de absorción.

A+AB = A

A(A+B) = A

Teorema 8. Leyes de De Morgan.

$$\overline{A+B} = \bar{A}\,\bar{B}$$
$$\overline{AB} = \bar{A} + \bar{B}$$

(Las leyes de De Morgan pueden ser generalizadas a "n" variables.)

Relaciones importantes que se deducen de las operaciones booleanas y de los teoremas anteriores:

0+A = A

1*A = A

0*A = 0

1+A = 1

Nota: es importante reconocer que llamar a las variables A, B o C es simplemente una elección para dar ejemplos pero que las variables pueden tomar cualquier nombre, <u>porque lo importante son las propiedades que cumplen que no cambian.</u>

Aconsejamos en el caso de los teoremas para una mayor comprensión hacer las tablas de verdad correspondientes y verificar que efectivamente se cumplen.

2.3PUERTAS LÓGICAS

Existe un convenio gráfico para representar dispositivos (electrónicos, hidráulicos, mecánicos, etc.) que lleven a cabo funciones booleanas elementales y que, en función de la combinación o combinaciones diseñadas, se obtendrán funciones más complejas. Las puertas lógicas son dispositivos electrónicos que desarrollan las funciones booleanas y son básicamente: Puertas OR, AND, NOT, NOR, NAND, OR Exclusiva y NOR Exclusiva.

2.3.1Las puertas OR

Desarrollan la suma booleana

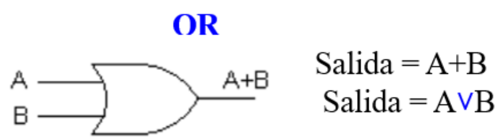


Tabla de verdad puerta OR

Entrada A	Entrada B	Salida $A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

2.3.2Las puertas AND

Desarrollan el producto booleano

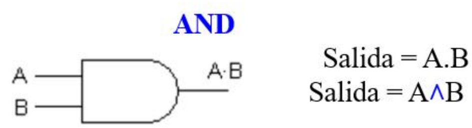


Tabla de verdad puerta AND

Entrada A	Entrada B	Salida $A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

2.3.3La puerta NOT

Realiza la función complementación o inversión booleana

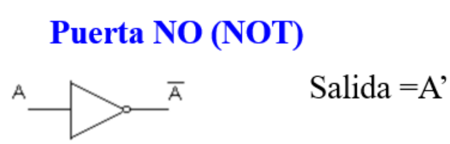


Tabla de verdad puerta NOT

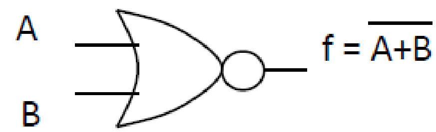
Entrada A	Salida $\overline{A}$
0	1
1	0

2.3.3 Las puertas NOR

Realizan la función inversa de una operación suma lógica, es decir, es la equivalente a una puerta OR complementada. La función lógica será, por tanto:

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

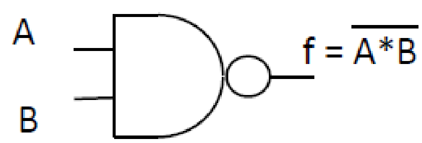
A	B	f
0	0	1
0	1	0
1	0	0
1	1	0



2.3.4 Las puertas NAND

Estas puertas realizan la función lógica $f = \overline{A * B}$

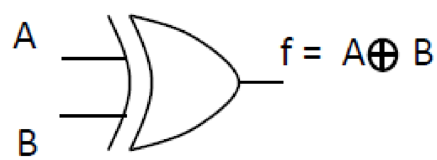
A	B	f
0	0	1
0	1	1
1	0	1
1	1	0



2.3.5 Las puertas OR EXCLUSIVAS (EOR o XOR)

Son puertas que a su salida proporcionan la función lógica $f = A\bar{B} + \bar{A}\,B = A \oplus B$

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

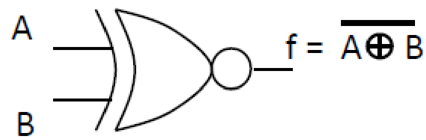


2.3.6Las puertas NOR EXCLUSIVAS (XNOR)

Son puertas que a su salida proporcionan la función lógica

$$f = \overline{A \oplus B} = \overline{A\bar{B} + \bar{A}B} = AB + \bar{A}\,\bar{B}$$

