UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

ESTRUCTURAS DISCRETAS

GRUPO 06

Ing. Orlando Zaldívar Zamorategui

Programa de cómputo para integrar los métodos de minimización de funciones booleanas. Método algebraico. Método de QUINE-MCCLUSKEY. Método mapas de Karnaugh.

Equipo 6

Ambrosio Escobar, Dominic Yael

Fernández Bautista, Ciria Andrea

Hernández Andrade, Miguel Ángel

Hernández Nájera, Itzel Adriana

Martínez García, José Eduardo

Martínez Martínez, Iván

Mendoza Camacho, Estrella de Maria

Rivera Rodríguez, Hugo Abraham

Semestre

2025-2

Fecha de asignación: **06/05/2025**

Fecha de entrega: **16/05/2025**

Fecha entregada:

16/05/2025

Índice

1.Objetivo

El objetivo de este trabajo es proporcionar una comprensión clara y accesible de la teoría de los sistemas algebraicos en el ámbito de la computación, con un enfoque particular en el álgebra booleana y el diseño de circuitos digitales. A través de un recorrido por los conceptos fundamentales y las técnicas de minimización de funciones booleanas, se buscará ilustrar cómo estas herramientas pueden ser aplicadas en la optimización de circuitos digitales.

Al final, el lector debe sentirse capacitado para implementar un programa de cómputo que utilice y compare tres métodos de minimización: el método algebraico, el método de Quine-McCluskey y el método de mapas de Karnaugh, reconociendo su importancia en la mejora de procesos dentro del diseño digital y la resolución de problemas complejos.

2.Introducción

En el campo de la computación y la ingeniería electrónica, el diseño eficiente de circuitos digitales es fundamental para el desarrollo de dispositivos y sistemas que operan de manera óptima. Las funciones booleanas son la base de la lógica digital, y su minimización es crucial para reducir la complejidad de los circuitos, disminuir el consumo de energía y mejorar el rendimiento general.

La minimización de funciones booleanas se puede abordar mediante diversos métodos, cada uno con sus propias características y aplicaciones. Entre estos métodos, se destacan el método algebraico, que utiliza reglas y teoremas del álgebra booleana para simplificar expresiones; el método de Quine-McCluskey, un enfoque sistemático y tabular que permite la minimización de funciones booleanas de múltiples variables; y el método de mapas de Karnaugh, que proporciona una representación visual que facilita la identificación de términos comunes y la simplificación de expresiones.

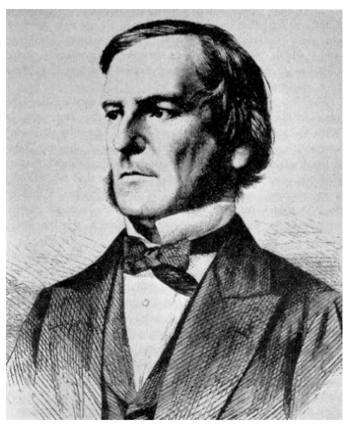
3.Desarrollo

3.1Álgebra de Boole

En 1938, Claude Shannon demostró cómo se podían utilizar las reglas de la lógica para diseñar circuitos. Estas reglas, enunciadas por primera vez en 1854 por George Boole en su libro Las leyes del pensamiento, son el fundamento del Álgebra de Boole. El primer paso en la construcción de un circuito consiste en representar su función booleana asociada mediante una expresión construida por medio de las operaciones básicas de un álgebra de Boole. De los procedimientos más relevantes se pueden destacar los diagramas de Karnaugh y el método Quine-McCluskey, para el diseño de circuitos eficientes.

3.1.1George Boole

Matemático británico nacido en Lincoln, Reino Unido en 1815, creador de un nuevo sistema de cálculo lógico que póstumamente sería llamado Álgebra de Boole. Dicho sistema, en el que las proposiciones se reducen a símbolos sobre los que puede operar matemáticamente, supuso un avance fundamental en el desarrollo de la lógica y, más de un siglo después, hallaría un formidable e insospechado campo de aplicación en la informática y los microprocesadores, cuyo funcionamiento se basa en la lógica binaria de Boole. Falleció en la actual Irlanda en 1864.



Geoge Boole.
Extraído de: López Medina, L. (Noviembre 2003). Método didáctico de simplificación de funciones booleanas.

3.1.2Funciones Booleanas

El Álgebra de Boole proporciona las operaciones y las leyes para trabajar en el conjunto (0, 1). Los interruptores electrónicos y ópticos se estudian utilizando este conjunto y las reglas del Álgebra de Boole. Las tres operaciones fundamentales del Álgebra de Boole que se usan habitualmente son el complemento, la suma y el producto, booleanos.

Complemento: 0', 1', NOT

Suma booleana: +, OR

Producto booleano: ·, AND

A continuación, se presentan sus operaciones y propiedades correspondientes:

Complemento.	Suma booleana.	Producto booleano.
0' = 1	1 + 1 = 1	1 · 1 = 1
1' = 0	1 + 0 = 1	1 · 0 = 0
	0 + 1 = 1	0 · 1 = 0
	0 + 0 = 0	0 · 0 = 0

3.1.3Jerarquía de operaciones

En caso de que las expresiones contengan paréntesis o más de un operador, se recurre a las reglas de precedencia de los operadores:

1. Complementos
2. Productos booleanos
3. Sumas booleanas

Los resultados acerca de un álgebra de Boole se pueden traducir directamente a resultados acerca de fórmulas proposicionales. Sus correspondencias serían de la siguiente forma:

1. Complemento: ¬
2. Suma booleana: v
3. Producto booleano: ∧

Sea B = [0, 1]. Entonces Bn = [(x1, x2, …, xn) | xi B, 1 i n] es el conjunto de todas las posibles n-tuplas de ceros y unos. La variable x se llama variable booleana si toma valores en el conjunto B, esto es, sus únicos valores posibles son 0 y 1. Una función de Bn en B se llama función booleana de grado n. La función F(x, y) = xy que va del conjunto de los pares ordenados de valores booleanos al conjunto {0, 1} es una función de grado 2 con:

F(0, 0) = 0

F(0, 1) = 0

F(1, 0) = 1

F(1, 1) = 0

Como se muestra en la tabla de verdad 1.

x	y	F(x, y)
1	1	0
1	0	1
0	1	0
0	0	0

Tabla de verdad 1.
Extraído de: Matemática discreta y sus aplicaciones. Rosen K. (2004).

