DEPARTMENT OF ELECTRONCS AND COMMUNICATION ENGINEERING

18ECC103J DIGITAL ELECTRONICS PRINCIPLES

LABORATORY MANUAL

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

FACULTY OF ENGINEERING AND TECHNOLOGY
S.R.M. Nagar, Kattankolathur-603203
Kancheepuram District

CONTENTS

S.R.M Institute of Science and Technology
Faculty of Engineering and Technology
Department of Electronics and Communication Engineering

Sub Code : 18ECC103J Year/Semester: II / III
Sub Title : Digital Electronics Principles Course Time: Sep'21–Jan/Feb'22
Pre_requisite: 18EES101J Basic Electronics Engineering

S. No.	Experiments Detail	Equipments Required	Components Required
1.	To study the truth table of logic gates	Digital IC Trainer Kit	All Digital gate IC's
2.	To design and verify the truth table for half adder, full adder,Half subtractor and Full subtractor	Digital IC Trainer Kit	74LS08 IC – Quad(2-input AND gates)Positive Logic 74LS32 IC – Quad(2-input OR gates)Positive Logic 74LS86 IC – Quad(2-input ExOR gates)Positive Logic
3	To design a circuit for comparing the magnitude of two-bit numbers	Digital IC Trainer Kit	74LS00 IC – Quad(2-Input NAND Gates)Positive Logic 74LS10 IC – Triple(3-Input NAND Gates)Positive Logic 74LS02 IC – Quad(2-Input NOR Gates)Positive Logic 74LS04 IC - Hex(Inverter)Positive Logic 74LS08 IC – Quad(2-input AND gates)Positive Logic 74LS32 IC – Quad(2-input OR gates)Positive Logic 74LS86 IC – Quad(2-input ExOR gates)Positive Logic
4	To design and verify the truth table for 8-3 Encoder &3-8 Decoder logic circuit. To design and verify the Truth table	Digital IC Trainer Kit	74LS04 IC - Hex(Inverter)Positive Logic 74LS08 IC – Quad(2-input AND gates)Positive Logic 74LS11 IC - Triple(3-input AND gates)Positive Logic 74LS32 IC – Quad(2-input OR gates)Positive Logic
5	To implement and verify the functional table of 4:1Multiplexer and 1:4 De-multiplexer	Digital IC Trainer Kit	74LS04 IC - Hex(Inverter)Positive Logic 74LS08 IC – Quad(2-input AND gates)Positive Logic 74LS11 IC - Triple(3-input AND gates)Positive Logic 74LS32 IC – Quad(2-input OR gates)Positive Logic
6	To construct a variety of Code converter Circuits.	Digital IC Trainer Kit	74LS86 IC – Quad(2-input Ex-OR gates)Positive Logic
7.	To implement the given SOP equation in Multiplexer and decoder	Digital IC Trainer Kit	IC74LS151- 8:1 Multiplexer IC74LS138-3:8 Decoder
8.	To familiarize with circuit implementations using ICs and test the behaviour of Verify the behaviour of Flip-flops.	Digital IC Trainer Kit	74LS73 IC – Dual(positive-edge triggered JK flip flop) 74LS74 IC –Dual(positive-edge triggered D flip flop)
9.	To design and verify the timing diagram of n bit Ripple Counter and to design Mod-n asynchronous Counter	Digital IC Trainer Kit	74LS73 IC – Dual(positive-edge triggered JK flip flop) 74LS00 IC – Quad(2-Input NAND Gates)Posit ive Logic
10.	To design n-bit Synchronous Counter from the given logic diagram and to implement it using Flip-flops.	Digital IC Trainer Kit	74LS08 IC – Quad(2-input AND gates)Positive Logic 74LS32 IC – Quad(2-input OR gates)Positive Logic 74LS04 IC – Hex(Inverter)Positive Logic 74LS73 IC – Dual(positive-edge triggered JK flip flop)
11.	To understand the operation of Shift Registers and Shift Register counters.	Digital IC Trainer Kit	74LS74 IC –Dual(positive-edge triggered D flip flop)

Laboratory Report Cover Sheet

SRM Institute of Science and Technology
Faculty of Engineering and Technology
Department of Electronics and Communication Engineering

18ECC103J Digital Electronics Principles
Third Semester, 2021-22 (odd semester)

Name	:	
Register No.	:	
Day / Session	:	
Venue	:	
Title of Experiment	:	
Date of Conduction	:	
Date of Submission	:	

Particulars	**Max. Marks**	**Marks Obtained**
Pre-lab questions	10	
In-lab experiment	20	
Post-lab questions	10	
Total	40	

REPORT VERIFICATION

Date	:	
Staff Name	:	
Signature	:	

Digital IC Trainer KIT Operation (Model No: 9002)

Specification : Digital IC Trainer Kit Model No. 9002 is available with 10 nos. of TTL compatible logic level inputs, TTL logic selectable by a toggle switch, Logic HIGH and logic LOW are displayed by LED, 10 nos of Logic Output indictors, Four crystal generated clock output of 1KHz, 100Hz, 10Hz and 1Hz. Facility for single pulse generation by a push button switch, Logic probe to check logic LOW, logic HIGH and pulse, Four seven segment displays with BCD inputs, Sockets onboard to fix the IC`s : 16pin-4nos. Built-in Power supply : 5V, 1Amp, +12V, 250mA

How to use the IC Trainer Kit?

Connect the 230 volts AC power supply and switch 'ON' the Toggle switch 'ON' the left side of the Top Panel, LED will glow. Digital IC Trainer Kit is ready for use. Select the TTL IC to be used for the experiment. Insert the IC properly in the breadboard/ZIF socket(lock the ZIF by moving lever upwards), Know the biasing voltage required for different families of IC's and connect power supply voltage and ground terminals to the respective pins of the IC.

Inputs such as logic clock, of different frequency, monopulse, logic levels, BCD inputs, can be selected from the patch panel. Outputs such as LED indicator, seven segment digital display can be selected depending upon the requirement.

Connect the pin connection of IC using wires as per the logic diagram, after verifying connection switch on the supply of IC Trainer Kit and verify the operation the circuit with the help of truth table.

Specification of Digital IC Trainer Kit

- Solderless bread board
- Logic input switches – 10 nos
- Logic output indicators – 10 nos
- Sockets onboard to fix the IC`s : 16 pin -4nos
- 3-digit 7 Segment display with BCD inputs(A,B,C,D)
- 26 pin flat-cable connector
- TTL clock – 1Hz, 10Hz, 100Hz, 1KHz and monopulse.
- Built-in Power supply – 5V,1A; +/- 12V,200Ma

Introduction to LOGISIM

LOGISIM

Logisim is a logic simulator which permits circuits to be designed and simulated using a graphical user interface. Simulation can be performed at varying degrees of physical abstraction, such as at the transistor level, gate level, register-transfer level (RTL), electronic system-level (ESL), or behavioural level. Logisim is free software designed to run under the Microsoft Windows, OS X, and Linux platforms. Its code is entirely in Java using the Swing graphical user interface library. The primary developer, Carl Burch, has worked on Logisim since its inception in 2001.

The software is used most often by students to design and experiment with digital circuits in simulation. Circuits are designed in Logisim using a graphical user interface similar to traditional drawing programs, an interface also found in many other simulators.

Logisim not only lets you simulate your logical circuits but also edit circuits while simulation is running. Therefore you can immediately rectify flaws without the need of separate edits and loading.

Using logic simulator software tools, electronic designers can digitally interpret the behaviour of circuits they create. This offers a great way to ensure accuracy of their design and correctness of the applied logic before jumping into the actual production. One such popular logic simulation software is Logisim.

Procedure to Design a Circuit

Users can create circuits with simple drag actions of the mouse. Logisim enables them to create circuits of any scale, including sub-circuits as well as complex structures. Colour-coded wires impart clarity to the representation.

Logisim opens as a simple workspace with panels on the top and the left. The workspace is dotted. You can drag logic gates, components and wires and connect them inside the workspace to create your circuit.

Connecting the components together is simple. Simply dragging the mouse between the intended components joins them with a digital wire. The wires can run horizontally and vertically.

All the elements of the circuit are colour-coded. For example, input ports are represented in green and output ports in blue. Also, active wires during simulation turn bright green, while wires with no electricity remain deeper green by default. Any flawed wiring is indicated in red. All input and output ports have a designated value of either 0 or 1 assigned to them, indicating their active status.

The upper edge of the software contains two menu bars. The topmost bar contains File, Edit, Project, Simulate, Windows and Help menus. You can perform different actions from these menus, such as open or continue a project, edit a circuit, access tutorials or run a simulation. Project menu contains many sub-menus like Add Circuit, Load or Unload Library, View Simulation Tree, Analyse Circuit and so on. Under Analyse Circuit, you can check values of the input, output and truth table that the logic circuit follows.

The other toolbar consists of some useful tools and shortcuts. Finger icon lets you toggle between a cursor and a finger, and change values in the circuit. So, if you click an input value of 0 with the finger, it will change to 1, meaning the input is receiving electric flow.

Cursor is the default mouse pointer. You can use it for interactive actions on the workspace, like selecting and moving components, connecting or deleting wires, resetting component properties and more. The icon for textbox lets you add labels to your circuit for convenience.

