



Electrónica



Diseño lógico

Fundamentos en electrónica digital

Héctor Arturo **Flórez** Fernández

de la
ediciones 
conocimiento a su alcance
www.edicionesdelau.com

Capítulo 2

Compuertas lógicas

Las compuertas lógicas son elementos de electrónica digital que permiten realizar operaciones lógicas entre cantidades binarias. El álgebra de Boole es el área matemática específica que estudia las operaciones lógicas y sus propiedades.

2.1 COMPUERTA NOT

La compuerta NOT, también conocida como negador o inversor, es una compuerta que permite realizar la operación lógica NOT. Esta operación indica que dada una entrada binaria, su salida será el valor contrario. Es decir, si a la compuerta NOT entra un cero lógico, su salida será un uno lógico y si entra un uno lógico, su salida será un cero lógico.

Cuando se habla de operaciones lógicas, se habla de niveles de verdad. Un cero lógico equivale a falso y un uno lógico equivale a verdadero.

Entonces los resultados de un circuito lógico, están determinados por un concepto denominado "Tabla de verdad", la cual es una tabla que ofrece un valor lógico de salida para una combinación de entrada.

La tabla de verdad de la operación lógica NOT es:

Tabla 2.1 Tabla de verdad de la compuerta NOT

Entrada	Salida
0	1
1	0

La compuerta NOT también cuenta con un símbolo que la hace distinguir de otras compuertas. El símbolo se representa en figura 2.1:



Figura 2.1 Símbolo de la compuerta NOT

Además, en el álgebra de Boole, la compuerta NOT también tiene su expresión. Una entrada o variable, debe especificarse por una letra, igualmente una salida. Entonces si la entrada de la compuerta NOT se denomina A y la salida X, la expresión Booleana de la operación lógica NOT es:

$$X = \bar{A}$$

El funcionamiento de la compuerta se puede observar en la siguiente figura:

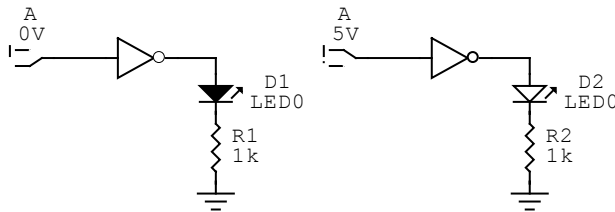


Figura 2.2 Funcionamiento de la compuerta NOT

En la gráfica anterior se puede apreciar que en la figura de la izquierda hay a la entrada un cero lógico por consiguiente a la salida se enciende el LED indicando un uno lógico. En la figura de la derecha hay a la entrada un uno lógico por consiguiente a la salida se apaga el LED indicando un cero lógico.

La referencia de esta compuerta es el número 74LS04, el cual tiene la estructura interna de la figura 2.3.

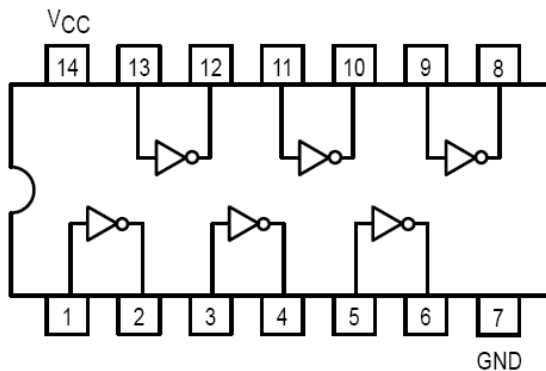


Figura 2.3 74LS04 Compuerta NOT

El circuito 74LS04, tiene internamente 6 compuertas NOT, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 Voltios denominado Vcc

2.2 COMPUERTA AND

La compuerta AND, es una compuerta que permite realizar la operación lógica AND. La operación lógica AND tiene siempre como mínimo dos entradas. La tabla de verdad de la operación lógica AND de dos entradas es:

Tabla 2.2 Tabla de verdad de la compuerta AND de dos entradas

Entradas		Salida
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

La tabla de verdad de la operación lógica de tres entradas es:

Tabla 2.3 Tabla de verdad de la compuerta AND de tres entradas

Entradas			Salida
A	B	C	X
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

Con base en las tablas anteriores se puede apreciar una característica fundamental de la operación lógica AND. La salida será uno lógico si y solo si todas sus entradas son uno lógico.

El símbolo de la compuerta AND es:



Figura 2.4 Símbolo de la compuerta AND

En el álgebra de Boole, la expresión de la compuerta AND es:

$$X = A.B$$

El funcionamiento de la compuerta se puede observar en la siguiente figura:

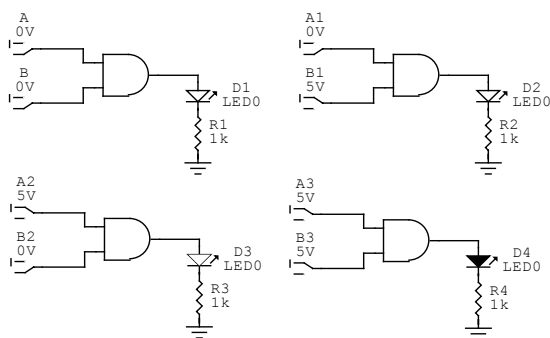


Figura 2.5 Funcionamiento de la compuerta AND

En la figura 2.5 anterior se puede apreciar que en la parte inferior derecha hay en las entradas un uno lógico por consiguiente a la salida se enciende el LED indicando un uno lógico. En el resto de las imágenes la salida es cero lógico, lo cual describe la tabla de verdad de esta operación lógica.

La referencia de la compuerta AND de dos entradas es el número 74LS08, el cual tiene la siguiente estructura interna.

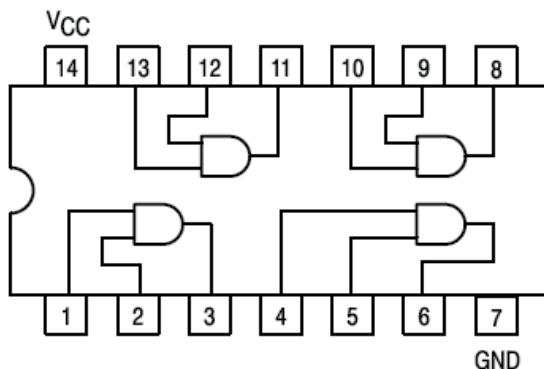


Figura 2.6 74LS08 Compuerta AND de dos entradas

El circuito 74LS08, tiene internamente 4 compuertas AND de dos entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 voltios denominado Vcc.

La referencia de la compuerta AND de tres entradas es el número 74LS11, el cual tiene la siguiente estructura interna.

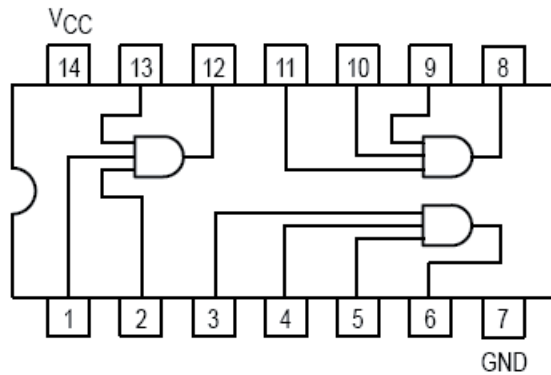


Figura 2.7 74LS11 Compuerta AND de tres entradas

El circuito 74LS11, tiene internamente 3 compuertas AND de tres entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 voltios denominado Vcc.

