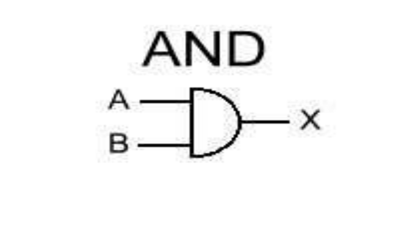
Álgebra Booleana

1. CONCEPTO DE ALGEBRA BOOLEANA

Es una rama especial del álgebra que se usa principalmente en electrónica digital. **El álgebra booleana** fue inventada en el año 1854 por el matemático inglés George Boole.

El álgebra de Boole es un conjunto cualquiera A en el que se han definido dos operaciones binarias denominadas suma lógica (+), también conocida como OR, y producto lógico (·), también conocido como AND, y una operación unitaria denominada complemento (a →ā), también conocida como NOT



El **álgebra de Boole** es un método para simplificar los circuitos lógicos (o a veces llamados circuitos de conmutación lógica) en electrónica digital.

Se llama tambien "**Cambio de álgebra**". Podemos representar el funcionamiento de los circuitos lógicos utilizando números, siguiendo algunas reglas, que son bien conocidas como "**Leyes del álgebra de Boole**".

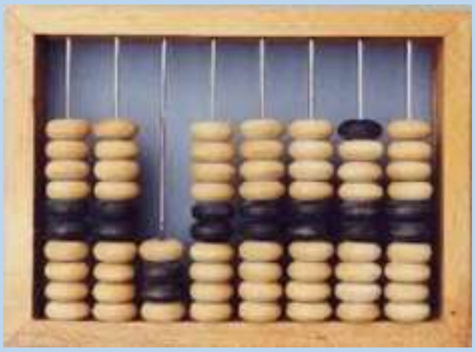
También podemos hacer los cálculos y las operaciones lógicas de los circuitos aún más rápido siguiendo algunos teoremas, que se conocen como "**Teoremas del álgebra de Boole**". Una función booleana es una función que representa la relación entre la entrada y la salida de un circuito lógico.

La lógica booleana solo permite dos estados del circuito, como True y False. Estos dos estados están representados por 1 y 0, donde 1 representa el estado "Verdadero" y 0 representa el estado "Falso".

Lo más importante para recordar en el **álgebra de Boole** es que es muy diferente al álgebra matemática regular y sus métodos. Antes de aprender sobre el álgebra de Boole, vamos a contar un poco sobre la **historia del álgebra de Boole** y su invención y desarrollo.

2. HISTORIA DEL ÁLGEBRA DE BOOLE

Como se mencionó anteriormente, el álgebra de Boole se inventó en el año de 1854, por el matemático inglés George Boole. Primero declaró la idea del álgebra de Boole en su libro "Una investigación de las leyes del pensamiento".



Después de esto, el álgebra de Boole es bien conocida como la forma perfecta para representar los circuitos lógicos digitales.

A fines del siglo XIX, los científicos Jevons, Schroder y Huntington utilizaron este concepto para términos modernizados. Y en el año de 1936, MHStone demostró que el álgebra de Boole es 'isomorfo' para los conjuntos (un área funcional en matemáticas).

En la década de 1930, un científico llamado Claude Shannon desarrolló un nuevo método de álgebra tipo "Cambio de álgebra" utilizando los conceptos de álgebra de Boole, para estudiar los circuitos de conmutación.

La síntesis lógica de las herramientas modernas de automatización electrónica se representa de manera eficiente mediante el uso de funciones booleanas conocidas como "Diagramas de decisión binarios".

El álgebra de Boole permite solo dos estados en un circuito lógico, como True y False, High and Low, Yes y No, Open and Close o 0 y 1.

3. LEYES E IDENTIDADES DEL ÁLGEBRA BOOLEANA

Al formular expresiones matemáticas para circuitos lógicos es importante tener conocimiento del álgebra booleana, que define las reglas para expresar y simplificar enunciados lógicos binarios. Una barra sobre un símbolo indica la **operación booleana NOT**, que corresponde a la inversión de una señal.

