COMSATS University Islamabad (CUI), Lahore Campus
Department of Electrical & Computer Engineering

CPE/EEE241 – Digital Logic Design

Lab Manual for Spring 2022

Lab Resource Persons

Sarmad Hassan

Theory Resource Persons

Amna Arif

Supervised By

Dr. Abbas Javed

Name: _____ Registration Number: *CIIT/ - - /LHR*

Program: _____ Batch: _____

Semester _____

Revision History

Sr. #	Update	Date	Performed by
1	Lab Manual Updating	Sep-12	Engr. M. Bilawal Rehman
2	Lab Manual Review	Sep-12	Dr. Naveed Bin Rais, Dr. Umer Farooq
3	Layout modifications	Aug-14	Engr. Asif Ali
4	Lab Manual Review	Aug-14	Dr. Naeem Shahzad, Dr. Naeem Awais
5	Layout modifications	Jan-15	Engr. Muzamil Ahmad
6	Lab Manual Review	Jan-15	Dr. Naeem Shehzad
7	Layout modifications	Sep-15	Engr. Asif Ali Engr. Muzamil Ahmad
8	Lab Manual Review	Sep-15	Dr. Kashif Imran
9	Lab Manual Update	Sep-17	Engr. Arslan Khalid, Engr. Nesruminallah
10	Lab Manual Update & Layout modifications	Feb-20	Sarmad Hassan & Arfa Tariq
11	Lab Manual Review	Feb-20	Dr. Abbas Javed

Preface

Digital Logic Design is among the core courses of Electrical Engineering presenting basic tools for the design of digital circuits. It serves as a building block in many disciplines that utilize and manipulate digital information such as microprocessor-based systems, computer organizations and architecture, digital systems design, digital control and digital communications etc.

This document has been prepared to serve as a laboratory manual for EEE 241 Digital Logic Design course for electrical engineering students at CIIT Lahore. The manual consists of a set of experiments designed to allow students to build, and verify digital circuits and systems. This set of experiments cover relevant topics prescribed in the syllabus and are designed to reinforce the theoretical concepts taught in the classroom with practical experience in the lab. By the end of the course, students are expected to have a good understanding of digital logic design and its implementation using ICs for basic logic gates and simple circuits.

Acknowledgement

Contribution of the following faculty members is highly appreciated:

- ✓ Engr. Syed Junaid Akhtar
- ✓ Engr. Wajeeha Khan
- ✓ Engr. Sara Sajid
- ✓ Engr. Abdul Moeed
- ✓ Engr. Hasnain Kashif
- ✓ Engr. Arslan Khalid
- ✓ Engr. Nesruminallah
- ✓ Engr. Arfa Tariq
- ✓ Engr. Sarmad Hassan

Books

Text Books

1. Digital Design, 5th Edition by Morris Mano & Michael D. Ciletti

Reference Books

1. Introduction to Logic Design, 2nd Edition by Alan B. Marcovitz
2. Thomas L. Floyd, Digital Fundamentals (7th Edition)

Learning Outcomes

- To get hands on experience of implementation of logic circuits using logic ICs, trainer and breadboards.
- To get familiar with the process of digital logic design (Combinational & Sequential logic)
- To be able to design digital logic circuit in application software.
- To be able to analyze the experimental results and to compare them with the theoretical knowledge.
- Ability to implement the Boolean functions using basic as well as universal gates.
- Ability to translate real-world problem into logic equations and logic diagrams.

Theory CLOs:

1. CLO1: Comprehend the working with different number systems, Boolean algebra and mapping methods using standard mathematical rules. (PLO1-C2)
2. CLO2: Analyze the working of combinational and sequential logic circuits using digital logic principles and Boolean algebra. (PLO2-C4)
3. CLO3: Plan and design the combinational and sequential logic circuits using digital logic principles, Boolean algebra and mapping methods. (PLO3-C5)

Lab CLOs:

4. CLO4: To analyze and design the combinational and sequential logic circuits using software and hardware platforms. (PLO3-C5)
5. CLO5: Follow the software and hardware tools to reproduce the response of the digital logic circuits using software and hardware platforms. (PLO5-P3)
6. CLO6: To explain and write effective lab reports of experiments performed during lab (PLO10-A3)
7. CLO7: To demonstrate the working of a digital logic circuit designed individually and by the teamwork using software and hardware platforms. (PLO9-A3)

CLOs – PLOs Mapping

PLO / CLOs	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12	Cognitive Domain						Affective Domain					Psychomotor Domain						
													C1	C2	C3	C4	C5	C6	A1	A2	A3	A4	A5	P1	P2	P3	P4	P5	P6	P7
CLO1	X												X	X																
CLO2		X											X	X	X	X														
CLO3			X										X	X	X	X	X													
CLO4 (LAB)			X										X	X	X	X	X													
CLO5 (LAB)					X																			X	X	X				
CLO6(LAB)										X									X	X	X									
CLO7(LAB)									X				X	X	X															

Lab CLOs – Lab Experiment Mapping

CLO / Lab	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5	Lab 6	Lab 7	Lab 8	Lab 9	Lab 10	Lab 11	Lab 12	Lab 13	Lab 14
CLO5	P1	P2	P2	P3	P3	P3	P3	P3	P3	P1	P3	P3	P2	P3
CLO6	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3

Grading Policy

The final marks for lab would comprise of Lab Assessment (25%), Lab S1 (10%), Lab S2 (15 %) and Lab Terminal (50%).

S-I 0.5*(S-I Exam result) + 0.5* (average of lab evaluation of Lab 1-4)

S-II 0.5*(S-II Exam result) + 0.5*[(average of lab evaluation of Lab 5-8) * 1.5]

Terminal 0.1*[(OEP marks out of 25)*2] +0.4*(Terminal Exam result out of 50) +0.25*[(average of lab evaluation of Lab 9-12) *5] + 0.10*[(average of lab evaluation of Lab 5-8) *5] + 0.15*[(average of lab evaluation of Lab 1-4) *5]

Lab Assignment Marks:

- Lab Assignment 1 marks = Lab report marks from experiment 1-3.
- Lab Assignment 2 marks = Lab report marks from experiment 4-6.
- Lab Assignment 3 marks = Lab report marks from experiment 7-9.
- Lab Assignment 4 marks = Lab report marks from experiment 10-14.

The minimum pass marks for both lab and theory shall be 50%. Students obtaining less than 50% marks (in either theory or lab, or both) shall be deemed to have failed in the course. The final marks would be computed with 75% weight to theory and 25% to lab final marks.

List of Equipment

IC 7400 (NAND Gate), IC 7402 (NOR Gate), IC 7404 (NOT Gate), IC 7408 (AND Gate), IC 7432 (OR Gate), IC 7447 (BCD to 7seg), IC 7486 (X-OR Gate), IC 74138 (De-multiplexer), IC 74148 (Encoder), IC 7420 (4 Input NAND GATE), IC 7474 (D Flip Flop), IC 7476 (J-K Flip Flop), IC 74147 (10 to 4 Line Encoder), IC 74139 (2 Line to 4 Decoder), IC 74153 (Dual Multiplexer 1 of 4), IC 74157 (quadruple 1 of 2 Data selector), IC 7485 (4-Bit Comparator), IC 7495 (4-Bit Shift Register), 7483 (4-Bit Adder), Trainer Board

Software Resources

Proteus Professional 6.9

Lab Instructions

- This lab activity comprises of three parts: Pre-lab, Lab Tasks, Post Lab Tasks, Lab Report and Conclusion and Viva session.
- The students should perform and demonstrate each lab task separately for step-wise evaluation.
- Only those tasks that are completed during the allocated lab time will be credited to the students.
- Students are however encouraged to practice on their own in spare time for enhancing their skills.

Lab Report Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

- Lab Objectives
- Methodology
- Conclusion

Laboratory Regulations and Safety Rules

The following Regulations and Safety Rules must be observed in all concerned laboratory location.

1) It is the duty of all concerned who use any electrical laboratory to take all reasonable steps to safeguard the HEALTH and SAFETY of themselves and all other users and visitors.
2) Be sure that all equipment is properly working before using them for laboratory exercises. Any defective equipment must be reported immediately to the Lab. Instructors or Lab. Technical Staff.
3) Students are allowed to use only the equipment provided in the experiment manual.
4) Power supply terminals connected to any circuit are only energized with the presence of the Instructor Lab. Staff.
5) Avoid any part of your body to be connected to the energized circuit and ground.
6) Switch off the equipment and disconnect the power supplies from the circuit before leaving the laboratory.
7) Observe cleanliness and proper laboratory housekeeping of the equipment and other related accessories.

8) Make sure that the last connection to be made in your circuit is the power supply and first thing to be disconnected is also the power supply.
9) Equipment should not be removed, transferred to any location without permission from the laboratory staff.
10) Students are not allowed to use any equipment without proper orientation and actual hands on equipment operation.
11) Smoking and drinking in the laboratory are not permitted.

All these rules and regulations are necessary precaution in Electronic Laboratory to safeguard the students, laboratory staff, the equipment and other laboratory users.

