DISEÑO DE CIRCUITOS A PARTIR DE EXPRESIONES BOOLEANAS

Si la operación de un circuito se define por medio de una expresión booleana, es posible construir un diagrama de circuito lógico a partir de dicha expresión. Por ejemplo, si se desea un circuito definido por la expresión x = A * B * C, a simple vista se observa que el circuito adecuado que cumple con esta expresión es una compuerta AND de tres entradas.

En otras palabras, la expresión anterior se puede interpretar como una función la cual es derivada de tres variables A, B, C, esta expresión será verdadera, será 1 o se cumplirá cuando las tres variables se cumplan o presenten el valor de 1, esto es; 1 = 1 * 1 * 1, hay que recordar que en el ámbito de la electrónica digital sólo se tienen contemplados dos valores posibles para las señales los cuales son 1 y 0.

Esta expresión no será verdadera cuando alguna de las variables presenten un estado distinto al 1. Lo anterior se puede observar en una tabla de verdad que define esta expresión tabla 1.

A	**B**	**C**	**X**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Tabla 1

En la tabla 1 se aprecian todas las posibles combinaciones de las variables y el resultado de la expresión esperado.

Si se requiere cumplir la siguiente expresión:

X = A + B’

En este caso se puede observar que la variable B presenta una señalización la cual indica que está negada, esto significa que la expresión anterior será verdadera cuando A = 1 y B = 0 el símbolo (+) indica que se trata de una operación OR.

Este caso requiere una explicación más detallada.

La compuerta OR es un circuito que representa a (+) en donde el resultado X tomará el valor de a o B tal y como se muestra el la tabla 2.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 2

Como se puede apreciar en la tabla 2 X toma los valores de A o B, sin embargo en la expresión X = A + B’, B’ indica que X= 1 cuando B esté negada o mejor dicho cuando B = 0 lo cual es lo mismo que indicar B’, en este caso de acuerdo a la tabla 2 se descartarán las demás combinaciones quedando como se indica en la tabla 3.

A	B	X
0	0	0
0	1	0
1	0	1
1	1	0

Tabla 3

La expresión anterior también se podría mostrar como: 1 = 1 + 0’

El símbolo ‘ o negación indica que a la variable B hay que negarla o “invertirla” para que se cumpla con la condición, una compuerta capaz de invertir la señal es precisamente el inversor.

El razonamiento que se aplica en estos casos sencillos es posible aplicarlo también a circuitos más complejos.

El circuito derivado del razonamiento anterior se muestra en la figura 1.

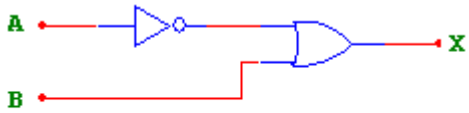


Figura 1

Suponga que se desea construir un circuito cuya salida es x = AC + BC’ + A’BC, esta expresión booleana contiene tres términos (AC, BC’, A’BC), en donde los tres componentes operan con una compuerta or de tres entradas iguales, en cada una de ellas se presentan las señales AC, BC’ y A’BC respectivamente figura 2.

Figura 2

Cada entrada de la compuerta or es un término de producto AND, lo cual significa que se puede usar una compuerta AND de dos y tres entradas y los elementos negados se obtienen mediante un inversor.

Lo anterior se muestra en la figura 3.

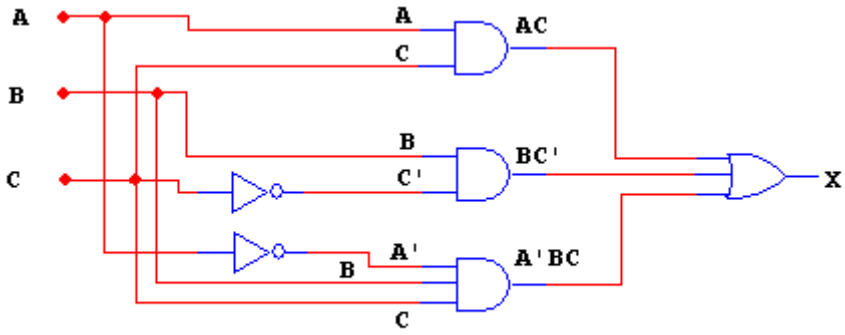


Figura 3

Este análisis general puede ser seguido siempre aunque existen técnicas más eficientes para la solución de este tipo de problemas una de ellas es la representación de las funciones lógicas mediante tablas de verdad y el uso de programas de diseño y simulación como *Electronics Workbench.*

Una forma de representar a una función lógica es mediante una tabla, la cual es llamada tabla de verdad, en la cual se indican los valores que toma la función para cada una de las combinaciones de los valores de las variables de entrada.

Esta tabla es realizada representando en la columna de la izquierda todas las posibles combinaciones de las variables de entrada como se muestra en las tablas 1, 2 y 3.

Un ejemplo más se muestra a continuación.
Dada la función X = BA + CBA' + B'A + A

Calculando el valor que toma la función para cada una de las posibles combinaciones de valores de las variables de entrada y representándolo en una tabla se obtiene la tabla 4.

C	B	A	X
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 4

El número de las posibles combinaciones de valores de las variables de entrada es 2^n, siendo n el número de variables de entrada, para este ejemplo se tienen tres variables de entrada por lo que se tiene $2^3 = 2 * 2 * 2 = 8$ combinaciones posibles.

Utilizando el dispositivo llamado convertidor lógico el cual se encuentra en el programa de simulación *work bench,* se obtiene y se comprueba la tabla de verdad de la función.

Esto se logra introduciendo la función que se tiene como ejemplo en el cuadro en blanco figura 4.

Ya que se ha introducido la función se activa la casilla de conversión de expresión algebraica a tabla de verdad, figura 5.

Se puede observar como el programa automáticamente muestra la tabla de verdad que satisface a la función.

Por el contrario si se tiene ya una tabla de verdad y se desea encontrar la función que lo representa se procede de la siguiente manera.

Como se puede observar en la figura 4 en la parte superior se encuentran las posibles señales de entrada A, B .... H al hacer clic con el Mouse en cada una de ellas siguiendo un orden ascendente A – H, el programa automáticamente generará las combinaciones de las variables de entrada como se muestra en la figura 6.

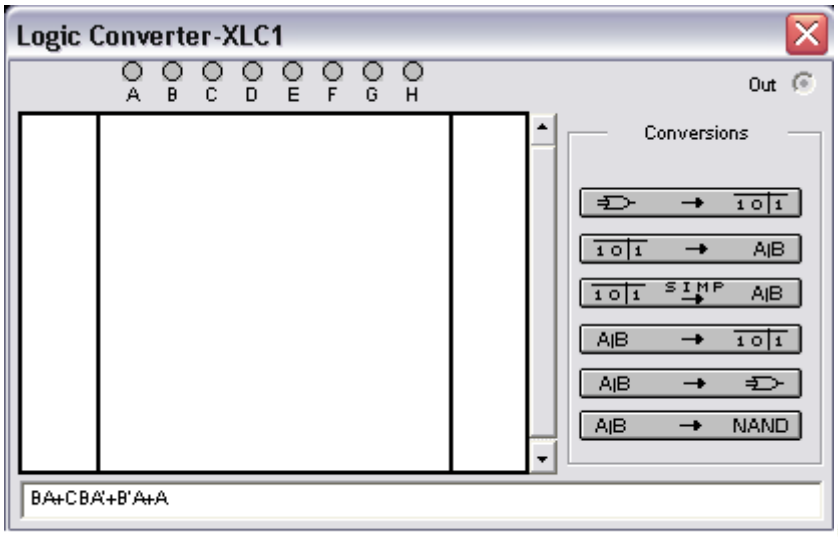
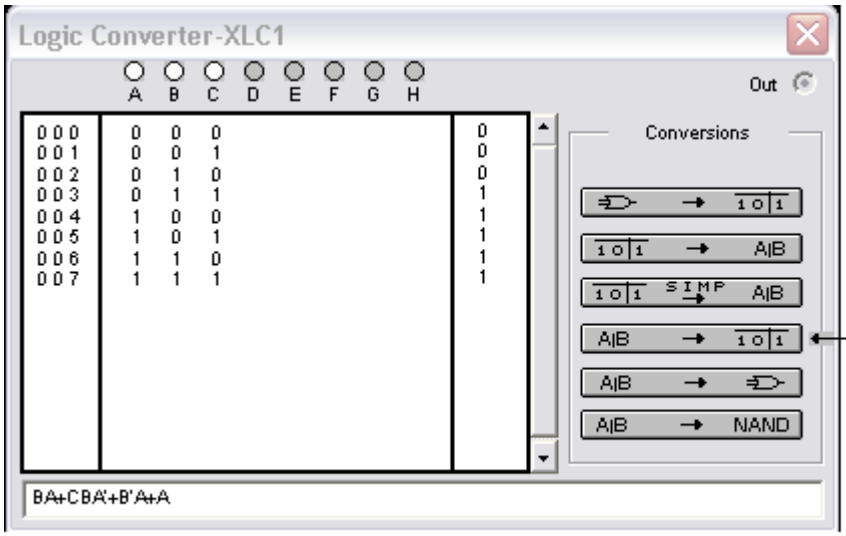


Figura 4



Conversion

Figura 5

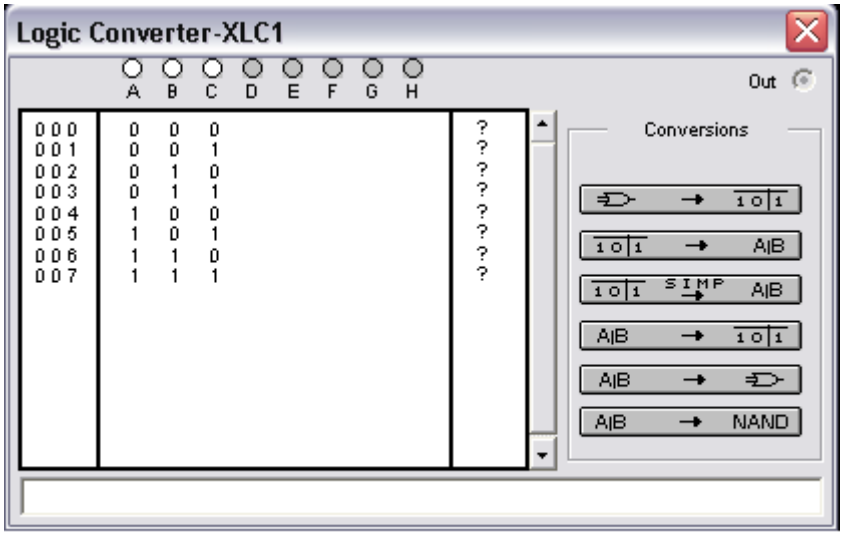


Figura 6

Se tienen tres entradas por lo que se han activado A, B, C y el programa ha generado automáticamente las combinaciones posibles, se puede observar en la columna de la derecha en la tabla generada los signos ?, esto significa que el programa está esperando que se haga clic en cada uno de estos símbolos para ingresar su valor correspondiente, de acuerdo a la tabla 4 se hace clic en los símbolos para modificar su valor y posteriormente se activa la casilla de conversión de tabla de verdad a función para que el programa muestra en el cuadro la función que satisface a la tabla generada figura 7.

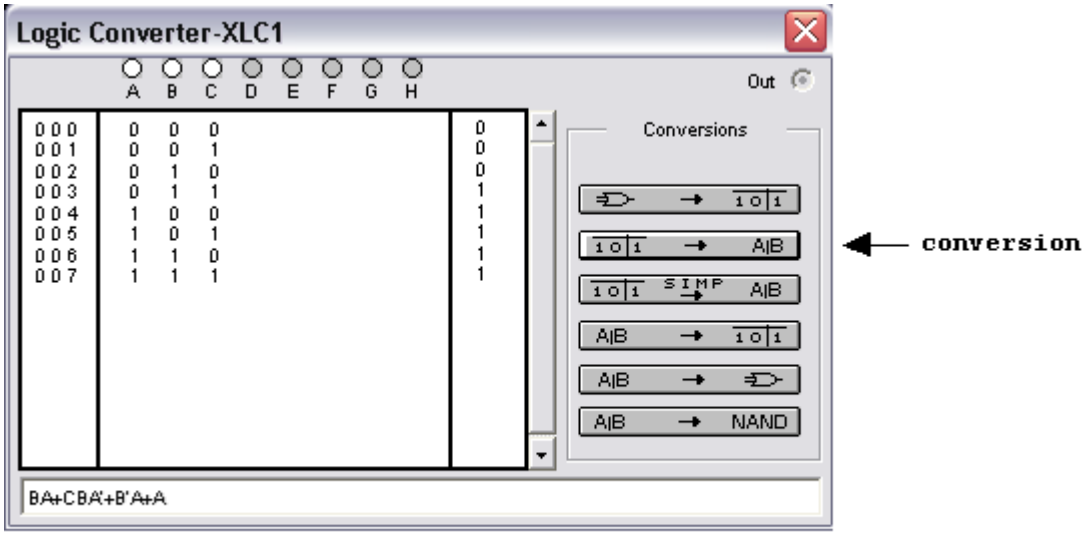


Figura 7

y el programa mostrará la función, es posible que el programa genere una función un poco diferente a la mostrada en la figura 7, esto es debido a que esta tabla, como cualquier otra, tiene varias soluciones o varias funciones "equivalentes" en donde todas pueden ser tomadas como verdaderas.

Ya que se tiene la tabla de verdad y/o la función este mismo programa puede generar el circuito que satisface a las condiciones únicamente haciendo clic en la casilla de conversión de tabla o función a circuito figura 8.

Al hacer clic en la casilla el programa automáticamente generará el circuito equivalente.

Esta aplicación también puede generar una tabla a partir del circuito, para lo cual se deberá tener el circuito como se muestra en la figura 8 y realizar clic en la casilla dedicada a esta operación la cual es la primera en el área de conversiones de la aplicación.

Representación de las funciones lógicas en su forma canónica.

Entre las múltiples expresiones algebraicas con las que se puede representar una función lógica, se mencionan dos tipos según la expresión esté formada por:

Sumas de productos como por ejemplo: X = ba + cba' + b'a + a

Productos de sumas como por ejemplo: X = (b+a)(c+b+a')(b'+a)

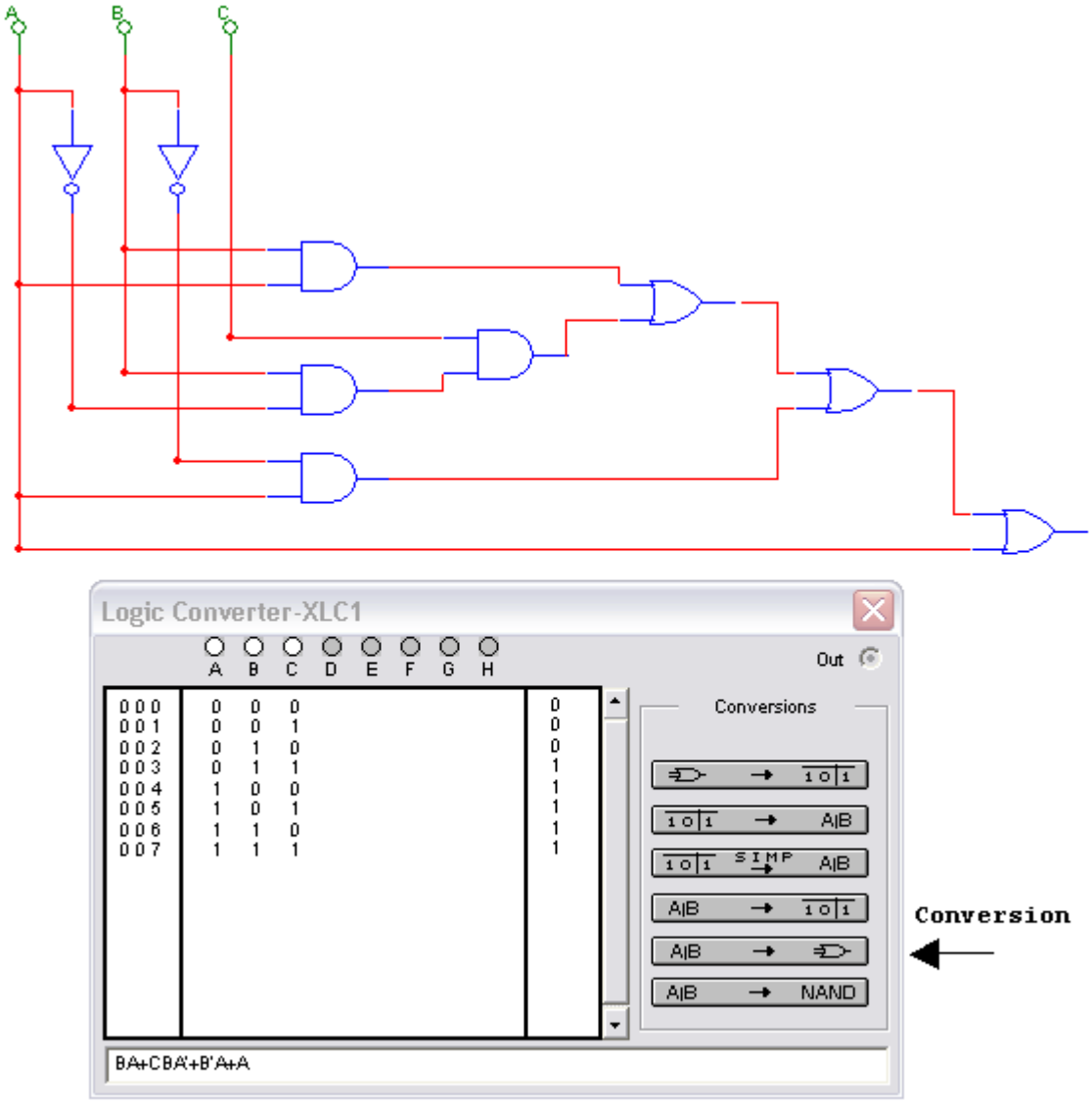


Figura 8

se define como termino canónico de una función lógica a todo producto o suma en el que aparecen todas las variables en su forma directa a o complementada a'.

A los términos producto se les llama productos canónicos o minitérminos. Esta denominación se debe al hecho de que, este término, toma el valor de 1 para una sola combinación de las variables de entrada, de ahí el prefijo mini. A los términos suma se les llama sumas canónicas o maxitérminos, denominándose así por el hecho de que, este término, toma el valor de 1 tantas veces como lo hagan los sumandos que lo forman.

Una función formada, exclusivamente, por términos de sumas canónicas o bien de productos canónicos recibe el nombre de función canónica. Si esta función tiene n variables, cada uno de sus productos sumas canónicas tendrá n variables. Como cada variable se puede representar en su forma directa o complementada, el número de productos canónicos posibles será 2^n, al igual que el de sumas canónicas.

BIBLIOGRAFIA

Acha Santiago. 2003. Electrónica Digital Introducción a la lógica Digital México. 1ª ed. Alfaomega * Ra-ma

Tocci Ronald J. 1996. Sistemas Digitales Principios y Aplicaciones. México. 6ª ed. Prentice-Hall.