UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

"APLICACIÓN DE ALGUNAS TÉCNICAS DE REDUCCIÓN EN CIRCUITOS COMBINACIONALES"

T E S I S

PARA OBTENER EL TITULO DE:

INGENIERO MECÁNICO ELECTRICISTA

ÁREA: ELÉCTRICA – ELECTRÓNICA

P R E S E N T A:

JUAN LUJANO CASTILLO

ASESOR: ING. ELEAZAR MARGARITO PINEDA DÍAZ

MÉXICO 2012

Este trabajo esta dedicado a mis padres que me han dado todo lo esencial de manera justa y en la cantidad correcta aun en las adversidades para ser, estar y crear desde el primer día hasta siempre.

A mis hermanos que al igual que mis padres me han dado la ayuda y el ejemplo en la vida de cómo hacer vereda donde no la hay para luego convertirla en camino real.

A ti Graciela por que se te hubiera gustado verme terminar.

Mi eterno agradecimiento al Ing. Eleazar ya que tuve la fortuna de tenerlo como asesor en este trabajo tan importante para mi, por que se de su enorme sabiduría y de la sencillez con que la comparte, además de su paciencia conmigo en todo el tiempo que me llevo finalmente terminar.

A la Universidad Nacional Autónoma de México que somos todos y en todo espíritu, gracias.

Agradezco a los amigos y compañeros que me dieron palabras de aliento para continuar y no claudicar en los diferentes momentos de este proceso Magda, Santiago, Juan, Mercedes, Diana, David, Vanessa, y además de todo gracias por la bibliografía Liliana. A la familia González Molina por su apoyo también incondicional, gracias Nelly por el apoyo en el diseño.

A los ánimos, al apoyo y a las críticas gracias Nancy, me ayudaron mucho. Y a dos personitas que sirvieron de inspiración para no quedarme sin terminar Erandi y Mikael.

ÍNDICE

INTRODUCCIÓN

Este trabajo trata un conocimiento importante y necesario en la formación del ingeniero, para relacionar su quehacer como facilitador y creador de soluciones ante las diversas necesidades que se le presenten. Este conocimiento al que se hace referencia es el saber del diseño de los circuitos combinacionales, punto clave para un buen diseño de sistemas digitales y siendo ambos elementos los que ayudan a automatizar todo lo que se desee, desde lo más simple hasta lo más complejo.

Pero precisamente una de las dificultades que en principio encuentra el estudiante es el poder relacionar el conocimiento teórico que se le imparte, con el conocimiento práctico y poderlos aplicar en la creación de un trabajo de diseño. El ingeniero puede ayudarse de muchas herramientas ya existentes para hacerlo, aunque por supuesto estas son un camino más largo para llegar a lograr dicha relación exitosa entre ambos aspectos, el teórico y el práctico; por lo que este trabajo de tesis se presenta como una herramienta que puede ayudar al estudiante en su formación y asimilación de los procesos para el diseño de circuitos combinacionales; pero de una manera mas sencilla, es decir la opción de una guía que reducirá y ordenara el proceso del diseño de los circuitos digitales según las necesidades de las tareas que se tengan que realizar.

Así que los objetivos de la tesis son: Caracterizar algunas técnicas de reducción de circuitos que se pueden emplear para tratar de minimizar la cantidad de compuertas empleadas en un diseño digital. Al mismo tiempo, hacer el diseño de algunos sistemas digitales empleando lógica combinacional, para mostrar su posible reducción empleando algunas de las técnicas descritas y facilitarle a los alumnos el aprendizaje de la electrónica digital.

Para lograr dichos objetivos en el capitulo 1 se desarrollaron una serie de generalidades necesarias para poder entender la materia de diseño lógico, es decir, conocer la materia prima con la que trabajan las componentes electrónicas llamadas compuertas lógicas, que son: los números binarios, la conversión de estos a los números que conocemos como decimales y viceversa. Así también para el manejo y el manipuleo de los números binarios se presenta la lógica binaria como elemento para describir por un medio matemático este proceso, además de ayudarnos con otras herramientas matemáticas que fortalecen el manejo general del aspecto binario, como lo son; los teoremas de De Morgan, los postulados de Huntington y el teorema de Shannon. También se introduce el concepto de sistema digital, que ya es finalmente una forma de entender la estructura de un sistema mas complejo pero de manera simple, y la diferencia de uno analógico a uno digital. Por supuesto, en la teoría de los circuitos combinacionales no puede faltar el algebra de Boole, la cual expresa correspondencias de las variables que maneja con las variables de los sistemas digitales, las características de esta teoría, y como esto nos lleva a las operaciones lógicas mas importantes que manejan las dos variables binarias, así como los símbolos eléctricos de las compuertas que representan a dichas operaciones lógicas según las distintas normas mas usadas y reconocidas en el mundo.

En el capítulo 2 los temas abordados son las características de algunas de las técnicas de reducción en los circuitos combinacionales, como son: el algebra de Boole, llamada así en honor del matemático ingles George Boole, quien crea esta matemática para la aplicación a la lógica deductiva, que además tiene como una propiedad derivativa a los minitérminos y maxitérminos de lo cual trata este trabajo, y que posteriormente vemos se usan reiteradamente en otras técnicas de reducción que se desarrollan; como lo son los mapas de Karnaugh, método que se muestra de una manera mas grafica relacionada muy fuertemente con la lógica de conjuntos y la cual tiene un uso viable para la reducción de 2, 3, 4, 5 y 6 variables, ya que para un mayor numero se vuelve un poco impráctico. Se encuentran técnicas también como las de tabulación, que como su nombre lo indica el acomodo de las distintas señales se da en tablas y que esta estrechamente relacionada con la técnica de Quine – Mc Cluskey que se apoya de este método tabular. Se analizan además las salidas múltiples, es decir cuando las variables que se utilizan son usadas para mas de un solo propósito. Por ultimo en el capitulo se trabaja por métodos algorítmicos variados, como lo es la búsqueda dirigida, el iterativo y el iterativo generalizado, que todos estos dejan atrás un cierto aspecto heurístico, y abren paso a soluciones mas optimas y de manera directa.

Se presentan en el capítulo 3 los datos de fabricante como conocimiento indispensable para el diseño de cualquier circuito electrónico, ya que estos contienen a la matrícula, nombre, la escala de integración, de qué tipo de familia son, de los tipos de encapsulados, las corrientes, la potencia disipada, los factores de carga, el tiempo de propagación, inmunidad al ruido, temperatura de trabajo y factor de calidad. Seguido a los datos de fabricante se presenta una guía de diseño de circuitos combinacionales, que consta de una serie de pasos en los que el estudiante tenga un punto de partida, es decir, la necesidad de una tarea y los tipos de tareas a los que el estudiante se enfrenta, como son las tareas a las cuales se les puede satisfacer con múltiples soluciones y de las cuales solo hay una única solución para posteriormente luego convertir esa necesidad a una expresión booleana, ya esta algebra booleana es una matemática que ayuda a la construcción de los diagramas de los circuitos digitales, en donde una tarea expresada en un enunciado común es convertible a la expresión booleana. De esta guía expuesta se trabajan ejemplos de diseño de circuitos que ayudaran al estudiante a observar la aplicación de esta guía, desde el planteamiento de la necesidad, seguido de la conversión de la necesidad a la expresión booleana, hasta lo que es ya el diagrama del circuito combinacional.

En el capitulo 4 se desarrolla la reducción de circuitos combinacionales con ayuda y base de las técnicas de reducción vistas en el capitulo 2. Algunas de estas técnicas son aplicadas a los diseños finales planteados en el capitulo 3. Este capitulo tiene como característica reunir a todos los conocimientos desarrollados a través de todos los capítulos anteriores y que quizás en cantidad de páginas sea el menos amplio, pero cuenta con la jerarquía de mostrar la aplicación total del conocimiento adquirido.

Así, en esta tesis se encuentra distribuido el conocimiento básico del diseño lógico combinacional y con la firme intención de poder llegar a la reducción de los circuitos combinacionales.

CAPITULO 1
GENERALIDADES

1.1. Números binarios

Para iniciar dentro del conocimiento y manejo de las características de las técnicas de reducción de circuitos combinacionales, es necesario conocer el tipo de materia prima y como es manejada por las compuertas lógicas. Esta materia prima a la que se hace referencia son niveles de voltaje que se representan por un nivel "alto" y un nivel "bajo". Los dos estados de los que hablamos generalmente son de encendido y apagado, para cuestiones de diseño se pueden representar por un "1" lógico en el caso de hablar de un nivel alto o de encendido y un "0" lógico en el caso de que el nivel al que se haga referencia sea un nivel bajo o de apagado, por lo que al diseñar los circuitos estaremos utilizando un sistema numérico de dos estados llamado *binario*. A este sistema numérico también se le conoce como un sistema numérico de base 2. Los dos dígitos en el sistema binario 1 y 0 se llaman *bits*.

El sistema decimal, que es el sistema numérico que conocemos, es fácil de utilizar y que todo mundo maneja sin mayor problema, y este será pues el sistema que se tome como referencia para poder introducirnos en el manejo del sistema binario, el lenguaje de nuestros circuitos lógicos. El sistema decimal o también llamado de base 10 es llamado así por que va de un dígito 0 hasta el 9. En este sistema la posición de los dígitos determina la magnitud del numero leído, por ejemplo: el número 10 es la conjunción de los dígitos 1 y 0, el cual se leerá como 1 decena y 0 unidades. Este concepto es sencillo y la conjunción de los números que se encuentran en el sistema decimal nos pueden llevar a acrecentar las posibilidades de mayores cálculos aritméticos, sin perdernos conforme estos aumenten.

Al leer un numero decimal como lo puede ser el 524 decimos que son; 5 centenas, 2 decenas y 4 unidades. Esto se puede expresar también de otra manera:

$$524 = 5 \text{ x } 10^2 + 2 \text{ x } 10^1 + 4 \text{ x } 10^0$$

524 = 5 centenas + 2 decenas + 4 unidades

Donde: $10^2 = 100$

$10^1 = 10$

$10^0 = 1$ (por definición)

10 = La base del numero decimal 524

A continuación mencionaremos algunas formas de cómo poder hacer conversiones de números binarios a decimales y viceversa, haciendo una analogía similar a la anterior, hay que recordar que el buen manejo de los números binarios nos ayudara a poder entender mejor como reducir los diseños de los circuitos lógicos, ya que estos funcionan, como dijimos, de dos estados representados por, encendido y apagado.

Conversión de números binarios a decimales.

Una forma sencilla de poder convertir los números binarios a decimales es, sustituyendo la base (en este caso será 2), luego el digito binario que desee convertir, posteriormente colocar los exponentes en la posición correcta (se inicia de derecha a izquierda con el 0, 1, 2, 3,…, n). Esto puede parecer tedioso, bueno, el manejo de los números binarios al principio no es tan sencillo y haciendo todo este procedimiento se vuelve rebuscado. Para estos casos y para facilitarnos la conversión nos apoyaremos en un modelo, que se muestra en la tabla 1.1, y en la que solo tendremos que colocar debajo de ella los decimales que se obtienen de la conversión.

Numero binario a convertir → 1 1 0 1

					1	1	0	1
1×2^8	1×2^7	1×2^6	1×2^5	1×2^4	1×2^3	1×2^2	0×2^1	1×2^0
256	128	64	32	16	8	4	0	1

Se colocan solo los valores que se encuentran sobre los unos ya que serán estos los números que se sumaran. → 8 4 0 1

Entonces, el numero binario 1101 resultará de la suma: 8 + 4 + 0 + 1 = 13 decimal

Tabla 1.1 Conversión binario a decimal.

La tabla 1.1 es muy útil ya que podemos colocar sobre ella la cantidad de números binarios que queramos. Esta tabla solo llega al $1 \times 2^8 = 256$ para un número binario de 9 bits, pero se puede hacer mas grande o mas pequeña según sea el numero binario.

Cabe mencionar, que en los números decimales, la posición de los dígitos dentro de un numero tienen un valor distinto, el digito que se encuentra mas a la derecha (unidades) tendrá un valor menor o será menos significativo y el que se encuentre a la izquierda será de mayor valor o más significativo. En los números binarios esto será de la misma manera, ya que el bit que se encuentra a la derecha del numero binario será el menos significativo LSB (Low Significative Bit) y el que se encuentre a la izquierda será el más significativo MSB (Main Significative Bit).

Conversión de números decimales a binarios.

Al haber una forma de convertir a un numero binario en decimal, por supuesto, existe también una forma de hacer lo inverso y convertir un numero decimal a binario. Este procedimiento lo podemos llevar a cabo apoyándonos de el método de *la división entre 2 repetida*, esto quiere decir que el numero decimal que queramos convertir a numero binario tendrá que ser dividido entre 2, y el cociente resultante nuevamente entre dos, así hasta el residuo entero sea igual a cero, entonces ahí pararemos. Los residuos fraccionarios generados de cada una de estas divisiones formaran nuestro número en binario (0 = 0 lógico y ≠ 0 es 1 lógico), considerando que el primer residuo obtenido será el bit menos significativo LSB, y el ultimo residuo producido el bit más significativo MSB. Un ejemplo para mostrar la aplicación del método de la división entre 2 repetida, para convertir el numero 19 decimal a binario se vería como:

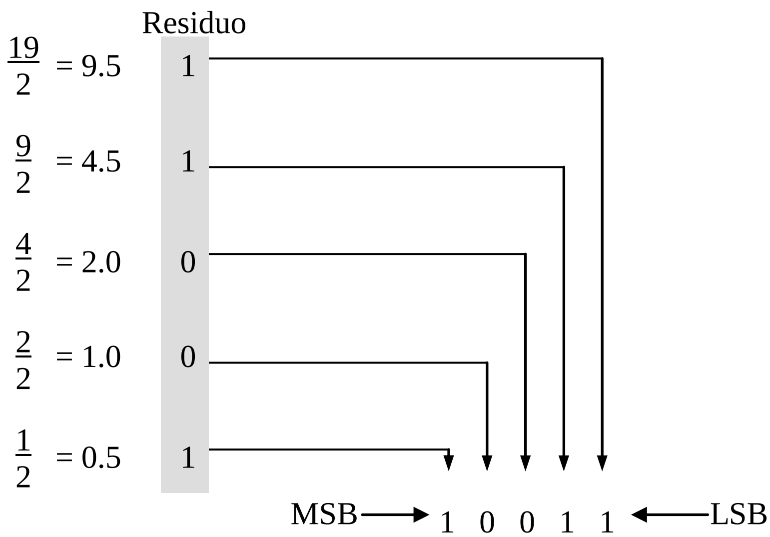


Figura 1.1. Método de la división entre dos repetida.

1.2. Código binario

Uno de los problemas que se presentan normalmente en el manejo de los números binarios es que, la mayoría de las personas estamos acostumbrados al manejo de los números decimales y no de los binarios. Por este motivo se han buscado maneras mas sencillas de cómo manejar a los sistemas digitales que deben recibir a los números binarios a través de códigos, los cuales son conjuntos de cadenas de n bits en la que cada una representa un carácter diferente y a cada cadena valida de n bits se le conoce como palabra de código. Existen una gran variedad de códigos como lo son el Código

Gray, Código de exceso tres, Códigos de detección de error, Códigos de distancia unitaria, Código ASCII, etc. En esta tesis se trabajara con el Código BCD, siendo este un código básico, fácil de manejar, además de que es el que tiene más uso en sistemas digitales.

Código BCD.

Este código BCD (Binary Code Decimal), es un código decimal codificado en binario, esto quiere decir, que cada digito decimal que conocemos del 0 al 9 tiene una representación binaria de cuatro bits. Una de las ventajas que existen al utilizar este código es que solo se tiene que recordar la representación de cada digito decimal en su representación de 4 bits, olvidándonos de realizar una conversión de binario a decimal, o de decimal a binario. Podemos observar en la tabla 1.2 que se nos muestra la relación del lado derecho del numero en BCD correspondiente a cada uno de los números decimales del lado izquierdo.

Digito Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Tabla 1.2 Código BCD

Con ayuda de la tabla anterior podemos representar cualquier numero decimal que decidamos. Por ejemplo el numero decimal 23 al convertirlo en BCD lo observamos como un numero constituido por un numero 2 y un numero 3, en donde; el decimal 2 = 0010 y el decimal 3 = 0011 en correspondencia al valor en BCD, por lo tanto el decimal 23 = 0010 0011.

Otros ejemplos pueden ser los siguientes:

a) El numero decimal 93

$\overbrace{1001}^{9}\ \overbrace{0011}^{3}$

Digito Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Por lo tanto: <u>93 = 1001 0011</u>

Figura 1.2. Código BCD para el numero 93.

b) El numero decimal 44

$\overbrace{0100}^{4}\ \overbrace{0100}^{4}$

Digito Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Por lo tanto: <u>44 = 0100 0100</u>

Figura 1.3. Código BCD para el numero 44.

c) El numero decimal 56

$\overbrace{0101}^{5}\ \overbrace{0110}^{6}$

Digito Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Por lo tanto: <u>56 = 0101 0110</u>

Figura 1.4. Código BCD para el numero 56.

d) El numero decimal 10

$\overbrace{0001}^{1}\ \overbrace{0000}^{0}$

Digito Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Por lo tanto: <u>10 = 0001 0000</u>

Figura 1.5. Código BCD para el numero 10.

1.3. Lógica binaria

La lógica binaria engloba el análisis de los valores de distintas variables, estos valores que pueden tener las variables que manejemos serán en nuestra lógica solamente dos y son el 1 y 0 lógicos, de aquí el nombre binario.

“La lógica binaria se usa para describir, de una manera matemática el procesamiento y manipuleo de la información binaria.”

La lógica binaria es fundamental como herramienta para comprender cual es la composición funcional de los sistemas digitales que posteriormente trataremos en este capítulo, además de ser introductorio para el manejo del álgebra de Boole tema expuesto en el capítulo 2.

Variables binarias y operaciones lógicas.

Obviamente al hablar de una lógica binaria debemos darnos cuenta que esta compuesta por variables binarias y operaciones lógicas que emergen por el manejo de estas variables. Las variables binarias se pueden observar como letras como lo pueden ser; A, B, C, X, Y, Z, etc., recordando que cada variable a la que hagamos referencia tendrá solo dos valores posibles: 1 y 0 lógicos.

De las operaciones lógicas hablaremos de las tres más básicas y con las cuales se puede trabajar de muy buena manera, ya que la variedad es mayor, por el momento las básicas serán suficientes y son: AND, OR y NOT. Estas funciones se encuentran en ingles, aunque nosotros las conoceríamos como funciones lógicas; Y, O y NO, pero para el manejo estandarizado que comúnmente manejan la mayoría de los textos lo manejaremos en ingles. A continuación describiremos la función principal de cada una de las operaciones lógicas básicas.

a) Operación Lógica AND:

La operación lógica AND nos sirve como una condicionante en la conjunción de dos variables, en la que si la primera variable toma el valor de 1 y la segunda de la misma manera toma el valor de 1, "si y solo si" sucede esto nuestro valor resultante de una tercera variable será igual a 1, de otra forma el resultado será 0. El operador para la operación AND se ve representado por un punto, una cruz o paréntesis; aunque muchas veces está ausente.

La operación AND de las variables x, y es: $x \bullet y = z$ ó $xy = z$ se lee como: x y y es igual a z, y consta de que $z = 1$ "si y solo si" $x = 1$ y $y = 1$, en caso contrario $z = 0$. La sustitución de valores en las 4 combinaciones posibles de las 2 variables y aplicando la operación AND se vería como:

$$\begin{array}{ccccc} x & \bullet & y & = & z \\ \downarrow & & \downarrow & & \downarrow \\ (1) & \bullet & (1) & = & 1 \\ (1) & \bullet & (0) & = & 0 \\ (0) & \bullet & (1) & = & 0 \\ (0) & \bullet & (0) & = & 0 \end{array}$$

Figura 1.6. Aplicación de la operación AND para la combinación de dos variables.

Obsérvese de la figura 1.6 como solo en la condición que nos marca el operador en la que x = 1 y también y = 1 el valor es z = 1, y en todas las demás combinaciones el valor de z = 0.

b) Operación Lógica OR:

Esta operación lógica nos indica que en la conjunción de dos variables, la primera siendo igual a 1 o la segunda igual a 1, o si ambas tienen un valor 1 el resultado de la tercera variable será 1, pero si la primera y segunda variable tienen valor 0 entonces el valor será 0. Se ejemplificara de la siguiente manera.

El operador en este caso se representara por un signo mas (+).

La operación OR de las variables x, y es: $x + y = z$ y se lee como x o y es igual a z, y consta de que z = 1, si x = 1 o y = 1, o también si x = 1 y y = 1, entonces z = 1, si de lo contrario x = 0 y y= 0 el valor de z = 0 solo para este caso.
En la sustitución de valores y aplicando el operador se vería como:

$$x + y = z$$

$$(1) + (1) = 1$$
$$(1) + (0) = 1$$
$$(0) + (1) = 1$$
$$(0) + (0) = 0$$

Figura 1.7. Aplicación de la operación OR para la combinación de dos variables.

c) Operación Lógica NOT

La operación en esta ocasión nos dice que de dos variables, una de ellas será lo inverso de lo que representa la otra.

La presencia de un apostrofe o una barra sobre la variable nos indica la existencia del operador NOT.

La operación NOT de la variable x es: $x´ = z$ ó $x = z$ y se lee como “x no es igual a z”, entendiéndose que una será lo que la otra no es y como solo existen dos valores para estas variables como ya lo habíamos visto, no es difícil deducir el otro valor de la variable.

En la sustitución de valores y aplicando el operador se vería como:

$$\begin{array}{cc} x´ & = z \\ \downarrow & \downarrow \\ (\ 1\ )´ & = 0 \\ (\ 0\ )´ & = 1 \end{array}$$

Figura 1.8. Aplicación de la operación NOT para la combinación de dos variables.

Obsérvese como en esta operación lógica solo existen dos únicas combinaciones posibles.

Para observar de una manera mas general la relación que existe entre los distintos valores que pueden tomar las variables x y y, y el resultado z que surja de las distintas combinaciones de entre las variables lo haremos de forma sencilla mostrando esto con **tablas de verdad**. Las tablas de verdad son simples tablas donde se ordenan las diferentes combinaciones posibles de las variables involucradas con respecto al resultado que estas arrojan. Por ejemplo, en la tabla 1.3 se muestran las tablas de verdad con dos variables para las operaciones lógicas AND, OR y NOT.

AND			OR			NOT	
x	y	$x \bullet y = z$	x	y	$x + y = z$	x'	z
0	0	0	0	0	0		
0	1	0	0	1	1	0	1
1	0	0	1	0	1	1	0
1	1	1	1	1	1		

Tabla 1.3 Tablas de verdad para las operaciones lógicas AND, OR y NOT.

1.4. Teoremas de De Morgan

Los Teoremas De Morgan tienen importancia en la simplificación de expresiones algebraicas que representan a un sistema digital, tema que se discutirá en el capítulo 2, en donde se tratara a fondo como técnica de simplificación; dentro de los cuales se encuentra a los 2 teoremas De Morgan que a continuación enunciaremos para 2 variables:

a) $(x + y)' = x' \bullet y'$

b) $(x \bullet y)' = x' + y'$

Como se puede observar los teoremas se están aplicando para solo dos variables digitales (x , y), pero se puede agregar mas. La comprobación de este teorema se realizara por medio de tablas de verdad, ya que es una manera muy sencilla de hacerlo, esto no quiere decir que todos los teoremas o postulados que se mostraran aquí tendrán que comprobarse, se tomaran como verdaderos, y si el lector quiere verificar las comprobaciones puede hacerlo por su parte.

Como primer paso de la comprobación, se realiza la representación en tabla de verdad de las 2 operaciones lógicas (OR y NOT) que aparecen en la expresión: $(x + y)'$, como se muestra en la tabla 1.4.

x	y	$x + y$	$(x + y)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Tabla 1.4. Tabla de verdad de $(x + y)'$

El segundo paso a realizar es el elaborar la tabla de verdad del segundo término de la igualdad $x' \bullet y'$, con ayuda de la tabla de verdad que aparece en la tabla 1.5; en donde se encuentran los resultados de las operaciones lógicas NOT y AND.

x	y	x'	y'	$x' \bullet y'$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Tabla 1.5 Tabla de verdad de $x' \bullet y'$

De las dos tablas anteriores se observa que la igualdad se cumple, ya que los valores obtenidos al realizar las operaciones lógicas son los mismos, por lo tanto queda comprobado el primer teorema De Morgan. De forma similar a la comprobación que se acaba de realizar se lleva a cabo la comprobación del segundo teorema, dicha comprobación se le deja al lector a manera de ejercicio.

1.5. Postulados de Huntington

En la necesidad de encontrar dos funciones equivalentes en donde alguna de ellas sea mas conveniente de usar, es uno de los objetivos que ha llevado al desarrollo de un sistema matemático deductivo y este sistema es el álgebra de Boole. Este tipo de álgebra esta compuesta por un conjunto de teoremas y postulados, aunque son varios estos postulados y teoremas algunos sirvieron de base a Huntington (1904), para enunciar sus postulados.

Los postulados de Huntington tienen que ver con varios operadores definidos sobre un conjunto de elementos básicos. Un conjunto es cualquier colección de elementos que tienen una o varias propiedades en común. Huntington encontró que todos los resultados e implicaciones del álgebra de Boole pueden obtenerse de los siguientes postulados.

I.- Existe un conjunto de K objetos o elementos sujetos a una relación de equivalencia, denotada por el signo " = " que satisface el principio de sustitución.

Donde la sustitución significa: $a = b$, b puede sustituir a a en cualquier expresión que contenga a b, sin afectar la validez de la expresión.

II.a.- Se define una regla de combinación " + " en tal forma que $a + b$ esta en K.

II.b.- Se define una regla de combinación " • " en tal forma que $a \bullet b$, que se abrevia ab, esta en K, siempre que tanto a como b estén en K.

III.a.- Existe un elemento 0 en K, tal que, para cada a en K, $a + 0 = a$.

III.b.- Existe un elemento 1 en K, tal que, para cada a en K, $a \bullet 1 = a$.

$$\left.\begin{array}{l} \text{IV.a.- } a + b = b + a. \\ \\ \text{IV.b.- } a \bullet b = b \bullet a. \end{array}\right\} \text{Leyes conmutativas}$$

$$\left.\begin{array}{l} \text{V.a.- } a + (b \bullet c) = (a + b) \bullet (a + c) \\ \\ \text{V.b.- } a \bullet (b + c) = (a \bullet b) + (a \bullet c) \end{array}\right\} \text{Leyes distributivas}$$

VI.- Para cada elemento *a* en *K*, existe un elemento *a* tal que:

$$a \bullet a' = 0 \qquad \text{y} \qquad a + a' = 1$$

VII.- Existen cuando menos dos elementos x y y en K, tales que $x \neq y$.

Se debe observar que en los postulados se hace mención de un conjunto *K* de objetos o elementos pero no dice nada que fije el numero o el tipo de elementos que constituyen a *K*.

Otro aspecto que debemos observar es la similitud entre estos postulados y los del álgebra ordinaria, solo que no obstante la similitud entre estos, se notará que la primera ley distributiva (la distribución sobre la suma), no se aplica al álgebra ordinaria, además de que el álgebra ordinaria no cuenta con una a' definida.

1.6. Teorema de Shannon

De la misma forma en que los postulados de Huntington nos sirven para la simplificación de funciones al intercambiarlas por otros equivalentes, pero de fácil manejo, de esa misma forma lo hará el teorema Shannon (1938). Este pues también forma parte del álgebra de Boole que como ya mencionamos analizaremos en el capitulo siguiente.

El teorema de Shannon es un método para obtener de una forma factorizada cualquier expresión booleana, lo que ayuda a encontrar soluciones con tres o mas niveles de lógica. La Factorización que se logra con el teorema reduce el numero necesario de componentes o el numero de literales requerido. La factorización incrementa la longitud máxima de la trayectoria de una señal desde la entrada hasta la salida de un circuito, por lo que el circuito se hace lento. Esto se compensa con que la factorización también disminuye la carga máxima de entrada de un circuito. El teorema se enuncia a continuación:

I.a.- $F (X_1, X_2, \dots , X_n) = X_1 \bullet F (1, X_2, \dots , X_n) + X_1' \bullet F (0, X_2, \dots , X_n)$

II.a.- $F (X_1, X_2, \dots , X_n) = [X_1 + F (0, X_2, \dots , X_n)] \bullet [X_1' + F (1, X_2, \dots , X_n)]$

Se observa que el teorema nos ayuda a separar cualquier función en dos subfunciones, una de ellas queda multiplicada por una variable denominada de separación, mientras que la otra queda multiplicada por el complemento. Las subfunciones son independientes de la variable de separación, la cual tiene el valor 1 lógico en una subfunción y un 0 lógico en la otra.

El teorema de Shannon puede escribirse de manera simplificada como:

III.a.- $F(X) = X \bullet F(1) + X' \bullet F(0) = [X + F(0)] \bullet [X' + F(1)]$

1.7. Sistemas digitales

Para poder conocer de una manera más sencilla lo que son los sistemas digitales, habrá que definir en primera estancia lo que es un sistema y de ahí partir al entendimiento de los sistemas digitales.

De una forma concreta un sistema es un conjunto de elementos que tienen algún tipo de relación entre si, estos elementos a su vez pueden tener una estructura de sistema, si esto es así entonces estos pasarán a ser subsistemas del sistema al que pertenecen.

De tal manera un sistema digital maneja datos que no son otra cosa que elementos discretos y de una forma finita. Los elementos discretos en un sistema digital son señales digitales, las cuales son niveles de voltaje o magnitudes de corriente y cada nivel de voltaje nos puede representar un elemento de los datos.

Las señales de un sistema digital con compuertas lógicas están restringidos a dos voltajes: 0v para el 0 lógico y 5v para el 1 lógico. Si comparamos un sistema digital con un sistema analógico, la diferencia es que en el funcionamiento el digital es mas preciso y rápido, mientras que el analógico basado en la manipulación de los valores representados en forma continua toma valores dentro de un intervalo de +5V a –5V, por lo que se muestra mas lento.

Con la finalidad de tener una idea mas exacta de lo que es un sistema digital ejemplificaremos haciendo mención de dispositivos digitales, que son conocidos.

Nuestro primer ejemplo es una calculadora electrónica, en la cual introducimos valores digitales por medio de un teclado, y la señal de salida se ve representa por un conjunto de valores digitales, antes de la pantalla de despliegue.

De una forma similar trabaja una computadora, aunque de alguna manera esta es más compleja, por lo cual nos apoyaremos de un diagrama a bloques (figura 1.9) para entender mejor su funcionamiento.

En este sistema para empezar a manejar los datos del dispositivo de entrada, se requiere de un programa que los traslade a la unidad de memoria. La unidad de control se encarga entonces de recuperar de la memoria instrucciones del programa, analiza y da las indicaciones a la unidad de procesamiento para que realice las instrucciones. Los resultados se remiten a la unidad de memoria para su almacenamiento y luego se transmiten al dispositivo de salida.

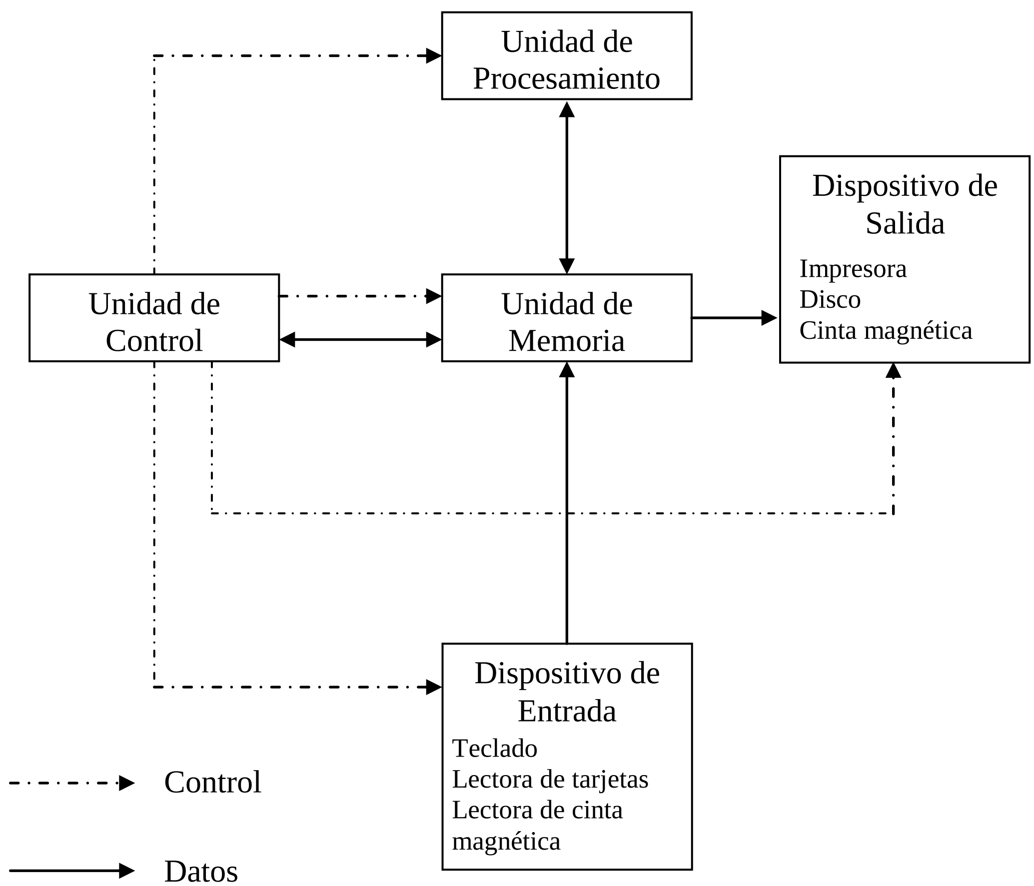


Figura 1.9. Diagrama a bloques del funcionamiento de una computadora.

En la figura 1.10 podemos observar otro ejemplo de un sistema digital solo que este se encuentra conviviendo con un sistema analógico, por lo cual podemos ver que a pesar de que los sistemas digitales son superiores a los sistemas analógicos, no quiere decir que estos últimos dejen de ser ocupados o funcionales, o no al menos por el momento.

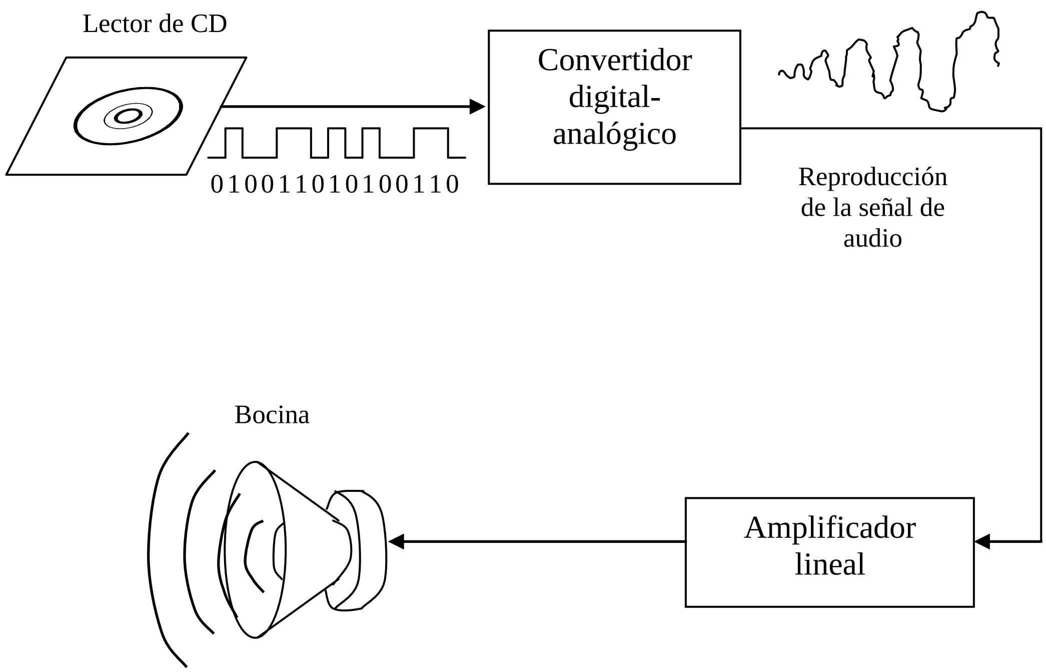


Figura 1.10. Sistema digital y analógico.

Donde se puede ver que un lector de CD manda su información a un convertidor digital – analógico, ya que la información contenida en el CD esta como datos binarios y el lector la lee y envía de esta forma, el convertidor transforma la información binaria en una señal de audio (señal analógica) enviándola a un amplificador lineal que amplificará la señal, para que pueda ser enviada a una bocina que emitirá el sonido o música.

1.8. Teoría de los circuitos combinacionales

Los circuitos combinacionales tienen sus preceptos en el algebra de Boole, de la cual se comenta mas a fondo en la parte inicial del capitulo 2 de este escrito y que es una base matemática de mediados del siglo XIX. Comprobada por Shannon en una obra llamada Algebra de Conmutación, que comprueba y relaciona las funciones básicas de la matemática de Boole a algo práctico; en otras palabras el algebra de Boole expresa correspondencias de las variables que maneja con las variables de los sistemas digitales actuales.

Las funciones booleanas que se forman por la composición de varias variables binarias, pueden agruparse en modulos o bloques unidos por operadores y que realizan operaciones especificas son un tipo de funciones "combinadas" o que se "combinan". Estas funciones combinadas formas lo que llamamos "Circuitos Combinacionales".

A través del diseño de circuitos combinacionales se han creado ya algunos arreglos que son muy utiles para arreglos posteriores mas complejos y que se pueden dividir en dos grupos, que son; los que facilitan la distribución de datos y los que permiten la selección de posibilidades. Los primeros serian multiplexores, demultiplexores y decodificadores, y los segundos son los que simplemente trasladan la misma información de un código a otro, llamados codificadores.

Dichos circuitos tienen salidas que dependen solamente de sus entradas actuales, podremos mencionar que por el contrario y en comparación con los circuitos llamados secuenciales, estos últimos dependen no solo de sus entradas actuales, sino también de la "secuencia" anterior de entradas pasadas, en pocas palabras son circuitos que contienen memoria, una memoria que se consigue mediante una retroalimentación de sus propias funciones booleanas.

Un circuito combinacional puede tener una gran cantidad de entradas y compuertas lógicas, pero no lazos de retroalimentación. Estos lazos de retroalimentación son una conexión que va desde la salida de alguna compuerta y se le permite regresar hasta alguna de las entradas; una conexión de ese tipo genera, naturalmente un fenómeno de "secuencia" en el circuito, característica principal de los circuitos secuenciales. Observe la figura 1.11 donde se muestra un diagrama a bloque de un circuito combinacional y un secuencial.

a) b)

Figura 1.11. Diagrama a bloque: a) circuito combinacional, b) circuito secuencial.

1.8.1 Características.

Las señales en un circuito eléctrico tienen una magnitud física llamada tensión eléctrica, la cual ha sido cambiada de cómo se emplea en un circuito analógico ha como se usa en un circuito digital. Cuando las tareas que se realizan por medio de sistemas analógicos, las señales de entrada son de una tensión continua y que se encuentra entre dos valores extremos: +Vp y -Vp, por ejemplo la tensión de un sistema analógico puede variar de +5 volts a – 5 volts. Por otra parte, en los sistemas digitales las señales de trabajo o magnitudes físicas solo pueden tomar valores discretos, por ejemplo en la salida de un sistema digital puede restringirse únicamente a dos valores como son 0 volts y 5 volts; que corresponden a los valores logicos 0 y 1. Se puede trasladar una señal analógica a una señal digital, para lograrlo es preciso utilizar algunos de los dispositivos electrónicos de los existentes, que convierten la señal analógica a una digital y que se llaman convertidores. La señal analógica convertida en digital se muestra en la figura 1.12.

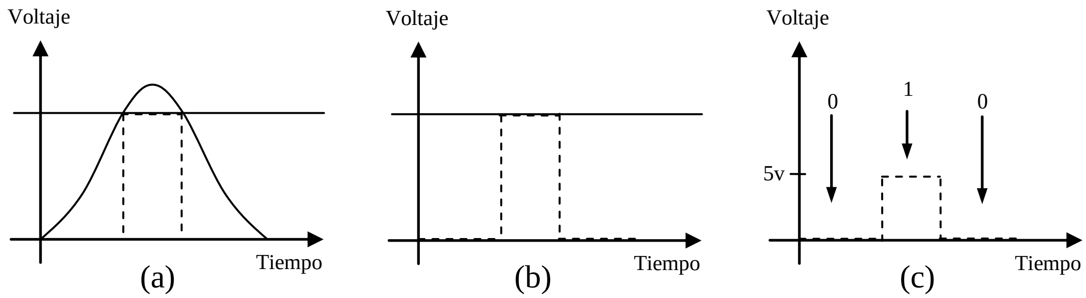


Figura 1.12. Digitalización de una señal analógica.

En la figura 1.12.a se toma una muestra de la señal analógica, en la figura 1.12.b se cuantifica a la señal muestreada y como se observa en la figura 1.12.c la señal se encuentra ya en solo dos niveles de voltaje llamados niveles de estado, alto y bajo, donde corresponde al digito 1 un nivel alto (H) y un 0 para un nivel bajo (L).

Este tipo de señal digital puede ser utilizada de dos maneras e interpretarse igual, y es a lo que también suele llamársele "lógica positiva" y "lógica negativa." En la denominada lógica positiva la señal que utilizan los circuitos combinacionales toman dos niveles, el alto (H) y que define el estado lógico 1, que seria representativo de una presencia de voltaje (5v) y el bajo (L) que representa al estado lógico 0 e indica ausencia de voltaje (0v), como se muestra en la figura 1.13.a;

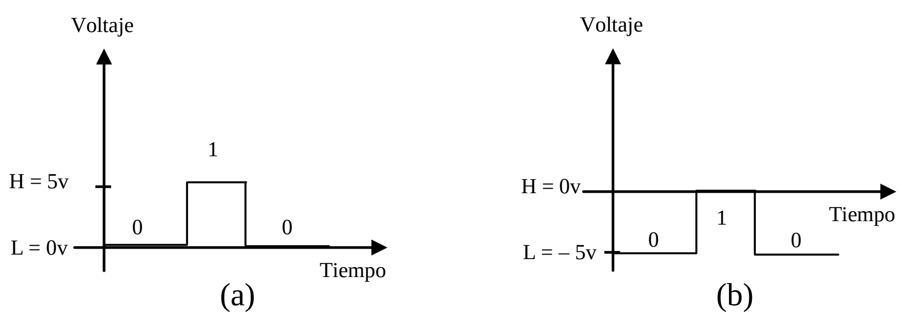


Figura 1.13. Lógica positiva.

Pero también lo es una señal digital con voltaje negativo, como se observa en la figura 1.13.b, donde la ausencia de una tensión eléctrica sería considerado como un 1 lógico, un valor de voltaje negativo como 0 lógico.

En la lógica negativa el uso de la señal es el caso contrario, o sea como se muestra en la figura 1.14. Donde particularmente en 1.14.a se observa como el valor mas alto o de voltaje +5v representa un 0 lógico y la ausencia de voltaje un 1 lógico; mientras que en 1.14.b un voltaje negativo de – 5v representa a un digito 1 lógico.

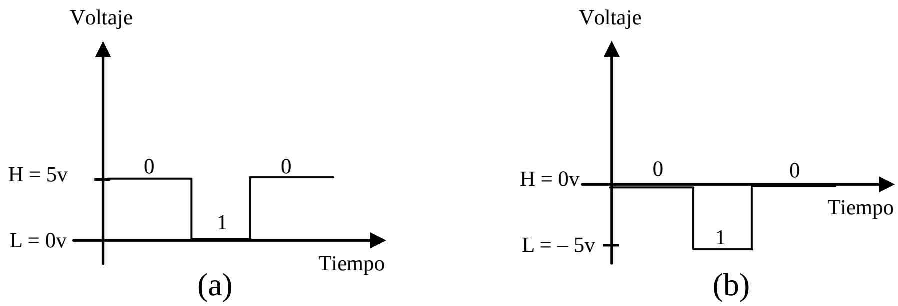


Figura 1.14. Representación de las señales con las que se puede trabajar en la lógica negativa.

Es necesario comentar que la lógica positiva es un poco mas familiar para el trabajo que se realiza en los sistemas combinacionales pero no exclusivo.

Otra de las características importantes en los circuitos combinacionales es que al trabajar con señales que representan dos estados como las que acabamos de ver, es posible ocupar una herramienta teórica de fundamentos totalmente matemáticos, llamada algebra de Boole, la que se fundamenta en el uso solamente de una base numérica de 2 dígitos (0 y 1) y que sirve para analizar y sintetizar circuitos lógicos combinacionales.

Como característica adicional es que el numero de las combinaciones de las señales de entrada se genera de manera exponencial, es decir que con el crecimiento del numero de entradas se puede duplicar, triplicar, etc.y se puede calcular con la relación 2^n, donde el exponente n = al numero de entradas.

1.8.2 Funcionamiento de una compuerta básica.

Con la intención de complementar las características de los circuitos combinacionales, es conveniente anexar en esta tesis al menos un ejemplo que pueda mostrar como estas características son contempladas para componentes básicas dentro de los circuitos combinacionales, como lo serian las compuertas. Para este caso se ejemplifica como funciona una compuerta AND, con el afán de complementar la información necesaria, pero a su vez cuidar no caer en una interminable cascada de conceptos, y así conocer como internamente realizan las operaciones lógicas las compuertas. En el punto 1.9 de este trabajo se abunda más sobre las operaciones de las compuertas básicas.

En la figura 1.15 se observa un diagrama de la compuerta AND de dos entradas con diodos semiconductores y una resistencia, además de la fuente de alimentación. El funcionamiento de este diagrama esta basado en los estados de los diodos. En el punto de conexión de los ánodos con la resistencia R se tiene la salida V_0, es decir el punto donde aparecerán los niveles de salida, ya sea alto o bajo según las señales que se apliquen a los diodos. Cabe mencionar que el diagrama de la figura 1.15 esta dibujado en una lógica positiva y podría analizarse de una forma similar para una lógica negativa. Con 5 volts en E y conectado al cátodo del D_1 y D_2 si en ninguna de las entradas E_1 y E_2 existiera un potencial como por ejemplo de 5 volts se asume que ambos diodos se encontrarían en estado de encendido ya que se encuentran polarizados para la conducción y en el punto de salida V_0 el voltaje sería de 0 volts, ya que en ambos diodos circularía alguna corriente dejando a $V_0 = 0.7$ volts (siendo el diodo de silicio) que sería tomado como que $V_0 = 0$ volts o un 0 lógico. Entonces con la combinación de las entradas $E_1 = 0$ y $E_2 = 0$ tenemos una salida $V_0 = 0$.

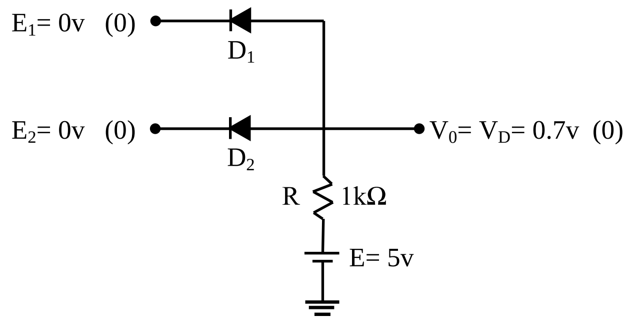


Figura 1.15. Compuerta lógica AND de dos entradas con Diodos.

Para el caso de que alguno de los diodos tuviera en su cátodo una polarización de 5 volts y el otro no, como por ejemplo para D_1, entonces la E_1 = 5 volts mientras que la E_2 = 0 volts, este D_1 se encontraría polarizado inverso por lo que su estado pasaría a apagado quedando abierto el circuito en esa sección, mientras que el D_2 seguiría permitiendo el flujo de una corriente I, dejando de la misma manera a V_0 = 0.7 volts e igual a un 0 lógico. Es decir; con una combinación de E_1 = 5v y E_2 = 0v el V_0 = 0v, (ver figura 1.16).

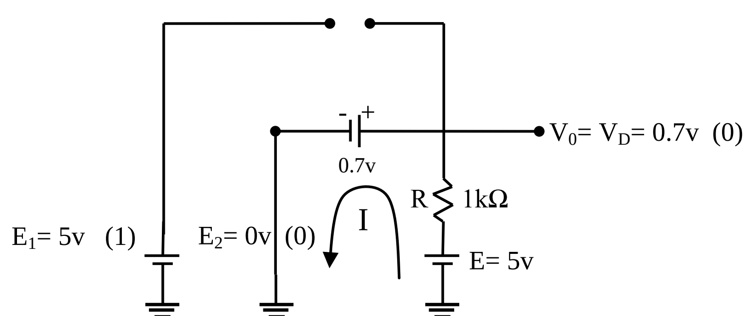


Figura 1.16. Cambio del estado de uno de los diodos en la compuerta lógica AND.

Para el caso de que la E_1 = 0 volts y la E_2 = 5 volts pasaría el mismo fenómeno, es decir; D_1 estaría en encendido y D_2 en apagado. La corriente fluiría por D_1 y aunque D_2 no conduzca aun así V_0 = 0.7v o considerado un 0 lógico. Así en el caso E_1 = 0v y E_2 = 5v, V_0 = 0v.

Por último cuando en E_1 = 5 volts y E_2 = 5 volts los diodos D_1 y D_2 están en estado de apagado, por lo que no conducirían corriente alguna, por lo que la corriente fluiría hacia V_0 obteniéndose en ese punto el voltaje de E, considerando obviamente el V_D que se estaría restando en la salida, que sería V_0 = 4.3 volts, considerada como salida lógica alta o un 1 lógico. De esta manera con la combinación de E_1 = 5v y E_2 = 5v, V_0 = 5v. De tal forma que las combinaciones totales expuestas quedarían como en la tabla 1.6 del punto 1.9.1.

1.9. Operaciones lógicas.

En el subtema 1.3 ya hablamos de lo que son las operaciones lógicas o al menos de tres de las mas importantes que son la AND, OR y NOT, pero son solamente tres de un total de 8 operaciones que se crean o surgen cuando se trata de compuertas lógicas

1.9.1. Operación AND

Empezamos retomando que una de las operaciones lógicas entre dos variables es la AND, operación lógica que como decíamos nos hace entender que implícitamente nos da una condición de “si y solo si” o como su propia conjunción en ingles traduciendo al español nos quiere dar entender, que no es mas que la conjunción “y”. Esta operación nos dice que entre dos variables que tengan el valor de 1 “y” 1 el valor de la operación lógica es 1 en caso de que alguna de las variables o ambas tomen el valor de un 0 lógico, entonces la respuesta al final será también un 0 lógico. En la tabla 1.6 se muestran las operaciones que resultan de la combinación de dos variables con la operación AND.

VARIABLES		RESULTANTE
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 1.6. Combinaciones de la operación AND con dos variables.

Cabe señalar que esta operación lógica AND en el algebra de Boole utiliza el símbolo "•" para unir dos o mas variables quedando expresiones como:

A • B, X • Y, C • D, etc.

1.9.2. Operación OR

Otro de los operadores entre variables en el algebra booleana es la operación OR, que equivaldría a la conjunción "o" y que nos da la opción de que entre dos variables la función que se obtendría será un 1 lógico si en tanto en la primera variable o en la segunda se encuentra un 1 lógico, solo será 0 cuando ninguna de las variables valga 1. En la tabla 1.7 se muestra una tabla con sus posibles combinaciones entre dos variables y el resultado al aplicar la operación.

VARIABLES		RESULTANTE
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 1.7. Combinaciones de la operación OR con dos variables.

El símbolo que representa a la operación es el signo + entre variables que estén agrupadas, por ejemplo:

X+Y+Z, A+B, C+D+E, etc.

1.9.3. Operación NOT

La operación NOT, que es la obtención del inverso de una variable, es decir que cuando una variable tiene el valor de 1 su inverso será 0 y de la misma manera si el valor en primera instancia de la variable es 0 su inverso será 1. Esta operación se puede indicar con un apostrofe o una línea en la parte superior de la variable, se vería de la siguiente manera:

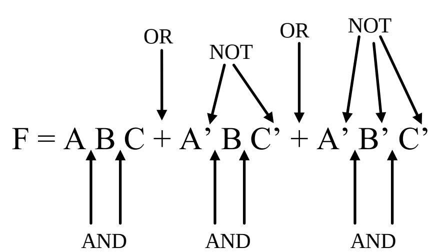
$$\overline{A}\,, \quad A', \quad \overline{X}, \quad X', \quad \overline{Y}, \quad Y', \quad \text{etc.}$$

La tabla de verdad para esta operación aparece en la tabla 1.8.

VARIABLE	VARIABLE NOT
A	A'
0	1
1	0

Tabla 1.8. Operación NOT cuando se aplica a una variable que vale 0 y 1 lógicos.

Ya ejemplificando el uso de las operaciones AND, OR y NOT y sus símbolos en una expresión booleana con varias variables podría quedar de la siguiente manera:



.

Figura 1.17. Símbolos de la operaciones AND, OR y NOT.

1.9.4. Operación NAND

Con la operación NAND es posible realizar una gran cantidad de expresiones booleanas. Si observamos un poco a detalle la palabra NAND nos daremos cuenta que esta formada por la contracción de la palabra NOT y AND, por lo que también deduciremos fácilmente que esta operación esta compuesta por las dos operaciones que dan forma su nombre.