Table of Contents

LAB # 1: To Identify the Responses of Different Logic Gates using Hardware and Software Platforms

Objectives

- To identify the basic Boolean functions of logic gates
- To identify various ICs used in proteus software and to perform basic functionality of logic gates

Pre Lab

Part A -Familiarize yourself with Logic Gates

The basic logic gates are the building blocks of more complex logic circuits. These logic gates perform the basic Boolean functions, such as AND, OR, NAND, NOR, Inversion, Exclusive-OR, Exclusive-NOR. The circuit symbols are shown below with their Boolean functions, and truth Tables. It can be depicted from the given symbols that each gate has one or two binary inputs, A and B, and one binary output, C. The small circle on the output of the circuit symbols designates the logic complement. The AND, OR, NAND, and NOR gates can be extended to have more than two inputs. A gate can be extended to have multiple inputs if the binary operation it represents is commutative and associative.

These basic logic gates are implemented as small-scale integrated circuits (SSICs) or as part of more complex medium scale (MSI) or very large-scale (VLSI) integrated circuits. Digital IC gates are classified not only by their logic operation, but also the specific logic-circuit family to which they belong. Each logic family has its own basic electronic circuit upon which more complex digital circuits and functions are developed. The following logic families are the most frequently used.

- TTL Transistor-transistor logic
- ECL Emitter-coupled logic
- MOS Metal-oxide semiconductor
- CMOS Complementary metal-oxide semiconductor

TTL and ECL are based upon bipolar transistors. TTL has a well-established popularity among logic families. ECL is used only in systems requiring high-speed operation. MOS and CMOS, are based on field effect transistors. They are widely used in large scale integrated circuits because of their high component density and relatively low power consumption. CMOS logic consumes far less power than MOS logic. There are various commercial integrated circuit chips

available. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

Read and understand pin configuration and list the truth tables of AND, OR, NOT, NAND, NOR and XOR gates.

In- Lab

Lab Tasks-Part-A

Study and verify the truth table of logic gates

Lab Task 1:

IC 7400 NAND gate

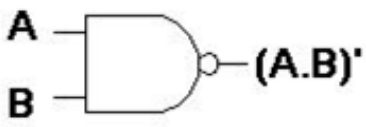


Fig: 1.1

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

Table: 1.1

Lab Task 2:

IC 7408 AND gate

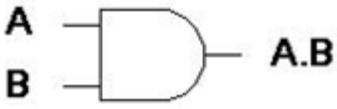


Fig: 1.2

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

Table: 1.2

Lab Task 3:

IC 7404 NOT gate

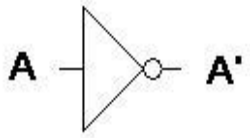


Fig: 1.3

Input	Output
A	
0	
1	

Table: 1.3

Lab Task 4:

IC 7402 NOR gate

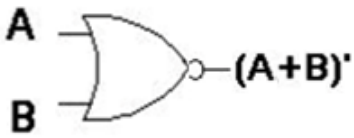


Fig: 1.4

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

Table: 1.4

Lab Task 5:

IC 7432 OR gate

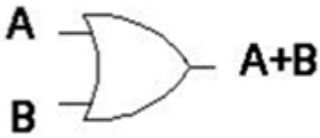


Fig: 1.5

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

Table: 1.5

Lab Task 6:

IC 7486 XOR gate

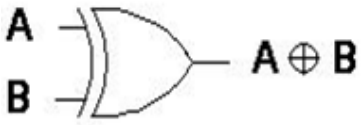


Fig: 1.6

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

Table: 1.6

Part B -Familiarize yourself with PROTEUS

Proteus is a software for microprocessor simulation, schematic capture, and printed circuit board (PCB) design. It is developed by Labcenter Electronics.

ISIS Schematic Capture - a tool for entering designs.

PROSPICE Mixed mode SPICE simulation - industry standard SPICE3F5 simulator combined with a digital simulator.

ARES PCB Layout - PCB design system with automatic component placer, rip-up and retry auto-router and interactive design rule checking.

VSM - Virtual System Modelling lets co-simulate embedded software for popular micro-controllers alongside hardware design.

System Benefits

Integrated package with common user interface and fully context sensitive help. But we only use the ISIS Schematic Capture

In Lab:

Lab Tasks-Part-B

Right click on the ISIS icon present on desktop and then open it.

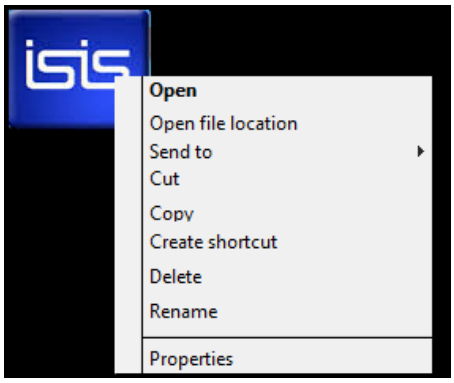


Fig: 1.7

The window below will be shown

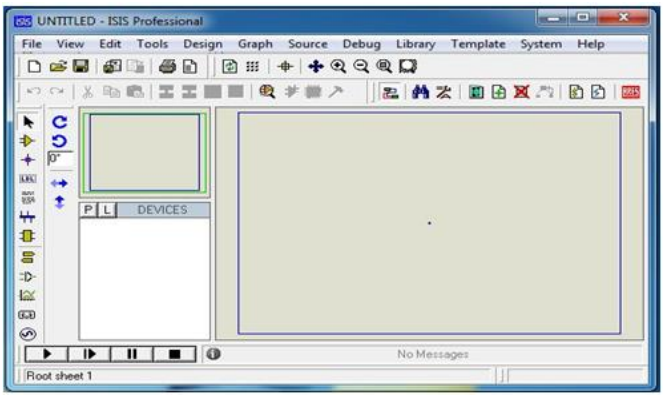


Fig: 1.8

Lab Task 1:

Implementation of AND gate on Proteus

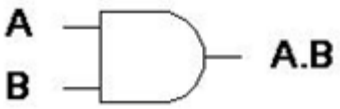


Fig: 1.9

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

Table: 1.7

Take the components From Library by pressing the tab “Library”

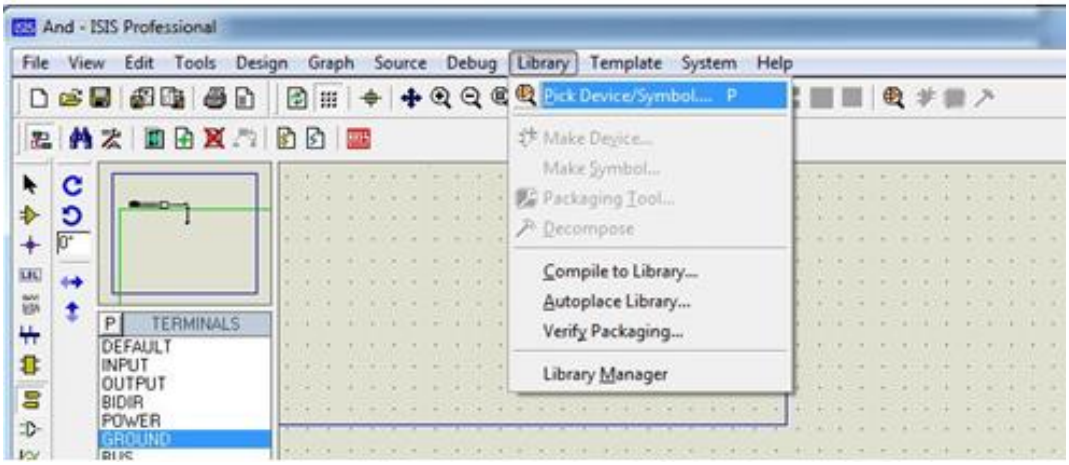


Fig: 1.10

Search the Components by names and then click **OK** to adding them in Devices Panel

