Boolean Algebra

Basic Definition.

- Boolean algebra like any other deductive mathematical system may be defined with a set of elements, a set of operators, and a number of unproved axioms or postulates.

- Boolean algebra is an algebraic structure defined on a set of elements B together with two binary operators + and $\cdot$ provided the following postulates are satisfied:

Postulate 1: Set and Operators

•*(Definition)*: A Boolean algebra is a **closed algebraic system** containing a set *K* of two or more elements and the two operators · and + which refer to logical "AND" and logical "OR".

1(a) Closure with respect to the operator +.

(b) Closure with respect to the operator ·.

Postulate 2: Identity Elements

- There exist 0 and 1 elements in K, such that for every element a in K
- a + 0 = a
- a · 1 = a

Note:
a · 0 = 0
a + 1 = 1

- Definitions:
- 0 is the identity element for + operation
- 1 is the identity element for · operation

4

Postulate 3: Commutativity

- Binary operators + and · are commutative.
- That is, for any elements a and b in K:
- $a + b = b + a$
- $a \cdot b = b \cdot a$

Postulate 4: Distributivity

- Binary operator (+) is distributive over (·) and (·) is distributive over (+).
- That is, for any elements a, b and c in K:
- $a + (b \cdot c) = (a + b) \cdot (a + c)$
- $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Postulate 5: Complement

- A unary operation, *complementation*, exists for every element of K.
- That is, for any elements a in K:

$$x + x' = 1$$
$$x \cdot x' = 0$$

7

Basic Theorems

8

Duality Principle

- Each postulate of Boolean algebra contains a pair of expressions or equations such that one is transformed into the other and vice-versa by interchanging the operators, + ↔ ·, and identity elements, 0 ↔ 1.

- The two expressions are called the duals of each other.
- Example: ____duals A + (BC) = (A+B)(A+C) ↔ A (B+C) = AB + AC

Properties of Boolean Algebra

- Properties stated as theorems.
- The postulates are basic axioms of the algebraic structure and need no proof.
- The theorems must be proven from the postulates.

Postulates and Theorems of Boolean Algebra

Post. 2	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
Post. 5	(a) $x + x' = 1$	(b) $x \cdot x' = 0$
Theorem 1	(a) $x + x = x$	(b) $x \cdot x = x$
Theorem 2	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Theorem 3 involution:	$(x')' = x$	
Post. 3 commutative:	(a) $x + y = y + x$	(b) $xy = yx$
Theorem 4 associative:	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$
Post. 4 distributive:	(a) $x(y + z) = xy + xz$	(b) $x + yz = (x + y)(x + z)$
Theorem 5 DeMorgan:	(a) $(x + y)' = x'y'$	(b) $(xy)' = x' + y'$
Theorem 6 absorption:	(a) $x + xy = x$	(b) $x(x + y) = x$

Theorem 1: Idempotency

- For all elements a in K: x + x = x;
  x.x = x.
- Proof:
  x + x = (x + x)1, (identity element)
  = (x + x)(x + x\`), (complement)
  = x + x x\`, (distributivity)
  = x + 0, (complement)
  = x, (complement)

Similar proof for x.x = x

Theorem 2: Null Elements Exist

- x + 1 = 1 ….. for (+) operator.
- x · 0 = 0 ….. for (·) operator.
- Proof: x + 1 = (x + 1)1, (identity element) =
  1(x + 1), (commutativity)
  = (x+ x\`)(x + 1), (complement)
  = x + x\`. 1, (distributivity)
  = x + x\`, (identity element)
  = 1, (complement)

Similar proof for x. 0 = 0.

Theorem 3: Involution Holds

- $(x`)` = x$
- Proof: $x + x` = 1$ and $x.x` = 0$, (complements)
  or $x` + x = 1$ and $x`.x = 0$, (commutativity)

i.e., x is complement of x`

Therefore, $x`` = x$

Theorem 6: Absorption

- $a + a\,b = a$
- $a\,(a + b) = a$
- Proof: $a + a\,b$ = $a\,1 + a\,b$, (identity element)
  = $a(1 + b)$, (distributivity)
  = $a\,1$, (Theorem 2)
  = a, (identity element)

Similar proof for $a\,(a + b) = a$.

Theorem 5: De Morgan

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

(b) $(xy)' = x' + y'$

- The theorems of Boolean algebra can be shown to hold true by means of truth tables.

For example, the truth table for the first De Morgan's theorem $\boxed{(x + y)' = x'y'}$ is shown below.

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

Theorem 4: Associativity

- Binary operators + and · are associative.
- That is, for any elements a, b and c in K:
- a + (b + c) = (a + b) + c
- a · (b · c) = (a · b) · c

Operator precedence

- The operator precedence for evaluating Boolean expressions is:
  - (1) paren theses,
  - (2) NOT,
  - (3) AND,
  - (4) OR.

- **Example:**
  - **(x+y)`**
  - **X`.y`**

18

Boolean Functions

19

Boolean functions

- Binary variable takes values 0 or 1.
- Boolean function is an expression formed with: binary variables, binary operators (AND, OR), and unary operator (NOT), parentheses, equal sign.
- Function can be 1 or 0. <u>EX:</u>

F1= xyz`….. When does F1=1?

Boolean function representation

Boolean functions can be represented by :

1. Algebraic expression:
2. Truth table:
   - We need 2^n combinations of 1's and 0's.
   - A column showing the combinations where function = 1 or 0.

