



SISTEMAS DIGITALES

LOGRO DE LA UNIDAD DE APRENDIZAJE

- Al término de la unidad, los alumnos, diseñarán e implementarán sistemas digitales básicos mediante el uso de simuladores que permite describir el funcionamiento interno de los circuitos digitales usados en la computadora.
- Al término de la unidad, los alumnos, describirán el funcionamiento de los sistemas digitales básicos usados en la computadora, haciendo uso de sistemas numéricos, voltajes y tiempos.

TEMARIO

- Álgebra Booleana
- Propiedades del álgebra de Boole
- Desarrollo de Sistemas Digitales
- Simplificación por Mapas de Karnaugh

ACTIVIDADES PROPUESTAS

- Los alumnos comprueban y utilizan el álgebra de Boole.
- Los alumnos desarrollan, en forma individual, distintos ejemplos de simplificación utilizando el método de Karnaugh.



1. ÁLGEBRA BOOLEANA

Para trabajar con los sistemas digitales se tuvo que desarrollar la matemática necesaria, esto fue posible, entre otros, gracias al matemático inglés George Boole (1810 - 1864), quien desarrolló la teoría en la que se basan los computadores para manipular los datos, desarrolló el álgebra de Boole. La lógica booleana está compuesta por una serie de operaciones que se aplican sobre las variables booleanas, las cuales sólo pueden tener dos valores o estados: verdadero o falso, set o reset, 1 ó 0, cerrado o abierto, etc. Las operaciones lógicas al actuar sobre las variables de entrada booleanas da lugar a una función booleana.

Para implementar un sistema digital necesitamos determinar el circuito lógico, el cual va a ser desarrollado por un conjunto de compuertas interconectadas, cada una de ellas realizando alguna operación lógica, cumpliendo lo establecido en el álgebra de Boole. En este circuito lógico se procesarán los bits, para ello se hace efectivo el "1" y el "0" a través de voltajes, se usa dos niveles de voltajes fijos: un nivel alto para el "1" y un nivel bajo para el "0".

Los sistemas digitales están representados a través de circuitos digitales, como el sistema digital es un conjunto de componentes que interactúan entre si y que tienen un objetivo, los componentes son los componentes básicos digitales, o sea las compuertas digitales (AND, OR, NOT, etc....). Al combinar las compuertas se da a lugar a otros tipos de elementos digitales como compuertas complejas, codificadores, memorias, flip-flops, microprocesadores, microcontroladores, etc.

Los sistemas digitales pueden resultar muy complejos, en realidad se construyen de un número muy grande de circuitos muy simples (compuertas digitales) pero siempre trabajan con variables digitales, las cuales solamente toman dos valores posibles: "0" y "1", si usamos interruptores, estos pueden tomar también dos estados posibles: "abierto" y "cerrado" o "Off" y "On", el estado abierto corresponde a "0" y el estado cerrado corresponde a "1".

Los posibles estados de las variables de entrada que afectan a los sistemas digitales se pueden representar en una tabla de estado también llamada tabla de estado.



Las tablas de verdad describen el funcionamiento del sistema digital, pueden tener varias columnas en la entrada, esto depende de la cantidad de variables de entrada, tiene además las columnas de las variables de salida, las cuales pueden ser una o más, dependiendo también del sistema digital. Podemos trabajar con una tabla para cada salida o una sola tabla para todas las salidas. La tabla presenta a la izquierda las variables de entrada y debajo de ellas los estados de las entradas, las cuales todas las combinaciones posibles de las entradas.

Número de combinaciones es igual a 2^n , donde n es el número de las variables de entrada de la tabla de verdad. Por ejemplo si la tabla tiene dos variables de entrada, tendrá 2^2 estados o sea 4 estados; si la tabla tiene 3 variables de entrada, entonces habrán: $2^3 = 8$ combinaciones, por lo tanto 8 estados.

La salida tomará el valor de cero o uno, en cada estado, dependiendo del sistema, tal como se puede apreciar en las dos tablas siguientes:

Tabla de estado

2 variables

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

3 variables

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

Para desarrollar un sistema digital debemos partir de su tabla de estado o tabla de verdad, en ella se debe haber establecido cómo debe responder cada un de las salidas frente a cada estado de entrada. Si tenemos la tabla de estado, de ella podemos generar la función lógica de la salida, luego la función lógica simplificada y finalmente hacer el circuito lógico en base a las compuertas lógicas.