Las funciones booleanas se pueden representar utilizando expresiones construidas con variables y operadores booleanos. Las expresiones booleanas en las variables x1, x2, …, xn se definen recursivamente como:

0,1,x1, x2, …, xn son expresiones booleanas; si E1 y E2 son expresiones booleanas, entonces E1, (E1E2) y (E1+E2) son expresiones booleanas.

Cada expresión booleana representa una función booleana. Los valores de esta función se obtienen sustituyendo las variables de la expresión por 0 y 1.

Ejemplo: calcular los valores de la función booleana F(x, y, z) = xy + z.

Solución: se resuelve con tabla de verdad para tres variables booleanas.

x	y	z	xy	$\bar{z}$	$F(x,y,z)=xy+\bar{z}$
1	1	1	1	0	1
1	1	0	1	1	1
1	0	1	0	0	0
1	0	0	0	1	1
0	1	1	0	0	0
0	1	0	0	1	1
0	0	1	0	0	0
0	0	0	0	1	1

Tabla de verdad 2.
Extraído de: Matemática discreta y sus aplicaciones. Rosen K. (2004).

3.1.4Propiedades de un álgebra de Boole

Las propiedades en un álgebra de boole son muchas para enlistar. Sin embargo, las más importantes se muestran en la siguiente tabla. Estas propiedades son particularmente útiles en la simplificación del diseño de circuitos. Se puede demostrar cada una con una tabla de valores.

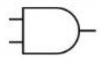
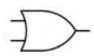
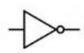
Expresión	Propiedad
$\overline{\overline{x}} = x$	Doble complemento
$x + x = x$ $x \cdot x = x$	Idempotencia
$x + 0 = x$ $x \cdot 1 = x$	Elemento neutro
$x + 1 = 1$ $x \cdot 0 = 0$	Acotación
$x + y = y + x$ $xy = yx$	Conmutativa
$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Asociativa
$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$	Distributiva
$\overline{(xy)} = \overline{x} + \overline{y}$ $\overline{(x + y)} = \overline{x}\,\overline{y}$	De Morgan
$x + xy = x$ $x(x + y) = x$	Absorción
$x + \overline{x} = 1$	Inverso para el uno
$x\overline{x} = 0$	Inverso para el cero

Propiedades del Álgebra de Boole.
Extraído de: Matemática discreta y sus aplicaciones. Rosen K. (2004).

3.2Método de Karnaugh

3.2.1Circuitos

El álgebra de Boole se utiliza para modelos de circuitos de dispositivos electrónicos. Cada circuito se puede diseñar utilizando las propiedades del álgebra de Boole vistas anteriormente. Los elementos básicos de los circuitos se llaman puertas lógicas, y cada una implementa una operación booleana. Al depender únicamente de los elementos de entrada (y no del estado actual del dispositivo), los diagramas con estas puertas lógicas son llamados redes lógicas o circuitos combinacionales. Es importante mencionar que estos diagramas se leen de izquierda a derecha y de arriba hacia abajo. Los flujos de la parte izquierda de cada puerta son variables booleanas entrantes, las cuales pueden variar en cantidad. El flujo de la parte derecha de cada puerta lógica es la salida, y solo puede haber una única. Esta se puede integrar a otra puerta posteriormente.

	AND
	OR
	NOT

Puertas lógicas.
Extraído de: Matemáticas Discretas, aplicaciones y ejercicios. (Villalpando Becerra, J. F., Garcia Sandoval, A., 2014).

3.2.2Maurice Karnaugh

Maurice Karnaugh, nacido en la ciudad de Nueva York, se licenció en el City College de Nueva York y se doctoró en la Universidad de Yale. Fue miembro del personal técnico de los Bell Laboratories de 1952 a 1966 y director de investigación y desarrollo en la división de sistemas federales de AT&T de 1966 a 1970. En 1970 se incorporó a IBM como miembro del personal de investigación. Karnaugh ha hecho contribuciones fundamentales a la aplicación de técnicas digitales tanto en la computación como en las telecomunicaciones. Sus áreas de interés más recientes incluyen los sistemas expertos y los métodos heurísticos de búsqueda.

Maurice Karnaugh.
Extraído de: Dispositivos y circuitos electrónicos. (Donald A. Neamen., 2012).

3.2.3Diagramas de Karnaugh

El diagrama de Karnaugh es un método gráfico cuyo objetivo es hallar términos que se pueden combinar en el caso de funciones booleanas que dependen de relativamente pocas variables. Este método fue introducido por Maurice Karnaugh en 1953. El diagrama de Karnaugh, o K-diagrama, no proporciona un método visual para simplificar una forma normal disyuntiva, pero no son adecuados para automatizar este proceso.

Es importante resaltar tres conceptos que serán aplicados posteriormente en la minimización con K-diagramas en el álgebra de Boole:

- Minitérmino: es un término producto donde aparecen todas las variables de la función, con o sin su complemento, una única vez.
- Maxitérmino: es un término de suma donde aparecen todas las variables de la función, con o sin su complemento, una única vez.
- Término canónico: es toda suma o producto que contenga a todas las variables de una función. Puede combinar minitérminos en una forma normal disyuntiva (sumas) o maxitérminos en una forma normal conmutativa (productos).

Las funciones se acomodan en forma de matriz, representando la función en ella como coordenadas.
Ejemplo: F(x, y) = xy' + x'y

	y	$\bar{y}$
x		1
$\bar{x}$	1	

Diagrama 1

Los minitérminos se traducen en la tabla colocando números uno donde se encuentren. La tabla se lee de la primera columna (la que tiene la variable x) y después la primera fila (la que tiene la variable y).