2.3 COMPUERTA OR

La compuerta OR es una compuerta que permite realizar la operación lógica OR. La operación lógica OR tiene siempre como mínimo dos entradas. La tabla de verdad de la operación lógica OR de dos entradas es:

Tabla 2.4 Tabla de verdad de la compuerta OR de dos entradas

Entradas		Salida
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

La tabla de verdad de la operación lógica de tres entradas es:

Tabla 2.5 Tabla de verdad de la compuerta OR de tres entradas

Entradas			Salida
A	B	C	X
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

Con base en las tablas anteriores se puede apreciar una característica fundamental de la operación lógica OR. La salida será cero lógico sí y sólo sí todas sus entradas son cero lógico.

El símbolo de la compuerta OR se representa en la figura 2.8:



Figura 2.8 Símbolo de la compuerta OR

En el álgebra de Boole, la expresión de la compuerta OR es:

$$X = A + B$$

El funcionamiento de la compuerta se puede observar en la siguiente figura:

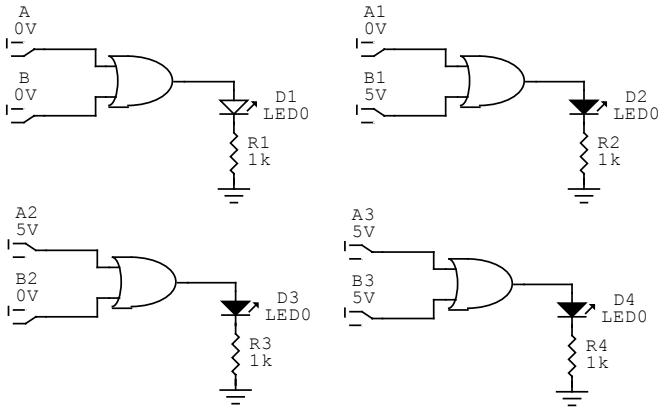


Figura 2.9 Funcionamiento de la compuerta OR

En la figura anterior se puede apreciar que en la parte superior izquierda hay en las entradas zeros lógicos por consiguiente a la salida se apaga el LED indicando un cero lógico. En el resto de las figuras la salida es uno lógico, lo cual describe la tabla de verdad de esta operación lógica.

La referencia de la compuerta OR de dos entradas es el número 74LS32, el cual tiene la siguiente estructura interna.

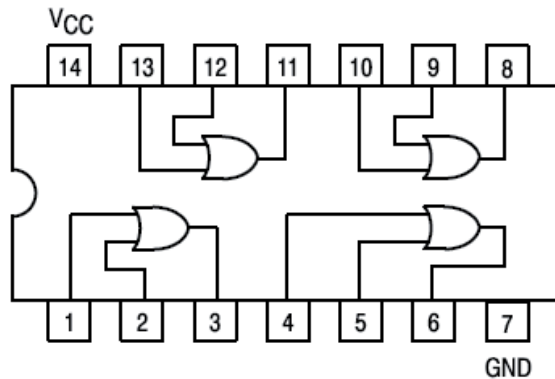


Figura 2.10 74LS32 Compuerta OR de dos entradas

El circuito 74LS32, tiene internamente 4 compuertas OR de dos entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 voltios denominado Vcc.

2.4. COMPUERTA NAND

La compuerta NAND, es una compuerta que permite realizar la operación lógica NAND. La operación lógica NAND tiene siempre como mínimo dos entradas. La tabla de verdad de la operación lógica NAND de dos entradas es:

Tabla 2.6 Tabla de verdad de la compuerta NAND de dos entradas

Entradas		Salida
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

La tabla de verdad de la operación lógica de tres entradas es:

Tabla 2.7 Tabla de verdad de la compuerta NAND de tres entradas

Entradas			Salida
A	B	C	X
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	0

Con base en las tablas anteriores se puede apreciar una característica fundamental de la operación lógica NAND. La salida será cero lógico sí y solo sí todas sus entradas son uno lógico.

El símbolo de la compuerta NAND es:

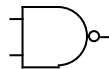


Figura 2.11 Símbolo de la compuerta NAND

En el álgebra de Boole, la expresión de la compuerta AND es:

$$X = \overline{A.B}$$

El funcionamiento de la compuerta se puede observar en la siguiente figura:

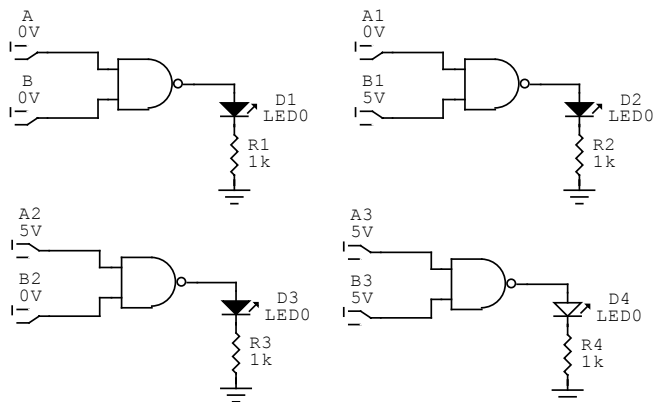


Figura 2.12 Funcionamiento de la compuerta NAND

Se puede apreciar que en la representación de la derecha inferior hay en las entradas uno lógicos por consiguiente a la salida se apaga el LED indicando un cero lógico. En el resto de las figuras la salida es uno lógico, lo cual describe la tabla de verdad de esta operación lógica.

La referencia de la compuerta NAND de dos entradas es el número 74LS00, el cual tiene la siguiente estructura interna.

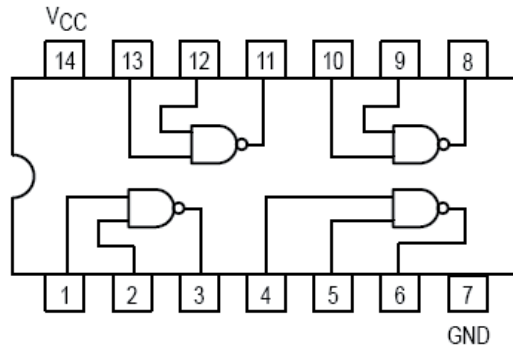


Figura 2.13 74LS00 Compuerta NAND de dos entradas

El circuito 74LS00, tiene internamente 4 compuertas NAND de dos entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 voltios denominado Vcc.

La referencia de la compuerta NAND de tres entradas es el número 74LS10, el cual tiene la siguiente estructura interna.