Para comprender fácilmente esto, lo vamos a explicar en base a un ejemplo sencillo, usaremos para ello la tabla de estado de la compuerta OR de dos entradas.



Tabla de estado

A	B	Z	
0	0	0	m0
0	1	1	m1
1	0	1	m2
1	1	1	m3

Función Lógica Z

$$Z = \bar{A}.B + A.\bar{B} + A.B$$

$$01 \quad 10 \quad 11$$

Estados donde Z = 1

En la tabla de estado se puede apreciar las variables de entrada A y B, las cuales generan 4 estados diferentes, por otro lado vemos que la salida Z es igual a cero en el estado "00" y uno en los tres estados restantes. Para determinar la función lógica de la salida debemos tener en cuenta que esta se obtiene como una suma de términos, la cantidad de términos depende de la cantidad de unos que tenga la salida. Cada término depende del producto de las variables de entrada y del estado en el cual la salida vale "1". Para cada estado las variables se verán afectadas si para ese estado valen cero, así por ejemplo, en el estado 01 A toma el valor cero, por lo tanto el término será $\bar{A}B$.

$$Z = \sum m_i$$

Resumiendo podemos decir que la función lógica de la salida Z es igual a la suma de términos, donde Z vale "1". Si las variables fuesen A, B y C y en el estado 011 la salida Z es igual a "1", entonces, el término para este estado será $\bar{A}BC$, donde vemos que ha sido negado la variable A, mientras que B y C no han sido alterados ya que se quiere que este producto sea igual a uno en ese estado. Note que la función lógica solo considera los UNOS.

Luego de haber determinado la función lógica, esta expresión no debe ser usada para implementar el circuito lógico, antes se debe simplificar y para ello se necesita conocer el álgebra de Boole, con cuyas identidades podemos obtener una expresión más sencilla y resulte un circuito con el menor número de compuertas, como el obtenido a continuación.

$$Z = \bar{A}.B + A.\bar{B} + A.B$$

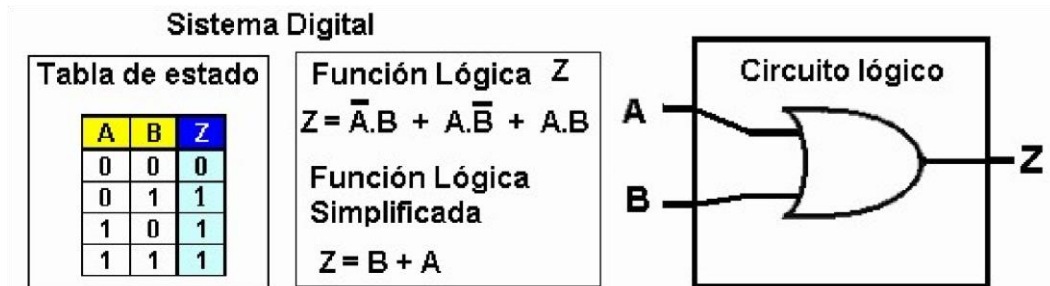
$$Z = \bar{A}.B + A(\bar{B} + B)$$

$$Z = \bar{A}.B + A \quad 1$$

$$Z = B + A$$

Función Lógica
Simplificada

Finalmente hemos obtenido la función lógica simplificada de Z y ésta nos va a servir para obtener el circuito lógico, el cual para este caso solo se trata de una compuerta OR.



Como vemos, para el desarrollo de un sistema digital, la determinación de la función lógica es sencilla, lo determinamos simplemente como una suma de términos, lo complicado es la simplificación y para ello debemos conocer las propiedades e identidades del álgebra de Boole.