Podemos comentar que esta operación esta así pues esta formada por dos operaciones básicas y podríamos deducir que es el complemento de la operación AND, ya que después de haber realizado esta ultima podremos afectarla por la operación NOT quedando los complementos de la operación AND, de una forma mas simple diremos que cuando la operación AND arroje como resultado un 1 lógico este se invertirá para obtener finalmente un 0 lógico y si el resultado arrojado por AND es un 0 lógico este se invertirá para dar mostrar un 1 lógico. A continuación se muestra la tabla 1.9 con las combinaciones de dos variables que a las cuales se les aplica la operación NAND.

VARIABLES		RESULTANTE
A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 1.9. Combinaciones de la operación NAND con dos variables.

Al observar la tabla 1.9 y compararla con la tabla 1.6 nos damos cuenta que en la resultante de la operación AND en la posición donde se encuentra un 1 lógico ahora en la tabla de NAND se encuentra un 0 lógico, y donde se encontraban ceros lógicos en la operación AND, en la operación NAND se encuentran ahora unos lógicos, por lo que se concluye que NAND es la operación complemento de AND.

De la misma manera en que se utiliza una simbología para representar cada una de las tres principales operaciones, en el algebra booleana existe también un símbolo que representa la operación NAND y es una pequeña flecha vertical dirigida hacia arriba que se coloca entre variables, a continuación mostramos como se verían algunas expresiones con operación NAND utilizando los símbolos que representan NOT y AND, además de con el símbolo que representa directamente la operación NAND:

F = (AB)’ ó bien F = A↑B, F= (XY)’ ó bien F = X↑Y, etc.

1.9.5. Operación NOR

Así como en el caso anterior NAND es el complemento de AND, también existe un complemento para OR que sería la operación llamada NOR y que es otra de las operaciones importantes al mismo nivel que NAND. De la misma manera la operación NOR esta formada por dos funciones básicas; NOT y OR y por supuesto su nombre es una contracción de estas dos mismas y no solo en la cuestión gramatical si no también en funcionamiento, ya que al agrupar dos o mas variables y aplicar la operación OR se obtienen una serie de valores (unos y ceros lógicos) como los de la tabla 1.7 a los cuales inmediatamente se les aplica la operación NOT, (como en la tabla 1.10) es decir que se esta encontrando el complemento de OR.

VARIABLES		RESULTANTE
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 1.10. Combinaciones de la operación NOR con dos variables.

De la tabla 1.10 observamos como para la operación NOR si alguna variable ó ambas toman valor de un 1 lógico el resultado arrojado será un 0 lógico y solo en el caso en el que ambas variables valieran 0 lógico la resultante será 1 lógico.

También en esta operación lógica como en las anteriores cuenta con un símbolo que la representa en el manejo del algebra booleana y que es muy parecido al de la función NAND, es decir que es una pequeña flecha vertical que se coloca entre las variables que están involucradas en la operación, solo que esta se encuentra dirigida hacia abajo, por lo que se muestra a continuación un ejemplo de como se escribiría una expresión booleana con dos variables utilizando las operaciones OR y NOT, además de cómo se vería la misma función expresada con el símbolo que corresponde a la operación NOR:

$F = (W + Z)'$ ó bien $F = W{\downarrow}Z$, $F = (C + D)'$ ó bien $F = C{\downarrow}D$, etc.

1.9.6. Operación OR – exclusiva

Nuestra siguiente operación es la llamada OR-exclusiva que se llega abreviar como XOR o también EOR y es llamada exclusiva ya que en su funcionamiento al aplicarla en dos variables A y B por ejemplo tiene la propiedad de arrojar como resultante un 1 lógico cuando una sola de la variables involucradas ya sea A o B cuenten con un valor de 1 lógico, es decir que excluye (se vuelve exclusiva) cuando ambas variables son iguales arrojando en su caso un 0 lógico. Esta operación en algunas bibliografías también es conocida como operación de comparación, en donde A ó B pero no ambas. A continuación mostramos su tabla con las combinaciones, donde vemos las operaciones que se realizan según sean los valores de las dos variables involucradas:

VARIABLES		RESULTANTE
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 1.11. Combinaciones de la operación XOR con dos variables.

Notamos claramente de la tabla 1.11 como cuando no coinciden los valores de la variable A con los de B el valor resultante es un 1 lógico y cuando si coinciden un 0 lógico. Ahora en lo que respecta a su representación en el algebra de boole también ejemplificaremos como se vera con los símbolos de las operaciones básicas y también con su propio símbolo de operación viéndose como:

F = XY’ + X’Y o bien F = X ⊕ Y,   F = AB’ + A’B o bien F = A ⊕ B, etc.

De las expresiones anteriores podemos justificar el porque también la operación es llamada de comparación, ya que vemos que en el primer termino de la primera expresión booleana la variable Y es inversa y la X no, mientras que en el segundo termino de la misma función ahora X es inversa y la Y no lo es.

1.9.7. Operación NOR – exclusiva

A continuación la ultima operación de este segundo grupo de operaciones importantes en el algebra booleana es la llamada NOR-exclusiva, también conocida por su abreviatura como XNOR o en algunos casos también como ENOR. La operación NOR-exclusiva es el complemento de la OR-exclusiva, así como la NAND lo es para la AND o también como la NOR para la OR. A este tipo de operación también le es llamada de equivalencia, o conocida como de igualdad-coincidencia.

Así que siendo el complemento de la OR-exclusiva, la NOR-exclusiva, es una operación que también se utilizada en solo dos variables, en las cuales si existe coincidencia en el valor de estas, digamos un 1 lógico en ambas, la operación nos arroja a la salida un 1 lógico también, y en el caso de que ambas variables tengan el valor de 0 lógico la salida será de un 1 lógico, es decir que las variables sean iguales sin importar que sean de valor 1 o 0 lógicos, pero si no existe coincidencia en el valor de las variables entonces el resultado final será un 0 lógico. A continuación la tabla 1.12 muestra las posibles combinaciones de valores que pueden tomar dos variables y el resultado al aplicárseles la operación lógica NOR-exclusiva.

VARIABLES		RESULTANTE
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Tabla 1.12. Combinaciones de la operación XNOR con dos variables.

De igual forma, cuenta con su símbolo en el algebra booleana y su representación utilizando las operaciones básicas principales siguientes:

F = XY + X’Y’ o bien F = X Θ Y,   F = AB + A’B’ o bien F = A Θ B, etc.

1.10. Símbolos eléctricos de las compuertas.

Uno de los conceptos que se manejan en este trabajo de tesis y que debe quedarnos muy claro es que cuando manejamos compuertas lógicas, se hace con la finalidad de realizar circuitos combinacionales para llevar a cabo tareas que se hayan creado por necesidades que tengamos que cubrir, y es necesario que en el proceso de manejar las operaciones lógicas y la construcción de los circuitos combinacionales exista un diseño escrito en documento que represente lo que se va a construir; de este modo debemos representar los operadores que realizan una función con variables de forma grafica, que a su vez representen a las componentes eléctricas que son las que llevan dichas operaciones pero con pulsos eléctricos que normalmente toman dos valores; uno de 5 volts y otro de 0 volts; siendo el de 5 volts correspondiente con el valor de un 1 lógico y el de 0 volts correspondiente al de un 0 lógico.

Comenzaremos por comentar que existen tres formas de representar gráficamente a las operaciones entre variables y así formar los esquemas de los circuitos combinacionales que también ya gráficamente las operaciones son realizadas por las componentes electrónicas llamadas compuertas y es importante tomar en cuenta que dichas compuertas son representadas por símbolos eléctricos.

Una de las formas de representar a las compuertas está definida por las normas MIL STD americanas que son aprobadas y recomendadas por la ANSI (American National Standard, www.ansi.org) del IEEE (Institute of Electrical and Electronics Engineers, www.ieee.org). La segunda forma de representar a las compuertas gráficamente esta dada por las normas DIN (Instituto Alemán de Normalización, Deutsches Institut für Normung, www.din.de) de origen alemán. La tercera forma de realizar la representación es a través de las normas que establece la CEI (Comisión Electrotécnica Internacional, www.iec.ch) que también ANSI/IEEE reconocen y recomiendan para emplearse. Tanto la primera como la tercera forma de representar a las compuertas son las mas usadas y son estándares que han sido adoptados por la industria privada, y la industria militar lo utiliza para su documentación interna así como para sus publicaciones. A continuación mostraremos como son gráficamente las compuertas que representan a las operaciones lógicas vistas en nuestro punto 1.9, es decir las operaciones AND, OR, NOT, NAND, NOR, XOR y XNOR a través de las tres normas ya mencionadas.

1.10.1. Símbolo AND

La primera compuerta a representar será la de la operación lógica AND para dos variables de entrada, que se muestra en la figura 1.18.

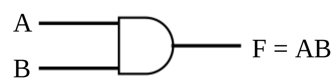


Figura 1.18. Símbolo eléctrico de la compuerta AND según las normas MIL STD.

De la figura 1.18, 1.19 y 1.20 vemos que se representa a una compuerta AND de dos entradas, las cuales se indican con las letras A y B. La salida o resultante de la operación con la letra F seguida de la operación booleana AND entre las dos variables de entrada.

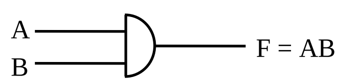


Figura 1.19. Símbolo eléctrico de la compuerta AND según las normas DIN.

De la figura 1.19 vemos que en realidad guarda cierto parecido a la figura 1.18, solo que cambian un poco en el tamaño siendo una mas pequeña que la otra, y de aquí nos damos cuenta como el hombre siempre ha tratado de homogenizar el conocimiento, ya que para ser honestos, el que existan solo tres formas de representar a las compuertas es poco en comparación de las tantas empresas en tantos países que se dedican al diseño de circuitos combinacionales y/o a la fabricación de componentes electrónicas, aunque es necesario comentar que las normas DIN ya son muy poco utilizadas, por lo que la mayoría de las veces encontramos con mas frecuencia el uso de las normas MIL STD y las CEI, eso implica una homogenización aun mayor.

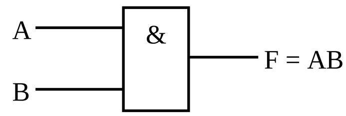


Figura 1.20. Símbolo eléctrico de la compuerta AND según las normas CEI.

Vemos en la figura 1.20 que según las normas CEI emplea rectángulos para la simbolización de las compuertas y dentro de él se muestra un símbolo en ingles que representa a la operación AND; así, para posteriores operadores el rectángulo no cambia grandemente solo internamente lo escrito en el, indicando que operación se realiza.

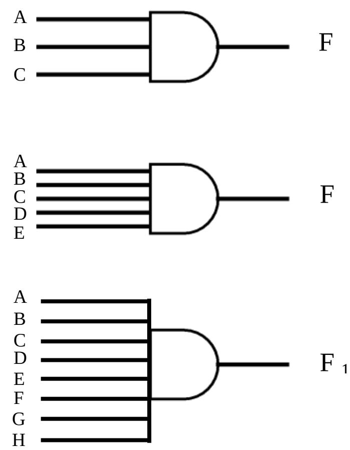


Figura 1.21. Símbolos eléctricos de la compuerta AND que cuentan con 3, 5 y 8 entradas según las normas MIL STD.

Observamos de la figura 1. 21 como son las compuertas AND cuando tienen mas de dos variables, el símbolo se mantiene como el mismo, solo se añaden la cantidad de entradas que se necesiten que como en la figura se tienen compuertas AND de 3, 5 y 8 variables, de esta ultima vemos que solo cuando la cantidad de variables es grande se alarga la línea que recibe las entradas, según se necesite, además a la salida se le coloco a la letra F un numero 1 como subíndice, esto con la intención de que sea diferenciada de la F que se utiliza como entrada.

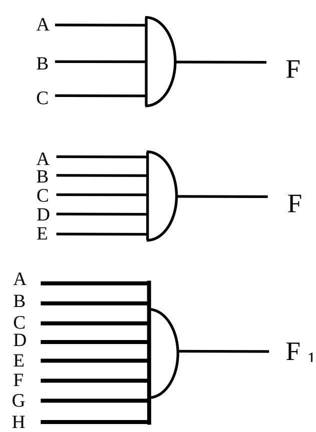


Figura 1.22. Símbolos eléctricos de la compuerta AND que cuentan con 3, 5 y 8 entradas según las normas DIN.

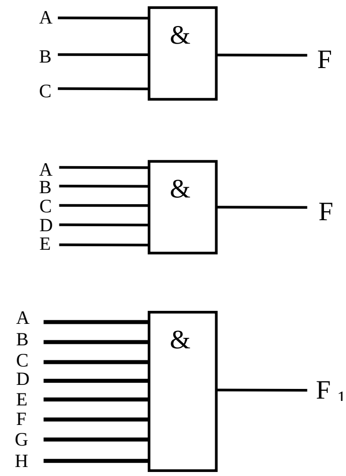


Figura 1.23. Símbolos eléctricos de la compuerta AND que cuentan con 3, 5 y 8 entradas según las normas CEI.

1.10.2. Símbolo OR

La compuerta OR es representada eléctricamente a continuación con figuras que se manejan con las mismas normas; MIL STD, DIN y CEI que se usaron para representar a la compuerta AND.

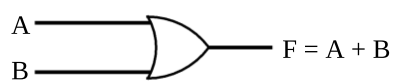


Figura 1.24. Símbolo eléctrico de la compuerta OR según las normas MIL STD.

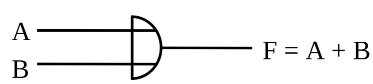


Figura 1.25. Símbolo eléctrico de la compuerta OR según las normas DIN.

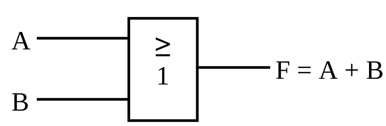


Figura 1.26. Símbolo eléctrico de la compuerta OR según las normas CEI.

En las figuras 1.24, 1.25 y 1.26 las representaciones de las compuertas OR son con dos variables de entrada y se indica con F la salida resultante y su operación lógica. También cabe señalar que de la figura 1.26 que es la representación de las normas CEI, dentro del rectángulo se escribe el signo matemático "igual o mayor que " y el numero 1, lo cual indica que la resultante al utilizar esta compuerta es 1, cuando al menos una entrada o ambas sean 1.

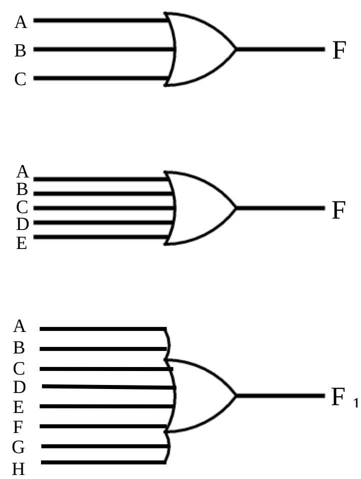


Figura 1.27. Símbolos eléctricos de la compuerta OR que cuentan con 3, 5 y 8 entradas según las normas MIL STD.

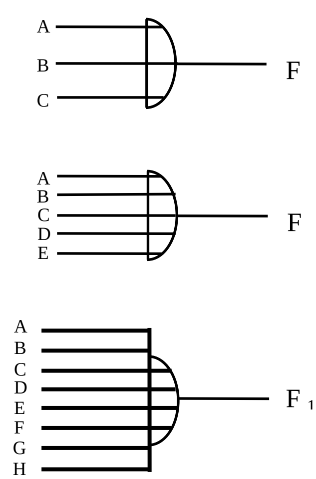


Figura 1.28. Símbolos eléctricos de la compuerta OR que cuentan con 3, 5 y 8 entradas según las normas DIN.

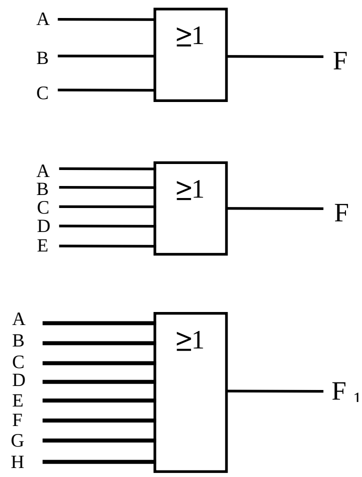


Figura 1.29. Símbolos eléctricos de la compuerta OR que cuentan con 3, 5 y 8 entradas según las normas CEI.

1.10.3. Símbolo NOT

De la misma forma, el símbolo para la operación lógica NOT se hace en las tres normas: MIL STD, DIN y CEI. En las tres se emplea un pequeño círculo en la terminal de salida para indicar la negación, es decir que la salida siempre será opuesta a la entrada. Aunque también en las normas CEI la negación se puede representar con un pequeño triangulo y no con el circulo, como aparece en la figura 1.32; además que dentro del rectángulo aparece el numero 1 para representar a su nombre.

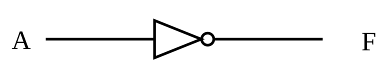


Figura 1.30. Símbolo eléctrico de la compuerta NOT según las normas MIL STD.

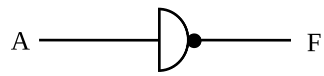


Figura 1.31. Símbolo eléctrico de la compuerta NOT según las normas DIN.

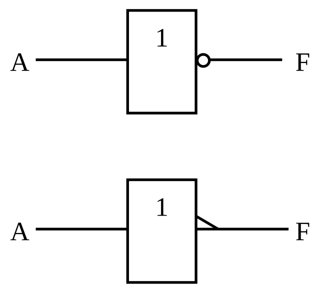


Figura 1.32. Dos representaciones eléctricas de la compuerta NOT según normas CEI.

1.10.4. Símbolo NAND

Nuestra siguiente compuerta a representar es la NAND, que haciendo la similitud con la operación lógica AND, basta agregar en la terminal de salida al circulo que representa a la operación lógica NOT, de esta misma forma las figuras eléctricas guardan esa misma similitud entre ellas y de igual forma en la norma CEI también en lugar del circulo se puede además usar un triangulo.

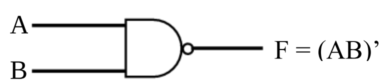


Figura 1.33. Símbolo eléctrico de la compuerta NAND según las normas MIL STD.

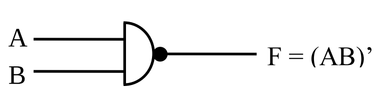


Figura 1.34. Símbolo eléctrico de la compuerta NAND según las normas DIN.

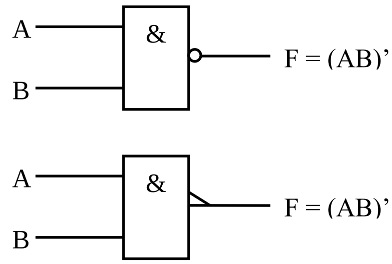


Figura 1.35. Dos símbolos eléctricos de la compuerta NAND según las normas CEI.

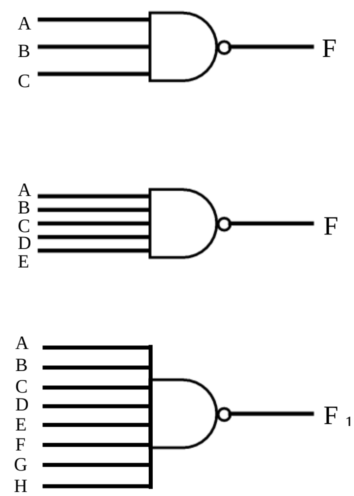


Figura 1.36. Símbolos eléctricos de la compuerta NAND que cuentan con 3, 5 y 8 entradas según las normas MIL STD.

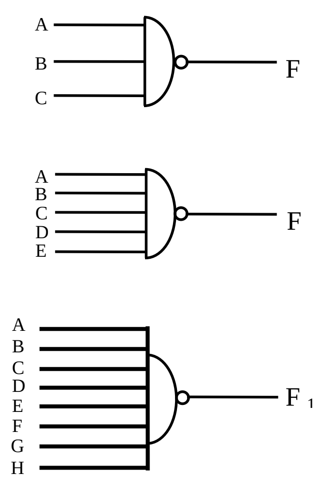


Figura 1.37. Símbolos eléctricos de la compuerta NAND que cuentan con 3, 5 y 8 entradas según las normas DIN.

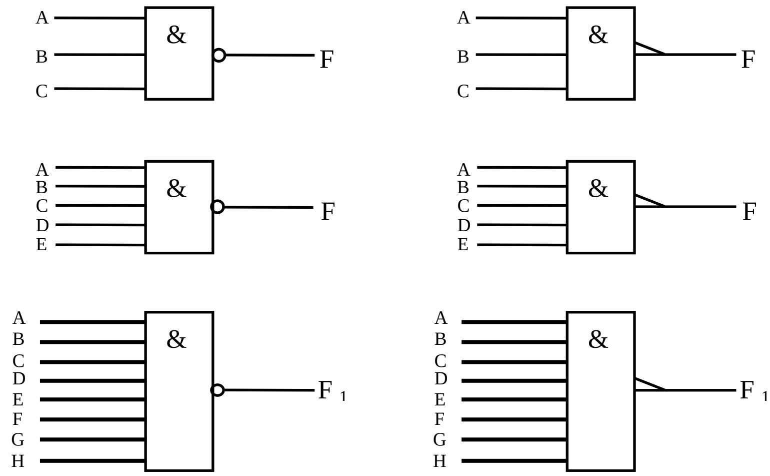


Figura 1.38. Dos grupos de símbolos eléctricos de la compuerta NAND que cuentan con 3, 5 y 8 entradas según las normas CEI.

1.10.5. Símbolo NOR

Para el caso de la compuerta NOR también se puede hacer la similitud con la operación lógica OR, de esta misma forma las figuras eléctricas guardan esa misma similitud entre ellas, es decir, que las representaciones de las compuertas OR son iguales a las NOR solo que estas ultimas tienen en la salida un circulo que indica la negación, y de igual forma en la norma CEI también en lugar del circulo se puede además usar el triangulo en esa misma posición.

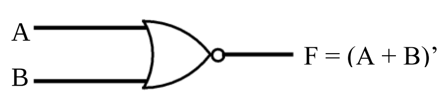


Figura 1.39. Símbolo eléctrico de la compuerta NOR según las normas MIL STD.

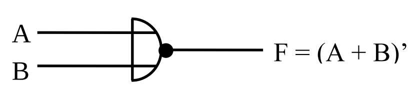


Figura 1.40. Símbolo eléctrico de la compuerta NOR según las normas DIN.

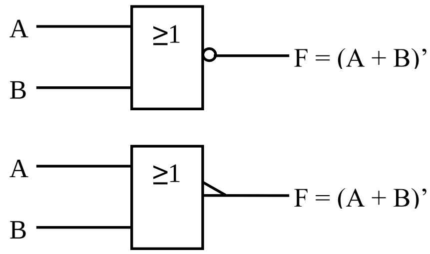


Figura 1.41. Dos símbolos eléctricos de la compuerta NOR según las normas CEI.

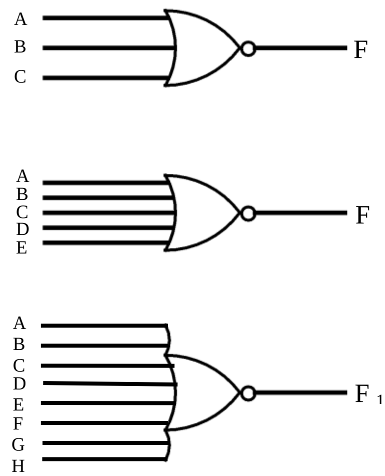


Figura 1.42. Símbolos eléctricos de la compuerta NOR que cuentan con 3, 5 y 8 variables de entrada, según la norma MIL STD.

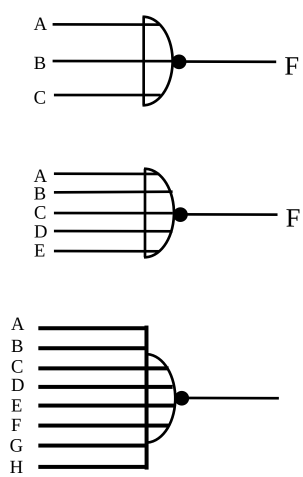


Figura 1.43. Símbolos eléctricos de la compuerta NOR que cuentan con 3, 5 y 8 entradas según las normas DIN.

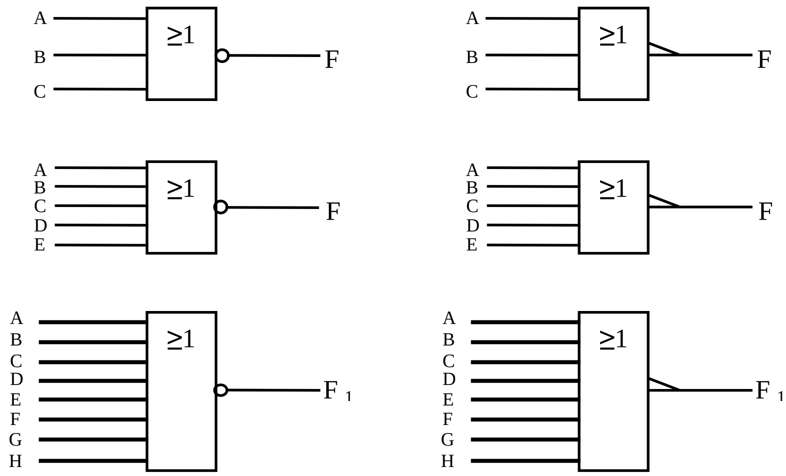


Figura 1.44. Dos grupos de símbolos eléctricos de la compuerta NOR que cuentan con 3, 5 y 8 entradas según las normas CEI.

1.10.6. Símbolo XOR

Los símbolos representativos de una compuerta XOR (OR-exclusiva) también están definidos para las normas MIL STD, DIN y CEI; los cuales aparecen en las figuras 1.45, 1.46 y 1.47 respectivamente.

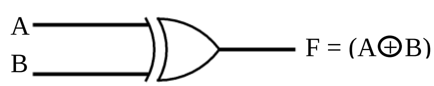


Figura 1.45. Símbolo eléctrico de la compuerta OR-exclusiva (XOR) según normas MIL STD.

Es importante señalar que esta compuerta esta construida para ser trabajada con dos variables de entrada, por lo que en sus símbolos eléctricos solo se ejemplifican con dos terminales de entrada.

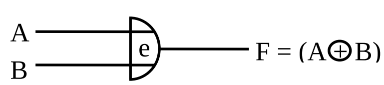


Figura 1.46. Símbolo eléctrico de la compuerta OR-exclusiva (XOR) según normas DIN.

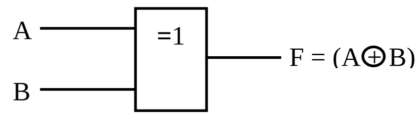


Figura 1.47. Símbolo eléctrico de la compuerta OR-exclusiva (XOR) según normas CEI.

1.10.7. Símbolo XNOR

Sabiendo que la relación que existe entre la OR-exclusiva y la NOR-exclusiva es que la ultima es complemento de la primera, es decir, que se diferencian porque la NOR-exclusiva es igual que la OR-exclusiva solo que con una función NOT al final. Por lo que las representaciones son las mismas pero con el correspondiente circulo que indica la inversión del resultado. A continuación sus representaciones en tres normas que venimos trabajando, figuras 1.46, 1.47 y 1.48.

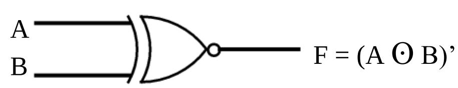


Figura 1.48. Símbolo eléctrico de la compuerta NOR-exclusiva (XNOR) según normas MIL STD.

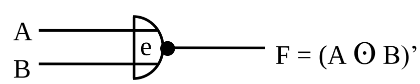


Figura 1.49. Símbolo eléctrico de la compuerta NOR-exclusiva (XNOR) según normas DIN.

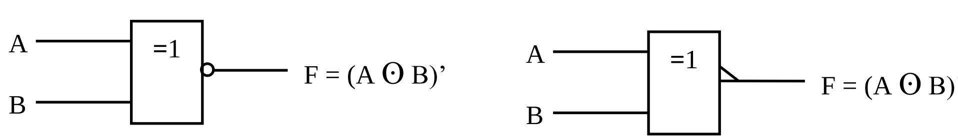


Figura 1.50. Símbolo eléctrico de la compuerta NOR-exclusiva (XNOR) según normas CEI.

CAPITULO 2

CARACTERÍSTICAS DE ALGUNAS DE LAS TÉCNICAS DE REDUCCIÓN EN CIRCUITOS COMBINACIONALES

2.1 Introducción

En este capítulo se hablará de las características de algunas de las técnicas de reducción para sistemas combinacionales, las cuales emplearemos en el Capítulo 4 para tratar de reducir a los sistemas digitales que se diseñaron en el capitulo 3.

Las técnicas de reducción para sistemas digitales son: el álgebra de Boole la cual maneja teoremas, principios, axiomas, postulados y leyes para poder reducir los sistemas digitales. Esta álgebra de Boole es llamada así por el matemático, George Boole (1854), lógico y matemático, quien enunciara esta álgebra en un escrito llamado "*Investigación sobre las leyes del pensamiento".* Observaremos posteriormente que esta técnica funciona con un procedimiento completamente matemático, ya que en sus orígenes era utilizada exclusivamente en el área matemática, aunque con el tiempo fue evolucionando hasta llegar a ser utilizada para el diseño y la reducción de sistemas digitales. Habrá que mencionar además, que este sistema es un sistema algebraico, las proposiciones lógicas se indican por símbolos y pueden relacionarse por medio de operadores matemáticos abstractos que corresponden a las leyes de la lógica. Además esta álgebra es fundamental en el diseño de los modernos ordenadores o computadores, aunque no es la única técnica, esta fue base de muchas otras.

Otra técnica de reducción que trataremos en este capítulo serán los minitérminos y maxitérminos que no es más que una propiedad derivada del álgebra de Boole. Los minitérminos al igual que los maxitérminos son las posibles combinaciones que se dan entre los términos que se manejaran, siendo los minitérminos aquellos términos que se combinan por medio de un operador AND o sea un "•", y los maxitérminos son aquellos términos que se combinan por un operador OR o sea " + ".

Además, conoceremos algunas técnicas que tienen un aspecto más grafico que matemático y que una vez que podamos entenderlas (que es uno de los objetivos de esta tesis), nos daremos cuenta de su fácil manejo y rapidez con que se hace una posible reducción de nuestro sistema digital. Un caso muy claro de estas técnicas es la llamada mapas de Karnaugh, en la que según la cantidad de señales será el tamaño del mapa a utilizar, ya que existen mapas para 2, 3, 4, 5, y 6 señales, siendo 6 señales la ultima cantidad de variables que utilizaremos.

Otras técnicas que tienen también formas un poco graficas y que veremos son el método de tabulación y las técnicas de Quince – Mc. Cluskey, ya que el método del mapa de Karnaugh es muy conveniente para la simplificación siempre y cuando el numero de señales a manejar no exceda de cinco o seis, pues se vuelve imposible lidiar cuando estas aumentan ya que el volumen del mapa también aumenta e impide una selección razonable. Además será interesante conocer algunas otras técnicas de reducción como la de salida múltiple y de algoritmos, que quizás no sean muy conocidas pero que nos ayudarán a tener un mayor conocimiento de la variedad de las técnicas de reducción y de cómo poder utilizarlas. Cabe mencionar que también existe software basado en estos últimos algoritmos que podrán ayudar a reducir circuitos combinacionales, como por ejemplo, Karnaugh minimizer, logisim, QMC, Boole-deusto, Reductio, entre otros.

2.2 Álgebra booleana

El álgebra booleana es una técnica matemática que se usa para resolver problemas de naturaleza lógica. Esta álgebra de Boole es llamada así por quien la introdujo: George

Boole (1854) matemático ingles, que crea esta matemática para que se pudiera aplicar a problemas de lógica deductiva, enunciada en un escrito llamado "Investigación sobre las leyes del pensamiento". Fue hasta 1938 que esta álgebra estuvo solo circunscrita al área matemática ya que en este año Claude Shannon, un científico de los laboratorios Bell analizo las características de esta álgebra, pudiendo utilizarla para analizar redes de contactos múltiples como los empleados en la telefonía.

Con la aparición de las computadoras el álgebra de Boole se ha vuelto una herramienta fundamental utilizada por ingenieros y matemáticos para auxiliarse en el diseño lógico y en este trabajo de tesis no será la excepción, ya que esta vez ayudará para que estudiantes de ingeniería puedan ayudarse de esta técnica de reducción para el diseño de circuitos combinacionales.

Aunque el álgebra de Boole fue utilizada para realizar análisis mas prácticos por Shannon, pero ya Huntington en 1904 formulo postulados o axiomas que ayudaron a definir mas claramente este tipo de álgebra y que en la actualidad han pasado a formar parte de un grupo de teoremas, principios y postulados que definen de una forma más completa el álgebra desarrollada por Boole.

El álgebra Booleana en un principio describe proposiciones cuya respuesta solo puede ser verdadera o falsa.

En computación se utiliza para analizar el estado en el que se encuentran los circuitos, pudiendo ser 1 y 0. De los estados 1 y 0 que son estados en los que pueden encontrarse los circuitos digitales, refiriéndolos a los números binarios podremos ver que se vuelven de una naturaleza completamente binaria.

Por ultimo, mencionaremos que esta álgebra de Boole es un sistema algebraico de dos o mas elementos y dos operadores binarios " + " (OR) y " • " (AND).

2.2.1 Teoremas, principios y postulados

El álgebra de Boole esta compuesta por teoremas, principios, postulados, leyes y axiomas; donde los postulados son suposiciones, de los cuales se deducen otras leyes y teoremas que componen el álgebra de Boole. De manera similar un axioma, en lógica y matemáticas es un principio básico que es asumido como verdadero sin recurrir a demostración alguna. En el caso de esta tesis los términos axioma y postulado se utilizaran como sinónimos, para no enfrascarse en una filosofía matemática que pueda llevar a la confusión. Ahora, un teorema es una regla que concierne a una relación fundamental entre las variables booleanas, además que tanto a los postulados como a los teoremas podemos aplicar algunos principios, que se encuentran en el álgebra de Boole.

En primera instancia se hablara de un principio que se aplica tanto en los postulados como en los teoremas y este es el principio de dualidad, el cual establece que las expresiones algebraicas deducibles de los postulados del álgebra de Boole permanecen validas si se intercambian los operadores "+" y "•", y se remplaza unos por ceros y ceros por unos. En la tabla 2.1 y 2.2 se nos muestra aplicado el principio de dualidad, ya que se acomodaron frente a los postulados y teoremas el dual de cada uno de ellos.

A continuación, en la tabla 2.1 mencionaremos los postulados y en la tabla 2.2 los teoremas que utilizaremos.

No. de Postulado	Postulado	Postulado Dual
Postulado 1	$X = 0$ ó $X = 1$	$X = 1$ ó $X = 0$
Postulado 2	$X + 0 = X$	$X \bullet 1 = X$
Postulado 3	$X + Y = Y + X$	$X \bullet Y = Y \bullet X$
Postulado 4	$X \bullet (Y + Z) = X \bullet Y + X \bullet Z$	$X + (Y \bullet Z) = (X + Y) \bullet (X + Z)$
Postulado 5	$X + X' = 1$	$X \bullet X' = 0$

Tabla 2.1. Postulados del álgebra de Boole.

No. De Teorema	Teorema	Teorema Dual
Teorema 1	$X + X = X$	$X \bullet X = X$
Teorema 2	$X + 1 = 1$	$X \bullet 0 = 0$
Teorema 3	$(X')' = X$	
Teorema 4	$X + (Y + Z) = (X + Y) + Z$	$X \bullet (Y \bullet Z) = (X \bullet Y) \bullet Z$
Teorema 5	$(X + Y)' = X' \bullet Y'$	$(X \bullet Y)' = X' + Y'$
Teorema 6	$X + (X \bullet Y) = X$	$X \bullet (X + Y) = X$
Teorema 7	$F(X_1,X_2,\ldots,X_n) = X_1 \bullet F(1,X_2,\ldots,X_n) + X'_1 \bullet F(0,X_2,\ldots,X_n)$	$F(X_1,X_2,\ldots,X_n) = [X_1+F(0,X_2,\ldots,X_n)] \bullet [X'_1 + F(1,X_2,\ldots,X_n)]$

Tabla 2.2. Teoremas del álgebra de Boole.

Antes de comenzar a tratar las funciones booleanas es conveniente mencionar que para trabajar con ellas se conozca la *prioridad de los operadores* para la evaluación de las expresiones booleanas, de las cuales se constituyen las funciones de Boole, en listando de forma progresiva según la prioridad del operador como de mayor prioridad el numero 1 y diremos de ultima prioridad el que este en listado como el numero 4.

1.- El paréntesis “( )”.

2.- NOT “ ′ ”.

3.- AND “ • ”.

4.- OR “ + ”

Por ejemplo, al evaluar una expresión algebraica como (X • Z)´, quedaría la evaluación primero dentro del paréntesis y luego se complementaria el resultado obtenido quedando X´ + Z´. De tal forma, que se puede observar la evaluación, en primera instancia, en el complemento tanto de X como de Z, para que posteriormente, al resultado obtenido, se le aplique la operación OR.

2.2.2 Funciones booleanas

Del subtema lógica binaria vimos que una variable binaria puede tener dos únicos valores que son: un 1 o un 0, pues bien, las funciones booleanas están compuestas por variables binarias, las cuales se encuentran en conjunción con los operadores lógicos o también llamados algunas veces operadores binarios, de los que ya también tenemos conocimiento, por supuesto sin olvidar el signo de igual. Todo lo anterior para poder obtener el valor de la función que será 1 o 0. Por lo que a continuación se muestra una función Booleana.

$$F_1 = X \bullet Y'$$

Llevando a cabo las operaciones lógicas que se presentan en la función y sustituyendo los dos posibles valores que pueden tomar tanto X como Y, obtenemos que solo cuando X = 1 y Y´ = 1 la función $F_1 = 1$, de cualquier otro modo, o en cualquier otra distinta combinación de las que podemos obtener en la función, el valor de $F_1 = 0$. Como se acaba de mencionar, tenemos que ir evaluando los valores que tome tanto Y como X, y para evitar precisamente el estar escribiendo la función una y otra vez para sustituir los diferentes valores, en diferentes combinaciones con el temor de poder equivocarnos, es recomendable utilizar una herramienta que nos facilitara el manejo y la observación de los distintos resultados que arroje nuestra función siendo esta la llamada tabla de verdad.

La elaboración de una tabla de verdad para la función F_1 la realizaremos tomando en cuenta que debemos conocer el numero de combinaciones que llevara nuestra tabla y esta será de 2^n donde; n es el numero de señales de entrada que tenga nuestra función y el 2 por el sistema numérico que estamos utilizando que es de base 2. El número de las combinaciones que obtengamos de unos y ceros para cada uno de los renglones de la tabla se contaran desde 0 a $2^n - 1$.

Entonces el numero de renglones para nuestra tabla de la F_1 será $2^2 = 4$ y el numero que llevara cada una de las combinaciones será de $2^2 - 1 = 3$ así nuestra tabla de verdad queda como la tabla 2.3.

No. de Renglones	Combinaciones	X	Y	$F_1 = X \bullet Y'$
1	0	0	0	0
2	1	0	1	0
3	2	1	0	1
4	3	1	1	0

Tabla 2.3. Tabla de verdad de $F_1 = X \bullet Y'$

Analizando la tabla 2.3. observamos que el numero de renglones es similar al numero de combinaciones, ya que mientras existen 4 renglones, el numero de combinaciones es también de $4 \neq 2^n - 1$, iniciando desde el 0 y llegando hasta el numero 3, ¿por qué de esta forma?, ¿por qué 0 a $2^n - 1$ será el numero de combinaciones?, la respuesta será que la tabla de verdad deberá guardar cierta coincidencia entre el numero de combinaciones con la combinación de 1´s y 0´s que se forma dentro de la tabla de verdad, o de cualquier otra, de tal manera que será el numero de la combinación y no el numero de combinaciones. Analicemos los renglones observando que el valor de la variable X como el valor de Y son la representación binaria de la combinación correspondiente, y apoyándonos de la tabla 1.1 del capítulo anterior observaremos para un mejor entendimiento lo siguiente:

No. de Renglón Combinación

1 → 0 →

X	Y	$F_1 = X \bullet Y'$
0	0	0

Numero Binario a convertir → 0 0

1×2^8	1×2^7	1×2^6	1×2^5	1×2^4	1×2^3	1×2^2	0×2^1	0×2^0
256	128	64	32	16	8	4	0	0

Se colocan solo los valores que se encuentran sobre los unos ya que serán estos los números que se toman en cuenta. → 0 0

Figura 2.1. Conversión del número binario 00 a decimal.

Quedando entonces que el numero binario 00 = 0 + 0 = 0 decimal, que es nuestra primera combinación.

En el siguiente caso toca el turno al segundo renglón de nuestra tabla.

No. de Renglón Combinación

2 1

X	Y	$F_1 = X \bullet Y'$
0	1	0

Numero Binario a convertir 0 1

1×2^8	1×2^7	1×2^6	1×2^5	1×2^4	1×2^3	1×2^2	0×2^1	1×2^0
256	128	64	32	16	8	4	0	1

Se colocan solo los valores que se encuentran sobre los unos ya que serán estos los números que se sumaran. 0 1

Figura 2.2. Conversión del numero binario 01 a decimal.

Quedando entonces que el numero binario 01 = 0 + 1 = 1 decimal, que es nuestra segunda combinación. Posteriormente para el tercer renglón se tiene:

No. De Renglón Combinación

3 2

X	Y	$F_1 = X \bullet Y'$
1	0	1

Numero Binario a convertir 1 0

1×2^8	1×2^7	1×2^6	1×2^5	1×2^4	1×2^3	1×2^2	1×2^1	0×2^0
256	128	64	32	16	8	4	2	0

Se colocan solo los valores que se encuentran sobre los unos ya que serán estos los números que se sumaran. 2 0

Figura 2.3. Conversión del numero binario 10 a decimal.

Quedando entonces que el numero binario 10 = 2 + 0 = 2 decimal, que es nuestra tercera combinación.

Para el último renglón queda que:

No. de Renglón   Combinación

4 —— 3

X	Y	$F_1 = X \bullet Y'$
1	1	0

Numero Binario a convertir

1 1

1×2^8	1×2^7	1×2^6	1×2^5	1×2^4	1×2^3	1×2^2	1×2^1	1×2^0
256	128	64	32	16	8	4	1	1

Se colocan solo los valores que se encuentran sobre los unos ya que serán estos los números que se sumaran.

2 1

Figura 2.4. Conversión del numero binario 11 a decimal.

Quedando entonces que el numero binario 11 = 2 + 1 = 3 decimal, que es el numero de la combinación.

Con la finalidad de que el lector observe que las diferentes combinaciones de 1´s y 0´s que se encuentran en cualquier tabla de verdad serán coincidentes con el numero de combinaciones que se obtengan con el rango de 0 a $2^n – 1$.

En el caso antes mencionado se realizo un análisis para una función F_1 que tenía dos variables, esto quiere decir para n = 2, en 2^n y $2^n – 1$, numero de renglones y numero de combinaciones respectivamente, entonces veremos otros ejemplos como la siguiente función:

$$F_2 = X + YZ$$

Para la función F_2 la cantidad de señales es de 3, por lo cual la tabla de verdad varia también, obteniendo de 2^n que; $2^{(3)} = 8$ renglones para asignarles a las tres variables, así como las combinaciones de 1´s y 0´s para cada una de los renglones contando desde 0 hasta $2^n - 1$, o sea; $2^{(3)} - 1 = 7$, por lo que los números decimales en cada renglón serán desde 0 a 7, quedando como en la tabla 2.4.

X	*Y*	*Z*	$F_2 = X + YZ$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Tabla 2.4. Tabla de verdad de $F_2 = X + YZ$

De la tabla 2.4 se observa que de los tres primeros renglones donde X, Y y Z son; 000, 001 y 010, $F_2 = 0$ y el resto harán que $F_2 = 1$

Para la función $F_3 = (X + Y) (X + Z)$ también aparecen tres señales (x, y, z), por lo tanto existirán ocho renglones y 7 combinaciones; quedando su tabla de verdad como aparece en la tabla 2.5.

X	Y	Z	$F_3 = (X + Y)(X + Z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Tabla 2.5. Tabla de verdad de $F_3 = (X + Y)(X + Z)$

En la tabla 2.5 se observa que al igual que en la tabla 2.4. para los tres primeros renglones $F_3 = 0$ y para todos los demás $F_3 = 1$, por lo que la función F_2 y F_3 son iguales aunque sus expresiones algebraicas son distintas. De lo anterior decimos que para cada función se puede tener una expresión algebraica distinta que la represente.

Una vez que vimos que una función puede representarse con diferentes expresiones algebraicas, podemos entonces darle a una función un manejo algebraico para que esta tenga otra expresión algebraica que la represente, haciendo lo posible de que sea lo mas pequeña o simple, siendo la intención, encontrar una reducción, aquí es en donde entonces aplicaremos nuestros teoremas principios y postulados.

2.2.3. Manejo de los teoremas, principios y postulados

Cabe recordar que las compuertas lógicas son la representación de las operaciones de Boole, por lo que una función que tenga poca cantidad de operaciones debe emplear también un menor numero de compuertas, en donde en cada literal representa una entrada en una compuerta y cada termino a la propia compuerta lógica (la forma grafica se observara en el capitulo 3).

En la simplificación de las funciones de Boole no se tiene un procedimiento definido para realizar esto, por lo que solo se tiene que tratar de simplificar lo más que se pueda aplicando los teoremas y postulados de Boole y obtener el menor numero de literales para eso veremos varios ejemplos; en donde se aplican ya los teoremas y postulados.

Ejemplo 2.1: Simplificar al menor numero de literales la siguiente función.

$$F = X'Y'Z + X'Y\,Z + X'Y\,Z' + X\,Y'Z + X\,Y\,Z$$

Observamos de la función que hay términos semejantes en los que solo difiere una variable, como lo es $X'Y'Z$ con $XY'Z$ y $X'YZ$ con $X'YZ'$, por lo que los organizamos de la siguiente manera.

$$F = X'Y'Z + X\,Y'Z + X'Y\,Z + X'Y\,Z' + X\,Y\,Z$$

Aplicando ahora el postulado 4: $X(Y + Z) = XY + XZ$

$$F = Y'Z(X' + X) + X'Y(Z' + Z) + XYZ$$

Aplicando el postulado 5: $X + X' = 1$

$$F = Y'Z(1) + X'Y(1) + XYZ$$

Reorganizando y aplicando el postulado 4: $X + (YZ) = (X + Y)(X + Z)$

$$F = (Y + Y')(Z + ZX) + X'Y$$

Por postulado 5: $Y + Y' = 1$ y simplificando a Z

$F = Z(1 + X) + X'Y$

Por teorema 2: $X + 1 = 1$ queda el resultado final

$F = Z + X'Y$

Ejemplo 2.2: Simplificar al menor número de literales la siguiente función.

$F = X Y'Z' + X Y Z + X'Y Z + X Y Z'$

Acomodando términos semejantes y simplificando

$F = X Y'Z' + X Y Z' + X Y Z + X'Y Z$

$F = XZ'(Y' + Y) + YZ(X + X')$

Aplicando por ultimo el postulado 5: $X + X' = 1$

$F = XZ' + YZ$

Ejemplo 2.3: Simplificar al menor numero de literales la siguiente función

$F = X'Y'Z' + X Y'Z + X Y'Z' + X'Y Z' + X Y Z'$

Aplicando el postulado 4: $X(Y + Z) = XY + XZ$

$F = X'Z'(Y' + Y) + XYZ' + XY'(Z + Z')$

Aplicando el postulado 5: $X + X' = 1$

$F = X'Z' + XYZ' + XY'$

Aplicando ahora el postulado 4: $X + (YZ) = (X + Y)(X + Z)$

$F = (X' + X)(Z' + YZ') + XY'$

Aplicando el nuevamente postulado 5: $X + X' = 1$

$F = Z' + YZ' + XY'$

Aplicando el postulado 4 y teorema 2:

$F = Z'(Y + 1) + XY'$

$F = Z' + XY'$

Ejemplo 2.4: Simplificar la siguiente función
$F = X'Y'Z + X'Y Z + X Y'Z + X Y Z$

Aplicando el postulado 4:
$F = X'Z(Y + Y') + XZ(Y' + Y)$

Aplicando postulado 5:
$F = X'Z + XZ$

Nuevamente el postulado 4:
$F = (X + X')Z$

Postulado 5: $F = Z$

2.3. Minitérminos y maxitérminos

Los minitérminos y maxitérminos son uno de los métodos quizás más conocidos o al menos más escuchados por los estudiantes de ingeniería y a la vez un método que no es complicado de entender, para lo cual también tenemos que ocupar algunos de los conceptos antes enunciados en esta tesis.

Una expresión algebraica puede ser representada con una tabla de verdad especifica y cualquier otra expresión que tenga la misma tabla de verdad es equivalente a la primera. Por esto se tiene que, los valores de salida de la expresión que ocupa los renglones de la tabla de verdad están asignados de acuerdo con el renglón que ocupan y se les da el nombre de minitérminos, por tanto como ya se había mencionado una función tendrá $2^n - 1$ combinaciones y así mismo una cantidad igual de minitérminos que la representaran de manera única.

Habrá que mencionar como definición entonces que; las diferentes combinaciones de variables binarias por medio del operador AND son llamados minitérminos, ya sean éstas en su forma complementada o la no complementada. Lo observaremos así de la siguiente manera:

X	*Y*	*Z*	*Minitérmino*
0	0	0	$X'Y'Z' = m_0$
0	0	1	$X'Y'Z = m_1$
0	1	0	$X'YZ' = m_2$
0	1	1	$X'YZ = m_3$
1	0	0	$XY'Z' = m_4$
1	0	1	$XY'Z = m_5$
1	1	0	$XYZ' = m_6$
1	1	1	$XYZ = m_7$

Tabla 2.6. Minitérminos de una función de tres variables (X,Y y Z)

Como aparece en la tabla 2.6 en el manejo de los minitérminos una variable tildada significa que esta tiene el valor de un 0 y la no tildada un valor de 1, además de que el operador entre estas variables es el AND, así entonces, existen otro tipo de combinaciones que de manera similar a las combinaciones que nos dan origen a los minitérminos, se encuentran definidas por variables tildadas que en este caso implican un valor de 1 y las no tildadas un valor 0, y siendo el operador OR el que se encuentre entre ellas, hablamos de los maxiterminos, que no son mas que una expresión que complementa a un correspondiente termino mínimo. Observemos en la tabla 2.7 a los maxiterminos y su similitud con los miniterminos.

X	*Y*	*Z*	*Maxiterminos*
0	0	0	$X+Y+Z= M_0$
0	0	1	$X+Y+Z'= M_1$
0	1	0	$X+Y'+Z= M_2$
0	1	1	$X+Y'+Z'= M_3$
1	0	0	$X'+Y+Z= M_4$
1	0	1	$X'+Y+Z'= M_5$
1	1	0	$X'+Y'+Z= M_6$
1	1	1	$X'+Y'+Z'= M_7$

Tabla 2.7. Maxitérminos de una función de tres variables (X,Y y Z)

Después de ver las dos tablas observamos como el numero de combinaciones para los minitérminos y maxitérminos es de $2^n - 1$ o sea el mismo y como una es el complemento de otra cualquier función puede ser representada o escrita por un producto de minitérminos o como la suma de maxitérminos.

Ejemplo 2.5: Desarrollar en minitérminos y maxitérminos la funcion F= XY + X'YZ + XYZ'

Para poder representar la función primeramente en forma de minitérminos se deberá añadir literal Z que le hace falta al primer término XY para que contenga a las tres literales de los otros dos términos, para eso utilizamos el Postulado 5 X + X' = 1, de la siguiente manera:

F = XY(Z+Z') + X'YZ + XYZ'

F= XYZ + XYZ' + X'YZ + XYZ'

Donde se observa que XYZ' se repite y por el Teorema 1 X +X = X se puede reducir a:

F= XYZ + XYZ' + X'YZ

Donde ya los 3 términos de la función contienen las tres literales y se procede a relacionarla con la tabla 2.6, dando una nueva tabla que se encuentra en tabla 2.8.

F= XYZ + XYZ' + X'YZ

X	*Y*	*Z*	*Minitérmino*
0	0	0	X'Y'Z'= m_0
0	0	1	X'Y'Z= m_1
0	1	0	X'YZ'= m_2
0	1	1	X'YZ= m_3
1	0	0	XY'Z'= m_4
1	0	1	XY'Z= m_5
1	1	0	XYZ'= m_6
1	1	1	XYZ= m_7

Tabla 2.8 Relación con minitérminos

Por lo que finalmente la función F queda expresada en función de los minitérminos:

$F = m_3 + m_6 + m_7$ o de otra manera mas compacta y valida seria $F = \sum m(3,6,7)$, esto con la intención de que sea manejable de una manera mas económica.

Así mismo para realizar el desarrollo en maxitérminos de la función necesitaremos del resto de los minitérminos que no aparecen en la función F, pero que invirtiendo a las variables como se mostro en la tabla 2.7.

