UTN
Facultad Regional Córdoba

TUP

Tecnicatura Universitaria en Programación

SISTEMAS DE PROCESAMIENTO DE DATOS

Unidad Temática N°4:
Álgebra de Boole

Material Teórico
1° Año – 1° Cuatrimestre

Índice

ALGEBRA DE BOOLE

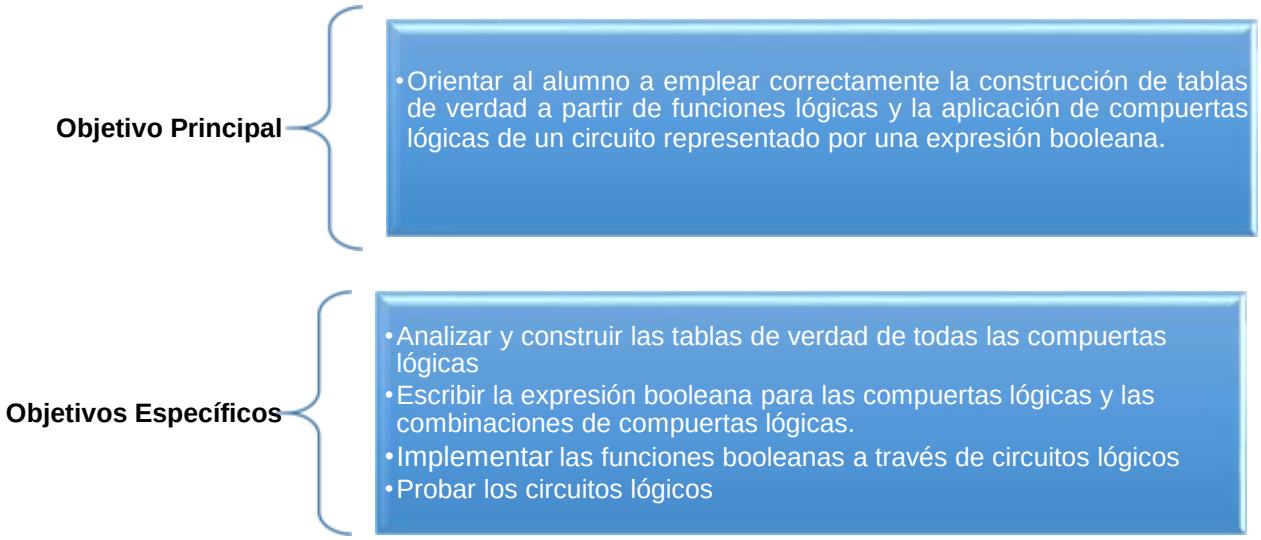
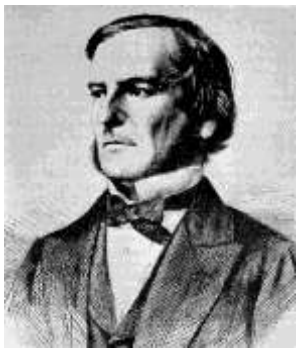


Gráfico 1: Elaboración propia.

INTRODUCCIÓN



El matemático inglés George Boole (1854) desarrolló una herramienta matemática que se utiliza para el estudio de computadores, denominada Algebra de Boole. Un siglo después, con el advenimiento de la electricidad y la electrónica, las estructuras algebraicas desarrolladas por Boole, fueron aplicadas por C.E. Shannon en la construcción de circuitos eléctricos, que constituyen la base del funcionamiento de cualquier dispositivo de Hardware como es el microprocesador, memorias, etc.

La aplicación en computadores es de tipo binario ⇒ 0/1.

El estado de un elemento del circuito lógico está representado por una variable que puede valer “1” o “0”.

El álgebra booleana es la teoría matemática que se aplica en la lógica combinatoria. Las variables booleanas son símbolos utilizados para representar magnitudes lógicas y pueden tener sólo dos valores posibles: 1 (valor alto) ó 0 (valor bajo).

Función Booleana:	Ejemplo
Expresión que indica la relación entre las variables y el número de variables: F= f(a,b,c,..) Son las representaciones analíticas de los circuitos lógicos, tiene asociado además un gráfico con compuertas lógicas y una tabla de valores de verdad.	$F(a,b,c,d)= \overline{abc} + b(\overline{c} + \overline{d})$
Funciones canónicas Son aquellas que tienen todas las variables de la función en cada uno de sus términos, ya sea en su forma directa o negada. Se presentan de dos formas: Forma Normal Disyuntiva o suma de productos y Forma Normal Conjuntiva o producto de sumas y pueden obtenerse directamente de la tabla de valores. Para obtener la suma de productos de la tabla, se debe tomar un Término suma por cada 1 de la tabla y si para la posición de ese 1 la variable está en 0 deberá aparecer como negada en el sumando en cambio está en 1 deberá aparecer sin negar.	**Ejemplo** $F(a,b,c) = abc + a\overline{b}c + \overline{a}bc$

<table>
<tr>
<td>

Tabla de Verdad

Tabla que recoge todas las combinaciones de las variables de entrada y los valores que toman las salidas.