1.1. PROPIEDADES DEL ÁLGEBRA DE BOOLE.

El álgebra de Boole presenta las siguientes propiedades:

- Ley de Idempotencia, es la propiedad para realizar una acción determinada varias veces y aun así conseguir el mismo resultado.

$$A \cdot A = A$$

$$A + A = A$$

- Ley Asociativa: $(A + B) + C = A + (B + C)$ / $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

- Ley Conmutativa: $A \cdot B = B \cdot A$ / $A + B = B + A$

- Ley Cancelativa: $(A \cdot B) + A = A$ / $(A + B) \cdot A = A$

- Ley de involución: $A = \overline{\overline{A}}$

- Ley Distributiva:
 $A + (B \cdot C) = (A + B) \cdot (A + C)$
 $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

- Absorción: $A + \bar{A} \cdot B = A + B$

- Leyes de Morgan: ley de las equivalencias que nos permiten trabajar con una sola compuerta, por ejemplo una NOR en lugar de usar dos inversores y una compuerta AND

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

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

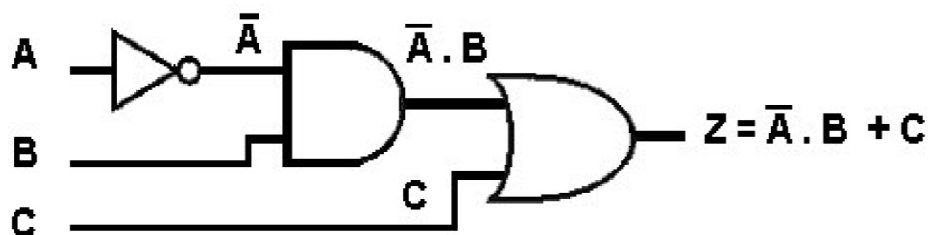
- Identidades del álgebra de Boole:

OR	AND
$A + 0 = A$	$A \cdot 0 = 0$
$A + 1 = 1$	$A \cdot 1 = A$
$A + A = A$	$A \cdot A = A$
$A + \bar{A} = 1$	$A \cdot \bar{A} = 0$

1.1.1 COMBINACIÓN DE OPERACIONES LÓGICAS

En un sistema digital y con mayor razón en un computador, las operaciones lógicas no se usan de forma aislada, sino que se usan en combinaciones en algunos casos extremadamente complejas, veamos un ejemplo simple, en donde la función lógica simplificada obtenida es la siguiente: $Z = \bar{A} \cdot B + C$

En esta expresión vemos primero un producto entre $\bar{A}$ y B , esto lo conseguimos con una compuerta AND la cual multiplica, necesitamos también un inversor para obtener $\bar{A}$ y finalmente una compuerta que suma la salida del AND con la entrada C . El circuito queda así:





Como todo sistema digital, éste debe tener su tabla de estado desde la cual se ha obtenido la función lógica que hemos utilizado para el circuito lógico. Si deseamos llenar la tabla de estado, primero debemos considerar que la tabla tiene como variables de entrada A, B y C, con 8 estados de entrada y la salida la obtendremos de la expresión inicial, reemplazando los valores que toma cada una de las variables en cada uno de los estados. Por ejemplo para el estado "000" obtendremos $Z = 1 \times 0 + 0 = 0$. De igual forma se puede obtener para los otros estados.

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

$$Z = \bar{A} \cdot B + C$$

1.2. DESARROLLO DE SISTEMAS DIGITALES

Para desarrollar un sistema digital se debe primero, entender claramente el problema, ello va a permitir deducir la tabla de estado que define el problema. Una vez que se deduce la tabla, el desarrollo es un simple proceso metódico, el cual nos permite determinar la función lógica, simplificar y luego implementar el circuito lógico.

Como vimos, la función lógica se obtiene como una suma de términos; para cada estado, donde la variable de salida es igual a 1. Cada término es un producto de las variables de entrada, pero irán negadas en el caso de que en dicho estado la variable tenga el valor de 0.

Luego se debe hacer el proceso de simplificación de la función lógica y para ello se recurre a las propiedades vistas anteriormente. Finalmente, se hará la representación de la función simplificada a través del circuito lógico, donde cada producto es reemplazado por compuertas AND, cada suma es representada por las compuertas OR y para la inversión o negación usamos al inversor o compuerta NOT.

Ejemplo de un sistema digital:

Sea el siguiente sistema de 3 variables de entrada A, B y C, y una variable de salida Z. Z es igual a 1 cuando el número de estado sea mayor que 2.

