Capítulo 2

Álgebra Booleana y Compuertas lógicas.

Temario

Objetivos

Al concluir este capítulo el lector estará en capacidad de:

1.- Construir la tabla de verdad de las operaciones lógicas básicas AND, OR y NOT

2.-. Construir la tabla de verdad de funciones compuestas, aplicando los operandos básicos.

3.- Construir las tablas de verdad de los operandos NAND, NOR y EXOR.

4.- Aplicar los teoremas y las leyes asociativa, distibutiva, conmutativa, de absorción y de complementaridad para reducir funciones lógicas a su mínima expresión.

5.- Aplicar las leyes de DeMorgan para simplificar funciones lógicas.

6.- Construir y transformar funciones lógicas, en formatos de maxitérminos y minitérminos.

7.- Obtener la mínima expresión de una función lógica empleando los Mapas de Karnaugh.

8.- Identificar los códigos y símbolos que representan a las compuertas lógicas AND, OR, NOT, NAND, NOR y EXOR.

9.- Diseñar los circuitos con compuertas lógicas considerando a la función como dato de entrada.

10.- Diseñar circuitos electrónicos empleando exclusivamente compuertas NAND, a partir de una función lógica expresada en minitérminos.

11.- Diseñar circuitos electrónicos empleando exclusivamente compuertas NAND, a partir de una función lógica expresada en maxitérminos..

12.- Diseñar circuitos electrónicos empleando exclusivamente compuertas NOR, a partir de una función lógica expresada en maxitérminos.

13.- Diseñar circuitos electrónicos empleando exclusivamente compuertas NOR, a partir de una función lógica expresada en minitérminos.

14.- Obtener las funciones lógicas y diseñar los correspondientes circuitos electrónicos, que controlan el trabajo de una máquina o proceso industrial a partir de los datos de operación de los mismos.

Introducción.

El análisis, síntesis y diseño de los sistemas digitales está basado en la herramienta algebraica conocida como *Álgebra Booleana* (George Boole, 1815). Está fundamentada en postulados básicos (axiomas), teoremas y leyes. Fue en 1854 cuando publicó su trabajo titulado *An Investigation into the Laws of Though*, el cual sirvió como base para la teoría matemática de probabilidades. Recientemente, el crecimiento y el correspondiente éxito de los sistemas computacionales e informáticos ha hecho que Boole sea considerado como uno de los padres fundadores de dichas áreas, debido a la enorme e innegable influencia de su teoría en el análisis y diseño de soluciones para sistemas digitales que van desde la tecnología de las telecomunicaciones; operación, manejo y transferencia de información digital; hasta los problemas de automatización.

Se emplearán variables booleanas para representar señales entradas y salidas de sistemas binarios, esto es, que tendrán la posibilidad de adquirir sólo alguno de dos posibles estados, 0 ó 1. Estos valores son sólo simbólicos, análogos a conceptos como Bajo/Alto; Falso/Verdadero; ON/OFF, entre otras expresiones. Así, tomando como ejemplo el estado de operación de una lámpara que ilumina la esquina de una calle se puede asociar el valor lógico de 1 al estado de "lámpara encendida," mientras que el estado de "lámpara apagada" quedará representado por el valor de 0. Empleando expresiones matemáticas, tendríamos lo siguiente:

X = Estado de la lámpara
X = 1 = Lámpara prendida
X = 0 = Lámpara apagada

2.1 Postulados.

1.- El álgebra Booleana es un sistema algebraico formado esencialmente por un conjunto *M* de elementos y dos operaciones básicas, más no las únicas, *"+ (OR)"* y " *(AND)*´ . Estos operandos actúan sobre el conjunto de variables de entradas que pueda poseer el sistema en cuestión. Lo más elemental es que se cuente con sólo dos de ellas, p ej. X y Y, de tal forma que el operando sobre ambas generará un resultado asociado a una función lógica de salida, F(X,Y), el cual será un subconjunto del universo de resultados formado por dos posibles valores lógicos, 0 ó 1.

Operación básica AND. Este operando expresa simbólicamente al concepto de intersección, empleado en la teoría de conjuntos. Si se cuenta con dos variables lógicas X y Y, el operando lógico sobre ellas generaría dos posibles resultados, conjunto vacío ó conjunto lleno. Al primero de ellos le corresponde el valor lógico de 0, por consiguiente al segundo le corresponderá el valor de 1. Su expresión simbólica sería:

F(X,Y) = XY

El total de combinaciones posibles que se pueden generar al asignarle valores a las variables, es de 2^n, ya que la base es binaria y *n* es la cantidad de variables existentes en el operando. Por lo tanto, para el caso de las dos variables, X y Y, la función lógica puede ser analizada mediante una tabla de verdad mostrando cuatro combinaciones posibles, $2^2 = 4$. La figura 2.1 muestra la tabla con las combinaciones correspondientes.

X (ENTRADA)	Y (ENTRADA)	F (SALIDA)
0	0	0
0	1	0
1	0	0
1	1	1

2.1 Tabla de verdad para la función lógica de dos variables

En el caso de un operando lógico AND de tres variables, la cantidad de combinaciones sería $2^3 = 8$. La tabla 2.2 muestra todas estas posibilidades.

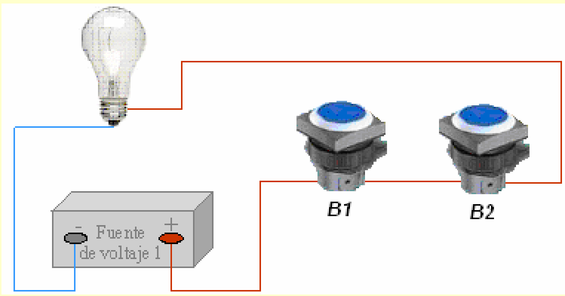
X (ENTRADA)	Y (ENTRADA)	Z (ENTRADA)	F(X,Y,Z) (SALIDA)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Tabla 2.2 Tabla de verdad para la función lógica de tres variables.

Ejemplo 2.1 Suponga que una instalación eléctrica cuenta con dos botones, B1 y B2, para una lámpara de emergencia. La condición de encendido es que se mantengan sostenidos ambos a la vez. Analizando la tabla 2.1, de manera inmediata se puede concluir que la instalación correspondiente obedece a las cuatro combinaciones expresadas en ella:

i) Si sólo se oprime B1, entonces la lámpara no enciende
ii) Si sólo se oprime B2, la lámpara no enciende.
iii) Si no se oprimen ni B1 ni B2, la lámpara no enciende
iv) Si se oprimen B1 y B2 a la vez, entonces se prenderá la lámpara.

La figura muestra la representación esquemática de la instalación eléctrica de la solución.



el operando AND, la operación lógica OR sobre X y Y, generará dos posibles resultados, conjunto vacío ó conjunto lleno, con sus correspondientes valores lógicos, 0 y 1, respectivamente.

$$F(X,Y) = X + Y$$

La cantidad de combinaciones seguirá siendo $2^2 = 4$. La figura 2.2 muestra la tabla con las combinaciones correspondientes.

X (ENTRADA)	Y (ENTRADA)	F (SALIDA)
0	0	0
0	1	1
1	0	1
1	1	1

2.2 Tabla de verdad para la función lógica de dos variables

Analizando con detenimiento la tabla 2.2, se observa que la única condicionante para que F(X,Y) = 1 es que cualquiera de las variables posea el valor lógico de 1. De esto último se deriva la expresión "OR": que X = 1 "o que" Y =1. Como efecto redundante es que ambas sean igual a uno.
Para el caso de tres variables, la cantidad de combinaciones sigue siendo $2^3 = 8$. La tabla 2.3 muestra los resultados correspondientes.

X (ENTRADA)	Y (ENTRADA)	Z (ENTRADA)	F(X,Y,Z) = XYZ
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

Tabla 2.2 Tabla de verdad para la función lógica de tres variables.

Ejemplo 2.2 Considere la situación de la lámpara y los botones del ejemplo 2.1. La aplicación de la función lógica generará también cuatro posibles combinaciones para el encendido:

i) Si sólo se oprime B1, entonces la lámpara encenderá
ii) Si sólo se oprime B2, igualmente encenderá la lámpara
iii) Si no se oprimen ni B1 ni B2, la lámpara no enciende
iv) Si se oprimen B1 y B2 a la vez, entonces se prenderá la lámpara.

La figura muestra la representación esquemática de la instalación eléctrica de la solución para la función "OR".

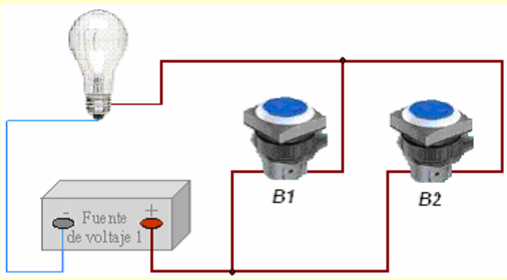


Figura 2.2 Esquema eléctrico para una lámpara, aplicando la función "OR"

Operación básica Inversora (NOT). Un primer razonamiento lógico inmediato al analizar los dos operandos tratados hasta el momento, sería el del manejo de la no existencia del estado X, dicho de otra forma, su parte complementaria. Esto es, el valor de exclusión 1 – X. La teoría establece que cualquier variable X tiene la oportunidad de poseer alguno de dos posibles valores, 1 ó 0, mencionados con anterioridad. Entonces, si en algún momento dado X = 1, el valor de exclusión, o complementario sería (1 – x) = 0. Por lo contrario, si X = 0, entonces (1-x) = 1. De aquí en adelante los términos a emplear serán los de X y su complemento (1 – x) = $\overline{X}$, la cual podrá diferentes adjetivos, tales como: “inversora”; “negación” o “NOT”. Su tabla de verdad correspondiente sería:

X	$\overline{X}$
0	1
1	0

Ejemplo 2.3. Considerando el ejemplo de la lámpara, se aplicará un botón para la operación del mismo, en este caso las dos posibilidades serían:

i) Si B1 no está oprimido, entonces la lámpara estará prendida
ii) Si B1 está oprimido, entonces la lámpara se encenderá.

Para el cableado eléctrico de este operando se requerirá de un botón cuya posición sea normalmente cerrada, esto es, que bajo condición normal de operación permita el paso de voltaje por el servicio correspondiente. Cuando se active o pulse, los contactos del botón dejan de transmitir la señal eléctrica. La figura 2.3 muestra el esquema eléctrico.

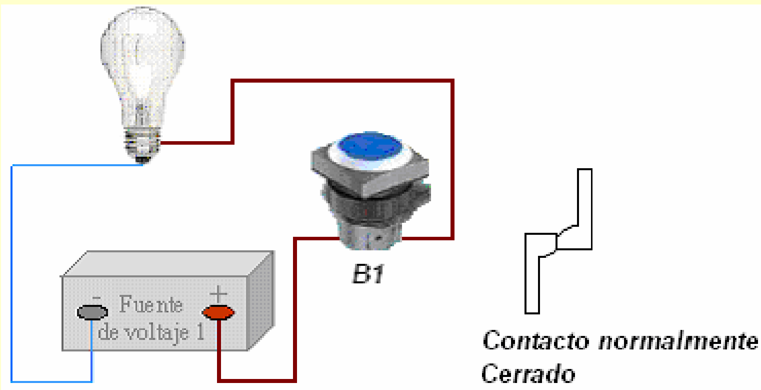


Figura 2.3 Instalación eléctrica par la función inversora NOT

2.2 Operaciones lógicas complementarias.

Además de las operaciones básicas, existen otras que son de relevancia para el desarrollo de la teoría y que son obtenidas como consecuencia de las primeras. Entre ellas estarían las funciones "NAND", "NOR" y "EXOR". A continuación se describen con más detalle cada una de ellas.

Operación NAND

Es una función compuesta entre la AND y la NOT. El orden de ejecución es fundamental, que en este caso primero se realiza la operación de intersección entre dos variables, X y Y, para posteriormente aplicarle la complementaridad. La tabla de verdad correspondiente sería tal como se muestra en la siguiente figura.

X	Y	XY(AND)	$\overline{XY}$ (NAND)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

La extensión a tres variables es inmediata, siguiendo la tabla anterior.

X	**Y**	**Z**	**XYZ**	$\overline{XYZ}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Operación NOR

Es una función compuesta entre la OR y la NOT. Al igual que con al NAND, el orden de sigue siendo vital. Primero se realiza la operación de unión entre dos variables, X y Y, para posteriormente aplicarle la inversión. La tabla siguiente muestra su lógica correspondiente.

X	Y	X + Y (OR)	$\overline{X+Y}$ (NOR)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Operación NOR para tres variables:

X	**Y**	**Z**	**X+Y+Z**	$\overline{X+Y+Z}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Operación EXOR
Es una aplicación específica de la operación OR, también se le conoce como función "OR Exclusiva". Su nombre obedece a que el resultado lógico 1 se consigue única y exclusivamente cuando, en el caso de tener dos variables de entrada, sólo una de ellas posee el valor lógico de uno. Para clarificar esto, revise la siguiente tabla.

X	Y	$X \oplus Y(EXOR)$
0	0	0
0	1	1
1	0	1
1	1	0

La extensión a tres variables requiere de un poco de cuidado, ya la incorporación y aplicación de la EXOR sobre la tercer variable se hace con respecto al resultado obtenido con dos variables, esto es, no se recomienda aplicar el operando directamente sobre X, Y y Z. Tome como referencia la tabla siguiente.

