CLASS : XI SCIENCE

SUBJECT : COMPUTER SCIENCE – PYTHON (083)

NOTE:

*FOR NOW STUDENTS NEED NOT WORRY ABOUT THE TEXT BOOK OR REGISTER.

*KINDLY GO THROUGH THE NOTES .

*THE NOTES SHOULD BE WRITTEN IN ANY ROUGH REGISTER UNDER THE HEADING

UNIT1 Computer Systems and Organisation

CHAPTER 3 BOOLEAN LOGIC

- Boolean logic: OR, AND, NAND, NOR, XOR, NOT, truth tables, De Morgan’s laws

BOOLEAN LOGIC

INTRODUCTION

- ✓ Developed by English Mathematician George Boole in between 1815 - 1864.
- ✓ It is described as an algebra of logic or an algebra of two values i.e True or False.
- ✓ The term logic means a statement having binary decisions i.e True/Yes or False/No.

APPLICATION OF BOOLEAN ALGEBRA

- It is used to perform the logical operations in digital computer.
- In digital computer True represent by '1' (high volt) and False represent by '0' (low volt)
- Logical operations are performed by logical operators. The fundamental logical operators are:
  1. AND (conjunction)
  2. OR (disjunction)
  3. NOT (negation/complement)

AND operator

It performs logical multiplication and denoted by (.) dot.

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

OR operator

It performs logical addition and denoted by (+) plus.

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

NOT operator

It performs logical negation and denoted by (-) bar. It operates on single variable.

X	$\bar{X}$	(means complement of x)
0	1	
1	0	

Truth Table

- Truth table is a table that contains all possible values of logical variables/statements in a Boolean expression.

No. of possible combination =

2^n, where n=number of variables used in a Boolean expression.

Truth Table

The truth table for XY + Z is as follows:

Dec	X	Y	Z	XY	XY+Z
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	0
3	0	1	1	0	1
4	1	0	0	0	0
5	1	0	1	0	1
6	1	1	0	1	1
7	1	1	1	1	1

Tautology & Fallacy

If the output of Boolean expression is always True or 1 is called Tautology.

If the output of Boolean expression is always False or 0 is called Fallacy.

Exercise

1. Evaluate the following Boolean expression using Truth Table.

(a) $X'Y'+X'Y$ (b) $X'YZ'+XY'$

(c) $XY'(Z+YZ')+Z'$

2. Verify that $P+(PQ)'$ is a Tautology.

3. Verify that $(X+Y)'=X'Y'$

Implementation

Boolean Algebra applied in computers electronic circuits. These circuits perform Boolean operations and these are called logic circuits or logic gates.

Logic Gate

A gate is an digital circuit which operates on one or more signals and produce single output.

Gates are digital circuits because the input and output signals are denoted by either 1(high voltage) or 0(low voltage).

There are three basic gates and are:

1. AND gate
2. OR gate
3. NOT gate

AND gate

- The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high.
- AND gate takes two or more input signals and produce only one output signal.

Input A	Input B	Output AB
0	0	0
0	1	0
1	0	0
1	1	1

OR gate

- The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high.
- OR gate also takes two or more input signals and produce only one output signal.

Input A	Input B	Output A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate

- The NOT gate is an electronic circuit that gives a high output (1) if its input is low .
- NOT gate takes only one input signal and produce only one output signal.
- The output of NOT gate is complement of its input.
- It is also called inverter.

Input A	Output $\overline{A}$
0	1
1	0

PRACTICAL APPLICATIONS OF LOGIC GATES

- ✓ Digital device is made up of logic gates only.
- ✓ Logic gates are used to make a few combinational circuits like multiplexers, demultiplexers, encoders, decoders etc.
- ✓ A few arithmetic circuits such as adder, subtracter, comparator etc.
- ✓ You make an Arithmetic and Logic Unit using them.

