INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
Facultad de Ingeniería y Ciencias Básicas

Unidad 1 / Escenario 2

Lectura Fundamental

Álgebra booleana y métodos de simplificación

Contenido

Palabras clave: Álgebra booleana, simplificación, mapas de Karnaugh

1. Álgebra booleana

El álgebra booleana debe su nombre a George Boole, matemático que desarrolló las reglas y leyes que hoy en día se utilizan en los sistemas lógicos para su análisis y simplificación. Según Floyd (2010): "El álgebra de Boole es una forma adecuada y sistemática de expresar y analizar las operaciones de los circuitos lógicos". Su importancia radica en que facilita el diseño y análisis de circuitos digitales. Cada circuito digital presente en el mundo actual tiene implícitas estas reglas para su funcionamiento.

En el escenario anterior se vieron los tres operadores básicos que utiliza el álgebra booleana: NOT (inversor o negador), OR (suma lógica) y AND (producto lógico). Utilizando estos operadores se pueden hacer diversas expresiones, que conecten las variables de entrada de un circuito con sus respectivas salidas. Las expresiones resultantes se pueden representar mediante la combinación de compuertas lógicas.

Por ejemplo, la expresión $X(\overline{Y}+Z)$ se podrá representar según se muestra en la figura 1. Esto mismo se puede hacer para representar expresiones más complejas.

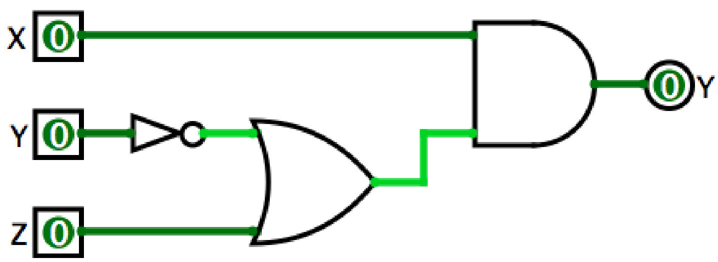


Figura 1: Magnitudes analógicas y digitales. *Elaboración propia, 2017.*

Antes de empezar a ver las leyes y reglas del álgebra de Boole, sugiero revisar en la actividad "Circuito y expresión booleana", que le permitirá verificar si la relación entre una expresión booleana y su circuito está clara.

1.1. Leyes del álgebra booleana

En el álgebra booleana se siguen las mismas leyes básicas del álgebra tradicional. En la tabla 1 se presentan las leyes, tanto para la suma como para la multiplicación.

	Suma	**Multiplicación**
Conmutativa	$A+B=B+A$	$AB=BA$
Asociativa	$A+(B+C)=(A+B)+C$	$A(BC)=(AB)C$
Distributiva	$A(B+C)=AB+AC$	

Tabla 1: Leyes del álgebra booleana. *Elaboración propia, 2017.*

La explicación de cada una de las leyes es la siguiente:

- La *ley conmutativa* indica que el orden de las entradas, al momento de realizar una operación OR o una AND, no altera el resultado final.

- La *ley asociativa* indica que si se tienen más de dos variables al realizar una operación OR o AND, no importa cómo estas se encuentren agrupadas, el resultado será el mismo.
- La *ley distributiva* indica que si se desea expandir una expresión, se multiplica dato por dato. Esto también permite factorizar expresiones con términos comunes.

1.2. Reglas del álgebra booleana

Las diferentes reglas del álgebra booleana se encuentran resumidas en la tabla 2, donde A, B y C son variables booleanas. Adicionalmente, en el escenario encontrará una explicación más completa de las diferentes reglas. Lo invito a revisar el recurso mencionado y posteriormente regresar a esta lectura.