X	Y	$X \oplus Y = W$	Z	$W \oplus Z$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0
0	0	0	1	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	1

2.3 Teoremas.

En los trabajos y artículos de George Boole se establecen los fundamentos del álgebra booleana. Axiomas, teoremas y leyes son establecidas y demostradas con el rigor debido, empleando la teoría de clases y sus respectivas relaciones. En todo caso, en este trabajo no se persigue el objetivo de volver a ejecutar dichas demostraciones, en todo caso sólo se harán las presentaciones, comprobaciones y por último las aplicaciones de las mismas. Para el lector más asiduo, se le recomienda el artículo "The Calculus of Logic", George Boole [Cambridge and Dublín Mathematical Journal, Vol. III (1848), pp. 183 – 98].

Dentro del conjunto de Leyes, existen tres que son básicas y que son aplicadas tanto en el álgebra lineal como en el álgebra booleana, que a saber son: conmutativa, asociativa y distributiva. A continuación se revisan y comprueban cada una de ellas.

Ley conmutativa: Sean las variables lógicas X y Y, de tal manera que la relación siguiente queda satisfecha:

$$X + Y = Y + X$$

Tomando como ejemplo a la lámpara que puede ser encendida mediante un botón B1 ó un botón B2, lo que nos dice esta ley es que no importa cuál botón se oprima primero, el resultado será el mismo en cuanto al encendido de la misma.
El lector puede realizar la comprobación correspondiente empleando tablas de verdad

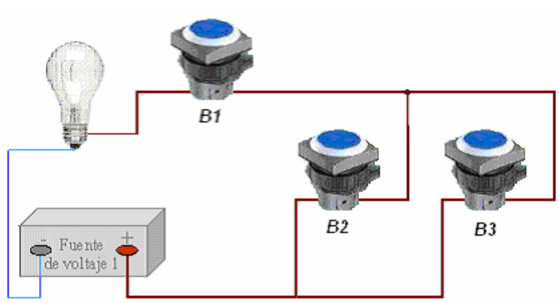
Ley distributiva. Sean las variables lógicas X y Y, la distributividad entre tres variables establece que:

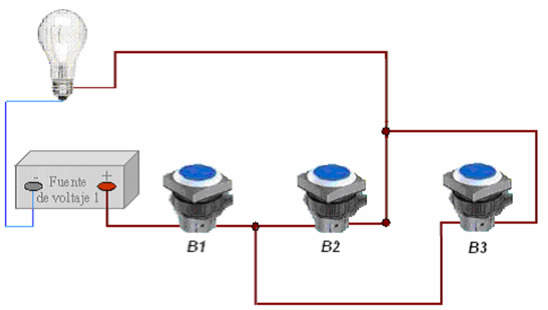
$$A(B + C) = AB + AC$$

Para la comprobación de la relación se puede revisar la tabla de verdad siguiente

A	B	C	(B+C)	AB	AC	A(B+C)	AB + AC
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

En cuanto a la aplicación de esta ley se puede seguir empleando el mismo ejemplo de la lámpara, tal como lo muestra la figura siguiente.





Ley asociativa. Sean las variables lógicas X y Y, la distributividad entre tres variables establece que:

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

Esta ley indica que, para efecto del resultado, no importa cómo se agrupen las variables para aplicar el operando OR, al final siempre será el mismo.

Propiedades

Existe un conjunto de propiedades básicas, útiles para la simplificación de funciones booleanas. Algunas de ellas son inmediatas de comprobar, mientras que otras requieren de un poco más de esfuerzo para verificar su relación. En la tabla siguiente se muestran algunas de ellas, e inmediatamente se realiza su comprobación correspondiente.

A) Operación con 1.

1.- $1 + 0 = 1$

2.- $1 + A = 1$

3.- $1*1 = 1$

4.- $1*A = A$

5.- $1*0 = 0$

B) Operaciones con 0

1.- $0 + 0 = 0$

2.- $0 + A = A$

3.- $0*A = 0$

C) Absorbentes

$A*A = A$

$A + A = A$

D) Complemento

.

$\overline{\overline{(X)}} = X$

$X\overline{X} = 0$

$X + \overline{X} = 1$

Ejemplo 2.4 Demostrar las siguientes relaciones.

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

$$XY + X\overline{Y} = X(Y + \overline{Y}) = X(1) = X$$

b) $X + XY = X$

$$X + XY = X(1 + Y) = X(1) = X$$

c) $(X + \overline{Y})Y = XY$

$$(X + \overline{Y})Y = XY + Y\overline{Y} = XY + 0 = XY$$

d) $(X + Y)(X + \overline{Y}) = X$

$$(X + Y)(X + \overline{Y}) = XX + X\overline{Y} + XY + Y\overline{Y}$$
$$= X + X(Y + \overline{Y}) + 0 = X + X(1) = X + X = X$$

e) $X(X + Y) = X$

$$X(X + Y) = XX + XY = X + XY = X(1 + Y) = X(1) = X$$

f) $X + \overline{X}Y = X + Y$

$$X + \overline{X}Y = X(1 + Y) + \overline{X}Y = X + XY + \overline{X}Y = XX + XY + \overline{X}Y$$

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

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

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

g) $(X+Y)(X+Z) = X+YZ$

$$(X+Y)(X+Z) = X+YZ$$

$$(X+Y)(X+Z) = XX + XZ + XY + YZ$$

$$= X + XZ + XY + YZ = X(1+Z) + XY + YZ$$

$$X + XY + YZ = X(1+Y) + YZ = X + YZ$$

Ejemplo 2.5 Empleando Álgebra Booleana, simplificar las siguientes expresiones lógicas.

a) $[A\overline{B}(C+BD) + \overline{A}\,\overline{B}]C$

$$(A\overline{B}C + AD0 + \overline{A}\,\overline{B})C$$

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

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

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

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

b) $\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$

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

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

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

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

$$BC + \overline{B}(A+\overline{C}) = BC + A\overline{B} + \overline{B}\,\overline{C}$$

2.4 Leyes de De Morgan

Augustus De Morgan fue un matemático muy importante del siglo XIX, cuyas mayores contribuciones fueron en el área de la lógica proposicional. Su trabajo más importante, *Formal logic,* incorpora el concepto de cuantificación de los predicados, sin el cual, hasta ese entonces resultaría prácticamente imposible resolver algunos problemas, desde el punto de vista de la lógica aristotélica. Dos tipos de proposiciones son las que construyen el universo de la lógica de De Morgan: simples y compuestas. La primera de ellas carecen de conectivos, mientas que la segunda emplea los conectores – operandos lógicos AND y OR para conectar a dos o más proposiciones simples, por ejemplo:

En un grupo específico de automóviles:

- La mayoría de ellos son de último modelo
- La mayoría de ellos son rojos.

Las proposiciones mostradas son del carácter simple, más sin embargo a partir de ellas se puede inferir una compuesta. Como inferencia se entiende al proceso de obtener una proposición cierta a partir de proposiciones simples las cuales son consideradas como verdaderas. Así, una inferencia inmediata compuesta a partir de las dos proposiciones simples verdaderas sería la siguiente:

- Algunos automóviles son de último modelo y de color rojo.

En donde se observa que como elemento de composición se emplea al conector lógico AND.

En términos particulares, del trabajo de De Morgan, son dos leyes o reglas las que guardan un interés específico en la teoría de los sistemas digitales:

$$\overline{(X+Y+Z+\ldots)} = \overline{X}\,\overline{Y}\,\overline{Z}..$$

$$\overline{XYZ\ldots} = \overline{X} + \overline{Y} + \overline{Z}\ldots.$$

Usualmente, pueden ser leídas de acuerdo a las siguientes expresiones:

“La negación de una suma es igual a la multiplicación de las negaciones” y “La negación de una multiplicación es igual a la suma de las negaciones”, respectivamente.

A continuación se revisan algunos ejemplos de aplicación.

Ejemplo 2.6. Aplicar las leyes de DeMorgan a las siguientes expresiones:

a) $F_1 = \overline{\overline{(A+B)}+C} = \overline{\overline{(A+B)}}\,\overline{C} = (A+B)\overline{C}$

b) $F_2 = \overline{A+B+\overline{C}} = \overline{(A+B)} \bullet \overline{\overline{C}} = \overline{A}\,\overline{B}C$

c) $F_3 = \overline{A \bullet B + (C + \overline{D})}$

$$F_3 = \overline{\overline{A} \bullet B} \bullet \overline{C + \overline{D}} = (\overline{\overline{A}} + \overline{B})(\overline{C} \bullet \overline{\overline{D}}) = (A + \overline{B})(\overline{C} \bullet D)$$

d) $F_4 = \overline{\overline{A}(B + C) + \overline{B} \bullet D}$

$$F_4 = \overline{\overline{A}(B + C)} \bullet \overline{\overline{B}D} = (\overline{\overline{A}} + \overline{(B + C)}) \bullet \overline{\overline{B}D}$$

$$F_4 = (A + \overline{(B + C)})(\overline{\overline{B}} + \overline{D}) = (A + \overline{B}\,\overline{C})(B + \overline{D})$$

$$F_4 = AB + A\overline{D} + \overline{B}\,\overline{C}B + \overline{B}\,\overline{C}\,\overline{D} = AB + A\overline{D} + \overline{B}\,\overline{C}\,\overline{C}$$

e) $\overline{AB + AC} + \overline{A}\,\overline{B}C$

$$\overline{AB}\,\overline{AC} + \overline{A}\,\overline{B}C$$

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

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

$$\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B}(1 + C) + \overline{B}\,\overline{C} = \overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

$$\overline{A}(1 + \overline{C}) + \overline{A}\,\overline{B} + \overline{B}\,\overline{C} = \overline{A} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

$$\overline{A}(1 + \overline{B}) + \overline{B}\,\overline{C} = \overline{A} + \overline{B}\,\overline{C}$$

2.5 Funciones lógicas.

Funciones lógicas: Una función lógica es aquella que se encuentra sujeta a los principios del álgebra booleana y que representa a un proceso o secuencia de operaciones.
En términos generales una función lógica puede ser representada en "minitérminos" o "maxitérminos", ya sea que estén o no en su forma canónica.
Una función lógica canónica, en minitérminos, es aquella que se expresa como la sumatoria de elementos compuestos en operación "AND"de todas las variables que intervienen en el proceso.

$$F(A,B,C) = \sum m(3,6,7) = \underbrace{\overline{A}BC + AB\overline{C} + ABC}_{\text{Minitérminos}}$$

Una función lógica canónica, en maxitérminos, es aquella que se expresa como la multiplicatoria de elementos compuestos en operación “OR”de todas las variables que intervienen en el proceso.

$$F(A,B,C) = \prod M(0,5) = \underbrace{(\overline{A} + B + \overline{C})(A + B + C)}_{\textit{Maxitérminos}}$$

Ejemplo 2.7 Expresar la siguiente función $f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C}$ en su equivalente de maxitérminos

Solución: $f(A,B,C) = m_0 + m_1 + m_4 = \sum m(0,1,4)$

Tabulemos tanto la función *f*, como su complemento

A	B	C	f	$\overline{f}$
0	0	0	1	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Se observa que el complemento de *f*, viene expresado por:

$$\overline{f} = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Al aplicar la ley de DeMorgan a esta última expresión, nos queda:

$$f = \overline{\overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC}$$

$$f = \overline{\overline{A}B\overline{C}}\;\overline{\overline{A}BC}\;\overline{A\overline{B}C}\;\overline{AB\overline{C}}\;\overline{ABC}$$

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

Relacionando la tabla con la función *f* expresada tanto en minitérminos como en maxitérminos, se concluye que existe una relación directa entre ambas, dada por:

$$\sum m(0,1,4) = \prod M(2,3,5,6,7)$$

En donde cada maxitérmino se obtiene de la suma de los complementos de cada renglón en donde la función se hace cero, de aquí que:

$$f = (A + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C}) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C}$$

Ejemplo 2.8. Expresar en maxitérminos la función $f(a,b,c,d)=\sum m(0,1,2,3,5,9,11,14)$

Solución:

$$f(a,b,c,d)=\sum m(0,1,2,3,5,9,11,14)=\prod M(4,6,7,8,10,12,13,15)$$

$$\prod M(4,6,7,8,10,12,13,15)=(a+\overline{b}+c+d)(a+\overline{b}+\overline{c}+d)(a+\overline{b}+\overline{c}+\overline{d})(\overline{a}+b+c+d)(\overline{a}+b+\overline{c}+d)..$$
$$(\overline{a}+\overline{b}+c+c)(\overline{a}+\overline{b}+c+\overline{d})(\overline{a}+\overline{b}+\overline{c}+\overline{d})$$

Ejemplo 2.9. Obtener el equivalente en maxitérminos de la siguiente función.