Se desarrolla la tabla de estado colocando un 1 en cada estado donde tenga un número mayor que 2, hay que tener en cuenta que cada estado tiene un número que lo identifica, tal como se puede apreciar en el siguiente gráfico.

Tabla de Estado

A	B	C	Z	
0	0	0	0	m0
0	0	1	0	m1
0	1	0	0	m2
0	1	1	1	m3
1	0	0	1	m4
1	0	1	1	m5
1	1	0	1	m6
1	1	1	1	m7

Función lógica:

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

$$0 \ 1 \ 1 \quad 1 \ 0 \ 0 \quad 1 \ 0 \ 1 \quad 1 \ 1 \ 0 \quad 1 \ 1 \ 1$$

Simplificación:

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

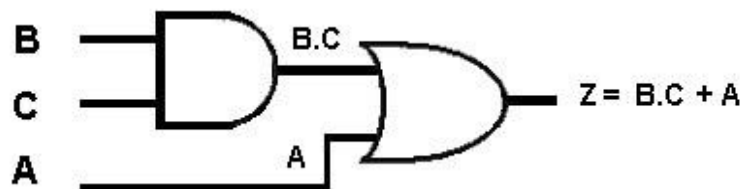
$$Z = \bar{A}.B.C + A.\bar{B} + A.B$$

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

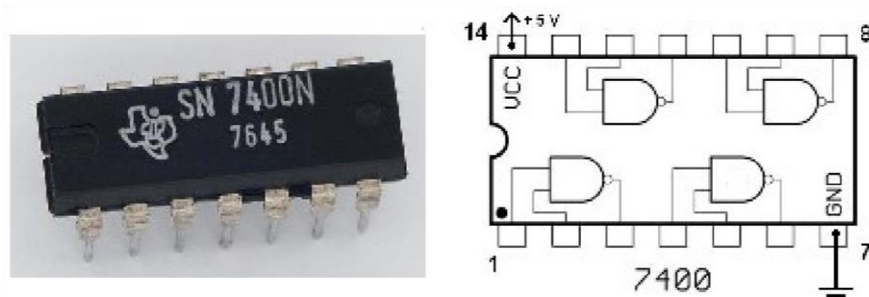
$$Z = \bar{A}.B.C + A$$

$$Z = B.C + A$$

En el gráfico anterior se puede apreciar además de la tabla de estado, la función lógica y la simplificación, donde se ha buscado algunas expresiones que permitan simplificar. Para ello debemos tener presente las propiedades de álgebra de Boole. Nos falta solo el circuito lógico, el cual vemos que es sencillo, se trata solo de dos compuertas una AND para la multiplicación y un OR para la suma.

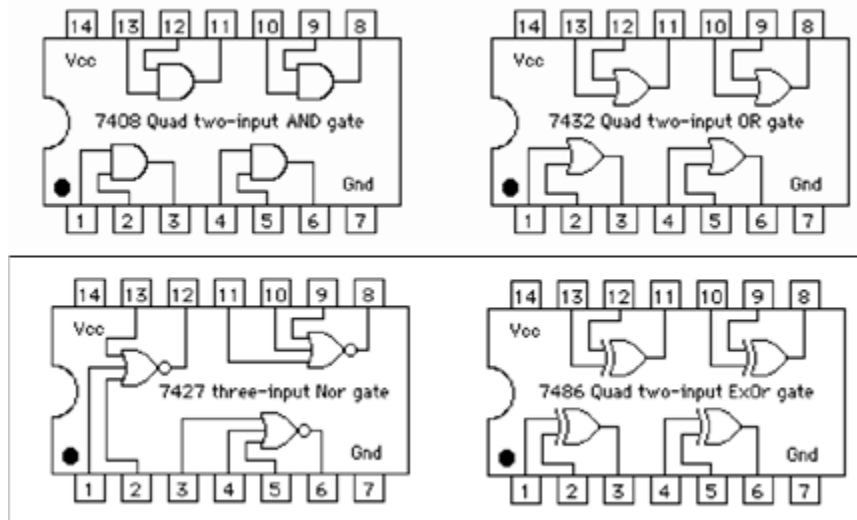


Si queremos implementar físicamente estos sistemas digitales, es necesario para ello las compuertas lógicas, las cuales vienen en chips con diferentes tipos de compuertas. Para tener una idea de ello veamos algunos ejemplos:



En el gráfico apreciamos el circuito integrado (chip) 7400, el cual tiene cuatro compuertas NAND de dos entradas, el chip tiene 14 pines, donde dos de ellos se usan para recibir la energía (5 voltios), el pin 7 para 0 voltios y el 14 para 5 voltios.