Función de Minitérminos $F = m_3 + m_6 + m_7$

Nueva función con los Minitérminos restantes $F = m_0 + m_1 + m_2 + m_4 + m_5$

Si se obtiene el complemento de nuestra nueva función quedaría:

$F' = (m_0 + m_1 + m_2 + m_4 + m_5)' = F = M_0 \cdot M_1 \cdot M_2 \cdot M_4 \cdot M_5$ y al igual que en los minitérminos los maxitérminos se pueden representar de una manera compacta y es como sigue:

$F = \prod M(0,1,2,4,5)$.

Algo muy importante que debemos observar de los minitérminos y los maxitérminos es que uno es el complemento del otro (tratándose de la misma función), ya que como observamos del caso anterior por ejemplo en el desarrollo de los minitérminos aparece el minitérmino m_3, entonces el maxitérmino M_3 no aparece en el desarrollo de los maxitérminos, de hecho en el caso anterior se realizo un desarrollo de maxitérminos a partir de un desarrollo de minitérminos, si hacemos lo contrario y desarrollamos los minitérminos a partir de los maxitérminos encontraríamos lo siguiente y que se dejaría al lector comprobarlo, con otros ejemplos:

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

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

Ejemplo 2.6: Desarrollar en minitérminos y maxitérminos la función $F = X' + YZ'$

Para iniciar el desarrollo, debemos observar que hacen falta literales, por lo que se procederá a agregarlas por medio de la utilización del Postulado 5; $X + X' = 1$ de la siguiente manera:

$F = X'(Y+Y')(Z+Z') + (X+X')\ YZ'$

$F = X'Y+X'Y'(Z+Z')+XYZ'+X'YZ'$

$F = X'YZ+X'YZ'+X'Y'Z+X'Y'Z'+XYZ'+X'YZ'$

Utilizamos el Teorema 1 ($X+X=X$) ya que se repite $X'YZ'$

$F = X'YZ+X'YZ'+X'Y'Z+X'Y'Z'+XYZ'$

Se recurre a la tabla 2.6 para buscar en ella los términos mínimos de la función y saber que numero de minitérmino es, quedando:

$F = m_3 + m_2 + m_1 + m_0 + m_6$ ó $F = \sum m(0,1,2,3,6)$ y

$F = M_4 \cdot M_5 \cdot M_7$ ó $F = \prod M(4,5,7)$

Ejemplo 2.7: Realice el desarrollo en maxitérminos y minitérminos de $F = (X'+Y)(X'+Z')$

En este caso utilizaremos el Postulado 5; ($X \cdot X' = 0$) para agregar las literales faltantes

$F = (X'+Y+ZZ')(X'+YY'+Z')$

$F = (X'+Y+Z)(X'+Y+Z')(X'+Y+Z')(X'+Y'+Z')$

Se simplifican los términos iguales

$F = (X'+Y+Z)(X'+Y+Z')(X'+Y'+Z')$

Se buscan en este ejemplo los términos máximos en primera instancia por lo que se utilizara la tabla 2.7.

$F = M_4 \cdot M_5 \cdot M_7$ ó $F = \prod M(4,5,7)$ y

$F = m_0 + m_1 + m_2 + m_3 + m_6$ ó $F = \sum m(0,1,2,3,6)$

2.4. Mapas de Karnaugh

Dentro de las técnicas de reducción que se trataran en esta tesis se encuentran los Mapas de Karnaugh, que si bien no es la primera técnica que se empezó a tratar aquí, si es la primera que logra dar un proceso viable, estandarizado y sistemático, que nos guía hacia la respuesta correcta, ya que el álgebra booleana o de la conmutación es una herramienta que utilizamos para este fin y es verdad que nos sirve para dar soluciones a la simplificación de funciones, solo que de una manera mas arbitraria, dejando a la experiencia y criterios propios del interesado, pero que nos hace la tarea mas difícil cuando se tienen varias señales de entrada. Siendo así por el contrario los mapas de Karnaugh, ya que estos logran ayudarnos a reducir funciones de hasta seis variables de una manera mucho más eficaz.

Es importante mencionar que si uno mira por primera vez a los mapas de Karnaugh, a simple vista pudieran no decirnos nada, y por esto se considera que una breve explicación para saber de donde surgen los mapas nos ayudara a entender que estos se originan de los conceptos de tablas de verdad, minitérminos y diagramas de Venn, conceptos que ya manejamos con anterioridad en este escrito.

De esta manera iniciaremos diciendo que los mapas se originan del manejo de los diagramas de Venn y como este conocimiento matemático se va convirtiendo en otro y así pasar a convertirse en los mapas de karnaugh. Lo anterior se mostrara en las figuras 2.5 a 2.16.

Comenzamos con un conjunto universal como vemos en la figura 2.5 con dos subdivisiones o subconjuntos A y B, que en realidad esta constituido así por el hecho de que se manejaran dos variables, lo que nos llevara a realizar la deducción grafica de un mapa de Karnaugh de dos variables.

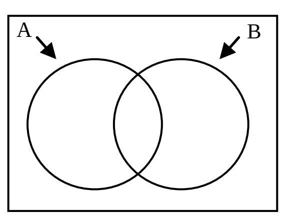


Figura 2.5. Diagrama básico de Venn

Mostrando la intersección de las subdivisiones en la figura 2.6 observamos que las variables ajenas a estas subdivisiones o sea que no están contenidas en A y B, son A' y B', además de las intersecciones AB', AB y A'B.

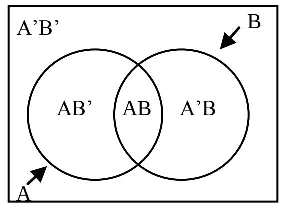


Figura 2.6. Intersecciones A'B', AB', AB y A'B

Así en la figura 2.7 llamamos a estas intersecciones minitérminos m_0, m_2, m_3 y m_1 sabiendo de antemano que estas intersecciones no son mas que otra cosa que la serie de combinaciones de una tabla de verdad de dos variables A y B, y que cada uno de los renglones de esta tabla de verdad equivalen a un minitérmino de ahí que se observe y se decida hacer el cambio, quedando de la siguiente manera.

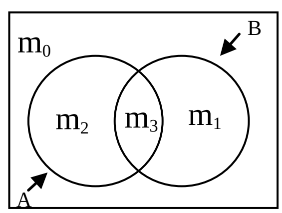


Figura 2.7. Diagrama básico de Venn con minitérminos

Dando un espacio regular a cada uno de los minitérminos nos queda una forma como la mostrada en la figura 2.8, debiendo observar cuidadosamente que las áreas adyacentes en el diagrama de Venn lo son de la misma manera en esta nueva forma que ha tomado el conjunto universal con el que iniciamos, donde también se señala con corchetes a las subdivisiones A y B, además de las intersecciones mostradas en la figura 2.7.

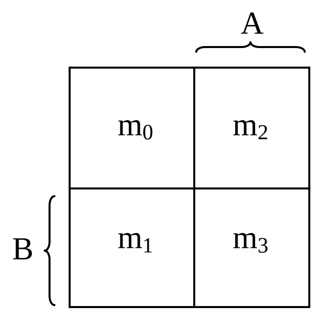


Figura 2.8. Mapa de Karnaugh con minitérminos

En la siguiente figura la 2.9 omitimos la letra m de cada uno de los minitérminos dejando solo los subíndices.

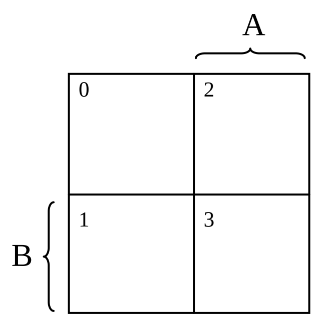


Figura 2.9. Primera forma del mapa de Karnaugh

De tal manera que llegamos finalmente al mapa de Karnaugh que conocemos, donde para cada cuadro se asocia una variable, esto en la figura 2.10, por ejemplo para A se indica que 0 equivale a una A’ y un 1 equivale a una A.

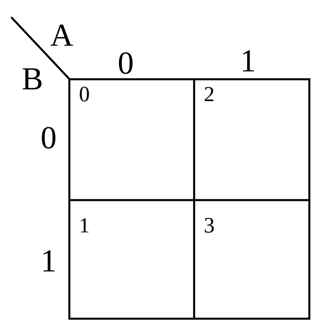


Figura 2.10. Mapa de Karnaugh

Y finalmente observaremos como el mapa de Karnaugh se relaciona directamente con una tabla de verdad de dos variables y que esta relación es de cada uno de los cuadros del mapa a cada uno de los renglones de la tabla, esto en la figura 2.11 y así ser testigos de la evolución de las técnicas de reducción para el diseño de circuitos digitales, el como de utilizar métodos mas dependientes de la habilidad y experiencia de las personas, así como de su criterio en el manejo de estas técnicas (algebra de Boole) , que va tomando forma en un proceso (que pasa por los miniterminos) o técnica viable, estandarizada y sistemática que como bien lo dice su nombre se convierta en un mapa que nos lleve a la mejor solución.

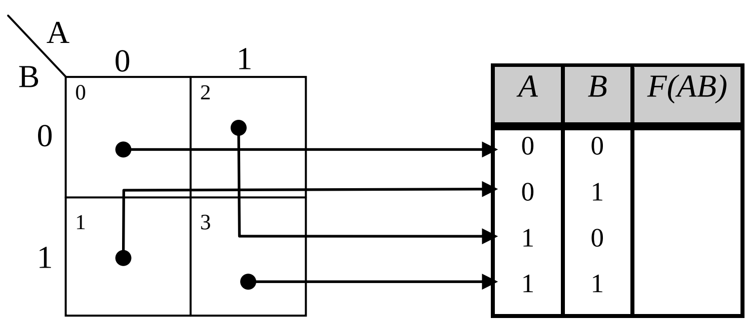


Figura 2.11. Relación mapa de Karnaugh – tabla de verdad.

Los mapas de Karnaugh son utilizados para la resolución de funciones de la conmutación en la que intervienen dos o más variables, por lo que a continuación veremos como de manera similar se origina ahora un mapa de tres variables, en este caso se mostraran solo las figuras de cómo va sucediendo el proceso de formación.

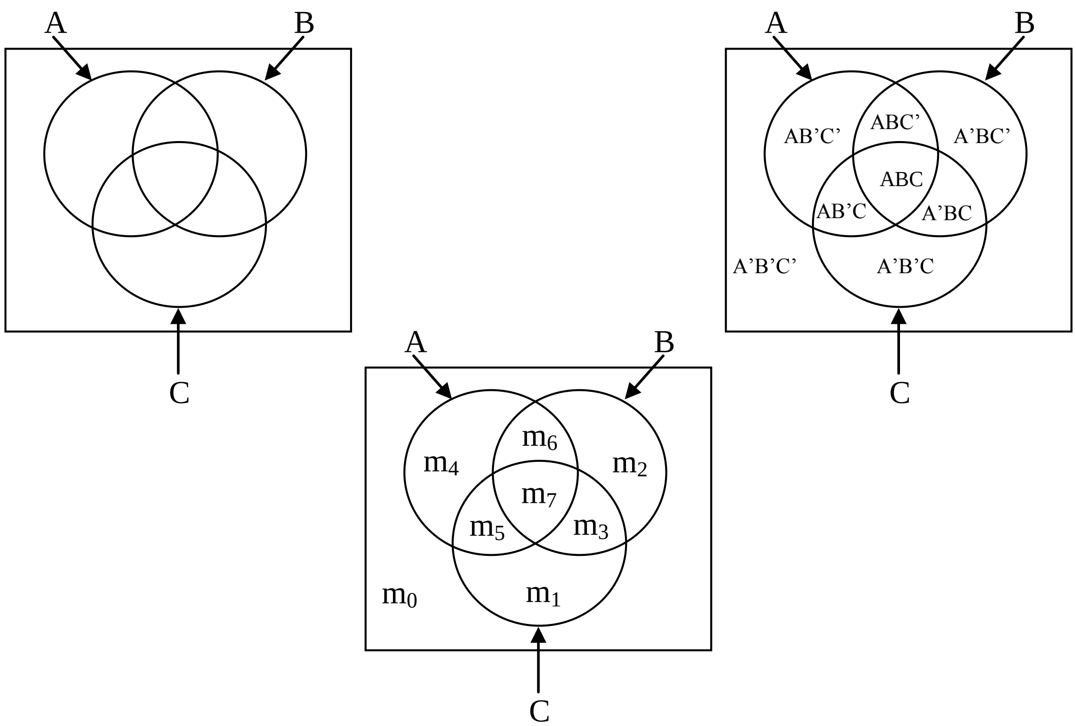


Figura 2.12. Diagramas de Venn para tres variables.

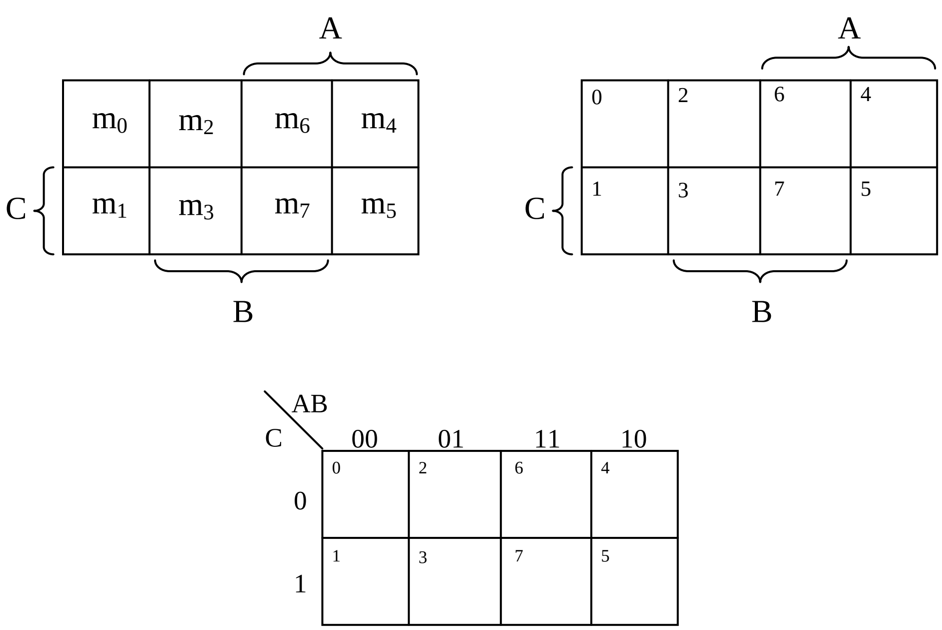


Figura 2.13. Mapas de Karnaugh de tres variables.

Cabe mencionar que de las figuras 2.12 y 2.13 se muestra como el minitérmino m_0 esta en contacto con los minitérminos m_1, m_2 y m_4, siendo que en los mapas el término m_4 no esta adjuntos al m_0, por lo que en este caso se realizara imaginariamente un doblez en el mapa de tal forma que el extremo derecho del mapa pueda tocar el extremo izquierdo, como si el mapa fuera un trozo de papel y quisiéramos hacer un tubo o cilindro con el, así los extremos estarían unidos.

En las siguientes figuras se mostraran los mapas de cuatro, cinco y seis variables

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Figura 2.14. Mapa de Karnaugh de cuatro variables.

DE \ ABC	000	001	011	010	100	101	111	110
00	0	4	12	8	16	20	28	24
01	1	5	13	9	17	21	29	25
11	3	7	15	11	19	23	31	27
10	2	6	14	10	18	22	30	26

Figura 2.15. Mapa de Karnaugh de cinco variables.

AEF \ BCD	000	001	011	010	100	101	111	110
000	0	4	12	8	16	20	28	24
001	1	5	13	9	17	21	29	25
011	3	7	15	11	19	23	31	27
010	2	6	14	10	18	22	30	26
100	32	36	44	40	48	52	60	56
101	33	37	45	41	49	53	61	57
111	35	39	47	43	51	55	63	59
110	34	38	46	42	50	54	62	58

Figura 2.16. Mapa de Karnaugh de seis variable.

2.4.1. Factorización

En general, lo que sucede internamente en los mapas de Karnaugh es un proceso de factorización y esta factorización consiste en que una vez colocada cualquier función sobre el mapa, esta pueda simplificarse a su mínima expresión.

El procedimiento de factorización dependerá obviamente del tipo de función que estemos interesados en minimizar, las variables que en esta intervienen y las dimensiones del mapa, que se verán a detalle en los posteriores temas, tanto en lo que será la "Introducción de Variables", como en "Diagrama de dos variables" hasta el tema de "Diagrama de cinco y seis variables".

Pero hay cinco puntos importantes que podemos mencionar en este instante que nos ayudaran en el momento de llevar acabo la simplificación de funciones sobre los mapas.

1.- Que cada uno de los cuadros del mapa que como sabemos representan un minitérmino sobre este mismo, siendo este de dos variables tendrá dos cuadros adyacentes que también podrán contener minitérminos, ahora esos cuadros respecto al que son adyacentes solo difieren precisamente en una variable; de manera similar sobre un mapa de Karnaugh de tres variables existirán tres cuadros adyacentes, y así sucesivamente, generalizando que cada cuadro de un mapa de n-variables tiene n-cuadros adyacentes con los cuales difiere en una sola variable.

2.- Al combinar los cuadros que contengan un minitérmino en un mapa de Karnaugh para iniciar la simplificación, siempre agruparemos cuadrados en potencias de dos; significa agrupar dos cuadros, cuatro cuadros, ocho cuadros, etc. Sabiendo que al agrupar dos cuadros eliminamos una variable, al agrupar cuatro eliminamos dos variables, etc. Así al agrupar 2^n cuadrados eliminamos n-variables.

3.- Se deberá agrupar tantos cuadros (minitérminos) como sea posible; cuanto mayor sea el grupo, habrá un número menor de literales en el término producto resultante.

4.- Que al formar los grupos estos sean el menor número posible y por supuesto que abarquen a todos los cuadros (minitérminos) de la función. Un minitérmino esta cubierto si esta incluido al menos en un grupo. Si hay menos grupos, será menor el número de términos producto en la función minimizada. Podemos utilizar cada minitérmino cuantas veces sea necesario en los puntos 4 y 5; sin embargo, debemos utilizarlo una vez. Tan pronto hayamos utilizado todos los minitérminos al menos una vez, nos detenemos.

5.- Hay que comenzar siempre al momento de combinar cuadros en el mapa por los cuadros donde existe el menor número de cuadros adyacentes (los cuadros más solitarios en el mapa). Los minitérminos con varios minitérminos adyacentes ofrecen más combinaciones posibles y, por tanto, deben combinarse mas adelante en el proceso de minimización.

2.4.2. Introducción de variables

Procederemos ahora a introducir las variables de las funciones que queramos minimizar dentro del mapa de Karnaugh, comenzando que si la función es de dos variables se utilice un diagrama para dos variables, si es de tres por lógica un diagrama para tres variables, etc. Las funciones que vayamos a introducir las podemos tener en varios aspectos como minitérminos o maxitérminos o también en su forma más sencilla como variables en una suma de productos o producto de sumas de las mismas. Algo también importante a comentar es que las variables que nosotros venimos utilizando son; X, Y, Z, etc. y los mapas que hemos mostrado hasta ahora utilizan variables como A, B, C, etc. Esto por supuesto es indistinto, si los mapas utilizan estas variables es por que son las que se ocuparon para realizar la analogía entre los diagramas de Venn y los mismos mapas. El lector puede utilizar las variables que guste siempre y cuando no modifique su orden, ya que al reordenar las variables también reordenaría la posición de los unos o ceros que representan a los factores de la función según sea el caso de estos dentro del mapa. Veremos a continuación un ejemplo de una función con el cual iremos paso a paso haciendo la introducción de sus variables en un mapa de Karnaugh, además de ir explicando un poco mas a detalle las partes que constituyen el mapa.

Ejemplo 2.8: Introducir las variables de la función $F = X'Y + XY' + XY$ a un mapa de Karnaugh

De lo primero que tendremos que hacer es observar que en nuestra función solo intervienen dos variables por lo que utilizaremos un diagrama o mapa para dos variables (figura 2.17). Del mapa podemos ver que este para cada variable puede tener el valor de 1 o 0, obviamente si la variable es negada nos representara un 0 y sin negar un 1 por lo que solo buscaremos los cuadros que en su intersección nos representen el elemento de la función que queremos.

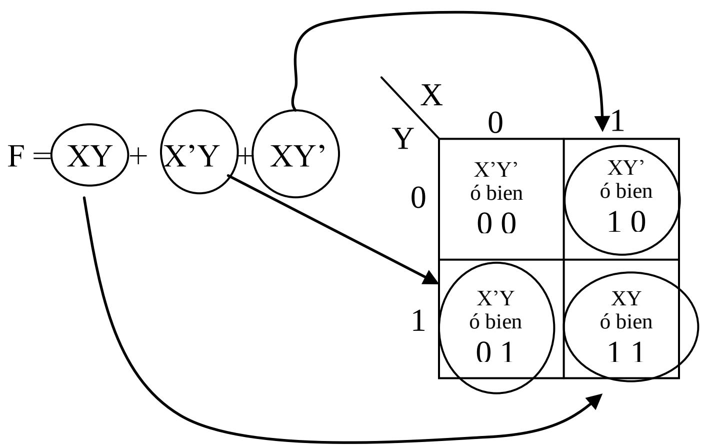


Figura 2.17. Relación de una función con el mapa.

Una vez hecho esto procederemos a colocar en los cuadros que contienen a nuestros elementos de la función en su lugar un 1 y en el resto no colocaremos por el momento nada, quedándonos como en la figura 2.18.

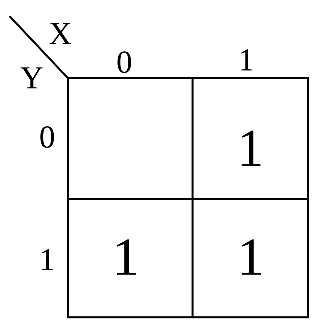
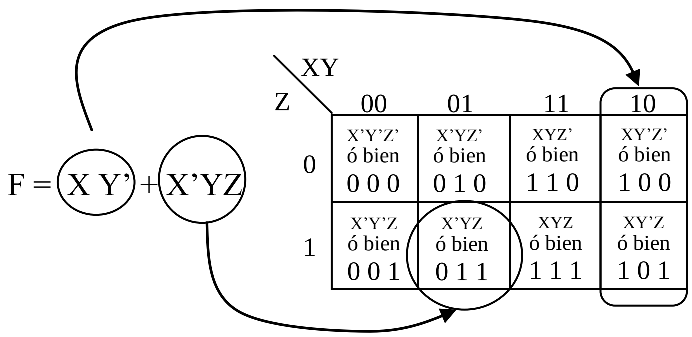


Figura 2.18. Mapa que contiene los elementos de F = X'Y+XY'+XY

Realizaremos a continuación un ejemplo más de la introducción de variables ahora con una función que contendrá tres variables, para que posteriormente veamos como se realiza la introducción de variables cuando nuestra función ya se ha convertido en un minitérmino o maxitérmino.

Ejemplo 2.9: Introducir las variables de la función F = X Y' + X'YZ en un mapa de Karnaugh

Como lo habíamos comentado esta función cuenta con tres variables por lo que necesitaremos un diagrama para este número de variables. Ya en este diagrama igual que en el ejemplo anterior buscaremos los cuadros que en su intersección coincidan con los elementos de nuestra función.



Quedando

Z \ XY	00	01	11	10
0				1
1		1		1

Figura 2.19. Mapa que contiene los elementos de F = X Y' + X'YZ

Algo importante que comentar es que en este ejemplo la función cuenta con solo dos elementos mientras que en la figura 2.19 observamos que se plasmaron tres unos, esto debido a que el elemento XY’ coincide con dos cuadros, los cuales debemos tomar en cuenta aunque no exista la tercera variable, debido a que esta ultima podría tomar los valores de 1 o 0.

También podremos introducir las variables de las funciones cuando estas ya estén representadas en forma de minitérminos o maxitérminos, pues cada uno de estos corresponde a un cuadro dentro del mapa haciendo aun más sencilla la introducción de variables.

Ejemplo 2.10: Introducir la función F(X,Y,Z) = m(0,3,5) = $m_0 + m_3 + m_5$

La función es de tres variables así que usaremos un diagrama para tres variables como el mostrado con anterioridad en la figura 2.13 en donde ya los cuadros tienen el minitérmino que les corresponde, plasmado en la parte superior izquierda de cada uno de los mismos, esto de acuerdo a la deducción que partió de los diagramas de Venn. Nos queda como se muestra en la figura 2.20

$F = m_0 + m_3 + m_5$

Z \ XY	00	01	11	10
0	^{0} 1	^{2}	^{6}	^{4}
1	^{1}	^{3} 1	^{7}	^{5} 1

Figura 2.20. Mapa con los minitérminos de $F = m_0 + m_3 + m_5$.

Si en dado momento quisiéramos introducir los maxitérminos a partir de la función $F = m_0 + m_3 + m_5$ sería sencillo ya que la deducción de ellos a partir de los minitérminos es muy practica, solo tomando en cuenta los términos faltantes y los representaríamos en el mapa de Karnaugh con ceros en lugar de unos. Observemos la figura 2.21 donde se muestra esto.

$F = \prod M(1,2,4,6,7) = M_1\ M_2\ M_4\ M_6\ M_7$

Z \ XY	00	01	11	10
0	^{0}	^{2} 0	^{6} 0	^{4} 0
1	^{1} 0	^{3}	^{7} 0	^{5}

Figura 2.21. Mapa con los maxitérminos $\prod M(1,2,4,6,7)$

O también se podrían plasmar tanto los minitérminos como los maxitérminos de una misma función dentro del mismo mapa como en la figura 2.22.

$F = m(0,3,5) = m_0 + m_3 + m_5$ y
$F = \prod M(1,2,4,6,7) = M_1\ M_2\ M_4\ M_6\ M_7$

Z \ XY	00	01	11	10
0	^{0} 1	^{2} 0	^{6} 0	^{4} 0
1	^{1} 0	^{3} 1	^{7} 0	^{5} 1

Figura 2.22. Mapa con minitérminos y maxitérminos.

En el caso anterior de la figura 2.22 se muestran dentro del mapa tanto los minitérminos como los maxitérminos pero habrá que aclarar que lo idóneo es plasmar en el mapa de Karnaugh ya sea solo los minitérminos o solo los maxitérminos, para que posteriormente se realice su resolución.

En otro caso que debemos observar es que por ejemplo en los casos anteriores hemos encontrado los maxitérminos directamente del mapa por la deducción lógica de que estos se encuentran dentro de los cuadros que no contienen un 1, pero también existe la forma de que estos se puedan encontrar o introducir cuando la función se encuentra en una forma de producto de sumas en lugar de una suma de productos, como vemos en el ejemplo 2.11

Ejemplo 2.11. Introducir las variables de F = (W + Y) (X +Y) (X’ + Y’ + Z) a un mapa de Karnaugh.

Lo primero a observar es que necesitaremos un diagrama de cuatro variables y que como ya habíamos comentado esta función se encuentra en producto de sumas, por lo que aquí introduciremos a los factores no como unos si no como ceros en los cuadros que correspondan, además de que las variables negadas de cada factor para poder localizarlos dentro del mapa toman el valor de uno y las no negadas el valor de cero, así es, lo contrario en cierta forma a encontrar minitérminos, ya que lo que estamos introduciendo son maxitérminos.

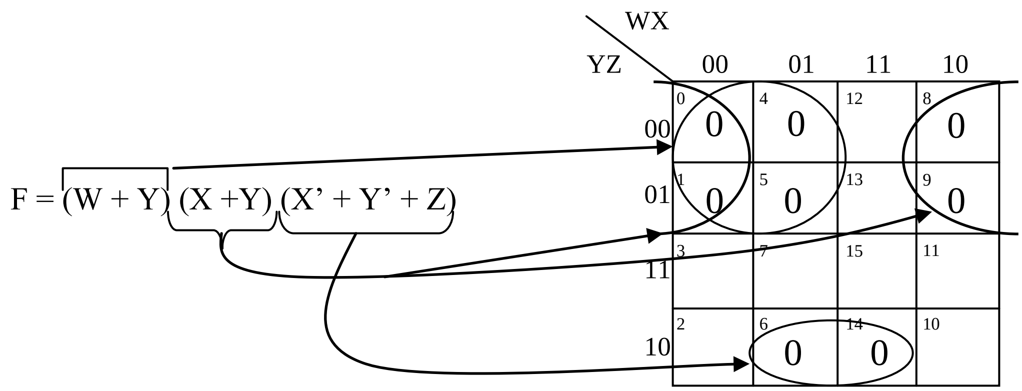


Figura 2.23. Mapa de una función en la forma producto de sumas.

Del mapa de la figura 2.23 directamente observamos y podemos obtener los maxitérminos que intervienen en esta función y son $F = \prod M(0,1,4,5,6,8,9,14) = M_0 M_1 M_4 M_5 M_6 M_8 M_9 M_{14}$ y si deseamos obtener los minitérminos serían; $F = m(2,3,7,10,11,12,13,15) = m_2 + m_3 + m_7 + m_{10} + m_{11} + m_{12} + m_{13} + m_{15}$. Quedando como ejercicio al lector el introducir los minitérminos en un mapa para que pueda observar como quedan colocados si así lo desea.

La introducción de variables de una función que contiene cinco de estas variables o mas dentro de un mapa de Karnaugh, se realiza de manera similar a como ya lo hemos visto recientemente, por lo que procederemos a ver su resolución en los siguientes temas que iniciaremos a continuación.

2.4.3. Diagrama con dos variables

En el diagrama con dos variables o mapa de Karnaugh con dos variables como se le prefiera llamar pero que con conciencia sepamos que hablamos de ellos como uno mismo, como iguales, veremos mas que cualquier otra cosa como este nos ayuda a simplificar una función compleja a otra mas simple creando así una técnica de reducción.

Comenzaremos por observar la figura 2.24 en donde se encuentran el diagrama con dos variables X y Y. El lector deberá elegir con que tipo de variables desea trabajar, aunque esto en realidad no modifica en nada el procedimiento, recordando nuevamente que siempre y cuando las variables no sufran un reordenamiento, por que esto también provocaría que la ubicación de tanto los unos como los ceros que representan los factores de la función quedan en un distinto lugar. Para nuestro caso el manejo lo continuaremos con las variables X y Y ya que estas han sido las variables que hemos manejado desde un inicio.

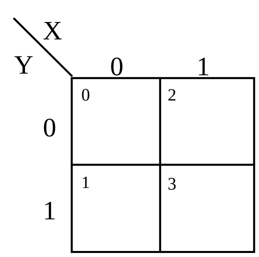


Figura 2.24. Diagrama con dos variables

Iniciaremos la utilización del diagrama para la simplificación de una función, tomando en cuenta por supuesto que las variables de esta ya estén plasmadas dentro del diagrama o mapa, además de retomar las indicaciones para realizar la factorización, mencionadas en el punto 2.4.1. Tenemos de esta forma a continuación el ejemplo 2.12 donde se encuentra la función y el diagrama que ya la contiene.

Ejemplo 2.12. Simplifique a su mínima expresión la función F = X’Y + XY’ + X’Y’ utilizando el mapa de Karnaugh de dos variables.
Introduciendo la función queda:

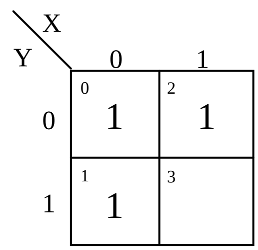


Figura 2.25. Diagrama de la Función F = X’Y + XY’ + X’Y’

Como ya habíamos comentado considerando los puntos mencionados en 2.4.1. agrupamos en pares (2, 4, 8, etc.), por lo que nuestro mapa de la figura 2.25 se vera como en la figura 2.26 que a continuación se muestra.

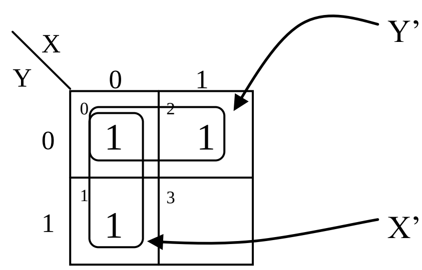


Figura 2.26. Diagrama con unos agrupados.

Del diagrama de la figura 2.26 y de la agrupación de la fila superior obtenemos solo la variable Y’ ya que al momento de agrupar dos cuadros eliminamos una variable que para el caso es X ya que esta cambia de un minitérmino a otro tomando valores de 0 y 1 mientras que Y no lo hace, se mantiene negada. En similitud a lo anterior de la agrupación de la primera columna que es de dos cuadros también podremos eliminar una variable que será la Y, la cual cambia de un minitérmino a otro tomando valores de 0 y 1, mientras que X se mantiene negada quedando asi la variable X’

Cabe comentar al lector que el hecho de eliminar la variable cuando cambia de 0 a 1 en la agrupación de cuadros viene justificada por el Postulado 5 donde X + X’ = 1 ya que si tenemos dos términos como lo son X’Y’ + XY’ esto seria igual a Y’(X + X’) quedando solo Y’ y la función seria; F = X’ + Y’. Con este principio los mapas de Karnaugh hacen la eliminación de variables para dos, tres o mas variables.

2.4.4. Diagrama con tres variables

Por supuesto que la intención del diagrama de tres variables que a continuación utilizaremos de la misma forma que el diagrama de dos que es el simplificar una función compleja a una mas simple.

Tomaremos en cuenta nuevamente las indicaciones de la factorización del punto 2.4.1 y ejemplificaremos con la simplificación de una función de tres variables ya inscritas en el diagrama.

Ejemplo 2.13. Simplifique a su mínima expresión la función F = X´Y´Z + X´YZ + XYZ´ utilizando el mapa de Karnaugh

Z \ XY	00	01	11	10
0	^{0}	^{2}	^{6} 1	^{4}
1	^{1} 1	^{3} 1	^{7}	^{5}

Figura 2.27. Mapa que contiene la función F = X´Y´Z + X´YZ + XYZ´.

Agruparemos en potencias de dos (2, 4, 8, etc.) los unos del diagrama y procederemos inmediatamente a simplificar tomando en cuenta que entre mas cuadros o unos agrupados eliminamos mas variables.

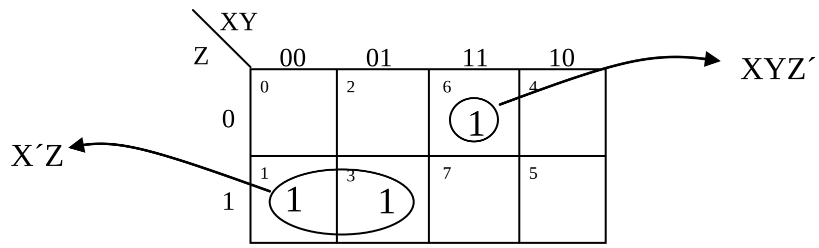


Figura 2.28. Mapa que contiene la función F = X´Y´Z + X´YZ + XYZ´ con las agrupaciones en potencias de dos.

Del diagrama y de la simplificación tenemos que solo se pudieron agrupar un par de unos de la fila inferior del diagrama en donde por tanto se elimina una variable que es Y, ya que esta cambia de un término a otro, o de un cero a un uno mientras que la X´ se mantiene al igual al igual que Z. Por otro lado en la parte superior del diagrama queda sin agrupar el uno que representa al termino XYZ´ y que por tanto no se puede reducir quedando de la misma forma.

Así de la reducción de la función en el mapa queda:

F = X´Y´Z + X´YZ + XYZ´ = XYZ´ + X´Z

Pero veamos mas ejemplos de simplificación con este diagrama, para que logremos observar las diferentes maneras de agrupar y reducir variables.

Ejemplo 2.14. Simplifique a su mínima expresión a F = X´Y´Z´ + X´Y´Z + X´Y Z´+ XY´ utilizando el mapa de Karnaugh.

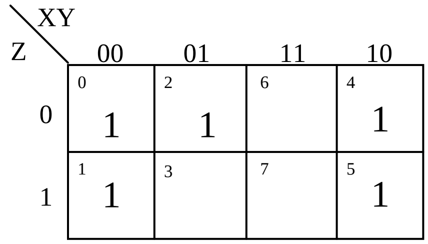


Figura 2.29. Mapa que contiene la función F = X´Y´Z´ + X´Y´Z + X´Y Z´+ XY´

Una vez mas agrupamos cuadros o unos en potencias de dos para iniciar la simplificación como se muestra en al figura 2.30

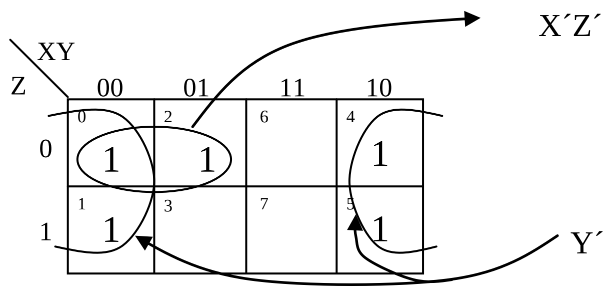


Figura 2.30. Mapa que contiene la función F = X´Y´Z´ + X´Y´Z + X´Y Z´+ XY´ agrupada en potencias de dos.

De la figura 2.30 observamos como existe una agrupación de cuatro cuadros, en la que podemos confirmar como se han eliminado ya dos variables quedando solamente una que es Y´. Esta agrupación se da uniendo los cuadros extremos como si el mapa fuera una hoja de papel y esta la trataremos de unir para formar un cilindro, así de esta manera los cuadros con minitérminos 4 y 5 y los minitérminos 1 y 0 se acercan y nos dan un conjunto de cuatro cuadros en donde se eliminan dos variables. Queda además en el mapa un solo uno el cual por la cercanía con otro se agrupa con este para poder simplificarse, no importando que este segundo ya había sido utilizado en la agrupación de cuatro unos.

Al final de la simplificación utilizando el mapa de Karnaugh tenemos la siguiente función:

F = X´Y´Z´ + X´Y´Z + X´Y Z´+ XY = X´Z´+ Y´

Ejemplo 2.15. Simplifique a su mínima expresión F = X´Y´Z´+ X´YZ + X´YZ + XYZ´+ XYZ

Se ingresa la función en el diagrama, agrupamos inmediatamente en potencias de dos como se muestra en la figura 2.31.

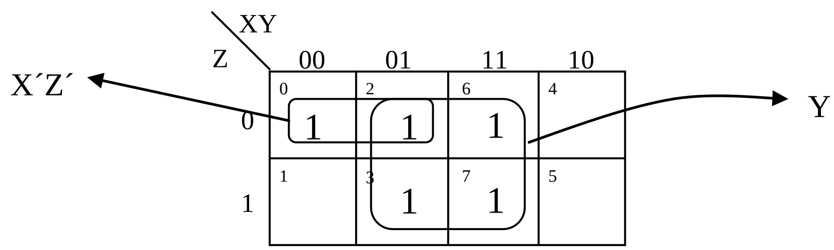


Figura 2.31. Mapa que contiene la función F = X´Y´Z´+ X´YZ + X´YZ + XYZ´+ XYZ agrupada en potencias de dos.

De la agrupación vemos como nos quedan los términos Y y X´Z´. Del conjuntar a loa cuatro unos se obtuvo una sola variable (Y) eliminándose 2 y al realizar lo mismo con los dos unos se elimina una sola variable y así queda X´Z´.

Por lo tanto F = X´Y´Z´+ X´YZ + X´YZ + XYZ´+ XYZ = X´Z´ + Y

De tal manera que con los ejercicios vistos con anterioridad analizamos que la simplificación hasta este momento es rápida, cómoda y eficiente.

2.4.5. Diagrama con cuatro variables

No existe impedimento alguno para que este diagrama de cuatro variables sea manejado con tal sencillez como lo es el de tres variables como se verá a continuación.

Ejemplo 2.16. Simplifique a su mínima expresión la función F = W´X´Y´Z´ + WX´Y´Z´ + W´X´YZ´ + WX´YZ´ + W´XYZ + WXYZ + WXY´Z + W´XY´Z.

Como primer paso obvio es realizar la introducción de la función al diagrama para inmediatamente agrupar los unos que representan a cada uno de los términos en potencias de pares. Así tenemos la figura 2.32 donde se observa la ubicación de estos.

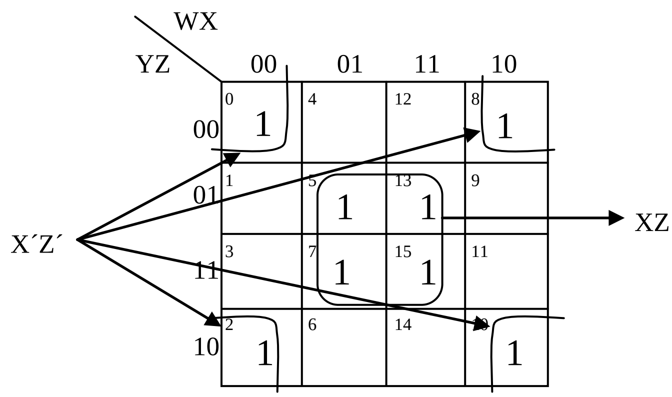


Figura 2.32. Mapa que contiene la función F = W´X´Y´Z´ + WX´Y´Z´ + W´X´YZ´ + WX´YZ´ + W´XYZ + WXYZ + WXY´Z + W´XY´Z.

De una de las agrupaciones vemos como las cuatro esquinas se unen, recordando que ya sabemos que las justificación viene de los diagramas de Venn y que físicamente en estos no se observa la vecindad de los minitérminos de las orillas entre ellos mismos así al unir en un solo conjunto los unos de las esquinas del mapa se obtiene una X´Z´ y del agrupamiento central de unos se obtiene XZ.

De esta manera la función simplificada de F = W´X´Y´Z´ + WX´Y´Z´ + W´X´YZ´ + WX´YZ´ + W´XYZ + WXYZ + WXY´Z + W´XY´Z = X´Z´ + XZ

Ejemplo 2.17. Llevar a su mínima expresión F = W´X´Y´Z´ + W´XY´Z´ + W´XY´Z + W´XYZ + WXYZ + WXYZ´ + WX´YZ´ + WX´Y´Z´.

Nuevamente introducimos la función que deseamos simplificar en el diagrama de cuatro variables y tenemos la figura 2.33 en la cual se observa la agrupación de los elementos de la función.

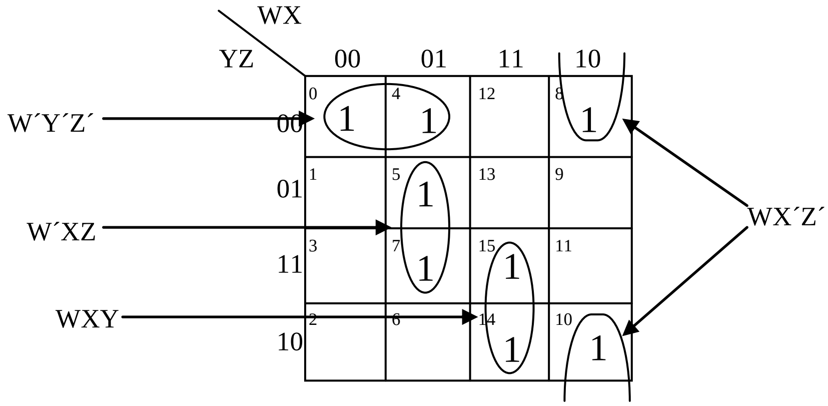


Figura 2.33. Mapa que contiene F = W´X´Y´Z´ + W´XY´Z´ + W´XY´Z + W´XYZ + WXYZ + WXYZ´ + WX´YZ´ + WX´Y´Z´.

Quedando en la agrupación superior izquierda del mapa en la figura 2.33 el resultado W´Y´Z´, luego del conjunto de los minitérminos m_5 y m_7 se obtiene W´XZ. Continuando la reducción se observa que del m_{14} y m_{15} se obtiene WXY y por último la agrupación del m_{10} y m_8 da como resultado WX´Z´.

Por lo tanto el resultado de la simplificación es F = W´Y´Z´ + W´XZ + WXY + WX´Z´

Cabe señalar que también de este mismo caso existe al menos una forma distinta mas de agrupar los elementos de la función como se muestra en la figura 2.34 en donde el resultado es también distinto al encontrado pero que al compararlos observamos que para ambos existe el mismo numero de términos y literales, por tanto con un mismo costo al momento de llevar al armado a estos circuitos.

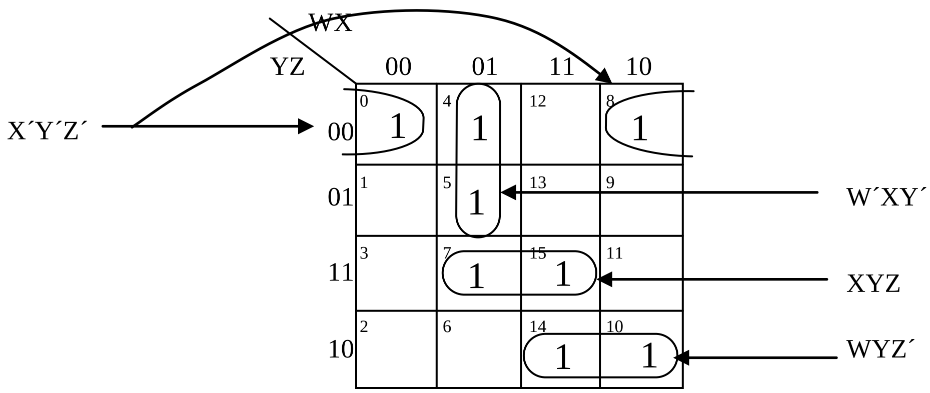


Figura 2.34. F = X´Y´Z´ + W´XY´ + XYZ + WYZ´

En ejemplo 2.18 también se lleva a cabo como podemos utilizar minitérminos que ya hayan sido utilizados en otras agrupaciones, de la misma forma en que en los ejemplos 2.14 y 2.15 de los diagramas de tres variables.

Ejemplo 2.18. Simplificar F(WXYZ) = ∑m(3, 4, 5, 6, 7, 8, 13, 15)

En el planteamiento de este ejemplo no se colocaron las literales de cada término de la función si no que solo se realizo la representación de los minitérminos de la misma. La resolución lleva el mismo proceso; introducimos la función al mapa y agrupamos la mayor cantidad de unos como se observa en la figura 2.35.

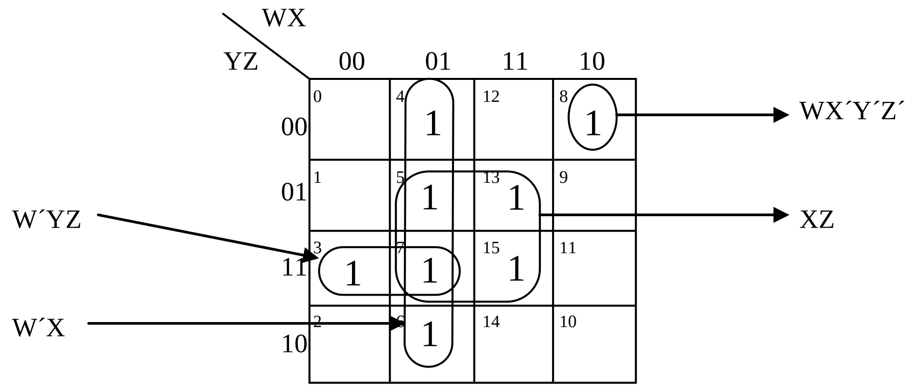


Figura 2.35. F(WXYZ) = ∑m(3, 4, 5, 6, 7, 8, 13, 15)

De una primera agrupación de minitérminos m_5, m_7, m_{13} y m_{15} se obtiene una simplificación de dos variables arrojando XZ, posteriormente se agrupa m_4, m_5, m_6 y m_7 para dar W´X, y aquí es donde observamos que se usa m_5 y m_7 minitérminos ya usados en nuestra primera simplificación y sucede nuevamente lo mismo al agrupar m_3 y m_7, pero ahora con m_7 que es la tercera vez que se ocupa al momento de agrupar para dar W´YZ. Ya por ultimo m_8 no se puede agrupar con nadie y nos da como resultado el mismo m_8 o sea WX´Y´Z´.

Así F(WXYZ) = ∑m(3, 4, 5, 6, 7, 8, 13, 15) = XZ + W´X + W´YZ + WX´Y´Z´

Finalmente la reducción queda F = XZ + W´X + W´YZ + WX´Y´Z´

2.4.6. Diagrama con cinco y seis variables

Este tipo de diagramas se vuelven mas difíciles de usar ya que el numero de cuadros aumenta, si sabemos que la cantidad de cuadros de un diagrama va a depender del numero de variables y sabiendo la relación que existe de estos con las tablas de verdad, por lo que esta cantidad de cuadros se define por 2^n don n = numero de variables, para un diagrama de 5 variables se necesitan 32 cuadros ($2^5 = 32$) y un diagrama de 6 variables 64 cuadros ($2^6 = 64$). De tal forma que su manejo se vuelve difícil. Para la simplificación de funciones de 7 o mas variables los mapas se vuelven complicados de ocupar.

A continuación observamos un diagrama de 5 variables.

DE \ ABC	000	001	011	010	100	101	111	110
00	0	4	12	8	16	20	28	24
01	1	5	13	9	17	21	29	25
11	3	7	15	11	19	23	31	27
10	2	6	14	10	18	22	30	26

Figura 2.36. Diagrama de cinco variables.

En la figura 2.36 las variables han cambiado de utilizar las ultimas letras del abecedario (W, X, Y, Z) a las primeras (A, B, C, D, … , etc.) sabiendo de antemano que no existe razón importante alguna. Como ya se había comentado con anterioridad, en esta tesis se hace de esta forma con la intención de no ocupar la letra “V” que es la anterior a “W” y que seria la literal siguiente a utilizar en esta situación de un diagrama de cinco variables y que es mas usada en este ámbito de la electrónica para representar el voltaje. Si el lector gusta podría usar esta literal, pero con la conciencia exacta de para que se esta utilizando. aunque hay otras opciones que como ya dijimos podrían ser las primeras letras del abecedario o hasta una sola literal, pero con subíndices como podrían ser X_1, X_2, X_3, X_4, X_5 para representar a cada una de las variables de un diagrama de cinco variables, X_1, X_2, X_3, X_4, X_5, X_6 para representar las variables de un diagrama de seis variables, y así sucesivamente, solo basta con no perder el orden de cómo son acomodadas todas estas en el mapa ya elaborado, ya que un mal acomodo podría causar algún tipo de confusión o problema al momento de la simplificación, problema que no existirá si se hace la analogía con lo presentado aquí.

Iniciamos así con la simplificación de una función de cinco variables como ejemplo de este punto.

Ejemplo 2.19. Minimizar en el mapa de Karnaugh la función de cinco variables F(A, B, C, D, E) = ∑m(0, 1, 4, 5, 6, 13, 14, 15, 22, 24, 25, 28, 29, 30, 31)

El primer paso es introducir la función en el mapa ayudándonos de que nuestro mapa ya tiene ubicado cada uno de los términos mínimos en cada cuadro.

DE \ ABC	000	001	011	010	100	101	111	110
00	^{0} 1	^{4} 1	^{12}	^{8}	^{16}	^{20}	^{28} 1	^{24} 1
01	^{1} 1	^{5} 1	^{13} 1	^{9}	^{17}	^{21}	^{29} 1	^{25} 1
11	^{3}	^{7}	^{15} 1	^{11}	^{19}	^{23}	^{31} 1	^{27}
10	^{2}	^{6} 1	^{14} 1	^{10}	^{18}	^{22} 1	^{30} 1	^{26}

Figura 2.37. Mapa de la función F(A, B, C, D, E) = ∑m(0, 1, 4, 5, 6, 13, 14, 15, 22, 24, 25, 28, 29, 30, 31).

Al agrupar debemos considerar que los dos cuadros grandes se relacionan uno con otro de tal manera que los cuadros interiores de cada uno de ellos están en una misma posición como si estos se colocaran alineados o uno frente al otro y que los términos o agrupaciones ubicados en uno de los cuadros grandes y coincidentes con otros términos o agrupaciones en el otro cuadro grande provocaran el poder realizar una nueva agrupación como si fuera una sola, de manera semejante a la de la figura 2.38.

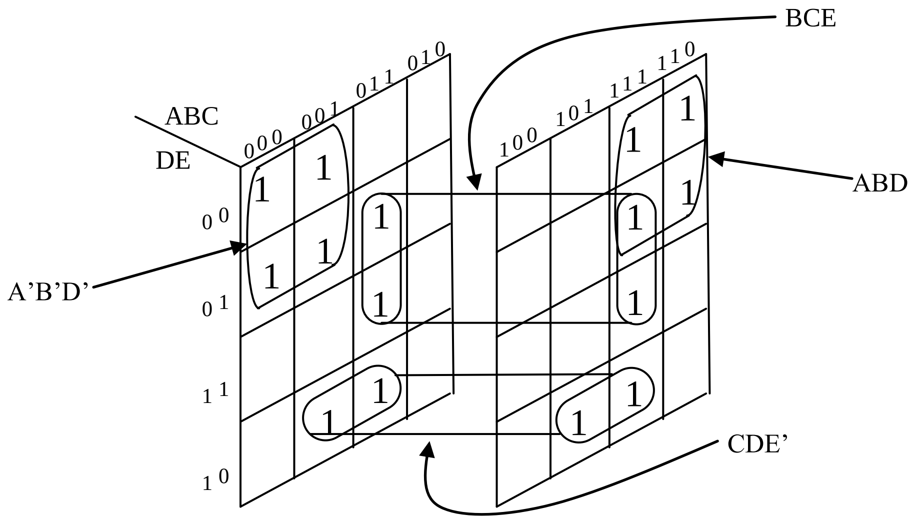


Figura 2.38. Mapa de la función F(A, B, C, D, E) = ∑m(0, 1, 4, 5, 6, 13, 14, 15, 22, 24, 25, 28, 29, 30, 31) que muestra como se relacionan los cuadros interiores de los dos principales cuadros, para realizar las agrupaciones.