$f(a,b,c)=ab+b\overline{c}+a\overline{b}c+a\overline{b}\overline{c}+ab\overline{c}$

Solución. Como se puede observar, la función está expresada en minitérminos, más no es canónica ya que a los dos primeros elementos de la suma les falta una de las tres variables, por lo tanto se procede a completar esos dos minitérminos.

$$ab=ab(c+\overline{c})=abc+ab\overline{c}$$
$$b\overline{c}=(a+\overline{a})b\overline{c}=ab\overline{c}+\overline{a}b\overline{c}$$

$$f(a,b,c)=abc+ab\overline{c}+ab\overline{c}+\overline{a}b\overline{c}+a\overline{b}c+a\overline{b}\overline{c}+ab\overline{c}$$

$$f(a,b,c)=abc+ab\overline{c}+\overline{a}b\overline{c}+a\overline{b}c+a\overline{b}\overline{c}=\sum m(2,4,5,6,7)$$

$$\sum m(2,4,5,6,7)=\prod M(0,1,3)=(a+b+c)(a+b+\overline{c})(a+\overline{b}+\overline{c})$$

Ejemplo 2.10 Obtener el equivalente en minitérminos de la siguiente función:

$$f(a,b,c)=(a+b)(a+\overline{c})(a+\overline{b}+c)$$

Solución. La función está expresada en maxitérminos, no canónica. Primero se tienen que completar los dos primeros maxitérminos.

$$(a+b)=(a+b+c)(a+b+\overline{c})$$

$$(a+\overline{c})=(a+b+\overline{c})(a+\overline{b}+\overline{c})$$

Por lo tanto:

$$f(a,b,c)=(a+b+c)(a+b+\overline{c})(a+b+\overline{c})(a+\overline{b}+\overline{c})(a+\overline{b}+c)$$

Factorizando: $f(a,b,c)=(a+b+c)(a+b+\overline{c})(a+\overline{b}+\overline{c})(a+\overline{b}+c)=\prod M(0,1,2,3)$

$$\prod M(0,1,2,3)=\sum m(4,5,6,7)=a\overline{b}\overline{c}+a\overline{b}c+ab\overline{c}+abc$$

2.6 Mapas de Karnaugh

Generalmente las funciones lógicas siempre pueden ser simplificadas empleando los teoremas y leyes del álgebra booleana. Sin embargo, cuando se trata de simplificar funciones relativamente complejas el procedimiento se vuelve muy laborioso, ya que no existe una metodología o procedimiento que pueda ser aplicado en forma sencilla y rápida, e inclusive se puede no tener la certeza de haber llegado a la mínima expresión. De allí entonces la necesidad de revisar procesos alternativos para el tratamiento algebraico, específicamente para la reducción o simplificación de las funciones lógicas. Como alternativas, el lector podrá encontrar en la bibliografía los métodos de *Mapas de Karnaugh* y *Quine – McCluskey*, de los cuales este último queda más allá de los alcances de este libro.

Los Mapas de Karnaugh son una herramienta para la simplificación de funciones lógicas. Tanto el tamaño como su formato de presentación son similares a una tabla de verdad y dependen de la cantidad de variables contenidas en la función a tratar. En él se vacía en celdas la información de las entradas y de sus respectivas salidas, formado una matriz con unos y ceros, correspondientes al valor lógico de las salida en correspondencia con todas las combinaciones posibles existentes de las entradas. Si se cuenta con dos variables de entrada la cantidad de celdas de la matriz o mapa será igual a $2^2 = 4$; si son tres variables entonces las celdas serán $2^3 = 8$, en general se tendrían 2^n celdas, siendo *n* la cantidad de variables existentes en la función. Para el caso más simple sería una función que depende de sólo dos variables, *F(a,b)* , y su correspondiente mapa de Karnaugh sería:

F(a,b)

b \ a	0	1
0		
1		

En donde 0 y 1 son los dos posibles estados que puede poseer cada variable. En cada cuadrante deberá de indicarse un 0 o un 1, dependiendo del estado lógico de cada uno de los minitérminos, o maxitérminos que contenga la función. En este caso no es relevante etiquetar primero con cero o con uno los títulos de la columna y/o de la fila. Pudo haberse comenzado con la etiqueta de “1” la primera columna para *a* y posteriormente asignar el “0” a la segunda columna. Para el caso de tres o más variables será importante definir el concepto de “celdas adyacentes”.

Suponga que se cuenta con una función *F(a,b,c,d)* a la cual se le desea construir su mapa. En este caso deberá de construirse una matriz de $2^4 = 16$ celdas, en arreglo de 4 x 4, esto es, cuatro filas y cuatro columnas, en donde cada una de ellas hará referencia a los conjuntos de variables *ab* y *cd,* respectivamente, alternando las cuatro combinaciones posibles: 00 01 11 10. Obsérvese que de par a par sólo se da un cambio en uno solo de los bits, por ejemplo, de 01 a 00 sólo cambia el bit menos significativo. Así, las *Celdas adyacentes* se definen como aquellas en la cual se observa sólo un cambio en uno de sus bits. Siguiendo este principio, para la función de cuatro variables, la matriz de Karnaugh pudiera tener cualquiera de los dos arreglos siguientes, según la siguiente figura.

cd \ ab	00	01	11	10
00				
01				
11				
10				

cd \ ab	00	10	11	01
00				
10				
11				
01				

Opciones *a* y *b* para arreglos de la matriz de Karnaug, para una función con cuatro variables.

Por inspección visual, se puede determinar que una celda es adyacente a otra siempre y cuando ambas se encuentren una junto a otra en la misma fila y/o en el mismo renglón, más no si son diagonales. Los extremos, 00 y 10 se consideran adyacentes entre sí. Esto mismo se aplica para las celdas extremas definidas por las cuatro intersecciones, 00 con 00; 00 y 10; 10 y 00; 10 intersección con 10.

Ejemplo 2.11. Obtener la representación en K de las funciones e indicar gráficamente a las celdas que contienes unos lógicos adyacentes.

a) $f(a,b)=\sum m(0,1,3)=00+01+11$

b) $f(a,b,c)=\sum m(1,3,4,6,7)=001+011+100+110+111$

c) $f(a,b,c,d)=\sum m(1,3,5,6,7,11,13,15)=001+011+101+110+111+1011+1101+1111$

Solución a) La distribución de los unos y ceros de asociados a F, según la combinación de las entradas es:

b \ a	0	1
0	1	0
1	1	1

Ahora, agrupando a los unos adyacentes:

b \ a	0	1
0	1	0
1	1	1

Obsérvese que los unos de 00 y 11 son diagonales, por lo tanto no son adyacentes entre ellos.

Solución b) Distribución de los unos. Las celdas que se encuentran vacías corresponden a aquellas en las cuales la función tiene un valor lógico de cero.

bc \ a	0	1
00		1
01	1	
11	1	1
10		1

La selección de los unos adyacentes tendría varias opciones.

Opción a:

bc \ a	0	1
00		1
01	1	
11	1	1
10		1

Opción b

bc \ a	0	1
00		1
01	1	
11	1	1
10		1

Se deja al lector la búsqueda de otra posible opción

Ejemplo 2.11.

c) $f(a,b,c,d) = \sum m(1,3,5,6,7,11,13,15) = 0001 + 0011 + 0101 + 0110 + 0111 + 1011 + 1101 + 1111$

Solución c) La distribución de los unos y ceros de asociados a *f*, según la combinación de las entradas es:

cd \ ab	00	01	11	10
00				
01	1	1	1	
11	1	1	1	1
10		1		

Obsérvese que pueden existir varias más opciones.

En las siguientes secciones se presentarán reglas de selección de celdas adyacentes, para poder aplicar la metodología de Karnaugh.

Para iniciar el proceso de minimización, el cual generará una expresión de tamaño mínimo en cuanto a la cantidad de términos y variables, primero se debe de dibujar el mapa con sus correspondientes unos. Posteriormente deberán de formarse grupos de unos, de acuerdo a ciertas reglas. Por último se determinará directamente la expresión mínima a partir de esas agrupaciones.

A continuación se indican los pasos a seguir para la agrupación de los minitérminos (unos).

1.- La cantidad de minitérminos (unos) a seleccionar deberán de ser escogidos de tal forma que cumplan con la expresión 2^n

2.- Los minitérminos seleccionados deberán de ser adyacentes contiguos -no diagonales.

3.- Construir los grupos del tamaño más grande posible, de acuerdo al punto uno.

4.- Todos los minitérminos deberán de quedar incluidos en al menos un grupo. Esto es, es posible que un minitérmino pertenezca a más de una agrupación.

Ejemplo 2.12. Aplicar las reglas de agrupación a la función del ejercicio 2.11.

c) $f(a,b,c,d) = \sum m(1,3,5,6,7,11,13,15) = 0001 + 0011 + 0101 + 0110 + 0111 + 1011 + 1101 + 1111$

Solución c) La distribución de los unos y ceros de asociados a *f*, según la combinación de las entradas es:

cd \ ab	00	01	11	10
00				
01	1	1	1	
11	1	1	1	1
10		1		

Un acercamiento de cada una de las cuatro agrupaciones se muestra enseguida.

cd \ ab	00	01
00		
01	1	1
11	1	1

01	11
1	1
1	1

cd \ ab	00	01
00		
01		
11		1
10		1

11	10
1	1

Queda al lector encontrar otra distribución de minitérminos.

2.- Sea x una variable asociada a un grupo de minitérminos, si ésta sufre cambio de estado conforme se desplaza por su respectivo grupo, ya sea en forma horizontal o vertical, entonces dicha variable no aparecerá en el minitérmino resultante. Las variables que no conmuten de estado aparecerán en la solución multiplicándose entre ellas. Como resultado final se tendrá una suma de multiplicaciones, en donde cada minitérmino de esta solución está asociado a una agrupación de unos del mapa.

Ejemplo 2.11. Obtener la mínima expresión para $f_1(a,b) = \sum m(0,1,3) = 00 + 01 + 11$

Observando el mapa, se detectan dos agupaciones de unos, en forma horizontal y vertical. Ambos cumplen con la regla de 2^n, siendo $n = 1$. El tamaño más grande posible es de dos. También cumplen con el principio de celdas adyacentes, no diagonales.

b \ a	0	1
0	1	0
1	1	1

Por lo tanto, la función reducida contiene dos minitérminos. Para el grupo horizontal la variable a se desplaza por los estados 0 y 1, por lo tanto no aparecerá en la solución. Para ese mismo grupo la variable b no conmuta, y permanece en su valor lógico 1, por lo cual su minitérmino reducido correspondiente es: b.
Para el grupo vertical la variable b es la que conmuta de 0 a 1, por lo tanto no aparecerá en su minitérmino solución. La variable a no conmuta de su valor lógico 0, por lo que el resultado de este grupo es $\bar{a}$.
La solución final es la suma de los minitérminos minimizados de cada agrupación:

$$f_1(a,b) = \bar{a} + b$$

Para verificar el resultado, comparemos la función original con la función minimizada

a	*b*	*f*	*f1*
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

De donde se concluye que las dos funciones son similares

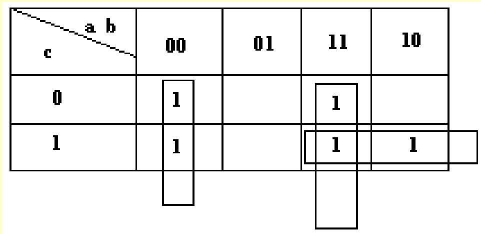
Ejemplo 2.12. Minimizar las siguientes funciones, empleando mapas de Karnaugh.

a) $f(a,b,c)=\sum_{\min}(0,1,5,6,7)$

b) $f(a,b,c,d)=\sum_{\min}(0,2,4,5,9,11,15)$

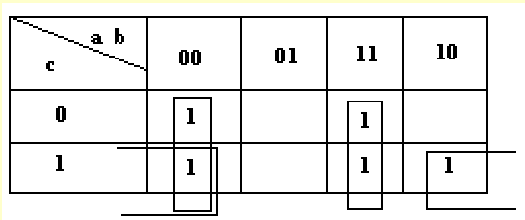
c) $f(a,b,c,d,e)=\sum_{\min}(0,,4,5,6,7,8,12,13,14,16,21,23,30,31)$

a) $f(a,b,c)=\sum_{\min}(0,1,5,6,7)$, primera opción



$$f=\overline{a}\,\overline{b}+ab+ac$$

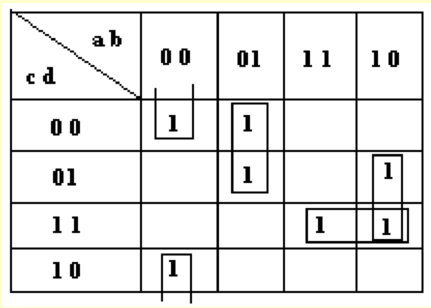
Segunda opción



$$f=\overline{a}\,\overline{b}+ab+\overline{b}c$$

Ambas soluciones son equivalentes. Obsérvese que cada una de ellas contiene tres minitérminos de dos elementos cada uno de ellos.

b) $f(a,b,c,d)=\sum_{\min}(0,2,4,5,9,11,15)$

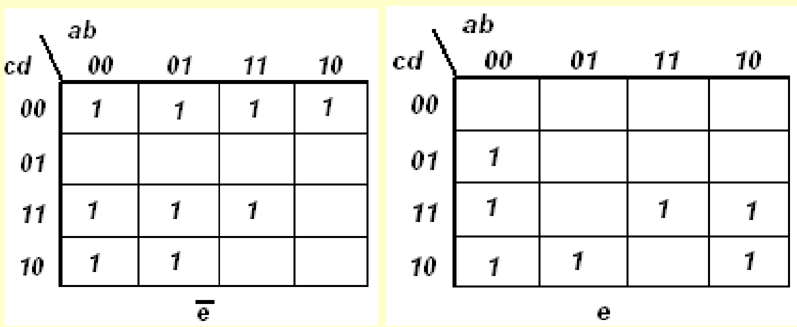


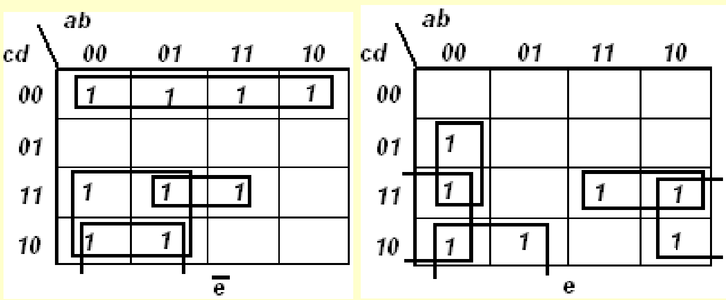
$$f=\overline{a}b\overline{c}+acd+a\overline{b}d+\overline{a}\,\overline{b}\,\overline{d}$$

