COMPUTER ORGANIZATION (CDA – 3103)

SPRING 2005

Lab # 2: Logical Gates & Boolean Algebra

In this lab, we will discuss the basic gates that are being used for the construction of any digital circuits.

1. The input(s) and output(s) of any digital circuits are binary values, each having one of two possible values; which are 0/1 or false/true or low/high.
2. The digital circuit performs a specific function (or operation) on given set of input values and generates a deterministic set of output values. This specific function of circuit can be represented in different ways.
    a. Symbols: Symbolic representation of a circuit.
    b. Logical expression: A logical expression representing the function of the digital circuit.
    c. Truth Table: Finite combination of input and their corresponding output values.
    d. Timing Diagram: Behavior of output over time with respect to the changes in the input.

1 LOGICAL GATES

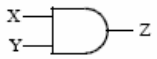
1.1 STANDARD GATES

1.1.1 OR Gate

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Either of the input is TRUE, output is TRUE. Otherwise output is FALSE.
Z = A + B

1.1.2 AND Gate



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Either of the input is FALSE, output is FALSE. Otherwise output is TRUE.
Z = A.B or AB

1.1.3 NOT Gate

Invert the input value. Also called complement or negation of the input.

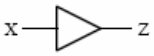
$Z = X'$ or $Z = \overline{X}$

X	Z
0	1
1	0

1.1.4 Buffer

Output same as input value.

$Z = X$



X	Z
0	0
1	1

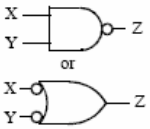
1.2 UNIVERSAL GATES

There are certain gates that can be used in different combination to get functionality of all other gates.

1.2.1 NAND Gate

Either of the input is FALSE, output is TRUE. Otherwise output is FALSE.

$Z = \overline{A.B}$ or $Z = \overline{AB}$

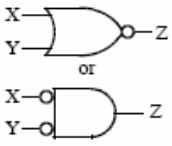


X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

1.2.2 NOR Gate

Either of the input is TRUE, output is FALSE. Otherwise output is TRUE.

$Z = \overline{A + B}$



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

1.3 OTHER GATES

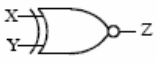
1.3.1 Exclusive OR Gate

Output is TRUE if inputs are complement of each other. Otherwise output is FALSE.

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

1.3.2 Equivalence Gate

Output is TRUE if inputs are identical. Otherwise output is FALSE.



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

2 BOOLEAN ALGEBRA

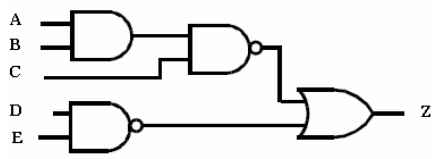
2.1 BASICS

This is an algebraic notation for representing any logical circuit.

1. The operator plus (+) is used for representing OR operation.
2. The operator dot (.) or absence of any operator represents AND operation.
3. The bar on the top of an expression or character 'c' represents complement operation.

2.1.1 Example

$Z = [(A.B).C]^c + [D.E]^c$ or $Z = \overline{[(A.B).C]} + \overline{[D.E]}$



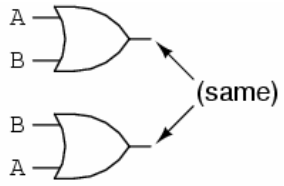
2.2 BOOLEAN ALGEBRAIC IDENTITIES

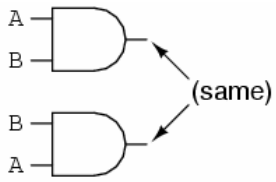
1. $A + 0 = A$
2. $A . 0 = 0$
3. $A + 1 = 1$
4. $A . 1 = A$
5. $A + A = A$
6. $A . A = A$
7. $A + A^c = 1$
8. $A . A^c = 0$
9. $(A^c)^c = A$