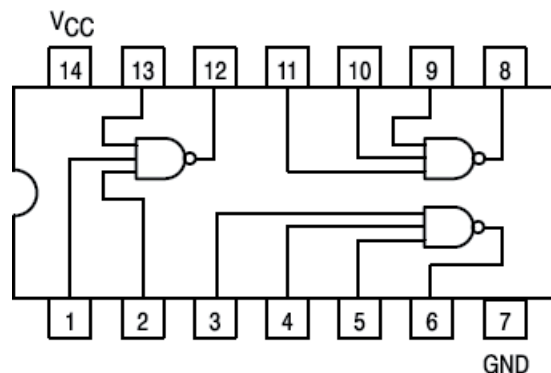


Figura 2.14 74LS10 Compuerta NAND de tres entradas

El circuito 74LS10, tiene internamente 3 compuertas NAND de tres entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 Voltios denominado Vcc.

2.5 COMPUERTA NOR

La compuerta NOR, es una compuerta que permite realizar la operación lógica NOR. La operación lógica NOR tiene siempre como mínimo dos entradas. La tabla de verdad de la operación lógica NOR de dos entradas es:

Tabla 2.8 Tabla de verdad de la compuerta NOR de dos entradas

Entradas		Salida
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

La tabla de verdad de la operación lógica de tres entradas es:

Tabla 2.9 Tabla de verdad de la compuerta NOR de tres entradas

Entradas			Salida
A	B	C	X
0	0	0	1
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	0

Con base en las tablas anteriores se puede apreciar una característica fundamental de la operación lógica NOR. La salida será uno lógico si y solo si todas sus entradas son cero lógico.

El símbolo de la compuerta NOR es:

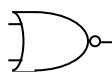


Figura 2.15 Símbolo de la compuerta NOR

En el álgebra de Boole, la expresión de la compuerta NOR es:

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

El funcionamiento de la compuerta se puede observar en la siguiente figura:

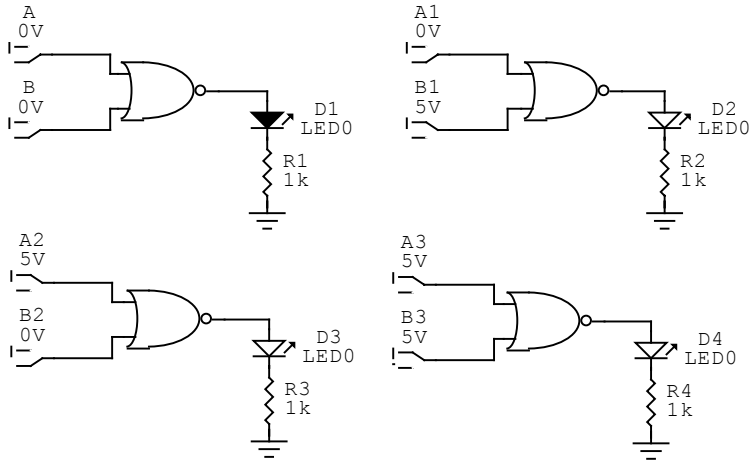


Figura 2.16 Funcionamiento de la compuerta NOR

En la gráfica anterior se puede apreciar que en la figura de la izquierda arriba hay en las entradas ceros lógicos por consiguiente a la salida se enciende el LED indicando un uno lógico. En el resto de las figuras la salida es cero lógico, lo cual describe la tabla de verdad de esta operación lógica.

La referencia de la compuerta NOR de dos entradas es el número 74LS02, el cual tiene la siguiente estructura interna.

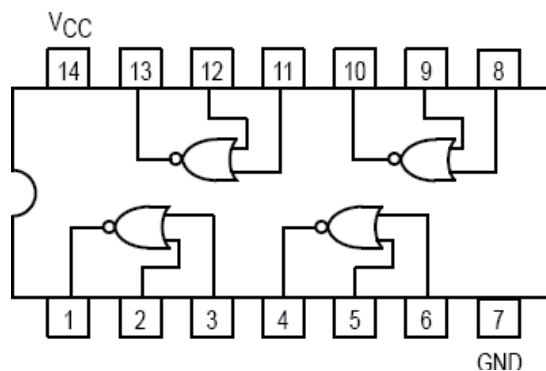


Figura 2.17 74LS02 Compuerta OR de dos entradas

El circuito 74LS02, tiene internamente 4 compuertas NOR de dos entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 voltios denominado Vcc.

La referencia de la compuerta NOR de tres entradas es el número 74LS27, el cual tiene la siguiente estructura interna.

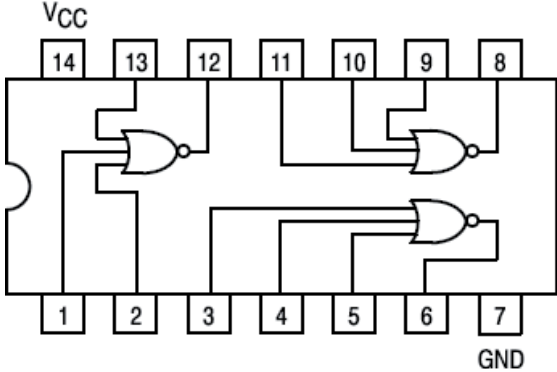


Figura 2.18 74LS27 Compuerta NOR de tres entradas

El circuito 74LS27, tiene internamente 3 compuertas NOR de tres entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 Voltios denominado Vcc.

2.6 COMPUERTA OR EXCLUSIVA

La compuerta OR Exclusiva o también conocida como XOR, es una compuerta que permite realizar la operación lógica XOR. La operación lógica XOR tiene siempre como mínimo dos entradas. La tabla de verdad de la operación lógica XOR de dos entradas es:

Tabla 2.10 Tabla de verdad de la compuerta XOR de dos entradas

Entradas		Salida
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

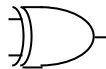
La tabla de verdad de la operación lógica de tres entradas es:

Tabla 2.11 Tabla de verdad de la compuerta XOR de tres entradas

Entradas			Salida
A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Con base en las tablas anteriores se puede apreciar una característica fundamental de la operación lógica XOR. La salida será uno lógico sí y solo sí la suma de unos lógicos a la entrada es impar.

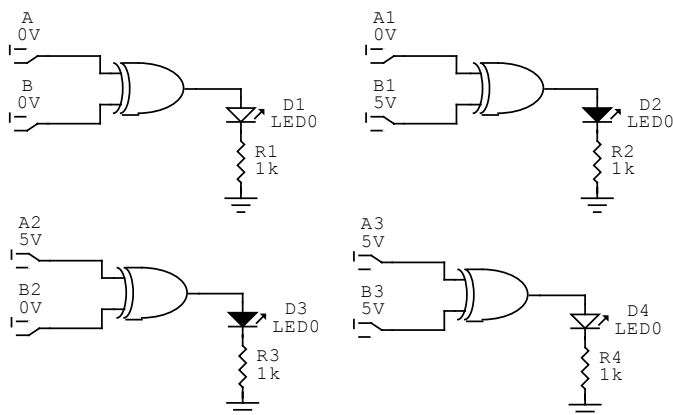
El símbolo de la compuerta XOR es:

**Figura 2.19** Símbolo de la compuerta XOR

En el álgebra de Boole, la expresión de la compuerta XOR es:

$$X = A \oplus B$$

El funcionamiento de la compuerta se puede observar en la siguiente figura:

**Figura 2.20** Funcionamiento de la compuerta XOR

En la gráfica anterior se puede apreciar que en la primera y última figura hay en las entradas número par de unos lógicos por consiguiente a la salida se apaga el LED indicando un cero lógico. En el resto de las figuras la salida es uno lógico, lo cual describe la tabla de verdad de esta operación lógica.