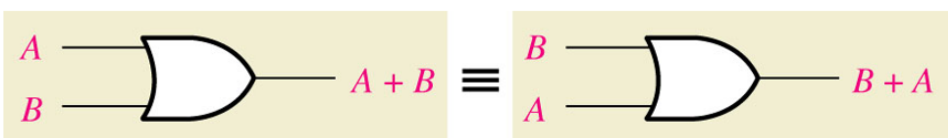
A	B	f
0	0	1
0	1	0
1	0	0
1	1	1



2.4 Algunas consideraciones

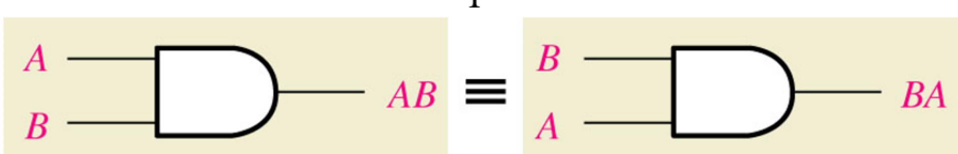
A+B = B+A

El orden en la OR no importa



AB = BA

El orden en la AND no importa



Nota: Durante el cursado de la materia aprenderemos con estas compuertas a interpretar y desarrollar, circuitos lógicos combinados complejos como un conjunto de puertas lógicas interconectadas.

2.4 Ejercitación propuesta del capítulo II

2.4.1 Aplicar los teoremas de De Morgan a las siguientes expresiones

a	$\overline{(A+B+C)D}$
b	$\overline{ABC+DEF}$
c	$\overline{A\bar{B}+\bar{C}D+EF}$
d	$\overline{\overline{(A+B)}+\bar{C}}$

2.5 Resolución a la ejercitación propuesta del capítulo II

2.4.1 Aplicar los teoremas de De Morgan a las siguientes expresiones

a)$\overline{(A+B+C)D}$
$\overline{(A+B+C)}+\bar{D}$
$(\bar{A}\,\bar{B}\,\bar{C})+\bar{D}$

b) $\overline{ABC+DEF}$
$\overline{ABC}\ \ \overline{DEF}$
$(\bar{A}+\bar{B}+\bar{C})\ (\bar{D}+\bar{E}+\bar{F})$

c) $\overline{A\bar{B}+\bar{C}D+EF}$
$\left(\overline{A\bar{B}}\right).\left(\overline{\bar{C}D}\right).(\overline{EF})$
$\left(\bar{A}+\bar{\bar{B}}\right).\left(\bar{\bar{C}}+\bar{D}\right).(\bar{E}+\bar{F})$
$(\bar{A}+B).(C+\bar{D}).(\bar{E}+\bar{F})$

d) $\overline{\overline{(A+B)}+\bar{C}}$
$\left(\overline{\overline{A+B}}\right)\bar{\bar{C}}$
$\left(\bar{\bar{A}}\bar{\bar{B}}\right)C$
$A * B * C$

2.4.2 Completar la siguiente tabla

A	B	AND	NAND	OR	NOR	XOR
0	0	0	1	O	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0

2.4.3 Verificar mediante las tablas de verdad de las compuertas las Leyes De Morgan

$$AB = \bar{A} + \bar{B}$$

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

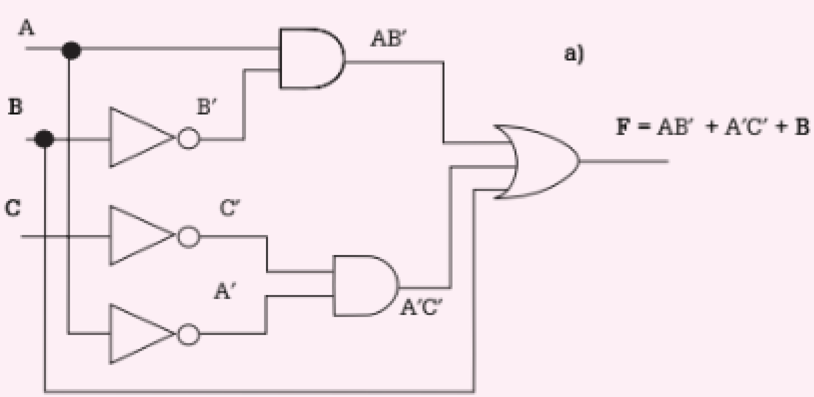
A	B	AB	$\overline{AB}$	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