Additive	Multiplicative
$A + 0 = A$	$0A = 0$
$A + 1 = 1$	$1A = A$
$A + A = A$	$AA = A$
$A + \overline{A} = 1$	$A\overline{A} = 0$

Commutative Property

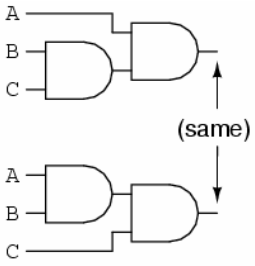
10. $A + B = B + A$
11. $A . B = B . A$

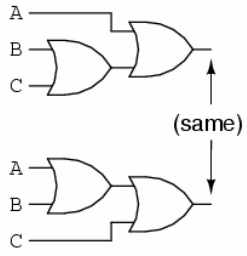




Associative Property

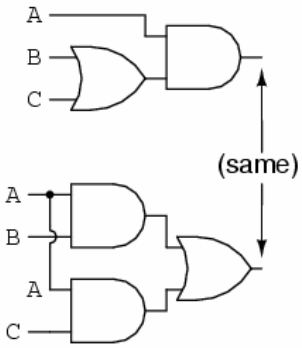
12. $A + (B + C) = (A + B) + C$
13. $A . (B . C) = (A . B) . C$





Distributive Property

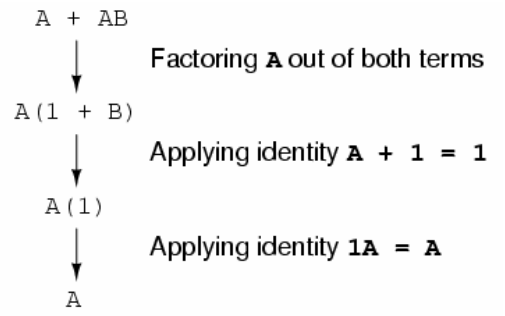
14. $A . (B + C) = A.B + A.C$

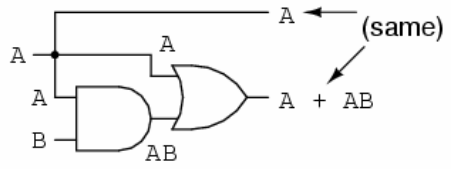


2.3 SIMPLIFICATION OF BOOLEAN EXPRESSION

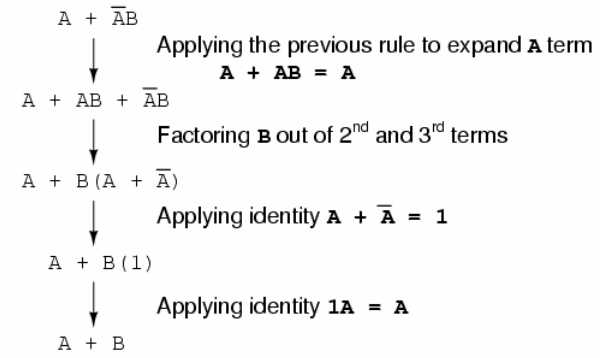
Boolean algebra finds its most practical use in the simplification of logic circuits. If we translate a logic circuit's function into symbolic (Boolean) form, and apply certain algebraic rules to the resulting equation to reduce the number of terms and/or arithmetic operations, the simplified equation may be translated back into circuit form for a logic circuit performing the same function with fewer components. If equivalent function may be achieved with fewer components, the result will be increased reliability and decreased cost of manufacture.

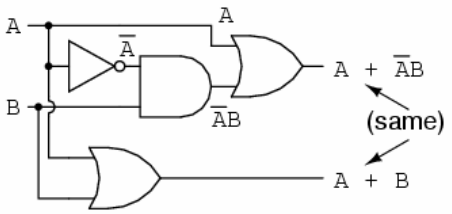
2.3.1 Example 1: Prove **A+A.B = A**





2.3.2 Example 2: Prove **A+A^{c}B = A+B**





*2.3.3 Example 3: Prove (***A+B)(A+C) = A + BC**