LECTURE 3: DIGITAL LOGIC CIRCUIT DESIGN

Binary Logic:

Binary logic deals with variables e.g. x, y, z, A, B, C, … etc., that take on two discrete values (e.g. 1 & 0, True & False, … etc.) and logic operations.

There are three basic logic operations:

1. AND → $x.y = z$ → reads $x\ AND\ y\ is\ qual\ to\ z$ and it means that $z=1$ if and only if ; otherwise $z=0$.

2. OR → $x+y=z$→ reads $x\ OR\ y\ is\ qual\ to\ z$ and it means that $z=1$ if $x=1$ or if $y=1$ or if both $x=1$ and $y=1$. If both $x\ and\ y=0$ then $z=0$.

3. NOT → $x'=z\ (or\ \overline{x}=z)$→ reads "$not\ x\ is\ equal\ to\ z$" meaning that z is what x is not.

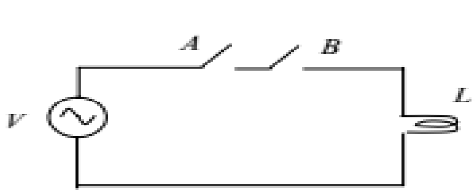
These logic operations can be illustrated in the form of truth tables:

AND			OR			NOT	
x	y	x.y	x	y	x+ y	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

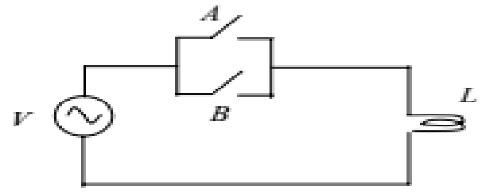
LECTURE 3: DIGITAL LOGIC CIRCUIT DESIGN

Switching Circuits & Binary Logic:

Binary logic can be demonstrated by switching circuits



$$L = A \cdot B$$

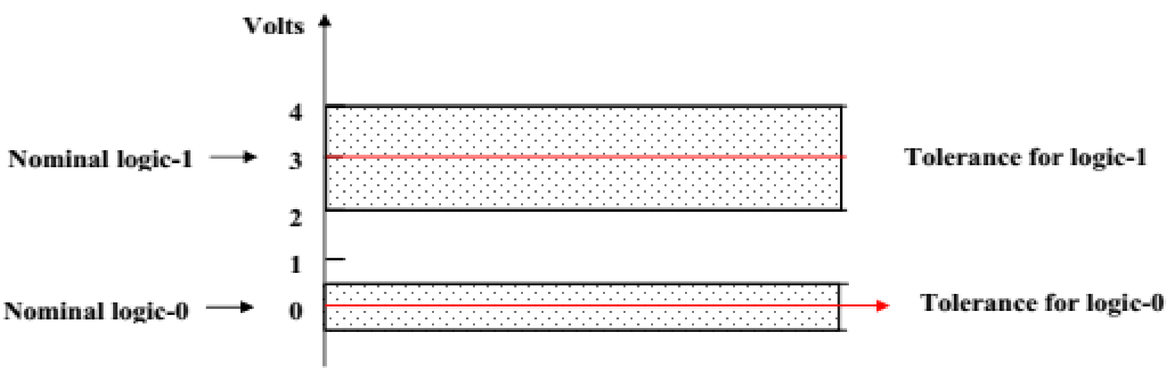


$$L = A + B$$

Binary Signals:

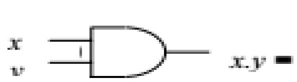
Electrical signals are used to change the state of electronic switches between the two states of conduction and non-conduction. An example is that the logical states of 1 and 0 can be represented by electrical voltages of +3 V and 0 V respectively, as

shown in the binary signal representation. Possible tolerance regions are indicates.

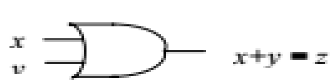


Basic Logic Gates and Signal Waveforms:

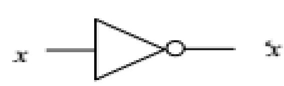
The conventional symbols for 2-input AND and OR gates and the single input inverter are shown:



2-input AND gate



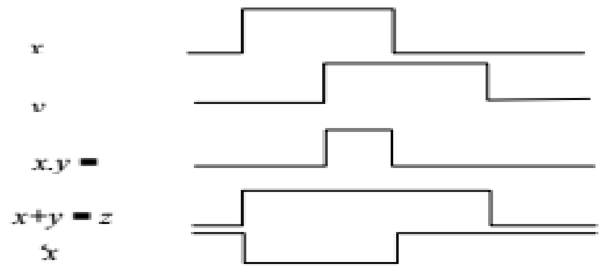
2-input OR gate



Inverter

The number of inputs for the “and” & “or” gates can be more than two.