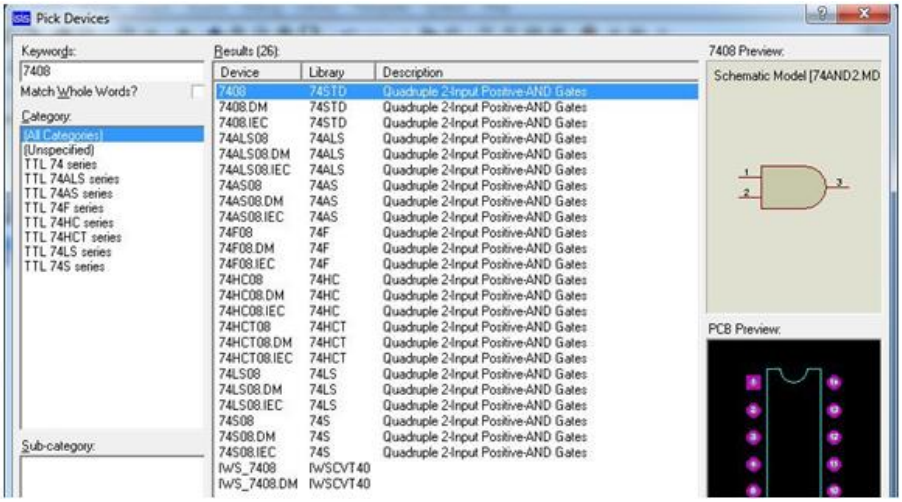


Fig: 1.11

Connect the components and save the design by specific name

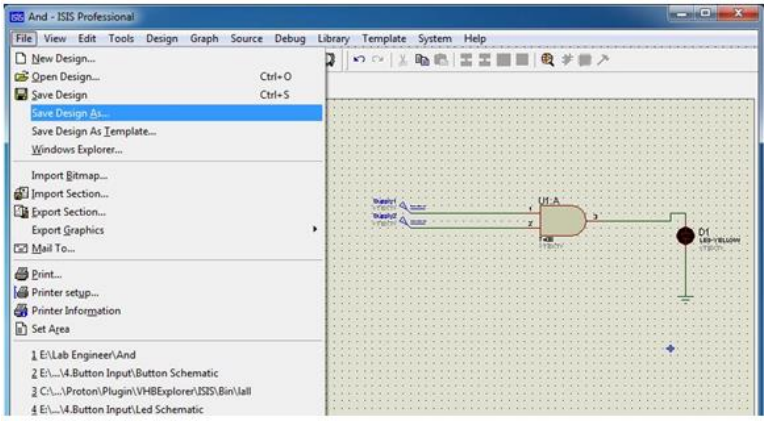


Fig: 1.12

Press the "**start"** button to execute the design and verify the results

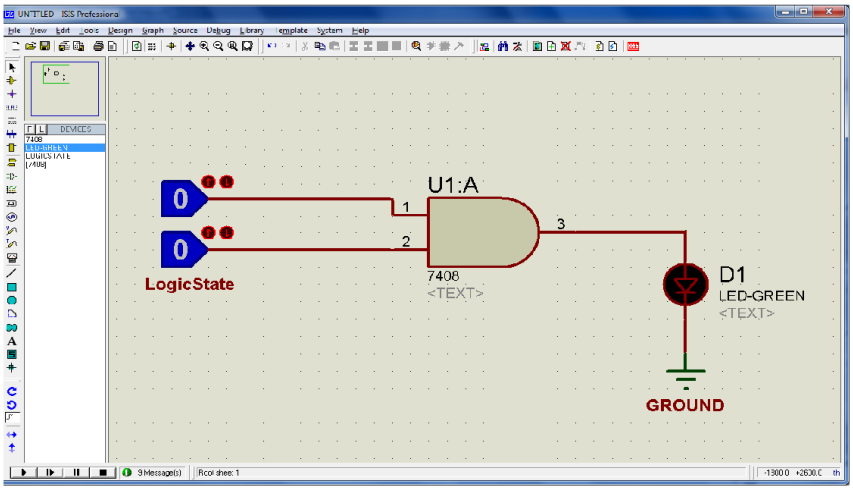


Fig: 1.13

Repeat the same procedure for the followings Gates.

Lab Task 2:

Implementation of OR gate on Proteus

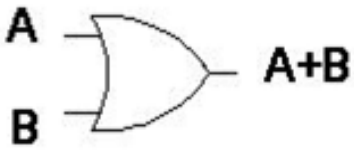


Fig: 1.14

Inputs		Output
A	**B**	
0	0	
0	1	
1	0	
1	1	

Table: 1.8

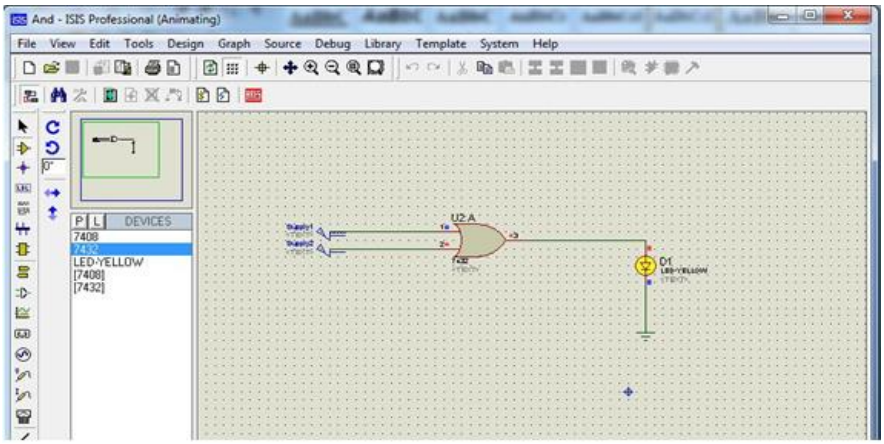


Fig: 1.15

Lab Task 3:

Implementation of NOT gate on Proteus

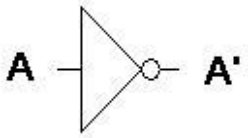


Fig: 1.16

Input	Output
A	
0	
1	

Table: 1.9

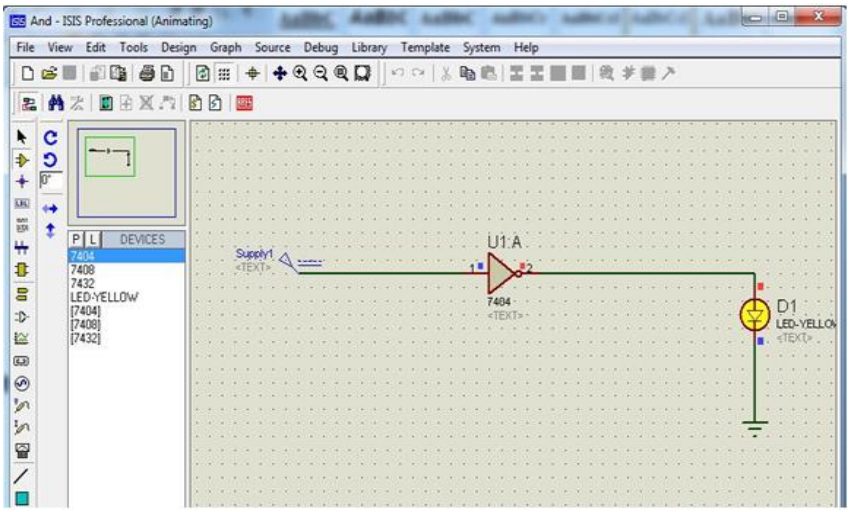


Fig: 1.17

Lab Task 4:

Implementation of NAND gate on Proteus

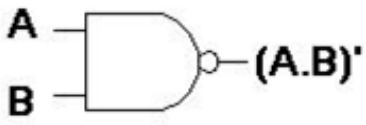


Fig: 1.18

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

Table: 1.10

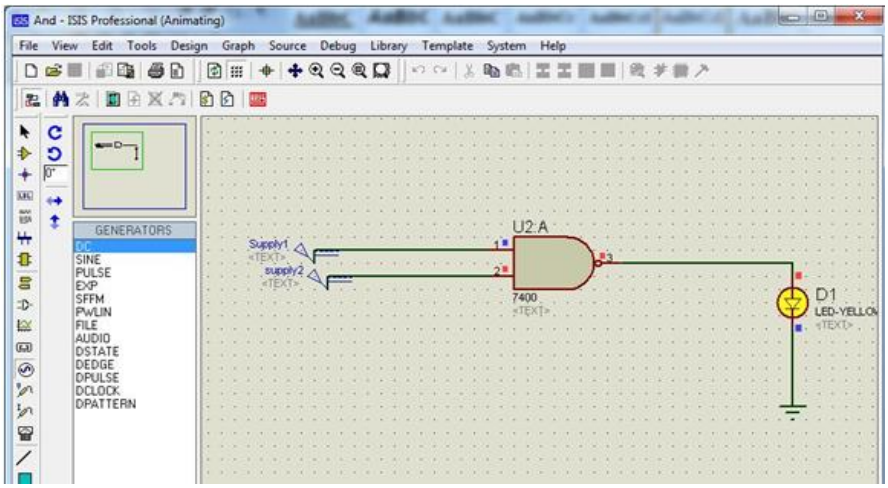


Fig: 1.19

Lab Task 5:

Inputs		Output
A	B	
0	0	
0	1	

Implementation of XOR gate on Proteus

1	0	
1	1	

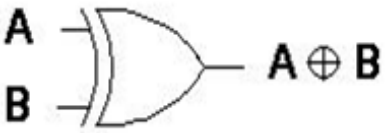


Fig: 1.20

Table: 1.11

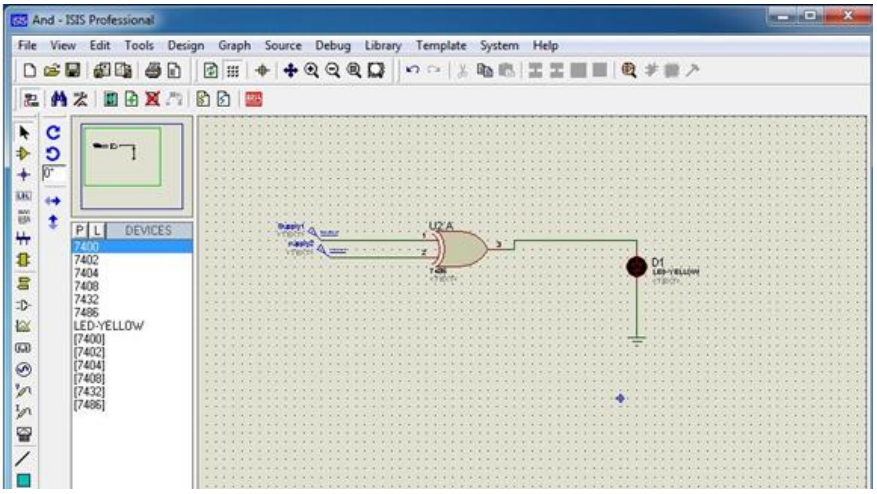


Fig: 1.21

Post Lab:

Draw the schematic for following logic circuit in Proteus and fill in the table. Answer the questions at the end.