Ejemplo 2: F(x, y) = xy' + x'y + x'y'

	y	$\bar{y}$
x		1
$\bar{x}$	1	1

Diagrama 2

Siempre que en el K-diagrama haya números uno adyacentes, los minitérminos representados por estas celdas se pueden combinar entre sí dando lugar a un término que depende sólo de una de las variables. En el ejemplo anterior xy' y x'y' están representadas por celdas adyacentes y pueden combinarse para producir y', ya que xy' + x'y' = (x + x')y' = y'. Además, si hay unos en las cuatro celdas, pueden combinarse los cuatro minitérminos para dar lugar a uno solo, que será la expresión booleana 1 que no depende de ninguna de las variables. Es decir:

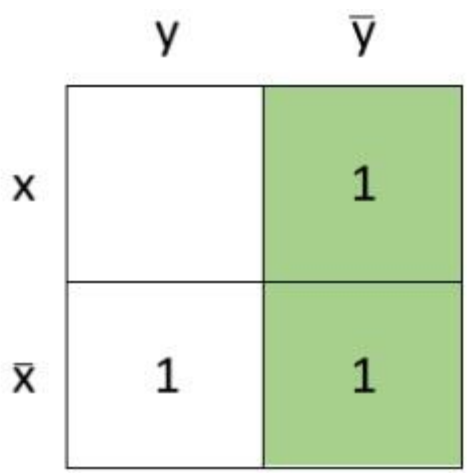


Diagrama 3

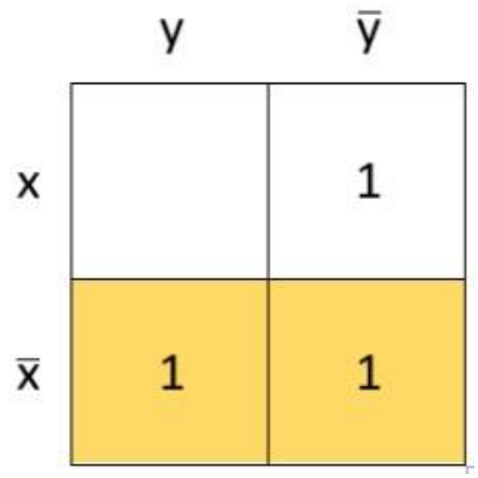


Diagrama 4

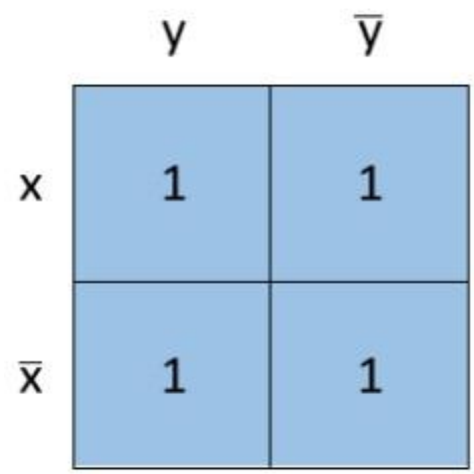
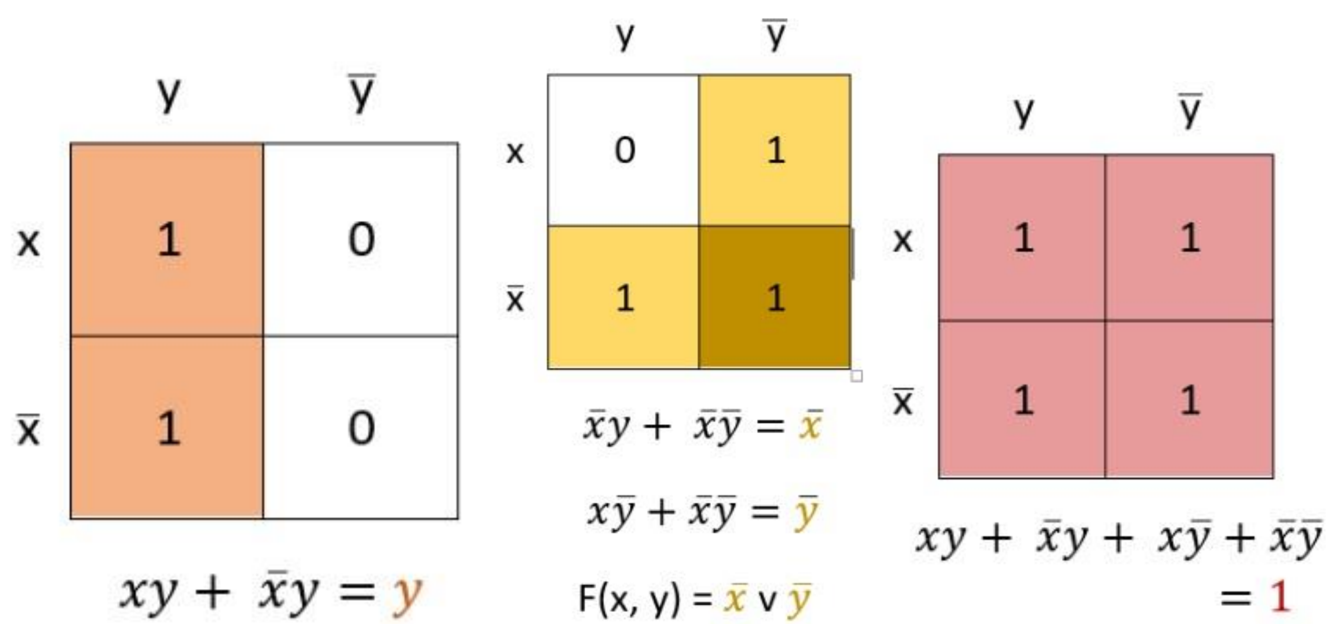


Diagrama 5

El objetivo es identificar los bloques de mayor tamaño posible y cubrir todos los unos con el menor número posible de bloques, usando en primer lugar los bloques más grandes y empleando siempre los bloques del mayor tamaño que sea posible.