3.1 .PROPIEDADES

Se dice que (A ,+, ·,¯) es un álgebra de Boole si cumple las siguientes propiedades:

- **El conjunto A es cerrado para las dos operaciones**, es decir,∀ a, b ∈ A:

$a + b \in A$ $a \cdot b \in A$

- **Conmutativa**: ∀ a, b ∈ A:

$a + b = b + a$ $a \cdot b = b \cdot a$

- **Identidad:** Existencia el elemento identidad para las dos operaciones (para la + el elemento identidad es el 0 y para el · es el 1), es decir, ∀ a ∈ A:

$a + 0 = a$ $a \cdot 1 = a$

- **Distributiva:** Cada una de las operaciones es distributiva en relación con la otra, es decir, ∀ a, b, c ∈ A:

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

Complementario: Todo elemento de A tiene su complementario que verifica:

$\forall a \in A\ \exists \bar{a}\ \ a + \bar{a} = 1$ $\forall a \in A\ \exists \bar{a}\ \ a \cdot \bar{a} = 0$

3.2- Leyes fundamentales

Compuertas Lógicas Digitales

Nombre	Símbolo	Función
AND		F = xy
OR		F = x + y
Inversor		F = x’
Separador		F = x
NAND		F = (xy)’
NOR		F = (x + y)’
XOR		F= xy’ + x’y = x ⊕ y
NORX		F= xy + x’y’= x ⊗ y

1. OR

A + 0 = A

A + 1 = 1

A + A = A

A + A = 1

2. AND

A + 0 = 0

A + 1 = A

A + A = A

A + A = 0

3. NOT

A¨ = A

Los dos puntos en la A corresponde a dos barras de negación.

4. Leyes Conmutativas

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

5. Leyes Asociativas

$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

6. Leyes Distributivas

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

7. Otras identidades útiles

$$A + (A \cdot B) = A$$

$$A \cdot (A + B) = A$$

$$A + (A \cdot B) = A + B$$

$$(A + B) \cdot (A + B) = A$$

$$(A + B) \cdot (A + C) = A + (B \cdot C)$$

$$A + B + (A \cdot B) = A + B$$

$$(A \cdot B) + (B \cdot C) + (B \cdot C) = (A \cdot B) + C$$

$$(A \cdot B) + (A \cdot C) + (B \cdot C) = (A \cdot B) + (B \cdot C)$$

8. ejemplo de aplicación:

Se va a simplificar la siguiente expresión aplicando las leyes e identidades booleanas mencionadas:

$$E = (X \cdot Y \cdot Z) + (Y \cdot Z) + (X \cdot Y)$$

Es posible aplicar la ley asociativa y la ley fundamental de que $A \cdot 1 = A$:

$$E = X \cdot (Y \cdot Z) + 1 \cdot (Y \cdot Z) + (X \cdot Y)$$

Ahora es posible factorizar el termino $(Y \cdot Z)$:

$$E = (X + 1) \cdot (Y \cdot Z) + (X \cdot Y)$$

Dado que $A + 1 = 1$ según las leyes fundamentales por lo tanto $X + 1 = 1$:

$$E = 1 \cdot (Y \cdot Z) + (X \cdot Y)$$

Al realizar la operación tendremos ya simplificada la expresión:

$$E = (Y \cdot Z) + (X \cdot Y)$$

Aún podemos simplificar la expresión al factorizar Y:

$$E = Y \cdot (Z + X)$$

4. SIMPLIFICACIÓN DE FUNCIONES BOOLEANAS

Al usar los teoremas y leyes booleanas, podemos simplificar las expresiones booleanas, mediante las cuales podemos reducir el número requerido de compuertas lógicas a implementar. Podemos simplificar la función Boolean utilizando dos métodos:

1. El método algebraico: mediante el uso de identidades (leyes booleanas).
2. El método gráfico: utilizando el método del Mapa de Karnaugh.

5. MÉTODO DE KARNAUGH

Según el ejemplo que tenemos una función F(A,B,C) de tres variables, cuya tabla de verdad es:

Tabla de verdad de una función F(A,B,C)	
ABC	F
000	0
001	0
010	1
011	1
100	1
101	1
110	1
111	1

Si la desarrollamos por la primera forma canónica o de minitérminos, para ello tomamos aquellas entradas que nos hacen "1" la función (en celeste):

F = A'BC' + A'BC + AB'C' + AB'C + ABC' + ABC

Observamos, que cuando utilizamos minitérminos, las entradas con valor "0" están complementadas, y las entradas con valor "1" no están complementadas, es decir:

010 = A'BC'

Veremos cómo aplicando el método de Karnaugh podemos simplificar esta función.