     EX: if we have F=xyz, *then* :

     we have $\mathbf{2^3=8}$ combinations
3. Logical diagram (using AND,OR,NOT gates).

- Consider the following functions:

F1=xyz`    F2=x+y`z

F3=x`y`z+x`yz+xy`   F4=xy`+x`z

The Truth table is:

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

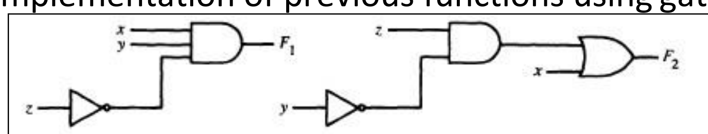
- We can see that F3 = F4
- The question: is it possible to find 2 algebraic expressions that specify the same function?

  Answer is yes.

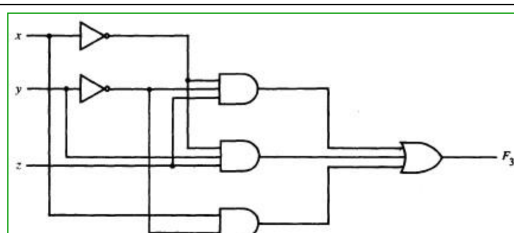
From Table above we find that F_4 is the same as F_3, since both have identical 1's and 0's for each combination of values of the three binary variables. In general, two functions of n binary variables are said to be equal if they have the same value for all possible 2^n combinations of the n variables.

- Manipulation of Boolean expression is applied to **find simpler expressions for the same function**

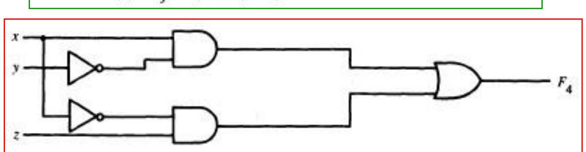
- Implementation of previous functions using gates:



(a) $F_1 = xyz'$ (b) $F_2 = x + y'z$



(c) $F_3 = x'y'z + x'yz + xy'$



(d) $F_4 = xy' + x'z$

Simpler is better

- From the diagrams it is obvious that the implementation of F_4 requires less gates and less inputs than F_3. Since F_4 and F_3 are equal Boolean functions, it is more economical to implement the F_4 form than the F_3 form. To find simpler circuits, one must know how to manipulate Boolean functions to obtain equal and simpler expressions. What constitutes the best form of a Boolean function depends on the particular application. In this section, consideration is given to the criterion of equipment minimization.

Algebraic Manipulation

- A *literal* is a primed or unprimed variable.
- each literal in the function designates an input to a gate,
- each term is implemented with a gate.
- The minimization of the number of literals and the number of terms will result in a circuit with less equipment.

Function Minimization using Boolean Algebra

•

Examples:

- $a + ab =$ $a(1+b)=a$
- $a(a + b) =$ $a.a + ab = a + ab = a(1+b) = a.$
- $a + a'b =$ $(a + a')(a + b) = 1(a + b) = a+b$
- $a(a' + b) =$ $a.\ a' + ab = 0 + ab = ab$

Try

- F = abc + abc’ + a’c

=(a.b(c+c’)) + a’.c

= ab.(1)+ a’c

=ab+ a’c

The other type of question

Show that;

1- $ab + ab' = a$

2- $(a + b)(a + b') = a$

1- $ab + ab' = a(b+b') = a.1=a$ 2- $(a + b)(a + b') = a.a +a.b' +a.b+b.b'$

$= a + a.b' +a.b + 0$

$= a + a.(b' +b) + 0$

$= a + a.1 + 0$

$= a + a = a$

More Examples

- Show that;
  (a) $ab + ab'c = ab + ac$
  (b) $(a + b)(a + b' + c) = a + bc$

(a) $ab + ab'c = a(b + b'c)$

$= a((b+b').(b+c))=a(b+c)=ab+ac$

(b) $(a + b)(a + b' + c)$

$= (a.a + a.b' + a.c + ab +b.b' +bc)$

$= \dots$

Complement of a Function

- The complement of a function F is F'
- may be derived algebraically through De Morgan's theorem.
- De Morgan's can be extended to three or more variables.

- $(A + B + C)' = (A + X)'$ Let $B + C = X$
  $= A'X'$ by theorem 5(a) (De Morgan)
  $= A' \cdot (B + C)'$ substitute $B + C = X$
  $= A' \cdot (B'C')$ by theorem 5(a) (De Morgan)
  $= A'B'C'$ by theorem 4(b) (associative)

- De Morgan's theorems for any number of variables are similar

Find the complement of the functions:

$$F_1 = x'yz' + x'y'z$$
$$F_2 = x(y'z' + yz).$$

$$F_1' = (x'yz' + x'y'z)'$$
$$= (x'yz')' \cdot (x'y'z)'$$
$$= (x + y' + z)(x + y + z')$$

$$F_2' = [x(y'z' + yz)]'$$
$$= x' + (y'z' + yz)'$$
$$= x' + (y'z')' \cdot (yz)'$$
$$= x' + (y + z)(y' + z')$$

- A simpler procedure for deriving the complement of a function
  1) take the dual of the function.
  2) complement each literal.

- Find the complement of the function $F_1 = x'yz' + x'y'z$ by taking their dual and complementing each literal.

1) The dual is: $(x' + y + z')(x' + y' + z)$
2) Complement each literal: $(x + y' + z)(x + y + z') = F_1'$