PRACTICAL APPLICATIONS OF LOGIC GATES

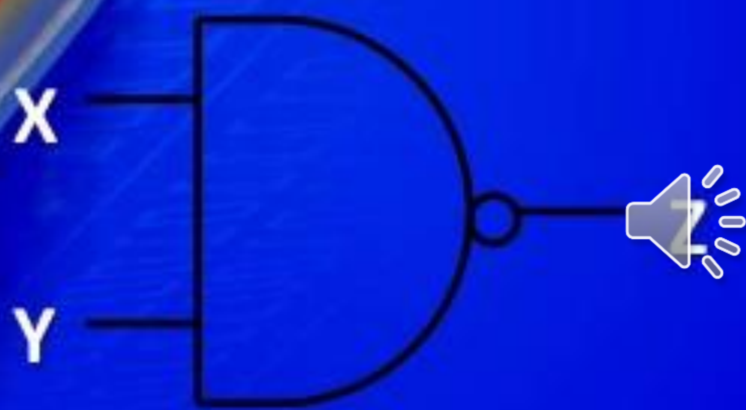
✓ Then flip-flops, counters and registers are made to store data.
Registers, RAM, ROM etc are made with these.

✓ However ROM can be implemented using decoders and multiplexers too.

NAND Gate

Known as a "universal" gate because ANY digital circuit can be implemented with NAND gates alone.