Es una tabla que despliega el valor de verdad de una proposición compuesta, para cada combinación de valores de verdad que se pueda asignar a sus componentes. Considérese dos proposiciones A y B. Cada una puede tomar uno de dos valores de verdad: o V (verdadero), o F (falso). Por lo tanto, los valores de verdad de A y de B pueden combinarse de cuatro maneras distintas: o ambas son verdaderas; o A es verdadera y B falsa, o A es falsa y B verdadera, o ambas son falsas.

</td>
<td>

Ejemplo

$F(a,b,c) = abc + a\bar{b}c + \bar{a}bc$

a	b	c	F(a,b,c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

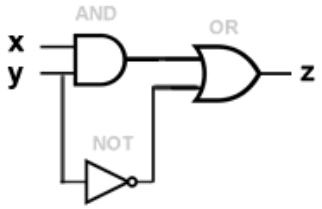
</td>
</tr>
<tr>
<td>

Circuito Lógico

Los circuitos lógicos se forman combinando compuertas lógicas. La salida de un circuito lógico se obtiene combinando las tablas correspondientes a sus compuertas componentes. Por ejemplo: es fácil notar que las tablas correspondientes a las compuertas OR, AND y NOT son respectivamente idénticas a las tablas de verdad de la disyunción, la conjunción y la negación en la lógica de enunciados, donde sólo se ha cambiado V y F por 0 y 1. Por lo tanto, los circuitos lógicos, de los cuales tales compuertas son elementos, forman un álgebra de Boole al igual que los enunciados de la lógica de enunciados.

</td>
<td>

Ejemplo



x	y
0	0
0	1
1	0
1	1

</td>
</tr>
</table>

Tabla 1: Elaboración propia.

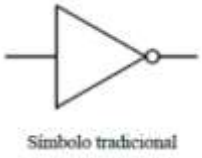
OPERACIONES DEL ÁLGEBRA DE BOOLE

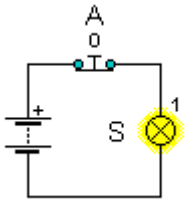
Las operaciones booleanas son posibles a través de los operadores binarios negación, suma y multiplicación, es decir que estos combinan dos o más variables para conformar funciones lógicas. Una compuerta es un circuito útil para realizar las operaciones anteriormente mencionadas.

Inversión o negación (complemento) COMPUERTA NOT

Esta operación se indica con una barra sobre la variable, es un operador algebraico que invierte el valor de una variable, es decir, si A denota la señal de entrada de un inversor, entonces Ā representa el complemento de tal señal. Su tabla de verdad, su símbolo lógico y eléctrico se representan a continuación:

A	Ā
0	1
1	0



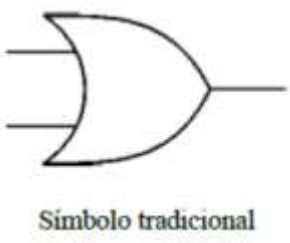


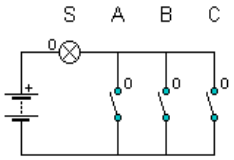
SUMA BOOLEANA

Compuerta OR

La suma booleana es 1 si alguna de las variables lógicas de la suma es 1 y es 0 cuando todas las variables son 0. Esta operación se asimila a la conexión paralela de contactos. En circuitos digitales, el equivalente de la suma booleana es la operación OR. Su tabla de verdad, su símbolo lógico y eléctrico se representan a continuación:

Decimal	A	B	A + B
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1



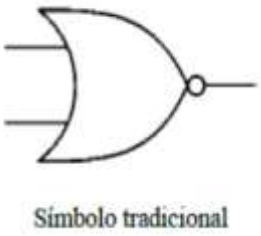


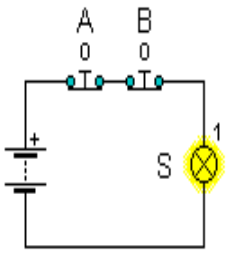
A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Compuerta NOR

El inverso de la función OR es la función NOR. Es una puerta OR con la salida invertida. Una compuerta NOR puede tener dos o más entradas; su salida es verdadera (1) si ninguna de sus entradas lo es, es decir si ambas entradas son falsas (0). Su tabla de verdad, su símbolo lógico y eléctrico se representan a continuación:

Decimal	A	B	$\overline{A + B}$
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0


Símbolo tradicional



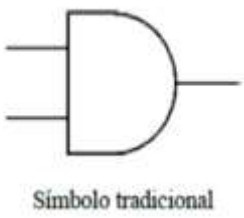
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