Otros ejemplos son los siguientes, donde vemos a 4 compuertas AND de dos entradas y el otro es de 4 compuertas OR e dos entradas.



En el gráfico anterior vemos un chip que tiene tres compuertas NOR de tres entradas, con lo que queremos indicar que hay diferentes tipos de compuertas. Finalmente mostramos a cuatro compuertas OR Exclusivas

Si analizamos a la OR Exclusive como un sistema, podemos encontrar que la función lógica determinada a partir de su tabla, no tiene posibilidad de simplificarla usando las propiedades del álgebra de Boole, esto es solo posible mediante una compuerta especial como lo es la OR Exclusive, a esta forma de simplificar ha obligado a la creación de compuertas OR Exclusive y OR Exclusive Negado.

Tabla de estado			
A	B	Z	
0	0	0	m0
0	1	1	m1
1	0	1	m2
1	1	0	m3

Función Lógica Z

$$Z = \bar{A}.B + A.\bar{B}$$

0 1 1 0

$$Z = \bar{A}.B + A.\bar{B} = A \oplus B \quad \text{Ex OR}$$

$$\bar{Z} = \bar{A}.\bar{B} + A.B = \overline{A \oplus B} \quad \text{Ex NOR}$$

1.1. SIMPLIFICACIÓN POR MAPAS DE KARNAUGH

La complejidad de los sistemas digitales que es implementado a partir de una función lógica, directamente relacionado con la complejidad de la expresión algebraica a partir de la cual la función se implementa. La representación de la tabla de verdad de una función es única pero la función puede aparecer en muchas formas diferentes como se expresa algebraicamente. La expresión puede simplificarse utilizando las relaciones básicas del álgebra Booleana. Este procedimiento sin embargo, es algunas veces difícil porque carece de reglas específicas para predecir cada uno de los pasos sucesivos en el proceso de manipulación. El método del mapa de Karnaugh proporciona un procedimiento simple, y directo para simplificar funciones Booleanas.

Este método puede mirarse como un arreglo gráfico de una tabla de verdad que permite una interpretación fácil para elegir el mínimo número de variables que se necesitan para expresar la función algebraicamente. Este método se basa en una nueva tabla, con la intención de formar grupos de unos y obtener la función lógica simplificada como una suma de grupos de "unos", los grupos de unos deben ser en cantidades como: 1, 2, 4, 8, 16, etc. Los unos para formar los grupos deben estar en horizontal o en vertical, nunca se debe agrupar en diagonal.

El mapa de karnaugh es un diagrama hecho de cuadrados, en el que, cada cuadrado representa un estado y en ese cuadrado debe ir el valor que toma la variable de salida. Por lo tanto en los cuadrados encontraremos o "1" o "0", dependiendo de la tabla. Para no equivocarse, conviene tener identificado cada cuadrado con un número, con el número de estado, de esta forma pasar de la tabla de estado a la tabla de Karnaugh sea sencillo.