Following the icons are symbols of five of the most used components in an electronic circuit input, output, as well as NOT, AND OR logic gates. Use the cursor to drag these elements on the toolbar and drop them on the workspace for use in the circuit as desired.

The left panel of the software consists of libraries and properties windows. The first folder represents the project. It is followed by folders called Base, Gates, Plexers, Arithmetic, Memory and Input/Output. All folders contain components from the library that can be used for building different kinds of circuits and sub-circuits. These components are categorised in folders for easy retrieval.

For instance, under Gates folder, along with the common three logic gates mentioned earlier, you get NAND, NOR, XOR, XNOR, Buffer, Odd Parity, Even Parity and so on.

Below the folders comes Properties window. Here you can edit properties of all components. The window displays circuit name, text label assigned, direction and font of the label, number of input ports of logic gates and so on. To modify any component, just click the component in the circuit with the cursor and do the modifications in this windows. Suppose you dragged in an

AND gate that has two input ports by default. If you need four input ports, click the AND gate symbol and in the properties window, click to change the number in Inputs entry field. The logic gate in the circuit will update immediately.

Features of Logisim

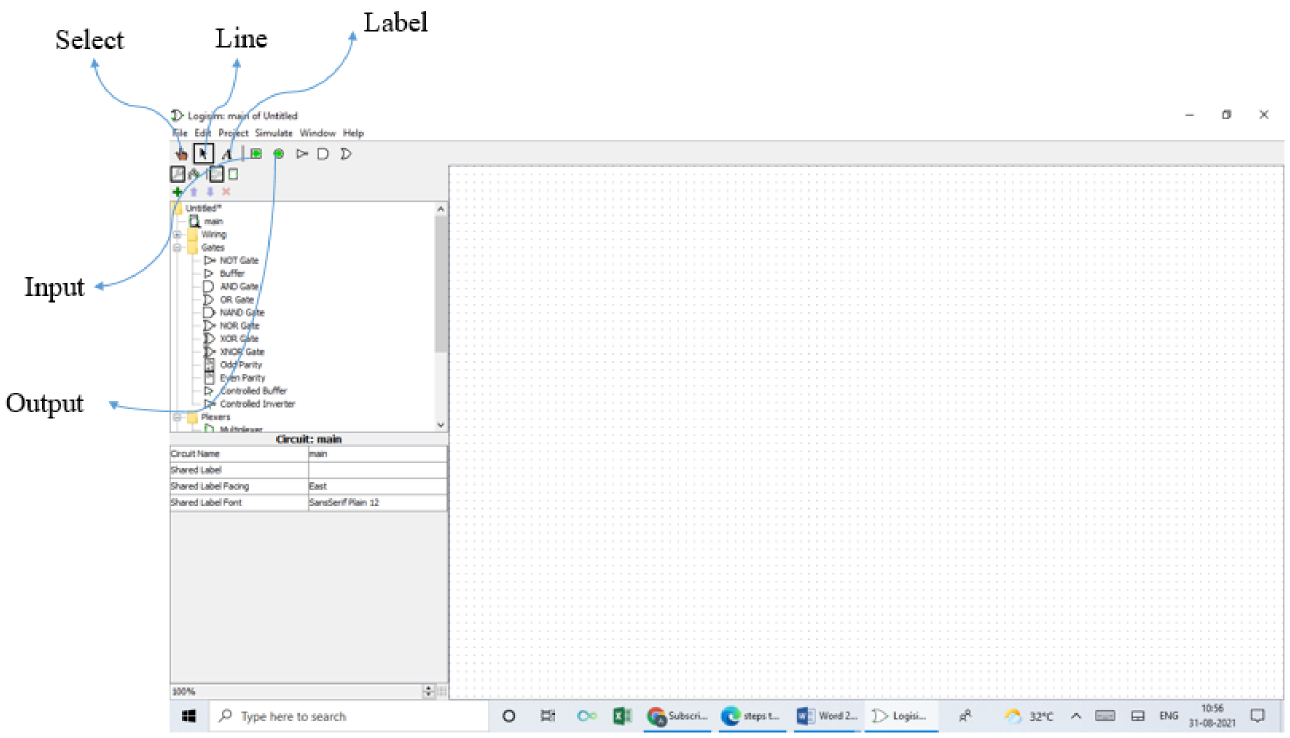
Logisim not only lets you simulate your logical circuits but also edit circuits while simulation is running. Therefore you can immediately rectify flaws without the need of separate edits and loading. Another useful feature is that any circuit built in Logisim software can be used as an existing library component. That means, you can use the created circuit as a sub-circuit of another bigger one. In addition, you can save an image file of the circuit created. Go to File->Export Image and then select the file type of your choice.

There are many more features that you can explore in Logisim be it the different components or the different simulation settings.