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

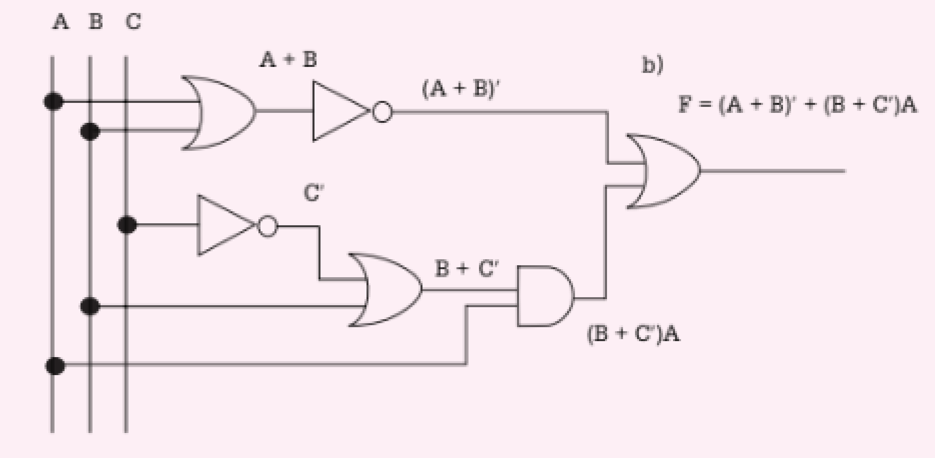
2.4.4. Representar las siguientes expresiones booleanas usando compuertas lógicas básicas:

a) F = AB′ + A′C′ + B
b) F = (A + B)′ + (B + C′)A

La representación de (a) es:



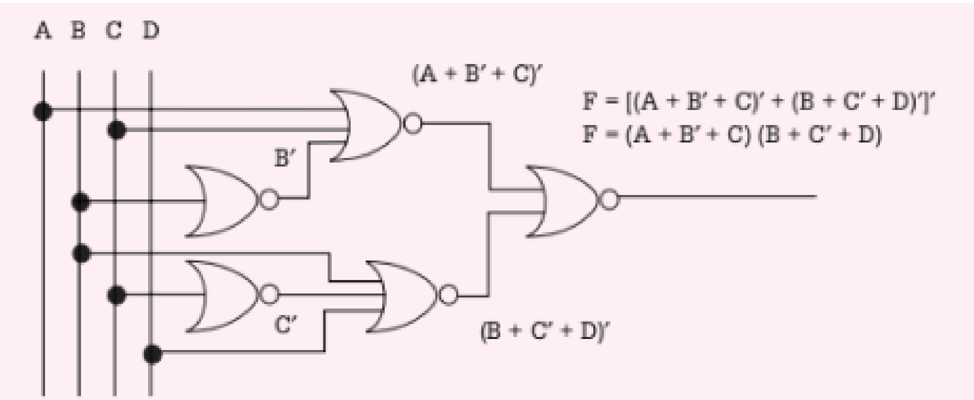
La representación de (b) es:



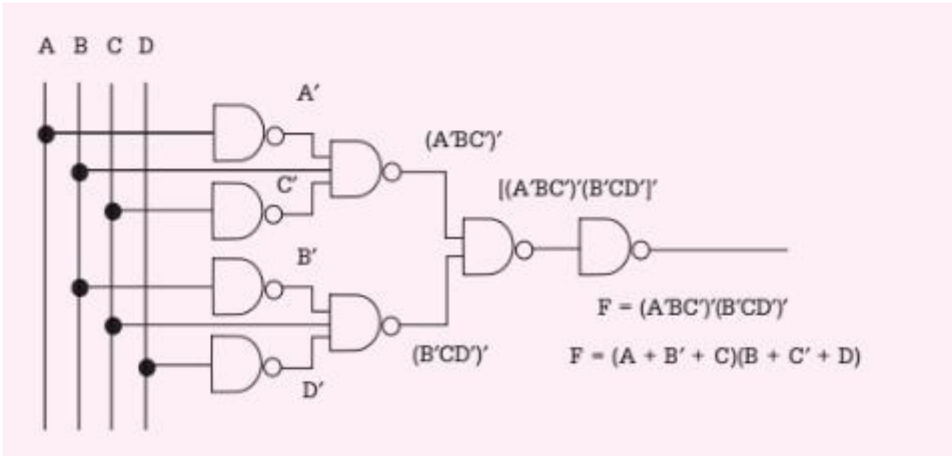
2.4.5 Representar la expresión booleana $F = (A + B' + C)(B + C' + D)$

usando sólo compuertas Nor.

En este caso se tiene el siguiente esquema



La misma expresión booleana representada con compuertas Nand quedaría de la siguiente manera:


A B C D
A′
C′
B′
D′
(A′BC′)′
(B′CD′)′
[(A′BC′)′(B′CD′)′]′
F = (A′BC′)′(B′CD′)′
F = (A + B′ + C)(B + C′ + D)