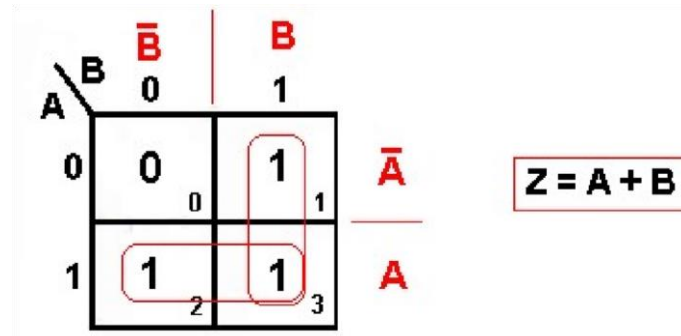
Los mapas de Karnaugh pueden usarse para los sistemas de dos, tres y más variables, nosotros trataremos solo de dos y de tres variables, con el objeto de conocer cómo se manejan y que importancia han tenido en el desarrollo de los sistemas digitales. Tengamos presente que en este curso no esperamos ser expertos en simplificación, sino que tocamos el tema a nivel introductorio para conocer los sistemas digitales, para conocer a la computadora como un sistema digital, saber que la computadora está hecha con compuertas y precisar que es un sistema muy complejo que recibe ceros y unos y entrega también ceros y unos.

1.1.1. Mapa de karnaugh de dos variables.

En el siguiente gráfico puede darse cuenta que cada estado de la tabla de verdad tiene un número que lo identifica y tiene su correspondiente cuadrado en el mapa. Como los estados son cuatro, los números de estados son: 0, 1, 2 y 3, y que en la tabla hay cuatro cuadrados identificados con: 0, 1, 2 y 3

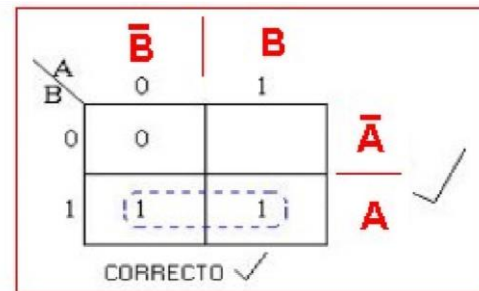
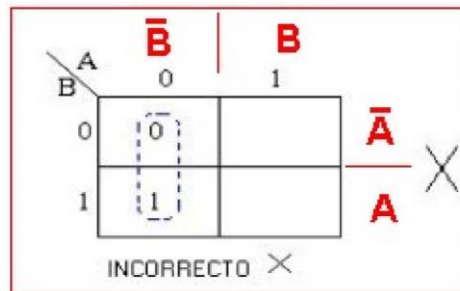
Tabla de Estado				Mapa de Karnaugh	
A	B	Z			
0	0	0	0	A 0	0
0	1	1	1		1
1	0	1	2	1	2
1	1	1	3		3

El mapa de Karnaugh está dividido de tal manera que se forman grupos en horizontal y vertical, en el ejemplo anterior tenemos dos grupos, uno de ellos corresponde a los cuadrados 1 y 3 y el otro grupo está formado por los cuadrados 2 y 3.

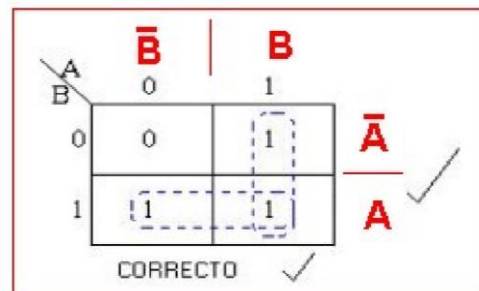
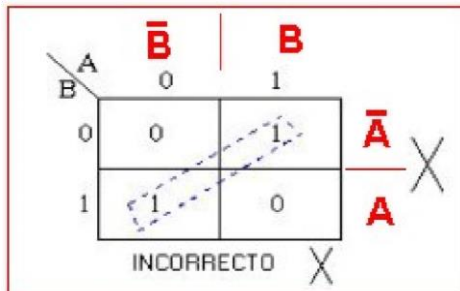


Veamos algunas reglas para el uso de los mapas de Karnaugh:

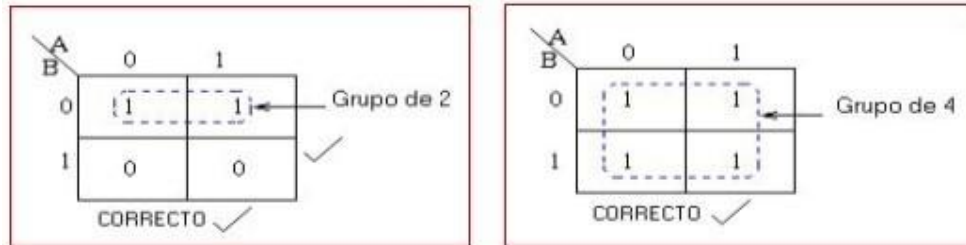
- Las agrupaciones son exclusivamente de unos, esto implica que ningún grupo puede contener un cero.