La referencia de la compuerta XOR de dos entradas es el número 74LS86, el cual tiene la siguiente estructura interna.

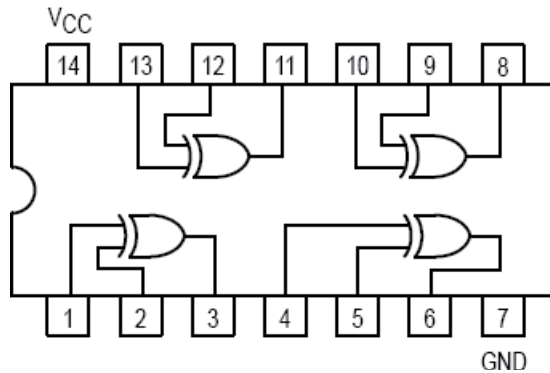


Figura 2.21 74LS86 Compuerta XOR de dos entradas

El circuito 74LS86, tiene internamente 4 compuertas XOR de dos entradas, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 voltios denominado Vcc.

2.7 COMPUERTA NOR EXCLUSIVA

La compuerta NOR Exclusiva o también conocida como XNOR, es una compuerta que permite realizar la operación lógica XNOR. La operación lógica XNOR tiene siempre como mínimo dos entradas. La tabla de verdad de la operación lógica XNOR de dos entradas es:

Tabla 2.12 Tabla de verdad de la compuerta XNOR de dos entradas

Entradas		Salida
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

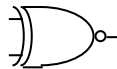
La tabla de verdad de la operación lógica de tres entradas es:

Tabla 2.13 Tabla de verdad de la compuerta XNOR de tres entradas

Entradas			Salida
A	B	C	X
0	0	0	1
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

Con base en las tablas anteriores se puede apreciar una característica fundamental de la operación lógica XNOR. La salida será cero lógico sí y solo si la suma de unos lógicos a la entrada es impar.

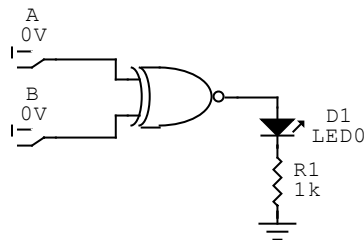
El símbolo de la compuerta XNOR es:

**Figura 2.22** Símbolo de la compuerta XNOR

En el álgebra de Boole, la expresión de la compuerta XNOR es:

$$X = \overline{A \oplus B}$$

La implementación de la compuerta XNOR se describe por la siguiente figura:

**Figura 2.23** Implementación de la compuerta XNOR

El funcionamiento de la compuerta se puede observar en la siguiente figura:

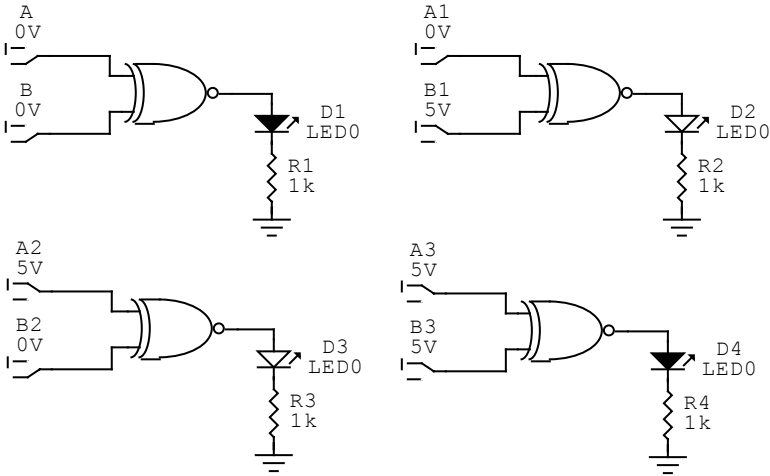


Figura 2.24 Funcionamiento de la compuerta XNOR

En la gráfica anterior se puede apreciar que en la primera y última figura hay en las entradas número par de unos lógicos por consiguiente a la salida se enciende el LED indicando un uno lógico. En el resto de las figuras la salida es cero lógico, lo cual describe la tabla de verdad de esta operación lógica.

La referencia de la compuerta XNOR de dos entradas es el número 74LS266, el cual tiene la siguiente estructura interna.

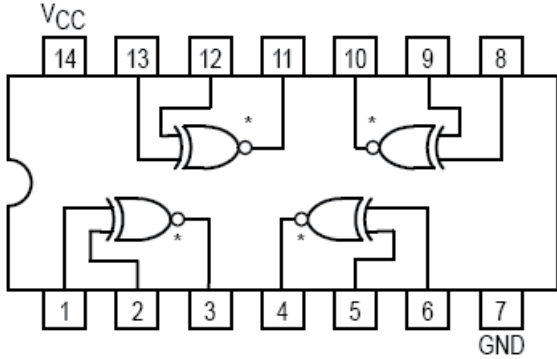


Figura 2.25 74LS266 Compuerta XNOR de dos entradas

El circuito 74LS266, tiene internamente 4 compuertas XNOR de dos entradas de colector abierto, además de un pin que se debe conectar a tierra denominado GND y un pin que se debe conectar a 5 voltios denominado Vcc. La salida de colector abierto indica que la salida debe conectarse a Vcc por medio de una resistencia para que el circuito ejerza el correcto funcionamiento.

2.8. IMPLEMENTACIÓN MEDIANTE COMPUERTAS LÓGICAS

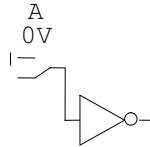
Una expresión lógica puede ser implementada a través de compuertas teniendo en cuenta la prioridad de las operaciones lógicas. La prioridad más alta la determina la operación lógica NOT, es decir que es ésta la primera que se debe implementar. En segundo lugar se encuentra la operación lógica AND y XOR y en último lugar la operación lógica OR, lo que indica que ésta es la última que se debe implementar.

Considerando la siguiente expresión lógica:

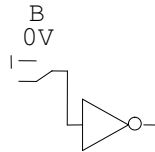
$$X = \overline{A}BC + \overline{ABC} + A \oplus \overline{B}$$

La implementación debe realizarse a través de los siguientes pasos:

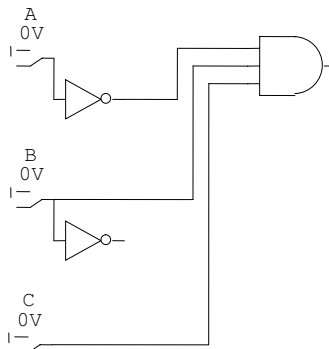
1. Se aplica la operación NOT a la variable A debido a que lo requiere el primer término.