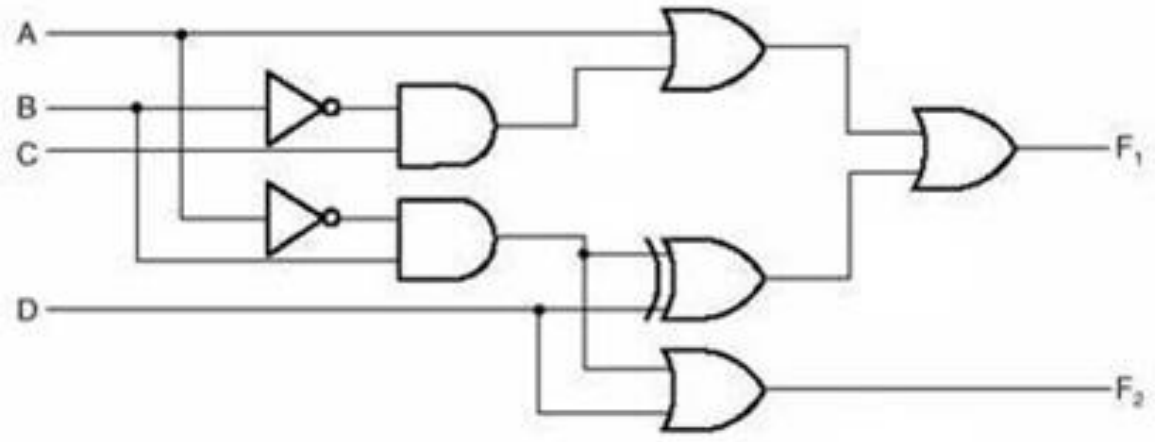


Fig: 1.22

A	B	C	D	F1	F2
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Table: 1.12

Post LAB:

Design XOR and XNOR circuits using AND, NOT and OR gates.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 2: To Explain the Universality of NAND and NOR GATES in Order to Design Other Logic Gates

Objectives

- To construct various logic gates using NAND gates
- To construct various logic gates using NOR gates

Pre-Lab:

Background theory:

Digital circuits are more frequently constructed with NAND or NOR gates than with AND and OR gates. NAND and NOR gates are easier to fabricate with electronic components and are the basic gates used in all IC digital logic families. Because of the prominence of NAND and NOR gates in the design of digital circuits, rules and procedures have been developed for conversion from Boolean function given in terms of AND, OR, and NOT into equivalent NAND and NOR logic diagram.

Read and understand the universal gates also list the truth tables of AND, OR, NOT, NAND, NOR and XOR gates.

In-Lab

Part 1 - Implementing any logic expression by using only NAND gates

If we can show that the logical operations AND, OR, and NOT can be implemented with NAND gates, then it can be safely assumed that any Boolean function can be implemented with NAND gates. Figure-1 below shows such implementation: -

Procedure

- ✓ Insert the IC on the trainer's breadboard.
- ✓ Use any one or more of the NAND gates of the IC for this experiment.
- ✓ Any one or more Logic Switches of the trainer (S1 to S9) can be used for input to the NAND gate.
- ✓ For output indication, connect the output pin of the circuit to any one of the LEDs of the trainer (L0 to L15).

Lab Tasks-Part-1

Lab Task 1:

Verification of AND function

- ✓ Connect the circuit as per figure 1(a) above.
- ✓ Connect +5V to pin 14 (Vcc) and Ground to pin 7 (GND) of the IC.
- ✓ By setting the switches to 1 and 0, verify that the output of the circuit conforms to that of an AND gate. Record your observation in the table below

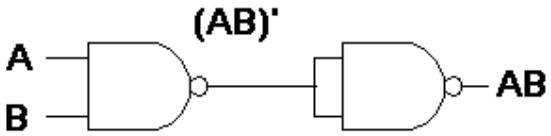


Fig: 2.1

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

Table: 2.1

Lab Task 2

Verification of OR function

- ✓ Connect the circuit as per Figure 1(b).
- ✓ Connect +5V to pin 14 (Vcc) and Ground to pin 7 (GND) of the IC.
- ✓ By setting the switches to 1 and 0, verify that the output of the circuit conforms to that of an OR gate. Record your observation in the table below

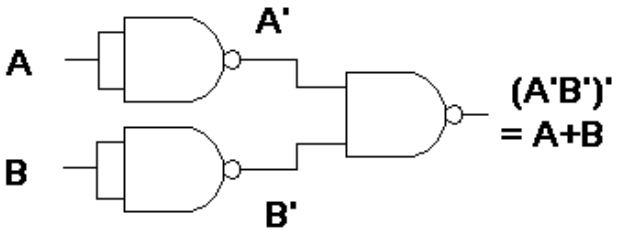


Fig: 2.2

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

Table: 2.2

Lab task 3

Verification of NOT function

- ✓ Connect the circuit as per Figure 1 (c)
- ✓ Connect +5V to pin 14 (Vcc) and Ground to pin 7 (GND) of the IC.
- ✓ By setting the switch to 1 and 0, verify that the output of the circuit conforms to that of a NOT gate. Record your observation in the table below

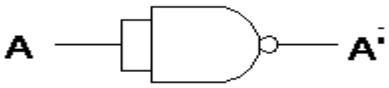


Fig: 2.3

Input	Output
A	
0	
1	

Table: 2.3

Part 2 - Implementing any logic expression by using only NOR gates

NOR function is the dual of the NAND function, hence all procedure and rules for NOR logic form a dual of the corresponding procedures and rules developed for the NAND logic. Figure 2.1 below shows how NOR gates can be used to create AND, OR, and INVERTER gates.

Procedure

- ✓ Insert IC on the trainer's bread board.
- ✓ Use any one or more of the NOR gates of the IC for this experiment.
- ✓ Any one or more Logic Switches of the trainer (S2 to S9) can be used for input to the NOR gate.
- ✓ For output indication, connect the output pin of the circuit to any one of the LEDs of the trainer (L0 to L15).

Lab Tasks-Part-2

Lab Task 1:

Verification of AND function

- ✓ Connect the circuit as per Figure 1 (a)
- ✓ Connect +5V to pin 14 (Vcc) and Ground to pin 7 (GND) of the IC.
- ✓ By setting the switches to 1 and 0, verify that the output of the circuit conforms to that of an AND gate. Record your observation in the table below

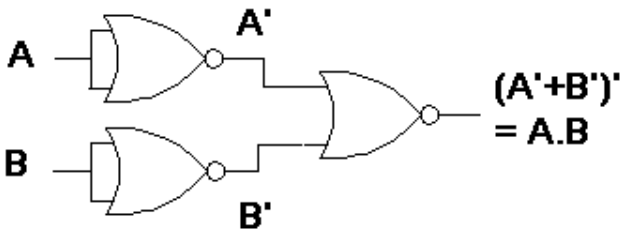


Fig: 2.4

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

Table: 2.4

Lab task 2:

Verification of OR function

- ✓ Connect the circuit as per Figure 1 (b)
- ✓ Connect +5V to pin 14 (Vcc) and Ground to pin 7 (GND) of the IC.
- ✓ By setting the switches to 1 and 0, verify that the output of the circuit conforms to that of an OR gate. Record your observation in the table below

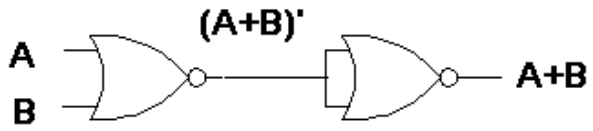


Fig: 2.5

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

Table: 2.5

Lab task 3

Verification of NOT function

- ✓ Connect the circuit as per Figure 1 (c)
- ✓ Connect +5V to pin 14 (Vcc) and Ground to pin 7 (GND) of the IC.

✓ By setting the switch to 1 and 0, verify that the output of the circuit conforms to that of an NOT gate. Record your observation in the table below

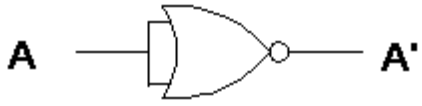


Fig: 2.6

Input	Output
A	
0	
1	

Table: 2.6

Post-Lab:

1- Design NOR, XOR and XNOR gate Using NAND gate only.