Propiedades y resultados de los K-diagramas



Bloques de diagrama de Karnaugh para dos variables: “x”, “y”.
Extraído de: Dispositivos y circuitos electrónicos. (Donald A. Neamen., 2012).

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
x	0	0	1	0
$\bar{x}$	0	0	1	0

$$x\overline{yz} + \overline{xyz} = \overline{yz}$$

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
x	0	0	0	0
$\bar{x}$	1	0	0	1

$$\bar{x}yz + \overline{xy}z = \bar{x}z$$

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
x	0	1	1	0
$\bar{x}$	0	1	1	0

$$xy\bar{z} + x\overline{yz} + \bar{x}y\bar{z} + \overline{xyz} = \bar{z}$$

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
x	0	0	0	0
$\bar{x}$	1	1	1	1

$$\bar{x}yz + \bar{x}y\bar{z} + \overline{xyz} + \overline{xy}z = \bar{x}$$

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
x	1	1	1	1
$\bar{x}$	1	1	1	1

$$xyz + xy\bar{z} + x\overline{yz} + x\bar{y}z +$$
$$\bar{x}yz + \bar{x}y\bar{z} + \overline{xyz} + \overline{xy}z = 1$$

Bloques de diagrama de Karnaugh para tres variables: “x”, “y”, “z”.
Extraído de: Dispositivos y circuitos electrónicos. (Donald A. Neamen., 2012).

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
wx	0	0	0	0
$w\bar{x}$	1	0	0	1
$\overline{wx}$	0	0	0	0
$\bar{w}x$	0	0	0	0

$$w\bar{x}yz + w\overline{xy}z = w\bar{x}z$$

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
wx	0	0	0	0
$w\bar{x}$	0	0	0	0
$\overline{wx}$	1	1	1	1
$\bar{w}x$	0	0	0	0

$$\overline{wx}yz + \overline{wx}y\bar{z} + \overline{wxyz} + \overline{wxy}z = \overline{wx}$$

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
wx	1	0	0	1
$w\bar{x}$	0	0	0	0
$\overline{wx}$	0	0	0	0
$\bar{w}x$	1	0	0	1

$$wxyz + wx\bar{y}z + \bar{w}xyz + \bar{w}x\bar{y}z = xz$$

	yz	$y\bar{z}$	$\overline{yz}$	$\bar{y}z$
wx	0	1	1	0
$w\bar{x}$	0	1	1	0
$\overline{wx}$	0	1	1	0
$\bar{w}x$	0	1	1	0

$$wxy\bar{z} + wx\overline{yz} + w\bar{x}y\bar{z} + w\overline{xyz} + \overline{wx}y\bar{z} + \overline{wxyz} + \bar{w}xy\bar{z} + \bar{w}x\overline{yz} = \bar{z}$$

Bloques de diagrama de Karnaugh para cuatro variables: "x", "y", "z", "w".
Extraído de: Dispositivos y circuitos electrónicos. (Donald A. Neamen., 2012).

3.3Método de Quine-McCluskey

El método de Quine-McCluskey cumple con la necesidad de utilizar un procedimiento para simplificar desarrollos en sumas de productos, que pueda automatizarse. Se puede utilizar para cualquier cantidad de variables. Fue desarrollado en los años cincuenta por W. V. Quine y E. J. McCluskey. Básicamente, el método de Quine-McCluskey consta de dos partes. La primera parte determina qué términos son candidatos para incluir en un desarrollo mínimo como suma booleana de productos booleanos y la segunda parte determina cuáles de esos términos se utilizan finalmente.

Ejemplo: obtener una expresión minimal equivalente a xyz + xy'z + x'y'z + x'y'z'.

Se representan los minitérminos del desarrollo mediante cadenas de bits. Donde aparezca una variable booleana en su estado normal (x), se escribirá 1 bit. En caso de aparecer con su complemento (x'), se escribirá 0 bits. Se sigue el proceso con el resto de las variables. Eventualmente se cuenta el número de unos que se encuentran en la columna de Cadena de bits, como se muestra en la siguiente tabla:

Tabla de valores		
Minitérmino	Cadena de bits	Número de unos
xyz	111	3
$x\overline{y}z$ $\overline{x}yz$	101 011	2 2
$\overline{xy}z$	001	1
$\overline{xyz}$	000	0

Tabla de valores.
Extraído de: Matemática discreta y sus aplicaciones. (Rosen K., 2004).

El método de Quine-McCluskey utiliza la siguiente secuencia de pasos para simplificar un desarrollo en suma de productos, en el siguiente ejemplo:

Ejemplo: minimizar con el método de Quine-McCluskey la siguiente fórmula booleana f(x,y,z,w) =wxyz + wxyz' + wxy'z' + wx'yz + wx'y'z + wx'y'z' + w'xy'z + w'x'yz + w'x'yz'

Notación en suma decimal: F(w,x,y,z)= ∑m(2,3,5,8,9,11,12,14,15)

Paso 1: se traducen las variables booleanas a bits, así como se cuenta el número de unos.

Tabla 1			
Término	Minitérmino	Cadena de bits	Número de unos
1	$wxyz$	1 1 1 1	4
2	$wxy\overline{z}$	1 1 1 0	3
3	$wx\overline{yz}$	1 1 0 0	2
4	$w\overline{x}yz$	1 0 1 1	3
5	$w\overline{xy}z$	1 0 0 1	2
6	$w\overline{xyz}$	1 0 0 0	1
7	$\overline{w}x\overline{y}z$	0 1 0 1	2
8	$\overline{wx}yz$	0 0 1 1	2
9	$\overline{wx}y\overline{z}$	0 0 1 0	1

Tabla 1

Paso 2: se organiza la tabla de menor a mayor en cuanto a número de unos, y se forman grupos respecto a la misma cantidad de unos.