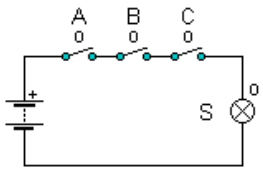
MULTIPLICACIÓN BOOLEANA

Compuerta AND

La multiplicación booleana es 1 si todas las variables lógicas son 1, pero si alguna es 0, el resultado es 0. Esta operación se asimila a la conexión en serie de contactos. En circuitos digitales, el equivalente de la multiplicación booleana es la operación AND. Su tabla de verdad, su símbolo lógico y eléctrico se representan a continuación:

Decimal	A	B	A.B
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1


Símbolo tradicional

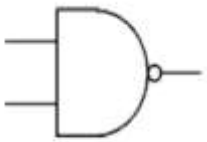


A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

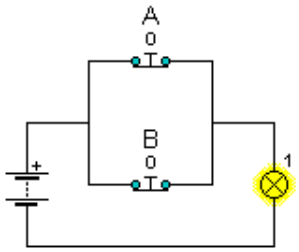
Compuerta NAND

El inverso de la función AND es la función NAND. Es una puerta AND con la salida invertida. Una puerta NAND puede tener dos o más entradas; su salida es verdadera si NO TODAS sus entradas son verdaderas, es decir si al menos una de ellas o ambas entradas son falsas (0). Su tabla de verdad, su símbolo lógico y eléctrico se representan a continuación:

Decimal	A	B	$\overline{A.B}$
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0



Símbolo tradicional

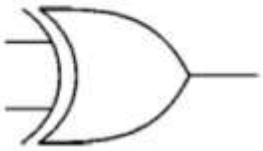


A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

COMPUERTA XOR

Es similar a una puerta OR pero excluyendo que ambas entradas sean verdaderas o 1. La salida es verdadera si las entradas A y B son DIFERENTES, es decir tienen distinto valor de verdad. La salida es falsa o 0 cuando ambas entradas son verdaderas o ambas son falsas. Las puertas XOR solo pueden tener dos entradas. Su tabla de verdad y su símbolo lógico se representa a continuación:

Decimal	A	B	$A \oplus B$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

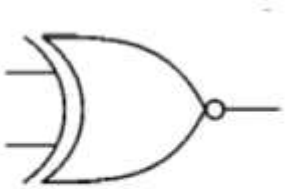


Símbolo tradicional

COMPUERTA N-XOR

El inverso de la función XOR es la función N-XOR. Es una puerta XOR con la salida invertida. Una puerta N-XOR puede tener sólo dos entradas; su salida es verdadera si sus entradas tienen el mismo valor de verdad, y es falsa en caso contrario. Su tabla de verdad y su símbolo lógico se representa a continuación:

Decimal	A	B	$\overline{A \oplus B}$
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1



Símbolo tradicional

ELABORACIÓN DE LA TABLA DE VERDAD

- A partir de una expresión o función booleana se construye una tabla de verdad que representa la salida del circuito lógico de dicha función, para todos los valores posibles de las variables de entrada.
- Esto requiere que se evalué la expresión booleana para todas las posibles combinaciones de valores posibles de las variables de entrada.
- En el caso de la expresión A(B+ CD) hay cuatro variables de entrada (A, B, C y D) y, por tanto, hay 24= 16 posibles combinaciones de valores o filas de la tabla de verdad donde se representan los resultados, con una sola salida representada por F.

ENTRADAS						SALIDA
A	B	C	D	CD	B + CD	F(A,B,C,D)= A(B+CD)
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	1	0
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	1	1	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	0	1	1
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	1	1

Tabla 2: Elaboración propia.

- Se construye la tabla de verdad utilizando las reglas de la adición y multiplicación booleanas, para resolver cada término en forma parcial. Comenzando por los términos que están encerrados entre paréntesis y respetando la jerarquía de las operaciones.
- Se obtienen los valores de salida ubicados en la última columna que corresponden a los valores de las variables de la función booleana, donde la expresión es igual a 1 o a 0 dependiendo del valor de las variables de entrada en la secuencia o fila correspondiente.

Análisis booleano del circuito lógico correspondiente a la función

- El Álgebra de Boole proporciona una manera concisa de expresar el funcionamiento de un circuito lógico formado por una combinación de puertas lógicas, de tal forma que la salida puede determinarse por la combinación de los valores de entrada.

- Para obtener la expresión booleana de un determinado circuito lógico, la manera de proceder consiste en:
    - Comenzar con las entradas situadas más a la izquierda.
    - Ir avanzando hasta las líneas de salida, escribiendo la expresión para cada puerta.