2- Design NAND, XOR and XNOR gate Using NOR gate only.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 3: To Show the output of simplified Boolean Expression by following the k-map, using Hardware and Software Platforms

Objectives

Part 1

- To display the Boolean expression in its simplified form using Karnaugh map
- To display the results of simplified Boolean expression using hardware and software platforms
- To show simplified Boolean expression using universal gates

Pre-Lab

Part 1 -Introduction to simplify the Boolean Expression

Canonical Forms (Normal Forms): Any Boolean function can be written in disjunctive normal form (sum of min-terms) or conjunctive normal form (product of max-terms).

A Boolean function can be represented by a Karnaugh map in which each cell corresponds to a minterm. The cells are arranged in such a way that any two immediately adjacent cells correspond to two minterms of distance 1. There is more than one way to construct a map with this property.

Karnaugh Maps

For a function of two variables, say, $f\ (x, y)$,

	x'	x
y'	f(0,0)	f(1,0)
y	f(0,1)	f(1,1)

For a function of three variables, say, $f(x, y, z)$

	x'y'	x'y	xy	xy'
z'	f(0,0,0)	f(0,1,0)	f(1,1,0)	f(1,0,0)
z	f(0,0,1)	f(0,1,1)	f(1,1,1)	f(1,0,1)

For a function of four variable $f(w, x, y, z)$

	w'x'	w'x	wx	wx'
y'z'	0	4	12	8
y'z	1	5	13	9
yz	3	7	15	11
yz'	2	6	14	10

Read and understand the methods for simplification of Boolean function. List a truth table, derive its Boolean expression and use K-Map for simplification of the derived Boolean expression.

In-Lab

Lab Tasks-Part-1

Lab Task 1:

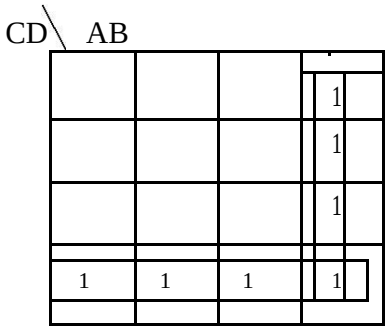
Realization of Boolean expression:

TRUTH TABLE

INPUTS				OUTPUT
A	**B**	**C**	**D**	**Y**
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Table: 3.1

$$Y = A'B'CD' + A'BCD' + AB'C'D' + AB'C'D + AB'CD' + AB'CD + ABCD'$$



$$Y = AB' + C\,D'$$

Lab Task 2:

For the given truth table simplify the Boolean expression by using K-Map

Truth Table

Inputs				Output
A	**B**	**C**	**D**	**Y**
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Table: 3.2

Karnaugh Maps:

Simplified Boolean Expression:

Part 2 –Boolean Expression using Basic gates

Lab Tasks-Part-2

Lab Task 1:

Implement the following simplified function by using basic gates as mention in Task 1 Part 1

$$Y = AB' + C\ D'$$

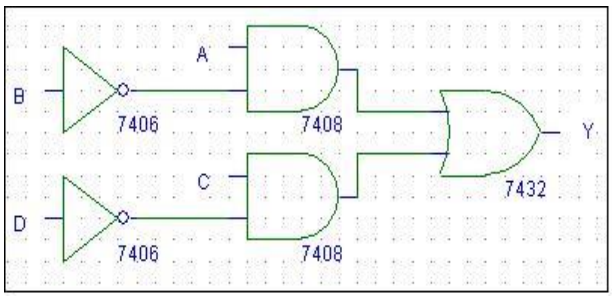


Fig: 3.1

Lab Task 2:

Implement the Task 2 of Part 1 by using basic gate

Part 3 –Boolean Expression using Universal gates (NAND & NOR)

A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

Post-Lab

Implement the Task 2 of Part 1 by using universal gates (NAND & NOR).

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 4: To Show the Behaviour of Binary Adder/Subtractor and reproduce its circuit using Universal Gates

Objectives

- To identify basic functionality of a Binary Adder using logic gates
- To identify basic functionality of a Binary Subtractor using logic gates
- To design a binary adder using universal gates (NAND, NOR)
- To design a binary subtractor using universal gates (NAND, NOR)

Pre-Lab

Part 1 -Familiarize yourself with Half Adder & Full Adder

Half Adder

A combinational logic circuit that performs the addition of two data bits, A and B, is called a half-adder. Addition will result in two output bits; one of which is the sum bit, S, and the other is the carry bit, C. The Boolean functions describing the half-adder are:

$$S = A \oplus B$$

$$C = A\ B$$

Full Adder

The half-adder does not take the carry bit from its previous stage into account. This carry bit from its previous stage is called carry-in bit. A combinational logic circuit that adds two data bits, A and B, and a carry-in bit, C_{in}, is called a full-adder. The Boolean functions describing the full-adder are:

$$S = (x \oplus y) \oplus C_{in}$$

$$C = xy + C_{in}\ (x \oplus y)$$

Half Subtractor

Subtracting a single-bit binary value B from another A (i.e. A-B) produces a difference bit D and a borrow out bit B-out. This operation is called half subtraction and the circuit to realize it is called a half subtractor. The Boolean functions describing the half-Subtractor are:

$$S = A \oplus B$$

$$C = A'\ B$$

Full Subtractor

Subtracting two single-bit binary values, B, C_{in} from a single-bit value A produces a difference bit D and a borrow out Br bit. This is called full subtraction. The Boolean functions describing the full-subtractor are:

$$D = (x \oplus y) \oplus C_{in}$$

$$Br = A'B + A'(C_{in}) + B(C_{in})$$

In-Lab

Lab Tasks-Part-1

Lab Task 1

Implement half adder by using basic gates and verify the results

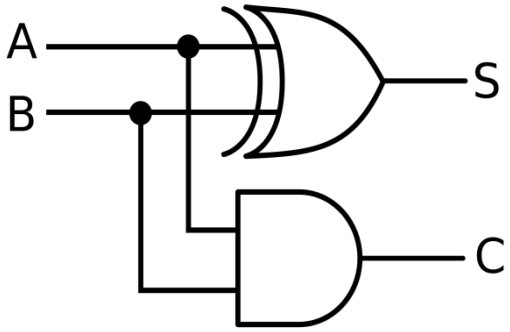


Fig: 4.1

Truth Table

INPUTS		OUTPUTS	
		Observed Values	
A	**B**	**S**	**C**
0	0		
0	1		
1	0		
1	1		

Table: 4.1

Lab Task 2:

Implement half adder by using NAND gates and verify the results

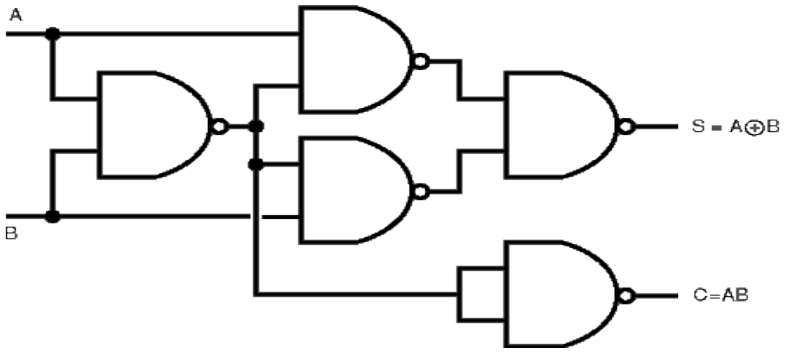


Fig: 4.2

Truth Table

INPUTS		OUTPUTS	
		Observed Values	
A	**B**	**S**	**C**
0	0		
0	1		
1	0		
1	1		

Table: 4.2

Lab Task 3

Implement full adder by using basic gates and verify the results

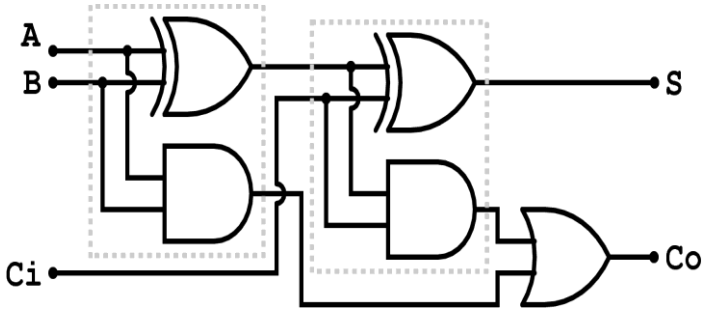


Fig: 4.3

Truth Table

INPUTS			OUTPUTS Observed Values	
A	B	C_{in}	S	C
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table: 4.3