Tabla 1				
Grupo	Término	Minitérmino	Cadena de bits	Número de unos
1	9 6	$\overline{wx}y\overline{z}$ $w\overline{xyz}$	0 0 1 0 1 0 0 0	1 1
2	3 5 7 8	$wx\overline{yz}$ $w\overline{xy}z$ $\overline{w}x\overline{y}z$ $\overline{wx}yz$	1 1 0 0 1 0 0 1 0 1 0 1 0 0 1 1	2 2 2 2
3	2 4	$wxy\overline{z}$ $w\overline{x}yz$	1 1 1 0 1 0 1 1	3 3
4	1	$wxyz$	1 1 1 1	4

Tabla 2

Paso 3: identificar cuáles cadenas de bits cambian en solo 1 bit, y reemplazarlo con un "-". Se compara el grupo n con el grupo n + 1 (el uno con el dos, el dos con el tres…)

Tabla 2					
Grupo	Término	Minitérmino	Cadena de bits	Representación en bits	Número de unos
1	9 8	$\overline{wx}y\overline{z}$ $\overline{wx}yz$	0 0 1 0 0 0 1 1	0 0 1 -	1
	6 3	$w\overline{xyz}$ $wx\overline{yz}$	1 0 0 0 1 1 0 0	1 - 0 0	
	6 5	$w\overline{xyz}$ $w\overline{xy}z$	1 0 0 0 1 0 0 1	1 0 0 -	
2	3 2	$wx\overline{yz}$ $wxy\overline{z}$	1 1 0 0 1 1 1 0	1 1 - 0	2
	5 4	$w\overline{xy}z$ $w\overline{x}yz$	1 0 0 1 1 0 1 1	1 0 - 1	
	8 4	$\overline{wx}yz$ $w\overline{x}yz$	0 0 1 1 1 0 1 1	- 0 1 1	
3	2 1	$wxy\overline{z}$ $wxyz$	1 1 1 0 1 1 1 1	1 1 1 -	3
	4 1	$w\overline{x}yz$ $wxyz$	1 0 1 1 1 1 1 1	1 - 1 1	

Tabla 3

Paso 4: se repite el mismo procedimiento del Paso 3, pero ahora formando grupos que difieran de dos bits, con base en la Tabla 2. Para esto, se toman en cuenta los términos cuyas representaciones en bits tengan un guion en la misma posición y difieran de un bit.

El proceso continúa, sin embargo, ya no existen más posibilidades en este ejemplo.

Paso 5: se traducen las representaciones en bits a variables booleanas. Estos términos serán llamados implicantes.

Tabla 3	
Grupo	Implicante
1	$\overline{xy}z$, $x\overline{zw}$, $x\overline{yz}$
2	$xy\overline{w}$, $x\overline{y}w$, $\overline{y}zw$
3	xyz, xzw

Tabla 4

Paso 6: se crea otra tabla donde se acomodan los implicantes y los valores de las variables booleanas en sistema decimal. Se colocará una X de acuerdo al minitérmino que aparezca en los valores decimales posteriores a la columna de los minitérminos (MT). Finalmente, se verifica que los implicantes seleccionados cubran todas las columnas.

1 1 1 1 = 15

1 1 1 0 = 14

1 1 0 0 = 12

1 0 1 1 = 11

1 0 0 1 = 9

1 0 0 0 = 8

0 1 0 1 = 5

0 0 1 1 = 3

0 0 1 0 = 2

Tabla 4										
IP	MT	2	3	5	8	9	11	12	14	15
xzw	11,15						X			X
xyz	14,15								X	X
$\overline{y}zw$	3,11		X				X			
$x\overline{y}w$	9,11					X	X			
$xy\overline{w}$	12,14							X	X	
$x\overline{yz}$	8,9				X	X				
$x\overline{zw}$	8,12				X			X		
$\overline{xy}z$	2,3	X	X							
$\overline{w}x\overline{y}z$	5			X						

Tabla 5

Tabla 4										
IP	MT	2	3	5	8	9	11	12	14	15
xzw	11,15						X			X
xyz	14,15								X	X
$\overline{y}zw$	3,11		X				X			
$x\overline{y}w$	9,11					X	X			
$xy\overline{w}$	12,14							X	X	
$x\overline{yz}$	8,9				X	X				
$x\overline{zw}$	8,12				X			X		
$\overline{xy}z$	2,3	X	X							
$\overline{w}x\overline{y}z$	5			X						

Tabla 6

Paso 7: Los implicantes que solo tengan una X se agregan de inmediato a la función minimizada. Sin embargo, para que el método se considere concluido, se deben buscar más opciones para cubrir todas las columnas de la Tabla 4. Se seleccionan las siguientes, por ejemplo:

Tabla 4										
IP	MT	2	3	5	8	9	11	12	14	15
xzw	11,15						X			X
xyz	14,15								X	X
$\overline{y}zw$	3,11		X				X			
$x\overline{y}w$	9,11					X	X			
$xy\overline{w}$	12,14							X	X	
$x\overline{y}\overline{z}$	8,9				X	X				
$x\overline{z}\overline{w}$	8,12				X			X		
$\overline{x}\overline{y}z$	2,3	X	X							
$\overline{w}x\overline{y}z$	5			X						

Tabla 7

Paso 8: Los implicantes coloreados serán los minitérminos de la función minimizada. A partir de este punto, con ayuda del Álgebra de Boole, se reduce más la expresión de ser posible.

Función minimizada:

f(x,y,z,w) = xyz + y'zw + xy'w + xz'w' + x'y'z + w'xy'z

4.Ejercicios

1. Resolver la siguiente función Booleana como producto MAXTERM.
   F(a,b,c)=a+b´*c

   Primero, se obtiene la tabla de verdad de la función y luego se toman los MAXTERM. Enseguida, se evalúa la función para todas las combinaciones y se toman los MAXTERM de la tabla para los cuales la función lógica vale 0.