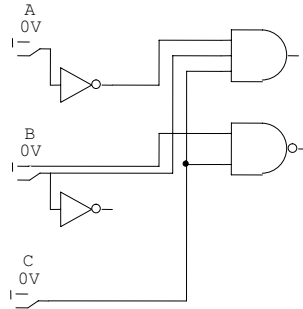
2. Se aplica la operación NOT a la variable B debido a que lo requiere el tercer término.



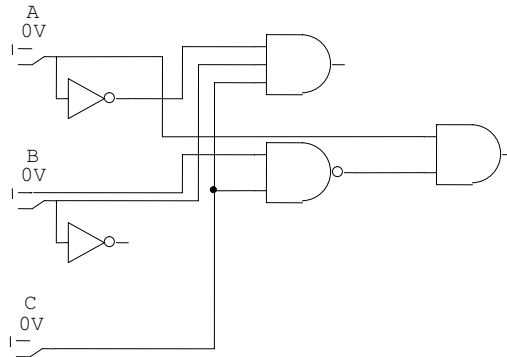
3. Se aplica la operación AND de tres entradas para el primer término, tomando la salida de la operación NOT de la variable A expuesta en el numeral 1, la variable B y la variable C.



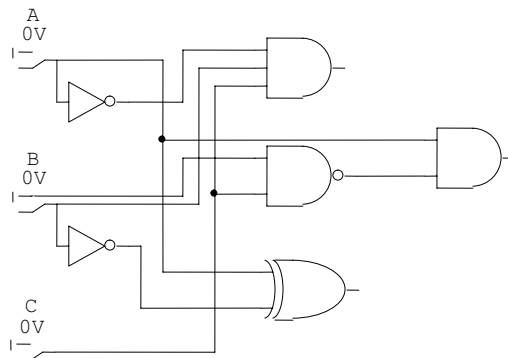
4. Se aplica la operación NAND entre la variable B y C, para usar en el segundo término.



5. Se aplica la operación AND de dos entradas para el segundo término, tomando la salida de la operación NAND expuesta en el numeral 4 y la variable A.



6. Se aplica la operación XOR para el tercer término, tomando la salida de la operación NOT de la variable B expuesta en el numeral 2 y la variable A.



7. Se aplica la operación lógica OR de tres entradas tomando la salida obtenida en la implementación del numeral 3, numeral 5 y numeral 6. Debido a que en la tecnología TTL no existe la compuerta OR de tres entradas, es necesario construirla con base en dos compuertas OR de dos entradas.

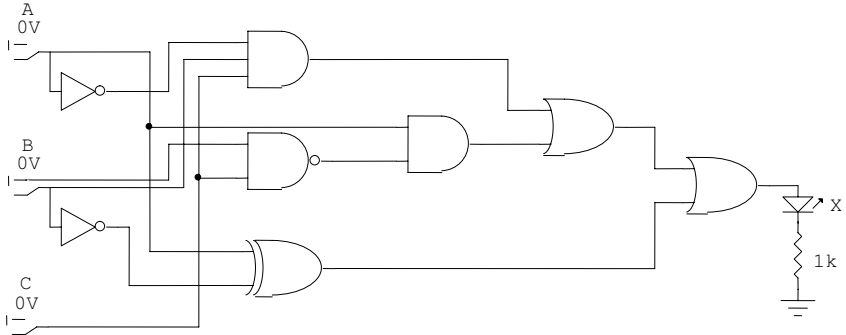


Figura 2.26 Implementación mediante compuertas lógicas



Ejercicios Propuestos

1. Implementar la expresión Booleana $X = \overline{ABC} + \overline{A}BC + \overline{A}\overline{B}C$
2. Implementar la expresión Booleana $X = (\overline{ABC} \oplus \overline{A}BC) + \overline{(\overline{A}B \oplus AC)}$
3. Implementar la expresión Booleana $X = (A \oplus B)C + AB$
4. Implementar la expresión Booleana $X = ABC + \overline{ABC}$

Capítulo 3

Álgebra de Boole

En 1854 George Boole publicó una obra titulada *Investigación de las leyes del pensamiento, sobre las que se basan las teorías matemáticas de la lógica y la probabilidad*. En esta publicación formuló la idea del álgebra de operaciones lógicas que se conoce hoy en día como Álgebra de Boole. El Álgebra de Boole es una forma que permite expresar y analizar operaciones de circuitos lógicos.

3.1 OPERACIONES BOOLEANAS

En el Álgebra de Boole, una variable es un símbolo que se utiliza para representar magnitudes lógicas. Cualquier variable puede tener un valor de 1 o 0. El complemento es el inverso de la variable. Un literal se define como una variable o su complemento.

3.1.1 Adición Booleana

La adición Booleana es equivalente a la operación lógica OR. En el álgebra de Boole, un término suma es una suma de literales. En circuitos lógicos, un término suma se produce mediante la operación lógica OR de dos literales. El resultado de la suma es 1 cuando uno o más de los literales es uno.

Las reglas de la suma son:

Tabla 3.1 Reglas de la suma Booleana

Entradas		Salida
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

3.1.2 Multiplicación Booleana

La multiplicación Booleana es equivalente a la operación lógica AND. En álgebra de Boole, un término producto es un producto de literales. En circuitos lógicos, un término producto se produce mediante la operación lógica AND de dos literales. El resultado de la multiplicación es 1 cuando todos sus literales son uno.

Las reglas de la multiplicación son:

Tabla 3.2 Reglas de la multiplicación Booleana

Entradas		Salida
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

3.2 LEYES DEL ÁLGEBRA DE BOOLE

Al igual que en otras áreas de las matemáticas, existe en el álgebra de Boole una serie de leyes que resuelven determinados problemas en las implementaciones de circuitos lógicos.

3.2.1 Ley Conmutativa

La ley conmutativa indica que el orden de las variables no altera el resultado. Entonces la ley conmutativa para la adición de dos variables se describe como:

$$A + B = B + A$$

La implementación de la sentencia anterior equivale a:

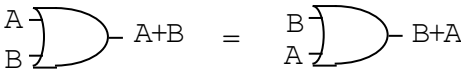


Figura 3.1 Ley conmutativa de la adición Booleana

La ley conmutativa para la multiplicación de dos variables se describe como:

$$A.B = B.A$$

La implementación de la sentencia anterior equivale a:

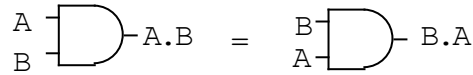


Figura 3.2 Ley conmutativa de la multiplicación Booleana

3.2.2 Ley Asociativa

La ley asociativa para la adición de tres variables se describe como:

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

La implementación de la sentencia anterior equivale a:

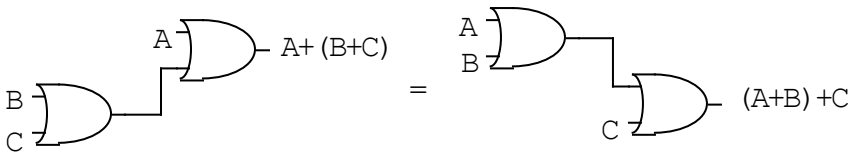


Figura 3.3 Ley asociativa de la adición Booleana

La ley asociativa para la multiplicación de dos variables se describe como:

$$A(B.C) = (A.B)C$$

La implementación de la sentencia anterior equivale a:

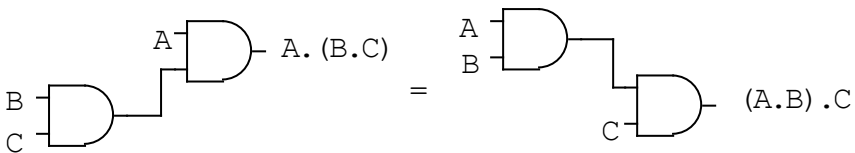


Figura 3.4 Ley asociativa de la multiplicación Booleana

3.2.3 Ley Distributiva

La ley distributiva para tres variables se describe como:

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

La implementación de la sentencia anterior equivale a:

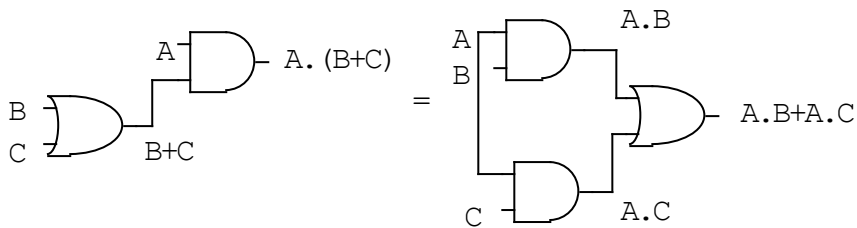


Figura 3.5 Ley distributiva Booleana

3.3 REGLAS DEL ÁLGEBRA DE BOOLE

Existen unas reglas básicas para la manipulación y simplificación de expresiones Booleanas. Estas reglas se describen en la siguiente tabla.

Tabla 3.3 Reglas del álgebra de Boole

Nº	REGLA	DESCRIPCIÓN
1	$A + 0 = A$	Al aplicar la operación OR a una variable y a cero el resultado es el valor de la variable. $A = 0 \Rightarrow 0 + 0 = 0$ $A = 1 \Rightarrow 1 + 0 = 1$
2	$A + 1 = 1$	Al aplicar la operación OR a una variable y a uno el resultado es uno. $A = 0 \Rightarrow 0 + 1 = 1$ $A = 1 \Rightarrow 1 + 1 = 1$
3	$A \cdot 0 = 0$	Al aplicar la operación AND a una variable y a cero el resultado es cero. $A = 0 \Rightarrow 0 \cdot 0 = 0$ $A = 1 \Rightarrow 1 \cdot 0 = 0$
4	$A \cdot 1 = A$	Al aplicar la operación AND a una variable y a uno el resultado es el valor de la variable. $A = 0 \Rightarrow 0 \cdot 1 = 0$ $A = 1 \Rightarrow 1 \cdot 1 = 1$
5	$A + A = A$	Al aplicar la operación OR a una variable consigo misma el resultado es el valor de la variable. $A = 0 \Rightarrow 0 + 0 = 0$ $A = 1 \Rightarrow 1 + 1 = 1$

6	$A + \bar{A} = 1$	Al aplicar la operación OR a una variable con su complemento el resultado es 1. $A = 0 \Rightarrow 0 + 1 = 1$ $A = 1 \Rightarrow 1 + 0 = 1$
7	$A.A = A$	Al aplicar la operación AND a una variable consigo misma el resultado es el valor de la variable. $A = 0 \Rightarrow 0.0 = 0$ $A = 1 \Rightarrow 1.1 = 1$
8	$A.\bar{A} = 0$	Al aplicar la operación AND a una variable con su complemento el resultado es 0. $A = 0 \Rightarrow 0.1 = 0$ $A = 1 \Rightarrow 1.0 = 0$
9	$\bar{\bar{A}} = A$	El doble complemento de una variable es igual a la variable.
10	$A + AB = A$	Aplicando la ley distributiva y las reglas 2 y 4 de esta tabla se puede deducir esta regla. $A + AB = A(1 + B)$ $A + AB = A.1$ $A + AB = A$
11	$A + \bar{AB} = A + B$	Aplicando varias reglas de esta tabla se puede deducir esta regla. $A + \bar{AB} = (A + AB) + \bar{AB}$ $A + \bar{AB} = (AA + AB) + \bar{AB}$ $A + \bar{AB} = AA + AB + \bar{A}\bar{A} + \bar{AB}$ $A + \bar{AB} = (A + \bar{A})(A + B)$ $A + \bar{AB} = 1.(A + B)$ $A + \bar{AB} = A + B$
12	$(A + B)(A + C) = A + BC$	Aplicando varias reglas de esta tabla se puede deducir esta regla. $(A + B)(A + C) = AA + AC + AB + BC$ $(A + B)(A + C) = A + AC + AB + BC$ $(A + B)(A + C) = A(1 + C + B) + BC$ $(A + B)(A + C) = A.1 + BC$ $(A + B)(A + C) = A + BC$

3.4 TEOREMAS DE DEMORGAN

DeMorgan, fue un matemático que estuvo relacionado con Boole. Propuso dos teoremas que están fuertemente relacionados con el álgebra de Boole. Los teoremas de DeMorgan, demuestran las equivalencias entre las compuertas NAND y Negativa OR y las compuertas NOR y Negativa AND.

El primer teorema de DeMorgan se enuncia como: el complemento de un producto de variables es igual a la suma de los complementos de las variables.

$$\overline{XY} = \overline{X} + \overline{Y}$$

Su tabla de verdad es:

Tabla 3.4 Tabla de verdad del primer teorema de DeMorgan

X	Y	\overline{XY}	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

El segundo teorema de DeMorgan se enuncia como: el complemento de una suma de variables es igual al producto de los complementos de las variables.

$$\overline{X + Y} = \overline{X} \cdot \overline{Y}$$

Su tabla de verdad es:

Tabla 3.5 Tabla de verdad del segundo teorema de DeMorgan

X	Y	$\overline{X + Y}$	$\overline{X} \cdot \overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Sus compuertas equivalentes son:

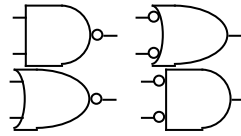


Figura 3.6 Equivalencias del teorema de DeMorgan

3.4.1 Aplicación de los teoremas de DeMorgan

El siguiente procedimiento presenta la aplicación de los teoremas de DeMorgan y el álgebra de Boole a una determinada expresión:

$$X = \overline{\overline{(A + BC)} + D \cdot \overline{(E + F)}}$$

Aplicando el segundo teorema de DeMorgan:

$$X = \left(\overline{(A + BC)} \right) \cdot \left(\overline{D \cdot \overline{(E + F)}} \right)$$

Aplicando la regla 9 del álgebra de Boole:

$$X = (A + BC) \cdot \left(\overline{D \cdot \overline{(E + F)}} \right)$$

Aplicando el primer teorema de DeMorgan:

$$X = (A + BC) \cdot \left(\overline{D} + \overline{\overline{(E + F)}} \right)$$

Aplicando la regla 9 del álgebra de Boole:

$$X = (A + BC) \cdot (\overline{D} + E + F)$$

3.5 SIMPLIFICACIÓN MEDIANTE EL ÁLGEBRA DE BOOLE

Cuando se desea implementar alguna expresión con dispositivos lógicos, es muy apropiado reducir la expresión a la forma más simple. Esto asegura el menor costo y esfuerzo, además asegura que la implementación es más eficiente. Mediante las leyes y reglas del álgebra de Boole, además de los teoremas de DeMorgan, se puede reducir una expresión a su forma más simple.