AND Gate- Simulation

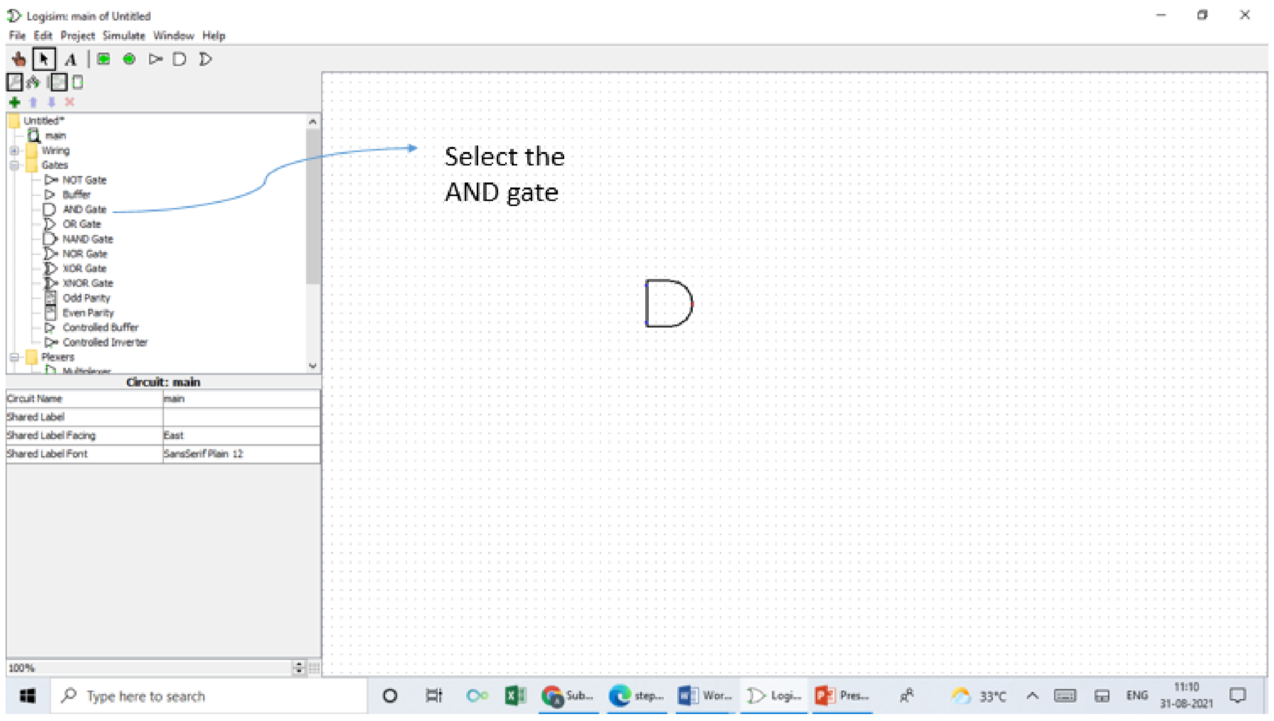
Step by Step Procedure

Open the LOGISIM Software



Circuit: main

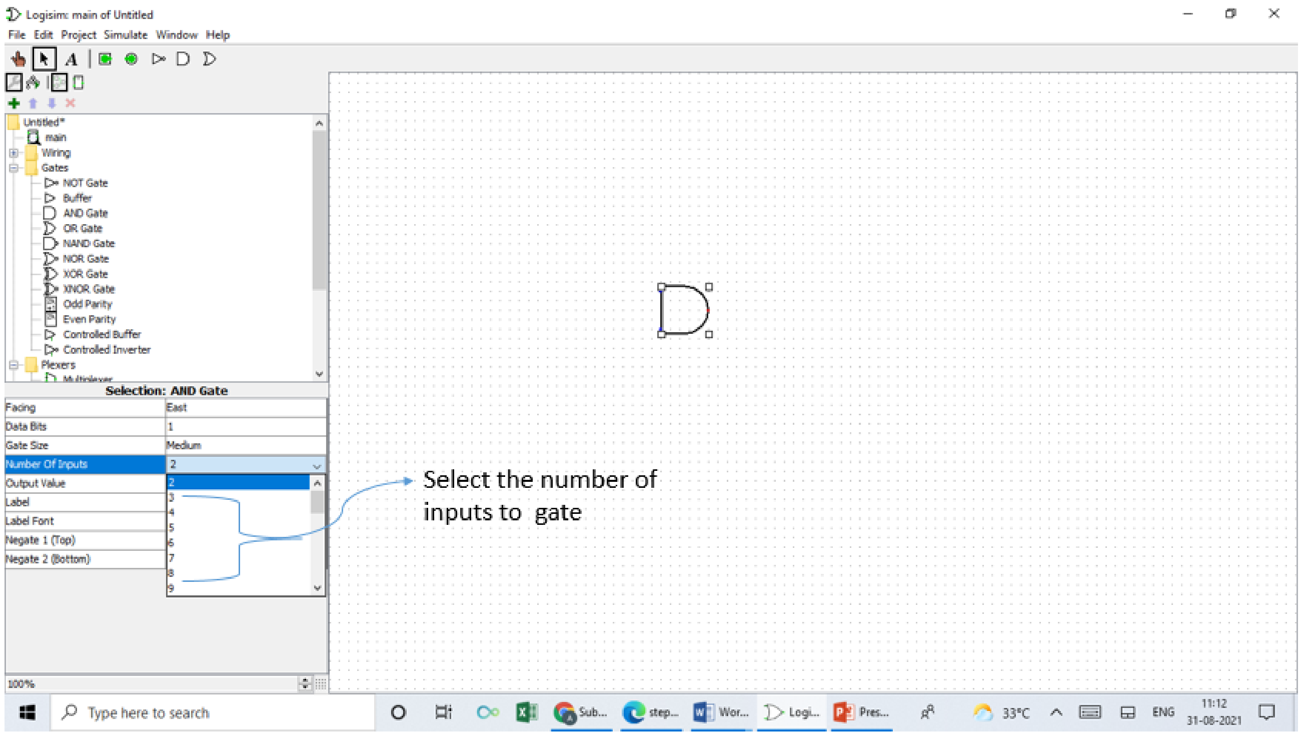
Circuit Name	main
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12



Select the AND gate

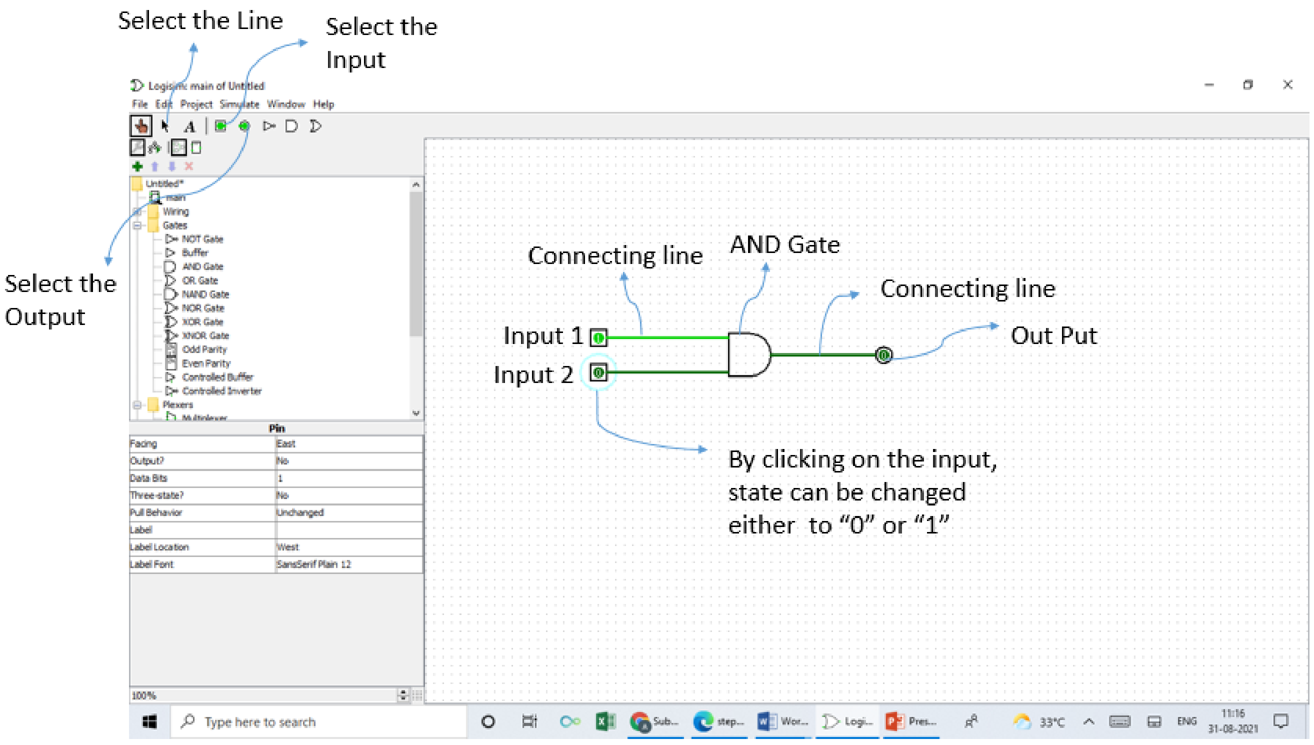
Circuit: main

Circuit Name	main
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12



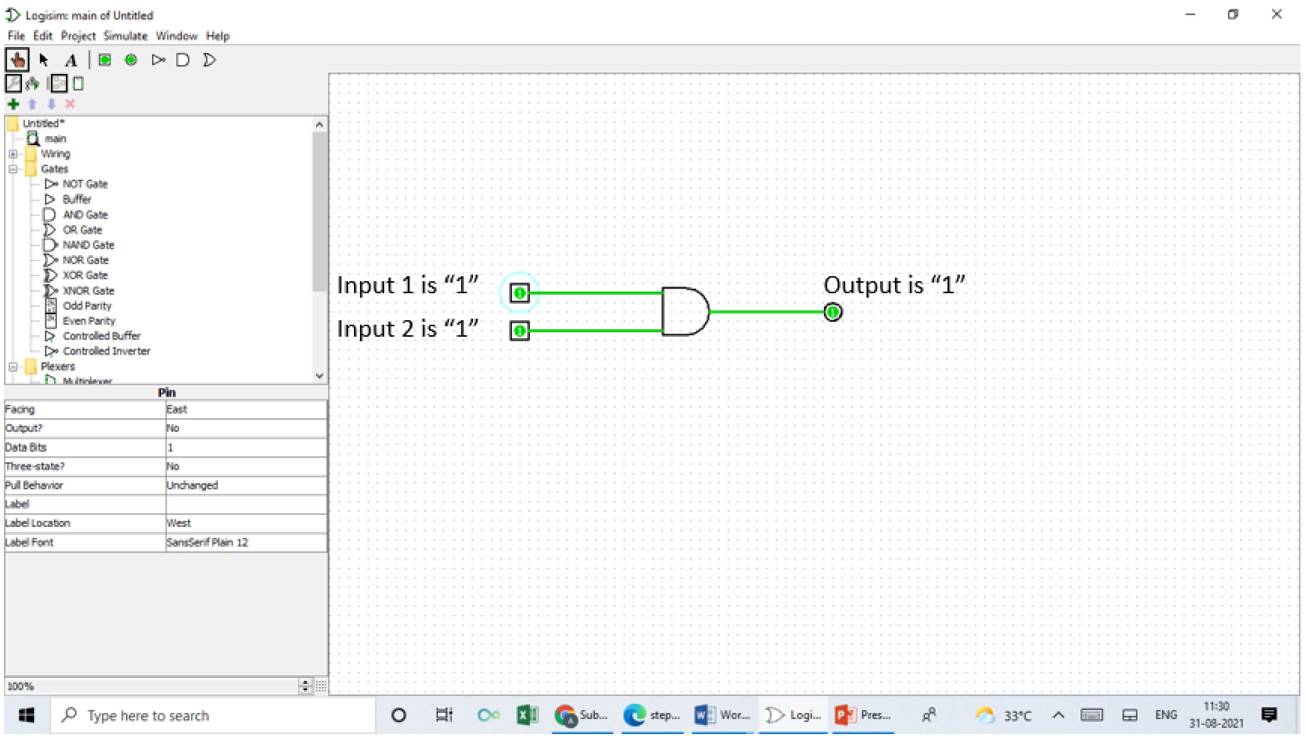
Selection: AND Gate

Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	
Label	
Label Font	
Negate 1 (Top)	
Negate 2 (Bottom)	