1.	$A + 0 = A$
2.	$A + 1 = A$
3.	$A \cdot 0 = 0$
4.	$A \cdot 1 = A$
5.	$A + A = A$
6.	$A + \overline{A} = 1$
7.	$A \cdot A = A$
8.	$A \cdot \overline{A} = 0$
9.	$\overline{\overline{A}} = A$
10.	$A + AB = A$
11.	$A + \overline{A}B = A + B$
12.	$(A + B)(A + C) = A + BC$

Tabla 2: Reglas del álgebra booleana. *Fuente: (Floyd, 2010).*

1.3. Teoremas de De Morgan

Se trata de dos teoremas de gran importancia que apoyan el álgebra booleana y que permiten pasar de expresiones tipo suma a productos, y viceversa. Estos teoremas se pueden ver en la figura 3.

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

Figura 2: Teoremas de De Morgan. *Elaboración propia, 2017.*

1.4. Simplificación mediante álgebra booleana

Según Floyd (2010): "Una expresión booleana simplificada emplea el menor número posible de puertas en la implementación de una determinada expresión". Esta es la importancia de la aplicación de las leyes y reglas del álgebra de Boole, junto con los teoremas de De Morgan. Estas últimas son ideales para eliminar negaciones (operaci ón NOT) que estén aplicadas a varias variables al tiempo. A continuación, se encuentran algunos ejemplos de simplificación de expresiones booleanas.

Dada la expresión:

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

Se puede aplicar primero el segundo teorema de De Morgan a toda la expresión, obteniendo:

$$\overline{\overline{A}}\;\overline{B}$$

La doble negación presente en A, se puede eliminar utilizando la regla 9 del álgebra de Boole. Por lo tanto, la expresión quedaría reducida a:

$$A\overline{B}$$

Un ejemplo más complejo puede ser la expresión:

$$\overline{(\overline{(\overline{W}+X)+\overline{Y}})(\overline{\overline{Y}+\overline{Z}})}$$

En este caso, nuevamente se aplica el primer teorema de De Morgan, para eliminar la negación grande que cubre todos los términos. Con este fin, se asume que $A = (\overline{(\overline{W}+X)+\overline{Y}})$ y $B = (\overline{\overline{Y}+\overline{Z}})$:

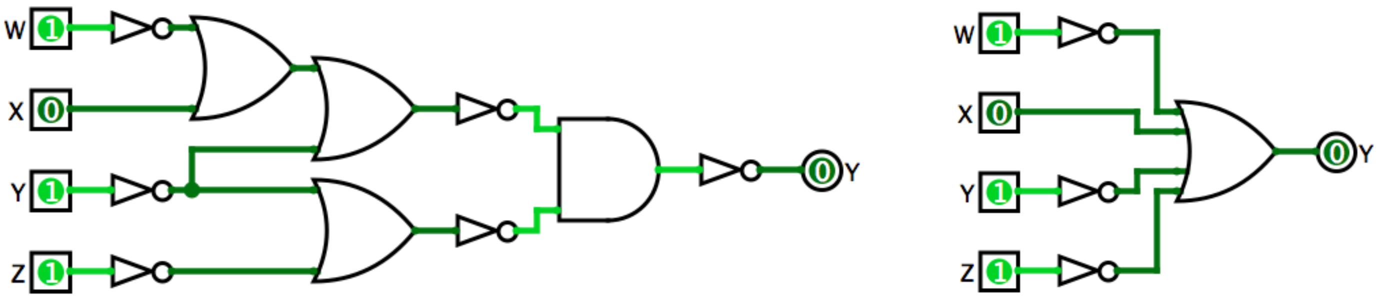
$$(\overline{\overline{\overline{(\overline{W}+X)+\overline{Y}}}}) + (\overline{\overline{\overline{\overline{Y}+\overline{Z}}}})$$

La nueva expresión tiene dobles negaciones que se pueden eliminar usando la regla 9. A la expresión resultante se le pueden eliminar los paréntesis y queda:

$$((\overline{W}+X)+\overline{Y})+(\overline{Y}+\overline{Z}) = \overline{W}+X+\overline{Y}+\overline{Y}+\overline{Z}$$

En la expresión obtenida aparecen los términos $\overline{Y}+\overline{Y}$. Aplicando la ley 5, esto es igual $\overline{Y}$. Por lo tanto, la expresión se reduce a:

$$\overline{W}+X+\overline{Y}+\overline{Z}$$



(a) Diagrama original.