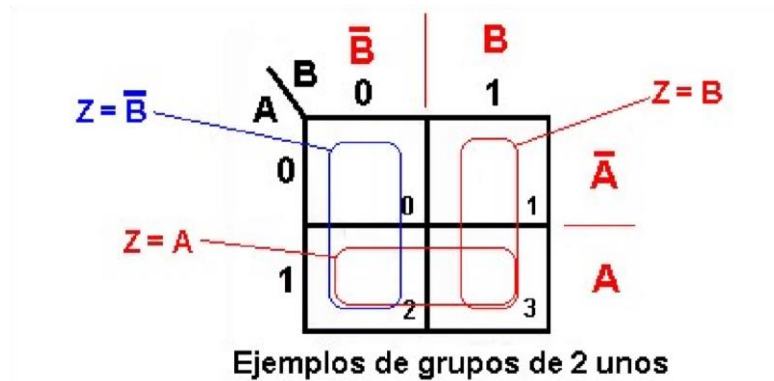
- Los grupos pueden formarse solo en horizontal o en vertical, esto implica que las diagonales están prohibidas.



- Los grupos deben contener 2^n elementos, es decir cada grupo tendrá 1, 2, 4, 8... unos.



- Veamos algunos ejemplos:



1.1.2. Mapa de Karnaugh de tres variables.

En este caso se debe pasar de la tabla de estado a la tabla o mapa de Karnaugh, el cual tiene 8 cuadrados, cada uno de ellos también los hemos identificado con el número de estado, para pasar fácilmente de una tabla a la otra, veamos las dos tablas a través del siguiente gráfico:

Tabla de Estado

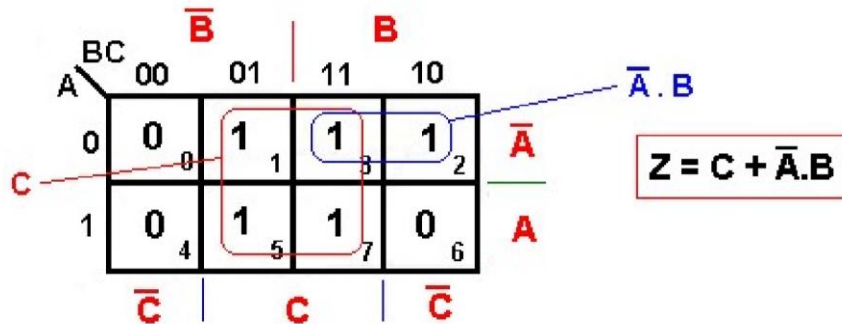
A	B	C	Z	
0	0	0	0	0
0	0	1	1	1
0	1	0	1	2
0	1	1	1	3
1	0	0	0	4
1	0	1	1	5
1	1	0	0	6
1	1	1	1	7

Mapa de Karnaugh

A \ BC	00	01	11	10
0	0 ₀	1 ₁	1 ₃	1 ₂
1	0 ₄	1 ₅	1 ₇	0 ₆

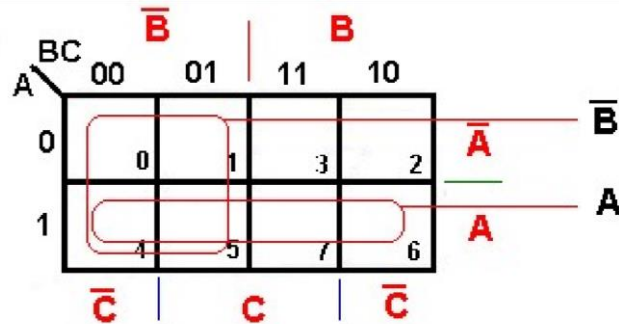
Una vez que se ha pasado de la tabla de estado a la tabla de Karnaugh, debemos agrupar los unos, se debe agrupar la mayor cantidad de unos, siguiendo la secuencia: 1, 2, 4, 8; finalmente, la salida Z se obtiene como la suma de grupos, tal como se puede ver en la siguiente figura:

Mapa de Karnaugh

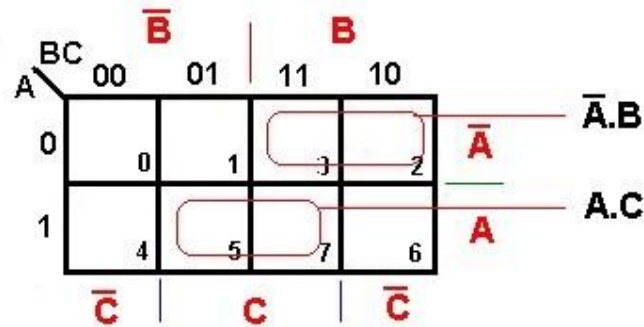


En el mapa de Karnaugh de tres variables, se ha distribuido los cuadrados para hacer más fácil la simplificación, de esta forma el grupo de 4 unos de la parte superior es igual a $\bar{A}$ y los cuatro cuadrados inferiores es igual a la variable A, de manera similar ocurre con las otras variables, tal como se puede apreciar en el siguiente gráfico, donde hemos puesto algunos ejemplos:

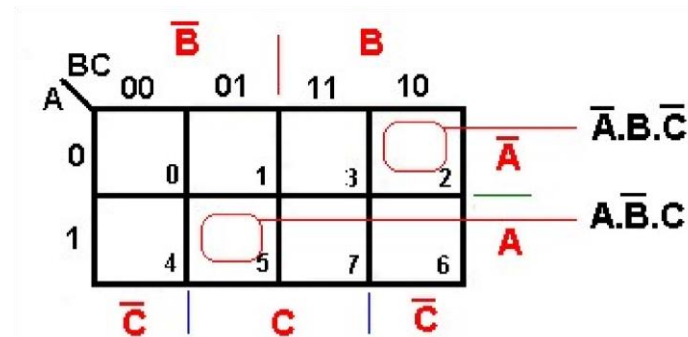
- Primero, grupos de cuatro "unos", a los que les corresponde una letra.



- Segundo, grupos de dos "unos", a los que les corresponde dos letras.

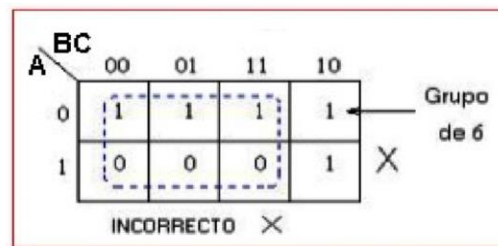
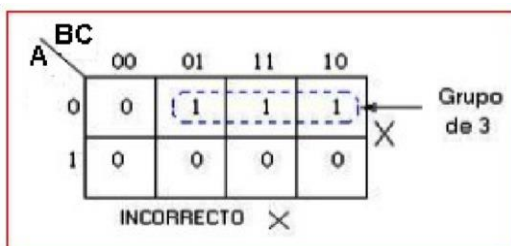


- Tercero, grupos de un "uno", a los que les corresponde tres letras.



Agreguemos algunas reglas para los mapas de Karnaugh.

- Los grupos deben contener 2^n elementos, es decir cada grupo tendrá 1, 2, 4, 8... unos.



- Cada grupo ha de ser tan grande como sea posible, tal como se ve en el ejemplo, donde cabe destacar que a pesar de que no se ha incumplido ninguna regla, el resultado no está simplificado correctamente.

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

CORRECTO ✓

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

INCORRECTO ✗

- Todos los unos tienen que pertenecer como mínimo a un grupo, aunque puedan pertenecer a más de uno.

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

Grupo 1: El 1 se encuentra en al menos un grupo

Grupo 2:

- Puede existir solapamiento de grupos.

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

Los grupos se solapan

CORRECTO ✓

- La formación de grupos también se puede producir con las celdas extremas de la tabla.

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

- Tenga en cuenta que la tabla debe ser considerada como un cilindro, tal como aparece en la siguiente figura.