La figura 2.38 nos muestra como es que se relacionan los recuadros, aunque la visualización y manejo del mapa para la simplificación es como se muestra en la figura 2.39 que a continuación se muestra y la relación de los recuadros se da por medio de flechas o líneas.

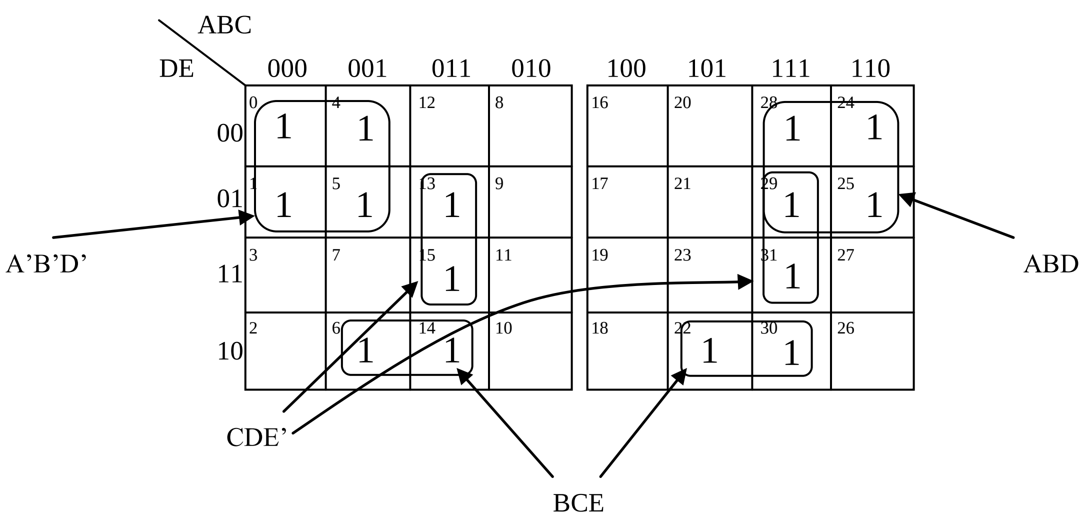


Figura 2.39. Mapa de la función F(A, B, C, D, E) = ∑m(0, 1, 4, 5, 6, 13, 14, 15, 22, 24, 25, 28, 29, 30, 31) que muestra sus agrupaciones y simplificaciones resultantes de cada agrupación.

En cualquiera de los casos mostrados en la figura 2.38 y 2.39 al iniciar la simplificación obtenemos que al conjuntar los grupos de minitérminos m_6 y m_{14} con el grupo que forma m_{22} y m_{30} para formar un nuevo grupo de cuatro minitérminos, por lo cual se eliminaron dos variables. De un cuadro mayor a el otro cuadro mayor de este mapa cambia la variable A por lo que se elimina automáticamente y la variable B en ambos subgrupos cambia por lo que solo quedan CDE' que son las que no cambian. Por otra parte y de manera semejante el grupo de minitérminos m_{13} y m_{15} se unen para formar grupo mayor de cuatro con m_{29} y m_{31} y así eliminar dos variables también quedando BCE. De la agrupación de cuatro unos del primer cuadro mayor no se puede alinear o agrupar con algún semejante en el otro cuadro mayor del mapa así que se realiza la simplificación con esa agrupación que son m_0, m_1, m_4, y m_5 quedando de estos A'B'D' y los mismo con la agrupación de cuatro unos en el segundo cuadro mayor donde m_{24}, m_{25}, m_{28} y m_{29} dan como resultado ABD'.

Por lo tanto la simplificación de la función F(A, B, C, D, E) = ∑m(0, 1, 4, 5, 6, 13, 14, 15, 22, 24, 25, 28, 29, 30, 31) es F = A'B'D' + ABD' + BCE + CDE'

Ejemplo 2.20. Minimizar con el diagrama una función de seis variables F(A,B,C,D,E,F) = ∑m(0,1,6,7,9,13,14,15,16,17,32,33,38,39,46,47,48,49,57,61)

La solución utilizando un mapa de seis variables es de manera muy similar a la de cinco en donde los cuadrantes o cuadros mayores se alinearon para observar la adyacencia entre cuadros.

En el ejemplo 2.19 observamos la figura 2.37 en donde la diferencia de los dos cuadrantes es A, ya que en un cuadrante existe A' y en el otro A, siendo esta la única diferencia por la que los cuadrantes al alinearse hacen que los cuadros sean adyacentes ya que su diferencia es solo A y A'.

De manera análoga, dividimos el mapa de seis variables en cuatro cuadrantes figura 2.40, cada uno de los cuales representa una combinación de las variables A y B en cada uno de los cuadrantes y que se encuentran señaladas, ya que el resto de las variables C, D, E, y F son las mismas en cada uno de ellos. En la figura 2.41, en donde imaginando a los cuadrantes acomodados uno frente del otro, de manera que las celdas mantendrán una adyacencia con otra celda del otro cuadrante de manera horizontal o como lo marcan las flechas que van de un cuadrante a otro, siendo el último adyacente con el primero, como se muestra.

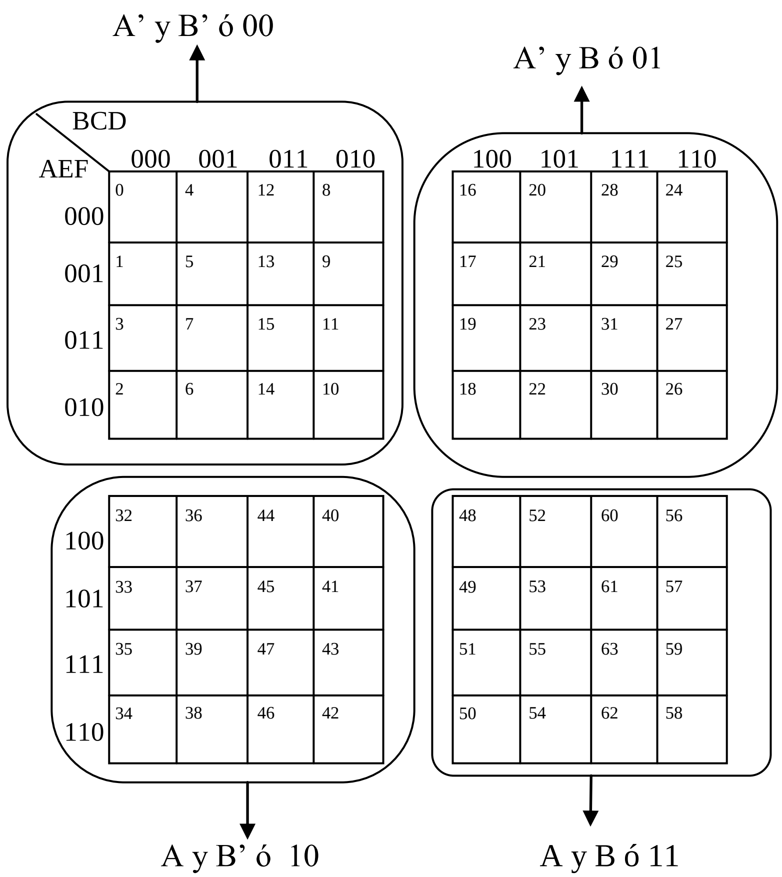


Figura 2.40. Mapa de seis variables que muestra las variables que cambian de cuadrante a cuadrante.

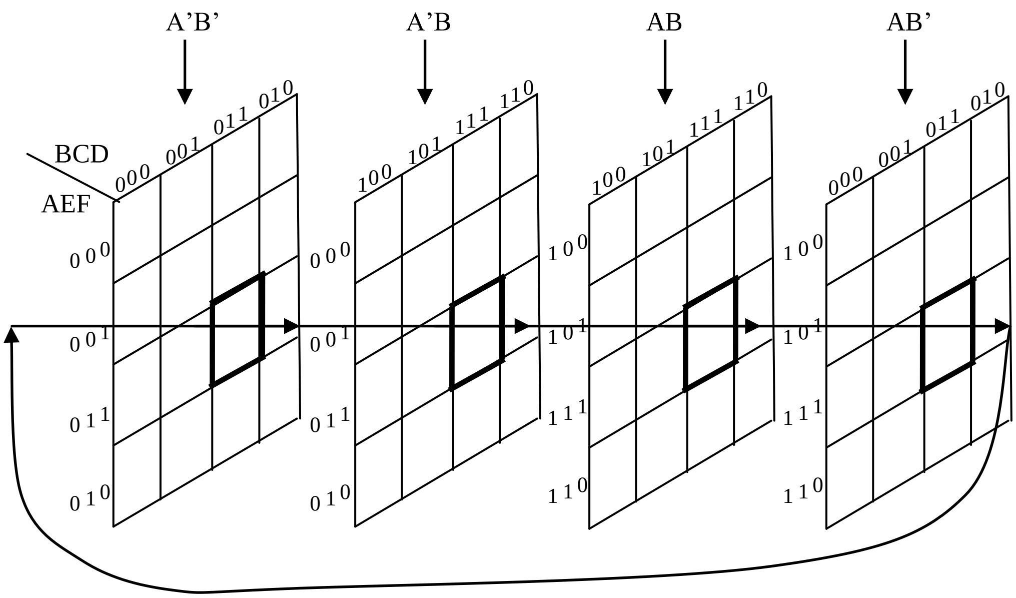


Figura 2.41. Mapa que muestra las adyacencias entre recuadros de los distintos cuadrantes.

Continuando introducimos la función de seis variables en el mapa de Karnaugh de seis variables y agrupamos como se muestra en la figura 2.42, y para iniciar la simplificación de la función nos imaginamos las adyacencias como las que se muestran en la figura 2.41.

Se agruparon los unos como ya se ha hecho en otras ocasiones y observamos en la figura 2.42 la relación en vecindad que existe entre la agrupación de dos unos que se encuentra en la parte superior izquierda del mapa, que son m_0 y m_1 con los m_{16} y m_{17} que se encuentran en la parte superior derecha del mismo, esta adyacencia esta justificada claramente en que de un grupo de unos a otro grupo de unos a pesar de estar en distintos recuadros, solo cambian en una variable, por lo que se procede a tomarlos como un solo grupo de cuatro unos, pero ahí no termina ese agrupamiento, si observamos en la parte inferior del mapa observamos que se encuentran otros dos grupos de dos unos, colocados en posiciones semejantes a los agrupados en la parte superior, estos son m_{32} y m_{33}, y m_{48} y m_{49}, que también guardan vecindad con los agrupados en la parte superior del mapa ya que tienen en diferencia una solo variable. De esta manera nos queda finalmente un grupo de ocho unos que es mucho mayor y que ayuda a eliminar tres variables y terminan quedando C'D'E'.

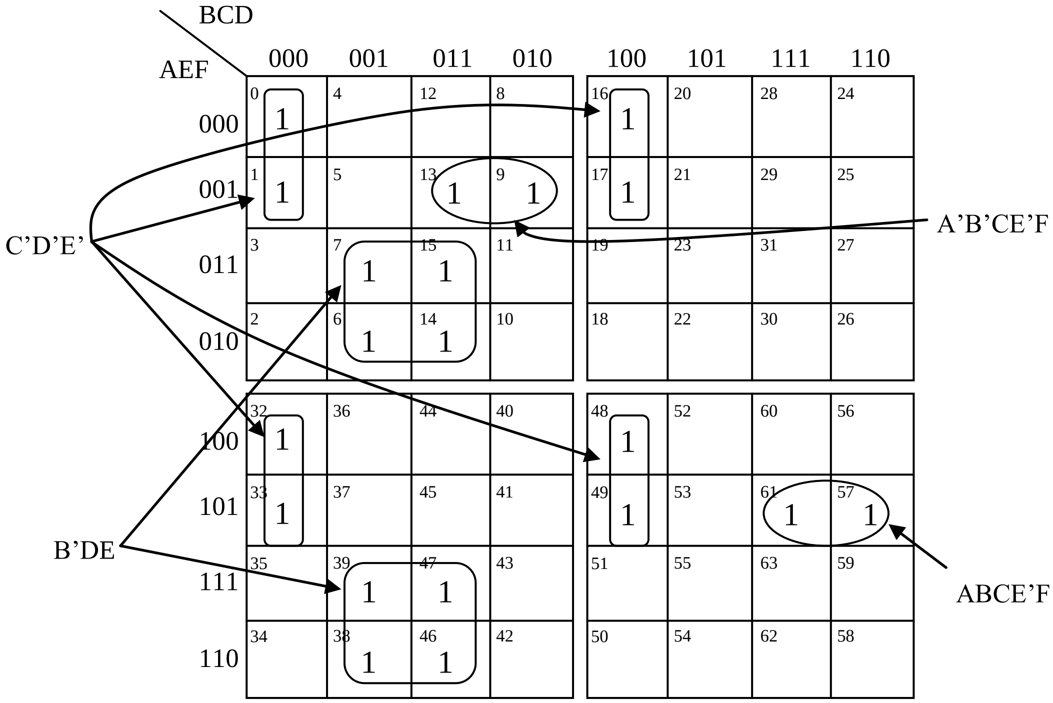


Figura 2.42. Muestra la forma de agrupar los minitérminos de la función F(A,B,C,D,E,F)=∑m(0,1,6,7,9,13,14,15,16,17,32,33,38,39,46,47,48,49,57,61)

Lo siguiente a simplificar son el grupo de m_6, m_7, m_{14} y m_{15} que observamos se encuentran en la parte superior izquierda del mapa de Karnaugh (figura 2.42) y que agrupamos a su vez con otro grupo de cuatro unos que se encuentran en una posición similar, pero estos en la parte inferior izquierda del diagrama. Estos se pueden unir por lo que hemos comentado, que es la existencia de una sola variable de diferencia entre estos grupos. Forman ocho unos por lo que también se eliminan tres variables y nos queda al final B'DE.

Por ultimo el grupo que forman m_{13} y m_9 no pueden ser agrupados a su vez con ningún otro grupo, así que de ellos emanan las variables A'B'CE'F, le sucede los mismo al grupo m_{61} y m_{57} quedando de estos ABCE'F.

Por lo tanto la función simplificada queda F = B'DE + C'D'E' + A'B'CE'F + ABCE'F

2.5. Método de tabulación

Este método, que independientemente a la sucesión de temas es mas que nada el limite entre la buena simplificación de funciones que permiten los mapas de Karnaugh y la dificultad que estos mismos representan del manejo de las variables al simplificarlas cuando estas son seis o mas, ya que este buen manejo esta supeditado a la experiencia que tenga la persona que use esta técnica, a su observación y habilidad de saber elegir correctamente las agrupaciones, para poder realizar la reducción optima.

Así entonces esta técnica de reducción por el método de tabulación viene a simplificar este proceso, pero mas que nada por que logra hacer de mejor manejo las funciones de seis o más variables. Este método brinda un procedimiento mas especifico y menos arbitrario, que nos asegura la obtención de una expresión mejor simplificada, con una base de pasos mas concreta, que por supuesto también tiene que tomarse cuidado en su uso, por que este puede causar errores por el uso tan rutinario que puede llegar a provocar.

Este método se divide principalmente en un proceso que consta de dos partes que son: la determinación de los términos y la selección de los términos, que serán los subtemas 2.5.1 y 2.5.2 respectivamente.

2.5.1. Determinación de los términos

El punto de inicio para este método, es en principio el tener a la mano nuestra función representada en minitérminos que de no ser así, tendríamos que convertirla de términos booleanos a minitérminos.

Una vez realizado lo anterior diremos que el concepto básico de ejecución es determinar por medio de una búsqueda exhaustiva y muy precisa de entre todos los términos cuales posiblemente sean incluidos en una función posteriormente mas simplificada. Estos términos sera posible encontrarlos por medio de formar parejas con los términos de la función que solo tengan como diferencia una sola variable, variable que se elimina para encontrar un nuevo termino que tenga una literal menos. Después de repetir varias veces esta eliminación y búsqueda con los nuevos términos que van surgiendo del emparejar términos diferenciados por una sola variable una y otra vez, nos daremos cuenta que fueron surgiendo términos resultantes y otros mas que no se pudieron emparejar, los cuales son los que nos van a interesar. Será entonces cuando hayamos realizado la designación de los términos, que en varias bibliografías son llamados los primeros implicados, haciendo la observación, utilizaremos en este trabajo determinación de los términos, o bien, determinación de los primeros implicados

Pero no hay mejor explicación que el que se realice una ejemplificación de esta primer parte del método de tabulación. Por esto elaboraremos la simplificación de una función y veremos paso a paso el proceso de simplificación y como se lleva a cabo cada uno de estos.

Ejemplo 2.21. Simplificar la función $F(A, B, C, D, E) = \sum m\ (0, 1, 3, 5, 7, 8, 9, 10, 13, 14, 15, 17, 21, 25, 29)$ utilizando el método de tabulación.

En primera instancia los minitérminos se convierten a su forma binaria. Se colocan en forma de lista según su orden y aprovecharemos para colocar frente a cada uno de ellos el numero de unos que como representación binaria tienen, por lo que se vera como en la tabla 2.9.

Minitérminos	*Forma Binaria*	Cantidad de 1's
m_0	00000	0
m_1	00001	1
m_3	00011	2
m_5	00101	2
m_7	00111	3
m_8	01000	1
m_9	01001	2
m_{10}	01010	2
m_{13}	01101	3
m_{14}	01110	3
m_{15}	01111	4
m_{17}	10001	2
m_{21}	10101	3
m_{25}	11001	3
m_{29}	11101	4

Tabla 2.9. Conversión de minitérminos a su forma binaria con la cantidad de unos que contienen.

Como segundo paso hay que separar los minitérminos y agruparlos por cantidad similar de unos, esto quiere decir que; el que no tenga ningún uno dejarlo solo, los términos que tengan un solo uno agruparlos, al igual que se agrupan los que tengan dos unos, luego los de tres y asi sucesivamente hasta que se hayan quedado agrupados todos los términos, en nuestro ejemplo se muestra esto en la tabla 2.10 en su primera sección, además de agregarle del lado izquierdo el numero decimal que representa al número binario. Nótese que se divide y agrupa con líneas horizontales en la tabla.

Como tercer paso debemos comparar cada uno de los minitérminos de un grupo con cada uno de los minitérminos del otro, tomando en cuenta que esta comparación debe hacerse solamente del grupo superior con el inferior inmediato, y además tomar las siguientes recomendaciones.

A) Al realizar las comparaciones, de un minitérmino con otro (par de minitérminos) si estos difieren en una sola variable se pueden combinar y las variables no coincidentes eliminar.

Decimal	A	B	C	D	E	
0	0	0	0	0	0	√
1	0	0	0	0	1	√
8	0	1	0	0	0	√
3	0	0	0	1	1	√
5	0	0	1	0	1	√
9	0	1	0	0	1	√
10	0	1	0	1	0	√
17	1	0	0	0	1	√
7	0	0	1	1	1	√
13	0	1	1	0	1	√
14	0	1	1	1	0	√
21	1	0	1	0	1	√
25	1	1	0	0	1	√
15	0	1	1	1	1	√
29	1	1	1	0	1	√

Combinacion	A	B	C	D	E	
0,1	0	0	0	0	X	√
0,8	0	X	0	0	0	√
1,3	0	0	0	X	1	√
1,5	0	0	X	0	1	√
1,9	0	X	0	0	1	√
1,17	X	0	0	0	1	√
8,9	0	1	0	0	X	√
8,10	0	1	0	X	0	
3,7	0	0	X	1	1	√
5,7	0	0	1	X	1	√
5,13	0	X	1	0	1	√
5,21	X	0	1	0	1	√
9,13	0	1	X	0	1	√
9,25	X	1	0	0	1	√
10,14	0	1	X	1	0	
17,21	1	0	X	0	1	√
17,25	1	X	0	0	1	√
7,15	0	X	1	1	1	√
13,15	0	1	1	X	1	√
13,29	X	1	1	0	1	√
14,15	0	1	1	1	X	
21,29	1	X	1	0	1	√
25,29	1	1	X	0	1	√

Combinacion	A	B	C	D	E
(0,1),(8,9)	0	X	0	0	X
(0,8),(1,9)	0	X	0	0	X
(1,3),(5,7)	0	0	X	X	1
(1,5),(3,7)	0	0	X	X	1
(1,5),(9,13)	0	X	X	0	1
(1,5),(17,21)	X	0	X	0	1
(1,9),(5,13)	0	X	X	0	1
(1,9),(17,25)	X	X	0	0	1
(1,17),(5,21)	X	0	X	0	1
(1,17),(9,25)	X	X	0	0	1
(5,7),(13,15)	0	X	1	X	1
(5,13),(7,15)	0	X	1	X	1
(5,13),(21,29)	X	X	1	0	1
(5,21),(13,29)	X	X	1	0	1
(9,13),(25,29)	X	1	X	0	1
(9,25),(13,29)	X	1	X	0	1
(17,21),(25,29)	1	X	X	0	1
(17,25),(21,29)	1	X	X	0	1

Combinacion	A	B	C	D	E	
(0,1)(8,9)	0	X	0	0	X	
(1,3)(5,7)	0	0	X	X	1	
(1,5)(9,13)	0	X	X	0	1	√
(1,5)(17,21)	X	0	X	0	1	√
(1,9)(17,25)	X	X	0	0	1	√
(5,7)(13,15)	0	X	1	X	1	
(5,13)(21,29)	X	X	1	0	1	√
(9,13)(25,29)	X	1	X	0	1	√
(17,21)(25,29)	1	X	X	0	1	√

Combinacion	A	B	C	D	E
[(1,5),(9,13)], [(17,21),(25,29)]	X	X	X	0	1
[(1,5),(17,21)], [(9,13),(25,29)]	X	X	X	0	1
[(1,9),(17,25)], [(5,13),(21,29)]	X	X	X	0	1

Combinacion ultima de la cuarta seccion	A	B	C	D	E
[(1,5),(9,13)], [(17,21),(25,29)]	X	X	X	0	1

Tabla 2.10. Agrupación, selección y combinación de minitérminos para la simplificación.

B) Al combinar dos minitérminos y eliminar una variable en el lugar de esta podemos colocar una “X” sabiendo que ahí hay variables que difieren o son diferentes, esta combinación se coloca en una siguiente sección (2da sección) o cuadro como se muestra en la tabla 2.10.

C) Como ya se había comentado las comparaciones se dan de cada uno de los términos del grupo siguiente inferior inmediato ya que todos estos se diferencian por que tienen un uno de mas y si se quisieran comparar con otro grupo de mas abajo o mas adelante la diferencia no seria solo de contener un uno mas, si no de dos o mas de tal forma que no se podría simplificar una variable.

D) Una vez utilizados los minitérminos en una combinación deberán marcarse con “√” para darnos cuenta que ya los usamos en alguna combinación.

E) Como se comento en el inciso B) los términos resultantes de las combinaciones se colocaron en una segunda sección junto con los decimales de los términos que se usaron en dicha combinación que dio origen al término nuevo resultante, ver nuevamente la tabla 2.10 en su segunda sección.

Así que dirigiendo estas recomendaciones a nuestro ejemplo observamos lo siguiente:

De la primera sección se realiza la combinación de cada uno de los términos del primer grupo que en nuestro ejemplo solo m_0 (00000) con los dos términos de nuestro segundo grupo, vemos que se pueden combinar m_0 con m_1 ya que m_0 (00000) y m_1 (00001), quedando en nuestra siguiente sección anotados con la combinación 0,1 con (0000X) y la “X” por que cambia en esa variable y en esa posición. La siguiente combinación y que también se puede llevar a cabo es m_0 pero ahora con m_8 por lo que m_0 (00000) y m_8 (01000) quedan en la siguiente sección como 0,8 con (0X000) quedando la “X” en la posición en la que las variables cambian. También recordemos que hay que revisar el señalizar con una “√” cuando los minitérminos se hayan usado en alguna combinación. Se procede así a realizar las combinaciones del segundo grupo con el tercero uno a uno. Se obtiene una vez realizadas todas las combinaciones la segunda sección de la tabla 2.10.

Para desarrollar la tercera sección de la tabla 2.10 se deben tomar en cuenta los mismos pasos que se llevaron a cabo para el desarrollo de la segunda sección, que son; realizar las combinaciones de cada uno de los minitérminos del primer grupo con cada uno de los minitérminos del segundo grupo inferior inmediato pero de la segunda sección de la tabla 2.10, los cuales coincidirán al tener una “X” en la misma posición y que en las demas posiciones solo difieran por una variable, para que en esa posición en que difieren se coloque una “X” mas e ir marcando con una “√” a los minitérminos ya utilizados, y por supuesto estar colocando los términos combinados en una nueva siguiente sección, colocando de su lado izquierdo los numeros decimales que hasta ahora han intervenido. Este proceso se sigue en secciones subsiguientes de la tabla que se va formando.

De la tabla 2.10 existe una sección igual a la sección tercera, solo que esta es simplificada, en realidad en esta tesis se anexa esta sección extra para que el lector observe que en el surgimiento de las combinaciones existen algunas dentro de los mismos grupos que son idénticas o se repiten, observe la tercera sección, por ejemplo la combinación (0,1)(8,9) 0X00X, es idéntica a la combinación (0,8)(1,9) 0X00X, así que solo se toma en cuenta una en la misma sección tercera pero simplificada que es (0,1)(8,9) 0X00X. Lo anterior esta justificado por los teoremas y postulados de De Morgan en donde X + X = X, así que solo se presenta una de las dos combinaciones idénticas, esto lo observara el lector en este único ejemplo, ya que es con una intención de que la comprensión sea amplia y no se tenga dudas de donde y por que desaparecen algunas combinaciones. Así que existen en la tercera sección combinaciones idénticas que se deberán simplificar o colocar simplemente alguna de estas. Mas adelante se dará por asentado que se conoce este paso y se colocaran combinaciones no repetidas.

En la cuarta sección también resultan combinaciones idénticas en su totalidad por lo que solo se deja una combinación.

Finalmente el cuarto paso es localizar los términos que no hayan sido marcados por "√" y así estos serán los términos buscados, estando completa la "Determinación de Términos", o también llamados primeros implicados.

En nuestro ejemplo las combinaciones no marcadas son de la segunda sección de la tabla 2.10, la combinación (8,10) con 010X0, (10,14) con 01X10 y (14,15) con 0111X. De la tercera sección la combinación (0,1)(8,9) con 0X00X, (1,3)(5,7) con 00XX1 y (5,7)(13,15) con 0X1X1 y finalmente en la cuarta sección y que ya no puede ser combinado es [(1,5)(9,13)][(17,21)(25,29] con XXX01 dando un total de siete combinaciones en la cuales se encuentra una función mas simplificada. Sabemos que estos siete términos en cada uno de ellos existen las variables ya sea negadas 0´s o no negadas 1´s y después del proceso las X´s en posiciones donde los ceros o unos desaparecieron o eliminaron, también se están eliminando variables de las 5 que se venían manejando en el orden de A, B, C, D y E. Por tanto las variables resultantes son como se muestra en la tabla 2.11

Combinación	*A B C D E*	*Termino final*
(8,10)	0 1 0 X 0	A´BC´E´
(10,14)	0 1 X 1 0	A´BDE´
(14,15)	0 1 1 1 X	A´BCD
(0,1)(8,9)	0 X 0 0 X	A´C´D´
(1,3)(5,7)	0 0 X X 1	A´B´E
(5,7)(13,15)	0 X 1 X 1	A´CE
[(1,5)(9,13)][(17,21)(25,29]	X X X 0 1	D´E

Tabla 2.11. Lista de la determinación de términos.

De la tabla 2.11 podemos obtener una función simplificada F = A´BC´E´ + A´BDE´ + A´BCD + A´C´D´ + A´B´E + A´CE + D´E, utilizando todos los términos finales.

Aunque se haya logrado una simplificación importante de la función inicial hasta esta ultima, esta es solo una de las dos etapas en las que se obtendrá la máxima simplificación, por que se ha observado a través de la practica que la determinación de términos en esta etapa del tabulado puede llevarnos a encontrar una función simplificada como ya no reducible o sea su máxima simplificación, pero también se detecta que existen funciones que han podido ser mayormente simplificadas con alguna otra técnica mientras que con la determinación de estos términos no, por eso quizás esta siempre tenga que complementarse con la selección de los términos encontrados, para así encontrar la simplificación máxima y que veremos a continuación.

2.5.2. Selección de los términos

El primer paso de la técnica del tabulado nos puede entregar una función en su mínima expresión, pero para estar seguros debemos ocupar lo que será la selección de términos para entregarnos con seguridad una función máximamente minimizada. Como ya se había comentado, deberemos aplicar un proceso tras otro para poder obtener la simplificación, y esta simplificación mínima se obtiene por medio de la selección de términos, términos que determinamos en el subtema anterior y los cuales trabajaremos para ilustrar el subtema presente.

Utilizaremos así la función del ejemplo anterior o más bien dicho sus términos determinados al final.

Esta selección funciona acomodando a nuestros términos determinados o comúnmente llamados primeros implicados en una tabla. Esta tabla contendrá en una primera columna nuestros términos determinados, las siguientes columnas se definirán de acuerdo a la cantidad de términos mínimos que contengas nuestros términos a manejar y son columnas que estarán bacías pero que en su parte superior tendrán como títulos a los minitérminos implicados. En las columnas de los minitérminos serán llenados por algún tipo de marca, y para no variar ocuparemos la marca "√". Estas marcas se pondrán a cada columna (minitérmino) que constituya a nuestros términos. Luego finalmente se seleccionan los términos que abarquen a todos los minitérminos, para obtener la función minimizada, pero mejor aplicamos la resolución por este método a nuestro ejemplo anterior para que se pueda visualizar mejor y se vaya explicando a detalle los pasos a seguir, y decir ¿por qué no? Si era o no la mínima expresión, y ver cual es en caso de que no fuera la mas simple.

Ejemplo 2.22. Minimizar la función final encontrada en el ejemplo 2.21, en el caso de que tuviera una minimización mayor.

Empezamos por decir que la función es; F = A´BC´E´ + A´BDE´ + A´BCD + A´C´D´ + A´B´E + A´CE + D´E

Tiene esta función siete términos que fueron encontrados después de ya un proceso anterior. Seguiremos los siguientes pasos.

A) Primero procederemos a armar la tabla para la selección de los términos. Esta contendrá varias columnas y filas.

En la primera columna ubicaremos a nuestros términos A´BC´E´, A´BDE´, A´BCD, A´C´D´, A´B´E, A´CE y D´E, y por supuesto las filas las definirán estos mismos términos. Véase la tabla 2.12.

Se recomienda dentro de este proceso de armado de la tabla para la selección incluir una columna donde se colocaran los números decimales que tiene el término.

	Términos	*No. de Combinación (Decimal)*	*0*	*1*	*3*	*5*	*7*	*8*	*9*	*10*	*13*	*14*	*15*	*17*	*21*	*25*	*29*
	A´BC´E´	8,1															
	A´BDE´	10,14															
	A´BCD	14,15															
	A´C´D´	0,1,8,9															
	A´B´E	1,3,5,7															
	A´CE	5,7,13,15															
	D´E	1,5,9,13,17,21,25,29															

Tabla 2.12. Para la selección de términos.

Las siguientes columnas están vacías pero en su fila superior encontramos el acomodo de los números que representan cada minitérmino que interviene en los términos y en forma ordenadamente ascendente, será aquí donde llenaremos con una marca.

Nótese también que debajo de la fila del último término se encuentra una fila vacía mas, que también será llenada, además también se encuentra una pequeña columna o espacio del lado izquierdo de la tabla y que también se llenaran pero con un "*".

B) El llenado es sencillo y solo consta de ir colocando "√" en las filas de los términos, pero solo en las columnas que coinciden con el numero decimal o de minitérmino que tengan dichos términos. Obsérvese la primera fila de la tabla 2.13 como los números decimales que corresponden a la combinación del término son 8,10 por lo cual se marca con "√" la columna 8 y 10. La fila dos tiene los decimales de combinación 10,14 por lo que se llena la columna 10 y 14 con la "√" y así sucesivamente hasta terminar con la última fila.

	Términos	*No. de Combinación (Decimal)*	*0*	*1*	*3*	*5*	*7*	*8*	*9*	*10*	*13*	*14*	*15*	*17*	*21*	*25*	*29*
	A´BC´E´	8,1						√		√							
	A´BDE´	10,14								√							
	A´BCD	14,15										√	√				
*	A´C´D´	0,1,8,9	√	√				√	√								
*	A´B´E	1,3,5,7		√	√	√	√										
	A´CE	5,7,13,15				√	√				√		√				
*	D´E	1,5,9,13,17,21,25,29		√		√			√		√			√	√	√	√
			√	√	√	√	√	√	√		√			√	√	√	√

Tabla 2.13. Selección de términos con la marca “√”

C) Lo siguiente es seleccionar. Para seleccionar los términos primero debemos observar cuidadosamente la tabla 2.13 y buscar aquellas marcas que se encuentran solas en las columnas esto quiere decir una sola marca en una columna, y esta marca pertenece a un término, al cual marcaremos con “*” del lado izquierdo, esto se hace con la intención de que estos minitérminos estén incluidos en la simplificación final, observemos la tabla 2.13, de la cual veremos como los minitérminos 0, 3, 17, 21, 25 y 29 son minitérminos que tienen al menos una columna con una sola marca. Así el minitérmino de 0 se cubre por A’C’D’ ya marcado con el “*”, luego el minitérmino 3 que esta contenido por A’B’E al cual también se le coloco del lado izquierdo su “*”, ya los minitérminos 17, 21, 25 y 29 los cubre el término D’E. A estos primeros términos ya seleccionados se les puede tomar como términos imprescindibles o esenciales, y de aquí se parte para seleccionar los demás.

Posteriormente vemos de la tabla 2.13 como los términos indispensables si contienen el minitérmino que se encuentra solo, pero a su vez contiene a uno o varios minitérminos mas, por lo que en la fila de abajo debemos colocar otra marca “√” primero para los minitérminos que se encuentran solos en la columna por estar cubiertos o contenidos en los términos esenciales y luego también colocar esas mismas marcas a los minitérminos que también se encuentran contenidos por esos términos indispensables. Por ejemplo el termino indispensable A’C’D’ cuenta con el minitérmino 0 (que se encuentra solo) y además el 1, 8 y 9 que proyectando a la ultima fila de la tabla tendrán que llevar una marca en su columna. De la misma forma para A’B’E que contiene al minitérmino 3 (que se encuentra solo) y a los minitérminos 1, 5 y 7, los cuales también tendrán su marca en la fila inferior ultima de la tabla. De la misma forma todos los minitérminos incluidos en el termino esencial D’E que son 1, 5, 9, 13, 17, 21, 25 y 29 que ya tienen su marca también. Obviamente que si ya alguno tenía marca solo no se marca nuevamente.

De toda la serie de marcas en la tabla vemos que hay minitérminos no marcados como lo son 10, 14 y 15, pero sabemos debemos contemplarlos. De la observación cuidadosa vemos como existen términos que los contienen, prácticamente todos los términos no marcados por el “*” contienen al menos un minitérmino de estos, pero debemos de elegir aquellos que pueden contener mas de uno y si es posible todos los minitérminos faltantes y que además contenga el menor numero de literales para que la simplificación sea verdaderamente la máxima, por eso en este caso elegimos el término A’BDE’ que contiene a dos minitérminos que son 10 y 14 y A’CE que contiene a 15 y que además tiene menos literales.

Así finalmente obtenemos la función simplificada, uniendo los términos esenciales con estos dos últimos que contienen los últimos minitérminos. Por lo que tenemos:

$$F = A'C'D' + A'B'E + D'E + A'BDE' + A'CE$$

De esta manera esta manera vemos finalizado el proceso. Cabe mencionar que el ejemplo se aplico para minitérminos o suma de productos, pero también se puede aplicar para producto de sumas, solo basta con que se tomen en cuenta los ceros en lugar de los unos, como si fueran los minitérminos, sabiendo de antemano que son los términos no incluidos en la función original y son finalmente el complemento. El proceso es el mismo y ya al final sigue quedando en suma de productos nada que no se pueda arreglar obteniendo de nuevo el complemento y dejándolo como producto de sumas.

Este método del tabulado puede parecer largo pero es mas bien por que en esta tesis se da la explicación de cómo es el proceso de ocupación, pero una vez que se aplica es quizás equitativo al estar haciendo los cuadros para realizar los diagramas de cuatro, cinco o seis variables, pero con la seguridad de una mayor efectividad al asegurar que se encuentra una función lo mas simplificada.

2.6 Técnicas de Quine – Mc. Cluskey

Esta técnica es llamada así debido a los personajes que la desarrollaron; Willard Van Orman Quine y Edward J. Mc.Cluskey; quienes se basaron en la minimización de funciones booleanas.

La técnica de Quine-Mc.Cluskey se utiliza para la simplificación de funciones es también un método tabular, de hecho el método tabular con anterioridad visto utiliza estas técnicas. De los ejemplos antes vistos en el tema del Método del Tabulado se vio como este método resolvió funciones completas y cabe aclarar con anticipación que en este punto también se analizara y explicara el proceso hacia la simplificación de la función completa, incompleta y la complemento; solo que se aplicará mas la técnica de Quine-Mc.Cluskey y así ir conjuntando cada vez mas herramientas que nos ayuden a simplificar de manera mas sencilla y eficaz a las funciones lógicas.

Además, de que por ser un método viable para la simplificación de funciones que contienen a un número mayor de variables, a diferencia de los otros métodos, como lo es el de mapas de Karnaugh. El concepto básico sobre el que trabaja este método es que nos ayuda a encontrar de una forma ordenada a todos los minitérminos adyacentes que se encuentran dentro de una función.

2.6.1. Función completa

Ya que el método tabular está basado en la técnica de Quine – Mc.Cluskey, esta técnica contiene los cuatro pasos fundamentales que se mencionan en el tabular como proceso del mismo y de los cuales hacemos mención nuevamente para la minimización de una función completa:

Primero.- Se enumeran o enlistan todos los minitérminos de la función a simplificar en una tabla, pero por supuesto en su forma binaria. Hacer la separación en grupos de minitérminos que se diferencien solo por un bit o variable, de tal forma que esto ayudara a localizar mas fácilmente los minitérminos adyacentes entendiendo que esta adyacencia de los minitérminos depende de que solo difieran entre ellos en una variable.

Segundo.- Se realiza la búsqueda minuciosa de los minitérminos adyacentes entre grupos vecinos y crear una nueva columna de minitérminos con adyacencia a también llamados implicantes, pero ahora estos tendrán n – 1 variables, y marcar con una "√" cada minitérmino incluido en la formación de estos nuevos implicantes. En la nueva columna donde se van colocando los nuevos implicantes pero en su representación binaria, se colocan guiones en posición de la variable que se haya eliminado. Este proceso se repite para la siguiente columna que se formen y se van creando así nuevas columnas, pero ahora con implicantes de n – 2 variables y así sucesivamente, hasta que no se puedan unir mas implicantes. De tal forma que al final de la creación de todas las columnas encontraremos implicantes que no tuvieron adyacencia con otros o no fueron absorbidos, estos serán los primeros implicantes o implicados de la función de conmutación.

Tercero.- Se construye una nueva tabla con los primeros minitérminos implicados de forma vertical y de forma horizontal los minitérminos no implicados, para colocar posteriormente una “X” cuando cierto primer implicado (fila) cubra un minitérmino (columna).

Cuarto.- Se selecciona a un número mínimo de primeros implicados o más conocidos en la sección anterior (2.5.2) como selección de términos, en el cual estos términos son seleccionados por cubrir a todos los minitérminos de la función.

Así, de esta forma, procederemos a llevar a cabo cada uno de los pasos anteriores, para realizar la simplificación de una función, por lo tanto iniciaremos la utilización del método con el siguiente ejemplo.

Ejemplo 2.23. Con el Método de Quine – Mc. Cluskey haga la simplificación de la función: $F(A,B,C,D,E) = \sum m\ (1,2,3,4,8,9,13,18,20,21,22,23,24,28)$.

Aplicaremos el primer paso para iniciar la simplificación y este nos indica que debemos agrupar los minitérminos según el numero de unos que tengan en su representación binaria; los que tengan un solo uno formaran un grupo, los que tengan dos unos formaran otro, y así sucesivamente y avecindarlos de manera en que cada grupo difiera de su vecino por solo un uno, como se muestra en la tabla 2.14

Como segundo paso se procederá a combinar minitérminos de entre grupos vecinos y marcamos con una “√” los minitérminos ya utilizados. Esta combinación nueva se colocara en otra sección anexa a la primera tabla, por lo cual se generara una tabla general con mas secciones, según avance el proceso, como se muestra en la tabla 2.15 en donde ya se encuentran generadas las demás secciones y que explicaremos como se van obteniendo.

Por el momento observamos como de la tabla 2.14 que tenemos se integra con una primera sección y como después de la combinación de los grupos ya se genera la segunda sección. Mencionamos además que la combinación de minitérminos de entre un grupo y otro solo se hacen de un grupo al próximo inmediato, porque entre ellos solo existe un uno de diferencia, esto quiere decir que el primero se combina solo con el segundo, y el segundo solo con el tercero, y así sucesivamente.

Minitérminos	*ABCDE*
1	00001
2	00010
4	00100
8	01000
3	00011
9	01001
18	10010
20	10100
24	11000
13	01101
21	10101
22	10110
28	11100
23	10111

Tabla 2.14. Acomodo de Minitérminos según cantidad de unos.

Una vez generada la segunda sección por las combinaciones vemos que en donde existió un bit de diferencia se coloco un “_“.

Cuando ya tenemos la segunda sección, con esta realizamos el mismo proceso que con la primera, para generar una tercera sección.

De la segunda sección debemos tomar en cuenta que de un grupo a otro solo podremos combinar aquellos minitérminos que se diferencian por solo una literal, pero que también solo podemos combinar los términos que tienen la misma literal ausente, o sea un guion en la misma posición.

PRIMERA SECCION

Minitérminos	A	B	C	D	E	
1	0	0	0	0	1	√
2	0	0	0	1	0	√
4	0	0	1	0	0	√
8	0	1	0	0	0	√
3	0	0	0	1	1	√
9	0	1	0	0	1	√
18	1	0	0	1	0	√
20	1	0	1	0	0	√
24	1	1	0	0	0	√
13	0	1	1	0	1	√
21	1	0	1	0	1	√
22	1	0	1	1	0	√
28	1	1	1	0	0	√
23	1	0	1	1	1	√

SEGUNDA SECCION

Minitérminos	A	B	C	D	E	
1,3	0	0	0	_	1	PI_2
1,9	0	_	0	0	1	PI_3
2,3	0	0	0	1	_	PI_4
2,18	_	0	0	1	0	PI_5
4,20	_	0	1	0	0	PI_6
8,9	0	1	0	0	_	PI_7
8,24	_	1	0	0	0	PI_8
9,13	0	1	_	0	1	PI_9
18,22	1	0	_	1	0	PI_{10}
20,21	1	0	1	0	_	√
20,22	1	0	1	_	0	√
20,28	1	_	1	0	0	PI_{11}
24,28	1	1	_	0	0	PI_{12}
21,23	1	0	1	_	1	√
22,23	1	0	1	1	_	√

TERCERA SECCION

Minitérminos	A	B	C	D	E	
(20,21),(22,23)	1	0	1	_	_	PI_1
(20,22),(21,23)	1	0	1	_	_	

Tabla 2.15. Manejo de los Términos para encontrar a los Primeros Implicados (PI_n).

Una aclaración que es necesario hacer, es que, cuando no se puedan combinar los términos, en lugar de colocarles una marca de "√", se colocara una "PI_n", donde; "PI" significa Primer Implicado y "n" es un subíndice que nos indica el número de los primeros implicados. Este enumerar se inicia a partir del último primer implicado encontrado, hacia atrás. Ver figura 2.15.

De la misma tabla debemos observar que en la tercera sección solo se señalo un primer implicado de los dos que están ahí, y la razón es que son idénticos, tanto tienen las mismas variables como también las ausentes.

Una herramienta extra que nos brinda Quine-Mc Cluskey para verificar que si hemos eliminado la variable correcta en las secciones primera, segunda y tercera es que;

restamos los números de entrada o de minitérminos que se hayan combinado, ya que el resultado que arroje será la variable que se debió eliminar. Por ejemplo en la combinación 4,20 que es _0100 de la segunda sección, se debe eliminar la variable que se obtenga de la resta de 20 – 4 o sea 16, esto es 20 – 4 = 16 y que vemos fue la variable eliminada, ya que el orden de las variables es 16, 8, 4, 2, 1 y en el lugar del 16 se encuentra el "_". Para la combinación (20, 21) – (22, 23) se tiene 101_ _ y las restas son:

21 – 20 = 1	22 – 20 = 2
23 – 22 = 1	23 – 21 = 2

De modo que deben estar eliminadas las variables que se encuentran en la posición de 1 y 2, procedemos a ver las posiciones y observamos que así es, ya que nuevamente las posiciones son 16, 8, 4, 2, 1 y los lugares que ocupan el 2 y 1 están con "_".

Se podría ahora que se tienen los primeros implicantes primos mostrarse como en una suma para rehacer la función, pero lo que se busca con este método es tener el menor numero de implicados, eso si, los necesarios también. Por esta razón seguimos con el próximo paso.

En este paso, el tercero, formaremos nuevamente una tabla de primeros implicantes.

Dicha tabla, la 2.16 se forma por los primeros implicados en forma vertical y por los miniterminos en forma horizontal. Como ya se había comentado con anterioridad se colocara una "X" cuando un primer implicado que se encuentra en una fila cubra un minitérmino de los que se encuentran en columna.

	1	2	3	4	8	9	13	18	20	21	22	23	24	28
				√		√	√		√	√	√	√		
** PI_1									X	X	X	X		
PI_2	X		X											
PI_3	X					X								
PI_4		X	X											
PI_5		X						X						
** PI_6				X					X					
PI_7					X	X								
PI_8					X								X	
** PI_9						X	X							
PI_{10}								X			X			
PI_{11}									X					X
PI_{12}													X	X

Tabla 2.16. Selección de los primeros implicados esenciales.

Una vez construida la tabla 2.16 se procede como cuarto paso a analizar esta, para poder seleccionar el numero mínimo necesario de primeros implicados.

Se analizan las columnas de los minitérminos una por una y encontramos que los minitérminos 4, 13, 21 y 23 están cubiertas cada uno por un solo primer implicado (a estos se les encierra la marca con un circulo y en la parte superior de la columna se marcan con una "√") y a estos primeros implicados PI_1, PI_6 y PI_9 los llamamos los primeros implicados esenciales (obsérvese como se les coloca un doble asterisco). Al elegir a estos primeros implicados esenciales, también estamos eligiendo los minitérminos que se encuentran en sus filas, que son; 9, 20, y 22, a los cuales también se les marcan con "√" en la parte superior de su columna.

De tal forma observamos que en la parte superior de la tabla 2.16 todos los minitérminos ya cubiertos que son: 4, 9, 13, 20, 21, 22 y 23 están marcados por la "√".

Después se procederá a elegir de los primeros implicados no esenciales (los no marcados o sobrantes) que cubran al resto de los minitérminos. Formamos a continuación una tabla con estos primeros implicados no esenciales y con sus correspondientes minitérminos a cubrir. Tabla 2.17.

En la elección de los primeros implicados de la tabla 2.17 debe tomarse en cuenta que debe ser el mínimo necesario para cubrir a los minitérminos de la parte superior, así que hay que observar cuidadosamente.

Podríamos empezar por ir descartando aquellos primeros implicados que no cubren muchos minitérminos y que estos minitérminos que cubren ya estén cubiertos por otros implicados. Así que descartaríamos a PI_3, PI_{11}, PI_{10} y PI_7, debido a que estos solo cubren un minitérmino, que además ya esta cubierto por algún otro primer implicado y con mayor numero de minitérminos cubiertos, los cuales son: PI_2, PI_5, PI_8, y PI_{12}.

	1	2	3	8	18	24	28
PI_2	X		X				
PI_3	X						
PI_4		X	X				
PI_5		X			X		
PI_7				X			
PI_8				X		X	
PI_{10}					X		
PI_{11}							X
PI_{12}						X	X

Tabla 2.17. Selección de los últimos primeros implicados que cubran al restante de minitérminos.

Por otra parte, también se tiene a PI_4 a pesar de que tiene cubiertos a dos minitérminos, estos ya están cubiertos por otros implicados, así que también queda fuera de la contemplación, por lo que queda cubierto todo.

Finalmente queda una función mínima de la original.

$F(A, B, C, D, E) = PI_1 + PI_2 + PI_5 + PI_6 + PI_8 + PI_9 + PI_{12}$

F(A, B, C, D, E) = 101- - + 000-1 + -0010 + -0100 + -1000 + 01-01 + 11-00

F(A, B, C, D, E) = AB’C + A’B’C’E + B’C’DE’ +B’CD’E’ + BC’D’E’ + A’BD’E + ABD’E’

De esta manera termina la simplificación de una función completa por el método de Quine – Mc. luskey. A continuación se llevara el análisis para la simplificación de la función incompleta.

2.6.2. Función incompleta

La minimización de las funciones incompletas se realiza de manera similar a la realizada en una función completa, de hecho de una forma idéntica, solo que en las funciones incompletas se están agregando términos prescindibles. Estos términos hay que recordar que se están utilizando para completar una función y así pueda tenerse una mejor agrupación de términos que nos pueda llevar a una mejor simplificación de las funciones. Ahora el proceso si cambia al instante de agregar los términos prescindibles ya que el cambio se da en el proceso de formar la tabla de los primeros implicados esenciales. Ya esta situación se tratara en el siguiente ejemplo en el cual también habrá que recalcar la diferencia a seguir en la simplificación de la función incompleta.

Ejemplo 2.24.- Simplificar la siguiente función utilizando el Método de Quine – Mc. Cluskey: F(A, B, C, D, E) = ∑m (2,3,7,10,12,15,27) + d (5,18,19,21,23)

El primer paso a realizar como se ha hecho en técnicas anteriores, es acomodar de manera ordenada los minitérminos, primero obteniendo los unos y ceros que correspondan a cada minitérmino y luego agruparlos por cantidad de unos, esto se realiza en la tabla 2.18 que a continuación se muestra.

Minitérminos	ABCDE
2	00010
3	00011
5	00101
7	00111
10	01010
12	01100
15	01111
18	10010
19	10011
21	10101
23	10111
27	11011

Minitérminos	ABCDE
2	00010
3	00011
5	00101
10	00111
12	01010
18	01100
7	01111
19	10010
21	10011
15	10101
23	10111
27	11011

Tabla 2.18. Acomodo por grupos según cantidad de unos por término.

Posteriormente procederemos a realizar la búsqueda los primeros implicados como también ya se realizo en su momento y así se obtiene la tabla 2.19.

Ya una vez realizada nuestra búsqueda procedemos a armar nuestra próxima tabla de primeros implicados esenciales para poder elegir de ellos cuales si son esenciales en verdad, pero si mencionar que como la función es incompleta la diferencia será aquí, esto debido a que en la lista de términos contemplada en el ejemplo que se esta tratando, existen términos prescindibles y que no hay necesidad de cubrir, serian; 5, 18, 19, 21, 23. Mientras que los que si deben estar y ser cubiertos en la tabla son; 2, 3, 7, 10, 12, 15 y 27, por lo que solo aparecerán estos en la tabla 2.20.

PRIMERA SECCION						
MINITERMINOS	A	B	C	D	E	
2	0	0	0	1	0	√
3	0	0	0	1	1	√
5	0	0	1	0	1	√
10	0	1	0	1	0	√
12	0	1	1	0	0	√
18	1	0	0	1	0	√
7	0	0	1	1	1	√
19	1	0	0	1	1	√
21	1	0	1	0	1	√
15	0	1	1	1	1	√
23	1	0	1	1	1	√
27	1	1	0	1	1	√

SEGUNDA SECCION						
MINITERMINOS	A	B	C	D	E	
2,3	0	0	0	1	-	√
2,10	0	-	0	1	0	PI_4
2,18	-	0	0	1	0	√
3,7	0	0	-	1	1	√
3,19	-	0	0	1	1	√
5,7	0	0	1	-	1	√
5,21	-	0	1	0	1	√
18,19	1	0	0	1	-	√
7,15	0	-	1	1	1	PI_5
7,23	-	0	1	1	1	√
19,23	1	0	-	1	1	√
19.27	1	-	0	1	1	PI_6
21,23	1	0	1	-	1	√

TERCERA SECCION						
MINITERMINOS	A	B	C	D	E	
(2,3-18,19)	-	0	0	1	-	PI_1
(2,18-3,19)	-	0	0	1	-	
(3,7-19,23)	-	0	-	1	1	PI_2
(3,19-7,23)	-	0	-	1	1	
(5,7-21,23)	-	0	1	-	1	PI_3
(5,21-7,23)	-	0	1	-	1	

Tabla 2.19. Búsqueda de los primeros implicados.

		√		√	√	√	√	√
		2	3	7	10	12	15	27
	PI_1	X	X					
	PI_2		X	X				
	PI_3			X				
**	PI_4	X			X			
**	PI_5			X			X	
**	PI_6							X
**	PI_7					X		

Tabla 2.20. Tabla de los primeros implicados esenciales con los minitérminos imprescindibles de la función incompleta.