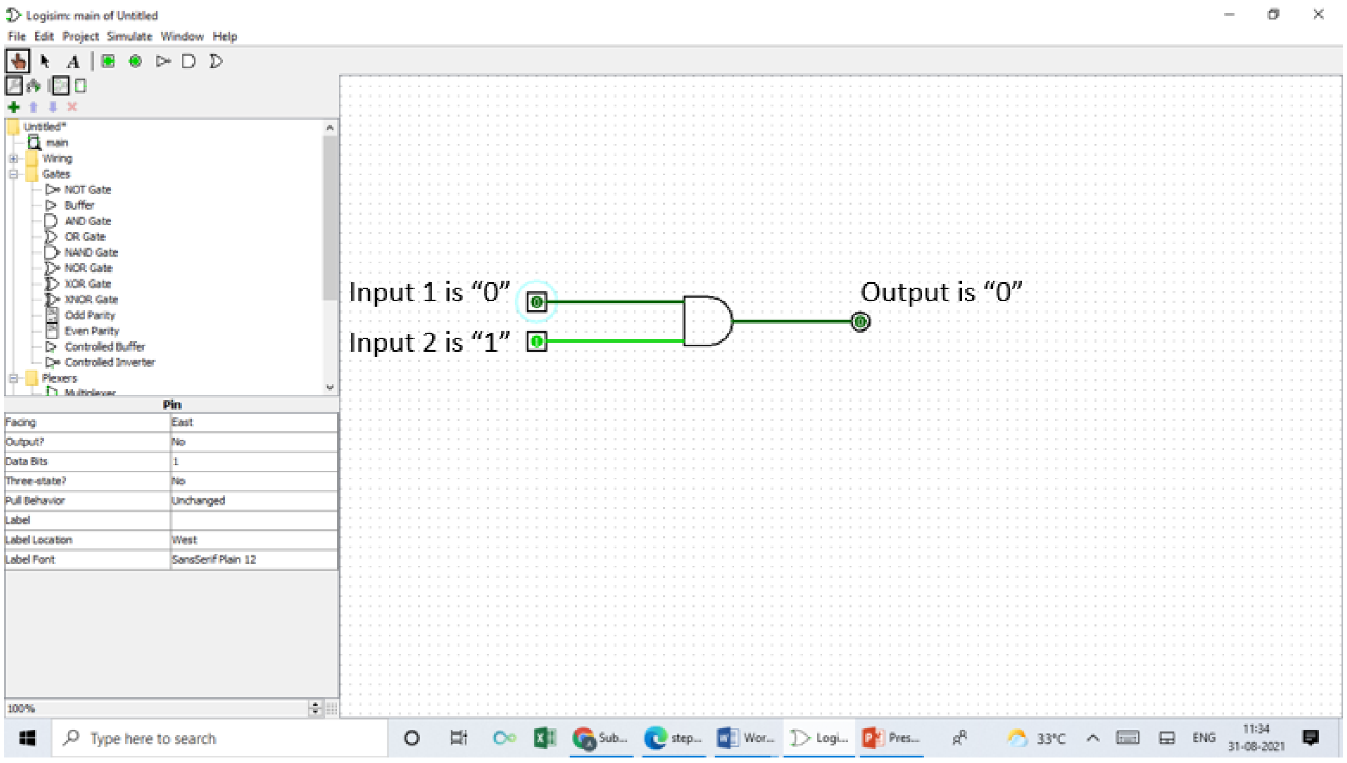
Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12



Pin

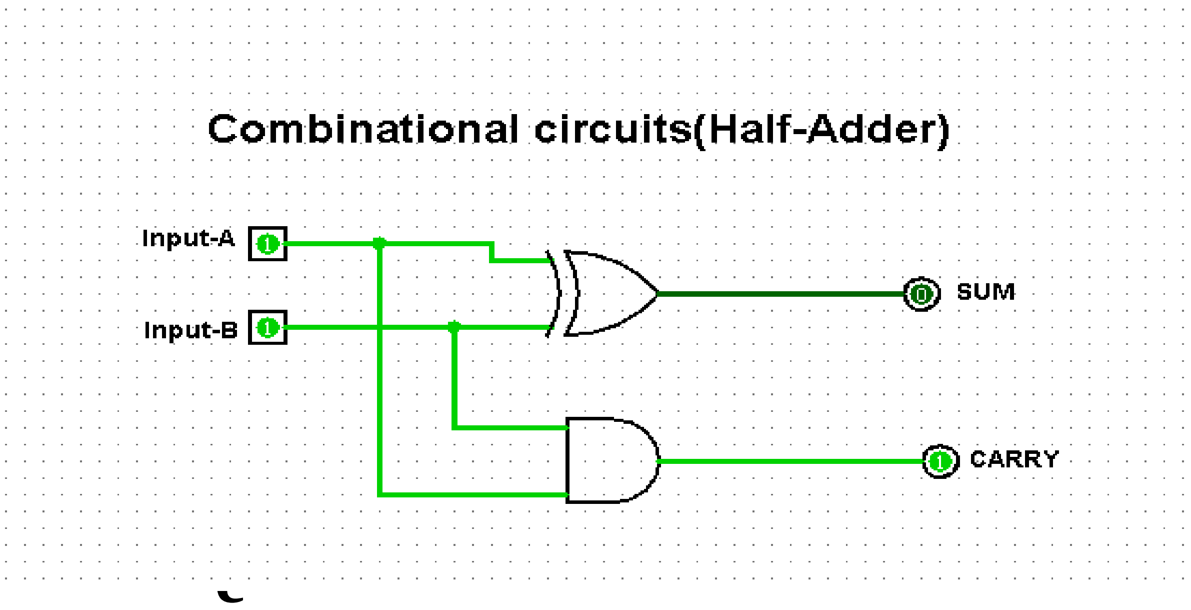
Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

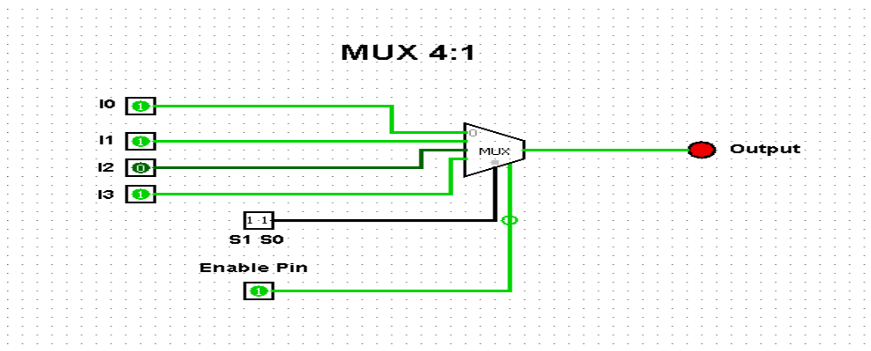


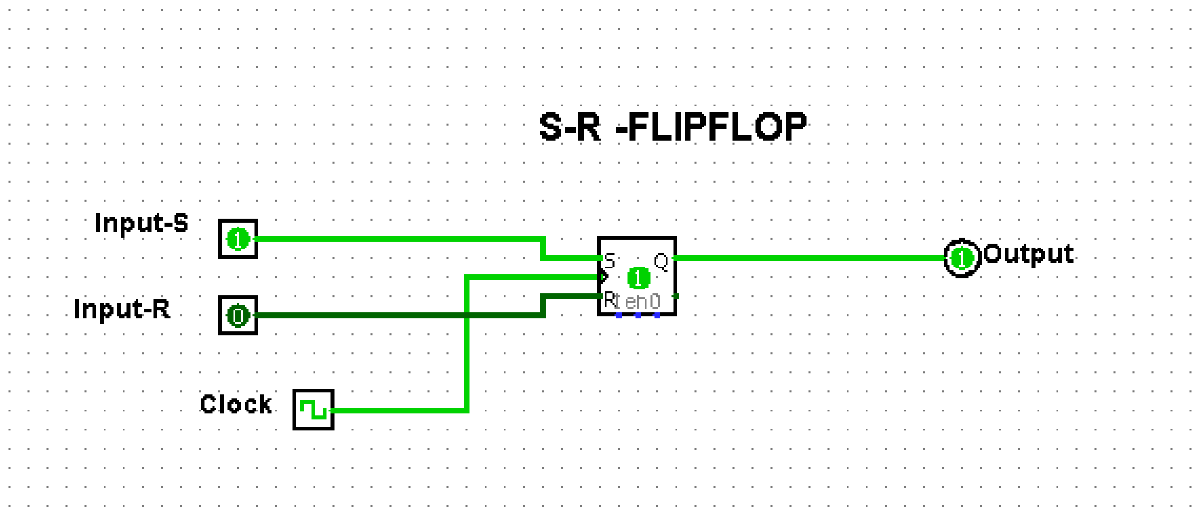
Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

Example Circuits







Introduction of Digital IC's

Logic Gates

Digital logic devices are the circuits that electronically perform logic operations on binary variables. The binary information is represented by high and low voltage levels, which the device processes electronically. The devices that perform the simplest of the logic operations (such as AND, OR, NAND, etc.) are called gates. For example, an AND gate electronically computes the AND of the voltage encoded binary signals appearing at its inputs and presents the voltage encoded result at its output.

The digital logic circuits used in this laboratory are contained in integrated circuit (IC) packages, with generally 14 or 16 pins for electrical connections. Each IC is labeled (usually with an 74LSxx number) to identify the logic it performs. The logic diagrams and pin connections for these IC's are described in the TTL Data Book by Texas Instruments1.

The transistor-transistor logic(TTL) IC's used in this laboratory require a 5.0 volt power supply for operation. TTL inputs require a voltage greater than 2 volts to represent a binary 1 and a voltage less than 0.8 volts to represent a binary 0.

Pin numbering is standard on IC's. Figure 1 illustrates the pin numbering for a 14-pin dual in-line package (DIP). With the IC oriented as shown, the numbering starts at the top left and proceeds counter-clockwise around the chip:

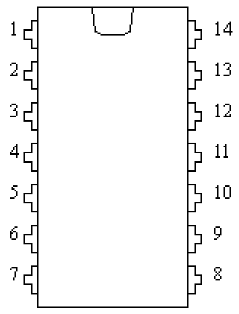


Figure 1: A 14-pin DIP

To construct circuits with IC's, a circuit board that allows easy connections to IC pins should be used. The circuit board contains rows of solder less tie points, a 5-volt power supply, a common circuit point (ground), toggle switches for input, and LEDs (light emitting diodes) for output.