Aplicamos Karnaugh para la tabla de verdad anterior, para ello dibujamos una tabla de la siguiente forma:

A \ BC	00	01	11	10
0	0	0	1	1
1	1	1	1	1

Se observa lo siguiente:

En total hay 8 casillas, cada una correspondiente a una fila de la tabla de verdad.

En cada casilla está colocado el valor de la función F, correspondiente a esa entrada.

En la tabla de verdad hay dos filas en las que **F=0** y seis filas en las que F=1.

En el nuevo diagrama hay dos casillas con "0" y seis con "1". Hay dos filas, en la primera fila están todos los valores de **F,** correspondientes a A=0, y en la segunda correspondientes a A=1.

Hay cuatro columnas, y el número que está en la parte superior de cada una de ellas nos indica los valores de las variables B y C en esa columna.

Dada una casilla cualquiera, mirando el número situado en la misma fila, a la izquierda del todo nos informa del valor de la variable A y los dos valores superiores, en la misma columna, nos dan los valores de B y C.

Así por ejemplo, si tomamos como referencia la casilla que está en la esquina inferior derecha, se corresponde con el valor que toma F cuando A=1, B=1 y C=0.

Entre dos casillas adyacentes cualesquiera, sólo varía una variable de entrada, quedando las otras dos con los mismos valores. Por ejemplo, si estamos en la casilla inferior derecha, en la que A=1, B=1 y C=0. Si vamos a la casilla que está a su izquierda obtenemos un valor de las variables de: A=1, B=1, C=1. Si lo comparamos los valores de las variables correspondientes a la casilla anterior, vemos que sólo ha cambiado una de las tres variables, la C. Lo mismo ocurre si nos desplazamos a cualquier otra casilla adyacente.

Ahora vamos a ver una propiedad “mágica” de esta tabla.

Si obtenemos la primera forma canónica, obtenemos una función con seis términos. Vamos a fijarnos sólo en los términos que obtenemos si desarrollamos sólo dos casillas adyacentes, como por ejemplos las marcadas en gris en la siguiente tabla:

A \ BC	00	01	11	10
0	0	0	1	1
1	1	1	1	1

Los valores de las variables en estas casillas son: A=1, B=1, C=1 y A=1, B=1, C=0. Si obtenemos los términos de la primera forma canónica y los sumamos:

ABC + ABC' = AB(C + C') = AB

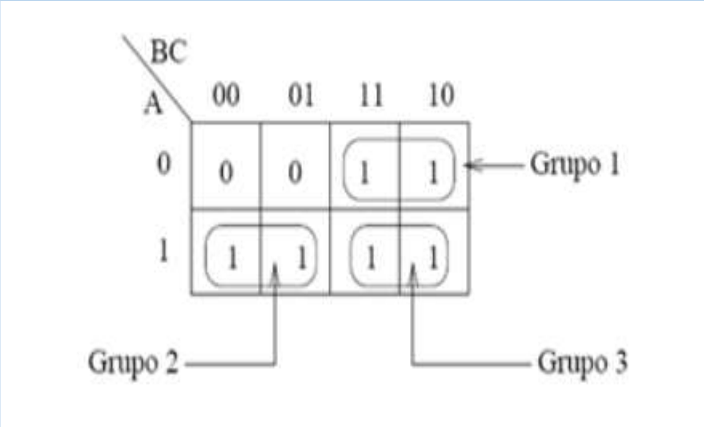
Por el hecho de agrupar los términos obtenidos de estas dos casillas y sumarlos, se han simplificado. Y esto es debido a la propiedad antes comentada de que entre dos casillas adyacentes sólo varía una de las variables, de manera que podemos sacar factor común.

Estos dos términos son los sumandos 5 y 6 de la primera forma canónica obtenida anteriormente, que al sumarlos y aplicar algunas propiedades se han simplificado.