De la tabla 2.20 tenemos que con PI_4, PI_5, PI_6 Y PI_7 se cubren casi todos los minitérminos, el único que faltaría seria el 3, pero como ya solo es uno no habrá necesidad de realizar una tabla de implicados mas reducida, ya que este termino 3 puede ser cubierto por PI_1 o PI_2. Por lo tanto habrá dos mínimas simplificaciones finales que serian:

$F(A, B, C, D, E) = PI_1 + PI_4 + PI_5 + PI_6 + PI_7$

O también

$F(A, B, C, D, E) = PI_2 + PI_4 + PI_5 + PI_6 + PI_7$

Ya hablando de las variables de cada uno de los primeros implicados tendríamos:

F(A, B, C, D, E) = B’C’D + A’C’DE’ + A’CDE + AC’DE + A’BCD’E’

Y por lo tanto también podría ser:

F(A, B, C, D, E) = B’DE + A’C’DE’ + A’CDE + AC’DE + A’BCD’E’

De esta forma se terminaría la simplificación de una función incompleta que como vemos ha sido muy similar a lo ya tratado con una función completa, con la diferencia de no haber incluir en una de las ultimas tablas los minitérminos prescindibles.

2.6.3. Complemento

Haciendo uso de la información planteada en el subtema 2.3, acerca de lo que es un complemento y ya teniendo ese antecedente; nos damos cuenta que el complemento de una función F(A, B, C, D, … , n), será F’(A, B, C, D, … , n). Así que como en los métodos anteriores se ha hablado de este complemento y en algunos casos se ha empleado; en el método de Quine – Mc.Cluskey no es la excepción por lo que los ceros pueden emplearse para obtener una solución en suma de productos mínima, que después puede ser transformada de un producto de sumas como se mostrara en el siguiente ejemplo, en donde se utilizara el complemento de la función obtenida del ejemplo 2.24.

Ejemplo 2.25. Minimizar el complemento de la función:

$F(A, B, C, D, E,) = \sum m\ (2, 3, 7, 10, 12, 15, 27) + \sum d\ (5, 18, 19, 21, 23)$

Para iniciar el proceso de minimización necesitamos primero el complemento de la función y este es el siguiente:

$F(A, B, C, D, E,) = \sum m\ (0,1,4,6,8,9,11,13,14,16,17,20,22,24,25,26,28,29,30,31) + \sum d(5, 18, 19, 21, 23)$

Además, observamos que los términos no esenciales también se incluirán ya que son necesarios para la simplificación de la función, esto, en los mismos pasos en los que se hizo para la función incompleta, esto quiere decir que en las tablas en donde se buscan los primeros implicados esenciales ya no estarán contemplados. Así tomamos la primera tabla en la que se encuentran los primeros implicados y es la tabla 2.21.

La revisión de la tabla 2.21 una vez terminada, arroja los primeros implicados, con lo que se procederá a realizar ahora la tabla para la búsqueda de los primeros implicados esenciales, y solo cabe recordar una vez mas que aquí ya no se integraron los términos no esenciales que en un principio si se utilizaron, siendo esta tabla de la que hablamos la 2.22.

Se encontraron de la tabla 2.22 los primeros implicados esenciales PI_1, PI_5, PI_6, PI_9, PI_{10} y PI_{11}. Ahora hay que observar en esta tabla detenidamente y notamos como los únicos términos no cubiertos son 4 y 20, pero estos podríamos cubrirlos con el PI_2 o también con el PI_3, de esta forma se cubrirían los términos restantes para PI_3, PI_4 y

PI_7. Además aclarar que el PI_8 esta cubierto en todos sus términos por el resto de los PI. Por lo tanto no se ve necesidad de formar una nueva tabla para analizar mas a detalle, así que el caso que analizamos ayuda enormemente ya que solo quedan dos términos por cubrir.

PRIMERA SECCION						
MINITER.	A	B	C	D	E	
0	0	0	0	0	0	√
1	0	0	0	0	1	√
4	0	0	1	0	0	√
8	0	1	0	0	0	√
16	1	0	0	0	0	√
6	0	0	1	1	0	√
9	0	1	0	0	1	√
17	1	0	0	0	1	√
20	1	0	1	0	0	√
24	1	1	0	0	0	√
11	0	1	0	1	1	√
13	0	1	1	0	1	√
14	0	1	1	1	0	√
22	1	0	1	1	0	√
25	1	1	0	0	1	√
26	1	1	0	1	0	√
28	1	1	1	0	0	√
29	1	1	1	0	1	√
30	1	1	1	1	0	√
31	1	1	1	1	1	√

SEGUNDA SECCION						
MINITER.	A	B	C	D	E	
0,1	0	0	0	0	-	√
0,4	0	0	-	0	0	√
0,8	0	-	0	0	0	√
0,16	-	0	0	0	0	√
1,19	0	-	0	0	1	√
1,17	-	0	0	0	1	√
4,6	0	0	1	-	0	√
4,20	-	0	1	0	0	√
8,9	0	1	0	0	-	√
8,24	-	1	0	0	0	√
16,17	1	0	0	0	-	√
16,20	1	0	-	0	0	√
16,24	1	-	0	0	0	√
6,14	0	-	1	1	0	√
6,22	-	0	1	1	0	√
9,11	0	1	0	-	1	PI_{11}
9,13	0	1	-	0	1	√
9,25	-	1	0	0	1	√
17,25	1	-	0	0	1	√
20,22	1	0	1	-	0	√
20,28	1	-	1	0	0	√
24,25	1	1	0	0	-	√
24,26	1	1	0	-	0	√
24,28	1	1	-	0	0	√
13,29	-	1	1	0	1	√
14,30	-	1	1	1	0	√
22,30	1	-	1	1	0	√
25,29	1	1	-	0	1	√
26,30	1	1	-	1	0	√
28,29	1	1	1	0	-	√
28,30	1	1	1	-	0	√
29,31	1	1	1	-	1	√
30,31	1	1	1	1	-	√

TERCERA SECCION					
MINITER.	A	B	C	D	E
(0,1-8,9)	0	-	0	0	-
(0,1-16,17)	-	0	0	0	-
(0,4-16,20)	-	0	-	0	0
(0,8-1,9)	0	-	0	0	-
(0,8-16,24)	-	-	0	0	0
(0,16-1,17)	-	0	0	0	-
(0,16-4,20)	-	0	-	0	0
(0,16-8,24)	-	-	0	0	0
(1,9-17,25)	-	-	0	0	1
(1,17-9,25)	-	-	0	0	1
(4,6-20,22)	-	0	1	-	0
(4,20-6,22)	-	0	1	-	0
(8,9-24,25)	-	1	0	0	-
(8,24-9,25)	-	1	0	0	-
(16,17-24,25)	1	-	0	0	-
(16,20-24,28)	1	-	-	0	0
(16,24-17,25)	1	-	0	0	-
(16,24-20,28)	1	-	-	0	0
(6,14-22,30)	-	-	1	1	0
(6,22-14,30)	-	-	1	1	0
(9,13-25,29)	-	1	-	0	1
(9,25-13,29)	-	1	-	0	1
(20,22-28,30)	1	-	1	-	0
(20,28-22,30)	1	-	1	-	0
(24,25-28,29)	1	1	-	0	-
(24,26-28,30)	1	1	-	-	0
(24,28-25,29)	1	1	-	0	-
(24,28-26,30)	1	1	-	-	0
(28,29-30,31)	1	1	1	-	-
(28,30-29,31)	1	1	1	-	-

TERCERA SECCION SIMP.						
MINITER.	A	B	C	D	E	
(0,1-8,9)	0	-	0	0	-	√
(0,1-16,17)	-	0	0	0	-	√
(0,4-16,20)	-	0	-	0	0	PI_2
(0,8-16,24)	-	-	0	0	0	√
(1,9-17,25)	-	-	0	0	1	√
(4,6-20,22)	-	0	1	-	0	PI_3
(8,9-24,25)	-	1	0	0	-	√
(16,17-24,25)	1	-	0	0	-	√
(16,20-24,28)	1	-	-	0	0	PI_4
(6,14-22,30)	-	-	1	1	0	PI_5
(9,13-25,29)	-	1	-	0	1	PI_6
(20,22-28,30)	1	-	1	-	0	PI_7
(24,25-28,29)	1	1	-	0	-	PI_8
(24,26-28,30)	1	1	-	-	0	PI_9
(28,29-30,31)	1	1	1	-	-	PI_{10}

CUARTA SECCION						
MINITER.	A	B	C	D	E	
(0,1-8,9)(16,17-24,25)	-	-	0	0	-	
(0,1-16,17)(8,9-24,25)	-	-	0	0	-	PI_1
(0,8-16,24)(1,9-17,25)	-	-	0	0	-	

Tabla 2.21. Primeros implicados del complemento.

		√	√		√	√	√	√	√	√	√	√		√	√	√	√	√	√	√	√
		0	1	4	6	8	9	11	13	14	16	17	20	22	24	25	26	28	29	30	31
**	PI_1	x	⊗			⊗	x				x	⊗			x	x					
	PI_2	x		x							x		x								
	PI_3			x	x								x	x							
	PI_4										x		x		x			x			
**	PI_5				x					⊗				x						x	
**	PI_6						x		⊗							x			x		
	PI_7												x	x				x		x	
	PI_8														x	x		x	x		
**	PI_9														x		⊗	x		x	
**	PI_{10}																	x	x	x	⊗
**	PI_{11}						x	⊗													

Tabla 2.22. Selección de los primeros implicados esenciales del complemento.

Quedan finalmente los términos que están marcados con doble asterisco, además de PI_2 ó PI_3. Así la simplificación del complemento de la función F es:

F'(A,B,C,D,E) = $PI_1 + PI_2 + PI_5 + PI_6 + PI_9 + PI_{10} + PI_{11}$ ó bien,
F'(A,B,C,D,E) = $PI_1 + PI_3 + PI_5 + PI_6 + PI_9 + PI_{10} + PI_{11}$ y que seria igual a:

F'(A,B,C,D,E) = C'D' + B'D'E' + CDE' + BD'E + ABE' + ABC + A'BC'E
ó también,
F'(A,B,C,D,E) = C'D' + B'CE' + CDE' + BD'E + ABE' + ABC + A'BC'E

Ahora la función puede obtenerse en forma producto de sumas mínimo como:

F(A,B,C,D,E) = (C + D)(B + D + E)(C' + D' + E)(B' + D + E') (A' + B' + E)(A' + B' + C') (A + B' + C + E')

o´ de la forma

$F(A,B,C,D,E) = (C + D)(B + C' + E)(C' + D' + E)(B' + D + E') (A' + B' + E)(A' + B' + C') (A + B' + C + E')$

Y también aquí decimos finalizado el apartado de complemento con respecto a las técnicas de Quine – Mc. Cluskey.

2.7 Salida múltiple

Hasta este subtema se ha manejado solo la simplificación y minimización de una sola función, pero en este subtema se hablara de cómo una serie de funciones pueden convivir unas con otras y utilizar una serie de mismas compuertas para formar circuitos mas complejos minimos y que a su vez en el momento de ser armados tengan un costo monetariamente menor. Solo cabe mencionar que las funciones que van a convivir deben tener la misma serie de entradas, es decir; $F_1(A,B,C,D)$, $F_2(A,B,C,D)$,…, $F_M(A,B,C,D)$.

Así que al hablar de la función que tiene una serie de variables, pasaremos a hablar de distintas funciones que pueden hacer las mismas series de variables, esto quiere decir; una minimización de variables a la entrada, ya que se usaran las mismas para distintas funciones, armando así redes que tengan distintas salidas, pero, las mismas entradas, formando lo que serian ya circuitos mas complejos. Aunque precisamente al hacer convivir a una serie de funciones haga que quizás no exista una suma de productos mínima para las funciones, como lo seria si estas se manejan de forma individual, pero que economizaríamos manejándolos en conjunto que por separado.

2.7.1. Salida con mapas de Karnaugh

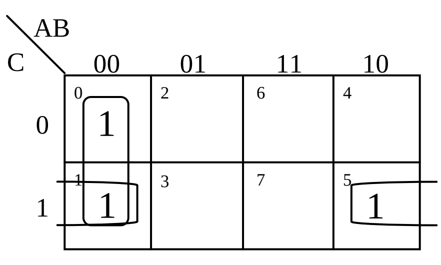
En principio, para manejar distintas funciones a través de los mapas de Karnaugh, es necesario hacer agrupaciones esenciales para el mapa de una función, y que a su vez esta agrupación de términos sea común en los distintos mapas del resto de las funciones, o sea que se encuentren dentro de estos, ya que precisamente esas agrupaciones de términos en común que tengan estos mapas, serán compuertas que se compartirán las funciones.

A continuación se realiza un ejemplo de cómo es una simplificación por separado de cada función y sus diagramas correspondientes, y como ya estas, a través de los mapas podemos encontrar términos o agrupaciones de términos en común que ayuden a formar un solo diagrama.

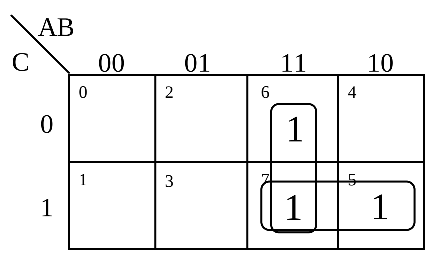
Ejemplo 2.26. Minimizar al mismo tiempo las funciones $F_1(A,B,C) = \sum m(0,1,2)$ y $F_2(A,B,C) = \sum m(5,6,7)$. También realizar el arreglo de un circuito con F1 y F2.

Lo que se realizara en este ejemplo es la simplificación de ambas funciones por separado y posteriormente se realizara la unificación.

Realizamos la colocación de los términos en el mapa para ambas funciones F_1 y F_2



$F_1 = A'B' + B'C$



$F_2 = AB + AC$

Figura 2.43. Mapa de la función F_1

Figura 2.44. Mapa de la función F_2.

Observamos que de la figura 2.43 la simplificación de la $F_1 = A'B' + B'C$ y de la figura 2.44 la $F_2 = AB + AC$ y quedarían representadas por los circuitos separados de la manera en que se muestran en la figura 2.45.

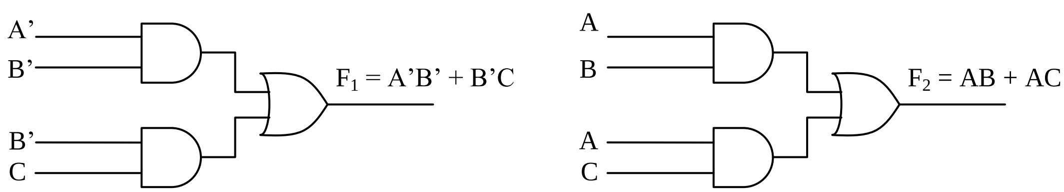


Figura 2.45. Diagrama de compuertas de las funciones F_1 y F_2.

Pero ahora veremos como a través de los mapas, de la observación y búsqueda de los términos o agrupaciones de términos en estos podemos unificar funciones para obtener estas mismas funciones reduciendo el uso de algunas compuertas, así como quizás de entradas.

De la figura 2.46 donde se muestra los mapas de Karnaugh nuevamente de F_1 y F_2, también observamos como ya solo se agrupan un par de minitérminos y no se agrupo uno que se noto es común entre los dos mapas y que es señalado en la figura.

Observando que el termino común en ambos mapas de ambas funciones es AB’C, el diagrama que representaría el circuito quedaría como en la figura 2.46 en el cual también queda utilizado el AB’C.

De esta manera obtenemos salidas múltiples en un solo circuito además que de la observación de la figura 2.45 y de la figura 2.47 concluimos que en los diagramas resultantes de la figura 2.45 de ambas funciones vemos que se utilizan 8 entradas y 6 compuertas, mientras que en el diagrama de la figura 2.47 tenemos 7 entradas y 5 compuertas.

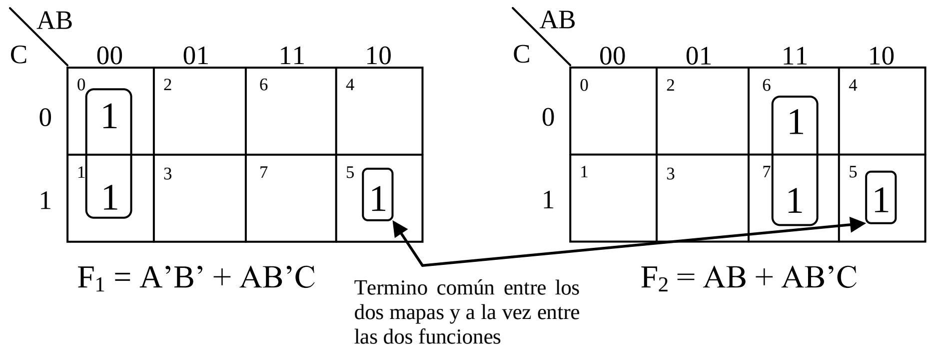


Figura 2.46. Mapas de la función F_1 y F_2 que muestra el termino común AB'C entre ambas.

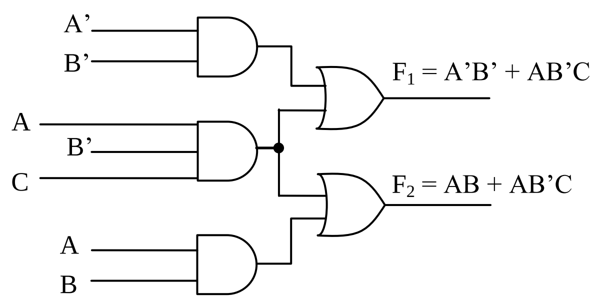


Figura 2.47. Diagrama del circuito donde se unifican F_1 y F_2.

Ejemplo 2.27. Simplificar en un mismo tiempo la función $F_1(A,B,C,D) = \sum m(1,5,9,13,15)$ y $F_2\,(A,B,C,D) = \sum m\,(4,6,12,14,15)$

En este ejemplo ya no se trabajan el desarrollo de las funciones por separado para hacer la comparación, si no ya directamente se realizan los mapas para buscar aquel o aquellos términos agrupados que puedan existir en común entre dichos mapas.

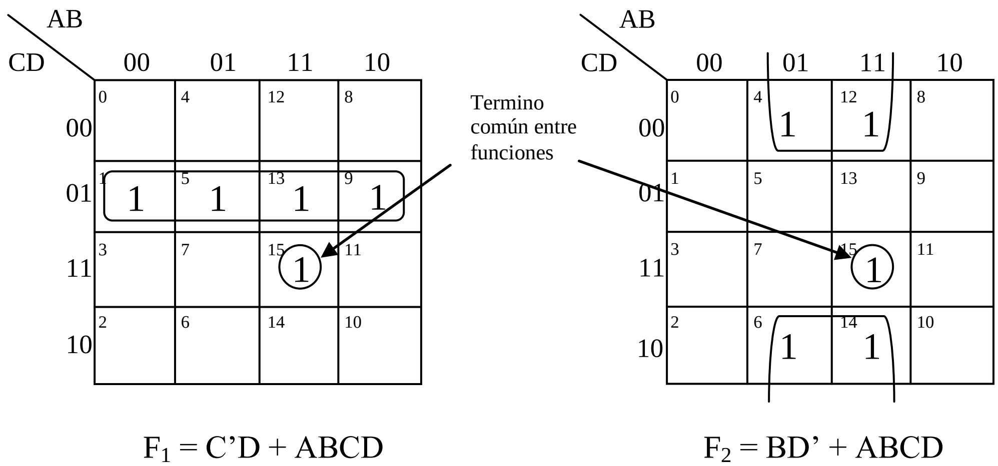


Figura 2.48. Mapas de las funciones F_1 y F_2 que muestran el termino común ABCD

Del análisis de ambos mapas se obtiene una simplificación para F_1 y F_2 en donde para F_1 el término C'D es esencial, mientras que el término ABCD es común a F_2 y de manera inversa F_2 tiene como termino esencial a BD' y al termino ABCD en común con F_1 como se muestra y señala en la figura 2.48.

Llevando a cabo el diagrama del circuito resultante donde conviven ambas funciones nos quedaría como a continuación muestra la figura 2.49.

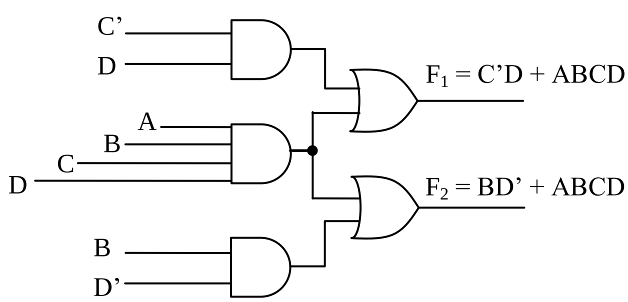


Figura 2.49. Diagrama del circuito que incluye a $F_1(A,B,C,D)$ = C'D + ABCD y F_2 (A,B,C,D) = BD' + ABCD.

De esta manera se concluye el proceso y queda al lector la simplificación de ambas funciones F_1 y F_2 por separado para que observe la reducción del uso de compuertas cuando estas se manejan de manera individual.

2.7.2. Salida a través de Quine – Mc.Cluskey

Con frecuencia es necesario implantar mas de una función de salida con algún conjunto dado de variables de entrada, y así obtener un diseño mas sencillo y de menor precio. Esto también se puede lograr a través de las técnicas de Quine – Mc. Cluskey, pero debemos tomar en cuenta las siguientes consideraciones.

Consideración 1.- A cada minitérmino se le asocia una marca o señal para identificar dentro de cual o cuales funciones se encuentra incluido.

Consideración 2.- Podemos combinar dos términos o también llamados minitérminos, solo si ambos tienen una o más señales en común y el término resultante de las combinaciones solo tiene señales comunes a ambos minitérminos.

Consideración 3.- Podemos eliminar cada término en la tabla de minimización solo si todas las señales de este aparecen en el término resultante de la combinación.

Por lo tanto una vez conociendo estas determinaciones podremos comenzar con la utilización de este método para el diseño de circuitos con varias funciones de salida.

Ejemplo 2.28. Obtener la simplificación mínima de las funciones:

$F_\alpha(A, B, C, D) = \sum m(0,2,7,10) + d(12,15)$

$F_\beta(A, B, C, D) = \sum m(2,4,5) + d(6,7,8,10)$

$F_\gamma(A, B, C, D) = \sum m(2,7,8) + d(0,5,13)$

PRIMERA SECCION							SEGUNDA SECCION							TERCERA SECCION						
MINITER.	A	B	C	D	Marca		MINITER.	A	B	C	D	Marca		MINITER.	A	B	C	D	Marca	
0	0	0	0	0	αγ	√	0,2	0	0	-	0	αγ	PI_2	(4,5-6,7)	0	1	-	-	β	PI_1
2	0	0	1	0	αβγ	PI_{10}	0,8	-	0	0	0	γ	PI_3	(4,6-5,7)	0	1	-	-	β	
4	0	1	0	0	β	√	2,6	0	-	1	0	β	PI_4							
8	1	0	0	0	βγ	PI_{11}	2,10	-	0	1	0	αβ	PI_5							
5	0	1	0	1	βγ	√	4,5	0	1	0	-	β	√							
6	0	1	1	0	β	√	4,6	0	1	-	0	β	√							
10	1	0	1	0	αβ	√	8,10	1	0	-	0	β	PI_6							
12	1	1	0	0	α	PI_{12}	5,7	0	1	-	1	βγ	PI_7							
7	0	1	1	1	αβγ	PI_{13}	5,13	-	1	0	1	γ	PI_8							
13	1	1	0	1	γ	√	6,7	0	1	1	-	β	√							
15	1	1	1	1	α	√	7,15	-	1	1	1	α	PI_9							

Tabla 2.23. Primeros implicados de las funciones F_α , F_β y F_γ .

De la tabla 2.23 vale la pena hacer los siguientes señalamientos:

La combinación de los minitérminos se daba de un grupo de estos a otro, cuando se encontraba la diferencia de un “1” a un “0” o viceversa, y de hecho sigue igual, pero además de esto también deberá haber al menos en común alguna marca (α, β, γ). Por ejemplo; las combinaciones que se pudieran dar del minitérmino 0 a los minitérminos 2, 4, y 8 serian (0,2), (0,4) y (0,8), ya que solo hay un bit de diferencia, pero como se puede observar en la sección 2 no aparece la combinación (0,4) y la razón es que estos no se pueden combinar por que mientras el minitérmino 0 tiene las marcas o señales “α γ” el minitérmino 4 tiene solo el “β”, de aquí que no aparezcan para combinarse de nuevo.

Ahora también es importante señalar que una vez utilizados los minitérminos en alguna combinación estos son marcados con una “√”. Pero vemos el caso del minitérmino 8 que no se encuentra marcado con “√”, una vez que si observamos marcado el 0 que fue un minitérmino con el que se combino, mientras que este no, por lo que se aclara que el minitérmino 0 es marcado por la combinación (0,2) ya que las señales del minitérmino “α γ” y el resultado en marcas de la combinación (0,2) es precisamente también “α γ” mientras que el minitérmino 2 tiene “α β γ” por lo que solo es marcado el 0 y no el 2, o sea que las señales del minitérmino 0 están contenidas en la combinación (0,2). De forma similar sucede con el minitérmino 8, el cual tiene como marcas “β γ” y estas no están incluidas (ambas) en la combinación (0,8). Por otro lado como ejemplo comentamos que el minitérmino 4 es marcado con “√” por que su marca “β” se encuentra contenida en las señales resultantes de la combinación (4,5) y (4.6) que precisan ser “β”.

Posteriormente se procede como se hace normalmente en Quine – Mc. Cluskey a realizar la tabla de los primeros implicados esenciales con la única diferencia que los primeros implicados se colocan para cada función con su marca (según sea el caso α, β, γ o sus combinaciones), y recordar que los términos prescindibles no aparecen en la parte superior.

	Fα				Fβ			Fγ		
	√	√		√	√	√	√	√		
	0	2	7	10	2	4	5	2	7	8
PI1 β						x	x			
PI2 α γ	x	x						x		
PI3 γ										x
PI4 β					x					
PI5 α β		x		x	x					
PI6 β										
PI7 β γ							x		x	
PI8 γ										
PI9 α			x							
PI10 α β γ		x			x			x		
PI11 β γ										x
PI12 α										
PI13 α β γ			x						x	

Tabla 2.24. Primeros implicados esenciales de la funciones F_α , F_β y F_γ.

Lo que en la tabla 2.24 se observa es que los minitérminos que se están cubriendo son aquellos para los cuales la marca de cada primer implicado esencial nos indica, por ejemplo para $PI_{10\alpha\beta\gamma}$ (las marcas que lo acompañan son α, β, γ) entonces el minitérmino a cubrir es 2, por lo que se llena el minitérmino 2 en F_α , F_β y F_γ.

También vemos que para PI_6, PI_8 y PI_{12} como no hay términos que cubrir quedan los renglones libres y en la próxima tabla reducida tampoco serán ya contemplados. Recalcando lo que se muestra en la tabla, que los implicantes primos esenciales son PI_1, PI_2 y PI_5.

		Fα	Fγ	
		7	7	8
**	$PI_{3\ \gamma}$			x
	$PI_{7\ \beta\ \gamma}$		x	
	$PI_{9\ \alpha}$	x		
	$PI_{11\ \beta\ \gamma}$			x
**	$PI_{13\ \alpha\ \beta\ \gamma}$	x	x	

Tabla 2.25. Reducción de los primeros implicados esenciales.

Ya para elegir los últimos implicados tenemos que PI_{13} cubre a dos términos y ya solo se esta entre elegir a PI_3 o PI_{11} que cubren al ultimo termino, solo que nos inclinaremos por elegir P_{I3} bajo el criterio de que tiene menos literales.

Por tanto los resultados mínimos para las tres funciones son:

Para $F_\alpha = PI_2 + PI_5 + PI_{13}$ = A'B'D' + B'CD' + A'BCD

Para $F_\beta = PI_1 + PI_5$ = A'B + B' CD'

Para $F_{\gamma.} = PI_2 + PI_3 + PI_{13}$ = A'B'D' + B'C'D' + A'BCD

Los PI's que tiene cada función son los que se encuentran o cruzan en cada función, esto lo podemos observar de las tablas 2.24 y 2.25.

A continuación se dibuja el circuito final que contiene a las tres funciones, mostrando como PI_2, PI_5 y PI_{13} son compartidos en una compuerta O, para generar a la función Fa.

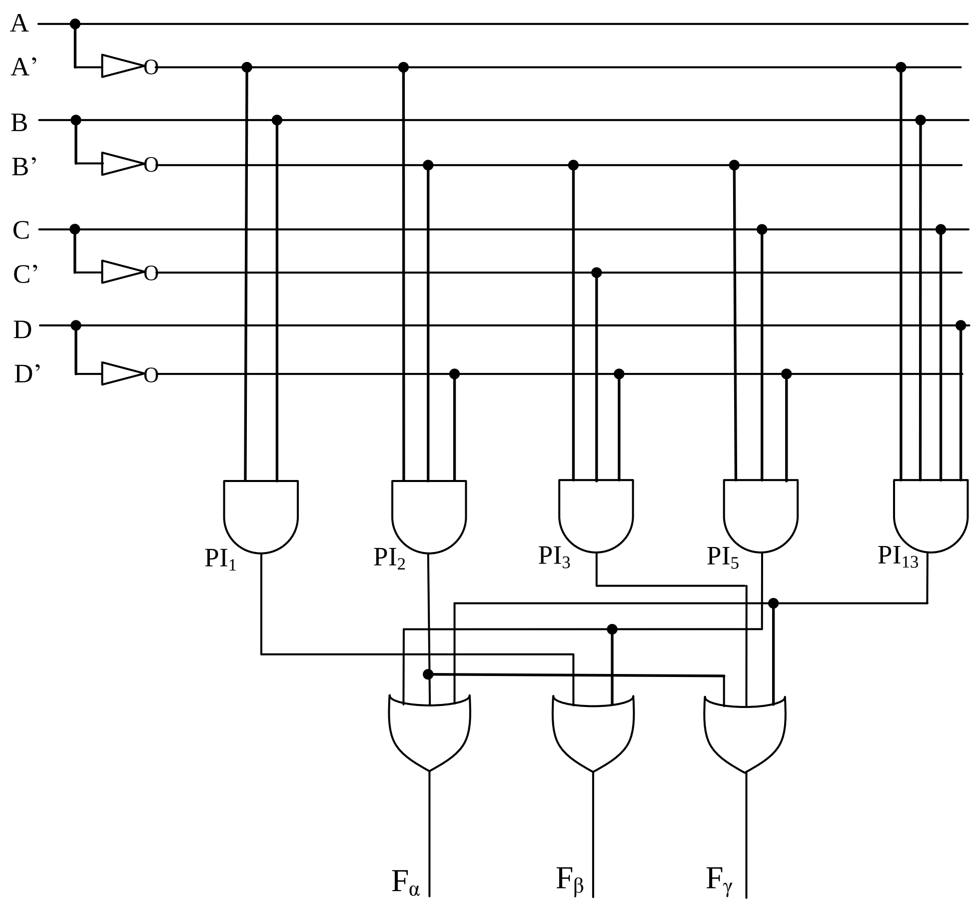


Figura 2.50. Diagrama del circuito que representa las funciones mínimas de F_{α}, F_{β} y F_{γ}.

2.8 Algoritmos

Llegamos a un punto en que nos damos cuenta que las técnicas anteriores son buenas, pero como ya lo habíamos comentado tienen cierto aspecto heurístico, esto quiere decir que nos sirven como una ayuda en el aprendizaje de cómo realizar una simplificación o reducción de un circuito, esto generalmente llevado por el método de prueba y error, y que por tanto no garantizan la obtención de una solución optima. Reconociendo la gran ayuda por ejemplo de la técnica de Quine-Mc.Cluskey, que si muy cierto ya es un algoritmo, sabemos que deja mucho al talento del diseñador para identificar un conjunto primero de implicantes mínimos para completar una cubierta y posteriormente de estos primeros implicados obtener unos segundos llamados esenciales y normalmente la habilidad del diseñador es la de observar cuidadosamente y el no cometer errores en su prueba.

Por otra parte, se han creado algunos otros métodos que pueden generar soluciones optimas de manera directa, y deseando realizar la simplificación de esta manera, nada mejor que los algoritmos.

Los algoritmos son una serie de listas bien definidas, ordenadas y finitas de operaciones que permiten hallar la solución de un problema. Por lo que dado un estudio inicial y una entrada a través de pasos sucesivos y bien definidos se llega a un estado final adecuado, obteniendo así una solución.

La importancia de los algoritmos es que pueden ayudarnos a resolver una gran variedad de problemas, en distintas áreas de la investigación humana, aunque su aplicación es mas en el área de las matemáticas, en mostrar la manera de llevar a cabo procesos y de resolver mecánicamente estos. Esto ayuda a no depender tanto de una habilidad especial por parte del diseñador, referido a lo que ya habíamos comentado con anterioridad.

2.8.1 Búsqueda dirigida.

El algoritmo de búsqueda dirigida contrario a lo que se realiza en el método de Quine – Mc.Cluskey, en donde se realiza una búsqueda de minitérminos esenciales directamente a través de encontrar adyacencias e ir eliminando aquellos minitérminos que podían ser cubiertos por otros, y ya posteriormente hacer un repaso mas en una última tabla para encontrar esos términos esenciales o definitivos. En el algoritmo de búsqueda dirigida se ahorra tiempo ya que se va directamente a la localización de estos términos que forman una cubierta mínima del resto de los términos de la función.

Se entiende que, si muy cierto el método del algoritmo de búsqueda dirigida hace una búsqueda directa de los minitérminos esenciales, también este esta basado en un enfoque de Quine – Mc.Cluskey en el que a través de las vecindades o adyacencias que existen entre un minitérmino y otro ayudan a realizar la simplificación, teniendo el bit de diferencia que existe entre estos.

Empezaremos por considerar que primero se tiene un conjunto inicial de minitérminos, contra los cuales se comparan todos los minitérminos adyacentes, con la finalidad de determinar la cobertura mas grande del minitérmino inicial. Así de esta forma se genera una estructura de árbol que termina con el minitérmino mas grande que cubre al o los minitérminos dados.

Hay que tomar las consideraciones necesarias para que los minitérminos que solo difieran en una variable o solo en un bit sean los que se puedan combinar entre si (se pudiera decir que tiene distancia uno). Ahora dos minitérminos tienen un punto de unión o adyacencia, cuando difiera en una sola variable, por lo que un concepto importante que obtenemos, es que ese punto de unión o adyacencia en cierta variable común a ambos términos es llamada “Dirección de Adyacencia” (DA), que se definiría como el conjunto de enteros con signo que describen como ir de un minitérmino a otro adyacente a este mismo o de adyacencia entre ellos.

El factor de ponderación de un bit o variable en la k – ésima posición a partir del bit menos significativo (LSB) es; 2^k, comenzando por k=0. Si el minitérmino a analizar, tiene un 1 en su k-ésima posición, es decir, que el minitérmino es m_1, y debe sustraerse el factor de peso, que para el caso sería el 1, quedándose con un cero en dicha posición. En la situación de que por ejemplo hablásemos del minitérmino m_0, tiene obviamente un cero en la k – ésima posición, así que ha este se le sumaria su factor de ponderación o séase un cero quedando un 1 en su posición. Así por lo tanto para el resto de los minitérminos su dirección de adyacencia estará definida por la secuencia de números $\{a^k\} = \{\pm 2k \mid k= 0, 1, 2, 3,\ldots , n – 1\}$

Para un mejor entendimiento de lo anteriormente descrito observe la tabla 2.26 en donde se encuentran las variables A, B, C, y D, para funciones de esta misma cantidad de variables, que pueden llegar a generar hasta un 4to nivel de combinación. También en esta misma tabla se encuentran los minitérminos que se obtienen de estas 4 variables, que vendrían siendo desde m_0 hasta m_{15} (los 16 minitérminos). También en otra columna sus variables eliminadas, sus correspondientes 0´s y 1´s, asi como su dirección de adyacencia.

Para nosotros ya son familiares los minitérminos y sus correspondientes 1´s y 0´s que los forman, asi que analizaremos como se obtienen las dos columnas restantes (variable eliminada y dirección de adyacencia.

Por ejemplo si el minitérmino 5 que es 0101 tiene una DA 8, –4, 2, –1, ya que como es 5 su posición, se debe restar esta en su DA, por eso en esta aparece –4 y –1, quedan un –5 que al restarlo de su posición 5 da como resultado 0, el resto de los enteros (8 y 2) son positivos. Ya sumando o restando según marque la DA completa (8, –4, 2, –1) al número 5, seria empezar primero por 5 – 1 = 4, siendo que, es el numero vemos en dirección de la variable eliminada “D”, y lo que sucede es que m_5 tiene adyacencia con m_4 en su variable “D” Ya que $m_5 = 0101$ y $m_4 = 0100$, por lo tanto hay una relación de adyacencia entre estos dos minitérminos. En seguida se realizaría la operación $5 + 2 = 7$ este número está colocado en la siguiente posición queriendo decir que la variable eliminada es “C”, observando nuevamente que m_5 tiene adyacencia o diferencia con m_7 en una variable, la cual es “C”, por que $m_5 = 0101$ y $m_7 = 0111$. Siguiendo en las operaciones seria $5 – 4 = 1$, o sea adyacencia con m_1 y variable eliminada en la posición “B”, ya que $m_5 = 0101$ y $m_1 = 0001$. Por ultimo $5 + 8 = 13$, y el 13 se coloca en la posición de “A” que es la variable que se elimina entre m_5 y m_{13}, ya que $m_5 = 0101$ y $m_{13} = 1101$.

Así de la tabla 2.26 podríamos ir analizando los números que se encuentran en la columna de variables eliminadas. Por ejemplo para m_0 seria 8, 4, 2, 1 es decir que m_0 se puede combinar con m_8 para eliminar la variable “A”, ya que esa es la posición del numero, luego en la posición de la variable “B” esta el 4, dice entonces que m_0 se combina con m_4 y la variable a eliminar es “B”, asi en la posición de “C” esta 2, es decir que m_0 se combina con m_2 y se eliminaría “C”, por ultimo en “D” esta 1 por que sería combinación m_0 y m_1, eliminándose esta variable. De esa forma similar observamos la relación que guardan el resto de los minitérminos con sus respectivas adyacencias.

Otro punto importante a tratar es que al combinarse dos minitérminos con adyacencia estos producen una primera combinación con $n – 1$ variables, si se combinan 4 minitérminos producen una combinación con $n – 2$ variables y si se combinan 8 minitérminos produce una combinación con $n – 3$ variables y así sucesivamente (algo parecido a lo que sucede con Karnaugh). Por lo que una combinación de mayor numero de minitérminos cubrirá a los minitérminos inicialmente planteados.

Los números con signo que indican la DA de un minitérmino con respecto de otro con el cual vaya a combinarse, recibe el nombre de “Dirección de Adyacencia

Requerida” (DAR). Esta DAR es la que nos ayudara a seguir un proceso de las combinaciones que se fueron dando y que pueden representarse por medio de un mapa o diagrama de árbol, en cuya raíz se encuentra uno de los minitérminos dados y cuyas hojas o bifurcaciones del diagrama son las combinaciones posteriores que lo cubren. Este modelo de diagrama de árbol se presenta en la figura 2.51.

ABCD	MINITERMINO	VARIABLE QUE SE ELIMINA				DIRECCION DE ADYACENCIA			
		A	B	C	D				
0000	0	8	4	2	1	+8	+4	+2	+1
0001	1	9	5	3	0	+8	+4	+2	-1
0010	2	10	6	0	3	+8	+4	-2	+1
0011	3	11	7	1	2	+8	+4	-2	-1
0100	4	12	0	6	5	+8	-4	+2	+1
0101	5	13	1	7	4	+8	-4	+2	-1
0110	6	14	2	4	7	+8	-4	-2	+1
0111	7	15	3	5	6	+8	-4	-2	-1
1000	8	0	12	10	9	-8	+4	+2	+1
1001	9	1	13	11	8	-8	+4	+2	-1
1010	10	2	14	8	11	-8	+4	-2	+1
1011	11	3	15	9	10	-8	+4	-2	-1
1100	12	4	8	14	13	-8	-4	+2	+1
1101	13	5	9	15	12	-8	-4	+2	-1
1110	14	6	10	12	15	-8	-4	-2	+1
1111	15	7	11	13	14	-8	-4	-2	-1

Tabla 2.26. Direcciones de adyacencia y relación que guardan las variables de los minitérminos para eliminarse.

El diagrama de árbol de la figura 2.51 muestra las distintas combinaciones de los minitérminos, mandatados por la Dirección de Adyacencia Requerida. Estas combinaciones son por ejemplo para m_0 o 0, combinaciones por supuesto que cubren a este minitérmino. De esta forma vemos como en principio la raíz se divide en las cuatro DAR (+1, +2, +4, +8) en donde al sumarse con el 0 de cada una de ellas se obtienen las primeras 4 combinaciones que son (0,1), (0,2), (0,4), (0,8).

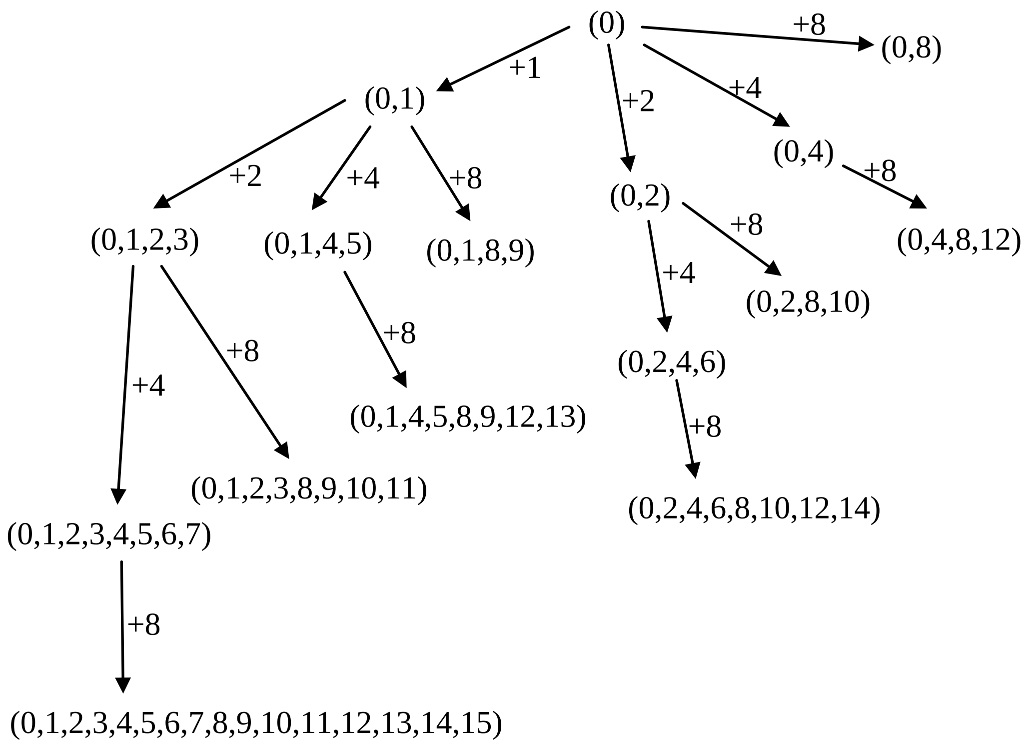


Figura 2.51. Árbol DAR para m_0.

También se observa como el lado derecho del árbol queda truncado, ya que después de la DAR +8 no existe una DAR mas con la cual sumar y crear una nueva combinación, por lo que ahí termina. Mientras que de la combinación (0,4) obtenida por la DAR +4 todavía hay una DAR que se puede aplicar, que seria la siguiente, o sea una DAR +8, por lo que esta logra una combinación mas, que es la suma del 0 + 8 = 8 y 4 + 8 = 12, y así queda la combinación nueva formada por la anterior y la suma con la DAR, es decir; (0,4,8,12). Ahora de la combinación (0,2) obtenida por la DAR +2, se observa como esta genera la bifurcación de DAR +4 y DAR +8 por que puede crecer hacia esas direcciones y al sumar +4 a (0,2) seria; 0 + 4 = 4 y 2 + 4 = 6 quedando la combinación de la anterior con la DAR como (0,2,4,6) y al sumarse +8 a (0,2), seria; 0 + 8 = 8 y 2 + 8 = 10, quedando (0,2,8,10), e igual que en la ramificación anterior esta quedaría truncada ahí, mientras tanto que la DAR +4 todavía le quedaría sumarse con una DAR +8, siendo ya esta la ultima y así consecutivamente con las siguientes ramificaciones. Por lo que el árbol DAR siempre crece mas hacia el lado izquierdo.

Para comenzar a trabajar con este método de algoritmo de búsqueda dirigida y como se recomienda a veces en los trabajos de mapas de Karnaugh, conviene comenzar con minitérminos que se combinan primero en lazos pequeños y después trabajar con lazos grandes, esto localiza combinaciones pequeñas con rapidez y evita tener combinaciones redundantes. La simplicidad en las combinaciones pequeñas provoca un tiempo computacional también mas corto.

Las reglas para la búsqueda de arboles DAR puede enunciarse de la siguiente manera:

1.- Separar todos los minitérminos en tres conjuntos, el conjunto C{1}, formado por todos los minitérminos iniciales verdaderos, el conjunto C{0}, formado por todos los minitérminos iníciales que son falsos o prescindibles y por último el conjunto C{x}, que consiste en todos los minitérminos sin especificar o que ya no importan.

2.- Desarrollar las DAR de todos los minitérminos iníciales verdaderos, aunque si es conveniente realizar una tabla primero con el total de minitérminos y sus DAR, de acuerdo al numero de variables, para que de ahí se escojan los minitérminos que se van a utilizar y sus direcciones de adyacencia para formar una pequeña tabla con estos minitérminos iníciales.

3.- Iniciar la búsqueda a partir del minitérmino verdadero que tenga el menor número de DAR, ya que este tiene el árbol mas simple. La búsqueda comienza en la parte izquierda del árbol. Si con esto no se agotan las DAR, entonces se pasa a la parte derecha del árbol y se continúa la búsqueda.

4.- Se cambian todos los minitérminos verdaderos que ya están cubiertos del conjunto C{1} al conjunto C{x}, y se reacomoda la lista de acuerdo con el número creciente de DAR en cada unos de los elementos que tenga el C{1}

5.- Se escoge el siguiente minitérmino inicial a desarrollar y se repite el proceso hasta abarcar todos los minitérminos.

Los pasos anteriores son una abstracción concreta de lo que se debe realizar, pero en el ejemplo siguiente daré mas detalles de cómo ir siguiendo el proceso.

Ejemplo 2.29. Utilizando el Algoritmo de Búsqueda Dirigida encontrar la solución minima para la función $F(A,B,C) = \Sigma m(0,1,2,3,4,6)$

Como primer paso recomendado es el reconocer los conjuntos de minitérminos que tenemos, por lo que corresponde a este ejemplo darnos cuenta que solo existe el de los minitérminos iníciales y verdaderos o sea que; C{1}= {0,1,2,3,4,6}.

El segundo paso esta diseñado para que se tengan las menores dudas para colocar en una tabla a todas las DAR de los minitérminos generados por las combinaciones de las tres variables A, B y C; la cual aparece en la tabla 2.27. Se realiza una comparación con la tabla 2.28 donde ya se encuentran solo las DAR de los minitérminos iníciales y sus respectivas DAR.

De la comparación de la tabla 2.27 con la tabla 2.28 nos podemos dar cuenta que en la tabla 2.28 que es la que finalmente vamos a utilizar, no se encuentran los minitérminos 5 y 7, ya que no están dentro de nuestros minitérminos iníciales, también debemos observar que en la columna de “se combina con” existen espacios en blanco, que son los lugares que dejaron el 5 y el 7 por no existir para combinar y que por consecuencia dejan espacios vacios en la columna de las direcciones de adyacencia requerida al no existir estas por la misma inexistencia de una dirección hacia otro minitérmino con quien combinarse o encontrarse en adyacencia.

A	B	C	MINITERMINO	SE COMBINA CON			DAR		
0	0	0	0	4	2	1	+4	+2	+1
0	0	1	1	5	3	0	+4	+2	-1
0	1	0	2	6	0	3	+4	-2	+1
0	1	1	3	7	1	2	+4	-2	-1
1	0	0	4	0	6	5	-4	+2	+1
1	0	1	5	1	7	4	-4	+2	-1
1	1	0	6	2	4	7	-4	-2	+1
1	1	1	7	3	5	6	-4	-2	-1

Tabla 2.27. Direcciones de Adyacencia Requeridas de las 2^{n-1} combinaciones de A, B y C.

A	B	C	MINITERMINO	SE COMBINA CON			DAR		
0	0	0	0	4	2	1	+4	+2	+1
0	0	1	1		3	0		+2	-1
0	1	0	2	6	0	3	+4	-2	+1
0	1	1	3		1	2		-2	-1
1	0	0	4	0	6		-4	+2	
1	1	0	6	2	4		-4	-2	

Tabla 2.28. Direcciones de Adyacencia Requeridas de la función $F(A,B,C) = \Sigma m(0,1,2,3,4,6)$.

Del tercer paso solo seleccionamos los minitérminos que tengan el menor número de DAR, que serian el minitérmino 1, 3, 4 y 6, con solo dos DAR, ya que el 0 y 2 tienen tres, así comenzamos con el árbol DAR del minitérmino 1.

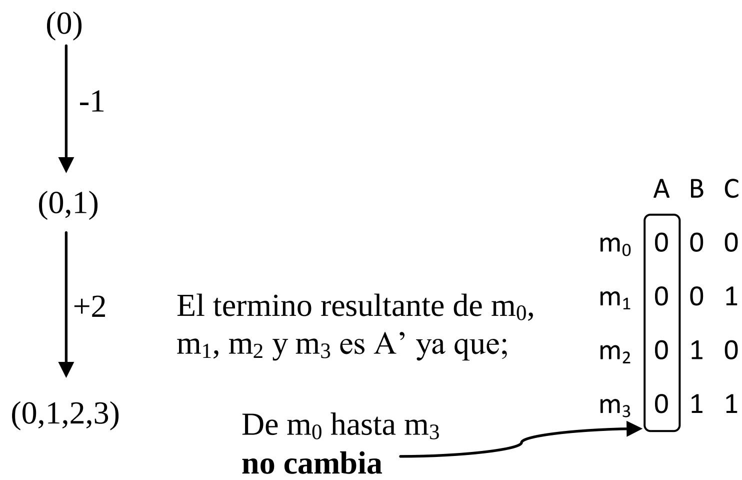


Figura 2.52. Árbol DAR del minitérmino m_1.

En la figura 2.52 observamos el diagrama de árbol que genera el minitérmino 1. En este nos damos cuenta que aparece el m_0 o 0 como principio o raíz, esto debido a que la DAR indica un –1 lo que provoca la operación (1 – 1) = 0, es decir que m_1 se combina con m_0 con Dirección de Adyacencia Requerida de –1 dando un 0 antes que 1 y la combinación es (0,1), cubriendo por obviedad al m_0 y m_1. Luego una DAR +2 que afecta la combinación nueva (0,1), por lo que 0 + 2 = 2 y 1 + 2 = 3, la combinación es (2,3), así de esta forma en la dirección de adyacencia +2 se cubre la combinación primera y la nueva quedando (0,1,2,3), como lo marca la figura 2.39, en la cual también aparece la nota de que la combinación de minitérminos que resulta (0,1,2,3) genera una A’ esencial, ya que en estos minitérminos los ceros y los unos que corresponden a las variables “B” y “C” cambian, mientras que en la variable “A” se mantienen los ceros, así la variable que subsiste es A’.

Lo que a continuación se procede a realizar es a colocar en el conjunto de los términos sin importancia los minitérminos que se obtuvieron y tomando como nuevos al resto de los minitérminos, es decir; C{x} = {0,1,2,3} y C{1} = {4,6}. De esta forma la siguiente combinación en orden numérico para trabajar en forma de árbol DAR es m_4.

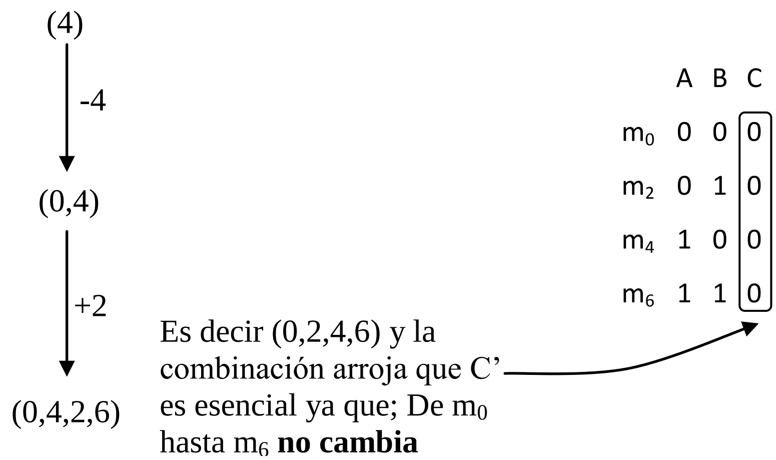


Figura 2.53. Árbol DAR de m_4.

De la figura 2.52 y 2.53 observamos que todos los términos quedan cubiertos con la simplificación, teniendo por tanto que F(A,B,C) = A' + C' y termina el proceso.