LOGIC GATES AND THEIR PROPERTIES

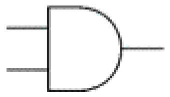
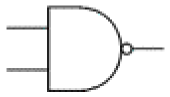
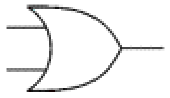
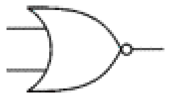
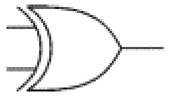
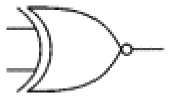
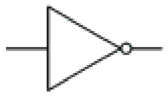
Name	Symbol	Description
AND gate		Output is 1 only if all the inputs are 1. AND gates can have two, three, or more inputs.
NAND gate		"NOT-AND gate"--opposite of AND gate--output is 1 only if all the inputs are NOT 1. Output is only 0 when all the inputs are 1. NAND gates can have more than two inputs.
OR gate		Output is 1 if either of the inputs are 1. Output is only 0 if both of the inputs are 0. OR gates can have more than two inputs.
NOR gate		"NOT-OR gate"--opposite of OR gate--output is only 1 if both of the inputs are 0. If either of the inputs, or both the inputs, are 1, then the output is 0. NOR gates can have more than two inputs.
EX-OR gate		"Exclusive-OR gate"--output is only 1 if either of the inputs are 1, but 0 when both the inputs are 0 or when both the inputs are 1.
EX-NOR gate		"Exclusive-NOR gate"--opposite of EX-OR gate--output is only 1 when both the inputs are 0 or both the inputs are 1. Output is 0 when either of the inputs is 1.
NOT gate		Also known as INVERTER gate.

Figure 2: Logic Gates: Its symbols and description

Truth Table of 2-input logic gates

AND gate

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

NAND gate

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

OR gate

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

NOR gate

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

EX-OR gate

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

EX-NOR gate

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

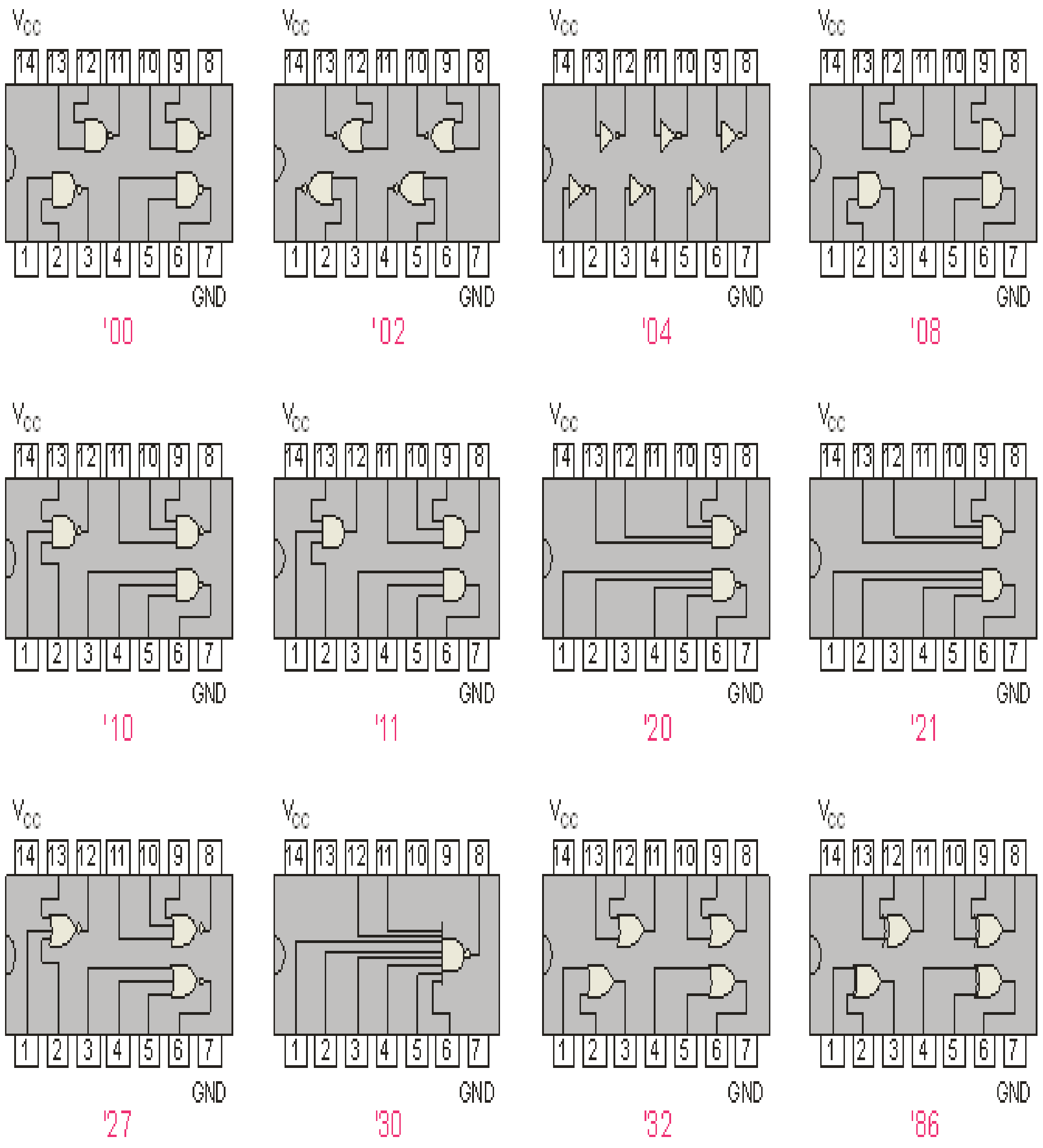


Figure 3: Pin configuration diagrams of some commonly used logic gates

Lab 1: Study of Logic gate

1.1 Aim

To Study and verify the truth table for Logic gates.

1.2 Hardware Requirement

Equipment	:	Digital IC Trainer Kit
Software	:	Logisim
Discrete Components	:	74LS08 Quad 2 input AND gate 74LS32 Quad 2 input OR gate 74LS00 Quad 2 input NAND gate 74LS02 Quad 2 input NOR gate 74LS86 Quad 2 input XOR gate 74LS04 Hex 1-Input NOT gate

1.3 Theory

In Boolean algebra, there are three basic operations, +,*, ~. Which are analogous to disjunction, conjunction, and negation in propositional logic. Each of these operations has a corresponding logic gate. Apart from these there are a few other logic gates as well.

Logic Gates –

- **AND gate (.) –** The AND gate gives an output of 1 if both the two inputs are 1, it gives 0 otherwise.
- **OR gate (+) –** The OR gate gives an output of 1 if either of the two inputs are 1, it gives 0 otherwise.
- **NOT gate (‘) –** The NOT gate gives an output of 1 input is 0 and vice-versa.
-

Logic gates are the heart of digital electronics. A gate is an electronic device which is used to compute a function on a two valued signal. Logic gates are the basic building block of digital circuits.

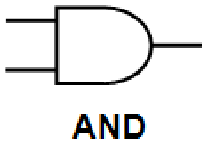
Basically, all logic gates have one output and two inputs. Some logic gates like NOT gate or Inverter has only one input and one output. The inputs of the logic gates are designed to receive only binary data (only low 0 or high 1) by receiving the voltage input.

The low logic level represents Zero volts and high logic level represents 3 or 5 volts positive supply voltage.

We can connect any number of logic gates to design a required digital circuit. Practically, we implement the large number of logic gates in ICs, by which we can save the physical space occupied by the large number of logic gates. We can also perform complicated operations at high speeds by using integrated circuits (IC).

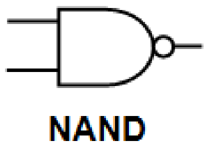
By combining logic gates, we can design many specific circuits like flip flops, latches, multiplexers, shift registers etc.

1.4 Circuit Schematic and Truth Table



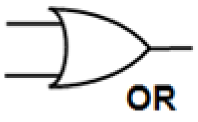
Inputs		Output
A	B	F
0	0	0
1	0	0
0	1	0
1	1	1

Figure 1.1 AND GATE



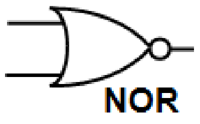
Inputs		Output
A	B	F
0	0	1
1	0	1
0	1	1
1	1	0

Figure 1.2 NAND GATE



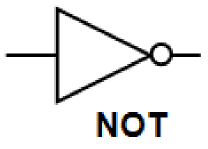
Inputs		Output
A	B	F
0	0	0
1	0	1
0	1	1
1	1	1

Figure 1.3 OR GATE



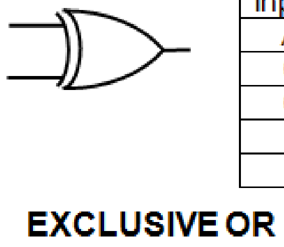
Inputs		Output
A	B	F
0	0	1
1	0	0
0	1	0
1	1	0

Figure 1.4 NOR GATE



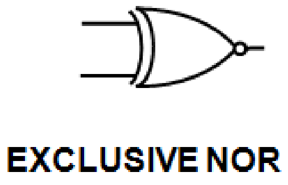
Input	Output
I	F
0	1
1	0

Figure 1.5 NOT GATE



Inputs		Output
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Figure 1.6 XOR GATE



Inputs		Output
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Figure 1.5 XNOR GATE

1.5 Pre lab Questions

1. Implement AND gate using NAND and NOR gate.
2. Implement OR gate using NAND and NOR gate.
3. Draw the truth for the given function X=A.B+C' ?
4. Which the following two input logic function can act as the universal gate? (i) AB (ii) A+B (iii) (A+B)'

1.6 Lab Procedure

Software

1. Double click logisim-win-2.7.1.exe on your desktop to open Logisim.
2. Open a new project named Digital Lab.
3. In the main circuit window, select the gates as per the logic schematic and place it.
4. Change the number of inputs for each gate.
5. Select input/output devices and make connections to make circuits.