Queda al lector encontrar una función equivalente, en caso de que exista.

c) $f(a,b,c,d,e)=\sum m(0,,4,5,6,7,8,12,13,14,16,21,23,30,31)$

$f=\bar{a}c\bar{e}+\bar{c}\bar{d}\bar{e}+bcd\bar{e}+\bar{a}c\bar{d}+\bar{b}ce+\bar{a}\bar{b}de+acde$





$$f(a,b,c,d,e)=\bar{a}c\bar{e}+\bar{c}\bar{d}\bar{e}+\bar{a}c\bar{d}+bcd\bar{e}+\bar{b}ce+abce+\bar{a}\bar{b}de$$

d) $f(a,b,c,d)=(\bar{a}+b+\bar{c}+d)(a+b+c+\bar{d})(\bar{a}+\bar{b}+\bar{c}+d)(a+\bar{b}+\bar{c}+d)(\bar{a}+\bar{b}+c+\bar{d})(a+b+c+d)$

De la expresión se obtiene que: $f(a,,b,c,d)=\prod M(0,1,6,10,13,14)$

cd \ ab	00	01	11	10
00	0			
01	0		0	
11				
10		0	0	0

Por lo tanto, la función en su mínima expresión será:

$$f(a,b,c,d)=(a+b+c)(\bar{b}+\bar{c}+d)(\bar{a}+\bar{c}+d)(\bar{a}+\bar{b}+c+\bar{d})$$

e) $f(a,b,c,d)=\sum m(0,1,2,4,5)+d(3,7,9,12)$

cd / ab	00	01	11	10
00	1	1	X	
01	1	1		X
11	X	X		
10	1			

$$f(a,b,c,d)=\bar{a}\bar{c}+\bar{a}\bar{b}$$

f) $f(a,b,c,d)=\prod M(0,1,3,6,11)D(4,12,14,15)$

cd / ab	00	01	11	10
00	0	X	X	
01	0			
11	0		X	0
10		0	X	

$$f(a,b,c,d)=(a+b+c)(a+b+\bar{d})(\bar{b}+d)(\bar{a}+\bar{c}+\bar{d})$$

2.7 Compuertas Lógicas

Las operaciones lógicas contenidas en las funciones lógicas pueden ser implementadas mediante compuertas lógicas digitales. Todos los elementos y funciones lógicas establecidas en la teoría del Álgebra Boolena se encuentran disponibles en circuitos integrados (IC).

Los IC se clasifican de acuerdo a varios criterios: forma en que se montan en una aplicación; Tecnologías de fabricación; de acuerdo a la complejidad. Con respecto a este último criterio, la clasificación es como sigue:

SSI: Small Scale Integration. Hasta doce compuestas lógicas.

MSI: Medium Scale Integration. Hasta 99 compuertas lógicas.

LSI: Large Scale Integration. Hasta 9999 compuertas lógicas.

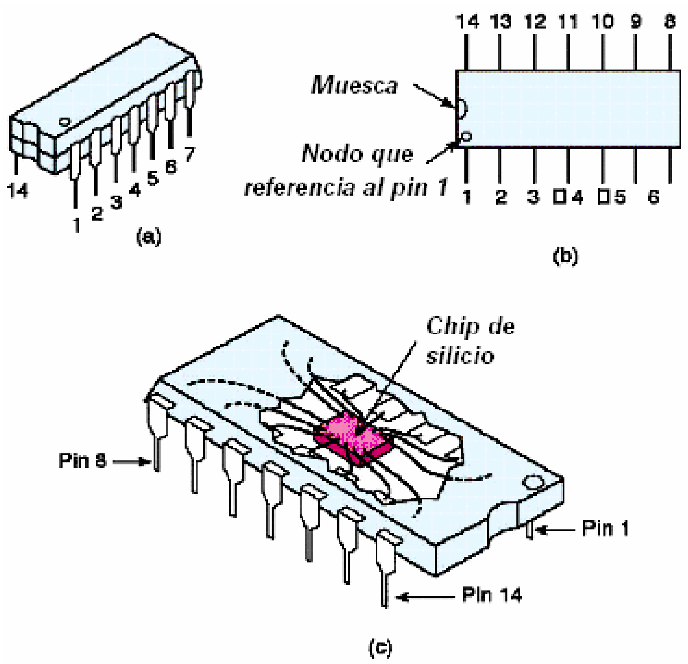
VLSI: Very Large Scale Integration. Hasta 99,999 compuertas lógicas.

ULSI: Ultra Large Scale Integration. Opera bajo memorias demasiado grandes, así como bajo el concepto de microprocesadores.

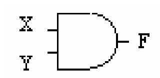
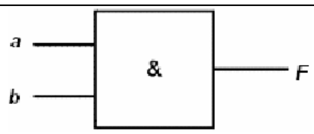
PLD: Programmable logic device. Dispositivos que se programan

Con respecto a las tecnologías, existe un conjunto más o menos amplio, pero las más útiles son las tecnologías TTL y CMOS

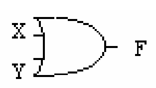
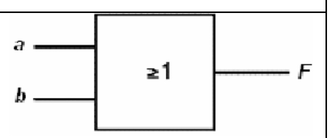
TTL: Transistor – Transistor Logic
CMOS: Complementary Metal Oxide semiconductor



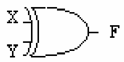
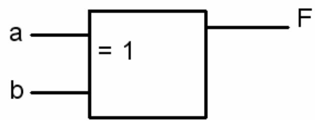
Función AND

<table>
<tr><th>Tabla de Verdad</th><th>Operación Lógica</th><th>Símbolo</th><th>Identificador</th><th>IEE/ANSI 91 1984</th></tr>
<tr><td><table><tr><td>X</td><td>Y</td><td>F</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>F = XY</td><td></td><td>7408</td><td></td></tr>
</table>

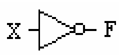
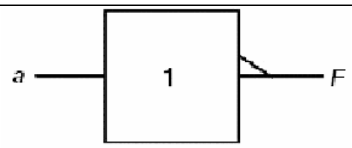
Función OR

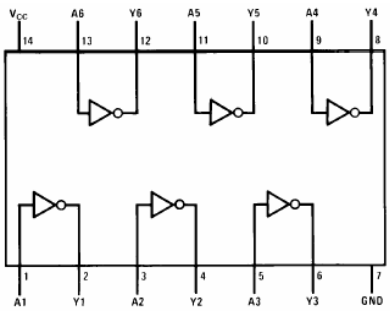
<table>
<tr><th>Tabla de Verdad</th><th>Operación Lógica</th><th>Símbolo</th><th>Identificador</th><th>IEE/ANSI 91 1984</th></tr>
<tr><td><table><tr><td>X</td><td>Y</td><td>F</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>F = X + Y</td><td></td><td>7432</td><td></td></tr>
</table>

Función OR exclusiva (EXOR)

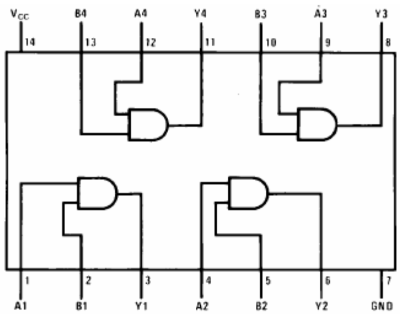
<table>
<tr><th>Tabla de Verdad</th><th>Operación Lógica</th><th>Símbolo</th><th>Identificador</th><th>IEE/ANSI 91 1984</th></tr>
<tr><td><table><tr><td>X</td><td>Y</td><td>F</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>$F = X \oplus Y$
$F = X\overline{Y} + \overline{X}Y + XY$</td><td></td><td>7486</td><td></td></tr>
</table>

Función NOT

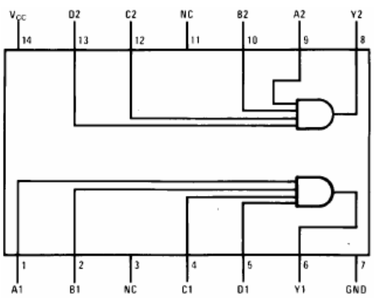
<table>
<tr><th>Tabla de Verdad</th><th>Operación Lógica</th><th>Símbolo</th><th>Identificador</th><th>IEE/ANSI 91 1984</th></tr>
<tr><td><table><tr><td>X</td><td>F</td></tr><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table></td><td>$F = \overline{X}$</td><td></td><td>7404</td><td></td></tr>
</table>



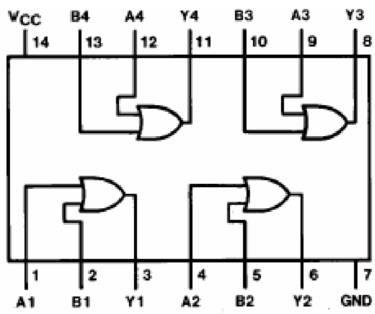
Compuerta 7404



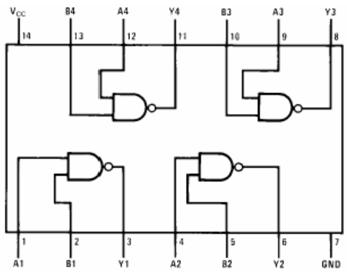
Compuerta 7408



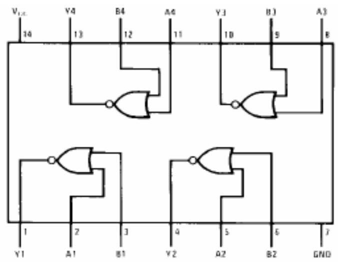
Compuerta 7421

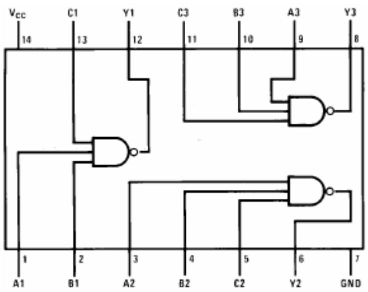


Compuerta 7432

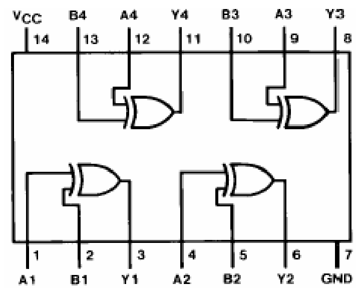


Compuerta 7400


Compuerta 7402

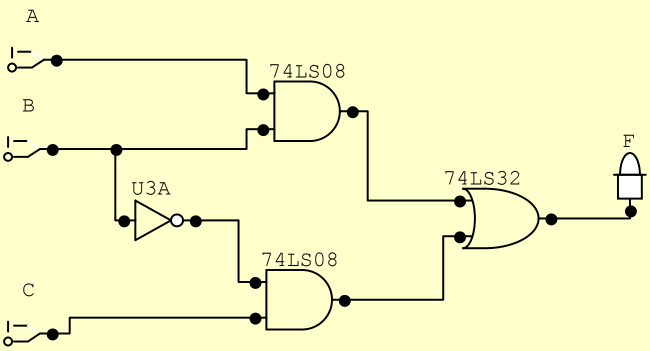


Compuerta 7410



Compuerta 7486

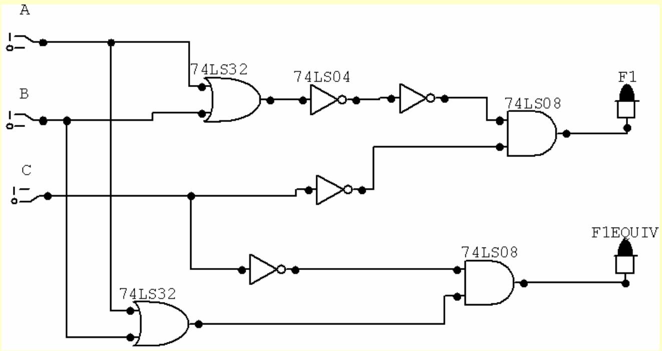
Ejemplo 2.13. Sea la función lógica $F = ab + \bar{b}c$, construir y simular su circuito electrónico con las compuertas correspondientes.



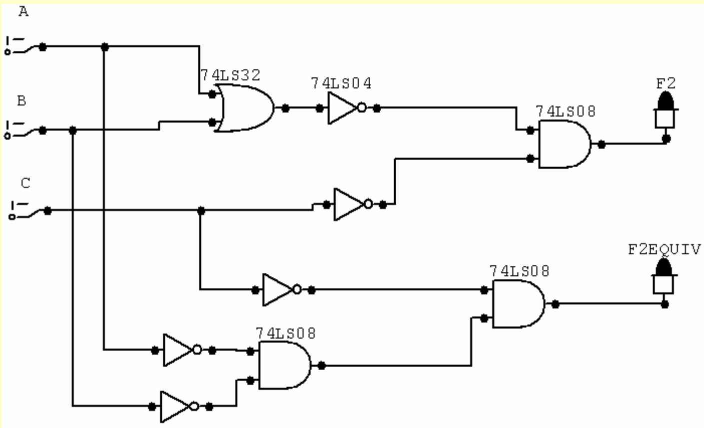
Función lógica F =a*b +b'c

Ejemplo 2.14. Simular las funciones siguientes, tanto en su forma original como en su forma equivalente. Hacer un comparativo de ambas.

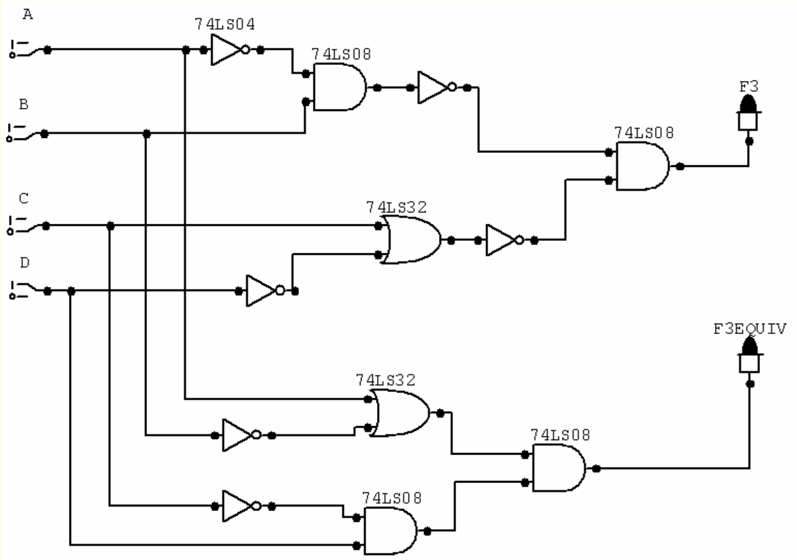
a) $F_1 = \overline{\overline{(A+B)}}\,\overline{C} = (A+B)\overline{C}$; $F_1 = \overline{\overline{(A+B)}} \bullet \overline{C} = (A+B)\overline{C}$



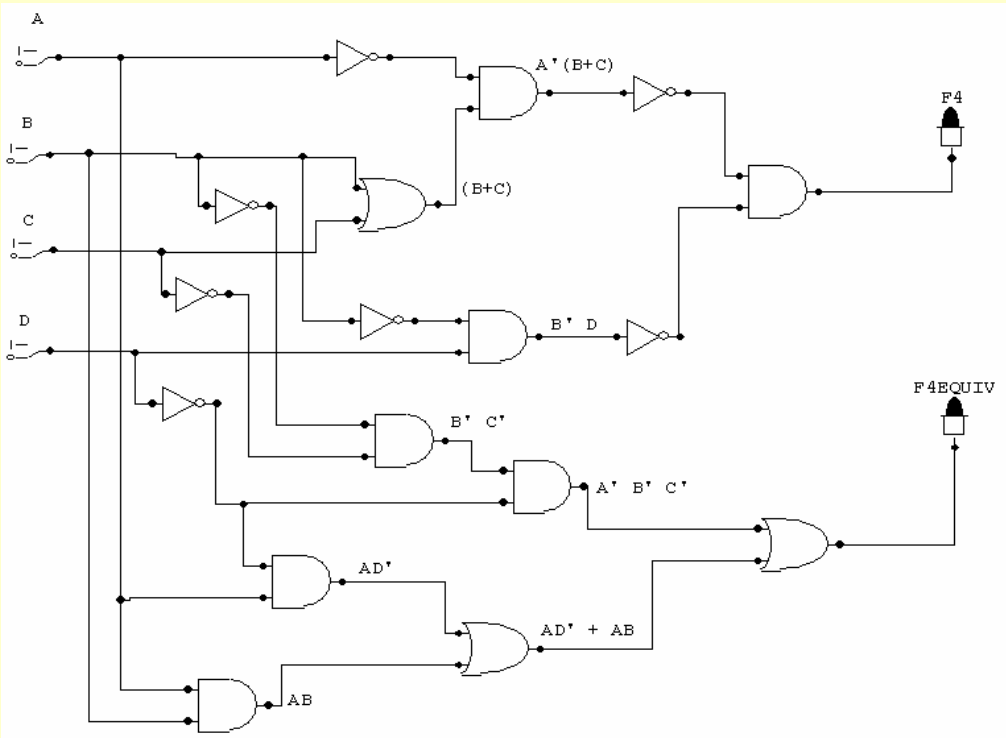
b) $F_2 = \overline{(A+B)} \bullet \overline{C} = \overline{A} \bullet \overline{B} \bullet \overline{C}$



c) $F_3 = \overline{\overline{A}B} \bullet \overline{\mathrm{C} + \overline{\mathrm{D}}} = (\overline{\overline{A}} + \overline{B})(\overline{C} \bullet \overline{\overline{D}}) = (A + \overline{B})(\overline{C} \bullet D)$



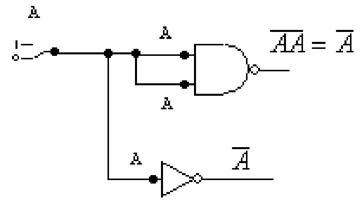
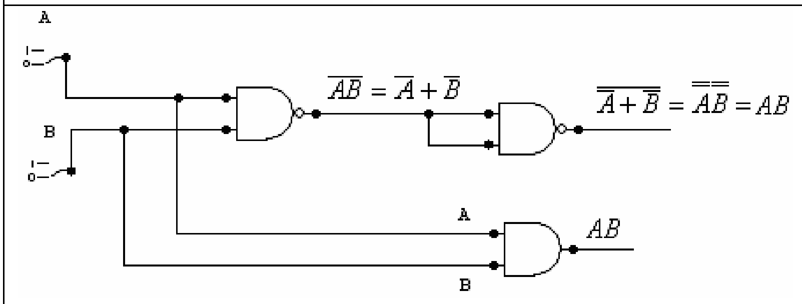
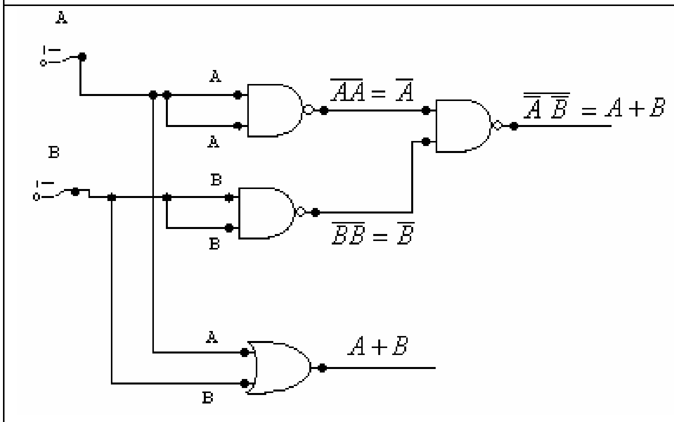
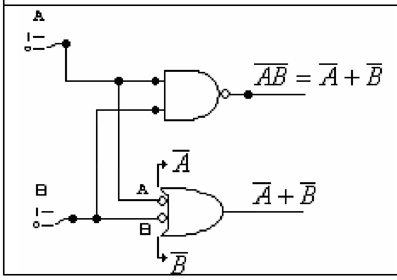
d) $F_4 = \overline{\overline{A}(B + C)} \bullet \overline{\overline{B}D} = AB + A\overline{D} + \overline{B} \bullet \overline{C} \bullet \overline{D}$

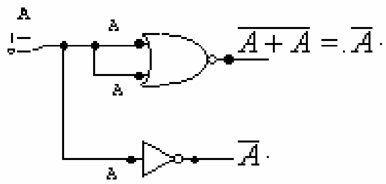
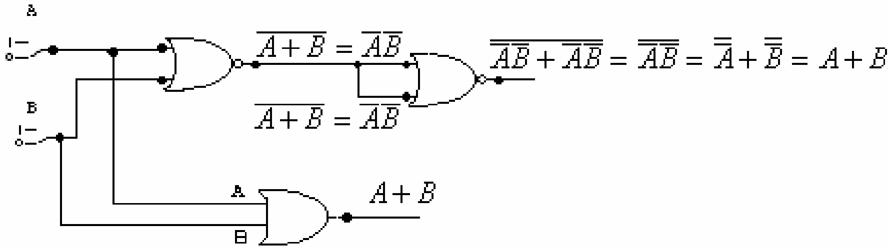
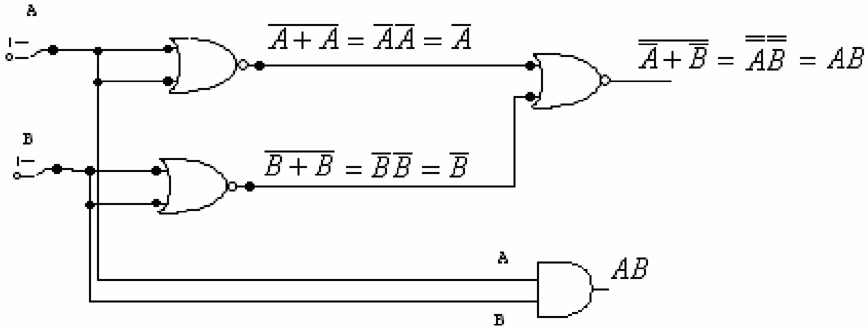
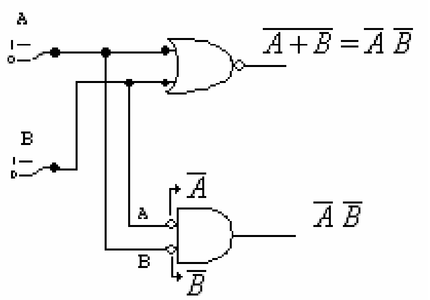


2.8 Compuertas Universales

Se dice que las compuertas NAND y NOR son universales porque cualquier sistema digital puede implementarse con ellas.
A continuación se muestran los arreglos correspondientes para generar las compuertas NOR, AND y OR, empleando la compuerta universal NAND.

Compuertas NAND.
Se dice que es universal porque cualquier función lógica booleana puede ser implementada con ella. Esto se logra debido a que los operandos básicos, AND, OR y NOT tienen pueden ser implementados empleando compuertas NAND, exclusivamente, más no necesariamente. Para comprobar esto, será necesario hacer referencia a las siguientes figuras y manipulando las leyes de DeMorgan.

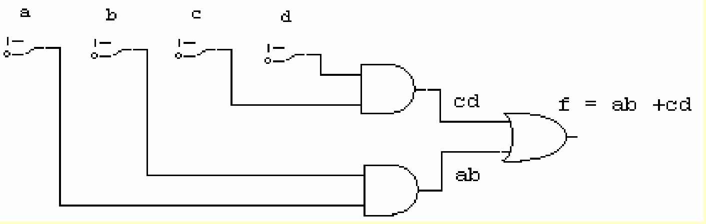
<table>
<tr><td>
</td><td>Compuerta NOT empleando NAND</td></tr>
<tr><td>
</td><td>Compuerta AND empleando NAND</td></tr>
<tr><td>
</td><td>Compuerta OR empleando NAND</td></tr>
<tr><td>
</td><td>Símbolos gráficos empleados para representar a la compuerta NAND</td></tr>
</table>

	Función NOT empleando compuertas NOR
	Función OR empleando compuertas NOR
	Función AND empleando compuertas NOR
	Símbolos gráficos empleados para representar a la compuerta NOR

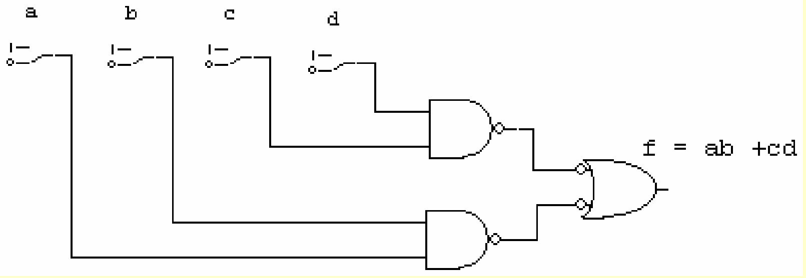
Ejemplo 2.14

Expresar la función lógica $f = ab + cd$ empleando compuertas NAND, exclusivamente.

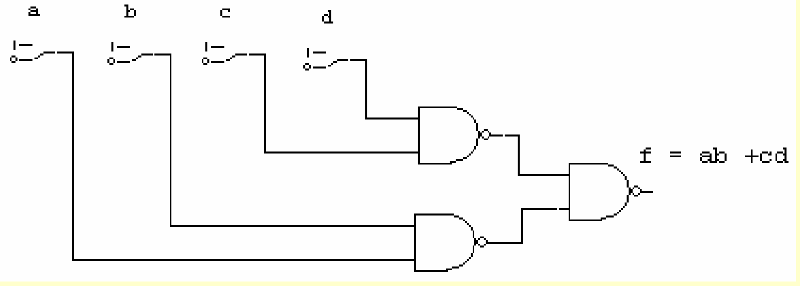
Paso 1. Función f sin compuertas NAND



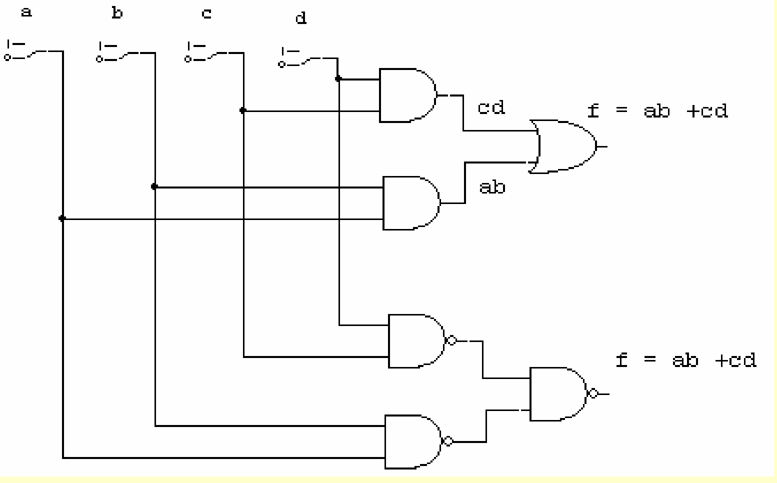
Paso 2. Función f con compuertas NAND y símbolo equivalente



Paso 3. Función f con compuertas NAND, exclusivamente.