Lab Task 4

Implement full adder by using NAND gates and verify the results

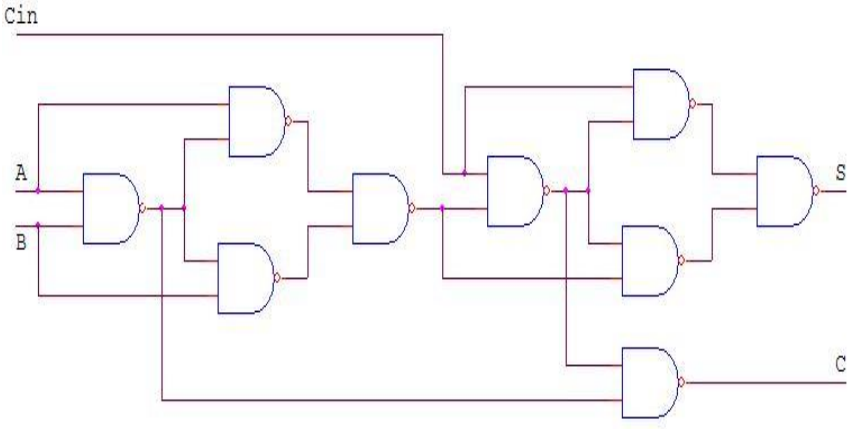


Fig: 4.4

Truth Table

INPUTS			OUTPUTS	
			Observed Values	
A	B	C_{in}	S	C
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table: 4.4

Lab Tasks-Part-2

Lab Task 1

Implement half subtractor by using basic gates and verify the results

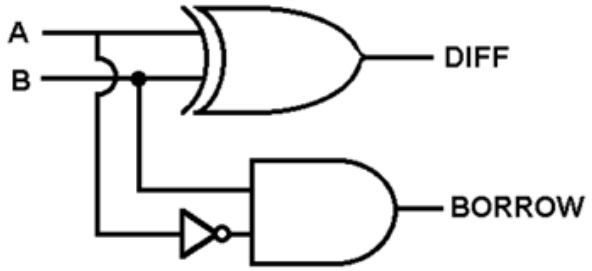


Fig: 4.5

Truth Table

INPUTS		OUTPUTS	
		Observed Values	
A	B	D	Br

0	0		
0	1		
1	0		
1	1		

Table: 4.5

Lab Task 2:

Implement half subtractor by using NAND gates and verify the results

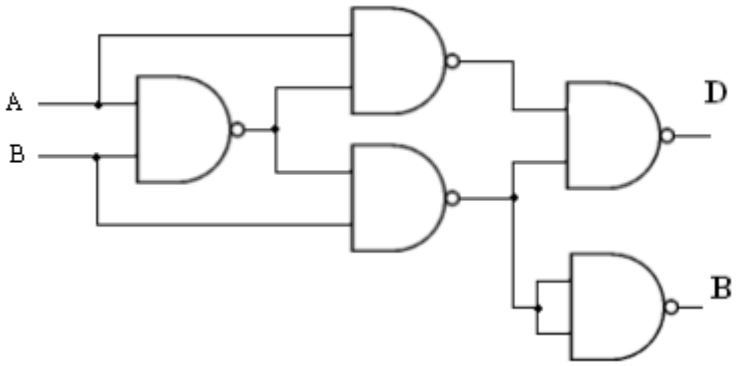


Fig: 4.6

Truth Table :

INPUTS		OUTPUTS Observed Values	
A	B	D	Br
0	0		
0	1		
1	0		
1	1		

Table: 4.6

Lab Task 3:

Implement full subtractor by using basic gates and verify the results

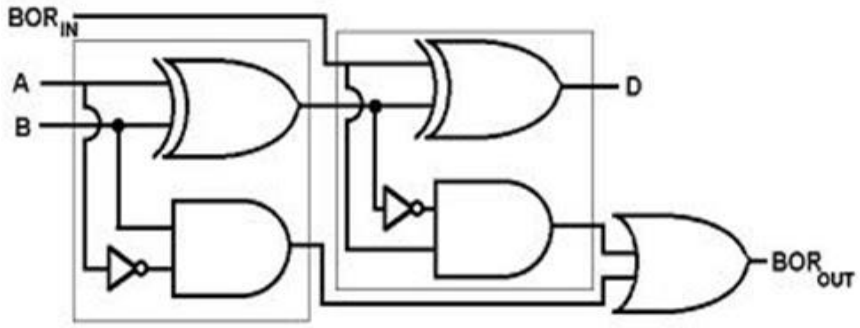


Fig. 4.7

Truth Table:

INPUTS			OUTPUTS Observed Values	
A	B	Bor_{in}	D	Bor_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table: 4.7

Lab Task 4

Implement full subtractor by using NAND gates and verify the results

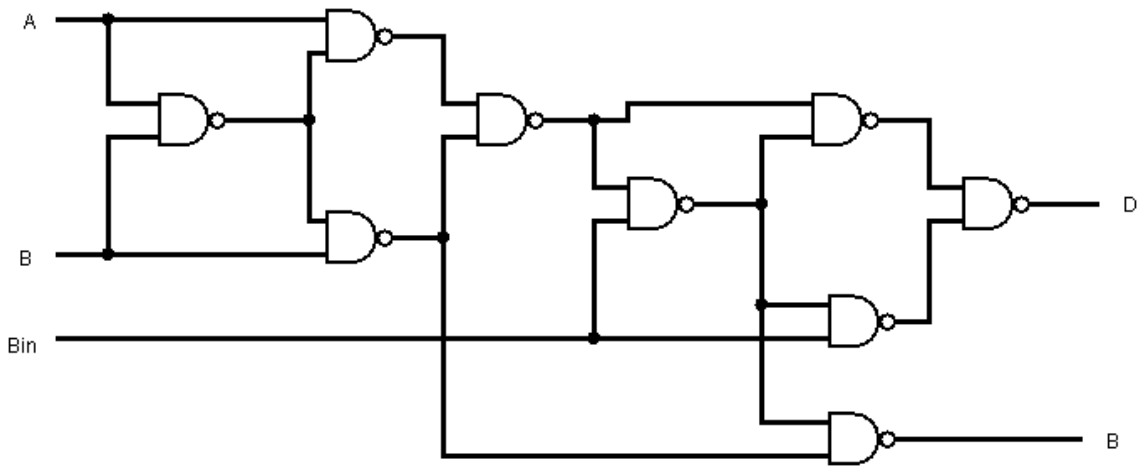


Fig: 4.8

Truth Table

INPUTS			OUTPUTS Observed Values	
A	B	Bor_{in}	D	Bor_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table: 4.8

Post-Lab:

Design half adder, full adder, half subtractor and full subtractor using NOR gates.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 5: To Follow the Steps of BCD to Excess 3 Code Conversion and Reproduce the Results using Dedicated IC

Objectives

- To understand and respond to the steps required to transform a four bit BCD to Excess 3 code conversion.
- To reproduce the steps required to transform Excess 3 code to BCD using IC 7483

Pre Lab

Part 1 -Familiarize yourself with BDC to Excess-3

Code converter is a combinational circuit that translates the input code word into a new corresponding word. The excess-3 code digit is obtained by adding three to the corresponding BCD digit. To Construct a BCD-to-excess-3-code converter with a 4-bit adder feed BCD-code to the 4-bit adder as the first operand and then feed constant 3 as the second operand.

Part 2 -Familiarize yourself with Excess-3 to BCD

To make the above mention circuit work as a excess-3 to BCD converter, we feed excess-3 code as the first operand and then feed 2's complement of 3 as the second operand. The output is the BCD code.

In Lab

Lab Tasks-Part-1

Lab Task 1

Use the following circuit to verify the result

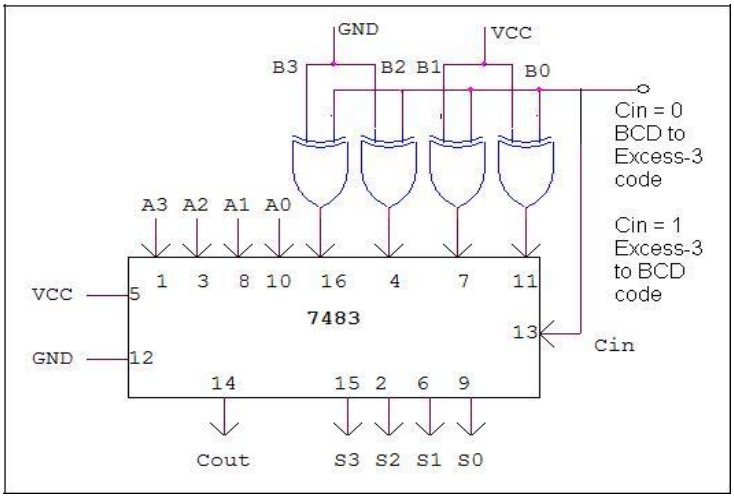


Fig. 5.1

Truth Table

INPUTS				OUTPUTS(Ex-3)			
BCD				Observed Values			
A	B	C	D	w	x	y	z
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				

Table: 5.1

Lab Tasks-Part-2

Lab Task 2:

Using the following circuit and verify the result

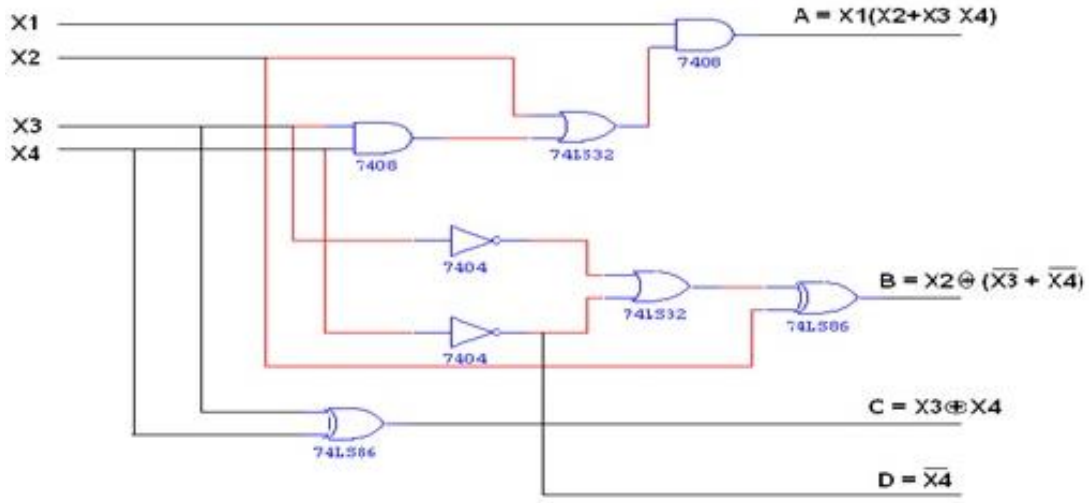


Fig: 5.2

Truth Table

INPUTS				OUTPUTS (Ex-3)			
Ex-3				BCD			
X1	X2	X3	X4	A	B	C	D
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
0	0	0	0				
0	0	0	1				
0	0	1	0				
1	0	1	1				
1	1	0	0				

Table: 5.2

Post Lab

Design Excess-3 to BCD converter using NAND Gates.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 6: To Follow the Steps of Binary to Gray Code Conversion and Reproduce the Results using Logic Gates

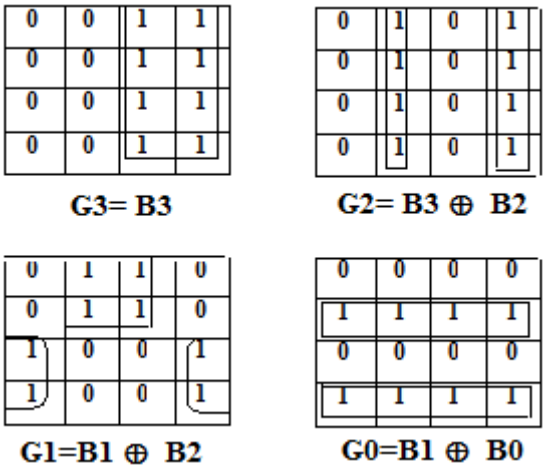
Objectives

- To understand and respond to the steps required to transform a four bit BCD to Gray code conversion
- To reproduce the steps required to transform Gray code to BCD using IC 7483

Pre Lab

Lab Tasks-Part-1

Binary Code to Gray Code Conversion



Boolean Expressions:

G3=B3; G2=B3 ⊕ B2
G1=B1 ⊕ B2; G0=B1 ⊕ B0

Lab Tasks-Part-2

Gray Code to Binary Code Conversion

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

B3 = G3

0	1	0	1
0	1	0	1
0	1	0	1
0	1	0	1

B2=G3 ⊕ G2

0	1	0	1
0	1	0	1
1	0	1	0
1	0	1	0

B1=G3 ⊕ G2 ⊕ G1

0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0

B0=G3 ⊕ G2 ⊕ G1 ⊕ G0

Boolean Expressions:

B3=G3; **B2=G3 ⊕ G2**

B1=G3 ⊕ G2 ⊕ G1; **B0=G3 ⊕ G2 ⊕ G1 ⊕ G0**

In Lab:

Lab Task 1:

Use following circuit to verify the results

Realization using XOR gates

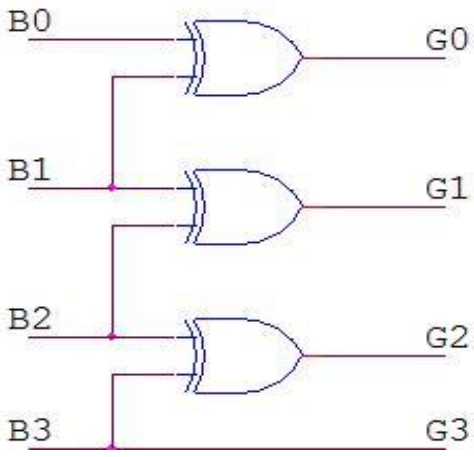


Fig: 6.1

Truth Table

Binary Inputs				Gray Observed Output			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Table: 6.1

Lab Task 2:

Realization using NAND gates

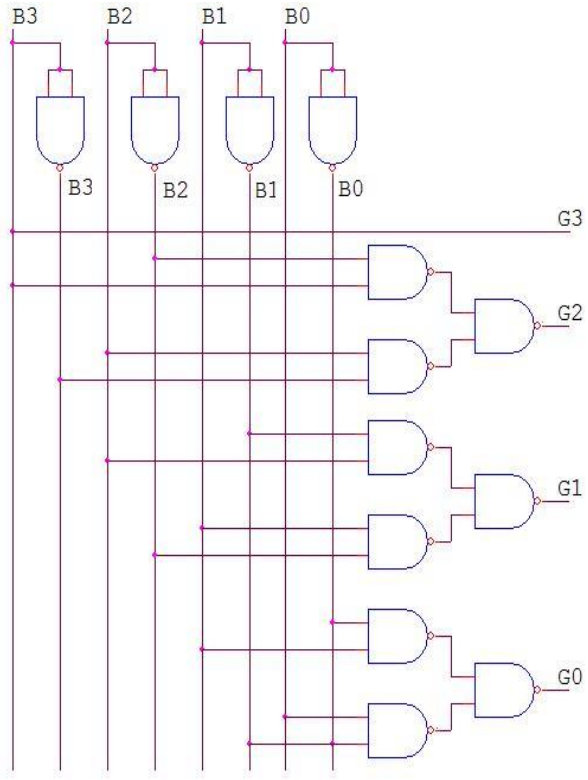


Fig: 6.2

Lab Task 3:

Use following circuit to verify the results

Realization using XOR gates

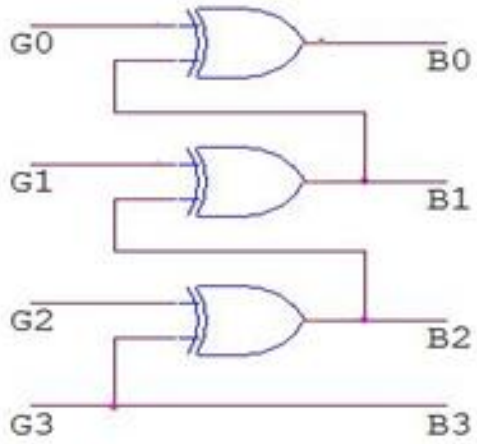


Fig: 6.3

Truth Table

Gray				Binary			
Inputs				Observed Output			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0				
0	0	0	1				
0	0	1	1				
0	0	1	0				
0	1	1	0				
0	1	1	1				
0	1	0	1				
0	1	0	0				
1	1	0	0				
1	1	0	1				
1	1	1	1				
1	1	1	0				
1	0	1	0				
1	0	1	1				
1	0	0	1				
1	0	0	0				

Table: 6.2

Lab Task 4:

Realization using NAND gates

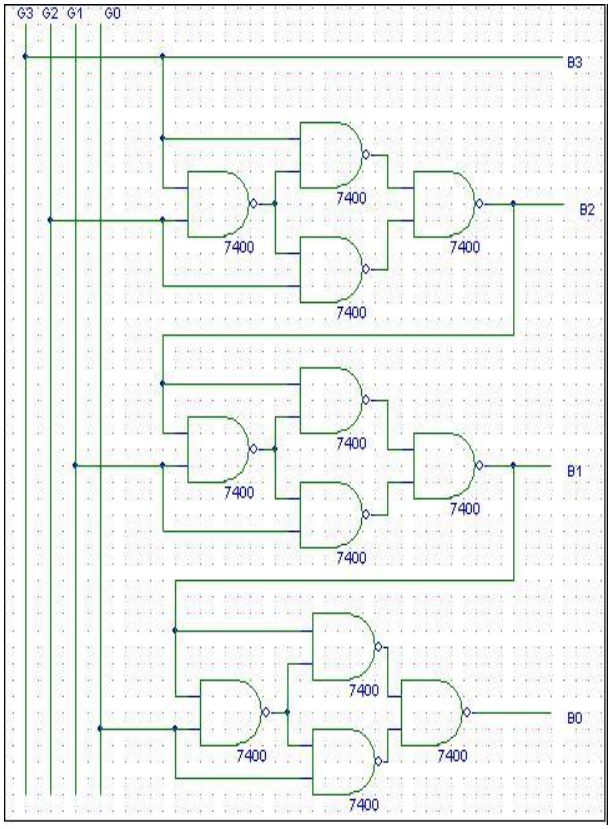


Fig: 6.4

Post Lab:

1. Design grey code to binary code converter using NOR gates.
2. Design binary code to grey code converter using NOR gates.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB: 7 To Show the Response of a Multiplexer/Demultiplexer and to Reproduce Adder/Subtractor using Multiplexer

Objectives

- To explain the basic functionality and working of multiplexer using logic gates
- To explain the basic functionality and working of demultiplexer using logic gates
- To display the results of various applications of multiplexer using hardware and software platforms

Pre Lab

Part 1 -Familiarize yourself with Multiplexers

Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output. By using control signals (select lines) we can select any input to the output. Multiplexer is also called as data selector because the output bit depends on the input data bit that is selected. The general multiplexer circuit has 2^n input signals, n control/select signals and 1 output signal.