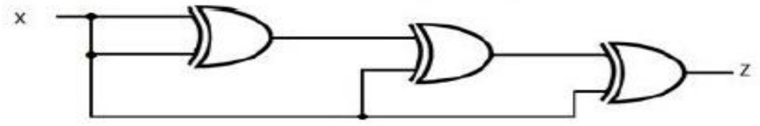
6. Save your file as Logic gates (eg:-and, or…) under the project from the File menu.
7. Once the circuit get done, test it by changing the input values and verify the output values as per the truth table.
8. Check the auto generated truth table from the Project menu and select Analyze Circuit and then select Table.

Trainer Kit

1. Connect the circuit as shown in the figures.
2. Connect the power supply and ground to the respective pin numbers to all the IC's.
3. Give the inputs and verify the output in the truth table.

1.7 Post lab Questions

1. Implement with minimum number of 2-input NOR gates to get a 2-input XNOR function.
2. For the following digital circuit, the output Z is given by:



3. Implement the following function F=ABC' only with 2- input NAND gate.

1.8 Result:

Lab 2: Design and implement Adder and Subtractor using logic gates

2.1 Aim

To design and verify the truth table for adders & Subtractors..

2.2 Software and Hardware Requirement

Software	:	Logisim
Discrete Components:		74LS08 Quad 2 input AND gate
		74LS32 Quad 2 input OR gate
		74LS86 Quad 2 input XOR gate
		74LS04 Hex 1-Input NOT gate
Equipment	:	Digital IC Trainer Kit

2.3 Theory

(i) Half Adder

A Binary adder is a circuit which is able to add together two binary numbers. The half adder adds two binary digits an addend and an augend to produce a sum and carry. The sum can be implemented by using an Exclusive OR gate and an AND gate can be used for carry generation.

The Boolean expression for the sum and carry are

Sum $S = A \oplus B$

Carry $C = A.B$

(ii) Full Adder

The full adder adds an addend, an augend and carry input generated by the previous stage addition. It has two outputs, sum and carry. Full adder circuit can be implemented using AND,OR and EX-OR gates. Full adder circuit can also be implemented with the help of two half adder circuits. The first half adder is used to add two inputs and generate sum and carry output. Then second half adder combines the sum and carry input and generate final sum and carry out.

The sum and carry can be expressed as

Sum $S = A \oplus B \oplus C_{in}$

Carry $C = (A \oplus B)C_{in} + AB = AB + BC_{in} + AC_{in}$

(iii) Half Subtractors

A half subtractor is a multiple output combinational logic network that does the subtraction of two bits of binary data. It has input variables and two output variables. Two inputs are corresponding to two input bits and two output variables corresponds to the difference bit and borrow bit. The binary subtraction is also performed by the Ex-OR gate with additional circuitry to perform the borrow operation. Thus, a half subtractor is designed by an Ex-OR gate including AND gate with A input complemented before fed to the gate.

The Boolean expression for the Difference and Barrow is

Difference D = $A \oplus B$

Barrow $B_o = \overline{A}\,\overline{A}.\ B$

(iv) Full Subtractor

A combinational logic circuit performs a subtraction between the two binary bits by considering borrow of the lower significant stage is called as the full subtractor. It has three input terminals in which two terminals corresponds to the two bits to be subtracted (minuend A and subtrahend B), and a borrow bit Bi corresponds to the borrow operation. There are two outputs, one corresponds to the difference D output and other borrow output Bo as shown in figure along with truth table.

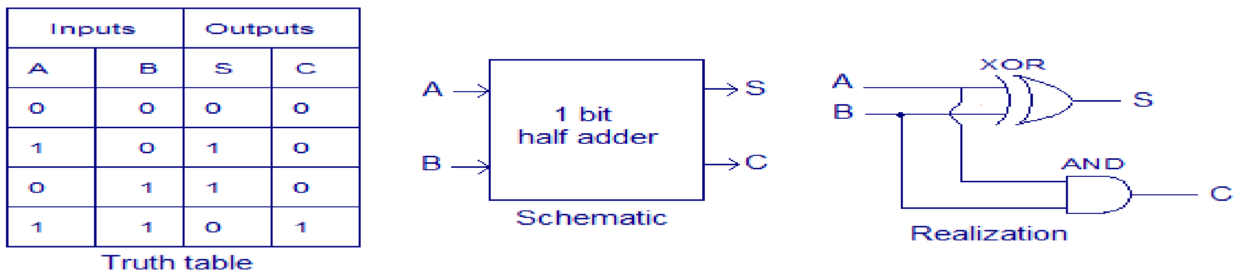
The Boolean expression for the Difference and Barrow are

Difference D = $A \oplus B \oplus B_{in}$

Barrow $B_o = \overline{A}B + \overline{A}\,\overline{A}B + \overline{A}B_{in} + B_{in}B$

2.4 Circuit Schematic and Truth Table

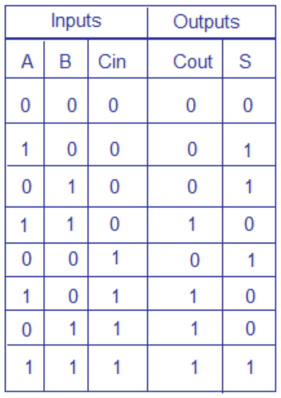
(i) **HALF ADDER**



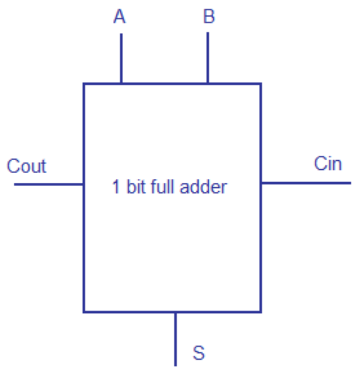
Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Figure 2.1: Half-Adder – Truth table and Schematic

(ii) FULL ADDER



Inputs			Outputs	
A	B	Cin	Cout	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1



1 bit full adder truth table & schematic

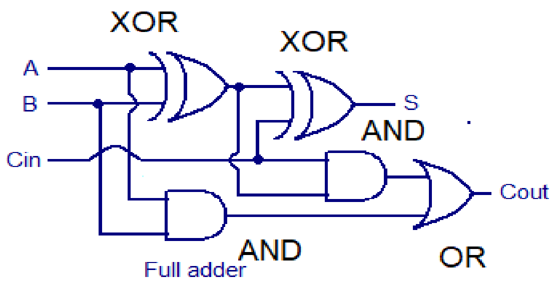


Figure 2.2: Full-Adder – Truth table and Schematic

Figure 5. Full Adder Circuit

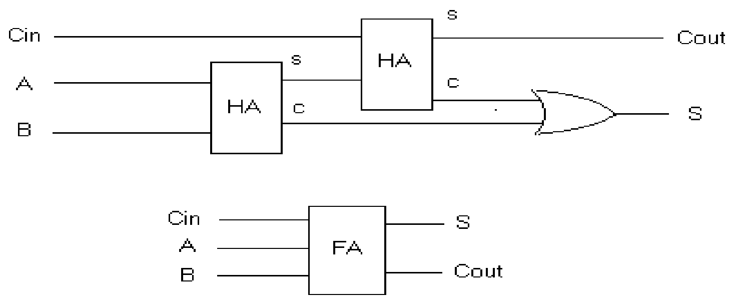
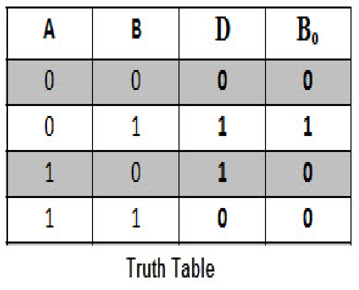


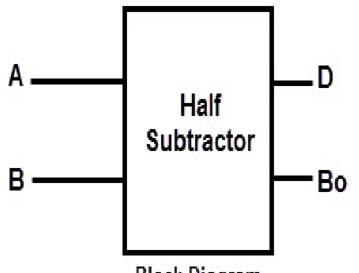
Figure 2.3: Full-Adder using two Half-Adders