Si nos fijamos en estas dos casillas adyacentes, la variable C, que es la única que varía de una a otra, ha desaparecido en la suma. De esta manera podemos afirmar lo siguiente:

"Si tomamos dos casillas adyacentes cuyo valor es '1' y desarrollamos por la primera forma canónica, desaparecerá una de las variables. Sólo permanecen las variables que no cambian de una casilla a otra".

De esta manera, vamos a ver que pasa si tomamos los siguientes grupos:



y sumamos los términos de estos grupos:

Grupo 1:A'BC + A'BC' = A'B(C + C') = **A'B**

Grupo 2: AB'C' + AB'C = AB' (C' + C) = **AB'**

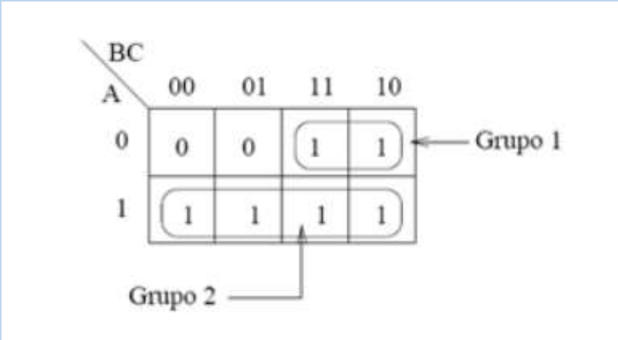
Grupo 3: El que teníamos antes: **AB**

Por tanto, la función F1 también la podemos expresar como suma de estos grupos:

F1 = A'B + AB' + AB

Como podemos observar, la función obtenida está ahora más simplificada. Pero...¿Se puede simplificar más?. Si.

Inicialmente la función F tenía seis sumandos, puesto que tenía 6 unos. Al hacer 3 grupos, ahora tiene 3 sumandos. ¿Podemos reducir el número de grupos? Si, vamos a ver qué pasa si tomamos los siguientes grupos:



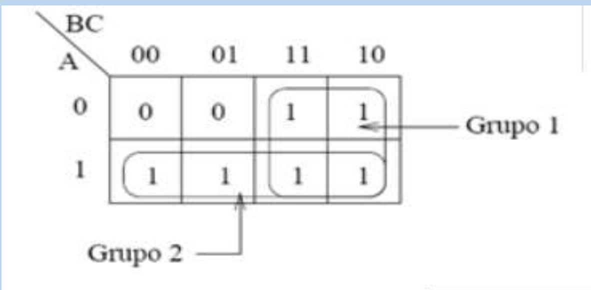
Ahora sólo hay 2 grupos. El nuevo grupo 2 está constituido por 4 casillas en las que F=1. La expresión de este grupo se obtiene sumando las expresiones de estas 4 casillas. Las nuevas expresiones de los grupos quedarían:

Grupo 1: Igual que antes: **A'B**

Grupo 2: AB' + AB = A (B' + B) = **A**

La nueva función F2 que obtenemos es: **F2 = A'B + A**

Todavía más simplificada que la anterior. Pero... ¿Es la más simplificada?. No, todavía podemos simplificarla más. ¿Por qué no podemos tomar 2 grupos de 4 casillas adyacentes?. Tomemos los grupos siguientes:



Las nuevas expresiones de los grupos son:

Grupo 1: A'B + AB = B (A' + A) = **B**

Grupo 2: Igual que antes: **A**

Por tanto, la nueva función **F3** simplificada es:

F3 = A + B

Por lo que la función queda simplificada al máximo.

5.1- Criterio de máxima simplificación:

Para obtener una función que no se puede simplificar más, hay que tomar el menor número de grupos con el mayor número de "1" en cada grupo.

Nos damos cuenta, que las tres funciones obtenidas son equivalentes, pero la más simplificada es F3.

F1 = A'B + AB' + AB

F2 = A'B + A

F3 = A + B

Por lo que deducimos que cuanto mayor sea el conjunto de "1" "adyacentes", mayor será la simplificación. Pero cuidado, sólo se pueden hacer grupos de 2 elementos, 4 elementos, 8 elementos, ... y siempre que éstos sean adyacentes.

Link SIMULADOR CIRCUITOS DIGITALES

https://dcaclab.com/es/lab?from_main_page=true