(b) Diagrama simplificado.

Figura 3: Resultado de la simplificación mediante álgebra booleana. *Elaboración propia, 2017.*

La expresión obtenida es más sencilla tanto de entender como de implementar utilizando compuertas lógicas. Esto se puede ver en la figura 3.

En este momento, lo invito a volver al escenario, a la siguiente actividad que le permitirá utilizar lo visto en esta sección sobre la simplificación de circuitos y expresiones lógicas mediante álgebra booleana.

2. Representación de funciones lógicas

En los sistemas digitales es posible representar cualquier tipo de problema mediante **tablas de verdad**. "Las tablas de verdad indican el valor que debe tomar la salida o salidas del sistema para cada una de las combinaciones de las entradas" (Tokheim, 2011). La ventaja de utilizar este método para analizar y diseñar circuitos lógicos radica en la facilidad para relacionar sus entradas y salidas.

La tabla 3 muestra un ejemplo de tabla de verdad, para un problema con tres entradas y una salida. En este ejemplo, el sistema sólo puede generar un "1" (o nivel ALTO) siempre que dos (sólo dos) de sus entradas tengan su valor en "1" al tiempo. Para cualquier otro caso, el sistema genera una salida en "0".

Entradas			**Salidas**
A	**B**	**C**	**Y**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	0

Tabla 3: Representación de un problema mediante una tabla de verdad. *Elaboración propia, 2017.*

Una vez representado un problema mediante su tabla de verdad, será posible obtener dos tipos de expresiones estándar: La suma de productos (SOP) y el producto de sumas (POS). Estas expresiones serán posteriormente utilizadas para generar expresiones simplificadas del problema y su representación mediante compuertas lógicas.

2.1. Suma de productos - SOP

Una expresión "suma de productos" (SOP, *Sum Of Products* en inglés) está conformada por varios términos producto (multiplicación booleana) de literales (variable afirmada o negada) que se agrupan en una suma booleana. Cuando dos o más términos están agrupados en una misma negación (ej: $\overline{ABC}$, es necesario aplicar las reglas de álgebra booleana hasta obtener las variables separadas (literales). Así, los siguientes son ejemplos de SOP:

$$\overline{A} + B + \overline{C}$$

$$AB + \overline{A}BC + C$$

Mientras que el siguiente no es una SOP (Nótese que hay varios términos agrupados en una misma negación):

$$\overline{AB} + \overline{\overline{A}BC + C}$$

Partiendo del problema de los números primos, la siguiente tabla de verdad (tabla 4) muestra los mintérminos y las expresiones correspondientes a cada una de las filas. Un mintérmino es un término en producto que contiene todas las variables de entrada para una salida en particular y está relacionado con su posición en la tabla de verdad. Por otra parte, las expresiones corresponden al producto de las variables de entrada, en el estado en el que se encuentran para una fila en particular: si el valor corresponde a "0" se niega la variable, esto hace que al aplicar el producto (compuerta AND) el resultado sea "1".

Entradas			**Salidas**		
A	**B**	**C**	**Y**	**mintérminos**	**Expresión**
0	0	0	0	m_0	$\overline{A}\ \overline{B}\ \overline{C}$
0	0	1	0	m_1	$\overline{A}\ \overline{B}\ C$
0	1	0	0	m_2	$\overline{A}\ B\ \overline{C}$
0	1	1	1	m_3	$\overline{A}\ B\ C$
1	0	0	0	m_4	$A\ \overline{B}\ \overline{C}$
1	0	1	1	m_5	$A\ \overline{B}\ C$
1	1	0	1	m_6	$A\ B\ \overline{C}$
1	1	1	0	m_7	$A\ B\ C$

Tabla 4: Tabla de verdad, mintérminos y expresiones asociadas. *Elaboración propia, 2017.*

Para obtener la expresión "suma de productos" a partir de esta tabla de verdad, *se tienen en cuenta únicamente los mintérminos asociados con una salida en "1"* (marcados en azul). Lo que se desea es obtener las expresiones necesarias para que al sumar dichos productos el resultado sea "1" únicamente en los casos requeridos por el problema. Esto se puede expresar de la siguiente manera:

$$Y = m(3,5,6) = m_3 + m_5 + m_6 = \overline{A}\ B\ C + A\ \overline{B}\ C + A\ B\ \overline{C}$$

El circuito resultante se puede ver en la figura 4 y corresponde a una posible solución del problema.