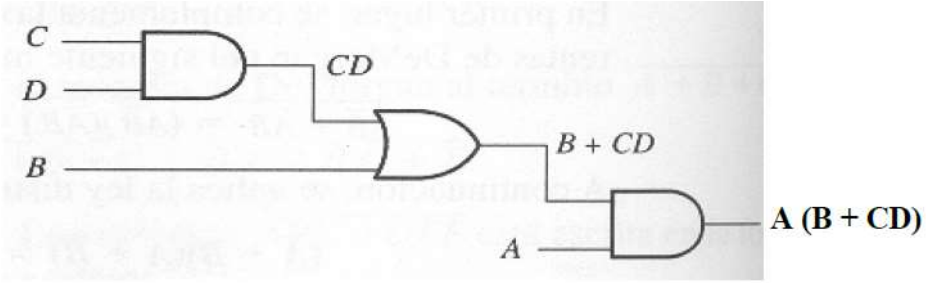


Gráfico 1: Elaboración propia.

- La expresión de la puerta AND situada más a la izquierda cuyas entradas son C y D es el producto lógico CD.
- La salida de la puerta AND situada más a la izquierda es una de las entradas de la puerta OR y B es su otra entrada. Por tanto, la expresión de salida para la puerta OR es B + CD.
- La salida de la puerta OR es una de las entradas de la puerta AND situada más a la derecha, siendo A su otra entrada. Por lo tanto la expresión de esta puerta AND será A (B + CD).

REPRESENTACIÓN EN LA FORMA CANÓNICA

Suma de miniterminos

Introducción

En Álgebra booleana, se conoce como término canónico de una función lógica a todo producto o suma en la cual aparecen todas las variables en su forma directa o inversa. Una Función lógica que está compuesta por operador lógico puede ser expresada en forma canónica usando los conceptos de *minterm.* Todas las funciones lógicas son expresables en forma canónica como una "suma de **minterms**". Esto permite un mejor análisis para la simplificación de dichas funciones, lo que es de gran importancia para la minimización de circuitos digitales.

Una función booleana expresada como una disyunción lógica (OR) de minterminos es usualmente conocida la "**suma de productos**".

Definición

Para una función booleana de n variables $x_1, x_2 \ldots\ldots x_n$, un producto booleano en el que cada una de las n variables aparece una sola vez (negada o sin negar) es llamado **minitérmino**. Es decir, un minitérmino es una expresión lógica de n variables consistente únicamente en el operador conjunción lógica (AND) y el operador complemento o negación (NOT).

Por ejemplo $a.b.c; a.\overline{b}.c; a.\overline{b}.\overline{c}$, son ejemplos de minterminos (minterms) para una función booleana con las tres variables a;b;c.

Nota: en algunas bibliografías, se puede encontrar la representación de la negación de la siguiente forma:

a./b.c=a.b'.c= $a.\overline{b}.c$; a./b./c =a.b'.c'= $a.\overline{b}.\overline{c}$

Metodología

	A	B	C	F_f	F_g
m_0	0	0	0	0	1
m_1	0	0	1	1	0
m_2	0	1	0	0	1
m_3	0	1	1	0	1
m_4	1	0	0	0	0
m_5	1	0	1	1	0
m_6	1	1	0	0	1
m_7	1	1	1	1	0

$$F_f = \bar{A}.\bar{B}.C + A.\bar{B}.C + A.B.C$$

$$F_G = ??$$

(Realizar la propuesta)

Tabla 3: Elaboración propia

Productos de maxiterminos

Introducción

Una Función lógica que está compuesta por operador lógico puede ser expresada en forma canónica usando los conceptos de *maxiter.* Todas las funciones lógicas son expresables en forma canónica como una "productos de **maxiters**". Esto permite un mejor análisis para la simplificación de dichas funciones, lo que es de gran importancia para la minimización de circuitos digitales.

Una función booleana expresada como una conjunción lógica (AND) de maxiterminos es usualmente conocida la "**productos de sumas**".

Definición

Para una función booleana de n variables $x_1, x_2 \ldots\ldots x_n$, una suma booleana en el que cada una de las n variables aparece una sola vez (negada o sin negar) es llamado **maxitermino**. Es decir, un maxitermino es una expresión lógica de n variables consistente únicamente en el operador conjunción lógica (OR) y el operador complemento o negación (NOT).

Por ejemplo $a + b + c; a + \bar{b} + c; a + \bar{b} + \bar{c}$, son ejemplos de maxiterminos para una función booleana con las tres variables a;b;c.

Nota: en algunas bibliografías, se puede encontrar la representación de la negación de la siguiente forma:

a+/b +c = a +b'+c= $a + \bar{b} + c$;

Metodología

- Para obtener el producto de sumas de la tabla, se debe tomar un termino producto por cada 0 de la tabla.
- Si para la posición de ese “0” la variable está en “1” deberá aparecer como negada.
- Si la variable está en “0” deberá aparecer sin negar.