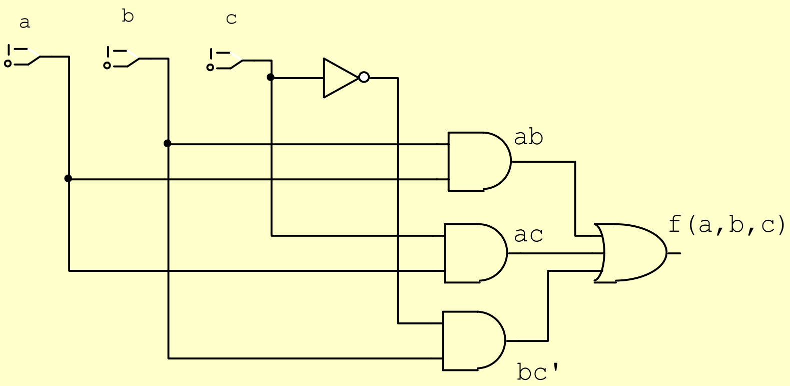
Por lo tanto, los dos circuitos siguientes son equivalentes



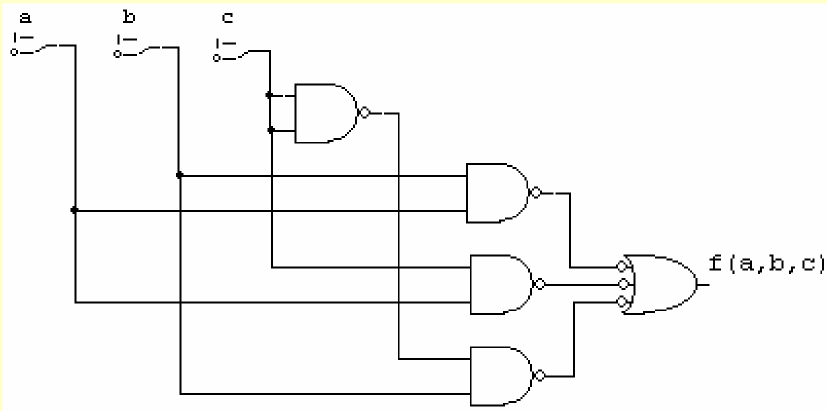
Ejemplo 2.15. Diseñar el circuito equivalente para la función $f(a,b,c) = ab + b\bar{c} + ac$

a) Empleando sólo compuertas NAND
b) Empleando sólo compuertas NOR.

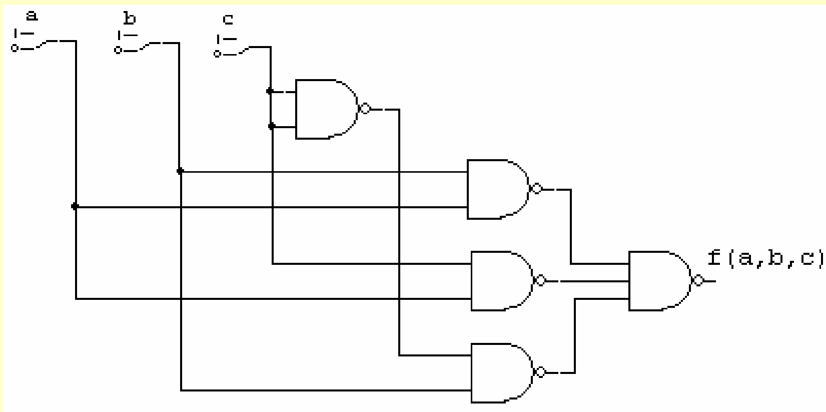
a) Primer paso



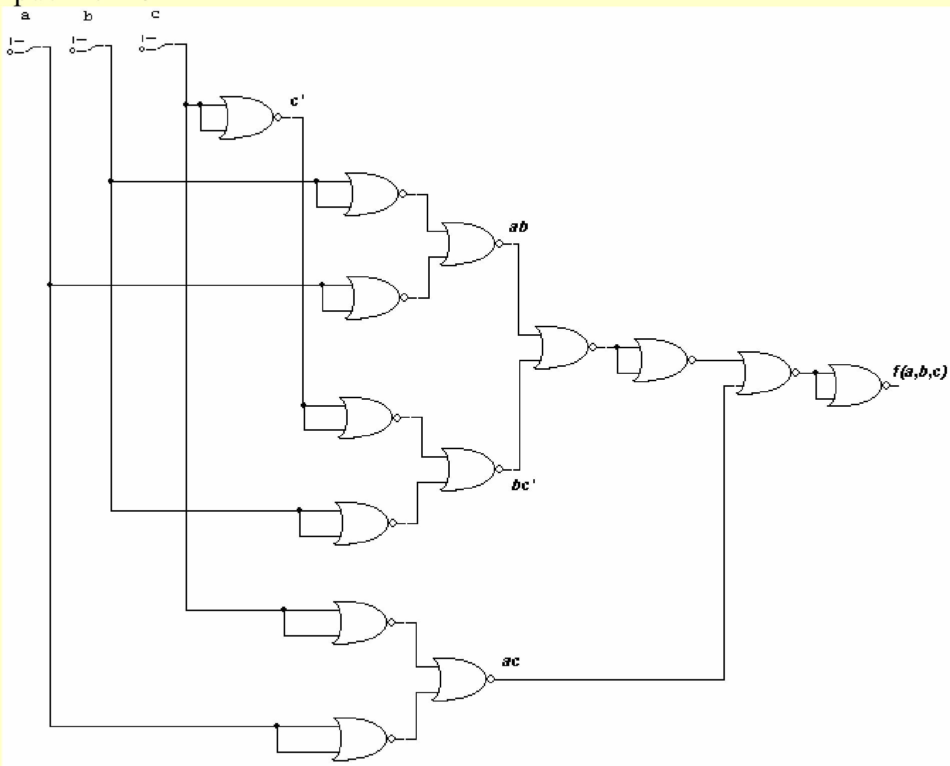
a) Segundo paso.



c) Tercer paso.

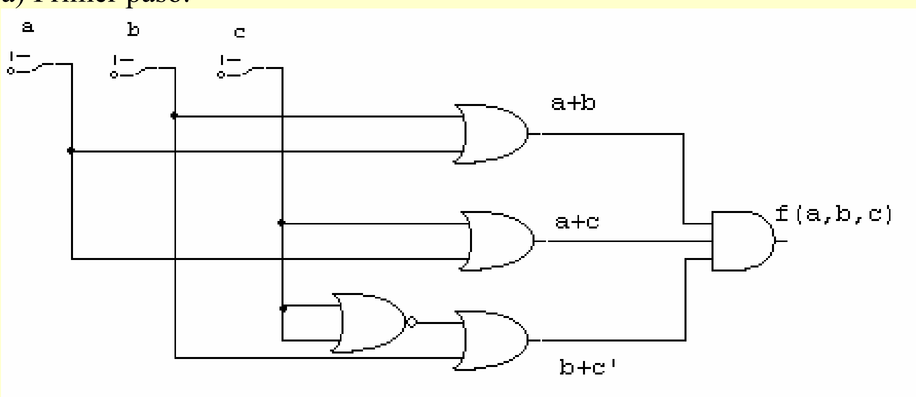


Ejemplo 2.16 Diseñar el circuito equivalente para la función $f(a,b,c) = ab + b\bar{c} + ac$ empleando sólo compuertas NOR.
Con compuertas NOR

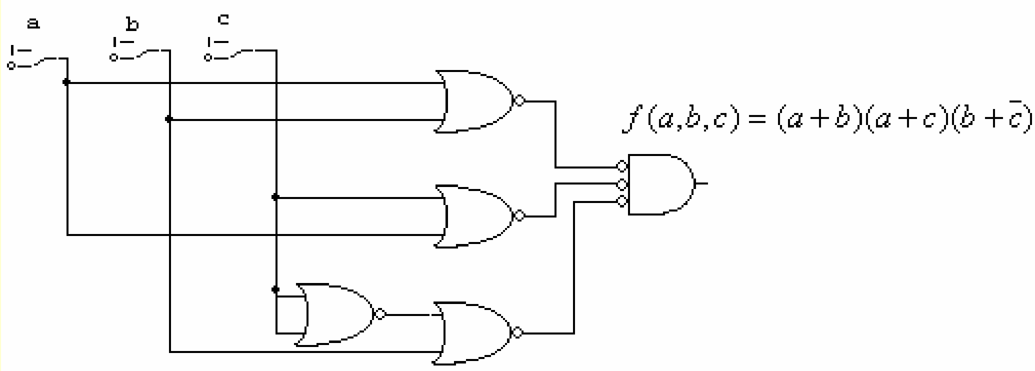


Ejemplo 2.17. Diseñar el circuito equivalente para la función $f(a,b,c) = (a+b)(a+c)(b+\bar{c})$, empleando sólo compuertas NOR

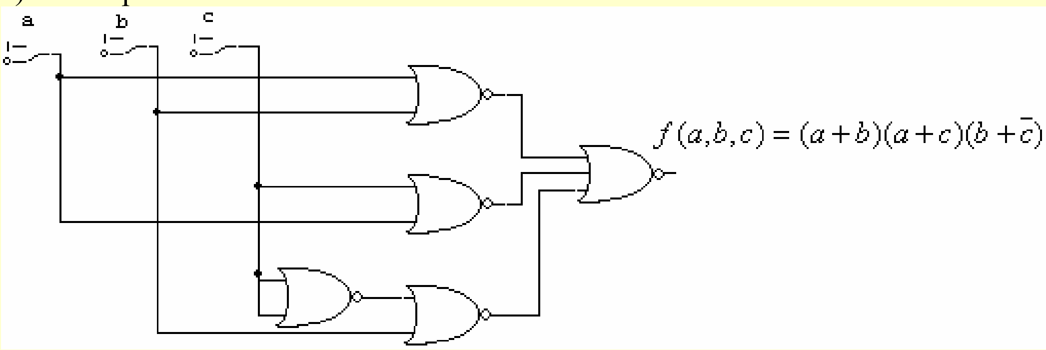
a) Primer paso.



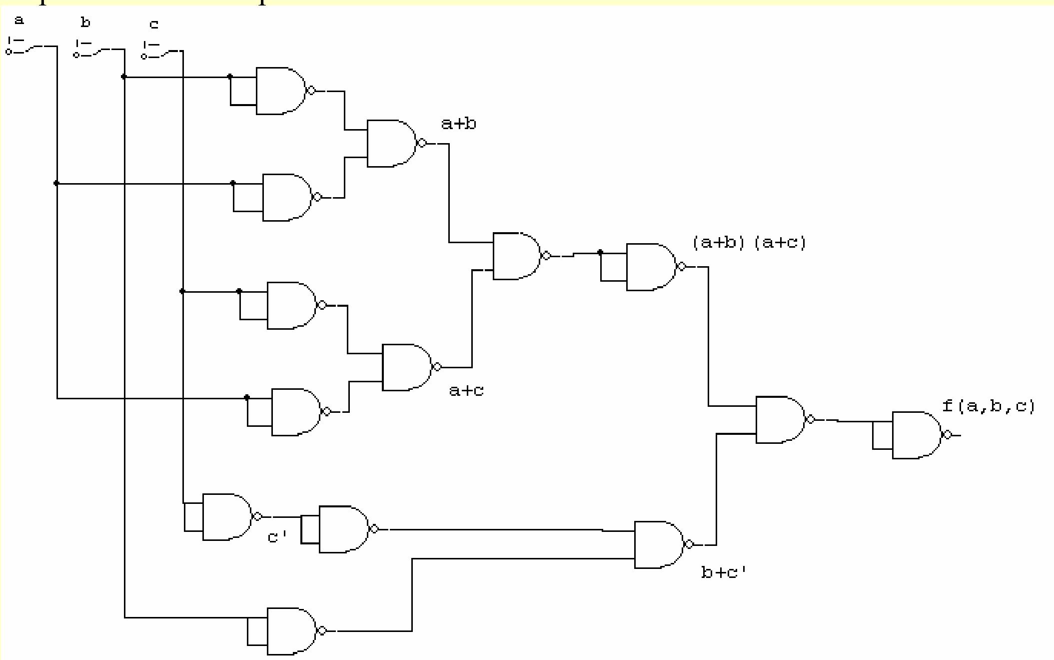
a) Segundo paso.



a) Tercer paso.



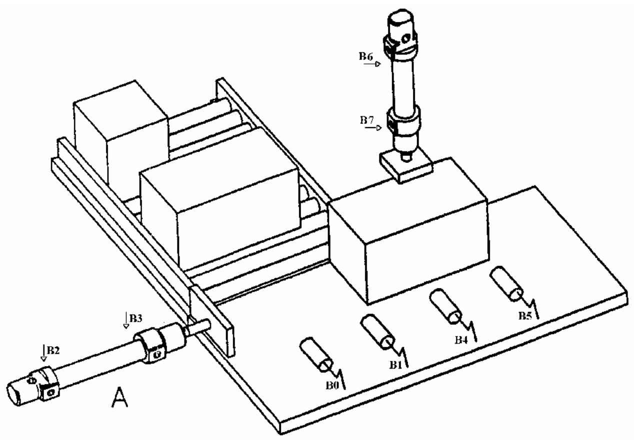
Ejemplo 2.18. Diseñar el circuito equivalente para la función $f(a,b,c) = (a+b)(a+c)(b+\bar{c})$, empleando sólo compuertas NAND



2.9 Ejemplos de aplicaciones.

Ejemplo 1 de aplicación.

