JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA
DEPARMENT OF COMPUTER SCIENCE & ENGINEERING
Course: Computer Organization & Architecture Lab
Course Code: CS208/18B17CI474
B. Tech. (CSE IV/VI Sem.)

Experiment # 1

Aim: Design of basic digital circuits using logic gates.

A *logic gate* is an elementary building block of a digital circuit. It performs a *logical* operation on one or more binary inputs and produces a single binary output 1 or a 0 when its logic requirements are met. The basic *logic gates* are categorized into seven: AND, OR, XOR, NAND, NOR, XNOR and NOT each of these are represented with the help of a distinct graphic symbol. NAND and NOR gates are known as **universal gates** because all other gates can be realized by using these gates. A *logic gate* often uses diodes or transistors that act (in '**on**' and '**off**' states) like electronic switches. Gates receive their input (binary information of 0 or 1) with the aid of physical quantities such as electric signals. For instance, an electric signal of lower voltage (e. g. 0 volts) and of higher voltage (e.g. 5 volts) may be used to represent a binary zero and one respectively. Similarly, the output from gates may be represented using some recognizable states (voltages) of an electric signal. In a computer system, logic gates are used to store the data, perform basic arithmetic/logical operations and other various types of manipulations in the bits.

Exercise#1: Design two inputs and five outputs All-in-One logic gate circuit shown in Fig.1 using Logisim simulator with (i) data width 1 (ii) data width 4.

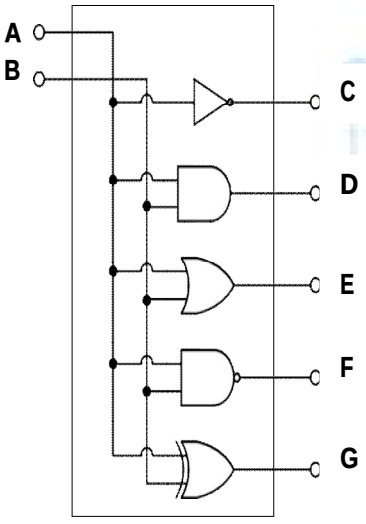

Fig. 1: Two inputs and five outputs All-in-One logic gate diagram.

$NOT\ gate\colon C = \bar{A}$

$AND\ gate\colon D = A.B$

$OR\ gate\colon E = A + B$

$NAND\ gate\colon F = \overline{A.B}$

$XOR\ gate\colon G = A \oplus B$

Boolean Expressions

Inputs		Outputs				
A	B	C	D	E	F	G
0	0	1	0	0	1	0
1	0	0	0	1	1	1
0	1	1	0	1	1	1
1	1	0	1	1	0	0

Truth Table

Exercise#2: Design two inputs and one output All-in-One logic gate (**without multiplexer**) diagram shown in Fig.2 using Logisim simulator with (i) data width 1 (ii) data width 8.

This All-in-One logic gate is a double input, single output gate that can be instructed to perform four different logic operations by placing a control value on the inputs X and Y. The instruction to this gate is provided by the operation select bits, which thus determine how the gate will act. Figure 2 shows the block diagram of such a gate. A and B form the data inputs and C is the single output. X and Y are the operation select lines. Total number of 2^n functions can be performed with n select lines.

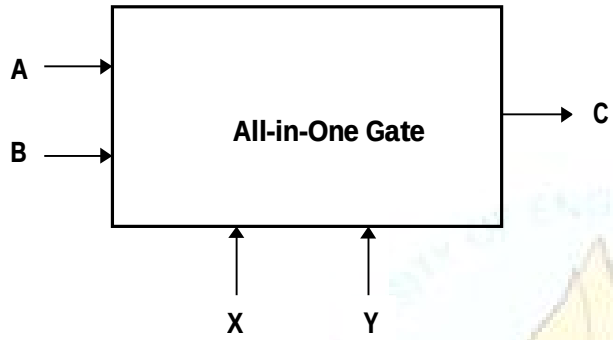

Fig. 2: Two inputs and one outputs All-in-One logic gate block diagram.

Select lines		Functions
X	Y	C
0	0	$AND\ gate\colon\ C = A.B$
1	0	$OR\ gate\colon\ C = A + B$
0	1	$NOR\ gate\colon\ C = \overline{A + B}$
1	1	$NAND\ gate\colon\ C = \overline{A.B}$

Truth Table

Design Steps:

- Find complete truth table with four inputs (A, B, X, Y) and one output C.
- Derive a Boolean expression for output C using Karnaugh (K) map.
- Implement the logic diagram as per derived Boolean expression and verify it.

Exercise#3: Design a three-input majority detector combinational digital circuit using Logisim simulator which shows output equal to 1 if the input variables have more 1's than 0's, the output is 0 otherwise.

Design Steps:

- Find truth table with three inputs and one output.
- Derive a Boolean expression for output using K-map.
- Implement the logic diagram as per derived Boolean expression and verify it.

Exercise#4: Design a combinational circuit with three inputs and three outputs. When the input is 0, 1, 2, or 3, the output is one greater than the input and when the input is 4, 5, 6, or 7, the output is one less than the input. Display the input and output digits using **Hex digit display with splitter.**

Design Steps:

- Find truth table with three inputs and three outputs.
- Derive a Boolean expression for each output using K-map.
- Implement the logic diagram as per derived Boolean expressions and verify it.
- Add one splitter of size 4x4 and one hex displays both at input and output side.
- Connect input bits with input splitter in order (i.e. input LSB with LSB of splitter and so on, ground MSB of splitter). In the same way, make the connection at output side splitter.