El propósito de la simplificación es usar el menor número posible de compuertas en una implementación.

Por ejemplo, mediante las técnicas del álgebra de Boole, simplificar la siguiente expresión:

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

Aplicando la ley distributiva:

$$X = AB + AB + AC + BB + BC$$

Aplicando la regla 5:

$$X = AB + AC + BB + BC$$

Aplicando la regla 7:

$$X = AB + AC + B + BC$$

Aplicando la regla 10:

$$X = AB + AC + B$$

Aplicando la regla 10:

$$X = AC + B$$

La implementación de las expresiones se presenta en la siguiente figura:

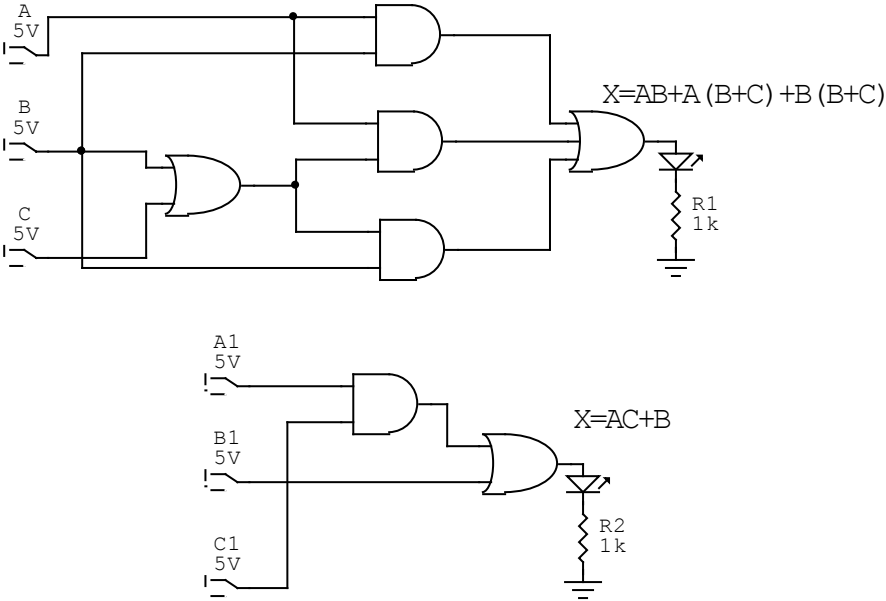


Figura 3.7 Implementación de simplificación mediante álgebra de Boole

3.6 MAPAS DE KARNAUGH

El mapa de Karnaugh proporciona un método para la simplificación de expresiones Booleanas. Si se aplica adecuadamente, genera las expresiones Booleanas más simples posibles.

Un mapa de Karnaugh es similar a una tabla de verdad, ya que muestra los posibles valores de las variables de entrada y de salida. En vez de organizar los valores en filas y columnas, el mapa de Karnaugh lo organiza en celdas, en donde cada celda representa un valor binario de las variables de entrada. El objetivo del mapa de Karnaugh es agrupar celdas adecuadamente. Los mapas de Karnaugh se aplican generalmente para expresiones con 3 o 4 variables. También se puede usar para 2 variables, aunque es muy simple y se puede usar para 5 variables aunque es demasiado complejo. El número de celdas de un mapa de Karnaugh es igual al número total de posibles combinaciones de las variables de entrada. Para 3 variables, el número de celdas es 8. Para 4 variables el número de celdas es 16.

3.6.1 Mapa de Karnaugh de tres variables

El mapa de Karnaugh de 3 variables es una tabla con un número de celdas equivalente a 2^3 celdas donde 3 es el número de variables del mapa. En este caso, A, B y C se utilizan para denominar a las variables. Los valores binarios de A y B se encuentran en la parte izquierda y los valores de C en la parte superior.

		C	
		0	1
AB	00	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$
	01	$\overline{A}B\overline{C}$	$\overline{A}BC$
	11	$A\overline{B}\overline{C}$	$A\overline{B}C$
	10	$A\overline{B}C$	$AB\overline{C}$

El valor de cada celda corresponde a la combinación especificada en el mapa de Karnaugh anterior. En la primera fila, se puede apreciar que A y B son negados, debido a que el valor de A y B en la parte de la izquierda son 00. Este concepto aplica para todas las celdas del mapa de Karnaugh.

3.6.2 Mapa de Karnaugh de cuatro variables

El mapa de Karnaugh de 4 variables es una tabla con un número de celdas equivalente a 2^4 celdas donde 4 es el número de variables del mapa. En este caso, A, B, C y D se utilizan para denominar a las variables. Los valores binarios de A y B se encuentran en la parte izquierda y los valores de C y D en la parte superior.

		CD			
		00	01	11	10
AB	00	$\overline{A}\overline{B}\overline{C}\overline{D}$	$\overline{A}\overline{B}C\overline{D}$	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}BC\overline{D}$
	01	$\overline{A}\overline{B}C\overline{D}$	$\overline{A}\overline{B}CD$	$\overline{A}B\overline{C}D$	$\overline{A}BCD$
	11	$A\overline{B}\overline{C}\overline{D}$	$A\overline{B}C\overline{D}$	$AB\overline{C}\overline{D}$	$ABC\overline{D}$
	10	$A\overline{B}C\overline{D}$	$A\overline{B}CD$	$AB\overline{C}D$	$ABCD$

El valor de cada celda corresponde a la combinación especificada en el mapa de Karnaugh anterior. En la primera fila, se puede apreciar que A y B son negados, debido a que el valor de A y B en la parte de la izquierda son 00. En la primera columna, se puede apreciar que C y D son negados, debido a que el valor de C y D en la parte de la superior son 00. Este concepto aplica para todas las celdas del mapa de Karnaugh.

3.6.3 Minimización de suma de productos mediante un mapa de Karnaugh

El mapa de Karnaugh se utiliza para reducir expresiones Booleanas. Una expresión de suma de productos minimizada está formada por el mínimo número de términos producto posibles con el mínimo número de variables por término. Cada término producto recibe el nombre de "minitérmino". Generalmente una expresión suma de productos minimizada puede ser implementada mediante un número de compuertas menor que su expresión estándar.

Por cada término de la expresión suma de productos, se coloca un 1 en el mapa en la celda correspondiente al valor del producto. Cuando se ha completado el mapa de Karnaugh correspondiente a la suma de productos dada, en dicho mapa deberá haber tantos unos como términos en la expresión. Las celdas que no contienen 1 son aquellas para las que la expresión es 0.

Entonces dada la siguiente expresión,

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

El mapa de Karnaugh es el siguiente:

		C	
		0	1
AB	00	1	1
	01		
	11	1	
	10	1	

Dada la siguiente expresión,

$$\overline{ABCD} + \overline{AB}CD + \overline{A}BCD + \overline{ABC}D + ABCD$$

El mapa de Karnaugh es el siguiente:

		CD			
		00	01	11	10
AB	00	1		1	
	01				
	11		1	1	
	10		1		

3.6.4 Simplificación de suma de productos mediante el mapa de Karnaugh

Para simplificar un mapa de Karnaugh con suma de productos, se deben seguir estos pasos:

1. Se colocan los unos en las celdas correspondientes dada la expresión que se desea simplificar.
2. Se agrupan las celdas adyacentes. Una celda adyacente a otra, es aquella en donde sólo cambia una variable. Los grupos se deben hacer de 2, 4, 8 o 16 celdas. Si un mapa, no contiene unos, entonces su resultado es 0. Si en un mapa se agrupan todas las celdas, su resultado es 1. Si un mapa tiene la mitad de celdas en 1 y la mitad de celdas en 0 y no se puede hacer una agrupación, la solución es la XOR o XNOR de las entradas.
3. Cada grupo debe contener el mayor número de celdas posible.
4. Se deben agrupar todos los unos del mapa, sin importar que una celda que contiene 1, se agrupe varias veces.
5. No se deben agrupar celdas que contienen 0.
6. Se analiza cada grupo. Se elimina la variable que en el grupo cambia. Si la variable se conserva, se mantiene su valor para ese minitérmino.

Para revisar el concepto, se muestra los siguientes ejemplos.

Dada la expresión:

$$X = \overline{ABCD} + \overline{AB}CD + \overline{A}BCD + \overline{ABC}D + ABCD + \overline{A}BCD + \overline{ABC}D + \overline{AB}CD$$

		CD			
AB \		00	01	11	10
00		1		1	
01		1	1		
11			1	1	
10			1	1	

Para el primero grupo, se tiene unos en las celdas:

$$\overline{A}\overline{B}\overline{C}\overline{D} \text{ y } \overline{A}\overline{B}CD$$

Entonces en este grupo se elimina la variable B, debido a que en una celda su valor es 0 y en la otra celda su valor es 1. El minitérmino quedará de la siguiente forma:

$$\overline{A}\overline{C}\overline{D}$$

Para el segundo grupo, se tiene unos en las celdas:

$$\overline{A}BC\overline{D} \text{ y } \overline{A}BCD$$

Entonces en este grupo se elimina la variable D. El minitérmino quedará de la siguiente forma:

$$\overline{A}BC$$

Para el tercer grupo, se tiene unos en las celdas:

$$A\overline{B}\overline{C}\overline{D}, A\overline{B}C\overline{D}, A\overline{B}CD \text{ y } A\overline{B}CD$$

Entonces en este grupo se elimina las variables B y C. El minitérmino quedará de la siguiente forma:

$$A\overline{D}$$

Para el cuarto grupo, se tiene unos en las celdas:

$$\overline{A}BC\overline{D} \text{ y } \overline{A}BCD$$

Entonces en este grupo se elimina la variable A. El minitérmino quedará de la siguiente forma:

$$\overline{B}C\overline{D}$$

Finalmente la solución de este mapa de Karnaugh es:

$$X = \overline{ACD} + \overline{ABC} + AD + \overline{BCD}$$

3.6.5 Simplificación de suma de productos usando Tabla de Verdad

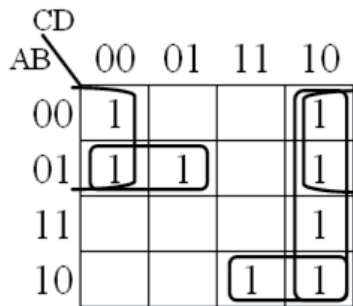
Para simplificar un mapa de Karnaugh con suma de productos, se deben seguir los pasos descritos anteriormente. En esta sección, se solucionará el mapa de Karnaugh tomando como valores de la expresión una tabla de verdad.

Dada la tabla de verdad:

Tabla 3.6 Simplificación mediante mapa de Karnaugh y tabla de verdad

Entradas				Salida
A	B	C	D	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Se debe colocar los unos de la salida en la celda cuya combinación corresponda. Posteriormente se hacen los grupos posibles.



Finalmente la solución de este mapa de Karnaugh es:

$$X = \overline{A}\overline{D} + \overline{A}B\overline{C} + \overline{A}BC + C\overline{D}$$

3.6.6 Condiciones indiferentes o valores No importa

En ocasiones, se tienen códigos de un número de bits pero no hay valor para algunas combinaciones. Por ejemplo para el caso del código BCD, que es un código de cuatro bits, pero sólo se consideran 10 combinaciones. En este caso a las celdas cuyas combinaciones no se consideran, se les coloca un valor "No importa", que se describe con "X". En el momento de hacer los grupos, se puede considerar estos valores como 1 o 0, a conveniencia de la solución del mapa.

Dada la tabla de verdad:

Tabla 3.7 Simplificación de mapa de Karnaugh con valores No importa

Entradas				Salida
A	B	C	D	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	X

1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Se debe colocar los unos de la salida y los valores No importa en la celda cuya combinación corresponda. Posteriormente se hacen los grupos posibles.

		CD			
		00	01	11	10
AB	00	1			1
	01	1	1		1
	11	X	X	X	X
	10			X	X

Finalmente la solución de este mapa de Karnaugh es:

$$X = \overline{AD} + \overline{BC}$$



Ejercicios Propuestos

1. Simplificar la expresión Booleana $X = \overline{ABC} + \overline{ABC} + \overline{AB}$
2. Simplificar la expresión Booleana $X = (\overline{ABC} + \overline{ABC}) + (\overline{AB} + AC)$
3. Simplificar la expresión Booleana $X = ABC + \overline{ABC}$
4. Implementar la expresión Booleana $X = (A \oplus B)C + \overline{ABC} + (A \oplus B)D$
5. Simplificar la expresión Booleana $\overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC}$
6. Simplificar mediante mapas de Karnaugh la siguiente tabla de verdad:

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

7. Simplificar mediante mapas de Karnaugh la siguiente tabla de verdad:

Entradas				Salidas		
A	B	C	D	X	Y	Z
0	0	0	0	0	1	1
0	0	0	1	1	0	1
0	0	1	0	0	1	0
0	0	1	1	0	0	1
0	1	0	0	0	0	0
0	1	0	1	1	1	1
0	1	1	0	0	1	0
0	1	1	1	1	1	1
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	0	0
1	1	0	0	1	1	1
1	1	0	1	1	1	0
1	1	1	0	0	0	0
1	1	1	1	0	1	0

8. Simplificar mediante mapas de Karnaugh la siguiente tabla de verdad:

Entradas				Salidas		
A	B	C	D	X	Y	Z
0	0	0	0	0	1	1
0	0	0	1	1	0	1
0	0	1	0	0	1	1
0	0	1	1	1	1	0
0	1	0	0	0	1	0
0	1	0	1	1	1	1
0	1	1	0	0	1	0
0	1	1	1	1	1	1
1	0	0	0	0	0	1
1	0	0	1	0	0	0