Tabla de verdad de la función lógica $a + b' \cdot c$ con MAXTERM.

Valor decimal	a	b	c	$F(a, b, c)$	MAXTERM
0	0	0	0	0	$M_0 = a+b+c$
1	0	0	1	1	
2	0	1	0	0	$M_2 = ab'c$
3	0	1	1	0	$M_3 = ab'c'$
4	1	0	0	1	
5	1	0	1	1	
6	1	1	0	1	
7	1	1	1	1	

Por lo tanto, la respuesta es:
F(a,b,c) = (a + b + c)(a + b´ + c)(a +b´ + c´)

También se puede expresar como:
F(a,b,c) = II M (0,2,3)
Por lo que el producto de los MAXTERM es 0,2,3.

2. Dada la siguiente tabla de verdad, hallar la expresión booleana minimizada.

A	B	C	F
0	0	0	1
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

Paso 1. Hallar los Minitérminos:

F = A’B’C’ + A’B’C + A’BC’ + A’BC + AB’C’ + AB’C + ABC’ + ABC

Paso 2. Simplificar mediante las propiedades hasta llegar a la expresión mínima:
F = A’B’(C’+C) + A’BC’ + A’BC + AB’C’ + AB’C + AB(C’ + C)
F = A’B’ + A’BC’ + A’BC + AB’C’ + AB’C + AB
F = A’B’ + A’BC’ + A’BC + AB’(C’ + C) + AB
F = A’B’ + A’BC’ + A’BC + AB’ + AB
F = A’B’ + A’B(C’ + C) AB’ + AB
F = A’B’ + A’B + AB’ + AB

F = A’(B’ + B) + A(B’ + B)
F = A’ + A
Finalmente la expresión queda: 1

3. Expresar como una suma de MINTERM la función booleana:
F(a,b,c)=a+b´*c
Primero se obtiene la tabla de verdad y luego se toman los MINTERM.
Posteriormente, se evalúa la función para todas las combinaciones y se toman los MINTERM de la tabla para los cuales la función vale 1.

Tabla de verdad de la función lógica $a + b' \cdot c$ con MINTERM.

Valor decimal	a	b	c	F(a, b, c)	MINTERM
0	0	0	0	0	
1	0	0	1	1	$m_1 = a'b'c$
2	0	1	0	0	
3	0	1	1	0	
4	1	0	0	1	$m_4 = ab'c'$
5	1	0	1	1	$m_5 = ab'c$
6	1	1	0	1	$m_6 = abc'$
7	1	1	1	1	$m_7 = abc$

Por lo tanto, la respuesta es:
F(a,b,c)=a’b’c+ab’c’+ab’c+abc’+abc
Otra notación que se puede utilizar es:
F(a,b,c)=∑m(1,4,5,6,7)
Lo que significa la suma de los MINTERM 1,4,5,6,7.

4. Usando el método Quine-McCluskey:
F(a,b,c,d,e)= ∑m(1,2,3,5,9,10,11,18,19,20,21,23,25,26,27)

Tabla de combinaciones

Mintér-minos	Forma Binaria A	B	C	D	E	No. de 1’s
m_1	0	0	0	0	1	1
m_2	0	0	0	1	0	1
m_3	0	0	0	1	1	2
m_5	0	0	1	0	1	2
m_9	0	1	0	0	1	2
m_{10}	0	1	0	1	0	2
m_{11}	0	1	0	1	1	3
m_{18}	1	0	0	1	0	2
m_{19}	1	0	0	1	1	3
m_{20}	1	0	1	0	0	2
m_{21}	1	0	1	0	1	3
m_{23}	1	0	1	1	1	4
m_{25}	1	1	0	0	1	3
m_{26}	1	1	0	1	0	3
m_{27}	1	1	0	1	1	4

MINIMIZACIÓN DE FUNCIONES BOOLEANAS – EQUIPO6

		CUBOS - 0	No. 1´s
√	1	0 0 0 0 1	1
√	2	0 0 0 1 0	
√	3	0 0 0 1 1	2
√	5	0 0 1 0 1	
√	9	0 1 0 0 1	
√	10	0 1 0 1 0	
√	18	1 0 0 1 0	
√	20	1 0 1 0 0	
√	11	0 1 0 1 1	3
√	19	1 0 0 1 1	
√	21	1 0 1 0 1	
√	25	1 1 0 0 1	
√	26	1 1 0 1 0	
√	23	1 0 1 1 1	4
√	27	1 1 0 1 1	

	CUBOS - 1	
√	1, 3	000X1
	1, 5	00X01
√	1, 9	0X001
√	2, 3	0001X
√	2, 10	0X010
√	2, 18	X0010
√	3, 11	0X011
√	3, 19	X0011
	5, 21	X0101
√	9, 11	010X1
√	9, 25	X1001
√	10, 11	0101X
√	10, 26	X1010
√	18, 19	1001X
√	18, 26	1X010
	20, 21	1010X
√	11, 27	X1011
	19, 23	10X11
√	19, 27	1X011
	21, 23	101X1
√	25, 27	110X1
√	26, 27	1101X

	Cubos - 2	
	1, 3, 9, 11	0X0X1
√	2, 3, 10, 11	0X01X
√	2, 3, 18, 19	X001X
√	2, 10, 18, 26	XX010
√	3, 11, 19, 27	XX011
	9, 11, 25, 27	X10X1
√	10, 11, 26, 27	X101X
√	18, 19, 26, 27	1X01X

Cubos - 3	
2, 3, 10, 11, 18, 19, 26, 27	XX01X

IMPLICANTES PRIMOS	
1, 5	00X01
5, 21	X0101
20, 21	1010X
19, 23	10X11
21, 23	101X1
1, 3, 9, 11	0X0X1
9, 11, 25, 27	X10X1
2, 3, 10, 11, 18, 19, 26, 27	XX01X

		1	2	3	5	9	10	11	18	19	20	21	23	25	26	27
	00X01	√			√											
	X0101				√							√				
*	1010X										√	√				
	10X11									√			√			
	101X1											√	√			
	0X0X1	√		√		√		√								
*	X10X1					√		√						√		√
*	XX01X		√	√			√	√	√	√					√	√
			√	√		√	√	√	√	√	√	√		√	√	√

Implicantes primos esenciales

1010X→AB’CD’

X10X1→BC’E’

XX01X→C’D

Una vez que tenemos los implicantes primos esenciales, ya tenemos una parte de la función simplificada. Ahora tendremos que ver los implicantes primos secundarios que tienen menor costo, es decir, que tienen menos variables en el término.