	A	B	C	F_f	F_g
M_0	0	0	0	0	1
M_1	0	0	1	1	0
M_2	0	1	0	1	1
M_3	0	1	1	0	1
M_4	1	0	0	0	0
M_5	1	0	1	1	0
M_6	1	1	0	1	1
M_7	1	1	1	1	0

$$F_f = (A + B + C).(A + \bar{B} + \bar{C}).(\bar{A} + B + C)$$

$$F_G = ??$$

(Realizar la propuesta)

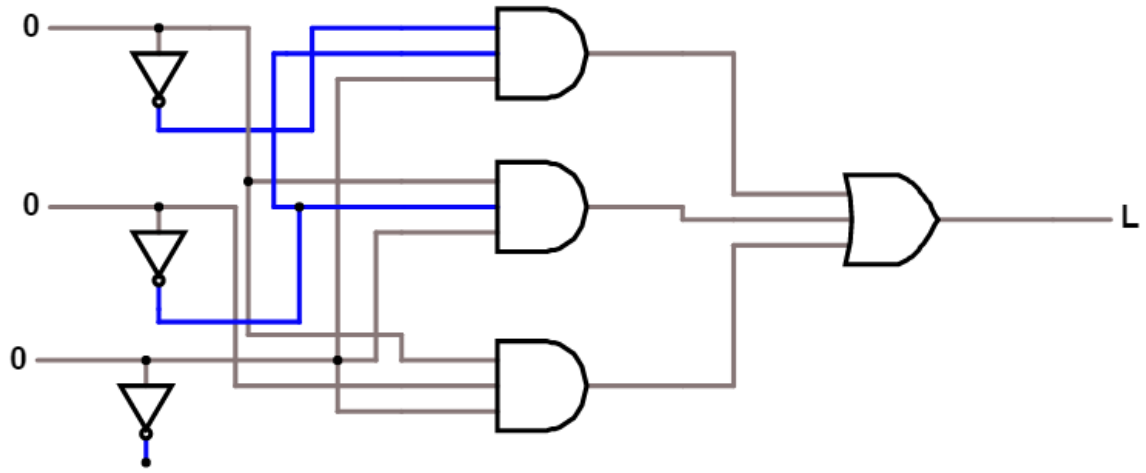
SIMULADORES DE CIRCUITOS DIGITALES

Software simulador de circuitos lógicos on line que se recomienda utilizar, se lo puede encontrar rn este link.

http://falstad.com/circuit/circuitjs.html

Circuitos con compuertas lógicas

En el circuito esquemático, se representa la función lógica F_f de la tabla (tabla 3) verdad anterior.



ALGEBRA DE BOOLE - POSTULADOS, PROPIEDADES Y TEOREMAS

Permiten simplificar funciones lógicas y expandir una función cualquiera para obtener la representación de la función canónica.

Como veremos mas adelante, la expresión original que se deduce de un problema propuesto, puede sufrir una sensible reducción mediante la aplicación de los postulados, propiedades y teoremas más significativos del álgebra de Boole . Este método de reducción de las expresiones recibe el nombre de simplificación algebraica.

En esta unidad, no se realizará la simplificación algebraica, si no que se va a utilizar un método gráfico denominado **"Mapa de Karnaugh"**.

Postulados.

Vamos a exponer los postulados más significativos del álgebra de Boole.

Postulado 1.

La suma lógica de una variable más un "1" lógico equivale a un "1" lógico.

$$a + 1 = 1$$

Postulado 2.

La suma lógica de una variable más un "0" lógico es igual al valor de la variable.

$$a + 0 = a$$

Postulado 3.

El producto lógico de una variable por un "1" lógico es igual al valor de la variable.

$$a.1 = a$$

Postulado 4.

El producto lógico de una variable por un 0 lógico es igual a un 0 lógico.

$$a.0 = 0$$

Postulado 5.

La suma lógica de dos variables iguales equivale al valor de dicha variable.

$$a + a = a$$

Postulado 6.

El producto lógico de dos variables iguales equivale al valor de dicha variable.

$$a.a = a$$

Postulado 7.

La suma lógica de una variable más la misma variable negada equivale a un "1" lógico.

$$a + \bar{a} = 1$$

Postulado 8.

El producto lógico de una variable por la misma variable negada equivale a un "0" lógico.

$$a.\bar{a} = 0$$

Postulado 9.

Si una variable es negada dos veces, esta no varia . Este postulado es valido para cualquier número par de negaciones.

$$\bar{\bar{a}} = a$$

Postulado 10 .

Si se invierten los dos miembros de una igualdad, ésta no sufre ninguna variación.

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

$$f = A.B \;\; ; \;\; \bar{f} = \overline{A.B}$$

Propiedades.

De la misma forma que en la matemática ordinaria, en el álgebra de Boole se cumplen las propiedades que exponemos a continuación .