NAND Gate

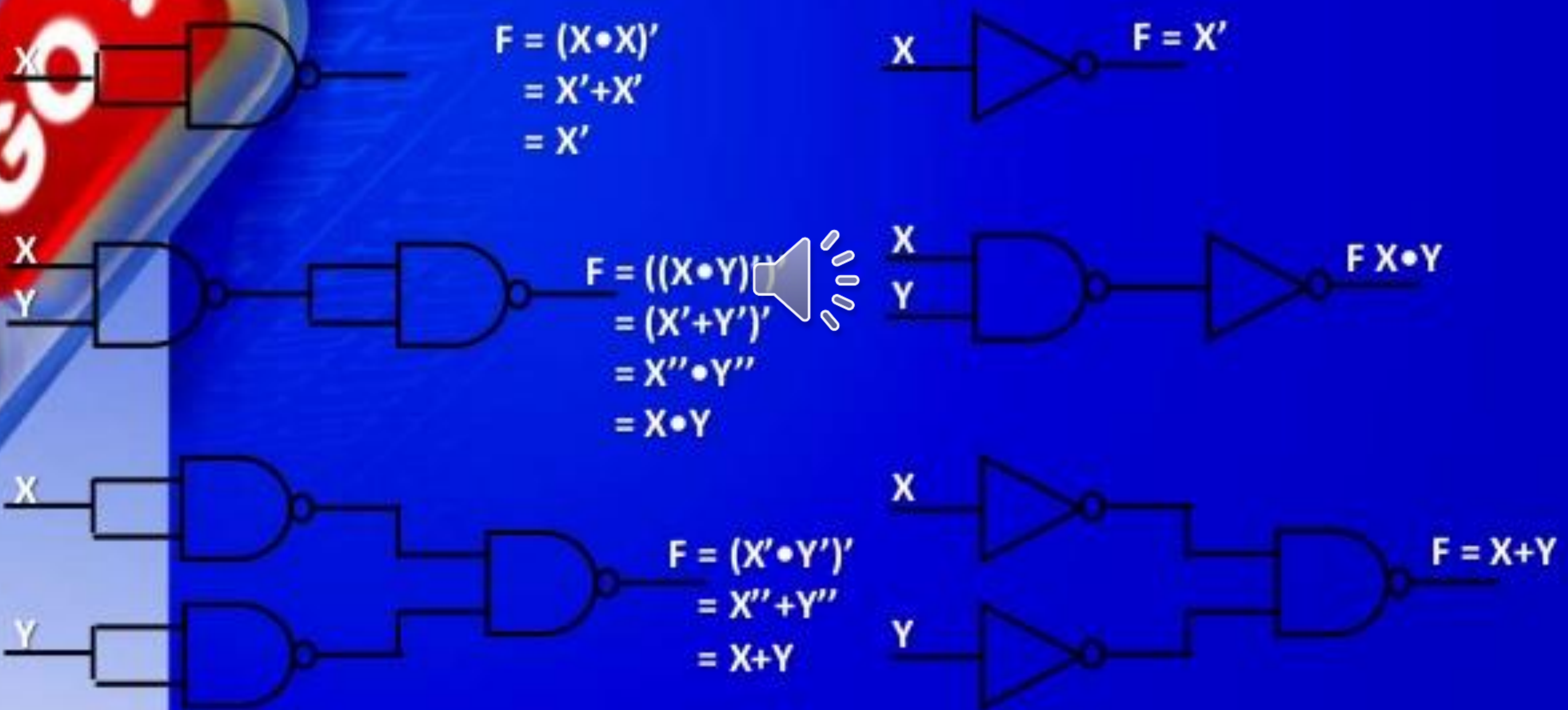


X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

Z = ~(X & Y)

nand(Z,X,Y)

NAND Gate



NOR Gate

Z = ~(X | Y)

nor(Z,X,Y)

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

Exclusive-OR Gate

$Z = X \wedge Y$

xor(Z,X,Y)

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

Exclusive-NOR Gate

Z = ~(X ^ Y)

Z = X ~^ Y

xnor(Z,X,Y)

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

Principal of Duality

In Boolean algebras the duality Principle can be is obtained by interchanging AND and OR operators and replacing 0's by 1's and 1's by 0's. Compare the identities on the left side with the identities on the right.

Example

$X + X'Y = X . X' + Y$

Basic Theorem of Boolean Algebra

T1 : Properties of 0

(a) $0 + A = A$

(b) $0\,A = 0$

T2 : Properties of 1

(a) $1 + A = 1$

(b) $1\,A = A$

Basic Theorem of Boolean Algebra

T3 : Commutative Law

(a) $A + B = B + A$

(b) $A\,B = B\,A$

T4 : Associate Law

(a) $(A + B) + C = A + (B + C)$

(b) $(A\,B)\,C = A\,(B\,C)$

T5 : Distributive Law

(a) $A\,(B + C) = A\,B + A\,C$

(b) $A + (B\,C) = (A + B)\,(A + C)$

(c) $A+A'B = A+B$

Basic Theorem of Boolean Algebra

T6 : Indempotence (Identity) Law

(a) $A + A = A$

(b) $A\,A = A$

T7 : Absorption (Redundance) Law

(a) $A + A\,B = A$

(b) $A\,(A + B) = A$

Basic Theorem of Boolean Algebra

T8 : Complementary Law

(a) $X+X'=1$

(b) $X.X'=0$

T9 : Involution

(a) $x'' = x$

T10 : De Morgan's Theorem

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

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

De Morgan's Theorem 1

Theorem 1 $\overline{A . B} = \overline{A} + \overline{B}$

$$\overline{A.B} = \overline{A} + \overline{B}$$

NAND = Bubbled OR

De Morgan's Theorem 1

Theorem 1 $\overline{A . B} = \overline{A} + \overline{B}$

A	B	$\overline{AB}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

De Morgan's Theorem 2

Theorem 2 $\overline{A + B} = \overline{A} . \overline{B}$

$\overline{A + B} = \overline{A} . \overline{B}$

NOR = Bubbled AND

De Morgan's Theorem 2

Theorem 2 $\overline{A + B} = \overline{A} \,.\, \overline{B}$

A	B	$\overline{A + B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \,.\, \overline{B}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

TO PROVE DE MORGAN'S LAW ALGEBRACIALLY

By Complementarity Law,

$P+P'=1$ and $P\cdot P'=0$

(Note: I shall only be using $P+P'=1$ as its dual is automatically true)

First Law:: De Morgan's 1st law states $(X+Y)'=X'\cdot Y'$

It is sufficient to prove that $(X+Y)+X'\cdot Y'=1$

$LHS=Y+(X+X'\cdot Y')$

$=Y+X+Y'$

$=(Y+Y')+X$

$=1+X$

$=1=RHS$

Second Law:: De Morgan's 2nd Law states that $(X\cdot Y)'=X'+Y'$

It is sufficient to prove that $X\cdot Y+(X'+Y')=1$

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

$Y' +(X'+X).(X'+Y)$

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

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

$=1+X'$

$=1=RHS$

Hence, De Morgan's Laws are verified algebraically.