Input output signals for logic gates may be represented in a signal waveform as shown.



Boolean Algebra:

It is defined with a set of elements, set of operators, and a number of unproved axioms or postulates. We are interested in two-valued Boolean Algebra.

Axiomatic Definition of Boolean Algebra:

Boole in 1854 introduced the treatment of logic and Shannon in 1938 introduced the 2-valued Boolean Algebra (Switching Theory).

We define Boolean Algebra by using the following *Huntington's postulates* defined on a set of two elements *B* and two binary operators: +, .

1. Closure **a.** with respect to + **b.** with respect to .

2. Identity **a.** wrt + called 0 such that x + 0 = 0 + x = x
b. wrt . called 1 such that x.1 = 1.x = x

3. Commutative **a.** x + y = y + x **b.** x.y = y.x

4. Distributive **a.** x.(y+z) = x.y + x.z **b.** x + (y.z) = (x + y).(x + z)

5. For $x \in B$ there is $x' \in B$ such that $x + x' = 1$, *and* $x . x' = 0$

6. There exists at least two elements $x, y \in B$ *such that* $x \neq y$.

The two-valued Boolean algebra:

Defined on a set of two elements B = {0, 1} and two binary operators . and +

X	Y	X.Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

X	X'
0	1
1	0

According to the definition of the two binary operators given in the tables, they become the AND and the OR logic operators of the Binary Logic.

LAWS AND THEOREMS OF BOOLEAN ALGEBRA

Duality:

If binary operators and identity elements are interchanged, then the dual is obtained.

Examples:

1. $(X + Y + Z + ...)^D = XYZ...$
2. $(XYZ ...)^D = X + Y + Z + ...$
3. $\{f(X_1,X_2,...,X_n,0,1,+,.)\}^D = f(X_1,X_2,...,X_n,1,0,.,+)$

Theorems and Postulates:

	Identity	Dual
Postulate 2	X + 0 = X	X.1 = X
Postulate 5	X + X' = 1	X.X' = 0
Theorem 1	X + X = X	X.X = X
Theorem 2	X + 1 = 1	X.0 = 0
Theorem 3	(X')'= X	
Cummutative law (Postulate 3)	X + Y = Y + X	X.Y = Y. X
Associative law (Theorem 4)	(X + Y) + Z = X + (Y + Z) = X + Y + Z	(XY)Z = X(YZ)= XYZ
Distributive law (Postulate 4)	X(Y + Z) = XY + XZ	X + (YZ) = (X + Y)(X + Z)
DeMorgan's Theorem (Theorem 5)	(X + Y + Z + ...)' = X'Y'Z'...	(XYZ...)' =X'+Y'+Z'+ ...
Absorption Theorem (Theorem 6)	X + X.Y= X	X(X + Y) = X

LECTURE 3: DIGITAL LOGIC CIRCUIT DESIGN

Proof of some Theorems:

Theorem 1

X + X = X

Proof:

= (X + X).1 = (X + X).(X + X')

= X + X.X' = X + 0 = X

Dual of Theorem 1

X.X = X

Proof:

= X.X + 0 = X.X + X.X'

= X(X + X') = X.1 = X

DeMorgan's Theorem

(X + Y)' =X'Y'

Using the definition of the complement X and X' such that X + X' = 1 and X.X' = 0

(X + Y) and X'Y' are complements if (X + Y) + X'Y' = 1 and (X + Y).X'Y' = 0

Therefore, this proves the theorem.