Propiedad 1.

Propiedad conmutativa.

$$a) \ldots a + b = b + a$$

$$b) \ldots a.b = b.a$$

Propiedad 2

Propiedad asociativa

$$a) \dots a + b + c = a + (b + a) = (a + b) + c$$

$$b) \dots a.b.c = a.(b.a) = (a.b).c$$

Propiedad 3

Propiedad distributiva.

$$a) \dots a.(b + c) = (a.b) + (a.c)$$

$$b) \dots a + b.c = (a + b).(a + c)$$

Teoremas.

Los teoremas que enunciaremos a continuación podrán ser demostrados apoyándonos en los postulados y propiedades descritos anteriormente

Teorema 1 : " Ley de Absorción ".

$$a) \dots a + (a.b) = a$$

Demostración:

$$a + (a.b) = a + a.b = a(1 + b) = a.1 = a$$

$$b) \dots a.(a + b) = a$$

Demostración:

$$a.(a + b) = a.a + a.b = a + (a.b) = a$$

Teorema 2 :

$$a) \dots a + (\bar{a}.b) = a + b$$

Demostración:

$$a + (\bar{a}.b) = (a + \bar{a}).(a + b) = 1.(a + b) = a + b$$

$$b) \dots b.\left(a + \bar{b}\right) = a.b$$

Demostración:

$$b.\left(a + \bar{b}\right) = (b.a)\left(b.\bar{b}\right) = b.a + 0 = a.b$$

Teorema 3: “Leyes de DeMorgan”.

Merece una especial mención las leyes de Morgan por su gran utilidad en los procesos de simplificación. La comprobación de estos teoremas se podrá realizar construyendo la tabla de verdad.

$$a) \ldots \overline{A + B} = \bar{A}.\bar{B}$$

$$b) \ldots \overline{A.B} = \bar{A} + \bar{B}$$

Lo mismo que otros teoremas y propiedades, las leyes de Morgan son generalizables, es decir, son aplicables a expresiones de más de dos variables.

Teorema 4 :

$$a) \ldots A.B + \bar{A}.C + B.C = A.B + \bar{A}.C$$

$$B) \ldots (A + B).(\bar{A} + C).(B + C) = (A + B).(\bar{A} + C)$$

La forma más cómoda de demostrar este teorema, como en el caso anterior, es mediante la construcción de la tabla de verdad.

EXPANSIÓN DE FUNCIÓNES A LA FORMA CANÓNICA.

En Álgebra booleana, se conoce como término canónico de una función lógica a todo producto o suma en la cual aparecen todas las variables en su forma directa o inversa. Una Función lógica que está compuesta por operador lógico puede ser expresada en forma canónica usando los conceptos de **minterm** y **maxterm**. Todas las funciones lógicas son expresables en forma canónica, tanto como una **"suma de minterms"** como **"producto de maxterms**". Esto permite un mejor análisis para la simplificación de dichas funciones, lo que es de gran importancia para la minimización de circuitos digitales.

Función incompleta

Toda función que no está representada en su forma canónica, se la puede denominar como función incompleta. Aplicando los teoremas, postulados y propiedades anteriormente estudiadas, se puede obtener la función canónica, partiendo de la función incompleta. A esta metodología se la denomina expandir una función lógica.

Esto también se puede lograr, utilizando método de la tabla verdad.

Expansión de una función por método algebraico

Ejemplos 1

En este caso, se va a expandir una función incompleta para obtener la su representación canónica.

$$f(1) = a + b.c$$

En la función se puede ver 3 variables que son a;b; y c. Para obtener la función canónica, debe tener en todos los términos todas las variables en forma directa o negada. Para lograrlo, se utilizan las leyes, postulados y propiedades del algebra de Boole.

Aplicamos **postulado 3**, al multiplicar por 1 a cada termino, no modifico la función.

$$f(1) = a.1 + b.c.1$$

Aplicamos **postulado 7**, cambiar el 1 con las variables que faltan al término.

$$f(1) = a.(b + \bar{b}) + b.c.(a + \bar{a})$$

Aplicamos p**ropiedad 3 *propiedad distributiva***.

$$f(1) = a.b + a.\bar{b} + b.c.a + b.c.\bar{a}$$

Aplicamos **propiedad 1 *propiedad conmutativa***.

$$f(1) = a.b + a.\bar{b} + a.b.c + \bar{a}.b.c$$

Aplicamos nuevamente el **postulado 3**, al multiplicar por 1 a cada termino, no modifico la función.

$$f(1) = a.b.1 + a.\bar{b}.1 + a.b.c + \bar{a}.b.c$$

Aplicamos **postulado 7**, cambiar el 1 con las variables que faltan al término.

$$f(1) = a.b.(c + \bar{c}) + a.\bar{b}.(c + \bar{c}) + a.b.c + \bar{a}.b.c$$

Aplicamos **propiedad 3 *propiedad distributiva***.

