



GOVT CO-ED POLYTECHNIC

BYRON BAZAR RAIPUR (C.G.)

LAB MANUAL

Branch : Electrical Engineering

Year & Semester : 2nd Year / 4th Semester

2000461(025) – Digital Electronics (Lab)

CONTENTS

S. No.	Title of Experiment	Page No.
1.	To verify the truth tables of different logic gates (AND, OR, NOT, NAND, NOR)	1
2.	To verify De-Morgan's theorem using the kits available in the lab.	3
3.	To design basic gates using NAND gates.	5
4.	Perform various arithmetic operations using different number systems.	6
5.	To design and implementation of full adder and full subtractors.	8
6.	To design an 8 X 1 Multiplexer using IC 74151.	10
7.	To design a synchronous UP-counter using IC 7490.	11
8.	To design a shift register using IC 7495.	12
9.	To identify different types of flip flops and corresponding ICs.	13
10.	To simulate a digital circuit using the virtual laboratory software.	15

Experiment No: 1

AIM: To verify the truth tables of different logic gates (AND, OR, NOT, NAND, NOR).

Apparatus Required: 5-in-1 logic gates module, connecting cables.

Theory:

There are different binary/Boolean operations which are basic functions of any digital circuit for generating and processing digital signals. Logic gates are symbolic representation of such functions. Five of such basic Boolean operation denoting gates are as follows:

- **AND** : - This gate gives output only when all the inputs are at logic level 1.
- **OR** : - This gate gives output only when either of the inputs is at logic level 1.
- **NOT** : - This gate gives the compliment of given binary input digit.
- **NAND** : - This gate gives output as complimented AND operation of its input.
- **NOR** : - This gate gives output as complimented OR operation of its input.

Logic Diagram & Truth Table:





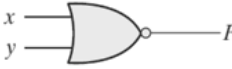
Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = x \cdot y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
NAND		$F = (xy)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

Fig. 1.1: Digital Logic Gates.

Procedure:

- Ensure that the *5-in-1 logic gates experiment kit* is connected to the power supply and the toggle switch is at *OFF* position.
- Choose any logic gate for verification of truth table, say AND gate.
- Connect the cables to two inputs of AND gate viz. A and B to the jack labelled *logic-1* & *logic-0* respectively for 1st set of observation.
- Connect one end of the cable to output of AND gate & other end to the jack labelled as *output indicator*.
- Switch the circuit *ON* by toggling the switch to *ON* position.
- Observe the output indicator LED, whether its glowing or not. Take note of it.
- Repeat above steps for more sets of observation for different logic gates.
- Record the observations in a tabular form in a practical file.

Result:

Thus, the truth table for different logic gates have been verified successfully.

Precautions:

- All the connection should be tight.
- It should be noted that the values of each component of the circuit does not exceed to their maximum ratings.
- Before the circuit connection, make sure everything is connected as per circuit diagram and power is switched OFF.
- Quickly inform teacher/lab attendant in case of short circuit.

Experiment No: 2

AIM: To verify De-Morgan's theorem using the kits available in the lab.

Apparatus Required: De-Morgan's theorem kit/module, connecting cables.

Theory:

The theorem is named after Augustus DeMorgan (1806–1871) who introduced a formal version of the laws to classical logic. DeMorgan's formulation was influenced by algebraization of logic undertaken by George Boole.

The theorem explains that the complement of the product of all the terms is equal to the sum of the complement of each term and vice-versa. The theorem can be understood and tested using logic shown below:

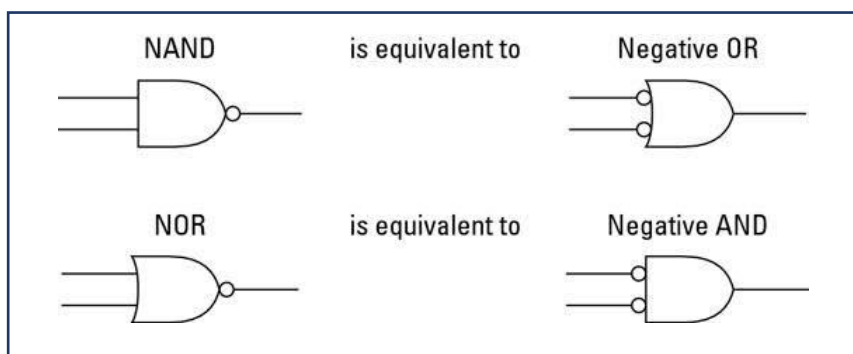


Fig. 2.1: Logic Gate's conversion based on De-Morgans Theorem.

Algebraically, *Theorem 1:* $\overline{A \cdot B} = \overline{A} + \overline{B}$

Theorem 2: $\overline{A + B} = \overline{A} \cdot \overline{B}$

Procedure:

- In order to test Theorem 1, make all the connections properly as per Fig. 2.2.
- To test L.H.S. of Theorem 1, follow *circuit (a)*. Take note of *output indicator*.
- Repeat above steps to test Theorem 2.
- Make sure that the kit is not powered up while making connections.
- Make the *Logic Inputs* initially toggled at *logic-0* level.
- Note the observations in a practical file.

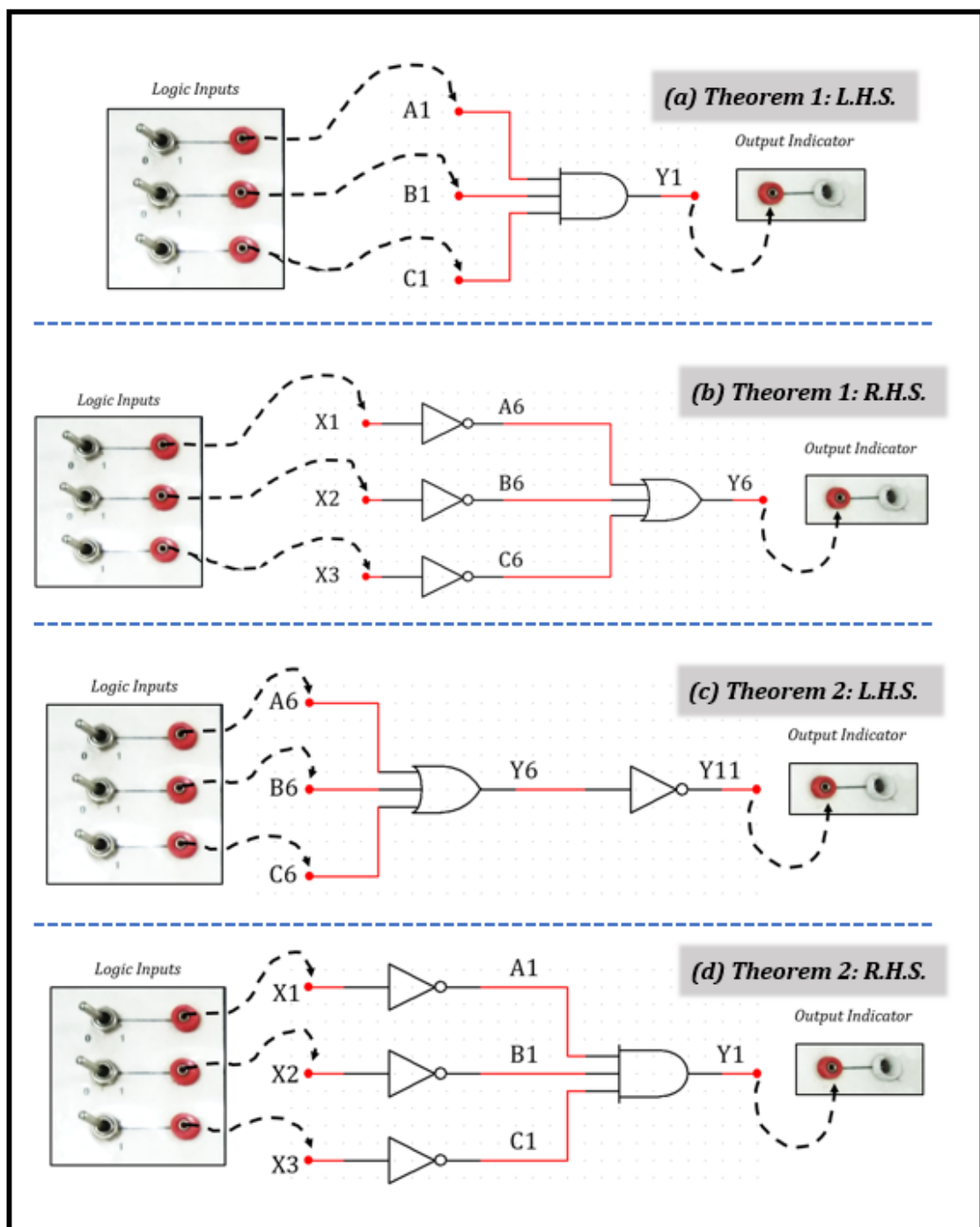


Fig. 2.2: Connection diagram for testing De-Morgans Theorem.

Result:

Thus, the DeMorgans theorem has been verified successfully.

Precautions:

- All the connection should be tight.
- It should be noted that the values of each component of the circuit does not exceed to their maximum ratings.
- Before the circuit connection, make sure everything is connected as per circuit diagram and power is switched OFF.
- Quickly inform teacher/lab attendant in case of short circuit.

Experiment No: 3

AIM: Design basic gates using NAND gates.

Apparatus Required: Desktop, internet connection etc.

Theory:

The NAND gate is a universal gate. It is so called because by using this gate we can make any gate like NOT, OR, AND, X-OR etc. With the help of this gate, we can also make multiplexers and de-multiplexers. Basic gates like AND, OR and NOT have been realised using combinations of NAND gate only, as shown in the table below:

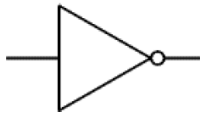
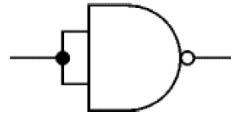
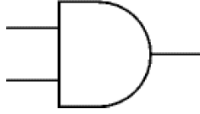
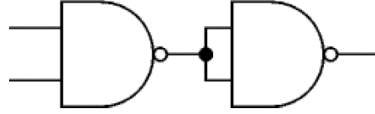
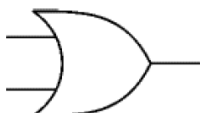
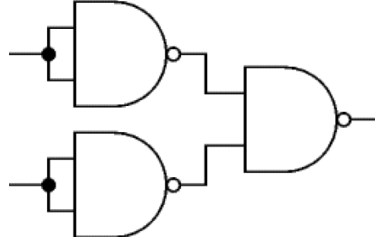
Gate		Equivalent in NAND gates
NOT		
AND		
OR		

Fig. 3.1: The NAND gate equivalents of NOT, AND, OR logic gates.

Procedure:

- Turn the computer *ON* and establish internet connection. Open browser & search *MultiSimlive.com* in the search bar.
- Login as a user. Move to working area online. Pick items from the components pane.
- Make proper circuit connections as per Fig. 3.1. Run the simulation. Note down observations as truth table in a practical file.

Result:

Thus, basic gates have been simulated successfully using NAND gate only.

Precautions:

- All the computer system should be logged out.
- Only visit allowed websites.
- Don't run unnecessary commands over the software. Ask teacher in case of system lag.

Experiment No: 4

AIM: To perform various arithmetic operations on scientific calculator using Base-N settings for different number systems.

Theory:

The number system relates to the quantity and symbols of data representation used by any digital circuit / microprocessor. Such number system is used by the ALU of processor in order to perform various arithmetic operations like addition, subtraction, division & multiplication.

Procedure:

- Turn the scientific calculator *ON* and set its mode to BASE-N setting by pressing *ON* \rightarrow *Mode Setup* \rightarrow *key no 4*.
- To enter in decimal mode, press key labelled as **DEC**.
- Carry out the arithmetic operations given below.

1. Decimal Number System –

Range of digits valid in this number system: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

(a) Addition:

$$(23)_{10} + (41)_{10} = (64)_{10}$$

(b) Subtraction:

$$(78)_{10} - (24)_{10} = (54)_{10}$$

$$(35)_{10} - (43)_{10} = - (8)_{10}$$

(c) Multiplication:

$$(39)_{10} \times (3)_{10} = (117)_{10}$$

(d) Division:

$$(124)_{10} / (4)_{10} = (31)_{10}$$

$$(125)_{10} / (4)_{10} = (31)_{10}$$

} As per calculator

- To enter in binary mode, press key labelled as **BIN**.
- Carry out the arithmetic operations given below.

2. Binary Number System –

Range of digits valid in this number system: 0, 1.

(a) Addition:

$$(1001)_2 + (111101)_2 = (1000110)_2$$

(b) Subtraction:

$$(1001)_2 - (0110)_2 = (0011)_2$$

$$(1001)_2 - (1100)_2 = (111100)_2$$

} As per calculator

(c) Multiplication:

$$(101)_2 \times (110)_2 = (11110)_2$$

(d) Division:

$$(11011011)_2 / (011)_2 = (1001001)_2$$

- To enter in octal mode, press key labelled as **OCT**.
- Carry out the arithmetic operations given below.

3. Octal Number System –

Range of digits valid in this number system: 0, 1, 2, 3, 4, 5, 6, 7.

(a) Addition:

$$(14)_8 + (37)_8 = (53)_8$$

$$(14)_8 + (39)_8 = \text{Syntax Error due to invalid number used....}$$

(b) Subtraction:

$$(56)_8 - (27)_8 = (27)_8$$

$$(25)_8 - (67)_8 = -(377777736)_8$$

} As per calculator

(c) Multiplication:

$$(53)_8 \times (27)_8 = (1735)_8$$

(d) Division:

$$(124)_8 / (4)_8 = (25)_8$$

- To enter in hexa-decimal mode, press key labelled as **HEX**.
- Carry out the arithmetic operations given below.

4. Hexa-Decimal Number System –

Range of digits valid in this number system: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

(a) Addition:

$$(2D)_{16} + (3F)_{16} = (6C)_{16}$$

(b) Subtraction:

$$(33A)_{16} - (149)_{16} = (1F1)_{16}$$

$$(152)_{16} - (321)_{16} = (FFFFF2C)_{16}$$

} As per calculator

(c) Multiplication:

$$(78C)_{16} \times (24)_{16} = (10FB0)_{16}$$

(d) Division:

$$(78C)_{16} / (6)_{16} = (142)_{16}$$

Result:

Thus, the basic arithmetic operations have been performed successfully, for all the number systems.

Experiment No: 5

AIM: Design and implementation of full adder and full subtractors.

Apparatus Required: Breadboard, power supply, X-OR, AND and OR gates, connecting wires etc.

Theory:

The digital circuits are used for arithmetic operations like addition and subtraction of binary digits, on a bit-by-bit manner inside ALU of microprocessors. Such addition or subtraction requires dedicated logic gates and can be realised using X-OR, AND & OR gates.

FULL ADDER: A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A , B and C_{in} ; A and B are the operands, and C_{in} is a bit carried in (from a past addition). The circuit produces two output bits which are sum, typically represented by the bit S and carry, which is represented by C_{out} . (See Fig. 5.1)

FULL SUBTRACTOR: It is a combinational circuit which is used to perform subtraction of three bits. It has three input bits, A (minuend) and B (subtrahend) and C (subtrahend) & two outputs bits, D (difference) and b (borrow). (See Fig. 5.2)

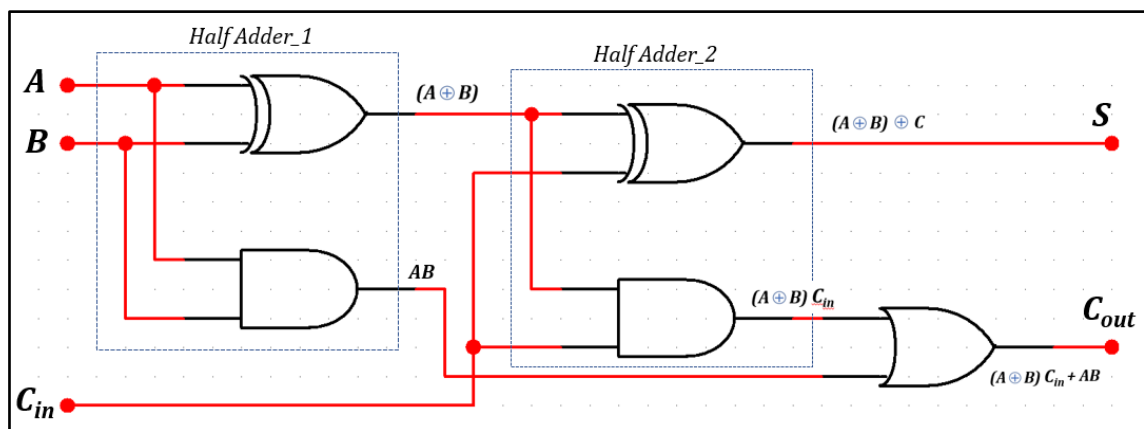


Fig. 5.1: Logic gate connection diagram for full adder.

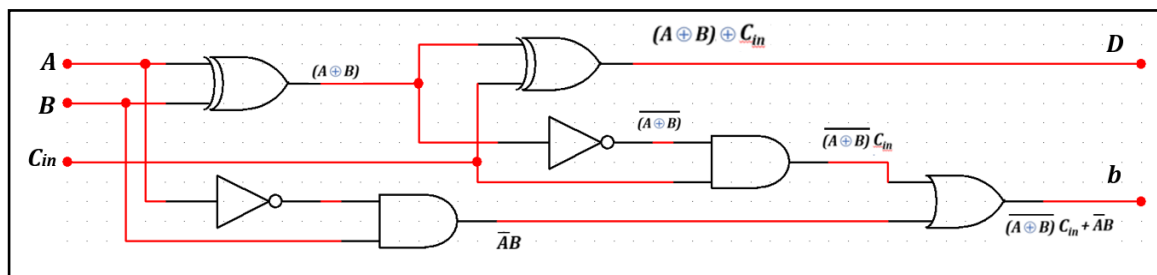


Fig. 5.2: Logic gate connection diagram for full subtractor.

Full Adder Truth Table

Input			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Subtractor Truth Table

Input			Outputs	
A	B	C_{in}	D	b
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

Procedure:

- Turn the power supply ON.
- Make connections on bread board as per Fig. 5.1 to realise full adder.
- Observe its output indicator bit/LED and note the observation in terms of truth table.
- Repeat above steps to realise full subtractor but make connections as per Fig. 5.2.
- Compare results with truth table.
- Note the observation in terms of truth table, in a practical file.

Result:

Thus, full adder & subtractor circuits have been designed and tested successfully.

Precautions:

- All the connection should be tight.
- It should be noted that the values of each component of the circuit does not exceed to their maximum ratings.
- Before the circuit connection, make sure everything is connected as per circuit diagram and power is switched OFF.
- Quickly inform teacher/lab attendant in case of short circuit.

Experiment No: 6

AIM: To study an 8 X 1 Multiplexer using IC 74151 based kit.

Apparatus Required: 8 X 1 MUX based kit/module.

Theory:

A multiplexer (MUX) or data selector is a logic circuit that accepts several data inputs and allows only one of them at a time to get through the output. The routing of that desired data input is controlled by select input, also called as address line /Input. Figure below shows a functional diagram of MUX & 8 X 1 MUX truth table:

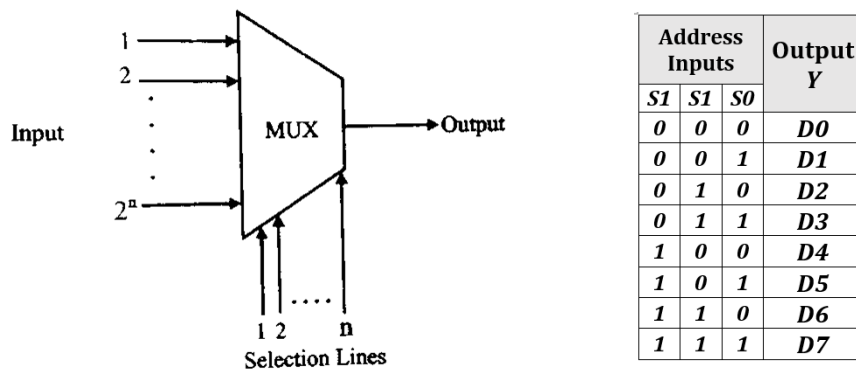


Fig. 6.1: MUX Functional diagram (Many-To-One data selector).

Procedure:

- Turn the power supply *ON*. Toggle the *Power* switch of the kit to *ON*.
- Toggle LE switch **disable** mode first. Set the toggle switches of address to any particular combination say (S₀=0, S₁=1, S₂=0). This will select data line D₂ to appear at output Y.
- Now Toggle LE switch to **enable** mode, this will enable data line D₂. Observe that, output Y changes as the D₂ state changes from *logic-0* to *logic-1* and vice versa.
- Repeat above steps for different set of address.
- Note the observation in terms of truth table, in a practical file.

Result:

Thus, an 8 X 1 MUX using IC 74151 based kit has been studied successfully.

Precautions:

- All the connection should be tight.
- It should be noted that the values of each component of the circuit does not exceed to their maximum ratings.
- Before the circuit connection, make sure everything is connected as per circuit diagram and power is switched OFF.
- Quickly inform teacher/lab attendant in case of short circuit.

Experiment No: 7

AIM: To design a synchronous UP-counter using IC 7490.

Apparatus Required: Breadboard, IC 7490, connecting jumpers etc.

Theory:

A digital counter is a set of flip-flops interconnected such that their combined states at any time is the binary equivalent of the no of pulses applied up to that time, i.e. it is a logic circuit that count number of pulses.

A 3-bit synchronous counter using JK-flip flop, requires 3 JK-flip flops (with output being Q_A , Q_B , Q_C , see Fig. 7.1).

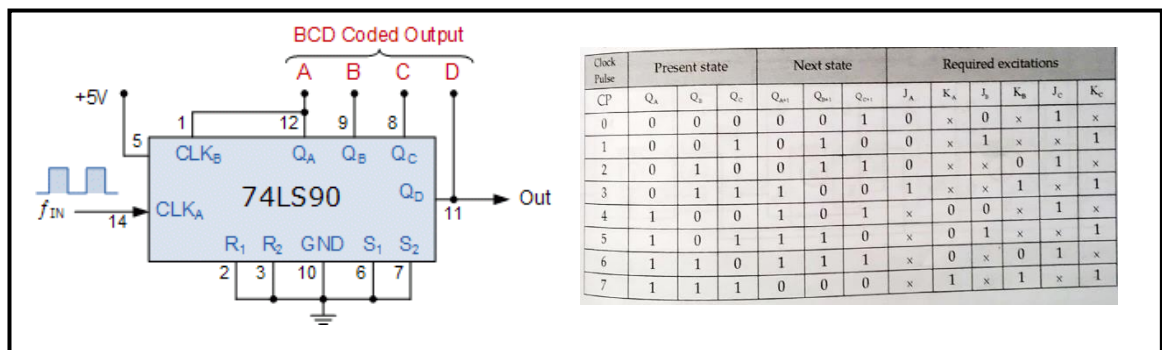


Fig. 7.1: Connection diagram & truth table of synchronous counter using 7490 IC.

Procedure:

- Turn the power supply *ON*. Make connections on bread board as per circuit diagram.
- Apply clock pulses to the IC 7490. Note the BCD number obtained on LED indicator.
- Repeat above steps for different pulse width of clock I/P.
- Note the observation in terms of truth table, in a practical file.

Result:

Thus, a synchronous UP-counter using IC 7490 has been designed successfully.

Precautions:

- All the connection should be tight.
- It should be noted that the values of each component of the circuit does not exceed to their maximum ratings.
- Before the circuit connection, make sure everything is connected as per circuit diagram and power is switched OFF.
- Quickly inform teacher/lab attendant in case of short circuit.

Experiment No: 8

AIM: To design a shift register using IC 7495.

Apparatus Required: Breadboard, IC 7495, connecting jumpers etc.

Theory:

A register is a digital circuit which stores binary information and is also known as memory register. Such a register, capable of shifting binary information (single bit) either to the right or the left is called a shift register.

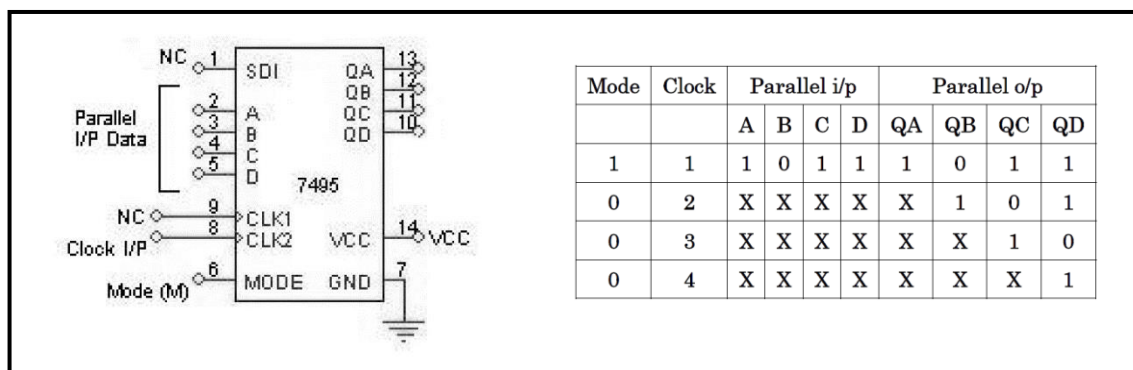


Fig. 8.1: Connection diagram & truth table of shift register using IC 7495.

Procedure:

- Turn the power supply *ON*. Make connections on bread board as per circuit diagram.
- Apply clock pulses to the IC 7495. Note the change in output bit one by one from QA → QB → QC → QD obtained on LED indicator.
- Repeat above steps for different pulse width of clock I/P.
- Note the observation in terms of truth table, in a practical file.

Result:

Thus, a shift register using IC 7495 has been designed successfully.

Precautions:

- All the connection should be tight.
- It should be noted that the values of each component of the circuit does not exceed to their maximum ratings.
- Before the circuit connection, make sure everything is connected as per circuit diagram and power is switched OFF.
- Quickly inform teacher/lab attendant in case of short circuit.

Experiment No: 9

AIM: To identify different flip-flops and their characteristic table using a lab kit.

Apparatus Required: Lab kit with flip flops, connecting cables etc.

Theory:

A flip flop is a sequential digital circuit in which the output at any instant depends on the present state of input as well as past output. Such circuits are used to make memory elements as they use a feedback path for storage of one bit binary information.

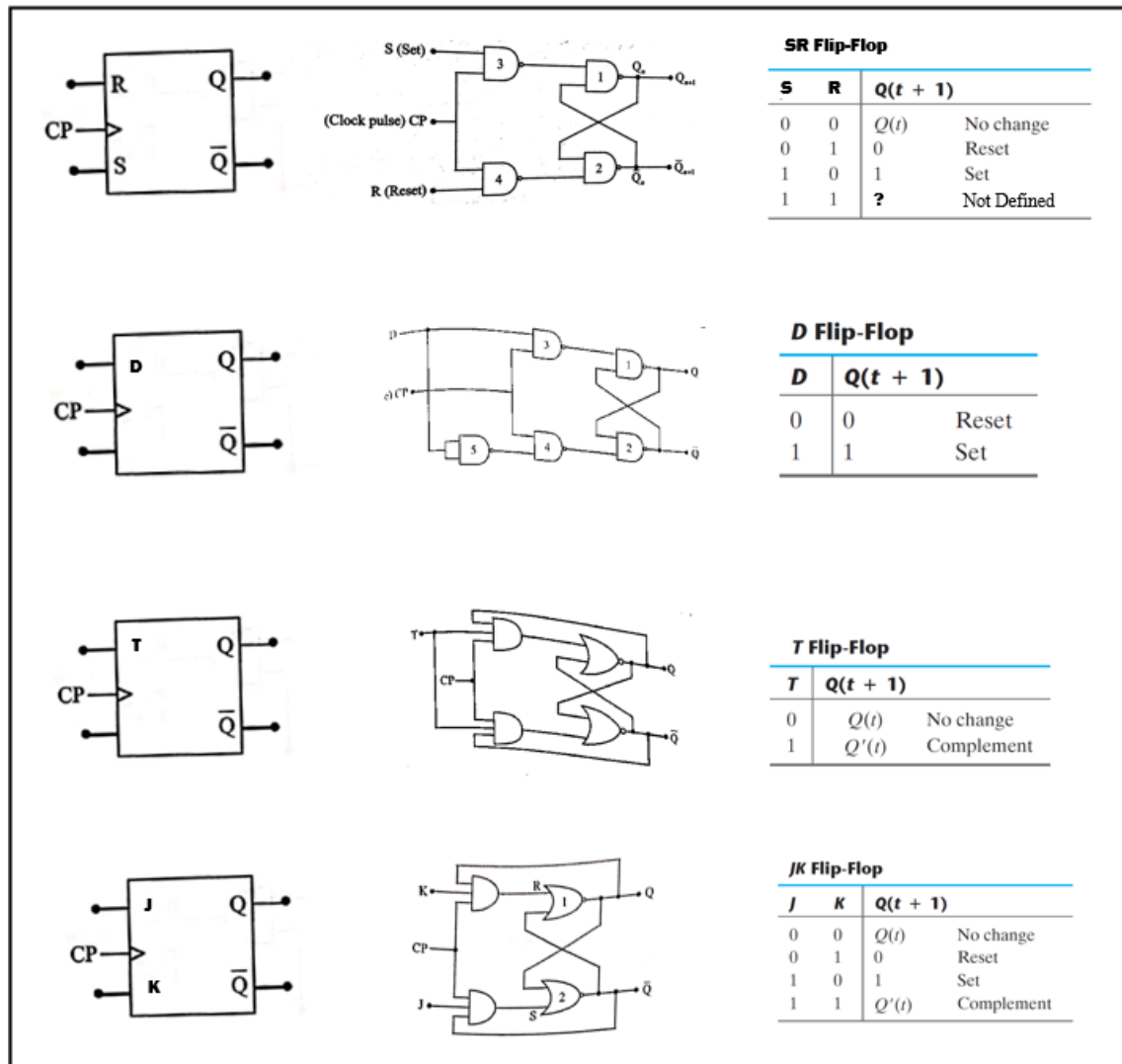


Fig. 9.1: Different Flip-Flops & their characteristic table.

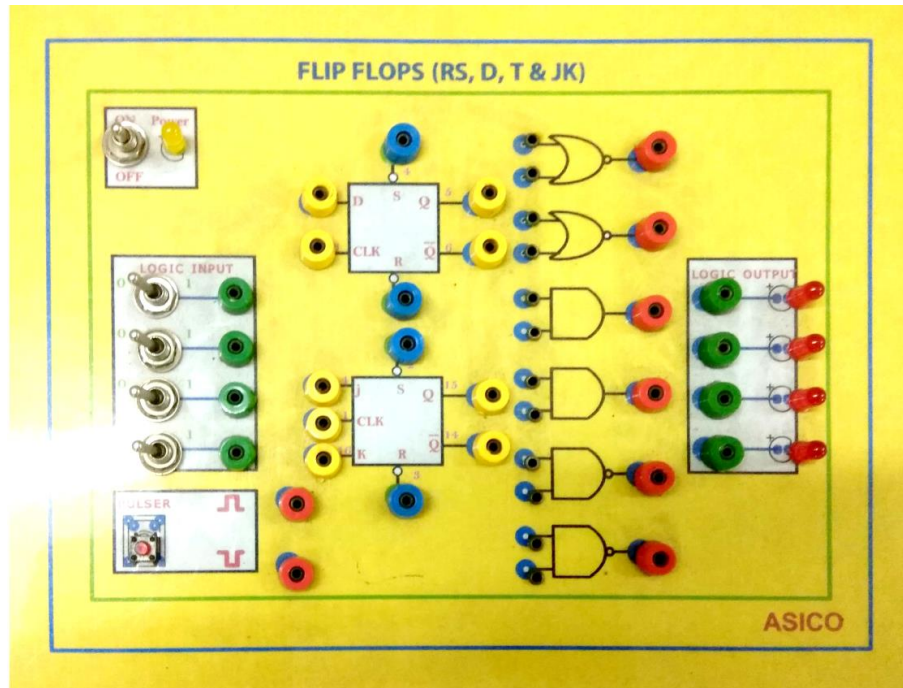


Fig. 9.2: Different Flip-Flops on our lab kit.