X + Y + X'Y' = X + (Y + X')(Y + Y') = X + (Y + X').1 = X + Y + X' = 1 + Y = 1

(X + Y)X'Y' = XX'Y' + YX'Y' = 0.Y' + 0.X' = 0 + 0 = 0

Absorption Theorem

X + XY = X → X.1 + XY = X(1+Y) = X.1 = X

Proving Distributive Law by Truth Table

X(Y + Z) = XY + XZ

X	Y	Z	Y + Z	X(Y + Z)	X.Y	X.Z	X.Y + X.Z
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Operator Precedence:

For evaluating Boolean Expressions:

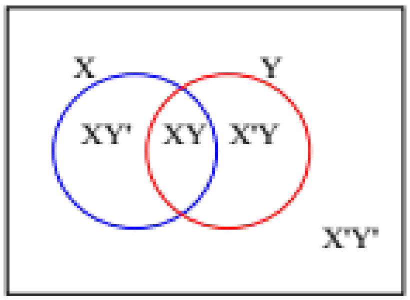
1. Parentheses 2. NOT 3. AND 4. OR

LECTURE 3: DIGITAL LOGIC CIRCUIT DESIGN

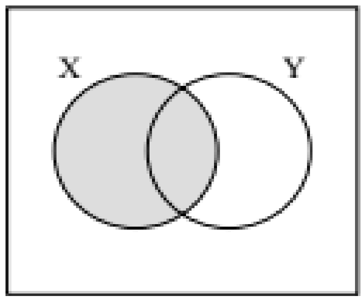
Venn Diagrams:

Venn diagrams may be used to prove Boolean algebra theorems and logic expressions.

2 Variable Venn Diagrams



X + XY = X



Boolean Functions

The following Boolean functions are represented in the truth table.

$F_1 = xyz'$

$F_2 = x + y'z$

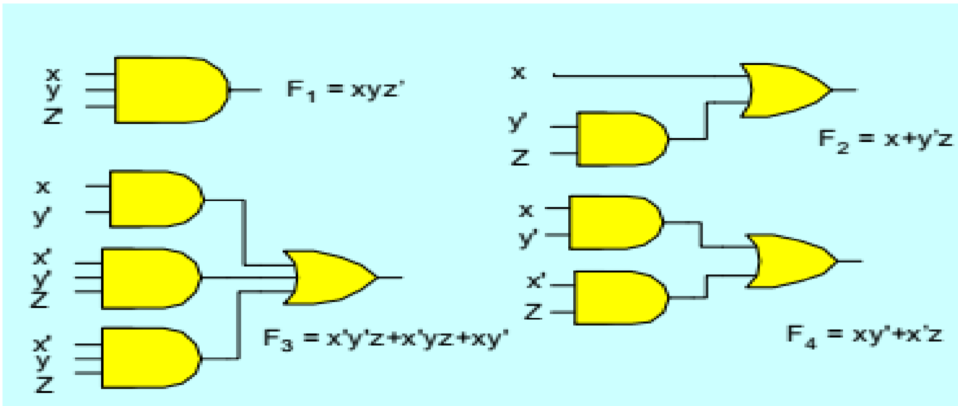
$F_3 = x'y'z + x'yz + xy'$

$F_4 = xy' + x'z$

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

LECTURE 3: DIGITAL LOGIC CIRCUIT DESIGN

The functions can be implemented, using the basic logic gates, as shown in the following logic diagrams.



Algebraic Manipulation

Boolean functions are made up of terms. Each term consists of a number of literals. A literal is a variable or the complement of a variable. Each term is represented by a logic gate and each literal represents an input to a logic gate. By reducing the number of terms, the number of literals, or both, a simpler logic circuit can be used to implement the Boolean function.

Reduction of the number of terms and/or number of literals is done by algebraic manipulation.

Examples

1. $x(x'+y) = xx'+xy=0+xy=xy$

2. The dual of (1) is →
$x+x'y=(x+x')(x+y)=1.(x+y)=x+y$

3. $(x+y)(x+y')=x+yy'=x+0=x$

4. $$\begin{aligned} xy+x'z+yz &= xy+x'z+(x+x')yz \\ &= xy+xyz+x'z+x'yz \\ &= xy(1+z)+x'z(1+y) \\ &= xy.1+x'z.1 = xy+x'z \end{aligned}$$

5. $(x+y)(x'+z)(y+z) = (x+y)(x'+z)$ → by duality.

Complement of a Function

The complement of a Boolean function may be obtained by either one of two methods:

1. Repetitive application of DeMorgan's theorem.
2. Taking the dual of the function and complementing each literal.

Example:

Find the complement of $F = x'yz' + x'y'z$

First method
$$F' = (x'yz' + x'y'z)' = (x'yz')'.(x'y'z)'$$
$$= (x + y' + z)(x + y + z')$$

Second method

dual of F → $F^{dual} = (x' + y + z')(x' + y' + z)$

then complement each literal

$$F' = (x + y' + z)(x + y + z')$$

Canonical and Standard Forms

Minterms and Maxterms

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7