$$f(1) = a.b.c + a.b.\bar{c} + a.\bar{b}.c + a.\bar{b}.\bar{c}) + a.b.c + \bar{a}.b.c$$

El siguiente paso, es verificar que los términos se repitan. Si hay algún termino repetido, se aplica el **postulado 5.** En este caso simplemente la tachamos por estar de más.

$$f(1) = a.b.c + a.b.\bar{c} + a.\bar{b}.c + a.\bar{b}.\bar{c} + \sout{a.b.c} + \bar{a}.b.c$$

Finalmente obtenemos de esta manera la representación en **su forma canónica** como suma de términos o ***miterminos.***

$$f(1) = a.b.c + a.b.\bar{c} + a.\bar{b}.c + a.\bar{b}.\bar{c} + \bar{a}.b.c$$

$$f(1) = \sum_{i=0}^{7} m3; m4; m5; m6; m7$$

Finalmente se puede aplicar **propiedad 1** propiedad conmutativa. Se puede demostrar utilizando la tabla verdad que son iguales

$$f(1) = a + b.c = \bar{a}.b.c + a.\bar{b}.\bar{c} + a.\bar{b}.c + a.b.\bar{c} + a.b.c$$

SIMPLIFICASION POR KARNAUGH

La complejidad de las compuertas lógicas digitales con que se llevan a cabo las funciones de Boole se relacionan directamente con la complejidad de la expresión algebraica de la cual se desprende la función. Aunque la representación de la tabla de verdad de una función única, puede aparezca de muchas formas diferentes. Las funciones de Boole pueden ser simplificadas por medios algebraicos de la manera que se vio antes.

El método del mapa presenta un procedimiento simple y directo para minimizar las funciones de Boole. Este método puede ser tratado no solamente en la forma pictórica de una tabla de verdad, sino como una extensión del diagrama de Venn. El método del mapa, propuesto primero por Veitch y modificado ligeramente por Karnaugh se conoce como el "diagrama de Veitch o el mapa de Karnaugh.

El mapa es un diagrama, hecho de cuadros. Cada cuadro representa un término mínimo. Como cualquier función de Boole puede ser expresada como una suma de términos mínimos, se desprende que dicha función, se reconoce gráficamente en el mapa a partir del área encerrada por aquellos cuadros cuyos términos mínimos se incluyen en la función.

Al reconocer varios patrones, se puede derivar expresiones algebraicas alternas para la misma función de las cuales se puede escoger la más simple.

Tipos de Mapa de Karnaugh.

Mapa de dos variables

En un mapa de dos variables hay cuatro términos mínimos para dos variables, es decir que el mapa consiste en cuatro cuadrados, uno para cada término mínimo otra forma de realizarlo como la imagen 1 donde se puede observar el mapa que se dibuja, referido a la tabla de verdad 1, ya que sirve para demostrar la relación entre los cuadrados y las dos variables. Los ceros y unos marcados para cada fila y columna designan los valores A y B respectivamente.

Si se marcan los cuadrados cuyos términos mínimos pertenecen a una función dada, el mapa de dos variables se convierte en otro método útil para representar una cualquiera de las 16 funciones de Boole de dos variables. Los cuadrados se escogen de los términos mínimos de la función.

A	B	s1
0	0	1
0	1	0
1	0	1
1	1	1

Tabla 3: Tabla de verdad de 2 bits. Elaboración propia.

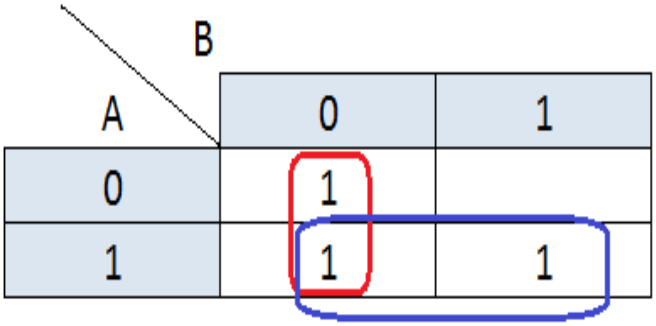


Tabla 4: Mapa de la tabla 4. Elaboración propia.

S1= B' +A

Luego de ubicar los respectivos valores 1 en el mapa los juntamos de a números pares excepto que sea imposible unir un término con otros, una vez hecho eso se procede a armar la función S1, para ello se observa en el aglomerado de los uno cual de mis variables A o B sigue constantes y cual cambia, la que cambia se obvia en la generación de la función S1, solo agregamos la que se mantiene constante, en el caso de la imagen 1 el rectángulo rojo nos muestra un 1 ubicado en A=0 ; B=0 y otro en A=1 ; B=0, se observa que A pasa de 0 a 1 y B se mantiene constante en 0 por lo tanto el primer factor que aparece es B que al ser cero será B negado.