Teorema: 'Sean a y b dos términos implicantes primos de una tabla reducida de tal manera que el costo de a sea menor o igual al costo de b. Entonces, si a domina a b o si a y b son intercambiables existe una suma mínima de productos que no incluye a b.

Utilizamos los implicantes que no se han tomado en cuenta como implicantes primos esenciales, con las columnas que no han sido eliminadas de la tabla de simplificación de los implicantes primos para buscar los implicantes primos secundarios.

		1	5	23
	00X01	√	√	
	X0101		√	
	10X11			√
	101X1			√
	0X0X1	√		

El implicante primo 00X01 cubre el mintérmino mi y el mintérmino m5 mientras que el implicante primo X0101 cubre únicamente al mintérmino m5 entre estos dos el que tiene menos costo es el 00X01, luego el implicante 10X11 y 101X1 ambos cubren únicamente al mintérmino m2s tomamos cualquiera de los dos.

		1	5	23
*	00X01	√	√	
*	10X11			√
	0X0X1	√		
		√	√	√

Implicantes primos secundarios:

00X01→A’B’D’E 10X11→AB’DE

Como se puede observar, existen implicantes primos esenciales y secundarios. La función resultante simplificada es la operación OR de todos estos implicantes.

F(A, B, C, D, E) = AB'CD' + BC'E + C'D + A'B'D'E + AB'DE

5. Obtener la expresión mínima de la siguiente fórmula booleana con el método de Karnaugh.

   F(A,B,C)=ABC' + A'BC' + ABC + AB'C

   Para construir el diagrama de Karnaugh, se hace una tabla de verdad para visualizar de forma binaria los minitérminos de la fórmula booleana.

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

Se construye el diagrama de Karnaugh para una función booleana de tres variables.

$A\backslash^{BC}$	00	01	11	10
0				
1				

Se colocan números uno en los recuadros donde los valores de las variables coinciden con los que están en la tabla de verdad, que, además, a su vez, tienen un número uno en la columna de salida. Las casillas vacías se rellenan con ceros.

$A\backslash^{BC}$	00	01	11	10
0	0	0	0	1
1	0	1	1	1

Se marcan las adyacencias que se identifiquen.

$A\backslash^{BC}$	00	01	11	10
0	0	0	0	1
1	0	1	1	1

Para transcribir las adyacencias a minitérminos, se verifica que los valores de las variables en el diagrama de Karnaugh se mantienen constantes. Por ejemplo, la adyacencia gris tiene a A como constante con 1, pero B no lo es, por lo que no se agrega. C sí lo es. Se hace el mismo análisis con la otra adyacencia para obtener la fórmula minimizada.

$$F(A,B,C) = AC + B\bar{C}$$

Los minitérminos obtenidos se conectan con sumas para obtener una función en una forma normal disyuntiva.

6. Obtener la expresión mínima de la siguiente fórmula booleana con el método de Karnaugh.

$$F(A,B,C,D) = \bar{A}BCD + A\overline{BC}D + AB\bar{C}D + \bar{A}B\bar{C}D + A\overline{BCD} + A\bar{B}CD$$

Para construir el diagrama de Karnaugh, se hace una tabla de verdad para visualizar de forma binaria los minitérminos de la fórmula booleana.

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

Se construye el diagrama de Karnaugh para una función booleana de tres variables.

$AB\backslash^{CD}$	00	01	11	10
00				
01				
11				
10				

Se colocan números uno en los recuadros donde los valores de las variables coinciden con los que están en la tabla de verdad, que, además, a su vez, tienen un número uno en la columna de salida. Las casillas vacías se rellenan con ceros.

$AB\backslash^{CD}$	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	0	0
10	1	1	1	0

Se marcan las adyacencias que se identifiquen.

$AB\backslash^{CD}$	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	0	0
10	1	1	1	0

Para transcribir las adyacencias a minitérminos, se verifica que los valores de las variables en el diagrama de Karnaugh se mantienen constantes. Por ejemplo, la adyacencia verde tiene a A, B y D constantes, pero no a C. Por lo tanto, la C se ignora, y las demás variables se incluyen en la forma minimizada.

$$F(A,B,C,D) = \bar{A}BD + B\bar{C}D + A\overline{BC} + A\bar{B}D$$

Los minitérminos obtenidos se conectan con sumas para obtener una función en una forma normal disyuntiva.

7. Expresar la siguiente función booleana como una suma de MINTERM mediante el uso de los teoremas de expansión canónica:

   F(a,b,c) = a + b´ * c

$$a + b'c$$
$$a(b + b')(c + c') + b'c(a + a')$$
$$(ab + ab')(c + c') + b'ca + b'c\,a'$$
$$abc + abc' + ab'c + ab'c' + ab'c + a'b'c$$
$$a'b'c + ab'c' + ab'c + abc' + abc$$

Entonces:

$$F(a,b,c) = a'b'c + ab'c' + ab'c + abc' + abc$$

8. Sea la función lógica:

   F(a,b,c) = a * (b + c)

   Obtener los MINTERMS y MAXTERMS asociados

En la tabla 9.2 se muestran los *MINTERM* y los *MAXTERM* asociados con cada combinación en una tabla de verdad de tres variables lógicas, con $2^3 = 8$ *MINTERM* y *MAXTERM*.

Tabla 9.2 MINTERM y MAXTERM de la función booleana $a \cdot (b + c)$.

