Anexo B

Funciones booleanas

El álgebra de Boole provee las operaciones y las reglas para trabajar con el conjunto $\{0, 1\}$. Los dispositivos electrónicos pueden estudiarse utilizando este conjunto y las reglas asociadas al álgebra de Boole. Las tres operaciones utilizadas mas comúnmente son complemento, suma booleana (OR) y producto (AND).

B.1 Funciones y expresiones booleanas

Sea $B = \{0, 1\}$. La variable x se denomina Variable booleana si asume únicamente valores del conjunto B. Una función de B^n, el conjunto $\{(x_1, x_2, \ldots , x_n) \mid x_i \in B, 1 \leq i \leq n\}$ en B se denomina función booleana de grado n.

Las funciones booleanas pueden representarse usando expresiones construidas a partir de variables y operaciones booleanas. Las expresiones booleanas en las variables $x_1, x_2, \ldots , x_n$ se definen en forma recursiva como sigue

$0, 1, x_1, x_2, \ldots , x_n$ son expresiones booleanas.
Si E_1 y E_2 son expresiones booleanas, entonces E_1, $(E_1 . E_2)$ y $(E_1 + E_2)$ son expresiones booleanas.

Cada expresión booleana representa una función. Los valores de esta función se obtienen sustituyendo 0 y 1 en las variables presentes en la expresión.

Las funciones booleanas F y G de n variables se dicen equivalentes si y solo si $F(b_1, b_2, \ldots , b_n) = G(b_1, b_2, \ldots , b_n)$, cuando $b_1, b_2, \ldots , b_n \in B$.

Una función booleana de grado 2 es una función de un conjunto con cuatro elementos, pares de elementos del conjunto $\{0, 1\}$ en B, un conjunto con dos elementos. De manera tal que existen 16 funciones booleanas diferentes de grado 2.

B.2 Identidades del álgebra booleana

Las identidades del álgebra booleana son particularmente útiles para simplificar el diseño de circuitos. Son proposiciones equivalentes y se pueden demostrar utilizando tablas de verdad. Las identidades se muestran en la tabla B.1

Identidad	Nombre
$\overline{\overline{x}} = x$	Doble complemento
$x + x = x$ $x . x = x$	Idempotencia
$x + 0 = x$ $x . 1 = x$	Identidad
$x + 1 = 1$ $x . 0 = 0$	Dominancia
$x + y = y + x$ $x . y = y . x$	Conmutatividad
$x + (y + z) = (x + y) + z$ $x . (y . z) = (x . y) . z$	Asociatividad
$x + y . z = (x + y) . (x + z)$ $x . (y + z) = x . y + x . z$	Distributividad
$\overline{(x.y)} = \overline{x} + \overline{y}$ $\overline{(x + y)} = \overline{x}.\overline{y}$	DeMorgan

Tabla B.1

B.3 Representación de funciones booleanas

Expansiones de suma-producto

Un *minitérmino* de las variables booleanas $x_1, x_2, \ldots, x_n$ es un producto booleano $y_1 . y_2 \ldots y_n$ en donde $y_i = x_i$ o bien $y_i = \overline{x_i}$. Un *literal* es una variable booleana o su complemento. Por lo tanto un minitérmino es un producto de n literales con un literal para cada variable.

Un minitérmino tiene un valor de *1* si y solo si cada variable y_i tiene un valor de *1*.

Tomando sumas booleanas de distintos minitérminos se puede construir una expresión booleana con un conjunto específico de valores. En particular una suma booleana de minitérminos tiene un valor de 1 cuando exactamente uno de los minitérminos en la suma tiene valor *1* y adquiere el valor 0 para cualquier otra combinación de valores de las variables.

Una expansión de suma-producto es una suma de minitérminos. Los minitérminos en la suma booleana corresponden a aquellas combinaciones de valores en los cuales la función adquiere el valor *1*.

A modo de ejemplo se puede encontrar la función booleana correspondiente a la tabla B.2

x	*y*	*z*	*F(x,y,z)*
0	*0*	*0*	*0*
1	*0*	*0*	*0*
0	*1*	*0*	*0*
1	*1*	*0*	*0*
0	*0*	*1*	*0*
1	*0*	*1*	*0*
0	*1*	*1*	*1*
1	*1*	*1*	*1*

Tabla B.2

Para representar F, se necesita una expresión que valga *1* en caso de que $x = 0$ e $y = z = 1$ o bien $x = y = z = 1$. Dicha expresión se puede construir por medio de una suma booleana de dos productos diferentes. Por lo tanto la función F quedaría

$$F(x,y,z) = \bar{x}.y.z + x.y.z$$

B.4 Completitud funcional

Toda función booleana puede representarse por una suma de minitérminos. Cada minitérmino es el producto booleano de variables booleanas o sus complementos. Esto demuestra que cada función booleana puede expresarse con los operadores +, . y $\bar{\ }$ [23]. Como cada función booleana se puede representar, se dice que el conjunto {+, . , $\bar{\ }$} es *funcionalmente completo*. Un conjunto menor funcionalmente completo puede construirse si se consigue expresar alguno de los operadores en términos de los otros dos. Por lo tanto si se utilizan las leyes de DeMorgan, se pueden eliminar las sumas booleanas utilizando la identidad

$$x + y = \overline{\bar{x}.\bar{y}}$$

Esto significa que el conjunto {$\bar{\ }$, .} es funcionalmente completo. De manera similar se puede deducir que el conjunto {$\bar{\ }$, +} también es funcionalmente completo aplicando la segunda ley de DeMorgan

$$x.y = \overline{\bar{x} + \bar{y}}$$

Sin embargo se pueden obtener conjuntos funcionalmente completos, aún mas pequeños. Se define la operación | o (NAND), que dados dos variables booleanas retorna el complemento del producto booleano y \ (NOR) que dadas dos variables booleanas retorna el complemento de la suma booleana. Si se construye un conjunto con cada uno de los operadores, { | } y { \ }, se puede demostrar que ambos conjuntos son funcionalmente completos.

$$\bar{x} = x \mid x$$
$$x.y = (x \mid y) \mid (x \mid y)$$

$$\overline{x} = x \setminus 0$$
$$x.y = (x \setminus 0) \setminus (y \setminus 0)$$

En virtud de la completitud funcional de { . , $\overline{\ }$ }, queda demostrada la completitud de { | } y { \ }.

B.5 Compuertas lógicas

El álgebra booleana se utiliza para modelar los circuitos electrónicos. Un dispositivo electrónico está constituido por un número de circuitos. Cada circuito puede diseñarse aplicando las reglas del álgebra de Boole. Los elementos básicos de los circuitos se denominan *compuertas*. Cada tipo de compuerta representa una operación booleana. En la figura B.1 se muestran los diversos tipos de compuertas. Cada una corresponde a una operación determinada.

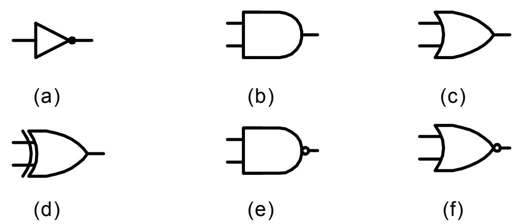

Figura B.1

La compuerta que se observa en la figura B.1 (a) se denomina inversor y representa la operación booleana de negación o NOT, y produce el complemento del valor dado como entrada. En la figura B.1 (b) se presenta la compuerta que representa el producto o AND y en la figura B.1 (c), la compuerta que representa la suma booleana u OR. Las tres últimas compuertas representan las operaciones XOR u OR exclusivo, NAND y NOR. La operación XOR a diferencia del OR, retorna 1 únicamente cuando los valores de entrada son distintos. El funcionamiento de las operaciones NAND (figura B.1 (e)) y NOR (figura B.1 (f)) se explicó en la sección anterior.

Las compuertas anteriores, se pueden utilizar para desarrollar circuitos lógicos combinatorios.

Circuito combinatorio: Se denomina circuito combinatorio a un circuito lógico cuya salida depende únicamente de la entrada y no del estado actual del circuito. En otras palabras, estos son circuitos que se construyen únicamente combinando las diversas compuertas lógicas y por lo tanto carecen de memoria.

B.6 Minimización de circuitos lógicos

La eficiencia de un circuito combinatorio depende del número y organización de la compuertas lógicas que lo comprenden. El diseño de un circuito lógico combinatorio comienza con su especificación mediante una tabla de verdad. A partir de la tabla se pueden utilizar las expansiones de suma-producto para diseñar un conjunto de compuertas lógicas que implementen el circuito. Sin embargo la expansión de suma-producto puede contener mas términos de los realmente necesarios. Los términos que difieren en una sola variable, de tal manera que en un término ocurre la variable y en otro término ocurre su complemento, se pueden combinar. A modo de ejemplo se considera una expansión de suma-producto con las características mencionadas anteriormente, junto con la forma de combinar los términos:

$$\bar{x}.y.z + x.y.z = (\bar{x} + x).(y.z) = 1.(y.z) = y.z$$

La expansión inicial utiliza tres compuertas lógicas y un inversor, mientras que la expansión final utiliza sólo una compuerta.

Para reducir el número de términos en una expresión booleana, se pueden utilizar las identidades definidas en la sección B.2 para encontrar los términos que se puedan combinar. Sin embargo esta tarea puede complicarse a medida que aumenta el número de variables.

Algunos de los métodos que se utilizan para simplificar expresiones booleanas los constituyen el *Mapa de Karnaugh* que es un método gráfico para encontrar los términos que se pueden combinar en una expresión y el método de *Quine-McCluskey* utilizado en expresiones con un gran número de variables [23] [24].