Para el segundo término tenemos un uno en A=1 ; B=0 y en A=1 ; B=1, aquí se observa que A se mantiene constante y B pasa de B=0 a B=1 por lo tanto como el valor de B cambia se obvia de la función y solo se deja A que al ser valor 1 no se negara quedando así el segundo y último término, dando la función:

$$S1= B' +A$$

Mapa de tres variables

En la imagen 2 se ilustra un mapa de tres variables. Hay ocho términos mínimos para las tres variables como se observa en la tabla de verdad número 2. El mapa por tanto consiste en ocho cuadrados. Nótese que los términos mínimos se arreglan no en una secuencia binaria, donde la característica de esta secuencia es que solamente un bit cambia de 1 a 0 o de 0 a 1, en la secuencia del listado.

El mapa dibujado en la imagen 2 se marca con los números de cada fila o cada columna para mostrar la relación entre los cuadrados de las tres variables

A	B	C	s1
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

.

Tabla 5: Tabla de verdad de 3 bits. Elaboración propia.

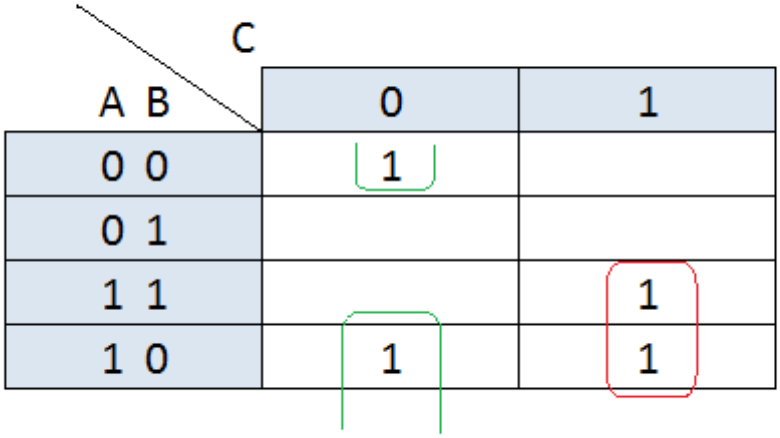


Tabla 6: Mapa de la Tabla 5. Elaboración propia.

Siguiendo el mismo principio que usamos antes podemos obtener la siguiente función:

$$S1 = B' C' + A C$$

Mapa de cuatro variables

En la imagen 3 se ilustra un mapa de cuatro variables. Hay dieciséis términos mínimos para las cuatro variables de la tabla de verdad número 3. El mapa por tanto consiste en dieciséis cuadrados.

A	B	C	D	s1
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Tabla 7: Tabla de verdad de 3 bits

A B \ C D	00	01	11	10
0 0		1		1
0 1		1	1	
1 1		1	1	
1 0		1		

Tabla 8: Mapa de la tabla 3 para cuatro variables. Elaboración propia

$$S= A' B' C D' + C' D + B D$$

Una vez que ordenamos los valores 1 en el mapa podemos armar la función S en base al mapa.

En el primer término tenemos un 1 que no pudo ser agrupado, por ende depende de todas las variables quedando A=0, B=0, C=1, D=0 quedando el término como A'. B'. C' .D'.

Para el segundo término, marcado en el mapa con rojo, podemos ver que tenemos que C y D se mantienen fijos pero A y B cambian, por ende esta parte de S depende solo de C=0 y D=1.

Para el último término, si analizamos A, B,C y D podemos observar que B se mantiene fija, al igual que D, pero A pasa de A=0 a A= 1 por lo tanto no tomamos el valor al igual que C, por lo tanto S dependerá de B y D quedando a si :

$$S= A' B' C D' + C' D + B D$$

BIBLIOGRAFÍA

Libros:

Johnsonbaugh, Richard. MATEMÁTICAS DISCRETAS. 1999 Edición 6. Editorial PEARSON EDUCACIÓN. México. Capítulo 7

Epp Susanna S. Matemáticas Discretas con aplicaciones. Cuarta Edición. 2012Edit. CENGAGE Learning. México. Capítulo 6.

Lipschutz Seymour. MATEMÁTICAS PARA COMPUTACIÓN. 1992. Edit. McGRAW-HILL. México. Capítulo 7y 8.

Morris Mano - Arquitectura de computadores, Capítulo 1.Sistemas binarios. Editorial Prentice Hall - 3ª edición. Biblioteca Central Ubicación: 004.22 MAN a

Páginas

http://falstad.com/circuit/circuitjs.html
http://163.178.104.150/ci1210/leccion%206.%20%C3%81lgebra%20de%20Boole/DEMOSTRACION%20LEYES%20DE%20ABSORCION.htm