2.8.2 Iterativo.

El método iterativo o algoritmo de consenso iterativo, es un método que nos ayuda a simplificar funciones que contienen a un alto número de variables. Ya que según tenga de variables una función, también tendrá de minitérminos la misma, y todo en una proporción de que en "n-variables" estas procrean 2^n–miniterminos. Así que entre más variables, mas minitérminos, y podemos realizar una pequeña demostración de lo que se dice; si suponemos que para una función que contenga tan solo 8 variables, existirán 2^8 = 256 minitérminos, o si se pensara en un número mayor de variables, como pudieran ser 16, entonces serian 2^{16} = 65536 minitérminos, ya ni hablar de 20 o más variables lo que estas generarían para una función cualquiera, volviéndose más complejo aun.

Por lo anterior, el método de algoritmo de búsqueda dirigida generaría una serie de tablas de gran tamaño por los listados de variables que se incrementarían.

El método del algoritmo de consenso iterativo esta concentrado mas a un manejo de los términos producto y no tanto a la cantidad de variables, porque si bien cierto, cuando el numero de variables aumenta, consigo aumenta el número de minitérminos, este es independiente del numero de los términos producto, ya que estos no necesariamente aumentan cuando las variables lo hacen dentro de la función.

El método comienza con el conocimiento de una función inicial F, en la que existan los términos producto cualesquiera, que pueden ir desde "0" hasta "n"; siendo n el numero de variables.

Este algoritmo tiene como herramientas iterativas a dos teoremas; el teorema del consenso, que seria AB + A'C = AB + A'C + BC y el teorema de absorción que es, A + AB = A, que se aplican a la materia prima, que en este caso serian los términos producto. Y lo que buscan estos teoremas es encontrar los primeros implicados.

Este método es el mas sencillo de entender ya que su justificación esta en el comparar y buscar directamente entre los términos iníciales la función, sin hacer ya mas, por lo que es mas veloz, sabiendo que al comparar dos minitérminos con diferencia de una variable, esta se anula, y el nuevo termino que resulto de la combinación cubre a los dos minitérminos. Este término por supuesto es un minitérmino n – 1, o sea con una variable menos, y este minitérmino a su vez se puede combinar con otro minitérmino que este en sus mismas condiciones o sea que tenga n – 1 variables, y si hay claro una solo variable diferente entre ellos, se generara un nuevo termino con n – 2 variables y así sucesivamente, esto quiere decir que una vez que se obtiene un nuevo termino este se consensa de nuevo con todos los términos de la función para ver con cuales se puede combinar, y esto se repite una y otra vez según sea necesario, así entonces se realiza un consenso reiterativo o mas bien un consenso iterativo.

Para ilustrar el método observemos el ejemplo 2.30.

Ejemplo 2.30. Simplificar la función F(A, B, C, D) = Σm(4,5,10,11,13)

La función se puede representar también como:

F(A, B, C, D) = A'BC'D' + A'BC'D + AB'CD' + AB'CD + ABC'D

Se reúnen términos producto que difieren uno de otro en solo una variable y por teorema de consenso quedaría la comparación de la siguiente manera:

$$\left.\begin{array}{l} A'BC'D' \\ A'BC'D \end{array}\right\} A'BC' \quad \text{Nuevo termino generado por consenso}$$

$$\left.\begin{array}{l} AB'CD' \\ AB'CD \end{array}\right\} AB'C \quad \text{Nuevo termino generado por consenso}$$

Los nuevos términos generados por consenso se anexan a la función original quedando:

F(A, B, C, D) = A'BC'D' + A'BC'D + AB'CD' + AB'CD + ABC'D + A'BC' + AB'C

Luego se aplica el teorema de absorción en el que precisamente el nuevo termino generado por consenso absorbe a los términos producto que lo generaron, es decir:

$$\left.\begin{array}{l} A'BC'D' \\ A'BC'D \end{array}\right\} \text{Los dos términos son absorbidos por } A'BC'$$

$$\left.\begin{array}{l} AB'CD' \\ AB'CD \end{array}\right\} \text{ Los dos términos son absorbidos por } AB'C$$

La función queda ya simplificada como:

$$F(A, B, C, D) = ABC'D + A'BC' + AB'C$$

Termina así el proceso del método iterativo, por lo que a continuación se muestra a otro ejemplo para el mejor entendimiento de la simplificación.

Ejemplo 2.31. Simplificar por el método de algoritmo iterativo a su mínima expresión la función $F(A, B, C, D) = \Sigma m(2,4,10,11,12,13)$

Debido a que la función es una suma de productos, de vería como:

$$F(A, B, C, D) = A'B'CD' + A'BC'D' + AB'CD' + AB'CD + ABC'D' + ABC'D$$

Al aplicar el primer paso que sería la intervención del teorema del consenso tenemos:

$$\left.\begin{array}{l} A'B'CD' \\ AB'CD' \end{array}\right\} B'CD' \text{ Nuevo termino generado por consenso}$$

$$\left.\begin{array}{l} AB'CD' \\ AB'CD \end{array}\right\} AB'C \text{ Nuevo termino generado por consenso}$$

$$\left.\begin{array}{l} A'BC'D' \\ ABC'D' \end{array}\right\} BC'D' \text{ Nuevo termino generado por consenso}$$

$$\left.\begin{array}{l} ABC'D \\ ABC'D' \end{array}\right\} ABC' \text{ Nuevo termino generado por consenso}$$

Se reescribe la función anexando los nuevos términos generados, la cual es.

F(A, B, C, D) = A’B’CD’ + A’BC’D’ + AB’CD’ + AB’CD + ABC’D’ + ABC’D + B’CD’ + AB’C + BC’D’ + ABC’

En la comparación por el teorema de absorción quedarían los siguientes términos producto que se generaron:

$$\left.\begin{array}{l} A'B'CD' \\ AB'CD' \end{array}\right\} \text{Estos términos son absorbidos por } B'CD'$$

$$\left.\begin{array}{l} A'BC'D' \\ ABC'D' \end{array}\right\} \text{Estos términos son absorbidos por } BC'D'$$

$$AB'CD \; \} \; \text{Este término es absorbido por } AB'C$$

$$AB'CD \; \} \; \text{Este término es absorbido por } ABC'$$

Finalmente, la solución mínima de la función F es:

F(A, B, C, D) = AB’C + ABC’ + BC’D’ + B’CD’

2.8.3 Iterativo generalizado.

El método del algoritmo de consenso iterativo generalizado es un método mejorado y originado del algoritmo de consenso iterativo, bastante más rápido porque requiere menos iteraciones que el algoritmo de consenso iterativo. El procedimiento de este algoritmo se enuncia a continuación:

1.- Eliminar todos los términos redundantes utilizando por ello el Teorema de Absorción.

2.- Los términos de la función se dividen en tres conjuntos llamados: c{1}, c{0} y c{x}, de acuerdo con el tipo de variable que el termino de la función contiene; ya sea 1, 0 o sin importancia, y para entender un poco mejor a que se refieren estos conjuntos, se muestra una pequeña definición de estas.

c{1} = Conjunto de términos en el cual la variable a analizar vale 1.

c{0} = Conjunto de términos en el cual la variable a analizar vale 0.

c{x} = Conjunto de términos en el cual la variable a analizar no se encuentra y por lo tanto es no importa ya que no es 0 o 1.

3.- Se examina cada término del conjunto c{1} con los del conjunto c{0} en lo que respecta a términos de consenso. Si existe un termino de consenso entonces este se agrega a un conjunto nuevo llamado c{c}, formado por términos de consenso. Es en este paso en el que hay que estar atentos a observar los términos a los cuales se les pueda aplicar el consenso, o de hecho que aparezcan términos iguales que puedan sustituir a términos dentro de la función.

4.- Se compara cada termino de consenso del conjunto c{c} con los otros términos c{c} que haya y los c{x}, eliminando los que son redundantes, como se comentaba en el punto anterior.

5.- Se comparan los demás términos de consenso de c{c} con los c{0}, c{1} y c{x}, eliminando cualquiera de estos últimos que sean cubiertos por un termino de consenso y que sean redundantes, o sea, iguales. Para ello se hace uso del teorema de absorción.

6.- Se combinan los cuatro arreglos y el proceso continua con la siguiente variable

A continuación se desarrolla el método por medio de ejemplos.

Ejemplo 2.32. Reducir la siguiente función aplicando el algoritmo de consenso iterativo generalizado F(A, B, C, D) = A’C’D’ + AB’C’D’ + A’BC + ABCD + ABCD’

Siguiendo el algoritmo y como primer paso se debe checar que haya términos redundantes que eliminar por medio del teorema de absorción (A + AB = A), en el caso de nuestra función no hay ningún termino redundante o que se pueda eliminar de manera inmediata por absorción así que se prosigue.

Los términos de la función se dividen según el tipo de variable y de que esta sea 1, 0 o no importa.

Para la variable A por ejemplo quedaría:

Cuando la variable vale 1 las variables que la acompañan en los distintos términos de la función serian; C{A=1} = {B’C’D’, BCD, BCD’}, mientras que cuando la variable vale 0 los distintos términos que acompañan son; C{A=0} = {C’D’, BC} y notamos que en todos los términos de la función existe la variable A tanto en valores de 1 como de 0, y no existen términos que no contengan la variable, por lo que el conjunto de valor no importa para la variable queda vacio y se representaría como C{A=x} = Ø. Una vez creados estos conjuntos se realiza la comparación entre estos para ver si se generan términos de consenso (teorema de consenso; AB + A’C = AB + A’C + BC), los cuales se puedan agregar a la función y si es posible inmediatamente compararlos con los términos de la función original para ver si este nuevo elemento cubre a otro termino dentro de la función claro que cuando este sea evidente, en caso de que no sea así solo hay que agregarlos. Para el caso que tratamos se generan de comparar C{A=1} con C{A=0} los términos B'C'D' y BCD que se anexan a la función quedando:

F_1(A, B, C, D) = A'C'D' + AB'C'D' + A'BC + ABCD + ABCD' + B'C'D' + BCD

Obsérvese que ahora a la función se le coloco un subíndice 1 para que se vaya diferenciando de la primera función. También como el término nuevo B'C'D' cubre al término AB'C'D' al compararse con el por lo que se elimina este último de la función, lo mismo sucede con BCD que cubre al término ABCD así que este ultimo también se elimina, por lo tanto la función ahora es:

F_2(A, B, C, D) = A'C'D' + A'BC + ABCD' + B'C'D' + BCD

Se continúa con el manejo de las variables ahora para la variable B:

C{B=1} = {A'C, ACD', CD}, C{B=0} = {C'D'} y C{B=x} = {A'C'D'}. De este análisis para la variable B debe hacerse hincapié en que aquí si hay términos no importa o sea que hay un término dentro de la última función que no tenían la variable B y que esta se toma como que podría ser 1 o 0. Y al comparar el C{B=x} con el C{B=0} se origina A'C'D', pero como este ya se encuentra dentro de la función, ya no se volvería a colocar en ella.

Ahora para la variable C:

C{C=1} = {A'B, ABD', BD}, C{C=0} = {A'D',B'D'} y C{C=x} = Ø. También en este caso no existen términos de consenso entre los conjuntos que se formaron, así que pasamos a la siguiente variable.

Para la variable D:

C{D=1} = {BC}, C{D=0} = {A’C’, ABC, B’C’} y C{D=x} = {A’B}. En los conjuntos de esta variable se genera por consenso el término ABC y BC, por lo que se anexan a la función quedando como:

F_3(A, B, C, D) = A’C’D’ + A’BC + ABCD’ + B’C’D’ + BCD + ABC + BC

De esta última función se tiene que los nuevos términos cubren a otros, por ejemplo el termino ABC al compararlo con el término ABCD’ cubre a este ultimo por lo que se elimina, por otra parte el término BC cubre a los términos A’BC, BCD y al nuevo recién generado ABC, quedando finalmente la función reducida a su mínima expresión como:

F_4(A, B, C, D) = A’C’D’ + B’C’D’ + BC

De esta manera termina el proceso del algoritmo.

Ejemplo 2.33. Aplicar el algoritmo iterativo generalizado a la función F(A, B, C, D) = A’B’C’ + A’B’D + A’BC + ABC +A’BC’D + AB’C’D’

Se revisa primeramente que no existan términos redundantes, ya que de ser así tendrían que ser eliminados por teorema de absorción y no es así para esta función.

Como siguiente paso formamos los grupos para la variable A:

C{A=1} = {BC, B’C’D’}, C{A=0} = {B’C’, B’D, BC, BC’D} y C{A=x} = Ø. En este paso se generan términos de consenso como BC, B’C’D’, que se comparan con la función y que ya pueden sustituir a algunos términos como por ejemplo BC puede sustituir a los términos A’BC y ABC, ya que los cubre, mientras que B’C’D’ puede sustituir por absorción a AB’C’D’ y la función quedaría:

F_1(A, B, C, D) = A’B’C’ + A’B’D + A’BC’D + B’C’D’ + BC

Continuamos formando los conjuntos pero ahora de la variable B:

C{B=1} = {C, A’C’D}, C{B=0} = {A’C’, A’D, C’D’ } y C{B=x} = Ø. Se generan los términos A’CD y A’C’D que se anexan a la función para que esta se vea de la forma:

F_2(A, B, C, D) = A’B’C’ + A’B’D + A’BC’D + B’C’D’ + BC + A’CD + A’C’D

De esta función el término nuevo A’C’D cubre al termino A’BC’D y esta misma función queda ya con la modificación, se realizo este paso para que pueda notarse que se deben observar cuidadosamente los términos, para localizar a los que se puedan cubrir con algún termino recién agregado.

F_2(A, B, C, D) = A’B’C’ + A’B’D + B’C’D’ + BC + A’CD + A’C’D

Continuando con la variable C:

C{C=1} = {B, A’D}, C{C=0} = {A’B’, A’D, B’D’} y C{C=x} = {A’B’D}. Se generan los términos A’BD, A’B’D, A’D, solo que A’B’D ya se encuentra en la función y ya no es necesario colocarla. Mientras que A’D sustituiría al termino A’B’D, A’C’D, A’CD y al mismo A’BD generado y que se iba a colocar pero que no se hace por que el A’D lo cubriría. La función es entonces:

F_3(A, B, C, D) = A’B’C’ + A’D + B’C’D’ + BC

Los conjuntos de la variable D:

C{D=1} = {A'}, C{D=0} = {B'C'} y C{D=x} = {A'B'C', BC}. Se genera A'BC pero lo cubriría BC por lo que tampoco se coloca. La función queda igual.

Por último y como lo menciona el algoritmo debemos checar si existen términos redundantes que sean cubiertos por un termino de consenso para eliminarlos por el teorema de absorción, esto después de haber trabajado con cada una de las variables de la función. Como se había comentado por consenso se buscan los términos redundantes de los cuales puede prescindir la función, los demás serán esenciales. En este ejemplo se observa que el termino A'D y B'C'D' generan por consenso el término A'B'C', que ya se encuentra en la función y que por lo tanto es redundante y no se coloca ya mas, queda finalmente:

$F_4(A, B, C, D) = A'D + B'C'D' + BC$

De esta forma queda concluido el capitulo 2, quedando cubiertas las características de algunas de las técnicas de reducción de funciones más importantes, las cuales permiten reducir a los circuitos combinacionales que se tratan en el siguiente capítulo.

CAPITULO 3

DISEÑO DE CIRCUITOS COMBINACIONALES

3.1. Datos del fabricante.

Es indispensable que para el inicio en el diseño de algún o algunos circuitos combinacionales, se tenga conocimiento de algunas características de los CI (circuitos integrados), que contienen dentro de ellos compuertas lógicas, que ayudan a la estructura de cualquier circuito combinacional a llevar a cabo ciertas tareas deseadas. Estas características son conocidas como datos del fabricante.

Dentro de los datos del fabricante encontramos también que estos se clasifican de acuerdo al grado de integración, el tipo de tecnología y por tanto a grupo o familia a la que pertenece, características técnicas y sus códigos de identificación.

Para empezar la clasificación de los CI según el grado de integración es; por la cantidad de puertas lógicas integradas en un chip o pastilla de silicio y reciben los siguientes nombres:

- SSI (Small scale integration); integración de pequeña escala, de 2 a 99 compuertas en un solo chip. Donde se encuentran las puertas lógicas mas comunes y operadores lógicos combinacionales básicos.

- MSI (Medium scale integration); es una integración a mediana escala que va de los 100 a 999 componentes en un solo chip, como los codificadores.

- LSI (Large scale integration); los chips contienen entre 1000 y 9999 componentes. Aquí quedan incluidos circuitos como registros, contadores, etc., así como los llamados ALU (Unidad lógica aritmética) además de memorias y microprocesadores.

- VLSI (Very large scale integration); este único chip contiene una muy gran escala de integración, entre 10000 y 99999 compuertas. Ya entran sistemas lógicos mas complejos como microprocesadores, memorias y microcontroladores, donde solo el chip tiene términos de procesador, de memoria y de entrada/salida de un ordenador.

- ULSI (Ultra large scale integration); al grado de integración se le adjudica el adjetivo de ultra gran escala, donde las compuertas en el chip van de entre 100000 y 999999 elementos. Los procesadores y microprocesadores son muy avanzados.

- GSI (Giga scale Integration); CI de una máxima complejidad, donde los transistores y diodos van mas allá de un millón y que son utilizados en los últimos y mas avanzados procesadores y microprocesadores digitales.

La clasificación de los circuitos integrados que pertenecen a grupos de CI que tienen características comunes y que se agrupan se les llama “familias”. Estas características similares que tienen por tipo de familia pueden ser por ejemplo con respecto al tiempo de propagación, potencia disipada, inmunidad al ruido, etc., características de las cuales hablaremos mas adelante y que para el instante inmediato solo hablaremos de los tipos de familias y algunas características que marcan la diferencia entre una y otra.

Estas familias de CI mas conocidas son las que se enlistan a continuación:

- RTL (Resistor transistor logic); son de los primeros circuitos integrados que se diseñaron, basados en compuertas discretas como las resistencias y transistores. Es de las familias mas antiguas. Su máxima frecuencia de trabajo es de 5Mhz.

- DTL (Diodo Transistor Logic); los arreglos internos de esta familia son los mas sencillos, que es un arreglo Lógica Diodo – Transistor esto provoca que haga tiempos de propagación de 25ns y frecuencia de trabajo de 10Mhz aproximadamente. Pero aun así las cualidades de esta familia son pocas.

- TTL (Transistor Transistor Logic); este grupo de CI es de las familias mas famosas, ya que han impulsado en gran medida el avance en la electrónica digital. Por supuesto su arreglo interno Transistor – Transistor da muchas mas ventajas en su funcionamiento que sus antecesores RTL y DTL. Algunas de sus ventajas son: que trabajan sin problemas a temperaturas de entre 0°c y 70°c. Son compatibles con otras familias como la DTL y subfamilias TTL. Tienen poco retraso en la propagación de la señal, así como buen acoplamiento con determinado numero de compuertas a la salida; frecuencias de trabajo muy veloces, reducción de márgenes de ruido, entre otras cualidades que otras familias no tienen. Contiene dentro de ella mas subfamilias que tienen características particulares, según requiera el diseñador, como lo son:

- TTL Standard
- TTL High speed
- TTL Schottky
- TTL Low power
- TTL Low power Schottky
- TTL Advanced low power Schottky
- TTL Fairchild advanced Schottky

- ECL (Emmitter coupled logic); este otro grupo es importante, ya que se basa en un arreglo de lógica acoplada por emisor. Este acomodo reduce aun mas los tiempos de propagación de la señal. Y la tensión de alimentación de estas compuertas es de –5.2v. Es de alta fabricación ya que logra tiempos de propagación de 1ns por compuerta y frecuencias de trabajo de hasta 400Mhz. Esta familia se subdivide en otras de acuerdo a sus tiempos de propagación como a continuación se enlista:

- ECL de 8ns; frecuencia máxima de funcionamiento de 30Mhz.
- ECL de 4ns; frecuencia máxima de trabajo de 75Mhz.
- ECL de 2ns; frecuencia máxima de funcionamiento de 125Mhz.
- ECL de 1ns; frecuencia máxima de funcionamiento de 400Mhz.

La mas usada dentro de estas subfamilias es la de 2ns ya que su velocidad es buena y optimiza el consumo de potencia. Estas son muy utilizadas en ordenadores y sistemas de comunicación de alta densidad de transmisión.

- CMOS (Complementary metal oxide semiconductor); la característica principal de esta familia es que son CI de bajo consumo de energía. Otras características mas son que las tensiones con las que se alimenta pueden ir desde los 3v hasta los 15v, permitiendo que se pueda acoplar con CI´s de la familia mas importante que es la TTL. Tiene buena inmunidad al ruido, entre el 30% y el 45% del nivel lógico entre los niveles alto y bajo. Esta familia es usada mucho en aparatos alimentados por baterías y de funcionamiento prolongado. Otras subfamilias de los circuitos integrados CMOS son:

- ABT (Semiconductor de tecnología avanzada de doble compuesto metal-oxido)
- AHC/T (Semiconductor metal-oxido de tecnología avanzada de alta velocidad)
- ALVC (Semiconductor metal-oxido avanzado de bajo voltaje)
- ALVT (Semiconductor de tecnología avanzada de doble compuesto metal-oxido de bajo voltaje)
- AUC (Semiconductor metal-oxido avanzado de bajísimo voltaje)
- AVC/M (Semiconductor metal-oxido avanzado de muy bajo voltaje)
- LVC (Semiconductor metal-oxido de bajo voltaje)
- LVT (Semiconductor doble compuesto metal-oxido con tecnología de bajo voltaje)

- I^2L (Integrated injection logic); otra principal familia es esta la I^2L o mas bien IIL (Lógica de inyección integrada). La cual tiene internamente arreglos de transistores bipolares multicolectores. Como características principales es que son también de bajo consumo de energía, de buena respuesta de transmisión de señal y esto aunado a la gran cantidad de compuertas que pueden integrarse dentro de los chips de 6 a 10 veces mas que en un chipcon tecnología TTL. Esto es ideal para equipos pequeños portátiles de bajo consumo de energía.

- HTL (High threshold logic); tienen una alta inmunidad al ruido, así que se emplean con frecuencia en la industria para el control y manejo de dispositivos electromecánicos y en circuitos con tiristores, donde se producen amplias transiciones de tensión.

Un dato mas del fabricante es que estos utilizan un código que va impreso sobre su capsula del CI, con el cual el mismo fabricante identifica de que CI se trata. El pequeño detalle es que cada fabricante pone a sus CI´s sus propios códigos y aunque otro fabricante tenga el mismo circuito integrado en producción, este no usa los mismos códigos, lo que hace complicado la búsqueda de un mismo circuito de distintos fabricantes. Solo comentaremos algunos de los mas importantes fabricantes y que para conocer mas de un circuito integrado, lo ideal es acudir al catalogo de CI´s que maneje el fabricante, o los fabricantes de los cuales se quiere utilizar un integrado. Algunos de los fabricantes son:
AEG TELEFUNKEN (www.telefunken.com)
AMI (www.ami.com)
FAIRCHILD (www.fairchildsemi.com)
FERRANTI (www.ferranti-technologies.co.uk)
INTEL (www.intel.com)
ITT (www.itt.com)
MOTOROLA (www.freescale.com)

MULLARD (www.ucl.ac.uk)
NATIONAL SEMICONDUCTORS (www.national.com)
PHILIPS (www.npx.com)
RAYTHEON (www.raytheon.co.uk)
SESCOSEM (www.st.com)
SGS/ATES (www.st.com)
SIEMENS (www.infineon.com)
SIGNETICS (www.signetics.com)
SOLITRON (www.solitrondevices.com)
TEXAS INSTRUMENTS (www.ti.com)
UNISEM (www.unisemgroup.com)

De las letras y números que como código están escritos sobre el encapsulado, estas quieren decir para cada fabricante sus características particulares, por ejemplo nos indican de que tipo de familia es; DTL, MOS o TTL, y si el circuito es digital, lineal, analógico o analógico digital, si es amplificador, demodulador, oscilador, etc., también algunos números pueden representar rangos de temperatura a los que trabaja o también en letras de que tipo de encapsulado se trata.

De los encapsulados se puede decir un poco mas, o mucho mas, ya que estos son bastante diversos. Pueden ser cerámicos o plásticos. En la figura 3.1 se muestra un esquema que sintetiza los tipos de encapsulados que hay, los mas usados o comunes de ver.

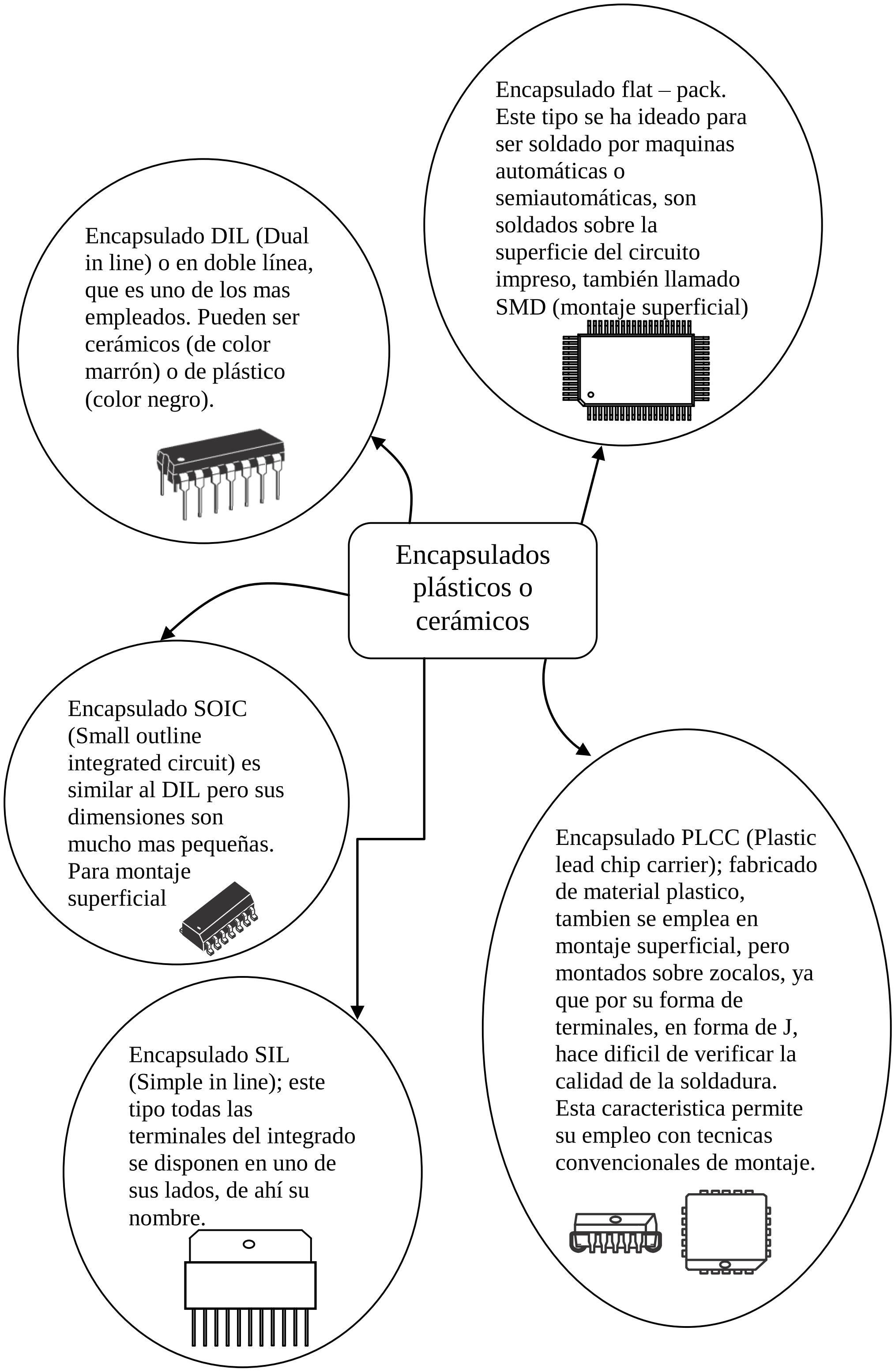


Figura 3.1. Encapsulados comunes en los circuitos integrados.

De los datos importantes de fabricante para saber que condiciones considerar para el diseño de un circuito combinacional son:

- Corriente
- Potencia de Disipación
- Factores de Carga
- Tiempo de Propagación
- Inmunidad al Ruido
- Temperatura de Trabajo
- Factor de Calidad

Corriente.

Una fuente de corriente directa utilizada para alimentar a un circuito integrado es capaz de suministrar y también de recibir corrientes, ya que en los circuitos integrados pueden circular estas en dos sentidos; el positivo y el negativo. La corriente positiva, será la corriente que circula hacia la entrada del integrado, mientras que la corriente negativa será la que circule desde la entrada del integrado y que es llamada también corriente inversa.

Por lo tanto es importante conocer que tipos de corriente existen, con las que se trabajan en las entradas y salidas de un circuito integrado, tanto en el nivel alto (H), como en el nivel bajo (L).

- Corriente de alimentación I_{cc}

Los circuitos contienen pines o patitas, de los cuales, dos son muy importantes y que se indican con las abreviaturas Vcc y GND, como se muestra en la figura 3.2.

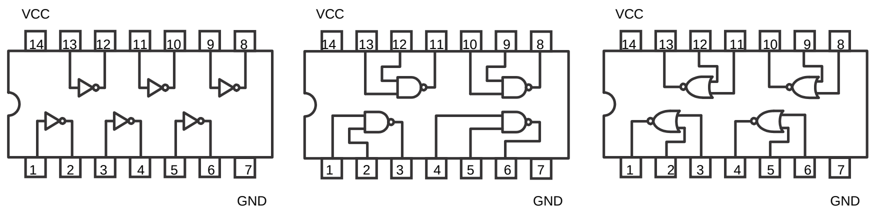


Figura 3.2. Terminales de alimentación en las compuertas.

Donde el pin marcado con Vcc es el que se le aplica un voltaje o tensión positiva proveniente de la fuente de alimentación, mientras que el pin marcado con GND es donde se conecta la llamada masa o negativo de la misma fuente de alimentación. Estos pines no siempre vienen colocados en la misma posición por lo que es indispensable consultar el manual del fabricante, ya que la estructura de un chip a otro puede cambiar aun siendo el mismo fabricante.

Por lo tanto, la I_{cc} es la corriente que circula hacia la terminal Vcc del integrado para alimentarlo. Este dato es importante que el fabricante lo proporcione, ya que con esto podremos permitirnos saber cuan exigente en alimentación son los CI que utilizaremos y la potencia que requerirá nuestro circuito combinacional. Lo que si debemos considerar es que el fabricante nos proporciona este dato con entradas y salidas del CI sin conexión. Pero no solo eso, el fabricante también nos especifica la corriente de alimentación al CI cuando sus salidas (salidas de las compuertas) a nivel lógico bajo I_{CCL} y la corriente de alimentación con las salidas a nivel lógico alto I_{CCH}. Normalmente estos datos se dan para niveles de voltaje nominal y máximo de alimentación Vcc que puede aplicársele al integrado en cuestión y a una temperatura ambiente de 25°C T_{amb}, ya que una variación, ya sea en el voltaje de alimentación como en la temperatura, hace que las corrientes puedan variar, ya sea en un mínimo, o un máximo.

- Corriente de entrada a nivel alto I_{IH}

Este tipo de corriente es la que circula en las entradas es decir, en las entradas de las compuertas que contenga el integrado, cuando se le es aplicado un voltaje a dicha o dichas entradas a nivel alto V_{IH} (especificado por el fabricante), por supuesto corriente positiva ya que circula hacia adentro del circuito.

- Corriente de entrada con nivel bajo I_{IL}

Esta corriente es ahora la que circula también en la entrada de las compuertas pero cuando se le aplica un voltaje en nivel bajo V_{IL} (especificado por el fabricante). Esta seria la corriente llamada negativa, ya que circula de la entrada hacia el pin GND.

- Corriente de entrada con tensión de entrada máxima I_I

En los casos anteriores habíamos hablado de las corrientes en nivel alto y bajo de voltaje en las entradas, especificadas por el fabricante, pero también este nos comenta que existe una corriente que se genera con una tensión o voltaje máximo permitido V_I y el cual todavía permite que trabaje el circuito.

- Corriente de salida con nivel alto I_{OH}

Así mismo como hay corrientes de entrada, también las hay de salida de las compuertas y una de ellas es la I_{OH} que se genera desde una salida cerrada cuando en esta se tiene un voltaje con nivel alto V_{OH}.

- Corriente de salida con nivel bajo I_{OL}

De la misma forma esta la I_{OL}, que es la corriente que se genera hacia una salida de compuerta cerrada cuando esta tiene un voltaje de salida a nivel bajo V_{OL}.

- Corriente de corto circuito de salida I_{OS}

Existe también la corriente que circula en el circuito cuando en una de sus salidas que esta en nivel alto se encuentra cortocircuitada a GND. La cual define el valor de resistencia limitadora de salida. Esto es que si la I_{OS} es muy grande puede deteriorar o destruir el integrado.

Potencia Disipada.

La potencia disipada es la potencia absorbida por el integrado y se abrevia como P_D. Se puede hacer la suma de las potencias disipadas en cada uno de los integrados que conformen un circuito combinacional, constituyen la potencia disipada total del diseño, lo que es importante saber, ya que ello provocara que conozcamos que tipo de potencia necesita el circuito, y por lo tanto, el tipo de fuente de alimentación a usar. Esto se puede calcular con la siguiente relación:

$P_D = (Vcc)(I_{IN})$ mW

Donde; Vcc es la tensión continua de alimentación en Volts , I_{IN} es la cantidad de corriente de entrada en mA.

Es necesario también comentar que la potencia disipada de una compuerta varia de acuerdo si esta trabajando en nivel alto o nivel bajo, por lo que en general los fabricantes dan como dato de potencia disipada un promedio de la potencia disipada de su compuerta cuando esta trabaja a nivel alto, con la potencia disipada cuando esta trabaja a nivel bajo quedando que:

$$P_D = \frac{P_{DH} + P_{DL}}{2}$$

La importancia de la potencia disipada viene en relación directa a los dos puntos siguientes a considerar:

1) Cuando la potencia disipada es menor en un circuito combinacional, este esta consumiendo menos cantidad de energía eléctrica, ya que la corriente eléctrica es menor. Por cuestiones de ahorro es bueno, ya que es importante que un circuito combinacional que tenga un grado mayor de compuertas, no tenga necesidad de grandes fuentes de alimentación.

2) El calor es otro aspecto importante a tomar en cuenta, ya que entre mayor sea la potencia disipada, mayor es la cantidad de calor que se genera, por lo que si el

circuito combinacional tiene una cantidad considerada de compuertas que están generando demasiado calor, se corre el riesgo de que este se destruya, por lo que tendrían que usarse radiadores de calor.

Con respecto a la potencia disipada podríamos concluir que; si esta es menor en un circuito combinacional entonces la calidad de este es mayor, aunque otro factor que se debe tomar en cuenta para la calidad es el tiempo de propagación, del cual se hablara mas adelante.

Factores de Carga.

A este dato de fabricante también se le conoce como capacidad de carga o cargabilidad, la cual es la capacidad de un integrado de que sus compuertas internas, desde sus salidas, puedan controlar cierto numero de otras compuertas lógicas en otro u otros circuitos, de la misma forma el numero de salidas provenientes de otras compuertas lógicas que puedan controlar las propias entradas de las compuertas del integrado.

Es decir, que un circuito integrado no puede controlar una cantidad infinita de otros circuitos integrados, y por lo tanto, sus compuertas, ya que lo que un integrado cede a otros es una cantidad de corriente, para así poderlo controlar, y por supuesto existe un limite para esto, es decir, que pueden conectarse a cierta cantidad de carga. Esta cantidad de carga o factor de carga se define por un numero entero, numero que nos dice la cantidad de circuitos que controla o se pueden controlar. Esta cantidad es la llamada capacidad de carga de salida o cargabilidad de salida, también conocida con el termino en ingles “fan – out” (abanico de salida).

De forma similar un circuito integrado puede formar parte de la carga de otro u otros circuitos, así que la corriente nominal de entrada limita el numero de circuitos integrados que pueden alimentar la entrada de algún circuito. Esta capacidad llamada capacidad de carga de entrada o cargabilidad de entrada, es llamada en ingles como "fan – in" (abanico de entrada)

En la figura 3.3 se observa un ejemplo de lo que serian las compuertas que se pueden conectar a una compuerta AND (que se encuentra encerrada) tanto en su salida como en su entrada; tiene como fan – out de 10, esto quiere decir conectadas 10 compuertas entre inversores y compuertas AND y OR, mientras que de fan – in solo una en cada patita de la compuerta AND, por lo que el fan – in seria de 1.

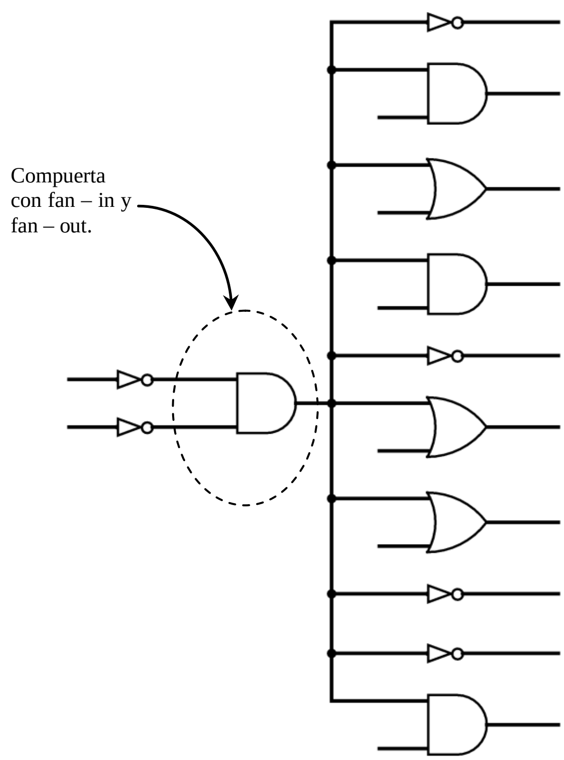


Figura 3.3. Fan – in y fan – out de una compuerta AND.

Normalmente en la práctica del armado de circuitos combinacionales, el fan – in suele ser de 1, aunque en algunos casos se da un fan – in de 2 o 3. Por su parte el fan – out suele variar de entre 6 y 10 para el común de los CI, aunque hay algunos especiales que pueden llegar a un fan – out de 30.

Tiempo de Propagación.

El tiempo que tarda un impulso aplicado a una entrada de una compuerta, en salir o aparecer en la salida de la misma compuerta, es al que llaman tiempo de propagación, o como se le conoce en ingles “Propagation delay time”, que seria en otras palabras la velocidad de funcionamiento de una compuerta o puerta lógica.

La velocidad de propagación es parte de las características mas importantes que se toman en cuenta para decir si se tiene un factor de calidad bueno o malo en la compuerta, ya que cuanto mayor sea su velocidad, mas operaciones aritméticas o lógicas harán en determinado tiempo.

Aproximadamente en los circuitos integrados actuales su tiempo de propagación puede variar de entre 1ns y 100ns.

Entonces, su velocidad de propagación es mayor en cuanto sea menor el tiempo que separa el flanco del impulso de entrada, del flanco del impulso de salida, considerando que se toman en cuenta los valores medios de los impulsos en su amplitud, así como se observa en la figura 3.4. En esta figura se observa como esta marcado con V_M la mitad de la amplitud de los impulsos de entrada y de salida. Tanto en el impulso de entrada que va de un nivel lógico bajo (L) a un nivel lógico alto (H, llamado flanco ascendente de impulso), como al nivel o impulso descendente que va de un nivel lógico alto a un nivel lógico bajo.

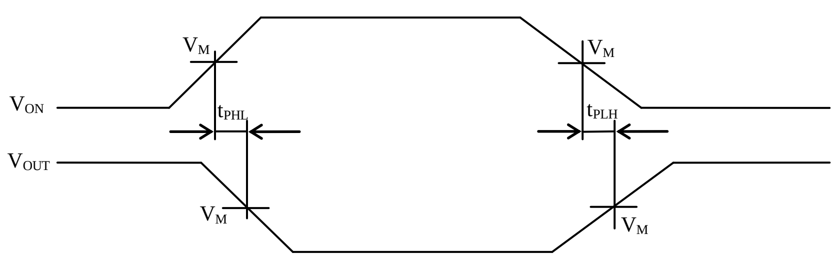


Figura 3.4. Muestra la separación de los impulsos de entrada con respecto a los de salida en los cambios de nivel (tiempo de propagación).

También existe un parámetro llamado tiempo de propagación medio T_{PM}, que seria la media aritmética entre los tiempos medios de propagación del cambio de nivel de la entrada, con el cambio de nivel a la salida.

Observe en la figura 3.4 como la amplitud o tiempo de transición de un nivel a otro en la entrada, es distinto al tiempo de transición de cambio de nivel de la salida, se observa como las inclinaciones son distintas.

Esta característica del tiempo de propagación se relaciona mucho con la potencia disipada, ya que es muy cierto que cuando un circuito combinacional esta diseñado y su velocidad de funcionamiento es mayor, el circuito esta en una forma mas ideal, pero esto contrae que a mayor velocidad existe mas potencia disipada y viceversa, y que exista equilibrio entre estos dos datos hace que el circuito combinatorio sea bueno, y es aquí donde el ingeniero debe considerar los distintos factores y datos del fabricante para realizar el mejor diseño.

Cabe mencionar que cuando armemos físicamente un circuito combinacional y este por ende contenga varias compuertas lógicas, los tiempos de propagación en cada compuerta se acumulan. Por ejemplo si el pulso de entrada tiene que pasar por al menos unas nueve compuertas, todas ellas del mismo tipo y que estas tengan un tiempo de propagación de unos 10ns, entonces hablaremos de un tiempo acumulado de 90ns.

Inmunidad al Ruido.

Con respecto al ruido habrá que definir primero a este, que vendría siendo cualquier tensión parasita superpuesta a la señal útil.

La elevada velocidad de conmutación de los circuitos integrados digitales que provoca señales transitorias puede afectar a la tensión de la fuente de alimentación, interfiriendo en el funcionamiento normal del circuito combinacional que el ingeniero diseñe. Una transición rápida de los niveles lógicos puede crear impulsos de ruido (rebotes) indeseables, que se transmiten a través de la capacidad distribuida entre las terminales de interconexión. Estas señales transitorias, o rebotes, pueden ser interpretadas como señales validas por las puertas lógicas, provocando niveles de salida erróneos y, como consecuencia, un falso funcionamiento del circuito.

Los circuitos combinacionales se deben diseñar con cierta inmunidad al ruido y, por consiguiente, que ignoren una parte de el, aunque no sea en su totalidad.

Por lo tanto la inmunidad al ruido seria la cantidad de ruido que puede superponerse a una señal lógica aplicada a una compuerta de un circuito integrado digital, sin que dicha compuerta cambien de estado incorrectamente. La magnitud de inmunidad de ruido se puede medir en volts o mili Volts.

Ahora existe cierto margen de tolerancia al ruido por parte de un integrado. La diferencia entre el nivel de tensión lógico transmitido y el aceptado como entrada valida es al que se le llama *margen de ruido* (noise margin).

Debemos además de tomar en cuenta que los niveles lógicos de los cuales hemos venido hablando, alto (H) y bajo (L), no son voltajes de valores fijos, sino que están expresados por una banda de voltaje o tensión.

Estas bandas de voltaje son distintas en las entradas y salidas de las compuertas, y también son distintas si se trata de algunos fabricados con tecnología bipolar o unipolar.

Haciendo referencia a las entradas de las puertas lógicas, cualquier tensión por debajo de +0.8v es interpretada como nivel lógico bajo (L o 0 binario) en cualquier tensión por encima de +2v pero inferior a la tensión de alimentación (es este caso +5v), es interpretado como nivel lógico alto (H o 1 binario). La banda de tensiones comprendida entre +0.8v y +2v es una banda prohibida, ya que en esa zona la puerta lógica no es capaz de interpretar el valor lógico de la señal de entrada, observe la figura 3.5.

Cualquier ruido parasito (margen de ruido), sobre una tensión de salida de +0.4v (nivel lógico L) es ignorado en la entrada del circuito a ella conectada, siempre y cuando sea inferior a +0.4v, ya que si el ruido supera este valor su adición a los 0.4v del nivel bajo de la salida superaría los +0.8v que exige como máximo la entrada para considerarlo nivel bajo. Por lo tanto, estos ruidos parásitos no perturbarán el funcionamiento del circuito, si se mantienen en ese nivel.

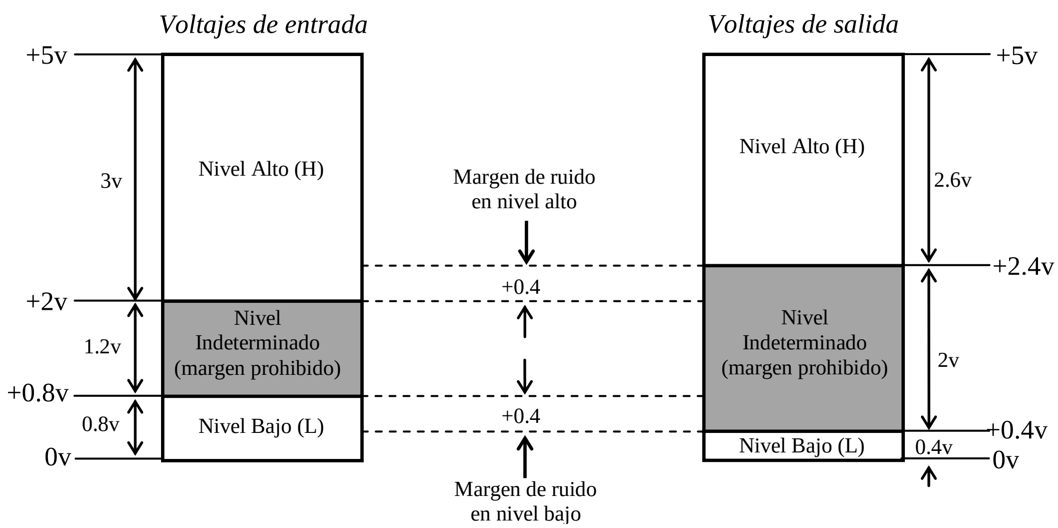


Figura 3.5. Niveles lógicos de entrada y salida y márgenes de ruido en un integrado.

Cualquier ruido parasito sobre una tensión de salida de +2.4v (nivel lógico alto H) es ignorado en la entrada de la compuerta lógica a ella conectada, porque cualquier valor de tensión por encima de +2v es interpretado en la entrada como nivel lógico H. Por lo tanto, estos ruidos parásitos no afectan tampoco al correcto funcionamiento del circuito

De manera general se puede decir que cuanto mayor es la diferencia entre los valores de las tensiones de entrada y salida, mas amplio es el margen de ruido, y por lo tanto mayor es la inmunidad al ruido en el integrado.

En otros tipos de compuertas lógicas, cualquier tensión por debajo de +1.5v es interpretado como nivel lógico bajo (L o 0 binario), y cualquier tensión por encima de +3.5v, pero inferior a la alimentación, es interpretada como nivel lógico alto (H o 1 binario). La banda de tensiones comprendida entre los valores +1.5v y +3.5v es una "banda prohibida", porque en esa zona la puerta lógica es incapaz de interpretar el valor lógico de la señal de entrada. Para la salida la menor tensión que expresa un nivel lógico alto (H) es +4.9v y la máxima la de la fuente de alimentación. También para la salida el nivel lógico bajo esta comprendido de 0v a +0.1v como máximo, y una "banda prohibida" que va de los +0.1v a los +4.9v. Observe la figura 3.6

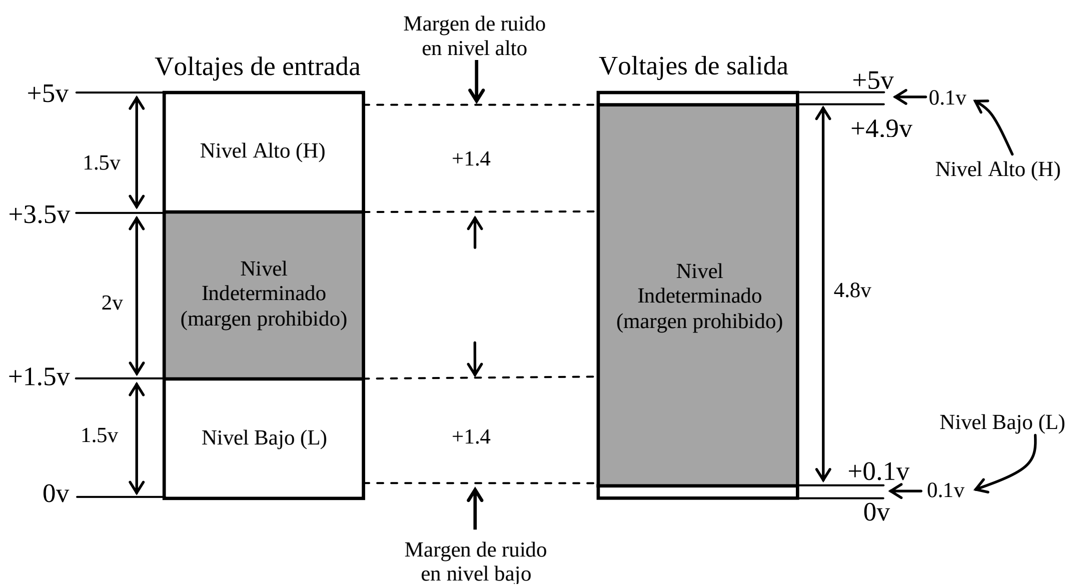


Figura 3.6. Niveles lógicos de entrada y salida y márgenes de ruido en otras compuertas.

De manera puntual tenemos que; cualquier ruido parasito (margen de ruido), sobre un voltaje de salida de +0.1v (nivel lógico bajo L) es ignorado en la entrada del circuito a ella conectada, siempre y cuando sea inferior a +1.4v, ya que si el ruido supera este valor su adición a los +0.1v del nivel lógico bajo de salida superaría los +1.5v que exige como máximo la entrada para considerarlo nivel bajo. Por lo tanto, estos ruidos parásitos no perturban el funcionamiento del circuito si se mantienen en los niveles mencionados.

Cualquier ruido parasito sobre un voltaje de salida de +4.9v (nivel lógico alto H) es ignorado en la entrada de la puerta lógica a ella conectada, porque cualquier valor de voltaje mayor a +3.5v es interpretado como nivel lógico alto H. Por lo tanto estos ruidos parásitos no afectan tampoco el correcto funcionamiento del circuito.

De la misma forma que cuanto mayor es la diferencia entre los valores de los voltajes de entrada y salida mas amplio es el margen de ruido y, por tanto, mayor es la inmunidad al ruido en el integrado.

Temperatura de Trabajo.

La corriente eléctrica que circula por un circuito integrado genera dentro de él calor. Por lo tanto la temperatura en un integrado es siempre superior a la del ambiente. Cada tipo de integrado tiene unos márgenes de temperatura de trabajo que no afectan a su correcto funcionamiento. Si la temperatura en el integrado esta fuera de dichos márgenes, el circuito trabajara deficientemente e incluso puede quedar destruido.

Factor de Calidad.

Aunque ya habíamos comentado algunos puntos con respecto a lo que es el factor de calidad de un circuito integrad, en este punto lo hacemos mas formalmente y podemos considerar para este a los siguientes dos puntos.

Primero; cuanto menor es su potencia de disipación.
Segundo; cuanto menor es su tiempo de propagación.

Aunque también habrá que considerar que una elevada velocidad de funcionamiento implica una mayor potencia disipada, es decir que ambos parámetros, son de alguna forma inversamente proporcionales.

La relación entre el tiempo de propagación y la potencia disipada se debe a las capacidades presentes en el integrado. Estas capacidades se cargan y descargan cuando se producen las transiciones entre los niveles lógicos a través de las resistencias transistores y diodos del integrado, formando constantes de tiempo RC.

Por otro lado el producto de una potencia por un tiempo expresa energía, es decir
$W = Pt$ y referido a los parámetros que venimos hablando se puede escribir de la siguiente manera; $W_P = P_D t_P$

Donde: W_P = Energía total disipada en pJ (1pJ = 10^{-12} joules)

P_D = Potencia disipada en mW

t_P = Tiempo de propagación en ns (1ns = 10^{-9} segundo)

Este producto puede tomarse a consideración como para conocer cierto factor de calidad de los circuitos integrados, ya que en el resultado de la ecuación, un integrado tendrá mejor calidad cuanto menor sea dicho resultado, es decir que la potencia de disipación junto con el tiempo de propagación sean los más pequeños.

3.2. Guía de diseño de circuitos combinacionales.

Una vez tratadas las técnicas de reducción y simplificación de circuitos combinacionales, es necesario e indispensable que el lector haya entendido a estas para que pueda reducir circuitos y para que pueda ver claramente la aplicación que las técnicas de reducción nos brindan, ya que muchas veces, si es muy cierto que en el proceso de formación del ingeniero llegan a quedar claras y con cierto manejo aceptable las técnicas de reducción y simplificación de circuitos; pero se mantiene insuficiente este conocimiento porque solo llega a quedar en eso, en un simple conocimiento adquirido y no como una aplicación.

Por tal motivo es indispensable que se pueda ligar el conocimiento hasta ahora visto con el siguiente paso de conocimiento, que es el diseño de los circuitos combinacionales, para posteriormente poder hacer la reducción de estos mismos circuitos; el momento donde la teoría se empieza a transformar en una realidad practica. Esto se lograra con la ayuda de una guía de diseño de circuitos combinacionales que se muestra a continuación y que consta de dos etapas importantes: la primera en la que el ingeniero deba tener o le hagan llegar la “necesidad de una tarea”, y la segunda la “conversión de dicha necesidad a una expresión booleana”.