Valor decimal	a	b	c	F(a, b, c)	MINTERM	MAXTERM
0	0	0	0	0	$m_0 = a'b'c'$	$M_0 = a+b+c$
1	0	0	1	0	$m_1 = a'b'c$	$M_1 = a+b+c'$
2	0	1	0	0	$m_2 = a'bc'$	$M_2 = a+b'+c$
3	0	1	1	0	$m_3 = a'bc$	$M_3 = a+b'+c'$
4	1	0	0	0	$m_4 = ab'c'$	$M_4 = a'+b+c$
5	1	0	1	1	$m_5 = ab'c$	$M_5 = a'b+c'$
6	1	1	0	1	$m_6 = abc'$	$M_6 = a'+b'+c$
7	1	1	1	1	$m_7 = abc$	$M_7 = a'+b'+c'$

9. Expresar la siguiente función booleana como un producto MAXTERM:

   F(a,b,c) = a + b’ * c

   Primero, se obtiene la tabla de verdad de la función y luego se toman los MAXTERM desde dicha tabla de verdad. Enseguida, se evalúa la función para todas las combinaciones y se toman los MAXTERM de la tabla para los cuales la función lógica vale 0.

Solución

En la tabla 9.2 se muestran los *MINTERM* y los *MAXTERM* asociados con cada combinación en una tabla de verdad de tres variables lógicas, con $2^3 = 8$ *MINTERM* y *MAXTERM*.

Tabla 9.2 MINTERM y MAXTERM de la función booleana $a \cdot (b + c)$.

Valor decimal	a	b	c	F(a, b, c)	MINTERM	MAXTERM
0	0	0	0	0	$m_0 = a'b'c'$	$M_0 = a+b+c$
1	0	0	1	0	$m_1 = a'b'c$	$M_1 = a+b+c'$
2	0	1	0	0	$m_2 = a'bc'$	$M_2 = a+b'+c$
3	0	1	1	0	$m_3 = a'bc$	$M_3 = a+b'+c'$
4	1	0	0	0	$m_4 = ab'c'$	$M_4 = a'+b+c$
5	1	0	1	1	$m_5 = ab'c$	$M_5 = a'b+c'$
6	1	1	0	1	$m_6 = abc'$	$M_6 = a'+b'+c$
7	1	1	1	1	$m_7 = abc$	$M_7 = a'+b'+c'$

Entonces la respuesta es:

F(a,b,c) = (a + b + c)(a + b’ + c)(a + b’ + c’)

Que puede expresarse como:

F(a,b,c) = m (0,2,3)

10. Demuestra la siguiente equivalencia booleana.

    XYZ + XZ + Z = Z

XYZ + XZ + Z = XYZ + Z (X+1)

= XYZ + Z (1)

= XYZ + Z

= Z (XY + 1)

= Z (1)

= Z

6.Software

- Software 1 **JavaScript Boolean Expression Simplifier** es una herramienta diseñada para **simplificar expresiones booleanas** escritas en JavaScript. Su objetivo principal es reducir expresiones lógicas complejas a su forma más sencilla, sin alterar su significado lógico. Esto facilita la **lectura, mantenimiento y optimización** del código.
- **Minimizador de Funciones Booleanas (Método Quine-McCluskey)**
  Este software implementa el **algoritmo de Quine-McCluskey** para la minimización de funciones booleanas, permitiendo trabajar con **una hasta ocho variables booleanas**. La función a minimizar se ingresa mediante su **tabla de verdad**, en la cual el usuario puede establecer manualmente los valores de salida (y) o generar un **ejemplo aleatorio** con el número de variables seleccionado.
  Cada combinación de entrada se representa en binario, y los **minitérminos** se identifican fácilmente (por ejemplo, la expresión AB'C'D se convierte en 1001). La interfaz permite marcar estas combinaciones de forma visual e interactiva.
- El programa "Mapa de Karnaugh - 4 Variables" es una herramienta interactiva diseñada para simplificar expresiones booleanas utilizando el método de Karnaugh. Este método es ampliamente utilizado en el diseño de circuitos digitales y en la optimización de funciones booleanas. A continuación, se detallan las características y el funcionamiento del programa.

7.Conclusión

La simplificación de funciones booleanas es esencial en el diseño de circuitos digitales, ya que permite reducir la complejidad lógica, el número de compuertas y, por ende, el consumo energético y los costos de implementación.

Durante el desarrollo del proyecto se abordaron tres métodos de minimización: el algebraico, los mapas de Karnaugh y el de Quine-McCluskey. Cada uno tiene características que lo hacen útil en diferentes contextos. El método algebraico es adecuado para expresiones sencillas, los mapas de Karnaugh resultan especialmente prácticos para representar y simplificar funciones de hasta cinco variables de forma visual, y el método de Quine-McCluskey, aunque más complejo, es ideal para ser implementado computacionalmente en funciones de muchas variables, permitiendo automatizar el proceso de simplificación.

Estos enfoques permiten una comprensión profunda de la lógica booleana y dotan al estudiante de herramientas fundamentales para la optimización de sistemas digitales.

8.Bibliografía

Godínez, C. H., & Herrera, C. J. (2011). *Álgebra lineal, teoría y ejercicios*.

Briand, E. (2011). *Introducción a la matemática discreta*.

Rosen, K. (2004). *Matemática discreta y sus aplicaciones* (5ta ed.).

López Medina, L. (2003). *Método didáctico de simplificación de funciones booleanas*.

Villalpando Becerra, J. F., & Garcia Sandoval, A. (2014). *Matemáticas discretas*.

Vilchez Quesada, E. (2015). *Estructuras discretas con Mathematica*.

López Medina, L. E. (2003). *Proyecto de funciones booleanas*.

Neamen, D. A. (2012). *Dispositivos y circuitos electrónicos*.

Grimaldi, P. R. (1985). *Discrete and combinational mathematics*.

Taylor, J., & Garnier, R. (2009). *Discrete mathematics*.