(iii) **HALF SUBTRACTOR**



A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Truth Table



Block Diagram

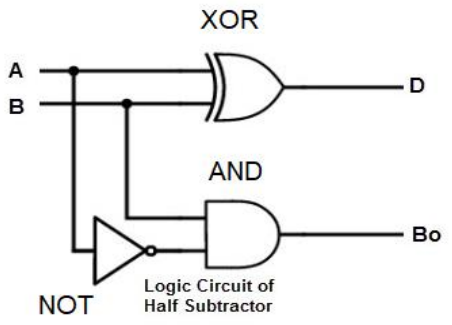
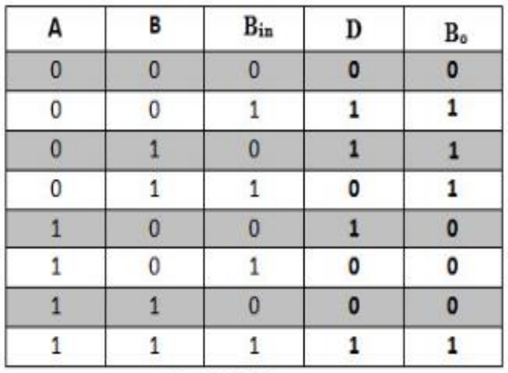


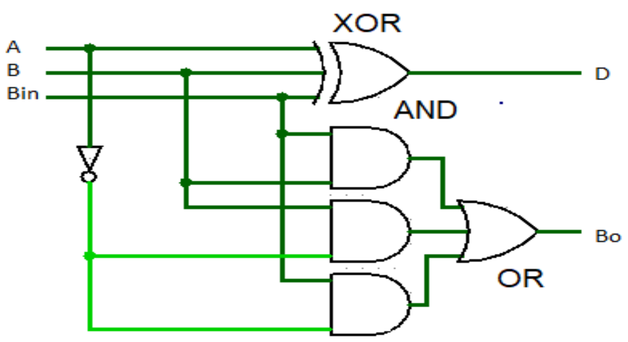
Figure 2.4: Half Subtractor Truth table and Circuit

(iv) **FULL SUBTRACTOR**



A	B	B_{in}	D	B_o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Truth Table



Logic Circuit

Figure 2.5: Full Subtractor Truth table and Circuit

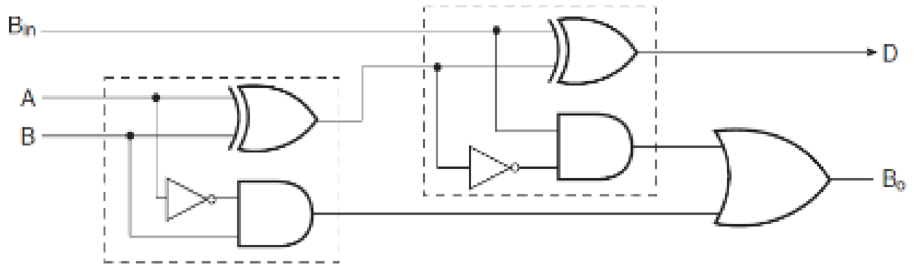


Figure2.6: Full Subtractor using Two Half- Subtractor

(v) **Lab Practice Circuits:**

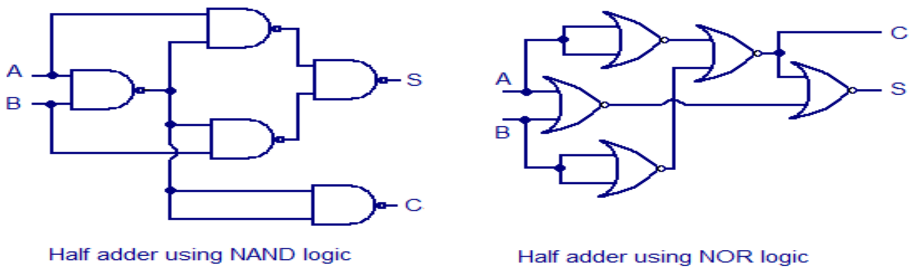


Figure 2.7: Half-Adder using NAND & NOR logic

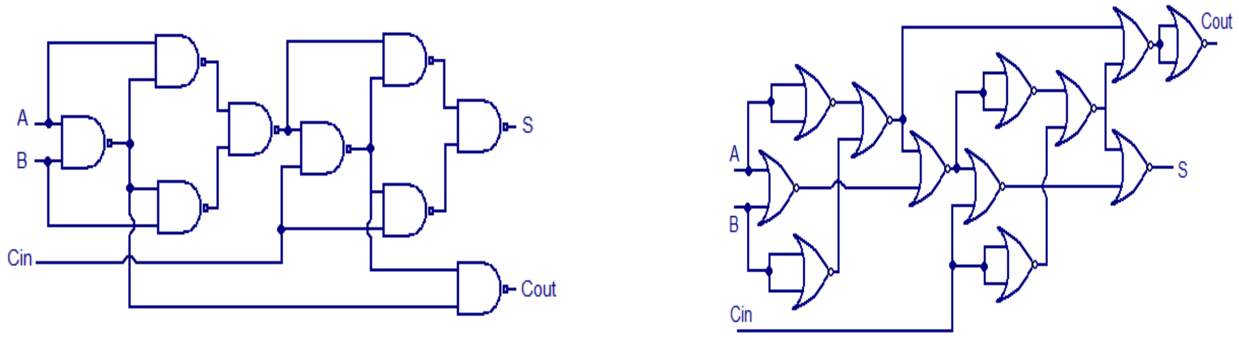


Figure 2.8: Full-Adder using NAND &NOR logic

2.5 Pre lab Questions

1. What is meant by a combinational circuit?
2. Discuss on the logic schematic and the uses of Half adder?
3. Design a circuit which can add 3 bits.
4. Realize a half adder circuit using a NAND gate.
5. Realize a Half subtractor circuit using NOR gate.
6. State the Significances and limitations of Karnaugh map.

2.6 Lab Procedure

Software

1. Double click logisim-win-2.7.1.exe on your desktop to open Logisim.
2. Open a new project named Digital Lab.
3. In the main circuit window, select the gates as per the logic schematic and place it.
4. Change the number of inputs for each gate.
5. Select input/output devices and make connections to make circuits.
6. Save your file as Adder and Subtractor under the project from the File menu.
7. Once the circuit get done, test it by changing the input values and verify the output values as
. per the truth table
8. Check the auto generated truth table from the Project menu and select Analyze Circuit and then select table.

Trainer Kit

1. Connect the circuit as shown in the figures.
2. Connect the power supply and ground to the respective pin numbers to all the IC's.
3. Give the inputs and verify the output in the truth table.

2.7 Post lab Questions

1. Two numbers (1101 and 1011) are applied to a 4-bit parallel adder. The input Carry is '1'. Determine the sum and output carry.
2. Obtain the output of Full adder using NAND gate using Logisim or LTSPICE.
3. Realize a Full subtractor using NOR gate using Logisim and verify its truth table.
4. Realize a 4-bit subtractor by using 4-bit adder and inverters.
5. Design a 4- bit parallel Adder/subtractor.
6. What are the advantages of complement arithmetic?

2.8 Result:

Lab 3: Design of Magnitude Comparator

3.1 Introduction

The purpose of this experiment is to introduce the design of 2-Bit & 8-bit Magnitude Comparator.

3.2 Software/Hardware Requirement

Equipment	:	Digital IC Trainer Kit
Software	:	Logisim
Discrete Components	-	74LS02 Quad 2-Input NOR gate 74LS04 Hex 1-Input NOT gate 74LS08 Quad 2-Input AND gate 74LS00 Quad 2-Input NAND gate 74LS266 Quad 2-Input XNOR gate 74LS85 4-bit magnitude comparator 74LS86 Quad 2-Input XOR 74LS10 Triple 3-Input NAND

3.3 Theory:

Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals at their input terminals and produces an output depending upon the condition of the inputs. For example, whether input A is greater than, smaller than or equal to input B etc.

Digital Comparators can compare a variable or unknown number for example A (A1, A2, A3, An, etc) against that of a constant or known value such as B (B1, B2, B3, ... Bn, etc) and produce an output depending upon the result. For example, a comparator of 2-bit, (A and B) would produce the following three output conditions. **A > B, A = B, A < B** This is useful if we want to compare two values and produce an output when the condition is achieved. For example, produce an output from a counter when a certain count number is reached. Consider the simple 2-bit comparator below.

2 – Bit Magnitude Comparator

Implementation

$A = A_1A_0$

$B = B_1B_0$

Here each subscript represents one of the digits in the numbers.