3.2.1. Necesidad de una tarea.

El primer paso que hay que seguir en esta guía del diseño de circuitos combinacionales y que es punto de partida para iniciar algún tipo de diseño, es precisamente el tener un móvil, es decir tener la necesidad de una tarea, esto implica tener la respuesta clara y concreta a preguntas como ¿Qué es lo que se necesita hacer? Ya sea para buscar una solución o adaptación a una necesidad de automatizar, o bien, para facilitar alguna tarea deseada. Es necesario comentar que estas necesidades al ingeniero se le presentan en ocasiones como parte de un logro u objetivo posterior o general.

Uno de los objetivos de esta tesis es la aplicación de algunas técnicas de reducción en los circuitos combinacionales, y es también verdadero que la aplicación de estas técnicas este pensada en solucionar una infinidad de tareas, así que mencionaremos algunas necesidades de tareas reales que he observado en distintos lugares y que quedan como ejemplos en este trabajo en los subtemas 3.3, 3.4 y 3.5.

Tomando en cuenta que el empleo de las técnicas de reducción en los circuitos combinacionales son importantes en la función que desarrolla el ingeniero en su actividad profesional, es también importante, pero aun mas, el diseñar. Por lo que debemos considerar que las necesidades deben ser contempladas de acuerdo al tipo de tarea, al cual se le vaya a dar solución para así diseñar sobre la misma necesidad, ya que algunas tareas son abiertas y algunas cerradas.

Cuando hablamos de tareas abiertas hablamos de aquellas a las que se les puede satisfacer su necesidad con un múltiple de soluciones y por supuesto el ingeniero debe seleccionar la mejor. Las tareas cerradas tienen una sola solución, que generalmente se obtiene con métodos matemáticos y son los casos que se plantean en esta tesis.

Definir bien lo que se quiere hacer es de los aspectos mas importantes a considerar, ya que se tienen que establecer los límites y delimitar bien la o las necesidades a satisfacer que es lo que se esta buscando. Definir bien la tarea es la parte más complicada en el proceso de diseño, no definir bien a esta nos puede llevar a que no se resuelva en realidad la tarea.

Debemos considerar que la solución a una necesidad es resolver una tarea que se encuentra sujeta a algunas restricciones o limitaciones, por lo tanto la selección de la mejor solución debe realizarse de acuerdo con determinados criterios. Considerar que posiblemente cambiando los criterios o su importancia relativa se obtenga una solución distinta es indispensable en el desarrollo de cualquier diseño.

Las necesidades pueden variar, desde satisfacer tareas sencillas hasta las más complicadas. Por ejemplo mantener iluminadas ciertas áreas de alguna casa solo cuando estas están en uso (exista la presencia de alguna persona) y que a su vez solo sea por las noches, con la finalidad (necesidad) de ahorrar energía eléctrica.

Otro ejemplo a mencionar podría ser que alguna empresa de transporte de pasajeros tenga la necesidad de instalar una alarma sonora en todos sus autobuses de transmisión automática, para que realice la tarea de indicar al conductor que ha dejado de sostener el volante de manejo con ambas manos, cuando el vehículo se desplace a velocidades altas con la intención de que el conductor vuelva a retomar responsablemente el control del vehículo, y así guardar la mejor de las seguridades para los pasajeros.

En alguna ciudad que exista la necesidad de movilizar sus sistemas de ayuda a tiempo en conflictos referentes a choques automovilísticos (como otro ejemplo), podría crearse un sistema de aviso automático, el cual contaría con la instalación de circuitos con sensores en las zonas de mayor tránsito y que en coordinación con circuitos adaptados en los autos, emitan una señal aérea inmediata a los servicios auxiliares con la finalidad de acudir prontamente a la ayuda, quizás más específicamente se podrían adaptar circuitos que emitan una señal cuando las bolsas de aire de algún automóvil se activen, estas serian captadas por sensores colocados en las distintas áreas de la ciudad, dando a conocer no solo la existencia de un accidente si no la ubicación.

Un ejemplo más podría ser la adaptación de ventanas automáticas, que controlen la ventilación de alguna fabrica debido a la necesidad de que no exista acumulación de gases dañinos, así como una elevada o muy baja temperatura dentro del sitio, que genere un riesgo a los trabajadores o a toda la estructura en general y que evite un ideal ambiente de trabajo. El grado de la apertura de dichas ventanas estará controlada por sensores que detecten concentraciones altas de gases, humo o temperaturas fuera del rango ideal para la operación de la fabrica. Si los sensores de temperatura detectaran una elevación mayor de la misma, las ventanas abrirían a un máximo e irían cerrando de acuerdo a como la temperatura se fuera estabilizando, pero que también si existiera la acumulación de gases detectada por los sensores instalados para esta función, tendrían estos prioridad sobre los de temperatura para que realicen la tarea de controlar las ventanas.

3.2.2. Conversión de una necesidad a una expresión booleana y a un diagrama.

Otro de los puntos clave en esta guía del diseño de circuitos combinatorios es como se convierte una necesidad a una expresión booleana, un algebra booleana que como sabemos son las matemáticas para construir a los diagramas de los circuitos digitales.

Podremos empezar mencionando que una vez que tenemos la tarea planteada como un enunciado, lo podríamos cambiar por una expresión del algebra booleana para posteriormente transformar estas ecuaciones en un diagrama con compuertas para que realicen en su totalidad lo inicialmente planteado. Las ecuaciones las podemos lograr apoyándonos de tres operadores lógicos básicos del algebra de boole. Estas conjunciones booleanas unen enunciados u oraciones como lo harían las letras "y", "o" y/o una "negación", en otras palabras condiciones, ya sea para que se lleve a cabo algo siempre y cuando se cumplan simultáneamente estas sin excluir a ninguna o en otro caso el de que se lleve a cabo alguna tarea siempre y cuando al menos uno de los enunciados o condiciones se cumpla, o también, que el enunciado o condición pueda invertirse; estamos hablando de los operadores lógicos AND, OR y NOT.

Una forma de ejemplificar el operador AND utilizando enunciados comunes seria en un supuesto que se desee ir al cine, pero para ello se necesitan varias condiciones en la cual si faltara alguna no se podría asistir. La oración podría estar planteada como: *"¿Se tiene dinero* **y** *tiempo* **y** *le gusta la película que se exhibe* **y** *existen boletos para dicha película?"*

Donde observamos que la conjunción "**y**" une todas las condiciones para que se asista al cine y en el caso de que no se cumpla alguna condición simple y sencillamente no se asistirá. Esta conjunción "y" es equivalente al operador AND. Por supuesto estas oraciones y el operador lo podemos convertir en expresión algebraica. A la primera condición le podemos asignar la letra A, a la segunda B y asi sucesivamente, mientras que el operador matemático es el de la multiplicación por lo que se representa por "•", la respuesta o conclusión de si se asiste al cine o no la representamos con la letra "F" pudiendo quedar una expresión algebraica booleana como F = A•B•C•D o también F = ABCD (ya que el operador aunque no aparezca en esta segunda opción se da por entendido que existe y se encuentra entre las variables). En conclusión "F" será igual a un "si se va al cine" si todas las oraciones cumplen también un si, pero si alguno cumple un no, entonces la respuesta de la expresión seria el "no asistir". El si podemos compararlo con un 1 lógico, mientras que el no lo compararíamos con el 0 lógico.

De una forma similar existe el operador OR para unir enunciados con la conjunción "o", en donde en la unión de dos condiciones una condición es llevada a cabo y la otra no, o si son varias condiciones el resultado podrá ser llevado a cabo si al menos una condición es cumplida. Por ejemplo si se desea viajar de un sitio a otro podrá hacerse por alguna de las siguientes opciones: *"Para llegar a mi destino puedo viajar en avión* ***o*** *en automóvil particular* ***o*** *en autobús"*, de cualquiera de las opciones se podría trasladar, es decir que con al menos una opción que se cumpla ocurre entonces el traslado. Y como habíamos comentado anteriormente las oraciones pueden ser representadas como las variables de entrada y estas ser unidas por la conjunción "o" y representada en el algebra de Boole con el operador matemático de una suma (+), de tal forma que podríamos obtener booleanamente una función como F = A + B + C, en el caso de que viajar por avión se considere como la variable A, en automóvil particular por B, y en autobús por C, y F será un si o 1 lógico en el caso de que al menos una de las condiciones también cumplan con un si o 1 lógico y un no o 0 lógico en caso contrario.

Lo anterior es una primera forma de ilustrar como los enunciados los podemos convertir a expresiones booleanas. A continuación enunciaremos mas ejemplos que llevan a establecer de manera mejor este proceso de conversión:

Se desea diseñar un sistema automático de iluminación que se active por las noches, para la escalera de una casa con activación por presencia o transito por ella. Este diseño debe permitir *iluminar dicha escalera exterior cuando se circula por ella "**y**" además sea de noche*, es decir que la iluminación no debe activarse cuando haya luz de día. La casa es de solo dos niveles, por lo que los sensores que se coloquen en ella para detectar presencia podrían solo ser dos, uno en cada uno de los extremos de la escalera (S_1 = sensor inferior, S_2 = sensor superior) y para controlar que se active el sistema solo por las noches se utilizaría una fotocelda (se representara con la letra C). Por lo que un enunciado todavía más específico que posteriormente podríamos convertir ya a una expresión booleana seria: *Iluminar una escalera exterior cuando se active el S_1 "**o**" el S_2 ya que se circula por ella "**y**" además la fotocelda C se active por ser de noche.* Como se puede observar en este ejemplo existe ya una combinación de conjunciones o condiciones tanto del tipo "y" o "AND" y del tipo "o" u "OR". Hay que observar que se marca la letra "o" y la letra "y" porque son nuestros operadores que posteriormente convertiremos en compuertas. Mientras tanto la expresión algebraica seria F = (S1 + S2) • C, el diagrama seria el que se muestra en la figura 3.7.

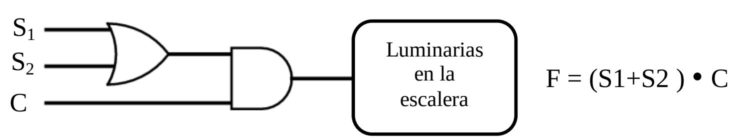


Figura 3.7. Diagrama del circuito combinacional de iluminación automática de una escalera.

De la figura 3.7 se observa como la compuerta OR donde se conectan las dos señales de los sensores S_1 y S_2 nos da la opción de que una o las dos se combinen con la señal de la fotocelda C a través de la compuerta AND para activar el sistema de iluminación.

Un ejemplo mas de cómo podemos realizar la conversión de una necesidad a una expresión booleana es el siguiente: En un invernadero por el tipo de plantas que se siembran, se requiere que dentro de el se activen aspersores de agua que se encuentran en el techo para su riego. Esto debe hacerse por un lapso de 5 minutos todas las tardes-noches, siempre y cuando la humedad de la tierra sea baja. La activación también puede ser manual a través de un interruptor siempre y cuando el sensor de la humedad marque deficiencia de humedad.

Aunque el enunciado está planteado de una manera sencilla es posible que no se logre ver totalmente las conjunciones que lo unen, pero podríamos plantearlo de la siguiente manera: "Los aspersores se encienden si en la tarde-noche (cuando empieza a obscurecer) se recibe un 1 lógico a través de la foto celda que se encuentra fuera del invernadero (entrada B), además un sensor de humedad (entrada D) que puede asignar un 1 lógico si dicha humedad es alta o bien un pulso lógico 0 si esta es baja **y** se oprime un interruptor manual (entrada A) pulsado 1 lógico y no pulsado 0 lógico para que los aspersores puedan verter el agua. La expresión algebraica debe entregar un 1 que active los aspersores si B se activa con un 1 **y** además D a través del inversor marca un 1 **o** bien D manda un 1 **y** se pulsa un interruptor activando un 1, quedando así: F = AD' + (BD')C, el cronometro es una entrada adicional llamada C. El diagrama que representa a dicha expresión aparece en la figura 3.8.

Observe en la figura 3.8 que en la entrada D el sensor de humedad arroja un 0 lógico con falta de humedad por lo que fue conveniente agregar un inversor para que invierta su valor a un 1 lógico antes de agregarse a las compuertas AND y se realizara la operación en conjunto con un valor de 1 lógico para activar los aspersores.

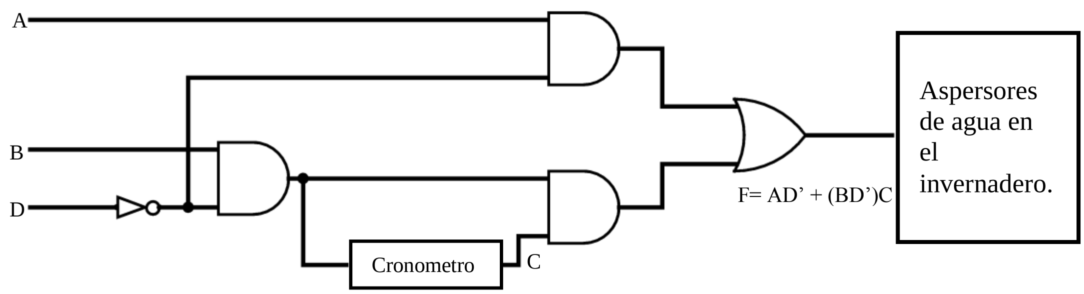


Figura 3.8. Diagrama del circuito combinacional de activación automática (B y D) o manual (A y D) de aspersores en un invernadero.

Se debe aclarar que el cronometro se activa con un pulso de 1 lógico y aunque este se mantenga el cronometro solo entrega un pulso de 1 lógico durante 5 minutos que dura el riego.

3.3. Diseño de un circuito combinacional para la apertura y cierre de un portón deslizante sobre riel.

Como ya se mostro en los subtemas anteriores, se necesitan al menos dos elementos para llevar a cabo el diseño de un circuito combinacional estos son los que se mencionan en nuestra guía de diseño de circuitos combinacionales y que son; tener una necesidad de una tarea y convertir dicha necesidad a una expresión booleana. Por lo que procederemos a realizar la aplicación de dichos elementos que aparecen en nuestra guía en el caso del diseño de un circuito para abrir y cerrar un portón que se desliza sobre un riel.

3.3.1. Necesidad de la tarea.

Se requiere controlar un portón deslizante sobre un riel de manera automática, por lo que se necesita que esta tarea sea controlada por un circuito combinacional que en el caso de que el portón se encuentre cerrado pueda ser abierto por medio de dos opciones: utilizando un control remoto o de manera manual con un interruptor de dos posiciones, y de la misma forma si inicialmente el portón se encuentra en una posición de abierto este inicie el trayecto para el cerrado. Por otra parte, se necesita que por cuestiones de seguridad para las personas como para el mismo sistema electromecánico que da movimiento de apertura y cierre de este portón, cuando inicie el proceso de apertura se detenga si en el trayecto sobre el riel se encuentra un obstáculo que impida el deslizamiento libre de dicho portón y continúe su apertura en cuanto el obstáculo haya sido removido; de este mismo modo cuando este en el proceso de cerrado. Cabe mencionar que el sistema electromecánico que da movimiento al portón funciona de la siguiente manera: cuenta con una interfaz de dos opciones de trabajo, la primera es de cerrado al recibir un 1 lógico el motor enciende en el sentido de las manecillas del reloj para que este cierre y en ausencia de este 1 lógico es decir, que si hay un 0 lógico este motor se detiene. La segunda opción de trabajo es de apertura, en el cual al recibir un 1 lógico el motor se mueve en el sentido contrario a las manecillas del reloj y se detiene si a esta opción le llega un 0 lógico, en cualquiera de estos dos casos el proceso continua si la obstrucción deja de existir. Además, si el portón esta asegurado en su cierre total con la chapa, el mecanismo no deberá funcionar hasta que esta se encuentre en la posición de abierta, en ambos casos, de cerrado y abierto, ya que si el portón se encuentra cerrado y asegurado con la chapa, el mecanismo no realizaría el trabajo provocando el daño del dispositivo electromecánico y si el portón se encuentra abierto con la chapa cerrada, esta evitara un buen ensamblaje en el momento del cierre por lo que continuara trabajando el sistema electromecánico.

3.3.2. Conversión de la necesidad a una expresión booleana y a un diagrama.

Siguiendo la guía de diseño de circuitos combinacionales, el segundo paso a realizar es la conversión de dicha necesidad a una expresión booleana, para este caso la necesidad es realizar el circuito combinacional que controle la apertura y cierre de un portón deslizante sobre un riel. Debemos tomar en cuenta que nuestro circuito combinacional va a arrojar dos salidas: una controlara el avance y el paro en la apertura del portón y la otra salida controlara el avance y el paro, pero ahora será del cierre del mismo; por lo que al crear nuestra expresión booleana en realidad estaremos creando dos, ambas estarán dependiendo de las mismas variables, por lo que existen dos planteamientos implícitos en un solo enunciado.

Empezamos por mencionar que la activación para el cierre o apertura del portón será con las salidas F_1 y F_2 donde F_1 controlara la apertura del portón con un 1 lógico y detendrá este proceso de apertura con un 0 lógico. De la misma manera F_2 gobernara el proceso de cerrado con un 1 lógico y detendrá este proceso con un 0 lógico.

Ahora bien las componentes que activaran si el portón deslizante abre, cierra o detiene su curso en cualquiera de los dos primeros casos son: un interruptor manual de activación para la apertura o cierre. Esta activación para abrir (si estaba cerrado) y cerrar (si estaba abierto) es un 1 lógico y estaría en estado inactivo con un 0 lógico (variable A). De forma similar la activación de esta apertura y cierre puede darse por medio de un dispositivo de control remoto que mantendría activo el proceso con un 1 lógico y desactivo con un 0 lógico (variable B). Un tercer dispositivo (sensor de presencia) es usado para indicar cuando el portón esta totalmente cerrado, que para este caso suministraría al circuito un 1 lógico y un 0 lógico cuando se encuentre totalmente abierto (variable C). Otras dos variables a tomar en cuenta son las de dos dispositivos colocados a los extremos del portón. El primero detecta la presencia de algún objeto que obstruya el tramo de riel al momento de que el portón abra, si es este el caso, el dispositivo otorgaría un 1 lógico que provocaría la detención de la apertura, si el trayecto esta libre el dispositivo mantendrá un 0 lógico y seguirá el proceso de apertura (variable D). El segundo dispositivo hace exactamente lo mismo pero en el proceso de cerrado, envía un 0 si no existe obstrucción alguna y un 1 si la hay, si es 0 la señal que envía, el cerrado continua sin problemas, mientras que si existe un 1 el portón se detendrá (variable E). Existe una última variable que condiciona las salidas F_1 y F_2, esta corresponde a una chapa electrónica que tiene la intención de brindar un bloqueo al portón, para brindar un extra en la seguridad, esta chapa envía un 1 lógico si se encuentra abierta o desbloqueada y un 0 lógico si se encuentra cerrada (variable F que no provocaría confusión con las salidas ya que estas últimas tienen subíndices). Esta evita que se abra o cierre el portón si se encuentra asegurada, cerrada o bloqueada.

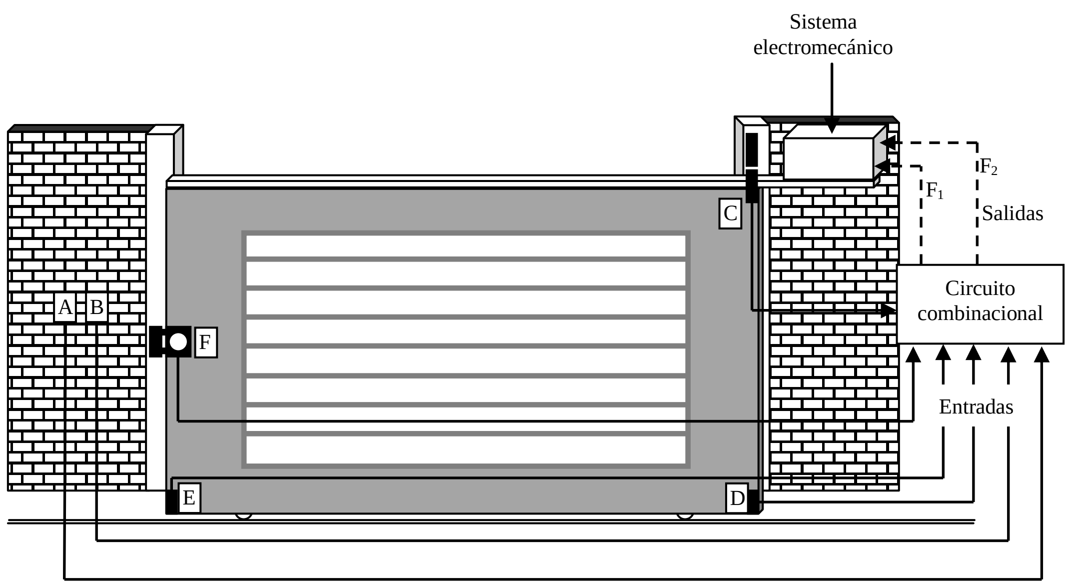


Figura 3.9. Portón deslizante sobre riel.

Con estos antecedentes se puede hacer un dibujo tener a una representación como la mostrada en la figura 3.9, además los enunciados para la apertura y cierre del portón que representan el funcionamiento del circuito combinacional quedarían de la siguiente manera:

Primeramente para controlar la apertura del portón el enunciado quedaría como: *"El portón abrirá si estando el portón en la posición de totalmente cerrado **"y"** la chapa abierta **"y"** activando la apertura por medio del interruptor manual **"y"** aun si el dispositivo de control remoto se encuentra desactivado **"y"** al continuar su avance no encuentra obstrucción alguna **"o"** de forma similar con las misma condiciones solo* *que esta vez se activara por medio del control remoto, es decir, que el portón este cerrado **"y"** que la chapa este abierta **"y"** se activara la apertura por el control remoto **"y"** aun este inactivo el interruptor manual **"y"** no haya obstrucciones en el riel **"o"** para el caso en el que llegara a activarse simultáneamente el interruptor manual **"y"** el de control remoto, **"y"** mientras también el resto de las condiciones sigan iguales; cerrado el portón **"y"** la chapa abierta **"y"** sin obstrucción el riel".*

En segundo caso para el cierre del portón el enunciado seria: *"El cierre del portón se llevara a cabo si este en primer lugar se encuentra abierto **"y"** se activa el interruptor manual **"y"** sin que el control remoto este activo **"y"** la chapa este abierta **"y"** en el recorrido no hubiera algún tipo de obstrucción **"o"** bien se puede activar el cerrado con el control remoto **"y"** aun el interruptor manual este en inactivo **"y"** la posición del portón sea de totalmente abierto **"y"** la chapa se encuentre abierta **"y"** no haya obstrucción el riel, **"o"** también puede existir el caso de que el cerrado se lleve a cabo si se mantienen activos el interruptor manual **"y"** el control remoto **"y"** el portón se encuentre en la posición de totalmente abierto **"y"** la chapa abierta **"y"** no haya obstrucciones en el riel".*

Para el caso de la apertura del portón y reduciendo las opciones de la tarea por las variables que representan a las mismas podríamos armar un nuevo enunciado un poco mas reducido en extensión y así ir tomando un aspecto mas de expresión algebraica quedando:

F_1 es un 1 si C=1 y F=1 y A=1 y B=0 y D=0, o bien F_1 será 1 si C=1 y F=1 y B=1 y A=0 y D=0, o también F_1 es 1 si A=1 y B=1 y C=1 y F=1 y D=0.

Cambiando las conjunciones "o" e "y" por los operadores lógicos a los que representan la expresión booleana quedaría:

F_1 = CFAB'D' + CFBA'D' + ABCFD'

Ordenando alfabéticamente cada bloque de términos tenemos:

$F_1 = AB'CD'F + A'BCD'F + ABCD'F$

Haciendo el mismo proceso para el cerrado del portón es decir que F_2 sea igual a 1 quedaría:

F_2 es 1 si C=0 y A=1 y B=0 y F=1 y E=0, o bien F_2 será 1 si B=1 y A=0 y C=0 y F=1 y E=0, o también F_2 es 1 si A=1 y B=1 y C=0 y F=1 y E=0

Cambiando las conjunciones "o" e "y" por los respectivos operadores quedaría:

$F_2 = C'AB'FE' + BA'C'FE' + ABC'FE'$

Ordenando alfabéticamente:

$F_2 = AB'C'E'F + A'BC'E'F + ABC'E'F$

Una vez que ya se tienen las dos expresiones booleanas se procede a armar el diagrama del circuito combinacional que incluya a ambas expresiones, y no es complicado con los antecedentes que ya hemos manejado en esta tesis, pero aun así podemos comentar que basta con colocar todas nuestras entradas del lado izquierdo de donde pensemos colocar nuestro circuito y del lado derecho colocar nuestras dos salidas e ir colocando nuestras compuertas entre nuestras entradas y salidas, compuertas que representan los términos booleanos de nuestras expresiones, en este caso nuestro diagrama eléctrico quedaría como se muestra en la figura 3.10.

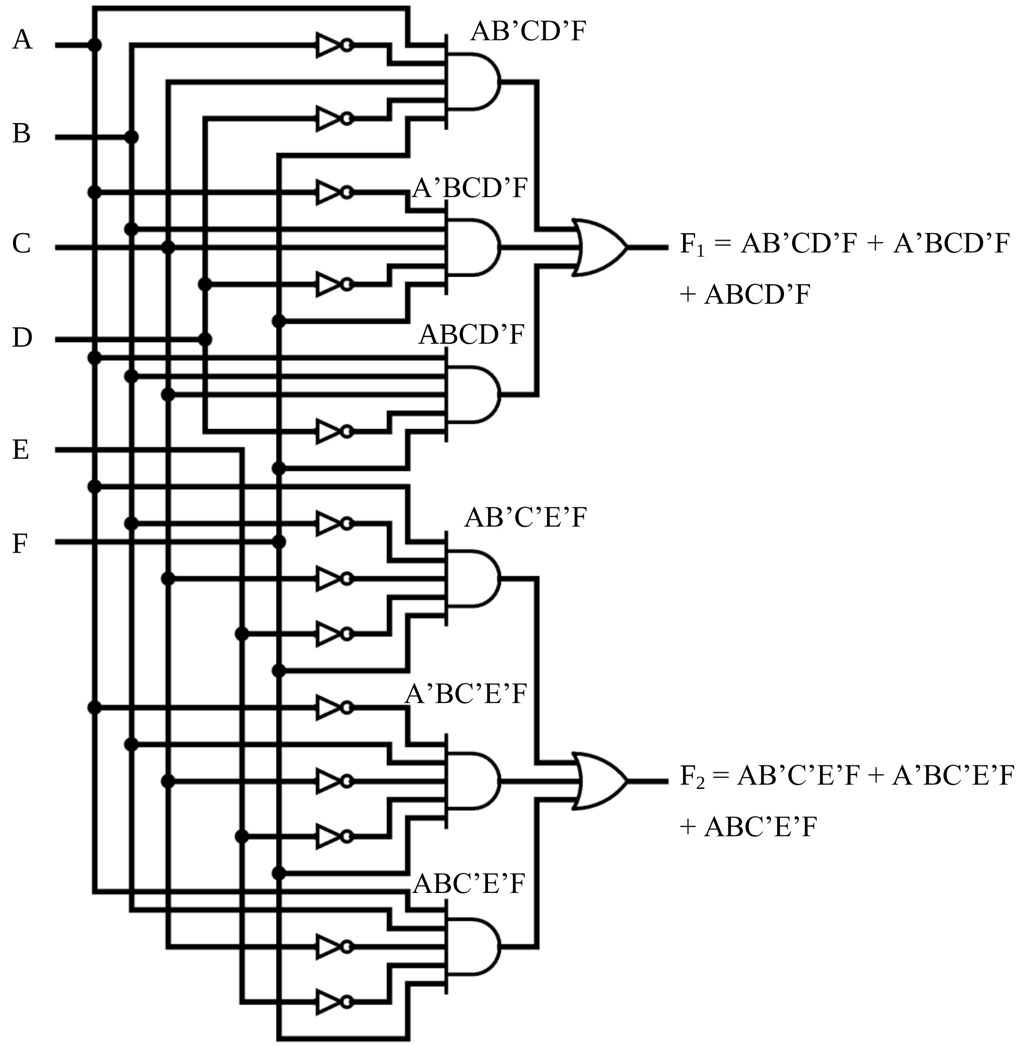


Figura 3.10. Diagrama eléctrico del circuito combinacional para la apertura y cierre un portón deslizante sobre riel.

De la figura 3.10 debe observarse que a pesar de que existan dos funciones de salida que realizan tareas relativamente distintas, en ambos casos estas ocupan las mismas variables de entrada, por lo que podemos aseverar que se trata de un solo circuito, y cabe mencionar que en algunas bibliografías el trato para este tipo de circuitos se maneja y observa de manera separada, es decir que el análisis lo hacen por separado, un circuito para cada función de salida, en nuestro caso lo dejaremos tal como esta.

3.4. Diseño de un circuito combinacional para activación de timbre escolar.

Para realizar el diseño de este circuito combinacional recurrimos a nuestra guía de diseño de circuitos combinacionales. Por principio tener en cuenta ¿Cuál o cuáles son las necesidades a cubrir por el circuito? y así designar la tarea a realizar, posteriormente o como siguiente pasos, hacer la conversión de dicha necesidad a una expresión booleana y finalmente convertir a dicha función en un diagrama de compuertas lógicas para que realicen la tarea.

3.4.1. Necesidad de la tarea.

Se desea controlar un timbre escolar, para que sea activado por varias razones, es decir que tenga la posibilidad no solo de avisar horarios de inicio y termino de un día de actividades escolares en alguna institución (a través de un reloj programable), sino que además pueda ser utilizado como una alarma que indique la existencia de fuego o humo dentro de las instalaciones escolares, esto con ayuda de algún o algunos sensores de humo en el sitio, además de poder servir como alarma nocturna, es decir; que se active si se detecto presencia en el interior del mismo modulo administrativo por las noches. Por último que este pueda ser activado manualmente para alguna actividad fuera de las consideraciones normales de operación automática que se mencionan con anterioridad.

3.4.2. Conversión de la necesidad a una expresión booleana y a un diagrama.

La conversión de nuestra necesidad a cubrir por una expresión booleana es el segundo paso de nuestra guía. Como ya sea había comentado en el subtema 3.2.2 debemos ordenar y adaptar la necesidad que ya planteamos en un enunciado unido coherentemente por las conjunciones “o” e “y”, o hacer las negaciones necesarias dentro de él para poderlo pasar a una expresión booleana, cambiar nuestras conjunciones por los operadores lógicos del algebra de boole.

Dentro de los elementos o componentes a considerar que controlaran la activación de nuestro timbre escolar (salida F) el cual será activado por un uno lógico y estará inactivo con un cero lógico serán; un interruptor manual de dos pasos que activaría el timbre con un 1 lógico y desactiva con 0 lógico (variable A), reloj programable que expulsa una señal digital 1 en ciertos horarios durante el día para avisar cambios en horarios, en otros momentos mantendrá un cero lógico a su salida (variable B), una fotocelda que nos indicara si es de día con un cero lógico y de noche con un 1 lógico (variable C), sensores de movimiento que nos indican cualquiera de alguno de ellos movimiento con un 1 y si no existe el movimiento mantendrían como respuesta un 0 (variable D), un sensor de humo que indique la existencia de tal con un 1 y en caso contrario mantenga un 0 lógico (variable E). En la figura 3.11 se puede observar el acomodo de los componentes.

El enunciado quedaría como; *"El timbre escolar se activara (recordar que se activa con un 1 lógico y se mantiene inactivo con un cero lógico) en el caso que se oprima el interruptor manual* ***"y"*** *sea de día,* ***"o"*** *en otro caso la activación del timbre estará supeditada a la señal que marque el reloj programable* ***"y"*** *sea también de día* ***"y"*** *aun estando el interruptor manual en apagado (en estado cero),* ***"o"*** *en un tercer caso el timbre escolar sonara si siendo de noche* ***"y"*** *se detecta movimiento al interior del modulo administrativo* ***"y"*** *aun siga en estado cero el interruptor manual,* ***"o"*** *como una cuarta opción de activación del timbre escolar es que sin importar si el interruptor manual está en estado cero* ***"y"*** *es de día* ***"y"*** *el sensor de humo detecta tal, el timbre se activara,* ***"o"*** *por ultimo si de la misma manera el interruptor manual sigue apagado* ***"y"*** *es de noche* ***"y"*** *se detecta humo habrá una activación del timbre".*

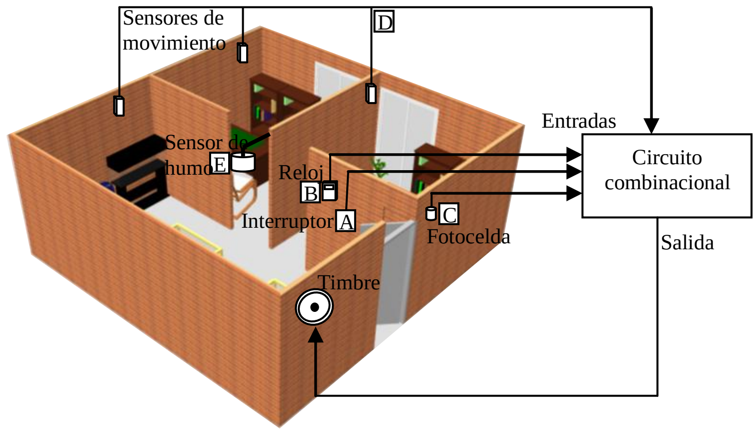


Figura 3.11. Modulo escolar.

Reduciendo un poco el enunciado y cambiando las opciones que intervienen en la tarea de activar el timbre escolar por sus variables correspondientes puede quedar de la siguiente forma:

F es 1 en el caso de que A=1 y C=0 (es decir C'=1) o B=1 y C'=1 y A'=0 o C=1 y D= 1 y A'=0 o A'=0 y C=1 y E=1 o A'=0 y C'=0 y E=1.

Quedando finalmente:

F = AC' + A'BC' + A'CD + A'CE + A'C'E

Al convertir cada uno de los términos del algebra booleana tenemos el siguiente circuito combinacional, donde se observan cada una de las combinaciones de la variables, hasta llegar a la función final F.

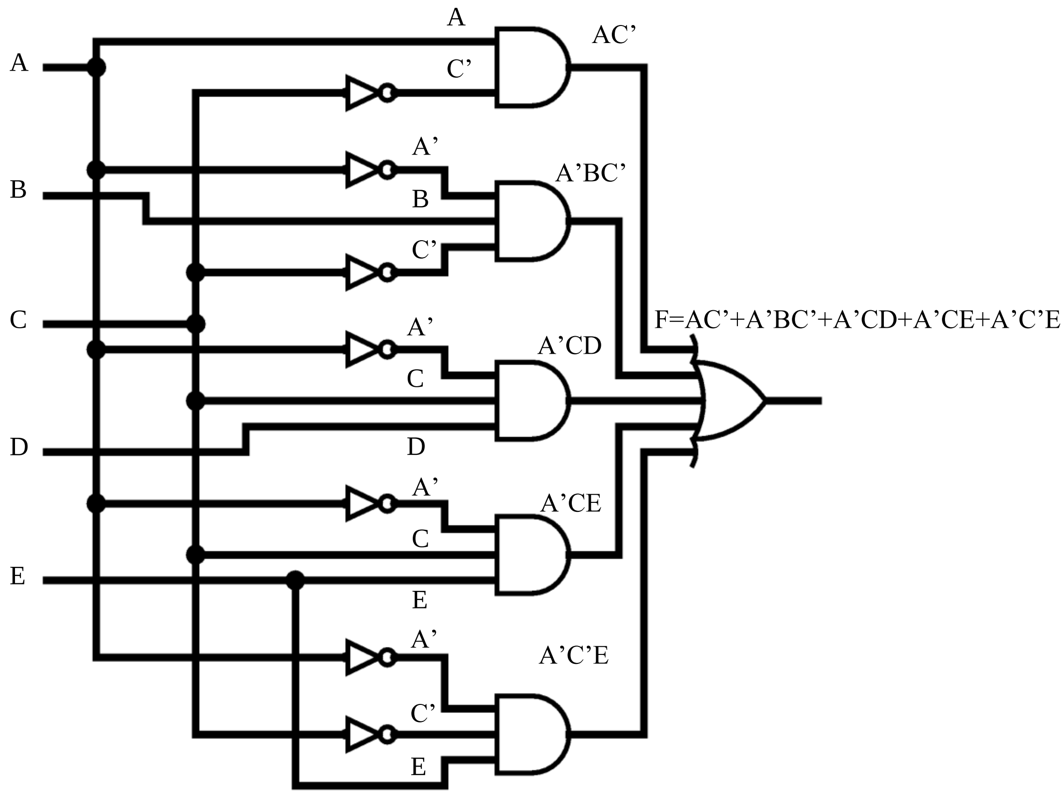


Figura 3.12. Diagrama del circuito combinacional de activación de timbre escolar.

3.5. Diseño de un circuito combinacional para control simultaneo del llenado de dos depósitos elevados independientes de agua y sistema de riego, con alimentación de dos cisternas.

La intención del diseño para este caso la iniciaremos por la necesidad de una tarea, y si es necesario mencionar que por necesidades reales, es decir que en dado momento y en diferentes lugares se me ha planteado la posibilidad de diseñar o crear una solución de automatización, por supuesto la respuesta al planteamiento es un sí, uno de los casos es el presente; en el que se cuenta ya con una serie de instalaciones hidráulicas, es decir, con dos tanques elevados independientes y que se quieren seguir manteniendo de la misma forma y dos cisternas que de la misma manera se encuentran una de otra independientes, ¿Cómo controlar el llenado de ambos ocupando las dos cisternas? y sea dicho de paso controlar el riego de un área verde. A continuación en la necesidad de la tarea se realizara el planteamiento completo.

3.5.1. Necesidad de la tarea.

Nuevamente el primer paso a seguir que nos marca nuestra guía es la de conocer la necesidad de una tarea que para este caso existe la situación de controlar el llenado de dos tanques elevados que se encuentran independientes uno del otro y el suministro de agua para estos dos tanques dependerá de dos cisternas que al igual que los tinacos se encuentran independientes. Cada cisterna cuenta ya con un sistema de bombeo (bomba de agua), faltaría hacer las conexiones hidráulicas pertinentes para un buen control del llenado, la propuesta es que de cada cisterna se bombee agua al mismo tiempo a ambos tanques y que cuando el llenado de un tanque sea total se cierre el suministro por medio de una válvula electrónica para que el otro continúe con su llenado, cuando ambos tanques estén llenos las dos bombas dejaran de funcionar. En caso de que solo una cisterna tenga agua, sea cual fuere, esta suministrara agua a ambos tanques elevados, o solo al que necesite ser llenado, además de que estas bombas suministren agua a aspersores para un área verde cuando la humedad sea escaza.

3.5.2. Conversión de la necesidad a una expresión booleana y a un diagrama.

El segundo paso a seguir como ya lo hemos hecho en casos anteriores y que nuestra guía nos marca, es convertir esta necesidad en una expresión booleana. Para lograr la conversión es pertinente plantear nuestras salidas (es decir que va a hacer nuestro circuito) y también nuestras estradas (las variables que tomara en cuenta el circuito para funcionar), para posteriormente organizar en uno o algunos enunciados coherentemente (si es una sola salida será un solo enunciado, si son varias salidas serán varios enunciados que se relacionan entre sí ya que utilizan las mimas variables de entrada).

Dentro de las componentes de entrada que se consideran para este circuito se encuentran dos sensores de nivel de agua (uno para cada cisterna), que indican si hay agua (suficiente) y entregan al circuito que se va a diseñar un 1 lógico, mientras que si existe ausencia de agua estos sensores marcan un 0 lógico, de esta forma un sensor para una cisterna seria la variable A, mientras que el otro sensor en la otra cisterna seria la variable B, podemos concluir de estas dos variables de entrada que cuando haya agua suficiente en ellas, ambas marcaran 1's lógicos en el caso de estar vacías marcaran 0's lógicos. De la misma forma otros dos sensores del mismo tipo estarán uno en cada tanque elevado, por lo que de ahí tendríamos dos variables mas la variable C para un tanque y para el otro la variable D, ambos indicando suficiente agua con un 1 lógico y ausencia de ella con un 0 lógico. Una última variable es un sensor de humedad ubicado en el área verde que se desea mantener regada. El sensor nos indicara si hace falta humedad o la humedad es suficiente, si esta es suficiente o la ideal el sensor marcara un 1 lógico que mandara hacia nuestro circuito combinacional o en caso de que la humedad sea insuficiente o escaza el sensor enviara un 0 lógico.

Estas variables activaran cinco salidas. La primera F_1 accionara la bomba de agua que se encuentra en la cisterna que contiene la variable de entrada A (observe figura 3.13), esta activación la realizara si a su salida se encuentra un 1 lógico, y se mantendrá apagada cuando esta salida sea un 0 lógico. La otra salida llamada F_2 se comportara de la misma manera, activando la bomba con un 1 lógico y desactivándola con un 0 lógico, solo que esta controlara a la bomba de la cisterna que contiene el sensor de la variable B (vea figura 3.13). Las otras tres salidas controlan válvulas electrónicas que permiten el paso del agua que es bombeada. La F_3 controla la válvula que permite o no el paso de agua al tanque elevado que contiene la variable C, con un 1 lógico abre la válvula y permite el paso del fluido y con un 0 lógico se mantiene cerrada. La F_4 al igual que la F_3 tiene el mismo funcionamiento solo que esta controla el acceso al tanque de la variable D. La F_5 de la misma forma controla el paso de agua o lo evita, pero esta vez para el área verde, la figura 3.13 muestra un diagrama de cómo está dispuesta la propuesta del llenado de los dos tanques y el sistema de riego.

Observe de la figura 3.13 que las flechas que entran al circuito combinacional son variables de entrada y las flechas que salen son las señales que controlan tanto las bombas como válvulas, además vea dos válvulas que están conectadas inmediatamente a la salida de las bombas, estas son de un solo sentido y funcionan mecánicamente, estas evitaran el regreso del fluido, esto una vez que sea solo una bomba la que realice el traslado del liquido.

Hay que aclarar que en este caso tendremos que formar varios enunciados ya que existen en este proceso de diseño varias salidas, y también para mayor facilidad en el momento de crear los enunciados la cisterna que contiene al sensor A o variable A la llamaremos cisterna A, de la misma forma será para la cisterna que contenga al sensor B, a la cual llamaremos cisterna B, lo mismo dispondremos para los tanques que serán; tanque C y D según corresponda (vea figura 3.13). Así mismo para las bombas y válvulas serán llamadas según la salida que las controle, por ejemplo la bomba 1 será la que sea manejada por la salida F_1, bomba 2 manejada por la F_2 y así sucesivamente con las válvulas.

Uno de los primeros enunciados y que encendería la bomba 1 es: *"La bomba 1 enciende si hay agua en la cisterna A **"y"** esto a pesar de que en la cisterna B no haya agua **"y"** en el tanque C no haya agua **"y"** tampoco en el tanque D **"y"** haga falta humedad al área verde **"o"** encenderá si hay agua en la cisterna A **"y"** siga sin haber agua en la cisterna B **"y"** haga falta agua en el tanque C **"y"** aun teniendo agua el tanque D **"y"** la humedad del área verde sea buena **"o"** el encendido*

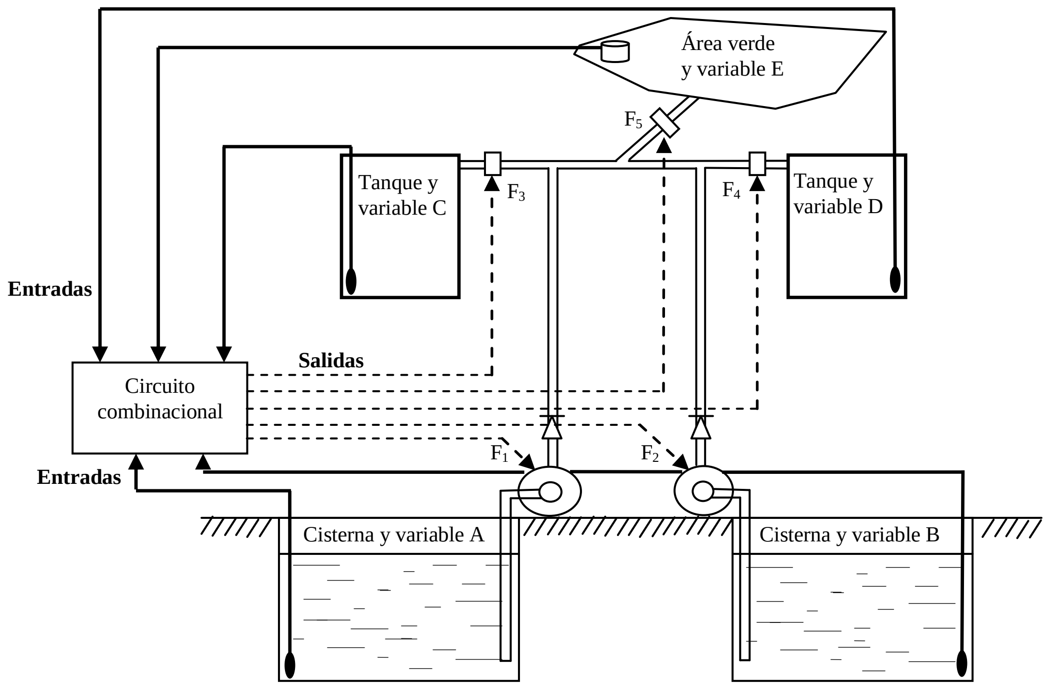


Figura 3.13. Tanques y cisternas independientes con sensores de nivel y sensor de humedad en área verde.

iniciaría si hay agua en cisterna A ***"y"*** *no en la cisterna B* ***"y"*** *haya agua en el tanque C* ***"y"*** *no haya agua en tanque D* ***"y"*** *humedad buena en área verde* ***"o"*** *también encenderá si hay agua en la cisterna A* ***"y"*** *no haya en la cisterna B* ***"y"*** *aun teniendo agua el tanque C* ***"y"*** *el tanque D* ***"y"*** *no haya humedad suficiente en el área verde* ***"o"*** *la bomba 1 encenderá también si existe agua en la cisterna A* ***"y"*** *en la B* ***"y"*** *no haya agua en el tanque C* ***"y"*** *en tanque D* ***"y"*** *no haya buena humedad en el área verde* ***"o"*** *encenderá si en cisterna A sigue habiendo agua* ***"y"*** *al igual en B* ***"y"*** *no haya agua en el tanque C* ***"y"*** *aun habiendo agua en el tanque D* ***"y"*** *la humedad sea buena en el área verde* ***"o"*** *haya agua en las cisternas A* ***"y"*** *B* ***"y"*** *tanque C, este último con agua suficiente* ***"y"*** *en pero el tanque D no tenga agua* ***"y"*** *que la humedad del área verde sea buena* ***"o"*** *bien que las dos cisternas tengan agua*

*A **“y”** B, **“y”** que el tanque C **“y”** D tengan suficiente agua **“y”** haga falta humedad en el área verde”.*

Realizando las reducciones pertinentes en lo que es la activación de la bomba 1 que es la salida F_1 queda de la siguiente forma:

F_1 será 1 cuando A=1 y B=0 y C=0 y D=0 y E=0 o bien A=1 y B=0 y C=0 y D=1 y E=1 o también A= 1 y B=0 y C=1 y D=0 y E=1 o igualmente si A=1 y B=0 y C=1 y D=1 y E= 0 o por otra parte encenderá si A=1 y B=1 y C=0 y D=0 y E=0 o nuevamente accionaria si A=1 y B=1 y C=0 y D=1 y C=1 o A=1 y B=1 y C=1 y D=0 y E=1 o A=1 y B=1 y C=1 y D=1 y E=0

Simplificando aun más quedaría:

F_1 = AB’C’D’E’ + AB’C’DE + AB’CD’E + AB’CDE’ + ABC’D’E’ + ABC’DE + ABCD’E + ABCDE’

El segundo enunciado que corresponde a la salida F_2 y que controla a la bomba 2 quedaría: *“La bomba 2 encenderá en los casos en que no haya agua en la cisterna A **“y”** si la haya en la cisterna B **“y”** el tanque C **“y”** el tanque D no tengan agua **“y”** además al área verde le haga falta humedad **“o”** también siga sin agua la cisterna A **“y”** la cisterna B cuente con agua **“y”** el tanque C no tenga agua **“y”** el tanque D si tenga liquido **“y”** el área verde contenga humedad suficiente **“o”** si la cisterna A sigue sin agua **“y”** la cisterna B contenga agua **“y”** el tanque C tenga agua **“y”** aun si el tanque D este sin agua **“y”** el área verde cuente con humedad suficiente, **“o”** siguiendo la cisterna A vacía **“y”** en pero la cisterna B con agua **“y”** el tanque C con agua **“y”** el tanque D con agua **“y”** haga falta humedad al área verde **“o”** en el caso de que la cisterna A **“y”** B contengan agua **“y”** el tanque C **“y”** el tanque D no tengan agua **“y”** al igual al área verde le haga falta humedad **“o”** en otra opción de encendido de la bomba la cisterna A **“y”** la cisterna B cuenten con agua **“y”** el* *tanque C necesite ser llenado **"y"** el tanque D tenga agua **"y"** el área verde cuente con humedad suficiente, **"o"** siguiendo con agua las dos cisternas A **"y"** B **"y"** el tanque C **"y"** en pero el tanque D no tenga agua **"y"** el área verde siga con la humedad suficiente **"o"** también la cisterna A contenga agua **"y"** la cisterna B con agua **"y"** tanque C **"y"** tanque D también con agua **"y"** el área verde necesite agua por la falta de humedad".*

Nuevamente se realizan las simplificaciones necesarias para el control de la bomba 2 que corresponde a la salida F_2 de nuestro enunciado anterior y que se vería de la siguiente manera.

F_2 será 1 cuando A=0 y B=1 y C= 0 y D=0 y E=0 o cuando A=0 y B=1 y C=0 y D=1 y E=1 o A=0 y B=1 y C=1 y D=0 y E=1 o también cuando A=0 y B=1 y C=1 y D=1 y E=0 o en donde A=1 y B=1 y C=0 y D=0 y E=0 o A=1 y B=1 y C=0 y D=1 y E=1 o cuando sea A=1 y B=1 y C=1 y D=0 y E=1 o sea A=1 y B=1 y C=1 y D=1 y E=0

Simplificando aun más la expresión quedaría:

F_2= A'BC'D'E' + A'BC'DE + A'BCD'E + A'BCDE' + ABC'D'E' + AB C'DE + ABCD'E + ABCDE'

El enunciado para la función F_3 que controla la apertura y cierre de la válvula 3 (observe figura 3.13) seria en los casos en que hiciera falta agua en el tanque C y hubiera agua en las dos cisternas o en alguna, solo de esta forma se abriría para el llenado, es decir recibiendo un 1 lógico, quedaría de la siguiente manera:

*"La válvula 3 abre cuando la cisterna A contiene agua suficiente **"y"** la cisterna B también **"y"** le hace falta agua al tanque C **"o"** si la cisterna A esta sin agua **"y"** la cisterna B tiene agua **"y"** le hace falta agua al tanque c **"o"** bien en otro caso si la*

*cisterna A tiene agua **“y”** la cisterna B no tiene agua **“y”** al tanque C le hace falta agua”*

Para este enunciado también realizamos las simplificaciones necesarias y la función se vería de la siguiente manera:

F_3 será 1 si A=1 y B=1 y C=0 o si A=0 y B=1 y C=0 o si A=1 y B=0 y C=0

Así como expresión booleana queda:

F_3 = ABC’ + A’BC’ + AB’C’

El enunciado para F_4 que controla la otra válvula y que es la que permite la entrada de agua al tanque D seria:

*“La válvula abre cuando la cisterna A tiene agua **“y”** la cisterna B también **“y”** cuando el tanque D no tiene agua **“o”** cuando la cisterna A tiene agua **“y”** la cisterna B no tiene agua **“y”** el tanque D tampoco **“o”** la cisterna A no tenga agua **“y”** la cisterna B tenga agua **“y”** el tanque D no tenga agua”*

Simplificando el enunciado tenemos:

F_4 será 1 cuando A=1 y B=1 y D=0 o cuando A=1 y B=0 y D=0 o cuando A=0 y B=1 y D=0

En forma de expresión booleana:

F_4 = ABD’ + AB’D’ + A’BD’

E igualmente para la última función que es la F_5 como lo fue para F_3 y F_4 esta controla la apertura y cierre de una válvula que deja pasar o evita caudal de agua para que se riegue una área verde y dependerá de la ausencia de humedad en dicha área y que haya agua en alguna o ambas cisternas para que la válvula abra con un 1 lógico. El enunciado seria como el siguiente:

"La válvula 5 dejara pasar agua hacia instalaciones hidráulicas de riego si existe agua en la cisterna A ***"y"*** *en la cisterna B* ***"y"*** *el sensor indica humedad insuficiente* ***"o"*** *cuando la cisterna A mantiene agua en ella* ***"y"*** *la cisterna B esta vacía* ***"y"*** *el sensor marca humedad insuficiente* ***"o"*** *en caso de que la cisterna A no tenga agua* ***"y"*** *la cisterna B tenga agua* ***"y"*** *el sensor marque humedad insuficiente".*

Simplificando el enunciado quedaría que:

F_5 sería igual a 1 cuando A=1 y B=1 y E=0 o A=1 y B=0 y E=0 o A=0 y B=1 y E=0

Haciéndola como expresión booleana seria:

$F_5 = ABE' + AB'E' + A'BE'$

Realizando la confección del circuito combinacional con las cinco entradas A, B, C, D y E, además de con las 5 salidas que contienen a las mencionadas entradas tenemos el circuito de la figura 3.14.