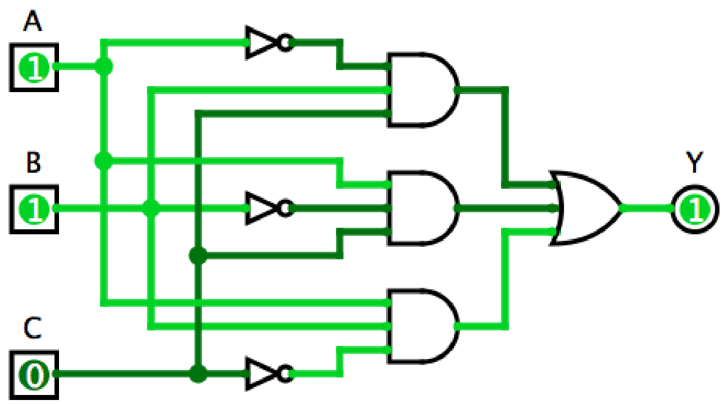


Figura 4: Circuito correspondiente a la SOP para el problema. *Elaboración propia, 2017.*

2.2. Producto de sumas - POS

Por otro lado, una expresión "producto de sumas" (POS, *Product Of Sums* en inglés) está conformada por varios términos suma (suma booleana) de literales que se agrupan en un producto booleano. Cuando dos o más términos están agrupados en una misma negación (ej: $\overline{A+B+C}$), es necesario aplicar las reglas de álgebra booleana hasta obtener las variables separadas (literales). Así, los siguientes son ejemplos de POS:

$$\overline{A}(B+C)(A+\overline{C})$$

$$(A+B)(\overline{A}+B+C)C$$

Partiendo del mismo problema de los números primos, la tabla 5 muestra los maxtérminos y las expresiones correspondientes a cada una de las filas. Un maxtérmino es un término suma que contiene todas las variables de entrada para una salida en particular y está relacionado con su posición en la tabla de verdad. Nótese que son las posiciones opuestas a los mintérminos de la tabla anterior. Además, en este caso los términos se toman con su valor opuesto (Si el término en la entrada en "1", se toma la variable negada).

Entradas			**Salidas**		
A	**B**	**C**	**Y**	**mintérminos**	**Expresión**
0	0	0	0	M_0	$A+B+C$
0	0	1	0	M_1	$A+B+\overline{C}$
0	1	0	0	M_2	$A+\overline{B}+C$
0	1	1	1	M_3	$A+\overline{B}+\overline{C}$
1	0	0	0	M_4	$\overline{A}+B+C$
1	0	1	1	M_5	$\overline{A}+B+\overline{C}$
1	1	0	1	M_6	$\overline{A}+\overline{B}+C$
1	1	1	0	M_7	$\overline{A}+\overline{B}+\overline{C}$

Tabla 5: Tabla de verdad, mintérminos y expresiones asociadas. *Elaboración propia, 2017.*

Para obtener la expresión "producto de sumas" de esta tabla de verdad, se tienen en cuenta únicamente los maxtérminos asociados con una salida en "0". Lo que se desea es obtener las expresiones necesarias para que el producto de sumas dé "0" en los casos que se requiere. Esto se puede expresar de la siguiente manera:

$Y = M(0,1,2,4,7) = (M_0)(M_1)(M_2)(M_4)(M_7) = (A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(\overline{A}+B+C)(\overline{A}+\overline{B}+\overline{C})$

El circuito resultante se puede ver en la figura 5, y corresponde a una solución del problema.

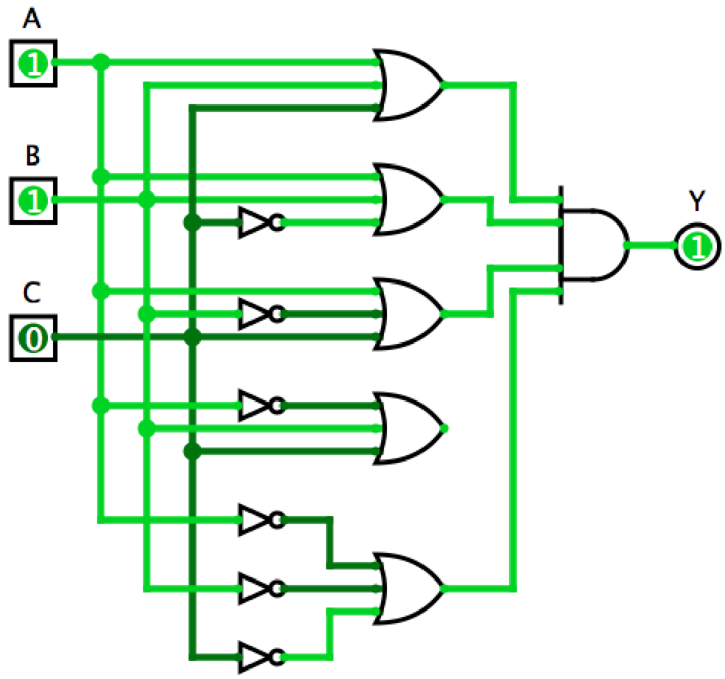


Figura 5: Circuito correspondiente al POS para el problema. *Elaboración propia, 2017.*

Una vez finalizada esta sección, es bueno volver al escenario para la siguiente actividad, que le permitirá practicar lo visto con respecto a la generación de circuitos dada una tabla de verdad.

3. Simplificación lógica de expresiones booleanas

Hay diferentes métodos para simplificación de expresiones booleanas. Uno de los más utilizados, cuando se trata de expresiones con pocas variables de entrada (máximo 5) es el método de los mapas de Karnaugh. Hay también métodos tabulares, como el Quine-McCluskey.

3.1. Mapas de Karnaugh

Los mapas de Karnaugh constituyen un método gráfico que facilita la simplificación de expresiones booleanas hasta de cinco variables. Sobrepasar ese límite dificulta la generación de los mapas y deja de ser un método práctico. Este método parte de las expresiones estándar, vistas en la sección anterior, y permite las expresiones más simplificadas posibles (expresiones mínimas).

En las lecturas complementarias encuentra una explicación completa de los mapas de Karnaugh, para la simplificación de funciones booleanas.

3.2. Método de Quine-McCluskey

Este es de tipo tabular y gráfico. A diferencia de los mapas de Karnaugh, permite trabajar con más variables. Además, el algoritmo se puede implementar en un programa de computador para resolución de problemas complejos, y asegurando la obtención de las expresiones mínimas del problema. Para finalizar este recorrido por los métodos de simplificación, es importante que regrese al escenario para hacer la actividad de evaluación final.

Índice de figuras

Índice de tablas

Referencias

Floyd, T. L. (2010). *Fundamentos de sistemas digitales.* Pearson Prentice Hall. (OCLC: 893578510)

Tokheim, R. L. (2011). *Electrónica Digital: Principios y Aplicaciones.* España: McGraw-Hill España. Descargado 2017-05-01, de https://library.biblioboard.com/content/05ed48f3-1297-46d7-97d2-ac0b5bdc2150 (OCLC: 963710488)

INFORMACIÓN TÉCNICA

Módulo: Sistemas digitales y ensambladores
Unidad 1: Álgebra booleana y métodos de simplificación
Escenario 2: Introducción a los sistemas digitales

Autor: Gabriel Eduardo Ávila Buitrago

Asesor Pedagógico: Jeimmy Lorena Romero Perilla
Diseñador Gráfico: Leonardo Stiglitch Campos
Asistente: Jhon Edwar Vargas Villa