Equality

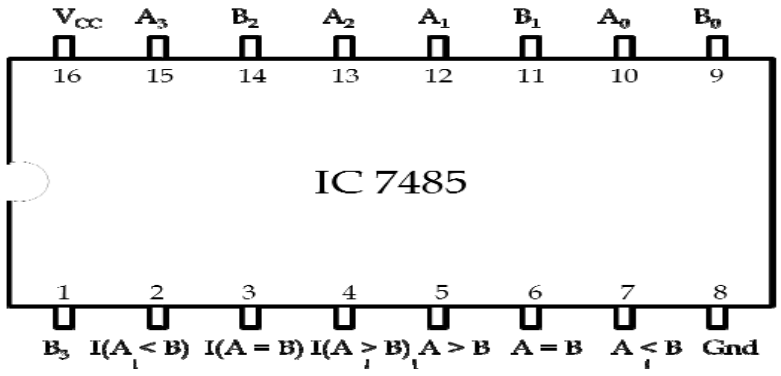
The binary numbers A and B will be equal if all the pairs of significant digits of both numbers are equal, i.e.,

$A_1 = B_1$ and $A_0 = B_0$

Since the numbers are binary, the digits are either 0 or 1 and the Boolean function for equality of any two digits Ai and Bi can be expressed as $x_i = Ai.B_i + \overline{A_i}.\overline{B_i}$

8- bit Magnitude Comparator using IC 7485:

PIN DIAGRAM FOR IC 7485:



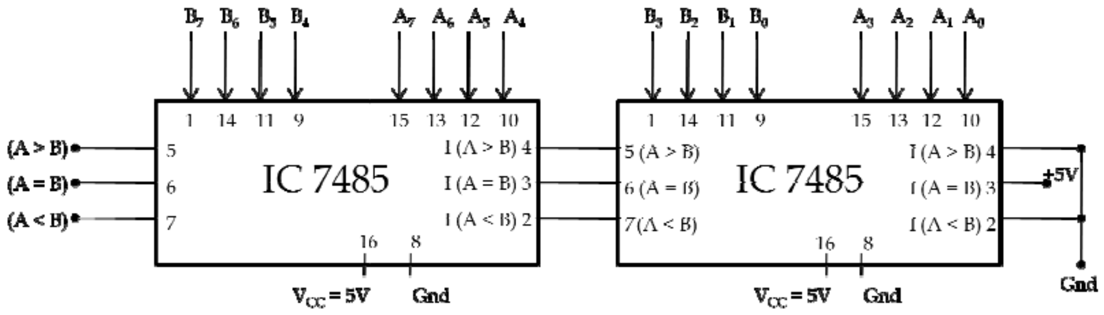


Figure. Connection diagram for 8- bit Magnitude Comparator using IC 7485

Function Table:

A_7, B_7	A_6, B_6	A_5, B_5	A_4, B_4	A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	A > B	A = B	A < B
$A_7> B_7$	x	x	x	x	x	x	x	1	0	0
$A_7< B_7$	x	x	x	x	x	x	x	0	0	1
$A_7= B_7$	$A_6> B_6$	x	x	x	x	x	x	1	0	0
$A_7= B_7$	$A_6< B_6$	x	x	x	x	x	x	0	0	1
$A_7= B_7$	$A_6= B_6$	$A_5> B_5$	x	x	x	x	x	1	0	0
$A_7= B_7$	$A_6= B_6$	$A_5< B_5$	x	x	x	x	x	0	0	1
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4> B_4$	x	x	x	x	1	0	0
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4< B_4$	x	x	x	x	0	0	1
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3> B_3$	x	x	x	1	0	0
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3< B_3$	x	x	x	0	0	1
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3= B_3$	$A_2> B_2$	x	x	1	0	0
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3= B_3$	$A_2< B_2$	x	x	0	0	1
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3= B_3$	$A_2= B_2$	$A_1> B_1$	x	1	0	0
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3= B_3$	$A_2= B_2$	$A_1< B_1$	x	0	0	1
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3= B_3$	$A_2= B_2$	$A_1= B_1$	$A_0> B_0$	1	0	0
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3= B_3$	$A_2= B_2$	$A_1= B_1$	$A_0< B_0$	0	0	1
$A_7= B_7$	$A_6= B_6$	$A_5= B_5$	$A_4= B_4$	$A_3= B_3$	$A_2= B_2$	$A_1= B_1$	$A_0= B_0$	0	1	0

x_i is 1 only if A_i and B_i are equal.

For the equality of A and B, all xi variables (for i=0,1) must be 1. So the equality condition of A and B can be implemented using the AND operation as $(A = B) = x_1 x_o$

The binary variable (A=B) is 1 only if all pairs of digits of the two numbers are equal.

Inequality
In order to manually determine the greater of two binary numbers, we inspect the relative magnitudes of pairs of significant digits, starting from the most significant bit, gradually proceeding towards lower significant bits until an inequality is found. When an inequality is found, if the corresponding bit of A is 1 and that of B is 0 then we conclude that A>B. (A>B) and (A < B) are output binary variables, which are equal to 1 when A>B or A<B respectively.

8 – Bit Magnitude Comparator

The comparison of two numbers is an operation that determines whether one number is greater than, less than or equal to the other number. A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitudes. The outcome of the comparison is specified by three binary variables that indicate whether (A > B), (A = B) or (A < B). The circuit for comparing two n- bit numbers has 22n entries in the truth table and this is difficult even with n = 3 as it requires 64 entries in the table.

Here we give the truth table for 2- bit magnitude comparator and thereby obtain the logic diagram by finding the expression for the output variables. Due to this disadvantage of making the truth table more complex for even 3- bit number we derive an algorithm to design a magnitude comparator. For any n- bit number, the algorithm is given as follows.

First, the most significant bit of both the numbers A & B is compared. In that bit position it is checked whether (A > B), (A = B) or (A < B). If (A > B) or (A < B) that is the final output. But if (A = B), then the next significant bit is compared. Likewise the procedure goes until all the bits are compared. IC 7485 is a 4- bit Magnitude Comparator whose pin diagram is as shown. On cascading two ICs it can be used to compare 8- bit numbers which is as shown in the connection diagram.

3.4 Circuit Schematic and Truth Table

TWO BIT MAGNITUDE COMPARATOR:

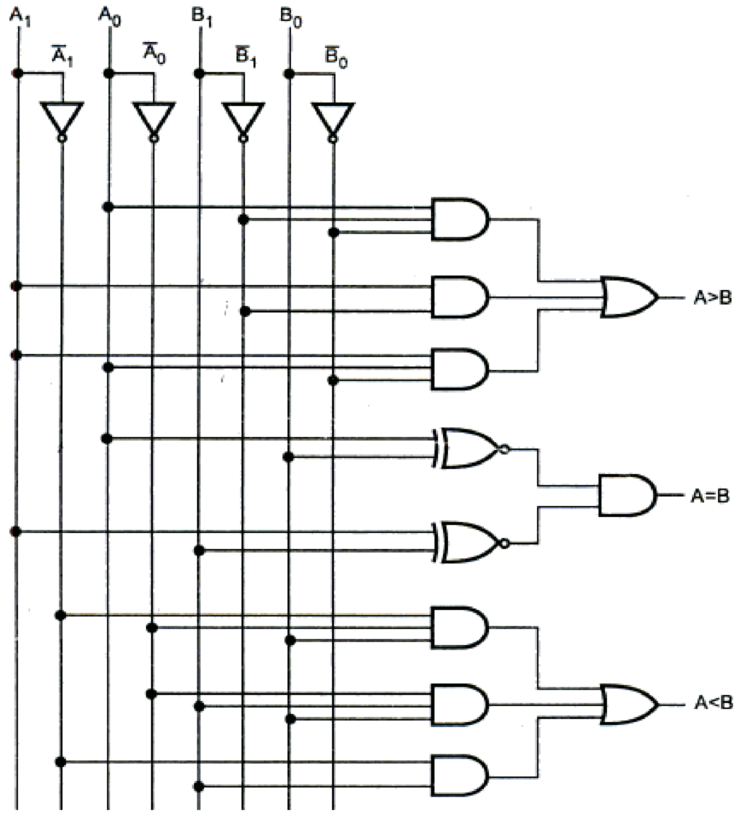


Figure 3.1: 2-bit Magnitude Comparator

Truth Table of 2-bit Magnitude Comparator

INPUTS				OUTPUTS		
A1	**A0**	**B1**	**B0**	**A < B**	**A = B**	**A > B**
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

3.5 Pre – Lab questions

1. Define magnitude comparator?
2. Write the output expressions for A<B and A>B.
3. Find out and label the output terminals (A<B, A=B and A>B) for the given 2 bit comparator circuit. (experimental circuit diagram)
4. List out the applications of comparators?
5. Which Logic gate is used to find A=B?

3.6_Lab Procedure

Trainer Kit

1. Construct the logic circuit of the 2-bit magnitude comparator shown in Figure 3.1.
2. Use different sets of inputs for A and B to check each of the outputs A<B, A=B and A>B.
3. Construct the logic circuit of the 8-bit magnitude comparator shown in Figure .
4. Use different sets of inputs for A and B to check each of the outputs A<B, A=B and A>B

Software

1. Double click logisim-win-2.7.1.exe on your desktop to open Logisim.
2. Open a new project named Digital Lab.
3. In the main circuit window, select the gates as per the logic schematic and place it.
4. Change the number of inputs for each gate.
5. Select input/output devices and make connections to make circuits.
6. Save your file as comparator under the project from the File menu.
7. Once the circuit get done, test it by changing the input values and verify the output values as per the truth table.
8. Check the auto generated truth table from the Project menu and select Analyze Circuit and then select table.

3.7 Post Lab question

1. How many IC 7485 required to design 24 bit comparator?
2. Give a summary of the points that you have learned from this experiment.
3. Design a 4-bit comparator using two 2-bit comparators.
4. Design a 8-bit comparator using two 4-bit comparators.

3.8 Result: