CAS 701

Boolean Algebra

Mahnaz Ahmadi
Oct. 14, 2004
McMaster University

References

J. Eldon whitesitt (Boolean Algebra and its Applications)
Ralph P. Grimaldi (Discrete and Combinational Mathematics)
Kenneth H.Rosen (Discrete Mathematics and its Applications)

Boolean algebra

Let B be a nonempty set that contains two special elements 0 and 1, and on which we define closed binary operations $+, \cdot$, and a unary operation $\bar{\ }$, $(B, +, ., \bar{\ }, 0, 1)$ is called a *Boolean algebra* if the following axioms are satisfied for all $x, y, z \in B$

Identity laws $\begin{cases} x + 0 = x \\ x \cdot 1 = x \end{cases}$

Inverse laws $\begin{cases} x + \bar{x} = 1 \\ x \cdot \bar{x} = 0 \end{cases}$

Commutative laws $\begin{cases} x + y = y + x \\ x \cdot y = y \cdot x \end{cases}$

Distributive laws $\begin{cases} x + (y \cdot z) = (x + y) \cdot (x + z) \\ x.(y + z) = (x.y) + (x.z) \end{cases}$

Associative laws $\begin{cases} (x + y) + z = x + (y + z) \\ (x.y).z = x.(y.z) \end{cases}$

From the axioms above we can derive the following theorems

Idempotent Laws $\begin{cases} x + x = x \\ x \cdot x = x \end{cases}$

Absorption Laws $\begin{cases} x + xy = x \\ x.(x + y) = x \end{cases}$

Dominance Laws $\begin{cases} x + 1 = 1 \\ x.0 = 0 \end{cases}$

Demorgan's Laws $\begin{cases} \overline{x + y} = \overline{x}.\overline{y} \\ \overline{x.y} = \overline{x} + \overline{y} \end{cases}$

Law of the Double Complement $\overline{\overline{x}} = x$

The principle of Duality

Any algebraic equality derived from the axioms of Boolean algebra remains true when the operators (.) and (+) are interchanged and the identity elements 0 and 1 are interchanged. This property is called the duality principle. For example:

$x + 1 = 1 \longrightarrow x.0 = 0$ (Dual)

Because of the duality principle, for any given theorem we get it's dual for free.

An application of Boolean algebra

Consider any set X , and let $P(X)$ stands for the collection of all possible subsets of the set X . (The power set of the set X).

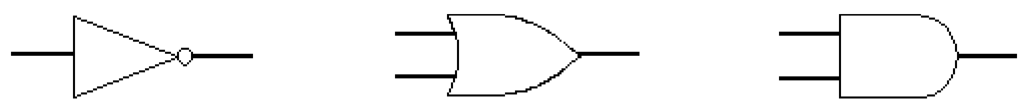
$(P(X),\bigcup,\bigcap,\bar{\ },X,\phi)$ Contains two special elements ϕ (corresponding to 0 in Boolean algebra) and X (Corresponding to 1 in Boolean algebra), and closed binary operations $\cup,\cap$ and a unary operation $\bar{\ }$ is an instance of *Boolean algebra* since the following axioms are satisfied for all $A,B,C \in P(X)$.

Identity laws	$\begin{cases} A\bigcup\phi = A \\ A\bigcap X = A \end{cases}$
Inverse laws	$\begin{cases} A\cup\overline{A} = X \\ A\cap\overline{A} = \phi \end{cases}$
Commutative laws	$\begin{cases} A\bigcup B = B\bigcup A \\ A\bigcap B = B\bigcap A \end{cases}$
Distributive laws	$\begin{cases} A\bigcup(B\bigcap C) = (A\bigcup B)\bigcap(A\bigcup C) \\ A\bigcap(B\bigcup C) = (A\bigcap B)\bigcup(A\bigcap C) \end{cases}$
Associative laws	$\begin{cases} A\bigcup(B\bigcup C) = (A\bigcup B)\bigcup C \\ A\bigcap(B\bigcap C) = (A\bigcap B)\bigcap C \end{cases}$

Logic Gates

In 1938 Claude Shannon showed how the basic rules of logic, first given by Geoge Bool in 1854 in his The Laws of Thought, could be used to design circuits.

