REVIEW OF LOGIC DESIGN

BOOLEAN ALGEBRA

The Boolean algebra is an algebra dealing with binary variables and logic operations.

A Boolean expression is an algebraic expression formed by using binary variables, the constants 1 and 0, the logic operation symbols and parenthesis.

Basic Properties and Theorems:

Closure w.r.t. + : x, y ϵ B ➔ x + y ϵ B

Closure w.r.t. • : x, y ϵ B ➔ x • y ϵ B

1- a)	$A + 0$	$= A$
b)	$A.1$	$= A$
2- a)	$A + 1$	$= 1$
b)	$A.0$	$= 0$
3- a)	$A + A$	$= A$
b)	$A.A$	$= A$
4- a)	$A + \overline{A}$	$= 1$
b)	$A . \overline{A}$	$= 0$
5- a)	$\overline{\overline{A}}$	$= A$
b)	$\overline{(A)}$	$= \overline{A}$

$0.0 = 0$
$0+0 = 0$
$1.1 = 1$
$1+1 = 1$
$1.0 = 0$
$0.1 = 0$
$0+1 = 1$
$1+0 = 1$

6-	a) $A.B = B.A$ b) $A+B = B+A$	Commutative
7-	a) $(A+B)+C = A+(B+C)$ b) $(A.B).C = A.(B.C)$	Associative
8-	a) $A.(B+C) = A.B+A.C$ b) $A+BC = (A+B).(A+C)$	Distributive
9-	a) $A+A.B = A$ b) $A.(A+B) = A$	Absorbtion
10-	a) $A+\overline{A}.B = A+B$ b) $A.(\overline{A}+B) = A.B$	Common Identities

De Morgan's Theorem:

a) $\overline{A + B} = \overline{A} . \overline{B}$
b) $\overline{A . B} = \overline{A} + \overline{B}$

LOGIC OPERATIONS

There are 2 to the power of 2^n different combinations (functions) for n binary variables. So, for n = 2, there are $2^4 = 16$ Boolean functions:

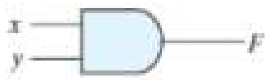
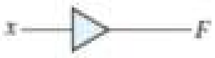
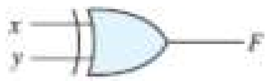
x y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

1. Two functions that produce a constant 0 or 1: F0, F15.
2. Four functions with unary operations complement and transfer: F3, F5, F10, F12.
3. Ten functions with binary operators that define 8 different operations: AND, OR, NAND, NOR, XOR, XNOR (EQUIVALANCE), Inhibition, Implication.

Boolean Expressions for the 16 Functions of Two Variables

Boolean Functions	Operator Symbol	Name	Comments
$F_0 = 0$		Null	Binary constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x, but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y, but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y, but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \subset y$	Implication	If y, then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x, then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

LOGIC GATES

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y F 0 0 0 0 1 0 1 0 0 1 1 1
OR		$F = x + y$	x y F 0 0 0 0 1 1 1 0 1 1 1 1
Inverter		$F = x'$	x F 0 1 1 0
Buffer		$F = x$	x F 0 0 1 1
NAND		$F = (xy)'$	x y F 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$F = (x + y)'$	x y F 0 0 1 0 1 0 1 0 0 1 1 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y F 0 0 0 0 1 1 1 0 1 1 1 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y F 0 0 1 0 1 0 1 0 0 1 1 1

XOR and XNOR

2-input XOR – XNOR:

3-input XOR – XNOR:

Properties of XOR Function:

$$x \oplus 0 = x$$
$$x \oplus 1 = x'$$
$$x \oplus x = 0$$
$$x \oplus x' = 1$$
$$x \oplus y' = x' \oplus y = (x \oplus y)'$$

Prove that:

a) *(x XOR y XOR z)' = (x XNOR y XNOR z)*

b) *(x XNOR y XNOR z) = ((x XOR y) XNOR z)*