Chapter 4

Boolean Algebra

Computer Application

History

- In 1854 George Boole introduced systematic treatment of logic and developed for this purpose an algebraic system now called ***Boolean Algebra***.
- In 1938 C. E. Shannon introduced a two-valued Boolean Algebra called ***Switching Algebra***, in which he demonstrated that this algebra can represented by electrical switching.

Chapter Outline

- Boolean Algebra (Switching Algebra)
  - Definitions
  - Basic Axioms
  - Basic Theorems
  - Representation of Boolean Functions

Boolean Algebra Definitions

- Boolean (Switching) Algebra structure consist:
  - **Set of elements (constant) B = {0, 1}.**
  - **Binary operation O = {+, .}.**
  - **Unary operation {‘}.**
  - **The following axiom:**

Closure:	a + b in B	a . b in B
Commutative:	a + b = b + a	a . b = b . a
Associative:	a + (b + c) = (a +b) + c	a . (b . c) = (a .b) . c
Identity:	a + 0 = a	a . 1 = a
	a + 1 = 1	a . 0 = 0
Distributive:	a + (b . c) = (a + b) . (a + c)	a . (b + c) = (a . b) + (a . c)
Complement:	a + a’ = 1	a . a’ = 0

Definitions

- **Boolean Algebra** : An algebraic structure defined with a set of elements B={**0,1**}, a set of binary operators (**+ , . , ‘**), and a number of unproved axioms.
- **Symbolic Variables** such as X, Y , Z represent the elements. A variable can take the value “0” or “1” which corresponds to the condition of a logic signal.
- <u>**Algebraic Operators**</u> :
  - Addition operator (**+**)
  - Multiplication operator (**.**)
  - Complement operator (**‘**)

Basic Axioms

- A variable can take only one of two values{0,1}

  (A_1) **X= 0 if X ≠1** (A_1’) **X=1 if X≠0**

- NOT operation (The complement Operation) :

  (A_2) **If X=0 then X’=1** (A_2’) **If X =1 then X’=0**

- AND and OR operations (Multiplication and Addition) :

  (A_3) **0 . 0 = 0** (A_3’) **0 + 0 = 0**

  (A_4) **1 . 1 = 1** (A_4’) **1 + 1 = 1**

  (A_5) **0 . 1 = 1 . 0 = 0** (A_5’) **1 + 0 = 0 + 1 = 1**

Theorems - Single Variable

- Null elements :
  (T_1) **X + 0 = X** (T_1') **X . 1 = X**
- Identity elements :
  (T_2) **X + 1 = 1** (T_2') **X . 0 = 0**
- Idempotency :
  (T_3) **X + X = X** (T_3') **X . X = X**
- Involution :
  (T_4) **(X')' = X**
- Complements :
  (T_5) **X + X' = 1** (T_5') **X . X' = 0**
- Induction Proof :
  Show that the theorems are true for both X=0 and X=1

Induction Proof

- T_3 : **X+X=X**

X	X	X+X
0	0	0
1	1	1

- T_5 : **X+X'=1**

X	X'	X+X'
0	1	1
1	0	1

Theorems - Multiple Variables

- Commutativity :
  (T_6) **X + Y = Y + X** (T_6') **X . Y= Y . X**
  - The inputs of AND and OR gates can be interchanged.
- Associativiy :
  (T_7) **(X+Y)+Z = X+(Y+Z)** (T_7') **(X.Y).Z= X.(Y.Z)**
  - The order of the input variables could be rearranged.
- Distributivity :
  (T_8) **X.(Y+Z)= X.Y+X.Z** (T_8') **X+Y.Z=(X+Y).(X+Z)**
  - Multiplication distributes over addition
  - Addition distributes over multiplication !!!

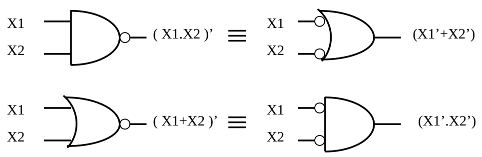
Theorems - Multiple Variables

- Covering :
  (T_9) **X+X.Y = X** (T_9') **X.(X+Y)= X**
- Proof :
  - T_9 : X+X.Y=X.1+X.Y (theorem T_1')
    =X.(1+Y) (theorem T_8- Distributivity)
    =X.1 (theorem T_2)
    =X (theorem T_1')
  - T_9': X.(X+Y) = (X+0).(X+Y) (theorem T_1)
    = X+(0.Y) (theorem T_8'- Distributivity)
    = X+0 (theorem T_2')
    = X (theorem T_1)

DeMorgan

- DeMorgans Theorems
  (T_{13}) **$(X_1.X_2. ... . X_n)' = X_1'+X_2'+...+X_n'$**
  (T_{13}') **$(X_1+X_2+ ... +X_n)'= X_1'.X_2'. ... . X_n'$**

Example : two-variable case



Duality

- Duality : Every Boolean expression remains **<u>valid</u>** if the **(AND, OR)** operators and **{0,1}**elements are interchanged.
- Mathematical definition : if F is a Boolean Function then FD, the dual, function is :
  - **FD(X_1,X_2, ... , X_n, + , . ,') def as F(X_1,X_2,....,X_n, . ,+ ,')**
- FD ≠ F
- Example :
  - $F(X_1,X_2,X_3) = X_1+X_2.X_3$
    $FD(X_1,X_2,X_3) = X_1.(X_2+X_3)$

Exercise : Demorgan/Duality

- F(A,B,C) = AB+AB'C+BC'

  Find : F' , FD
- F' = [AB+AB'C+BC']'= (A'+B')(A'+B+C')(B'+C)

  FD = (A+B)(A+B'+C)(B+C')

  FD(A',B',C') = (A'+B)(A'+B+C')(B'+C)

  FD'(A',B',C') = [(A'+B')(A'+B+C')(B'+C)]'

  = AB+AB'C+BC'