3.2 Boolean Algebra 01.09.23.

- ☐ Boolean algebra is a mathematical system for manipulating variables that can have one of two values.
  - In formal logic, these values are "true" and "false"
  - In digital systems, these values are "on"/"off," "high"/"low," or "1"/"0".
  - So, it is perfect for binary number systems
- ☐ Boolean expressions are created to operate Boolean variables.
  - Common Boolean operators include AND, OR, and NOT.

Boolean Algebra

- ☐ The function of Boolean operator can be completely described using a ***Truth Table***.
- ☐ The truth tables of the Boolean operators AND and OR are shown on the right.
- ☐ The AND operator is also known as the ***Boolean product*** ".". The OR operator is the ***Boolean sum*** "+".

X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

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

Lab: 1. W.A.P. to implement AND, OR - operator for (i) two i/p (ii) three i/p

2. W.A.P. to implement the: (i) $f(x,y,z) = x\bar{z} + y$ (ii) $f(x,y,z) = (x+y)(x+\bar{z})(\overline{x\bar{z}})$

3. W.A.P. - - — (i) $f(x,y,z) = \Sigma m(0,2,5,7)$; $\Sigma m(1,3,5)$

(ii) $f(x,y,z) = \Pi M(1,3,4,6)$, $\Pi M(2,5,6,7)$

Boolean NOT

- The truth table of the Boolean NOT operator is shown on the right.
- The NOT operation is most often designated by an **overbar** "‾".
  - Some books use the prime mark (') or the "elbow" (¬), for instead.

NOT X

X	$\overline{X}$
0	1
1	0

Boolean Function

- A Boolean function has:
  - At least one Boolean variable,
  - At least one Boolean operator, and
  - At least one input from the set of {0,1}.
- It produces an output that is a member of the set {0,1} – Either 0 or 1.

Now you know why the binary numbering system is so handy for digital systems.

Boolean Algebra

- Let's look at a truth table for the following Boolean function shown on the right. :

  $F(x,y,z) = x\bar{z}+y$

- To valuate the Boolean function easier, the truth table contains a extra columns (shaded) to hold the evaluations of partial function.

$F(x,y,z) = x\bar{z}+y$

x	y	z	$\bar{z}$	$x\bar{z}$	$x\bar{z}+y$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

Rules Of Precedence

- Arithmetic has its rules of precedence
  - Like arithmetic, Boolean operations follow the rules of precedence (priority):
  - NOT operator > AND operator > OR operator .
- This explains why we chose the shaded partial function in that order in the table.

$F(x,y,z) = x\bar{z}+y$

x	y	z	$\bar{z}$	$x\bar{z}$	$x\bar{z}+y$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

Rules Of Precedence

Use Boolean Algebra in Circuit Design

- Digital circuit designer always like achieve the following goals:
  - ***Cheaper*** to produce
  - Consume ***less power***
  - run ***faster***
- How to do it? -- We know that:
  - Computers contain circuits that implement Boolean functions → Boolean functions can express circuits
  - If we can simplify a Boolean function, that express a circuit, we can archive the above goals
- We always can reduce a Boolean function to its *simplest* form by using a number of Boolean laws can help us do so.

Boolean Algebra Laws

- Most Boolean algebra laws have either an ***AND (product)*** form or an ***OR (sum)*** form. We give the laws with both forms.
  - Since the laws are always true, so X (and Y) could be either 0 or 1

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0 + x = x$
Null Law	$0x = 0$	$1 + x = 1$
Idempotent Law	$xx = x$	$x + x = x$
Inverse Law	$x\bar{x} = 0$	$x + \bar{x} = 1$

Boolean Algebra Laws ('Cont)

- The second group of Boolean laws should be familiar to you from your study of algebra:

Identity Name	AND Form	OR Form
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x + (y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$

Boolean Algebra Laws ('Cont)

- The last group of Boolean laws are perhaps the most useful.
  - If you have studied set theory or formal logic, these laws should be familiar to you.

Identity Name	AND Form	OR Form
Absorption Law	$x(x+y) = x$	$x + xy = x$
DeMorgan's Law	$\overline{(xy)} = \bar{x} + \bar{y}$	$\overline{(x+y)} = \bar{x}\bar{y}$
Double Complement Law	$\overline{(\bar{x})} = x$	

DeMorgan's law

- ☐ DeMorgan's law provides an easy way of finding the negation (complement) of a Boolean function.
- ☐ DeMorgan's law states:

$$\overline{(xy)} = \bar{x} + \bar{y} \quad \text{and} \quad \overline{(x+y)} = \bar{x}\,\bar{y}$$

- ☐ Example — More Examples?
  - ■ I **will** come to school tomorrow if
    - ☐ (**A**) my car is working, and
    - ☐ (**B**) it won't be snowing

=

  - ■ I **won't** come to school tomorrow if
    - ☐ (**A**) my car **is not** working, or
    - ☐ (**B**) it **will** snowing

16

DeMorgan's Law

- ☐ DeMorgan's law can be extended to any number of variables.
  - ■ Replace each variable by its negation (complement)
  - ■ Change all ANDs to ORs and all ORs to ANDs.
- ☐ Let's say F (X, Y, Z) is the following, what is $\bar{F}$?

$$F(X,Y,Z) = (XY) + (\bar{X}Y) + (X\bar{Z})$$

Simplify Boolean function

- Let's use Boolean laws to simplify:

as follows: $F(X,Y,Z) = (X+Y)(X+\overline{Y})(\overline{X\overline{Z}})$

$(X + Y)(X + \overline{Y})\boxed{(\overline{X\overline{Z}})}$	
$(X + Y)(X + \overline{Y})\boxed{(\overline{X} + Z)}$	DeMorgan's Law
	Double complement Law
$\boxed{(XX + X\overline{Y} + YX + Y\overline{Y})}(\overline{X} + Z)$	Distributive Law
$((X + Y\overline{Y}) + X(Y + \overline{Y}))(\overline{X} + Z)$	Commutative and Distributive Laws
$((X + 0) + X(1))(\overline{X} + Z)$	Inverse Law
$X(\overline{X} + Z)$	Idempotent and Identity Laws
$X\overline{X} + XZ$	Distributive Law
$0 + XZ$	Inverse Law
XZ	Identity Law