Antes del inicio de cada turno, un operador debe de revisar en modo automático la buena operación de un proceso de transporte y estampado de cajas con producto terminado. Ver figura siguiente.



La rutina que debe de cumplirse es la siguiente:

El avance del cilindro A se debe de dar si B0, B1 y B2 están activos. El retroceso de A se dará cuando se activen B3, B4 y B5. Esta misma condición se dará para el avance del cilindro B. El retroceso de B se dará cuando estén activos B4, B5 y B7.

a) Obtenga la función (es) para la operación de los cilindros.

b) Diseñe el circuito electrónico solución para cada cilindro, a partir de la función lógica mínima.

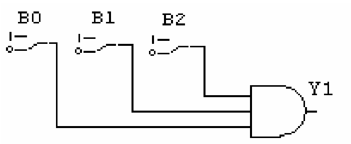
De acuerdo a las especificaciones de operación, a cada desplazamiento de cilindro le corresponde su propia función lógica, por lo tanto serán cuatro de ellas, las cuales se definirán de la siguiente manera:

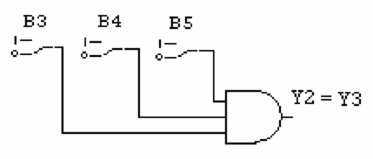
Y1 = avance del cilindro A.
Y2 = retroceso del cilindro A.
Y3 = avance del cilindro B.
Y4 = retroceso del cilindro B.

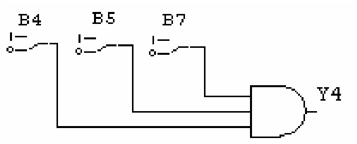
Así, tanto para el avance como para el retroceso, todas las funciones lógicas son en operando AND, de tres variables cada una de ellas. Por otro lado, se tiene que el retroceso de A y el avance de B son simultáneos, por lo cual ambas funciones serán las mismas. El resultado final es como se muestra a continuación.

$$Y_1 = B_0 \bullet B_1 \bullet B_2$$
$$Y_2 = B_3 \bullet B_4 \bullet B_5$$
$$Y_3 = B_3 \bullet B_4 \bullet B_5$$
$$Y_4 = B_4 \bullet B_5 \bullet B_7$$

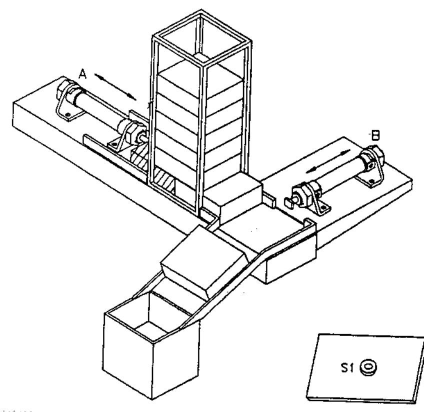
b) Los circuitos electrónicos se obtienen de manera inmediata, a partir del inciso anterior. No es necesario realizar ni una operación algebraica, ya que sólo contienen un solo minitérmino, cada una de ellas.

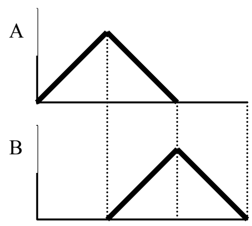






Ejemplo 2 de aplicación. Se cuenta con un sistema de empaque, el cual consiste de dos cilindros neumáticos de doble efecto, A y B, de empuje y llenado, respectivamente. Así mismo se cuenta con un alimentador de cajas, y un receptor de las mismas. El modo de operación del sistema es tal como se muestra en el diagrama espacio – fase. Ver figura siguiente





Las condiciones de arranque serán las siguientes: El cilindro A iniciará su movimiento de avance sólo si primero se garantiza que tanto A como B se encuentran retraídos, además de que exista presencia de caja en el alimentador, lo cual queda establecido por el sensor *Sc*.

Sean a_0, a_1, b_0, b_1 los sensores de detección de inicio y final de carrera de los cilindros A y B, respectivamente. Según lo estipulado con anterioridad, la condición de arranque se dará mediante Y1, el avance de A, al aplicar el operando lógico AND entre los sensores que garantizan la posición inicial de ambos cilindros y el sensor de presencia de caja.

$$Y_1 = S_c a_0 b_0$$

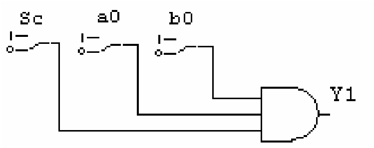
Así, entonces, de acuerdo a Y1, en caso de no tener existencia de cajas, el cilindro no iniciará su desplazamiento de avance.
Posteriormente, y de acuerdo al diagrama de espacio fase, una vez que el cilindro A alcanza su carrera final deberán de suceder dos eventos a la vez: retroceso inmediato de A y avance de B. Esto implica que exista la combinación AND entre los sensores final de carrera de A e inicio de B, por lo tanto, la función lógica para ambos será:

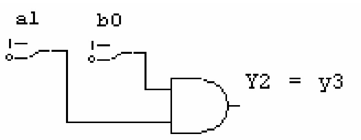
$$Y_2 = Y_3 = a_1 b_0$$

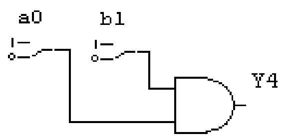
El retroceso de B se logrará una vez que se tenga garantizado que el cilindro A haya regresado a su posición de origen. Mientras tanto el cilindro B tendría que esperar en reposo extendido, por lo cual se concluye que para Y4 se tenga:

$$Y_4 = a_0 b_1$$

Los circuitos electrónicos par implantar cada una de las funciones son los siguientes:

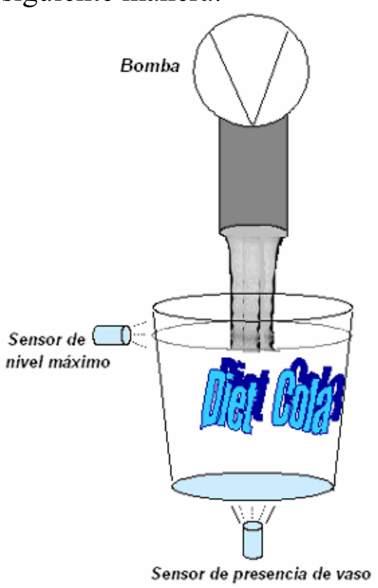






Ejemplo de aplicación 3.

Se tiene una pequeña máquina dispensadora de refrescos en vaso la cual opera de la siguiente manera:



Una bomba eléctrica sirve gaseosa siempre y cuando un sensor de presencia de vaso esté activado. Se apagará cuando se active el sensor de nivel máximo ó que el sensor térmico de la bomba lo indique, o que bien la desconectemos mediante un botón de emergencia. Deberá de existir indicaciones luminosas que muestren que la bomba está sirviendo refresco. Así mismo, deberá de existir una señal adicional que indique existencia de problemas de temperatura en la bomba, a través del sensor térmico. Esta indicación es fundamental, ya que exceso de temperatura podría causar daños al motor.
Por último, existirá un botón de encendido/apagado, y por operación obvia, mientras éste se encuentre en estado OFF toda la funcionalidad quedará inhibida.

a) Determine las funciones lógicas para las señales luminosas de “máquina sirviendo”; “problemas térmicos”; y “máquina encendida”.
b) Diseñar los circuitos electrónicos del inciso anterior.

Solución.
En primera instancia, se realiza una recopilación de las señales que intervienen en el proceso, tanto en forma cuantitativa como cualitativa. De manera imprescindible, en una aplicación existen señales de entrada y de salida. De manera complementaria, pueden existir señales adicionales auxiliares utilizadas principalmente como elementos de apoyo en la estructura lógica de alguna función aplicada a algún elemento o proceso a controlar. Se puede decir que las señales de entrada y de salida se asocian directamente a elementos físicos de campo, sensores y actuadores/cargas, mientras que las señales auxiliares están más relacionadas con variables internas auxiliares programadas en el mismo circuito de control, diseñadas ex profeso para la construcción de lógicas complejas.

De esta manera, analizando la redacción del ejercicio se determina la existencia de cuatro señales de salida, que a saber son:

Foco de alarma: A_L
Foco de máquina sirviendo: M_S
Foco de máquina encendida: M_E
Bomba: B

Las señales de entrada serían:

Sensor de nivel máximo: N_M
Sensor de presencia de botella: S_P
Sensor térmico: S_T
Botón de emergencia: B_E
Botón de encendido: O_N

a) En la redacción se establece que debe de existir una lámpara asociado al estado de servicio de la bomba, esto significa que la operación lógica de ambos será la misma.
Así, el servicio de refresco se realizará cuando se valide es sensor de presencia de botella en combinación lógica AND con el NOR de todas las señales que deshabilitan al estado de servicio (máquina sirviendo). El botón de ON siempre deberá de estar presente.

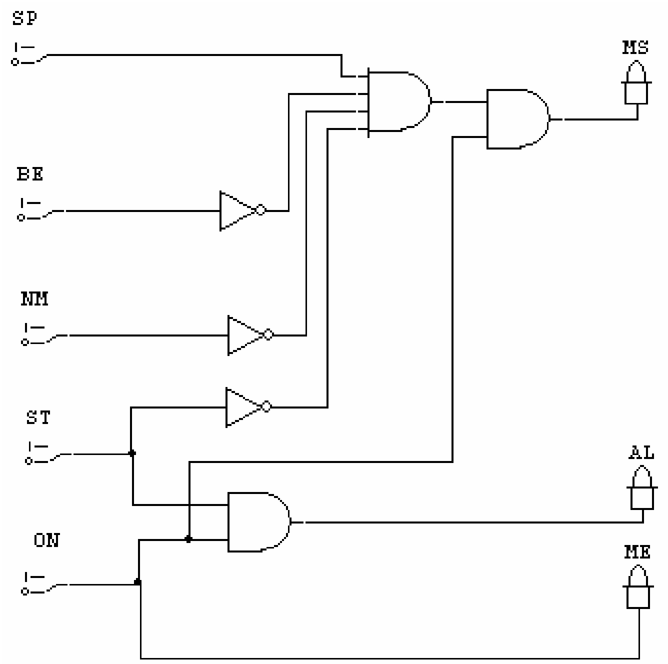
$$B = M_S = S_P(\overline{N_M + S_T + B_E})O_N$$

El encendido de las lámpara de alarma y de máquina operando se lleva a cabo de manera directa de acuerdo a las condiciones del sensor térmico y del botón de arranque, resepectivamente.

$$A_L = S_T O_N$$
$$M_E = O_N$$

c) En el caso del circuito de la bomba se puede implementar tal y como lo indica el inciso anterior o bien se puede aplicar DeMorgan para obtener una expresión en minitérminos.

$$B = M_S = S_P \overline{N_M}\,\overline{S_T}\,\overline{B_E} O_N$$

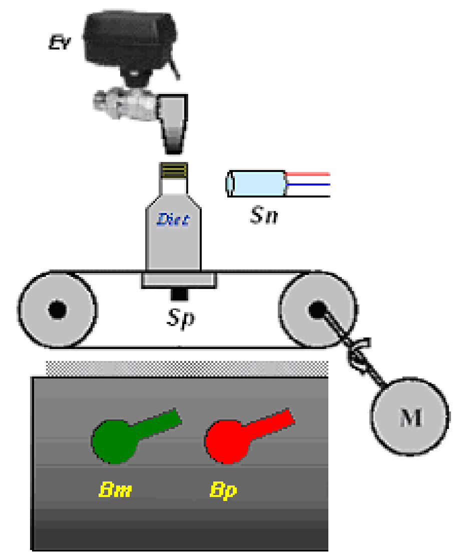


Circuitos para el encendido de las lámparas aplicando DeMorgan para la bomba.

Ejemplo de aplicación 4.

Diseñar el circuito electrónico que controle el llenado de botellas de refresco, hasta un cierto nivel, de acuerdo a las siguientes condiciones de trabajo:

- Se cuenta con un botón de arranque (marcha), *Bm*, el cual genera la señal de arranque para el motor *M* del sistema de transporte.
- Cuando el sensor de presencia *Sp* detecte una botella el motor *M* se detendrá e inmediatamente comenzará el proceso de llenado, mediante una válvula de servicio controlada por una bobina electomagnética (electroválvula), *Ev*.
- El control de nivel se hace con un detector (*Sn*) colocado a una altura equivalente a la posición superior de la botella. Cuando este se active, la electroválvula *Ev* deberá de detenerse.
- Un instante *T* relativamente pequeño posterior a la desactivación de la electroválvula deberá de volver a prender el motor *M*. Este tiempo debe ser lo suficientemente grande como para garantizar que los diferentes elementos dinámicos hayan llegado de una forma estable al reposo antes de imprimir de nuevo una inercia a la botella.
- En caso de accidente o de necesidad de detener la marcha del motor, se contará con un botón de paro de emergencia, *Bp*. Cuando este se activa, todas las cargas se deshabilitan. El re-arranque podrá darse de nuevo mediante *Bm*.



Solución. Este ejercicio involucra el manejo de una señal variable interna de proceso, el Temporizador, la cual se puede implementar ya sea vía Hardware o Software. En el caso de seleccionar la primera opción, en el mercado se cuenta con relevadores temporizados, los cuales poseen un selector de tiempo, *preset,* ajustable de acuerdo a las necesidades. Una vez que el dispositivo (relevador) recibe una orden externa de trabajo, éste ejecutará la acción durante el tiempo *T* preestablecid, el cual una vez que se cumple el relevador dejará de enviar su señal de actuación.
En general los temporizadores pueden ser analizados y diseñados con los principio de los sistemas digitales, bajo el principio de sistemas secuenaciales que serán revisados en el capítulo 4 de este texto. Por lo pronto, y para efecto de plantear una solución al ejercicio, se considerará que al temporizador como una variable existente propiamente en el proceso, por lo cual solamente haremos uso del mismo en el diseño de la solución sin entrar en sus detalles específicos de operación.