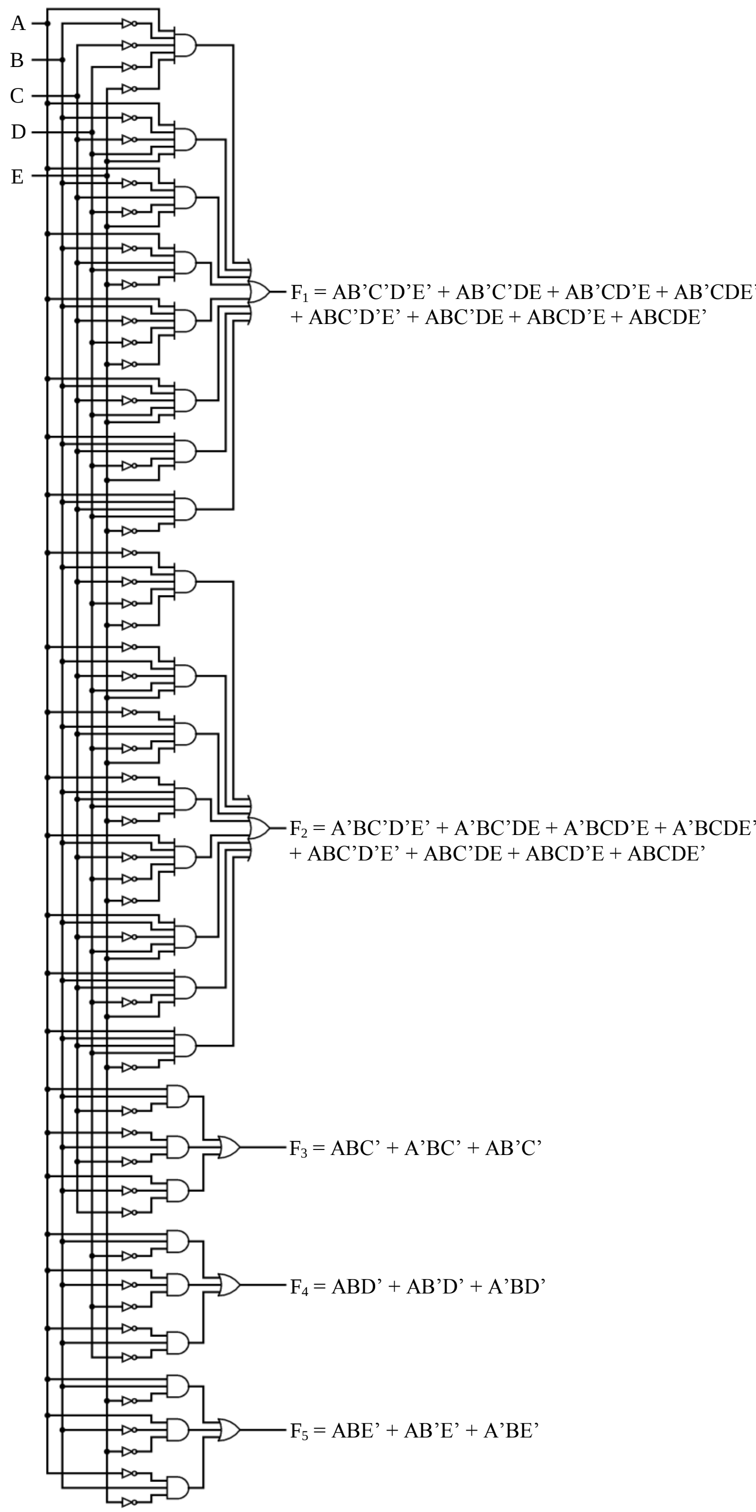


Figura 3.14. Circuito combinacional que controla el llenado de dos tanques y sistema de riego abastecidos con dos cisternas independientes.

CAPITULO 4

REDUCCIÓN DE CIRCUITOS COMBINACIONALES

4.1 Reducción del circuito combinacional para la apertura y cierre de un portón deslizante sobre riel por algebra de Boole

Se ha llevado un proceso en este trabajo de tesis para llevar de la mano al lector, de tal forma que ahora llegamos al capitulo en donde fusionamos algunos de los aspectos mas importantes abordados en el mismo. Estos aspectos que se fusiona y componen nuestro capitulo 4 son; el diseño y la reducción de los circuitos combinacionales, en especifico de los que se diseñaron en el capítulo 3.

Iniciaremos con la reducción del circuito combinacional planteado en el subtema 3.3 que es un circuito combinacional que se diseño para controlar la apertura y cierre de un portón que se desliza sobre un riel. Esta reducción como lo dice por supuesto el titulo de nuestro tema la realizaremos por medio del algebra booleana, herramienta que vislumbramos en el subtema 2.2 y que será el momento de aplicar.

El circuito combinacional para controlar la apertura y cierre del portón sobre riel arrojo como resultado dos funciones de salida para manejar dicha necesidad, la cuales se tendrán que reducir en caso de que exista dicha reducción y formar un nuevo diseño del circuito con las nuevas funciones, para así de esta manera observar si realmente se redujeron la cantidad de compuertas a utilizar o no. Las funciones del circuito son:

$F_1 = AB'CD'F + A'BCD'F + ABCD'F$

$F_2 = AB'C'E'F + A'BC'E'F + ABC'E'F$

Empezamos por simplificar a F_1, es decir tratar de que la función tenga el menor número de literales, para obligar así a que haya una mejor posibilidad de utilizar una menor cantidad de compuertas.

Observamos de F_1 que existen dos términos muy semejantes que solo varían en una literal, estos términos son:

A'BCD'F y ABCD'F

Observamos varía la literal A y aquí aplicaríamos el postulado 4; $X \bullet (Y + Z) = X \bullet Y + X \bullet Z$ que se encuentra en la tabla 2.1 quedando:

$F_1 = AB'CD'F + (A' + A)BCD'F$

Posteriormente aplicamos el postulado 5; $X + X' = 1$ y la función estaría como:

$F_1 = AB'CD'F + (1)BCD'F$

Luego aplicamos el postulado 2 dual; $X \bullet 1 = X$ al termino (1)BCD'F por lo cual tendríamos:

$F_1 = AB'CD'F + BCD'F$

Nuevamente aplicamos el postulado 4; $X \bullet (Y + Z) = X \bullet Y + X \bullet Z$ a los dos términos que han quedado, dejando fuera como un elemento en común a las literales CD'F:

$F_1 = (AB' + B)CD'F$

Trabajando ahora solo el termino (AB' + B) y aplicando el postulado 4 dual; X +

$AB' + B = (B + A)(B + B')$

Así la función que teníamos de dos términos la tendríamos:

$F_1 = [(B + A)(B + B')] \, CD'F$

Aplicamos el postulado 5; $X + X' = 1$ a $(B + B')$ daría:

$F_1 = [(B + A)(1)] \, CD'F$

Y luego aplicaríamos el postulado 2 dual; $X \bullet 1 = X$ al termino $(B + A)(1)$ quedando así solamente $(B + A)$ y:

$F_1 = (B + A)CD'F$

Expandiendo la función con postulado 4; $X \bullet (Y + Z) = X \bullet Y + X \bullet Z$ finalmente quedaría:

$F_1 = BCD'F + ACD'F$

Esta última función con respecto a la inicial tiene un término menos y menos literales, por el momento la expresión booleana ha quedado reducida, pero aun falta la expresión F_2 la cual se intentara reducir a continuación.

Para la función $F_2 = AB'C'E'F + A'BC'E'F + ABC'E'F$ el proceso es similar, buscamos que términos son en común dentro de ella y también encontramos que dos de los términos son muy parecidos que serian $A'BC'E'F$ y $ABC'E'F$, de donde observamos que solo cambia la literal A, aplicaríamos el postulado 4; $X \bullet (Y + Z) = X \bullet Y + X \bullet Z$ para dejar a las literales $BC'E'F$ como un elemento en común quedando:

$F_2 = AB'C'E'F + (A' + A)BC'E'F$

A esta última función le aplicamos el postulado 5; $X + X' = 1$ para que quede:

$F_2 = AB'C'E'F + (1)BC'E'F$

Para eliminar el uno que ha quedado dentro de la función en $(1)BC'E'F$ utilizamos el postulado 2 dual; $X \bullet 1 = X$ para que se vea:

$F_2 = AB'C'E'F + BC'E'F$

Nuevamente aplicamos el postulado 4; $X \bullet (Y + Z) = X \bullet Y + X \bullet Z$ a los dos términos que han quedado, dejando fuera como un elemento en común a las literales $C'E'F$ y se pueda observar de la siguiente manera:

$F_2 = (AB' + B)C'E'F$

Ahora trabajando solo el termino $AB' + B$ y utilizando el postulado 4 dual; $X + (Y \bullet Z) = (X + Y) \bullet (X + Z)$ el termino tomaría la forma:

$AB' + B = (A + B)(B' + B)$

Y entonces la función F_2 seria:

$F_2 = [(A + B)(B' + B)]C'E'F$

Utilizaríamos postulado 5; $X + X' = 1$ aplicado a $B' + B$ y observamos como:

$F_2 = [(A + B)(1)]C'E'F$

Otra vez aplicamos el postulado 2 dual; $X \bullet 1 = X$ dando:

$F_2 = (A + B)C'E'F$

Expandiendo F_2 con el postulado 4; $X \bullet (Y + Z) = X \bullet Y + X \bullet Z$ la función queda finalmente como:

$F_2 = AC'E'F + BC'E'F$

De esta última función F_2 y comparándola con la función inicial F_2 antes de la reducción observamos que esta ya solo cuenta con dos términos y no tres, además de que en los dos términos restantes se tienen menos cantidad de literales.

Así pues las nuevas dos funciones de salida del circuito combinacional ya con reducción a través del algebra booleana son:

$F_1 = BCD'F + ACD'F$
$F_2 = AC'E'F + BC'E'F$

El circuito eléctrico que formaríamos ahora con estas dos nuevas expresiones se muestra en la figura 4.1.

Si realizamos la comparación de la figura 3.10 con la figura 4.1, sabemos que ambas son de la misma necesidad de tarea que se requiere atender y de esa misma comparación vemos que la reducción ha sido fructífera a través de este método (algebra booleana) ya que obtenemos un circuito combinacional con menor número de compuertas y menor cantidad de literales entrando a cada una de ellas, esto genera que las operaciones que se realizan también son menores, dando no solo un circuito más viable en costos si no también en rapidez de funcionamiento al momento de realizar determinada tarea.

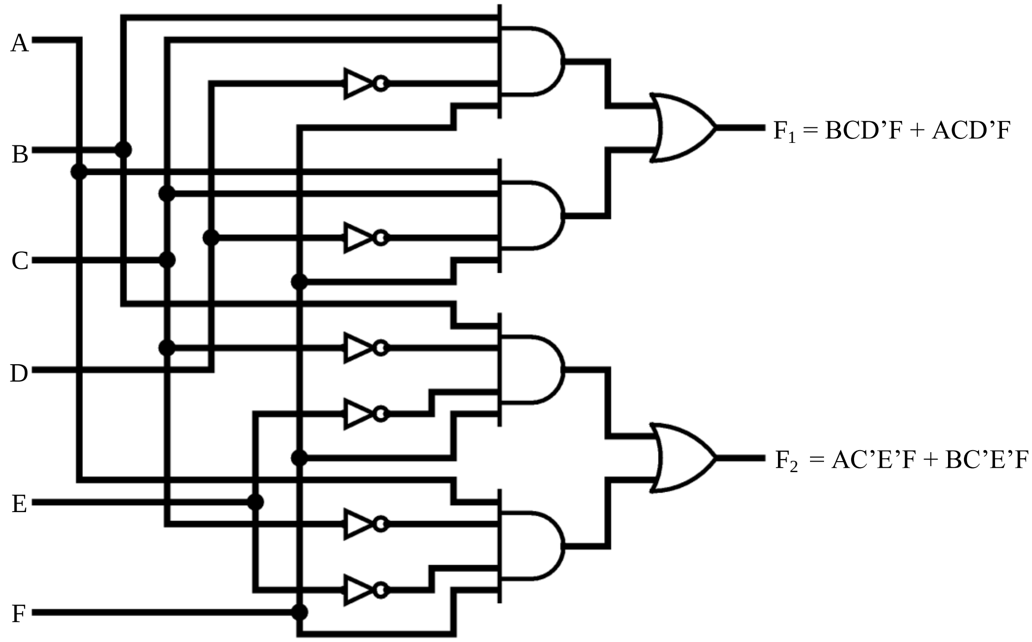


Figura 4.1. Circuito combinacional con reducción para el control de apertura y cierre de un portón sobre riel.

4.2 Reducción del circuito combinacional para activación de timbre escolar por mapas de Karnaugh.

Ahora toca para realizar otra simplificación de circuitos combinacionales aplicar la técnica de reducción por mapas de Karnaugh, es una de las más conocidas, mencionadas y usadas, aunque ya en el capítulo 2 se comenta por que no es la más viable a fin de cuentas, pero en este caso aplicaremos la técnica para hacer evidente que cualquiera de las que mencionamos puede ser usada y más cuando la necesidad de nuestra tarea genero una función con cinco variables.

Del subtema 3.4 conocemos a la función de salida que se obtuvo al momento de diseñar el circuito combinacional para controlar la activación de un timbre escolar, la cual fue:

F = AC’ + A’BC’ + A’CD + A’CE + A’C’E

Para iniciar recapitularemos que la función que vayamos a reducir se tiene que introducir en un formato de mapa de Karnaugh, en el caso de la función para controlar la activación del timbre escolar este será de cinco variables ya que esa es la cantidad de variables de entrada que se manejan en el, este formato será como el que se muestra en la figura 2.36 del capítulo 2. Para nosotros este formato se vería como se muestra en la figura 4.3 mapa en el que ya se encuentra introducida la función.

Observamos que en el mapa de karnaugh en la figura 4.3 los términos se pueden introducir fácilmente ya que cada cuadro tiene un valor que indica que minitermino debe ir ahí, pero como ya se ha comentado en el punto 2.4.2 introducción de variables, no siempre es así, ya que se pueden introducir los términos de las funciones si estos se encuentran con todas o algunas de sus literales. Ya que cada cuadro mayor (el que contiene a todos cuadrados pequeños) tiene en sus orillas las variables que contienen y que valor toman, y aunque nuestros términos no tienen todas las variables estos se colocan en el mapa, ubicando números unos en los cuadros que contengan nuestras variables las que si existen, ya que las variables que no estén pueden tomar valores de 1 o 0. Por ejemplo, se introduce solo el termino AC’ de toda la función

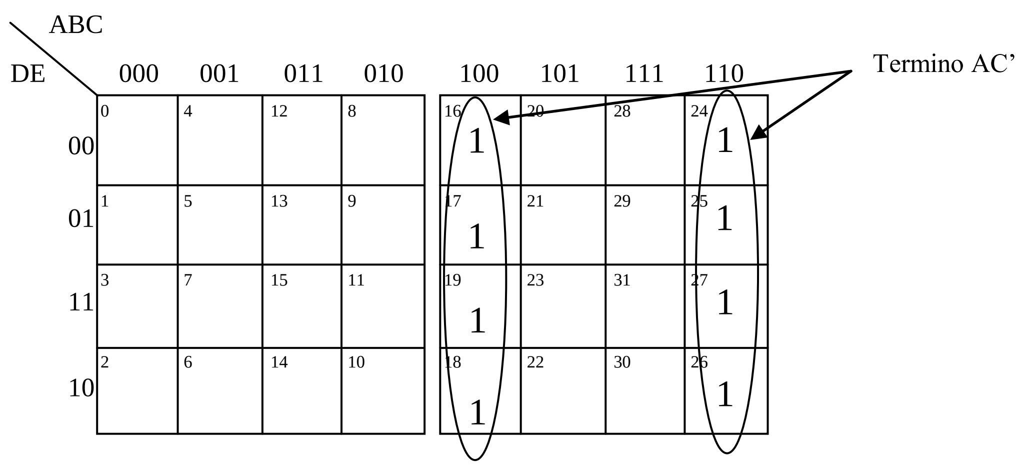


Figura 4.2. Mapa de Karnaugh de cinco variables en el que se introduce solo el termino AC’.

De la figura 4.2 observamos como en la parte de arriba de la selección de unos por medio de los óvalos los valores son 110 y 100 que representan las variables ABC’ y AB’C’ respectivamente, que hacen coincidir el valor de la variable A que es igual a 1 y C’ que su valor es 0 en ambos casos, pero como no tenemos el valor de B ya que este puede ser como en el caso del valor 110 igual a 1, también puede ser su valor igual a 0 en el valor 100, por eso se seleccionan ambos, y toda la columna hacia abajo es seleccionada por que tampoco tenemos los valores de las variables D y E que podrían ser unos o ceros. A continuación se muestra un pequeño ejemplo de toda la función que quedaría como se muestra en la figura 4.3.

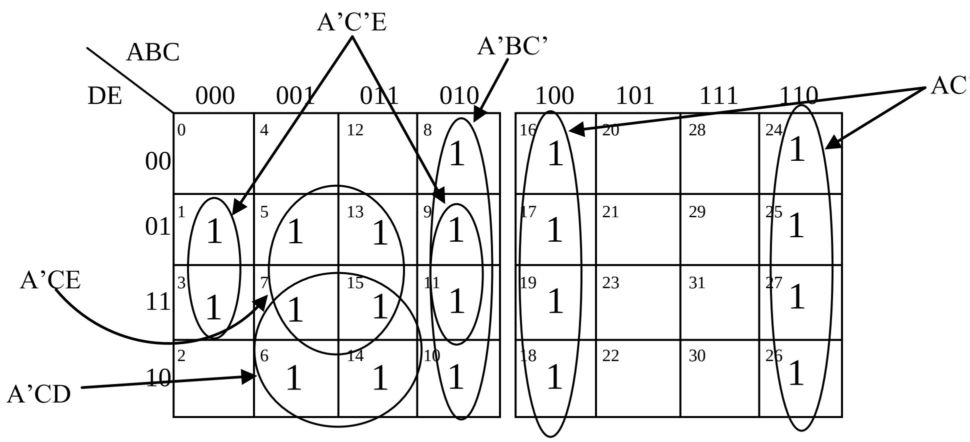


Figura 4.3. Mapa de Karnaugh que contiene a F = AC’ + A’BC’ + A’CD + A’CE + A’C’E.

Si es necesario aclarar que de la figura 4.3 no se han seleccionado unos para agrupar y simplificar, si no que se están haciendo los señalamientos de cómo están inscritos los términos de la función en el mapa, ya en la figura 4.4 se muestra como se seleccionan los unos para agruparlos y así eliminar las más variables posibles.

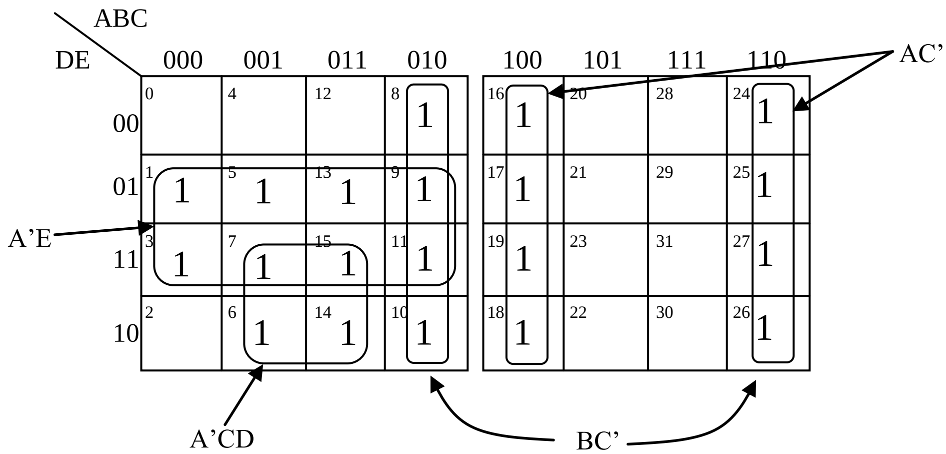


Figura 4.4. Mapa de Karnaugh que muestra la selección de unos para la simplificación de variables en la función F = AC’ + A’BC’ + A’CD + A’CE + A’C’E.

En la figura 4.4 vemos como el termino AC’ en el momento de agruparlo no hubo termino o términos con quien mas agruparlos de tal manera que pudiera eliminarse una variable mas, cabe recordar que cuando se agrupan dos unos solo se elimina una variable, cuando se agrupan cuatro unos se eliminan dos variables, cuando se agrupan ocho unos las variables que se eliminan son tres y así sucesivamente. Por lo que si observamos al momento de no poderse agrupar mas unos en el termino AC’ si notamos que en realidad desde un inicio hay ocho unos agrupados por lo cual el termino solo contiene a sus dos literales. Ahora si bien esta agrupación de unos no simplifico al mismo término, si nos ayuda para simplificar algunos otros, como lo son los señalados en la figura 4.4 para que así quede el termino BC’, otra agrupación de ochos que se observa en la misma figura 4.4 en el cuadro mayor de la izquierda nos arroja solo dos literales que son A’E y por ultimo nos queda el termino A’CD con cuatro unos agrupados, dándonos como función simplificada a:

F = AC’ + BC’ + A’CD + A’E

Si hacemos la comparación con la función sin simplificación que es F = AC' + A'BC' + A'CD + A'CE + A'C'E, nos damos cuenta que de cinco términos que había solo quedaron cuatro y también el numero de literales redujeron, ya que en la función primera había solo un termino con dos literales y los restantes cuatro tenían tres literales, mientras que ahora solo quedan tres términos con dos literales y uno solo con tres literales. Pero será mas grafico observar si resulto la reducción por este método al momento de pasar la función simplificada a compuertas lógicas, es decir observando el circuito combinacional de la siguiente figura 4.5.

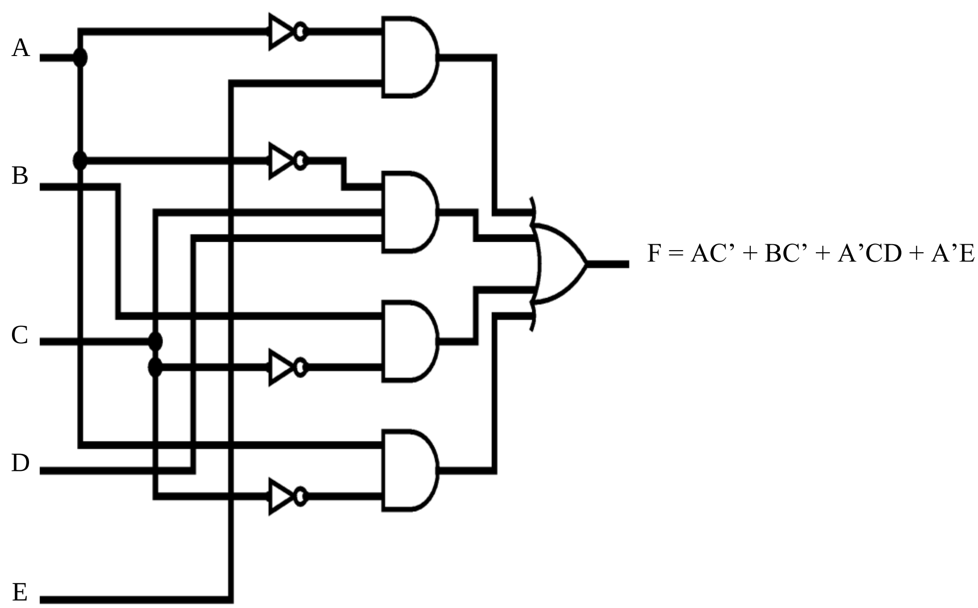


Figura 4.5. Circuito combinacional reducido para la activación de un timbre escolar.

De la comparación de la figura 3.12 del capítulo anterior con la figura 4.5 y entendiendo que ambos son circuitos que atienden la necesidad de activar el timbre escolar planteado en el punto 3.4 podemos concluir que si existe una reducción del circuito combinacional, ya que del circuito mostrado en la figura 3.12 que consta de 7 inversores, 5 compuertas AND y una compuerta OR, en la figura 4.5 se muestra ya una reducción, contando ahora con solo 4 inversores, 4 compuertas AND y en común la compuerta OR. También se puede observar como las entradas de las compuertas AND son ahora menos, la mayoría con solo dos entradas y la compuerta OR reduce una entrada, por lo cual podemos afirmar que en este circuito (figura 4.5) que realizara la misma tarea que el anterior (figura 3.12) se producen menos operaciones, por lo que daría más calidad al circuito combinacional, encontrando más favorable la reducción.

4.3 Reducción del circuito combinacional para control simultaneo del llenado de 2 depósitos elevados independientes de agua y sistema de riego, con alimentación de Dos Cisternas por Quine – Mc. Cluskey.

Como tercera propuesta de reducción se tiene al circuito que controla el llenado de dos depósitos elevados independientes de agua y también el control de un sistema de riego, que son alimentados con agua por dos cisternas, circuito manejado ya en el capítulo 3 en el punto 3.5, en el cual se manejan cinco entradas y cinco salidas; para lo cual se va aplicar una simplificación a cada una de las funciones de salida utilizando la tecnica de Quine – Mc. Cluskey (método visto en el capítulo 2), y después se va a constituir nuevamente el circuito para ver si existe una verdadera reducción al final del proceso.

Así pues empezaremos por la simplificación de la primera de las cinco funciones de salida del circuito, que será F_1 = AB’C’D’E’ + AB’C’DE + AB’CD’E + AB’CDE’ + ABC’D’E’ + ABC’DE + ABCD’E + ABCDE’, y se irán recordando los pasos a seguir.

Como primer paso tendremos que convertir todos los términos que conforman a la función de salida a su representación binaria, para así manejarse como miniterminos, y seguir el primer paso de la técnica de Quine –Mc. Cluskey tratada en el punto 2.6.1 de nuestro segundo capitulo, para así enlistar dichos miniterminos en una tabla y posteriormente hacer la separación en grupos de miniterminos que se diferencien por un solo bit o variable; para localizar mas fácilmente los miniterminos adyacentes.

La función F_1 en su representación binaria quedaría como: F_1 = 10000 + 10011 + 10101 + 10110 + 11000 + 11011 + 11101 + 11110, y acomodando los términos en su expresión binaria en una tabla por cantidad de unos quedarían como se muestran en la tabla 4.1.

En segundo paso debemos realizar la búsqueda minuciosa de los términos adyacentes entre grupos vecinos crear la nueva columna que contenga los miniterminos con adyacencia o implicantes, hay que recordar que debemos señalar con una "√" cada minitermino incluido en la formación de los nuevos miniterminos. También cabe recordar que debemos colocar un guion en los términos de la siguiente columna en la posición de la variable que se haya eliminado. Este proceso se repetirá para todas las columnas que se vayan formando y terminara cuando no haya mas implicantes que se puedan unir.

Miniterminos	ABCDE	
16	10000	Grupo 1 términos que contiene solo un uno
24	11000	Grupo 2 términos que contiene dos unos
19	10011	
21	10101	Grupo 3 términos de tres unos
22	10110	
27	11011	
29	11101	Grupo 4 términos de cuatro unos
30	11110	

Tabla 4.1. Acomodo de términos de la función de salida F_1 en su forma binaria y en grupos según la cantidad de unos.

Para nuestro caso este segundo paso aplicado a nuestra función F_1 nos dará una tabla como la 4.2, donde podemos observar que del primer renglón de la segunda columna se tiene un termino con un solo uno, mientras que en los siguientes renglones se tienen términos ya con tres unos, por lo que no se encontraran mas adyacencias y ahí terminaría la búsqueda de los implicantes.

Miniterminos	ABCDE		Miniterminos	ABCDE	
16	10000	√	16, 24	1_000	PI_0
24	11000	√	19, 27	1_011	PI_1
19	10011	√	21, 29	1_101	PI_2
21	10101	√	22, 30	1_110	PI_3
22	10110	√			
27	11011	√			
29	11101	√			
30	11110	√			

Tabla 4.2. Manejo de los términos de F_1 para encontrar los primeros implicados (Pi_n).

Después de terminada nuestra tabla 4.2 utilizaremos una herramienta para verificar errores, es una prueba que se realiza a los miniterminos utilizados en la búsqueda de adyacencias, por ejemplo de la segunda columna en el primer renglón de la tabla 4.2 vemos que se utilizaron los miniterminos 16 y 24 con los cuales se realiza una resta, al mayor restándole el menor que seria 24 – 16 = 8 encontrando que el 8 es el valor de la posición en que estaba colocada la variable que se elimino, si la combinación arrojo 1_000 vemos claramente que la posición con un guion era la del valor 8 es decir:

1 _ 0 0 0

16 8 4 2 1

Si en las demás restas con los miniterminos de cada implicante pasa lo que acabamos de ver quiere decir que no ha habido errores, en caso contrario en que la resta arroje un resultado que no coincida con la posición con guion entonces habrá que revisar el proceso hasta este punto.

En las demás restas como 27 – 19 = 8, 29 – 21 = 8 y 30 – 22 = 8 observamos que en todas el resultado es el numero 8 y que coincide con el valor de la posición en la que se encuentra el guion en todos esos implicantes, observe la tabla 4.2.

Una vez encontrados los primeros implicantes como tercer paso se procede a construir una nueva tabla con los primeros implicantes de forma vertical y de forma horizontal los miniterminos, así colocaremos una "X" cuando un implicante en un renglón cubra un mintermino que se encuentre en una columna. Así quedara la tabla 4.3 para la función que estamos simplificando.

		√	√	√	√	√	√	√	√
		16	19	21	22	24	27	29	30
**	PI_0	X				X			
**	PI_1		X				X		
**	PI_2			X				X	
**	PI_3				X				X

Tabla 4.3. Selección de los implicados esenciales en la función F_1.

De esta tabla 4.3 podremos continuar con un cuarto paso que es ya obtener los implicados o implicantes esenciales, que son los señalados con doble asterisco al lado izquierdo de cada uno de ellos, estos son elegidos una vez que se observa que estos sean el numero minimo de implicantes que cubren a los miniterminos de la función.

Vemos de la tabla 4.3 que todos los primeros implicantes que se introdujeron son esenciales, por lo que se mantienen igual y la función ahora quedaría como:

$F_1 = PI_0 + PI_1 + PI_2 + PI_3$

Luego sustituimos cada implicante esencial por las variables que lo componen observando la tabla 4.2 y nos da la función simplificada final:

$F_1 = AC'D'E' + AC'DE + ACD'E + ACDE'$

Con este mismo proceso en que simplificamos a la función F_1 lo haremos para las funciones F_2, F_3, F_4 y F_5.

Iniciamos con el primer paso la simplificación de la función F_2 es decir convertirla en su representación binaria y acomodar cada termino en una tabla donde se acomodaran estos según la cantidad de unos que contengan.

Si $F_2 = A'BC'D'E' + A'BC'DE + A'BCD'E + A'BCDE' + ABC'D'E' + AB\ C'DE + ABCD'E + ABCDE'$, entonces la función en binario quedaría como:

$F_2 = 01000 + 01011 + 01101 + 01110 + 11000 + 11011 + 11101 + 11110$

Acomodando en tabla y por grupos quedaría como en la tabla 4.4 que a continuación se muestra.

Miniterminos	ABCDE	
8	01000	Grupo 1 que contiene solo un uno
24	11000	Grupo 2 que contiene dos unos
11	01011	
13	01101	Grupo 3 de tres unos
14	01110	
27	11011	
29	11101	Grupo 4 de cuatro unos
30	11110	

Tabla 4.4. Acomodo de términos de la función de salida F_2 en su forma binaria y en grupos según la cantidad de unos.

El segundo paso en la técnica de Quine – Mc. Cluskey en el trabajo de simplificación se hace a través la búsqueda minuciosa de los términos adyacentes entre grupos vecinos, crear la nueva o nuevas columna que contenga los miniterminos con adyacencia o implicantes, y señalar con una "√" los miniterminos utilizados, para que nos quede una tabla como la 4.5.

Miniterminos	ABCDE		Miniterminos	ABCDE	
8	01000	√	8, 24	_1000	PI_0
24	11000	√	11, 37	_1011	PI_1
11	01011	√	13, 29	_1101	PI_2
13	01101	√	14, 30	_1110	PI_3
14	01110	√			
27	11011	√			
29	11101	√			
30	11110	√			

Tabla 4.5. Manejo de los términos de F_2 para encontrar los primeros implicados (Pi_n).

Como habíamos comentado después del punto dos en la simplificación de la función anterior también aquí utilizaremos la pequeña herramienta de restar entre los miniterminos que intervinieron para obtener a cada implicante, observando en la tabla

4.5 que en el caso de todos los implicantes obtenidos, la posición que tiene el guion ya que se ha eliminado una variable es la del valor 16, así que las restas deberán dar como resultado ese valor. A continuación mostramos las restas.

24 – 8 = 16
27 – 11 = 16
29 – 13 = 16
30 – 14 = 16

De lo observado en las operaciones de sustracción tenemos que hasta aquí se lleva bien el proceso, de tal forma que iniciamos con el tercer paso que es; construir una nueva tabla con los implicantes que se obtuvieron en la tabla 4.5, para así saber cuáles son los implicantes esenciales que cubren a todos los miniterminos de la función de salida F_2, esta tabla es la 4.6.

En la tabla 4.6 al igual que en la 4.3 de la función F_1 se ve que todos los primeros implicados son esenciales, ya que cada uno de ellos cubre un minitermino de la función F_2 por lo que no habría mas reducción para esta función.

		√	√	√	√	√	√	√	√
		8	11	13	14	24	27	29	30
**	PI_0	X				X			
**	PI_1		X				X		
**	PI_2			X				X	
**	PI_3				X				X

Tabla 4.6. Selección de los implicados esenciales en la función F_2.

El cuarto paso es seleccionar los implicantes esenciales, y formar nuestra nueva función con ellos. Como vemos de la tabla 4.6 todos los implicantes con doble asterisco a la izquierda son esenciales y los colocaremos en la función de la siguiente manera:

$F_2 = PI_0 + PI_1 + PI_2 + PI_3$

Posteriormente sustituimos cada implicante esencial por las variables que lo componen observando la tabla 4.5 y obtendríamos la función simplificada final:

$F_2 = BC'D'E' + BC'DE + BCD'E + BCDE'$

Para las simplificaciones posteriores de las funciones F_3, F_4 y F_5, reduciremos un poco la explicación entendiendo que es el mismo proceso de las dos funciones anteriores, con la única diferencia de que manejaremos los términos con la cantidad de variables que contienen los términos.

Iniciamos la simplificación de la función $F_3 = ABC' + A'BC' + AB'C'$, que aunque sabemos que los términos que están en esta función son con mas variables solo manejaremos dicha simplificación con solo tres variables, así en las siguientes funciones.

Primer paso; pasamos la función a su representación binaria es decir quedaría como:
$F_3 = 110 + 010 + 100$
Para acomodarla en una tabla en donde agruparemos por cantidad de unos, vemos la tabla 4.7 donde ya se colocan los términos.

Miniterminos	ABC	
2	010	Grupo 1 contienen solo un uno
4	100	
6	110	Grupo 2 de dos unos

Tabla 4.7. Acomodo de términos de la función F_3 por cantidad de unos.

Como paso dos se realiza la búsqueda minuciosa de las adyacencias entre los grupos para eliminar variables, quedando la tabla 4.8 donde observamos los primeros implicantes.

Miniterminos	ABC		Miniterminos	ABC	
2	010	√	2, 6	_10	PI_0
4	100	√	4, 6	1_0	PI_1
6	110	√			

Tabla 4.8. Manejo de los términos de F_3 para encontrar los primeros implicados (Pi_n).

En este caso se encuentran dos primeros implicantes, realizamos las restas en busca de errores las cuales son; 6 – 2 = 4, valor que equivale al guion que se ocupa en el PI_0, así que esta bien, y 6 – 4 = 2, valor que también equivale al guion que se ocupa en el PI_1, no localizando errores a continuación realizamos nuestro tercer paso; el de buscar los implicantes esenciales. En nuestra tabla 4.9 se acomodan tanto los primeros implicantes como los miniterminos que cubren estos.

		√	√	√
		2	4	6
**	PI_0	X		X
**	PI_1		X	X

Tabla 4.9. Selección de los implicados esenciales en la función F_3.

Aplicamos el cuarto paso una vez que observamos que los dos implicantes PI_0 y PI_1 son esenciales teniendo ahora:

$F_3 = PI_0 + PI_1$

La misma función pero ya con las variables que corresponde a PI_0 y PI_1 nos queda la función final:

$F_3 = BC' + AC'$

Ahora nuevamente empezamos con el primer paso para la función $F_4 = ABD' + AB'D' + A'BD'$, que el manejo es el mismo ya que tenemos la misma cantidad de variables, aunque difiera en una variable con la función F_3. La función en su representación binaria será: $F_4 = 110 + 100 + 010$. En la tabla 4.10 vemos su acomodo por grupos de unos para su posterior análisis de adyacencia.

Miniterminos	ABD	
2	010	Grupo 1 contienen solo un uno
4	100	
6	110	Grupo 2 de dos unos

Tabla 4.10. Acomodo de términos de la función F_4 por cantidad de unos.

Nótese como en la tabla 4.10 ahora en lugar de la variable C se coloca la variable D, que es en lo único que cambia esta función con la anterior, pero como ya comentamos el manejo es el mismo. El paso 2 será buscar las adyacencias para encontrar a los primeros implicados como se muestra en la tabla 4.11.

Miniterminos	ABD		Miniterminos	ABD	
2	010	√	2, 6	_10	PI_0
4	100	√	4, 6	1_0	PI_1
6	110	√			

Tabla 4.11. Manejo de los términos de F_4 para encontrar los primeros implicados (Pi_n).

Seguimos viendo que el proceso es casi el mismo y al realizar las restas para checar errores, las restas resultan iguales a la función anterior; 6 – 2 = 4 y 6 – 4 = 2 y checamos de la tabla 4.11 que corresponden a los valores de donde están colocados los guiones y continuamos con el tercer paso que es buscar los implicantes esenciales, para eso usamos la tabla 4.12.

También por la similitud que tenemos con la función anterior los implicantes esenciales son PI_0 y PI_1 así nuestra función F_4 queda:

$F_4 = PI_0 + PI_1$

Checando las variables que tienen ambos implicantes esenciales de la tabla 4.11 tenemos que la función finalmente quedaría como:

$F_4 = BD' + AD'$

		√	√	√
		2	4	6
**	PI_0	X		X
**	PI_1		X	X

Tabla 4.12. Selección de los implicados esenciales en la función F_4.

Por último para $F_5 = ABE' + AB'E' + A'BE'$ su representación binaria es $F_5 = 110 + 100 + 010$ y se coloca en la tabla 4.13 por grupos de unos.

Miniterminos	ABE	
2	010	Grupo 1 contienen solo un uno
4	100	
6	110	Grupo 2 de dos unos

Tabla 4.13. Acomodo de términos de la función F_5 por cantidad de unos.

Igual que en los casos anterior si observamos la tabla 4.13 vemos que cambiamos la variable que antes era C en la función F_3 y D en la función F_4 por la variable E en nuestra función F_5. Procedemos como segundo paso a la búsqueda de los primeros implicantes y lo plasmamos en la tabla 4.14.

Miniterminos	ABE	
2	010	√
4	100	√
6	110	√

Miniterminos	ABE	
2, 6	_10	PI_0
4, 6	1_0	PI_1

Tabla 4.14. Manejo de los términos de F_5 para encontrar los primeros implicados (Pi_n).

Como paso tres buscamos los implicantes esenciales a través de nuestra tabla 4.15 en donde ya se han colocado los primeros implicantes para así ver cuáles son los esenciales, aunque sabemos que no habrá sorpresas ya que la función es muy parecida a las anteriores.

		√	√	√
		2	4	6
**	PI_0	X		X
**	PI_1		X	X

Tabla 4.15. Selección de los implicados esenciales en la función F_5.

Por cuarto y último paso, nos dirigimos a poner la función resultante en función de PI_0 y PI_1 como se muestra a continuación:

$F_5 = PI_0 + PI_1$

Y pasamos la función a su forma en la que contiene a sus variables según los implicantes de la tabla 4.14 quedando:

F_5 = BE’ + AE’

De esta forma terminamos de utilizar la técnica de Quine – Mc. Cluskey para simplificar la función que representa al circuito combinacional en cuestión, pero ahora tendremos que constituir nuevamente dicho circuito para saber si ha habido una reducción en el. Para ir vislumbrando un poco mejor si se ha realizado algún cambio en tamaño del circuito no simplificado en comparación del simplificado y para ayudarnos a armar este ultimo colocaremos una tabla donde compararemos las funciones antes de la reducción y después de ella, ya que en el caso de este sistema combinacional tenemos un mayor numero de funciones de salida, y nos referimos a la tabla 4.16.

Funciones sin simplificar	**Funciones simplificadas**
F1 = AB’C’D’E’ + AB’C’DE + AB’CD’E + AB’CDE’ + ABC’D’E’ + ABC’DE + ABCD’E + ABCDE’	F1 = AC’D’E’ + AC’DE + ACD’E + ACDE’
F2 = A’BC’D’E’ + A’BC’DE + A’BCD’E + A’BCDE’ + ABC’D’E’ + AB C’DE + ABCD’E + ABCDE’	F2 = BC’D’E’ + BC’DE + BCD’E + BCDE’
F3 = ABC’ + A’BC’ + AB’C’	F3 = BC’ + AC’
F4 = ABD’ + AB’D’ + A’BD’	F4 = BD’ + AD’
F5 = ABE’ + AB’E’ + A’BE’	F5 = BE’ + AE’

Tabla 4.16. Muestra las funciones antes de la simplificación y después de aplicarles la técnica de Quine – Mc. Cluskey.

A continuación mostraremos en la figura 4.6 el circuito combinacional que se constituyo con las nuevas funciones simplificadas.

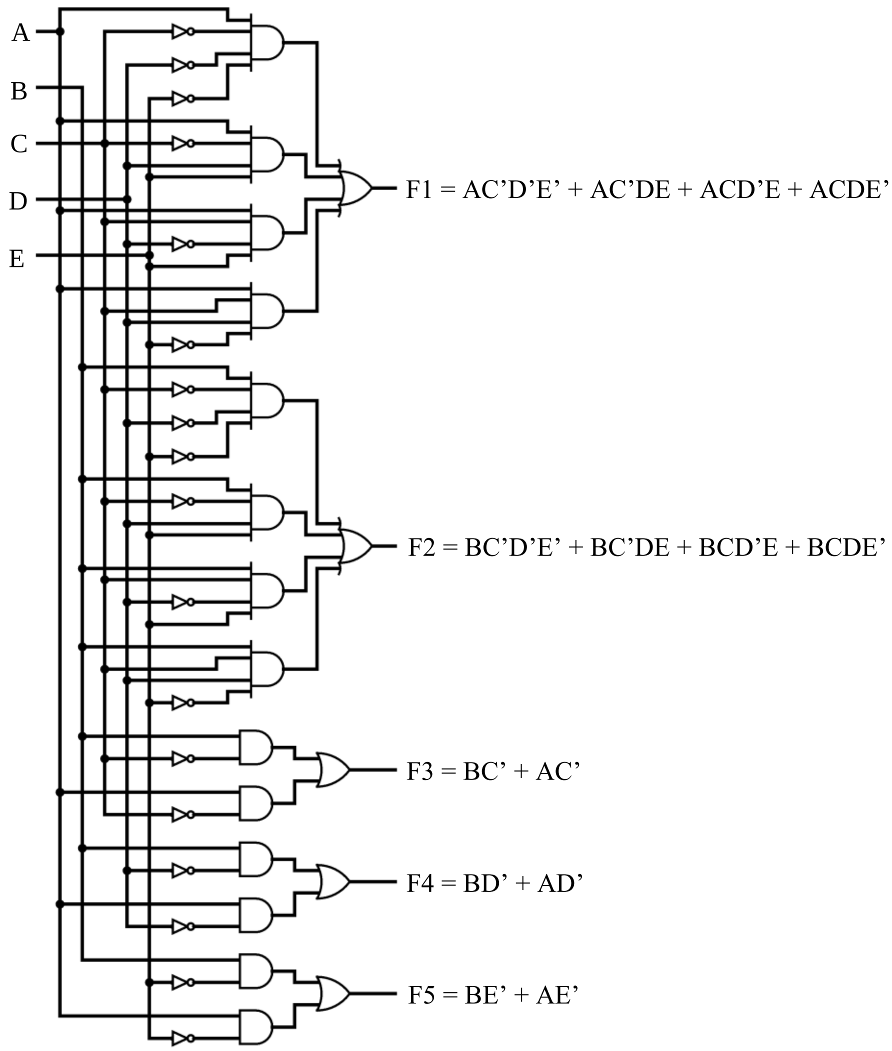


Figura 4.6. Circuito combinacional con una reducción, que controla el llenado de dos depositos elevados independientes y un sistema de riego, alimentado por dos cisternas.

CONCLUSIONES

Al terminar este trabajo se logra llegar a una serie de comentarios finales sobre su proceso de elaboración, de los primeros elementos a concluir son del capitulo 1, en donde los números binarios y sus operaciones elementales aportan claramente a través de la teoría de los circuitos combinacionales el concepto de que la componente física mínima es la compuerta, elemento básico y fundamental que opera y materializa las operaciones lógicas y como al interconectarse con otras dan forma a lo que son los llamados circuitos combinacionales, y a su vez como estos últimos cuentan con diferencias marcadas al ser comparados con los que serian los circuitos secuenciales, así esta sección es especifica acerca de que solo trataremos circuitos combinacionales desde el inicio hasta el final de todo el escrito. Este capítulo nos ayuda a entender como los binarios de ser simples elementos discretos, es decir; niveles de voltaje o magnitudes de corriente que representan datos se comienzan a agrupar y a formar códigos, estos a su vez pasan a formar parte de los sistemas digitales. Nos muestra el manejo de operaciones básicas para con los mismos números binarios a través de teoremas y postulados; pero además el apartado no solo se queda hasta ahí, si no que muestra también una faceta grafica donde se muestran los símbolos eléctricos que representan las operaciones lógicas según las distintas normas internacionales, normas que son el intento de homogeneizar su uso.

Del segundo capítulo se puede concluir que las primeras teorías creadas para la reducción son de índole matemática y que al relacionarlas con las compuertas son vistas de la misma manera, es decir; como rutas abiertas o cerradas, coincidencia importante encontrada en el desarrollo de esta investigación y que ayuda a relacionar la teoría con la práctica que es uno de los objetivos más importantes del trabajo realizado ya que es imperante para el alumno que encuentre esa relación para poder iniciar su trabajo hacia el camino del manejo del diseño. Posteriormente se llega a los mapas de Karnaugh que son técnicas que al igual que las primeras dejan mucho peso a la percepción y habilidad del diseñador para ejecutarlas, se percibe aquí claramente como si se tiene la experiencia necesaria en estos métodos se puede llegar a una reducción mínima, de no ser así se puede llegar a reducciones parciales, no equivocadas pero tampoco mínimas, Se recomienda su utilización para cantidades de señales reducidas y que definitivamente estos son métodos más del tipo gráfico. Hay técnicas que tienen una forma más algorítmica, con un proceso ya mas definido que ayuda a llegar a una única opción mínima de reducción quitando responsabilidad a la experiencia o habilidad del diseñador.

Se logra abarcar en el capitulo 3 lo que son los datos del fabricante para comprender el aspecto práctico de las compuertas, siendo estas las empleadas en el diseño digital combinacional. Se elabora con éxito una guía certera que muestra en dos etapas como se hace el diseño de algunos sistemas digitales empleando la lógica combinacional y en referencia a las etapas, es en primera instancia la necesidad de una tarea y como segunda a la conversión de esa necesidad a la expresión booleana así como a su diagrama. Aquí vemos cumplido parte de los objetivos planteados al inicio del trabajo; el poder facilitar a los alumnos el aprendizaje de la electrónica digital, ya que es sorprendente como a través del planteamiento de una idea en un enunciado este se puede convertir en un diseño lógico. Esta zona del escrito es el facilitador para el diseño, lo cual ayuda a elaborar algunos diseños digitales con éxito.

Finalmente, en el capitulo cuatro se logra dar una continuidad coherente y lógica al trabajo desarrollado al obtener diseños realmente mas simplificados y listos para ser comparados con sus diseños iniciales, y que a su vez se conjugan armónicamente con la guía de diseño de los circuitos combinacionales y en una fusión excepcional con los demás temas, desde las generalidades hasta el que se encaminen hacia las técnicas de reducción para posteriormente aplicarlas.

BIBLIOGRAFÍA

BALABANIAN Norman y CARLSON Bradley. Principios de diseño lógico digital. Ed. Cecsa. P. 55 a 58.

BARNU Porat. Introducción a la tecnología digital. Ed. Limusa. P. 1 a 50.

BOYLESTAD Robert L. y NASHELSKY Louis. Electrónica: Teoría de circuitos. Prentice Hall. P. 59 a 77.

BREEDING Kenneth J. Digital design fundamentals. Ed Prentice Hall. P. 25 a 250.

Dirección general de orientación y servicios educativos. Guía de carreras UNAM 2003 – 2004. Ed. UNAM. P. 130.

FABRICIUS Eugene D. Diseño lógico moderno y teoría de la conmutación. Ed. Cecsa. P. 47, 48, 57 a 61, 74, 75, 145 a 148, 157 a 166.

FLOY Tomas L. Digital fundamentals. Ed. Prentice Hall. P. 4.

FLOY Tomas L. Fundamentos de sistemas digitales. Ed. Prentice Hall. P. 125 a 130, 134, 139, 142 a 144, 148, 149, 271, 273 a 280, 308 a 312.

GARCIA Sanchez J. E. et al. Circuitos y sistemas digitales. Ed. Tebur flores. P. 1, 2, 3.

GRECH M. Pablo. Introducción a la ingeniería. Un enfoque a través del diseño. Ed. Prentice Hall. P. 72 a 74, 78, 80, 82, 83, 88.

LIPSCHUTZ Seymour. Teoría de conjuntos y temas afines. Ed. Mc Graw Hill. P. 104, 216, 217.

MANO Morris. Diseño digital. Ed. Prentice Hall. P. 27 a 155.

MANO Morris. Lógica digital y diseño de computadores. Ed. Prentice Hall. P. 15 a 200.

MARCOVITZ Alan B. Diseño digital. Ed. Mc Graw Hill. P. 21, 24, 25, 39.

McCLUSKEY E. J. Electrical and electronic engineering series. Ed. Mc Graw Hill. P. 132, 133, 136, 137, 140 a 146, 157 a 162.

MOTOROLA INC. Fast and ls ttl data. Ed. Motorola. P. 5 – 37, 7 – 13, 4 – 6, 4 – 8, 4 – 12, 4 – 19, 4 – 21.

NASHELSKY Louis. Fundamentos de tecnología digital. Ed. Limusa. P. 138.

NELSON P. Víctor et al. Análisis y diseño de circuitos lógicos digitales. Ed. Prentice Hall. P. 104, 111, 113, 114, 120, 123, 128, 137, 173 a 231.

PETERSON Gerald R. y Hill J. Frederick. Teoría de la conmutación y diseño lógico. Ed. Noriega. P. 65, 66, 68, 163, 164, 178, 181 a 187.

POLLAN Santamaria Tomás. Electrónica digital. Ed. Prensas universitarias de Zaragoza. P. 19, 21, 22.

ROTH H. Charles Jr. Fundamentos de diseño lógico. Ed. Thomson. P. 6, 7, 32, 33, 54, 70, 71, 76 a 78, 109 a 120, 142 a 147, 177, 178, 189.

RUIZ VASSALLO Francisco. Electrónica digital fácil. Para electricistas y técnicos de mantenimiento. Ed. Alfa omega. P. 19 a 23, 31 a 35, 38, 42, 44, 47 a 49, 79, 85, 86, 94, 96, 99, 102, 113 a 132, 145 a 163.

SHIVA Sajjan G. Introducción al diseño lógico. Ed. Trillas. P. 13, 14 20 a 30.

WAKERLY John F. Diseño digital principios y prácticas. Ed. Prentice Hall. P. 80, 82 a 85, 193, 194, 210 a 216, 233 a 235, 311, 313, 314.

WRIGHT Paul H. Introducción a la ingeniería. Ed. Limusa. P. 121 a 123, 137, 142.

SITIOS WEB

http://physics.usask.ca/~angie/ep326/LS_Dat_Book.pdf
http://karnaugh.shuriksoft.com/
http://ozark.hendrix.edu/~burch/logisim/index_es.html
http://www.sweethome3d.com
http://sourceforge.net/projects/boole-deusto/
www.ansi.org
www.din.de
www.ieee.org
www.telefunken.com
www.ami.com
www.fairchildsemi.com
www.ferranti-technologies.co.uk
www.intel.com
www.itt.com
www.freescale.com
www.ucl.ac.uk

www.national.com

www.npx.com

www.raytheon.co.uk

www.st.com

www.st.com

www.infineon.com

www.signetics.com

www.solitrondevices.com

www.ti.com

www.unisemgroup.com