Procedure:

- Turn the power supply *ON*. Make connections on lab kit using cables as per Figure.
- Apply clock pulses to the flip-flop, say D- flip flop. Note the change in logic output red LED, one by one as the clock pulse varies.
- Repeat above steps for different flip-flops.
- Verify the characteristic table for each flip flop.
- Note the observation in a tabular form in a practical file.

Result:

Thus, flip-flops and their characteristic tables have been verified successfully.

Precautions:

- All the connection should be tight.
- It should be noted that the values of each component of the circuit does not exceed to their maximum ratings.
- Before the circuit connection, make sure everything is connected as per circuit diagram and power is switched OFF.
- Quickly inform teacher/lab attendant in case of short circuit.

Experiment No: 10

AIM: To simulate a digital circuit using the virtual laboratory software.

Apparatus Required: Computer system with internet, LTSpice software.

Theory:

The need of virtual laboratory is often needed in circuit simulations. The use of electronic/electrical circuit modelling software reduces much effort of engineers in the design and implementation phase of any project. In this experiment, LTSpice based software is used to implement simple logic gates with ICs and verification of such gates have been performed afterwards.

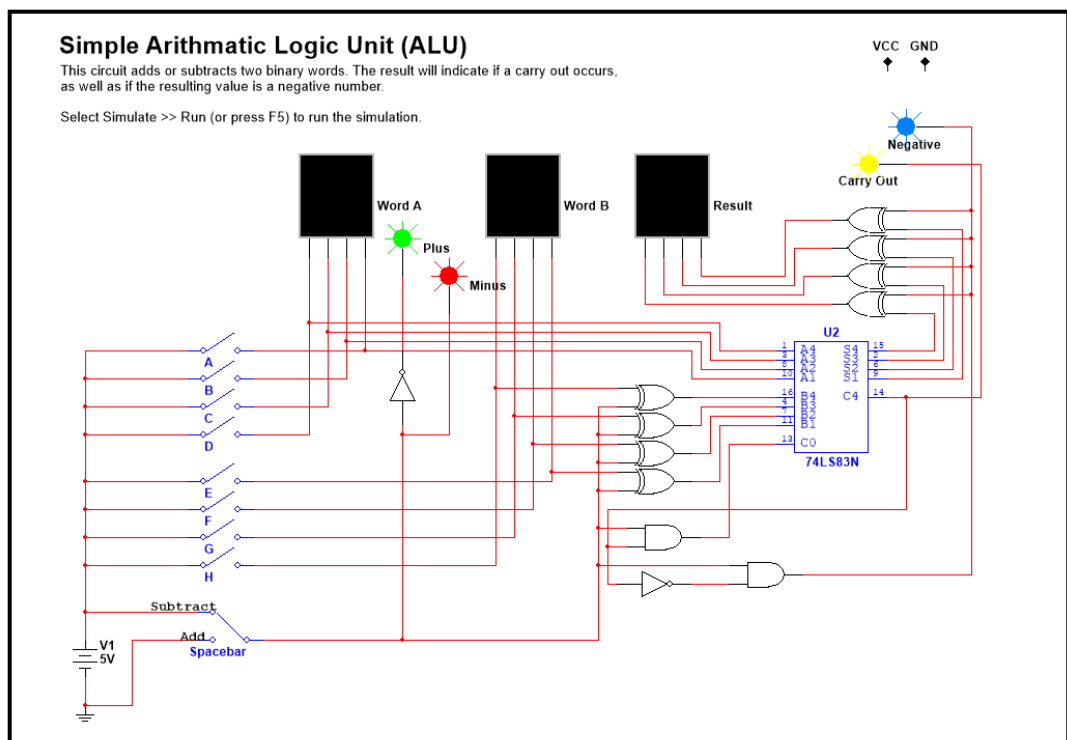


Fig. 10.1: Simulation of digital circuit on virtual lab software.

Procedure:

- Turn the computer *ON* and establish internet connection. Open browser & search *LTSpice* software in the search bar. Login as a user. Move to working area online. Pick items from the components pane.
- Make proper circuit connections as per Fig. 10.1. Run the simulation. Note down observations in a practical file.

Result:

Thus, the digital circuit simulation on lab software has been simulated successfully.

Dos & Don'ts in Digital Electronics Lab

Dos	Don'ts
<ul style="list-style-type: none">• Enter/exit lab quietly.• Raise your hand before asking any doubt.• Always have a clean & dry hand.• Touch keyboard & mouse gently.• Keep your work space clean.• Search only approved websites• Identify leads of IC before connection.• Find out the current and voltage rating of IC before experiment.	<ul style="list-style-type: none">• No food or drinks in the lab.• Do not mark on any part of kit.• Do not change any key settings of computer.• No magnets allowed in computer lab• Do not pull any cable/cord of any system.• Ask teacher before taking any printout.• Do not pull any cable/cord of jack.