- Boolean algebra is used to design and simplify circuits of electronic devices.
- Each input and output can be thought as a member of the set $\{0,1\}$.
- The basic elements of circuits are called gates. Each type of gate implements a Boolean operation.



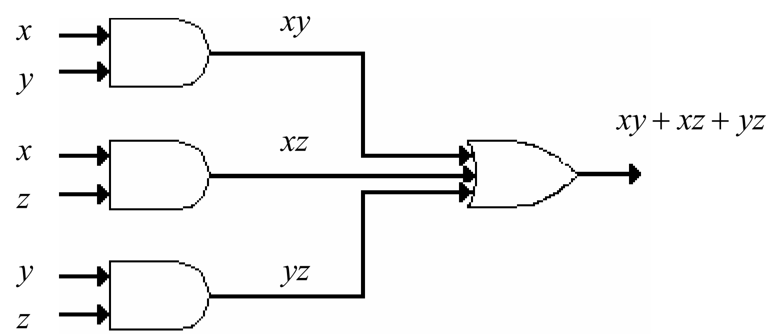
Inverter *OR* *AND*

  - *Inverter*, produces the complement of an input Boolean variable as its output
  - *OR*, which produces the Boolean sum of two or more Boolean variables
  - *AND*, which produces the Boolean products of two or more Boolean variables

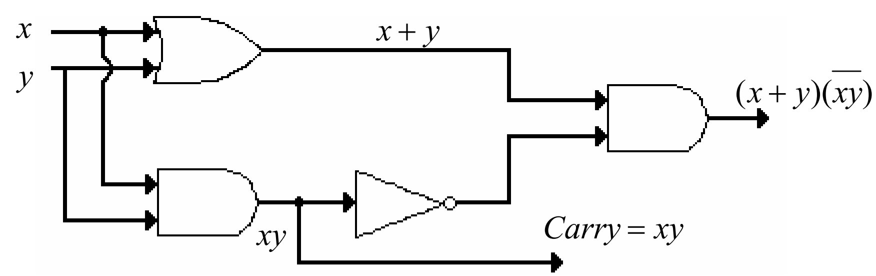
Each circuit can be designed using the rules of Boolean algebra

For example:

1) A circuit for Majority Voting:



2) The Half Adder:



Minimization of Circuits

Boolean algebra finds its most practical use in the simplification of logic circuits. To do this:

- Translate a logic circuit's function into symbolic (Boolean) form
- Apply certain algebraic rules to the resulting equation to reduce the number of terms and/or arithmetic operations

The simplified equivalent function with fewer components will:

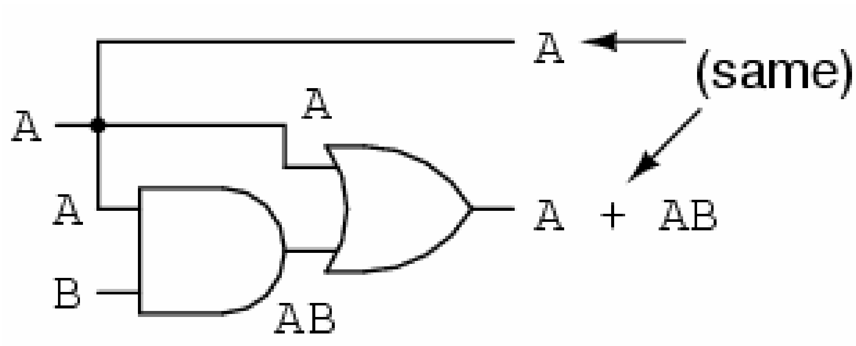
1) Increase efficiency and reliability
2) Decrease cost of manufacture.

There are several ways to reduce the number of terms in a Boolean expression representing a circuit such as:

1) Rules of Boolean algebra
2) Karnaugh Maps
3) The Quine-McCluskey Method

1) Rules of Boolean algebra

$A + AB = A$ (Absorption Laws)



A+AB = A1+AB	Identity Law
= A (1+B)	Distributive Law of . Over +
= A1	Dominance Law (and Commutative Law of +)
= A	Identity Law

2) Karnaugh Maps

Karnaugh map is a graphical method to reduce the number of terms which was introduced by Maurice Karnaugh in 1953.

This method is usually applied only when the function involves *six or fewer* variable.

Karnaugh map uses the sum-of-products (d.n.f) expansion of a circuit to find a set of logic gates that implement circuits and to find terms to combine.

Terms in a sum-of –products expansion that differ in just one variable can be combined, so that in one term this variable occurs and in the other term the complement of this variable occurs.

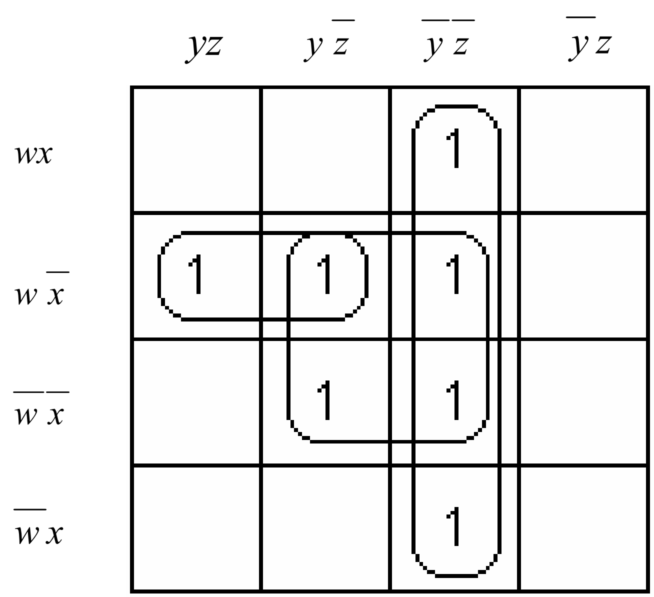
A 1 is placed in the square representing a minterm if this minterm is present in the expansion.

The goal is to identify the largest blocks of 1s in the map and to cover all the 1s using the fewest blocks needed, using the largest blocks first.

Suppose a Boolean function in four variables is:

$wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z} + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}\bar{z}$

A Karnaugh map in four variables is a square that is divided into 16 squares, that means there are 16 possible minters in the sum-of-products expansion of a Boolean function in the four variables w, x, y, z



The minimal expansion is: $\bar{y}\bar{z} + w\bar{x}y + \bar{x}\bar{z}$

3) The Quine-McCluskey Method

It was developed in the 1950s by W.V.Quine and E.J.McCluskey.

The Quine-McCluskey Method is procedure for simplifying sum of products expansions that can be mechanized.

It can be used for Boolean function in any number of variables.

The Quine-McCluskey method consists of two parts.

1) Finds those terms that are candidates for inclusion in a minimal expansion as a Boolean sum of Boolean products
2) Determines which of these terms to actually use

Represent the minterms in the expansion by bit strings.

The first bit will be 1 if x occurs and 0 if $\overline{x}$ occurs. The second and third bit as the same represent for y and z .

Minterms that can be combined are those that differ in exactly one literal. Hence, two terms that can be combined differ by exactly one in the number of 1s in the bit strings that represent them.

A product in two literals is represented using a dash to denote the variable that doesn't occur.

Find a minimal expansion equivalent to:

$$xyz + x\overline{y}z + \overline{x}yz + \overline{x}\,\overline{y}z + \overline{x}\,\overline{y}\,\overline{z}$$

			Step1		Step2	
	Term	**Bit String**	**Term**	**String**	**Term**	**String**
1	xyz	111	(1,2) xz	1-1	(1,2,3,4) z	--1
2	$x\overline{y}z$	101	(1,3) yz	-11		
3	$\overline{x}yz$	011	(2,4) $\overline{y}z$	-01		
4	$\overline{x}\,\overline{y}z$	001	(3,4) $\overline{x}z$	0-1		
5	$\overline{x}\,\overline{y}\,\overline{z}$	000	(4,5) $\overline{x}\,\overline{y}$	00-		

	xyz	$x\overline{y}z$	$\overline{x}yz$	$\overline{x}\,\overline{y}z$	$\overline{x}\,\overline{y}\,\overline{z}$
z	X	X	X	X	
$\overline{x}\,\overline{y}$				X	X

The minimal expansion is: $z + \overline{x}\,\overline{y}$