Dispositivo	**Símbolo**
Botón de marcha.	*Bm*
Botón de paro de emergencia	*Bp*
Sensor de nivel	*Sn*
Sensor de presencia	*Sp*
Electroválvula	*Ev*
Motor	*Ev*
Temporizador	*T*

Por efectos de simplicidad de diseño, y sin pérdida de generalización, se considerará que los botones de marcha y paro son de giro monoestable (enclavados), esto es, una vez que se establecen en un estado operativo, permanecen allí mientras no exista fuerza muscular externa que los obliga a conmutar, contrario a lo que realiza un botón pulsador, *pushboton,* que se restablece a su estado original una vez que desaparece la fuerza que lo empuja.

$$E_v = S_P \overline{S_n}\,\overline{B_p} \quad M = (B\overline{S_P} + T)B_m \overline{B_P}$$

Ejercicios

1.- El equivalente de la función lógica $(a+\bar{b})(\bar{a}+c)(\bar{b}+c)$ es:

a) $(a+\bar{b})(\bar{a}+c)$

b) $(\bar{a}+\bar{b})(\bar{a}+c)$

c) $(a+\bar{b})(\bar{a}+\bar{c})$

d) $(a+\bar{b})$

2.- El equivalente de la función lógica $a(b+c)+\bar{a}b$ es:

a) $\bar{b}(\bar{a}+\bar{c})+\bar{a}b$

b) $\bar{b}(\bar{a}+\bar{c})$

c) $\bar{b}(\bar{a}+c)$

d) $\bar{b}(a+\bar{c})$

3.- Según el álgebra Booleana, la relación $ab+\bar{a}c$ es igual a:

a) $ab+a\bar{c}+bc$

b) $\bar{a}b+\bar{a}c+bc$

c) $ab+\bar{a}c+bc$

d) $ab+\bar{a}c+\bar{b}c$

4.- La función lógica $f=ab+a\bar{b}c$ tiene como equivalente a:

a) $f=ab+a\bar{c}$

b) $f=ab+ac$

c) $f=a+ac$

d) $f=0$

5.- El equivalente en maxitérminos de la función $f(a,b,c)=\sum m(0,1,4,6,7)$ es:

a) $(a+\bar{b}+c)(a+\bar{b}+\bar{c})(\bar{a}+b+c)$

b) $(a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+b+\bar{c})$

c) $\bar{a}\bar{b}c+\bar{a}bc+a\bar{b}c$

d) $(a+\bar{b}+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+b+\bar{c})$

6.- El equivalente en maxitérminos de la función $f(a,b,c)=\sum m(0,2,4,5,7)$ es:

a) $(a+\bar{b}+c)(a+\bar{b}+\bar{c})(\bar{a}+b+c)$

b) $\bar{a}\bar{b}c+\bar{a}bc+a\bar{b}c$

c) $(a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+\bar{b}+c)$

d) $(\bar{a}+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+\bar{b}+c)$

7.- El equivalente de la función lógica $f(a,b,c) = a(a+\bar{c})$ es:

a) $f(a,b,c) = \prod M(0,1,2,3)$

b) $f(a,b,c) = \prod M(0,1,2,8)$

c) $f(a,b,c) = \prod M(0,2,3)$

d) $f(a,b,c) = \sum M(0,1,2,3)$

8.- El equivalente de la función lógica $f(a,b,c) = ab + a\bar{c} + \bar{a}c$ es:

a) $f(a,b,c) = \sum m(1,3,4,6,7)$

b) $f(a,b,c) = \sum m(1,2,4,6,7)$

c) $f(a,b,c) = \sum m(1,3,4,6)$

d) $f(a,b,c) = \prod M(1,3,4,6,7)$

9.-Determine el equivalente en minitérminos de la función lógica
$f(a,b,c,d) = (a+bc)(a+\bar{b}+c+\bar{d})(a+b+d)(a+\bar{b}+\bar{c}+\bar{d})$

10.- Obtenga la función correspondiente a partir del siguiente mapa de Karnaugh.

cd\ab	00	01	11	10
00				
01	1	1		
11	1	1		1
10				1

e

cd\ab	00	01	11	10
00				1
01		1	1	1
11		1	1	1
10				1

$\bar{e}$

11.- Determinar la función F de acuerdo al mapa de Karnaugh que se muestra.

cd\ab	00	01	11	10
00			X	
01	1			
11	1	X		1
10				1

e

cd\ab	00	01	11	10
00	1			X
01				
11	1	X	1	
10				1

$\bar{e}$

12.-Determinar la función F de acuerdo al mapa de Karnaugh que se muestra.

cd\ab	00	01	11	10
00			X	
01	1			
11	1	X		1
10				1

e

cd\ab	00	01	11	10
00	1			X
01		1		
11	1		1	
10				1

$\bar{e}$

13.- El mapa de Karnaugh que se muestra en la siguiente imagen representa a la función:

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

a) $f = (a + c)(b + c)(\bar{b} + \bar{c})$

b) $f = (a + \bar{c})(b + c)(\bar{b} + \bar{c} + d)$

c) $f = (a + c)(b + c)(\bar{b} + \bar{c} + d)$

14.- Utilice Mapas de Karnaugh para llevar a su mínima expresión a las funciones siguientes:

a) $f = a\bar{b}c + ab\bar{d} + abc\bar{d} + bcd$

b) $f(a,b,c,d) = \sum m(0,1,2,4,5,6,12,13,14)$

c) $f(a,b,c,d,e)\sum m(0,4,12,8,1,13,15,16,17,29,31,23)$

d) $f(a,b,c,d,e) = a\bar{b}\bar{c}\bar{d}\bar{e} + \bar{a}\bar{b}c\bar{d}e + \bar{a}b\bar{c}de + \bar{a}bcde + \bar{a}\bar{b}\bar{c}\bar{d}\bar{e} + \bar{a}\bar{b}c\bar{d}\bar{e} + \bar{a}bcd\bar{e} + abc\bar{d}\bar{e}$

e) $f(a,b,c,d,e) = a\bar{b}\bar{c}\bar{d}e + a\bar{b}c\bar{d}e + \bar{a}b\bar{c}de + \bar{a}bcde + a\bar{b}\bar{c}\bar{d}\bar{e} + a\bar{b}c\bar{d}\bar{e} + \bar{a}bcd\bar{e} + \bar{a}bc\bar{d}\bar{e}$

f) $f(a,b,c,d,e) = \bar{a}b\bar{c}\bar{d}e + ab\bar{c}\bar{d}e + \bar{a}b\bar{c}de + \bar{a}bcde + \bar{a}b\bar{c}\bar{d}\bar{e} + ab\bar{c}\bar{d}\bar{e} + \bar{a}bcd\bar{e} + \bar{a}bc\bar{d}\bar{e}$

g) $f(a,b,c,d,e) = a\bar{b}\bar{c}\bar{d}\bar{e} + \bar{a}b\bar{c}de + \bar{a}bcde + a\bar{b}c\bar{d}e + \bar{a}\bar{b}\bar{c}\bar{d}\bar{e} + \bar{a}bcd\bar{e} + \bar{a}bc\bar{d}\bar{e} + a\bar{b}c\bar{d}\bar{e}$

h) $f(a,b,c,d,e) = a\bar{b}\bar{c}\bar{d}e + \bar{a}b\bar{c}de + \bar{a}bcde + a\bar{b}\bar{c}\bar{d}e + \bar{a}\bar{b}c\bar{d}\bar{e} + \bar{a}bcd\bar{e} + \bar{a}bc\bar{d}\bar{e} + a\bar{b}\bar{c}\bar{d}\bar{e}$

i).- $f(a,b,c,d,e) = \sum m(2,3,10,11,14,15,18,21,22,23)$

j) $f(a,b,c,d) = \prod M(0,1,3,6,9,11,14,15)$

k).- $f(a,b,c,d,e) = (a + b + c + \bar{d} + e)(a + b + c + d + \bar{e})(\bar{a} + b + c + \bar{d} + e)..$
$(\bar{a} + \bar{b} + c + \bar{d} + \bar{e})(a + b + \bar{c} + d + e)$

l) $f(a,b,c,d,e) = (\bar{a} + \bar{b} + c + d + e)(a + b + \bar{c} + d + \bar{e})(\bar{a} + b + c + \bar{d} + \bar{e})..$
$(\bar{a} + \bar{b} + c + \bar{d} + e)(a + b + \bar{c} + d + e)$

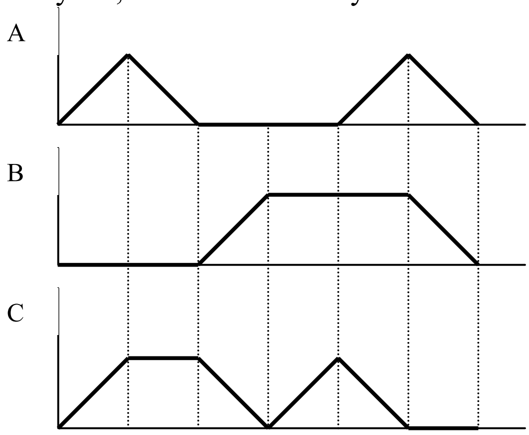
m) $f(a,b,c,d) = (a + b + c + \bar{d})(a + b + \bar{c} + \bar{d})(a + \bar{b} + c + d)(a + \bar{b} + \bar{c} + d)(\bar{a} + b + c + \bar{d})(\bar{a} + b + \bar{c} + \bar{d})..$
$(\bar{a} + \bar{b} + c + d)(\bar{a} + \bar{b} + \bar{c} + d)$

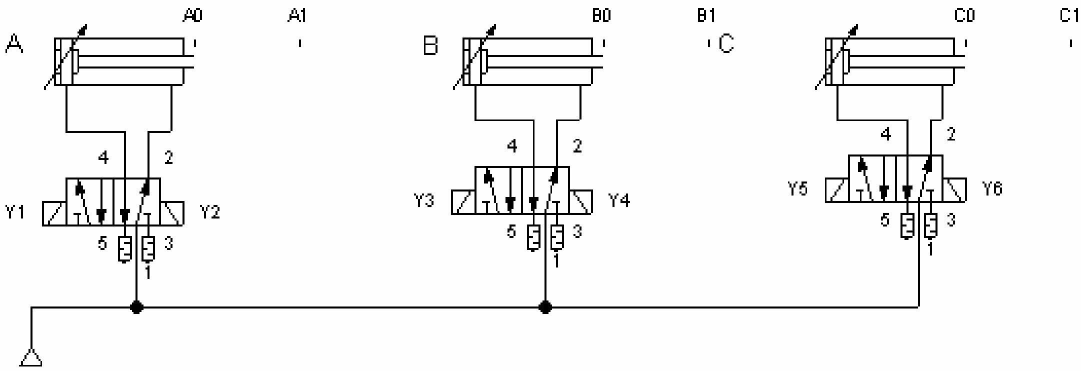
n) $f(a,b,c,d) = (a + c + \bar{d})(a + b + \bar{d})(a + \bar{b} + c + d)(a + \bar{b} + \bar{c} + d)(\bar{a} + b + c + \bar{d})(\bar{a} + b + \bar{c} + \bar{d})$

o) $f(a,b,c,d) = \prod M(0,1,3,6,9,11)D(4,8,14,15)$

15.-Empleando mapas de Karnaugh, simplificar la función $f(a,b,c,d) = \sum m(1,3,4,7,11) + d(5,12,13,14,15)$ y expresar el resultado en maxitérminos.

16. Considere que en una línea de producción se desea instalar una nueva máquina para cumplir con los nuevos objetivos de trabajo. Dicha máquina posee tres cilindros neumáticos, *A*, *B* y *C*, respectivamente, y debe de cumplir con una secuencia específica, de acuerdo a su diagrama de espacio fase. Determinar las funciones lógicas para cada una de las señales de avance y retroceso de cada cilindro.
El inicio de cada ciclo se dará mediante la activación de un botón pulsador Bm;
Y1 = Avance de cilindro A.
Y2 = Retroceso de cilindro A.
Y3 = Avance de cilindro B.
Y4 = Retroceso de cilindro B.
Y5 = Avance de cilindro C.
Y6 = Retroceso de cilindro C.
A0 y A1, detectores inicial y final de carrera del cilindro A.
B0 y B1, detectores inicial y final de carrera del cilindro B.
C0 y C1, detectores inicial y final de carrera del cilindro C.





Bibliografía

Floyd, T., *Digital Fundamentals*, *8th* edition, Prentice Hall, New Jersey, 2003.

Mano, M., *Digital Design*, *Third Edition*, Prentice Hall, Englewood Cliffs, NJ, 2002

Nelson, P., Nagle H., Carroll D., and Irwin J., *Digital Logic Analysis and Design*, Prentice Hall, Englewood Cliffs, NJ, 1995.

Roth, C., *Fundamentals of Logic Design*, *5th* edition, Thomson Brooks/Cole, Belmont CA, 2004.

Tocci J., Widmer S., *Sistemas Digitales, principios y aplicaciones*, Octava edición, Naucalpan de Juárez, México, 2003.

Wakerly F., *Diseño Digital y Prácticas*, Prentice Hall, Naucalpan de Juárez, México, 1992.