Unit 2 Boolean Algebra

2.1 Introduction

We will use variables like x or y to represent inputs and outputs (I/O) of a switching circuit. Since most switching circuits are 2 state devices (having only 2 different possible values) the symbols "0" & "1" are used for the value

Thus for variable x, x = 0 or x = 1

Note: Although the symbols 0 & 1 *look like* binary numbers, they are not. They have no numeric value! They merely represent the 2 possible states of a switching circuit.

2.2 Basic Operations

Complement: *Often called inverse; designated by a '(prime). Complement of 1 is 0 and of 0 is 1*

Circuit Representation

Often called NOT operation because x = 1 if x is not equal to 0

AND: *Referred to as a Boolean multiplier, Designated by ●*

Definition:
0 ● 0 = 0   *0 ● 1 = 0*   *1 ● 0 = 0*   *1 ● 1 = 1*

Note: *0 & 1 are Boolean constants not binary #'s*

For C = A ● B

A	B	C = A * B
0	0	0
0	1	0
1	0	0
1	1	1

Note: *C = 1 if and only if (iff) A and B are both 1*

Circuit Representation:

Note: *Often the "●" is omitted leaving AB instead of A● B*

<u>OR:</u> *Referred to as logic addition, Described by "+"*

Definition:

0 + 0 = 0 *0 + 1 = 1* *1 + 0 = 1* *1 + 1 = 1*

For C = A + B

A	B	C = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Note: *C = 1 iff A <u>or</u> B (or both) = 1*

Circuit Representation

<u>Switches:</u>

X= 0 if switch is open *x = 1 if switch is closed*

<u>Series switches</u>

T= 0 open circuit between 1 and 2

T = 1 closed circuit between 1 and 2

T= 1 iff A and B are closed

T = A ● B

<u>Parallel switches</u>

T = 1 iff A or B is closed

T = A + B

Thus series switch perform AND parallel OR operations

2.3 Boolean Expressions and Truth Tables

Example Boolean expressions

AB’ + C

[A(C+D)]’ + BE

Order of operations

Parentheses
Complementation
AND
OR

Circuits for above equations

The expression can evaluate by substituting 0 or 1 as values for each variable

<u>Literal:</u> *each occurrence of a variable or the complement of an expression*

<u>Truth Tables</u>: (*table of combinations) Specify values of a Boolean expression for every possible combination of values of the variables in the expression*

Ex F = A’ + B

A	B	A’	F = A’ + B
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

An expression with n variables, each having a value of 0 or 1 has a different number of combinations of values of variable = 2^n

See Table 2-1 for AB’ + C Example

2.4 Basic Theorems

$X + 0 = X$ $X \bullet 0 = 0$
$X + 1 = 1$ $X \bullet 1 = X$

Idempotent laws

$X + X = X$ $X \bullet X = X$

Involution law

$(X')' = X$

Laws of complementary

$X + X' = 1 \qquad X \bullet X' = 0$

Any of these theorems can be proven by substituting in a value of 0 or 1 for X

Note: An entire expression can be substituted for x

Switch examples

A * A

A + A

A + 1

A + 0

2.5 Commutative, Associate, Distributive Laws

Commutative Laws

$XY = YX \qquad X + Y = Y + X$

Associative Laws

$(XY)Z = X (YZ) = XYZ$

$(X + Y) + Z = X + (Y + Z) = X + Y + Z$

Proof using truth table:

X	Y	Z	XY	YZ	(XY)Z	X(YZ)
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

Circuit Gates

When ANDing variables the result is a1 iff all values are 1

When OR the result is a 1 if any value is a 1 and 0 iff all variables are zero

Distributive Laws

$X (Y+Z) = XY + XZ$

$X + YZ = (X + Y) (X + Z)$

Proof

This law is a very useful Boolean manipulation

2.6 Simplification Theorems

$XY + XY' = X$ $(X+Y)(X+Y') = X$

$X + XY = X$ $X(X + Y) = X$

$(X + Y')\,Y = XY$ $XY' + Y = X + Y$

Each can be proven with truth tables or algebraically

Ex. X + XY $= X \bullet 1 + X \bullet Y$
$= X\,(1 + Y)$
$= X$

Ex. Y + XY' using switches

Ex. F = A (A' + B) using gates

Example Problems:

$Z = A'BC + A'$

$Z = [A + B'C + D + EF] [A + B'C + (D + EF)']$

$Z = (AB + C) (B'D + C'E') + (AB + C)'$

2.7 Multiplying Out, Factoring

Multiplying

Sum of products- *when all products are products of only single variables*
End result when an expression is fully multiplied out

Ex. $AB + CD + EF$
or
$A + BC + DE$

Distributive laws are used to obtain the sum of products

Note: *When multiplying out try second distributive law first, if possible*

Ex. $(A + BC) (A + D + E)$

The problem could be done by multiplying out fully but that would be time consuming

(A + BC) (A + D + E)

Factoring

Product of sums- *all sums are sums of single variables*

An expression is fully factored iff it is in its product of sum form

Ex. A + B'CD

Ex. AB' + C'D

Ex. C'D + C'E' + G'H

Note: *When factoring apply the ordinary distributive law before the second law*

Gates

Sum of products - expressed as one or more AND gates feeding a single OR gate

Ex. A + BCD + BCE

Product of sums - expressed as one or more OR gates feeding a single AND gate

Ex. (A + C) (A + B) (A + D)

2.8 DeMorgan's Law

Used to find inverse or complement of Boolean expressions

$(X + Y)' = X'Y'$

$(XY)' = X' + Y'$

Proof:

X	Y	X'	Y'	X + Y	(X + Y)'	X'Y'	XY	(XY)'	X' + Y'
0	*0*	*1*	*1*	*0*	*1*	*1*	*0*	*1*	*1*
0	*1*	*1*	*0*	*1*	*0*	*0*	*0*	*1*	*1*
1	*0*	*0*	*1*	*1*	*0*	*0*	*0*	*1*	*1*
1	*1*	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*0*

Generalizad for n-variables

The complement of the products is the sum of the complements

The complement of the sum is the product of the complements

Ex. $[(A' + B) C']'$

Ex. $[(AB' + C)D' + E]'$

Note: *The complement operation is only applied to single variables in the final expression*

Dual*: Boolean expression found by replacing, AND with OR, OR with AND, 0 with 1, and 1 with 0. Variables and complements are left unchanged*

Ex. $(XYZ)^D = X + Y + Z$ $(X + Y + Z)^D = XYZ$

Dual can be found by complementing the entire expression then complementing each individual variable

Ex. $(AB' + C)^D$

Laws and Theorems of Boolean Algebra

Operations with 0 and 1:

1. $X + 0 = X$ — 1D. $X \cdot 1 = X$
2. $X + 1 = 1$ — 2D. $X \cdot 0 = 0$

Idempotent laws:

3. $X + X = X$ — 3D. $X \cdot X = X$

Involution law:

4. $(X')' = X$

Laws of complementarity:

5. $X + X' = 1$ — 5D. $X \cdot X' = 0$

Commutative laws:

6. $X + Y = Y + X$ — 6D. $XY = YX$

Associative laws:

7. $(X + Y) + Z = X + (Y + Z) = X + Y + Z$ — 7D. $(XY)Z = X(YZ) = XYZ$

Distributive laws:

8. $X(Y + Z) = XY + XZ$ — 8D. $X + YZ = (X + Y)(X + Z)$

Simplification theorems:

9. $XY + XY' = X$ — 9D. $(X + Y)(X + Y') = X$
10. $X + XY = X$ — 10D. $X(X + Y) = X$
11. $(X + Y')Y = XY$ — 11D. $XY' + Y = X + Y$

DeMorgan's laws:

12. $(X + Y + Z + \ldots)' = X'Y'Z' \ldots$ — 12D. $(XYZ \ldots)' = X' + Y' + Z' + \ldots$

Duality:

13. $(X + Y + Z + \ldots)^D = XYZ \ldots$ — 13D. $(XYZ \ldots)^D = X + Y + Z + \ldots$

Theorem for multiplying out and factoring:

14. $(X + Y)(X' + Z) = XZ + X'Y$ — 14D. $XY + X'Z = (X + Z)(X' + Y)$

Consensus theorem:

15. $XY + YZ + X'Z = XY + X'Z$ — 15D. $(X + Y)(Y + Z)(X' + Z) = (X + Y)(X' + Z)$

Roth, JR., C. H., & Kinney, L. L. (2010). *Fundamentals of logic design*. (Sixth ed., p. 55). Stamford, CT: Centage.