Part 2 -Familiarize yourself with Demultiplexers

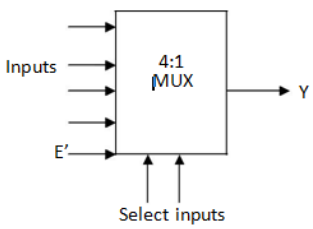
De-multiplexers perform the opposite function of multiplexers. They transfer a small number of information units over a larger number of channels under the control of selection signals. The general de-multiplexer circuit has 1 input signal, n control/select signals and 2^n output signals. De-multiplexer circuit can also be realized using a decoder circuit with enable.

In lab

Lab Tasks-Part-1

Lab Task 1:

4:1 Multiplexer



Output Y= E'S1'S0'I0 + E'S1'S0I1 + E'S1S0'I2 + E'S1S0I3

Realization using NAND Gates

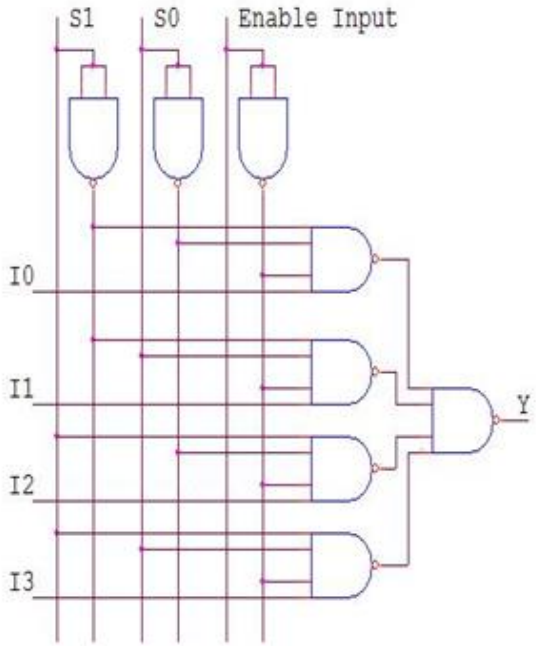


Fig: 7.1

Lab Task 2:

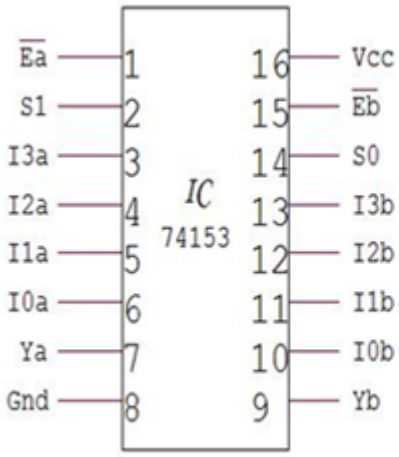
Truth Table

Select Inputs		Enable Input	Inputs				Outputs
S1	**S0**	**E**	**I0**	**I1**	**I2**	**I3**	**Y**
X	X	1	X	X	X	X	
0	0	0	0	X	X	X	
0	0	0	1	X	X	X	
0	1	0	X	0	X	X	
0	1	0	X	1	X	X	
1	0	0	X	X	0	X	
1	0	0	X	X	1	X	
1	1	0	X	X	X	0	
1	1	0	X	X	X	1	

Table: 7.1

Lab Task 3:

Verify IC 74153 MUX (Dual 4:1 Multiplexer)



Select Inputs		Enable Input	Inputs				Output
S1	S0	E	I0	I1	I2	I3	Y
X	X	1	X	X	X	X	
0	0	0	0	X	X	X	
0	0	0	1	X	X	X	
0	1	0	X	0	X	X	
0	1	0	X	1	X	X	
1	0	0	X	X	0	X	
1	0	0	X	X	1	X	
1	1	0	X	X	X	0	
1	1	0	X	X	X	1	

Fig: 7.2 Table: 7.2

Tasks-Part-2

Lab Task 1:

Realization using NAND Gates

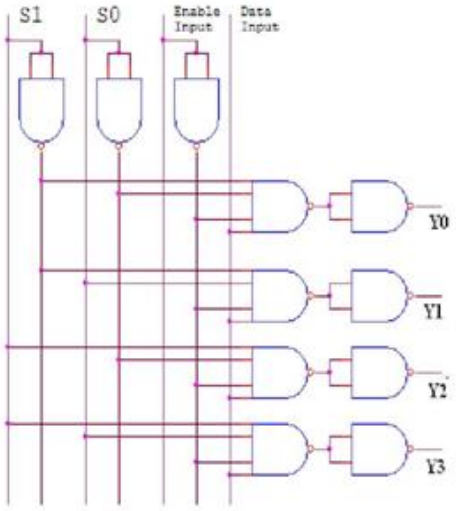


Fig: 7.3

Lab Task 2:

Truth Table

Enable Input	Data Input	Select Inputs		Outputs			
E	D	S_1	S_0	Y_3	Y_2	Y_1	Y_0

1	0	X	X				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				

Table: 7.3

Lab Task 3:

Verify IC 74139 DEMUX

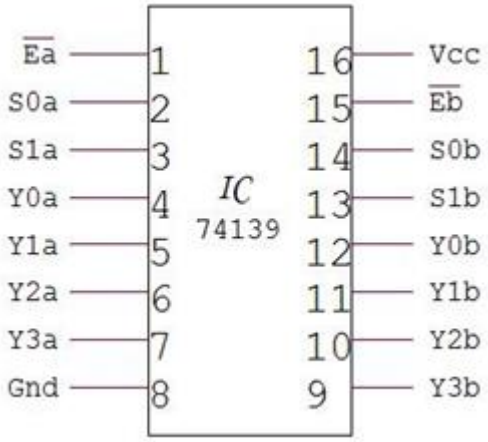


Fig: 7.4

Truth Table

Inputs			**Outputs**			
Ea	**S_1**	**S_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
1	X	X				
0	0	0				
0	0	1				
0	1	0				
0	1	1				

Table: 7.4

Part 3 – Applications of Multiplexers:

To set up a Half/Full Adder and Half/Full Subtractor using IC 74153.

Lab Tasks-Part-3

Lab Task 1: Half Adder using MUX:

Design:

SUM

I0	I1
0	1
2	3
A	**A'**

CARRY

I0	I1
0	1
2	3
0	**A**

Circuit Diagram:

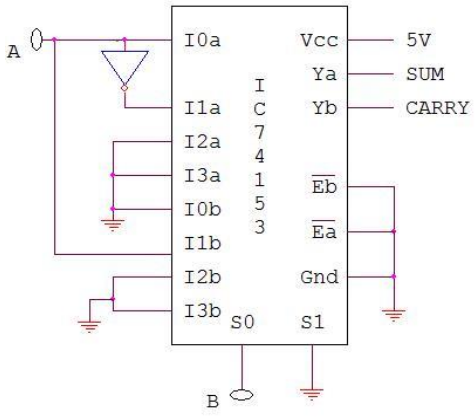


Fig: 7.5

Truth Table:

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

Table: 7.5

Lab Task 2: Full Adder using MUX:

Design:

SUM

I0	I1	I2	I3
0	1	2	3
4	5	6	7
A	**A'**	**A'**	**A**

CARRY

I0	I1	I2	I3
0	1	2	3
4	5	6	7
0	**A**	**A**	**1**

Circuit Diagram:

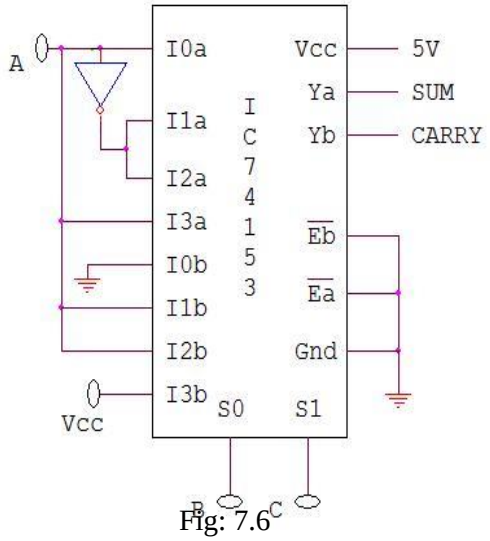


Fig: 7.6

Truth Table:

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

Table: 7.6

Lab Task 3: Half Subtractor using MUX:

Design:

DIFFERENCE	
I0	**I1**
0	1
2	3
A	**A'**

BORROW	
I0	**I1**
0	1
2	3
0	**A'**

Circuit Diagram:

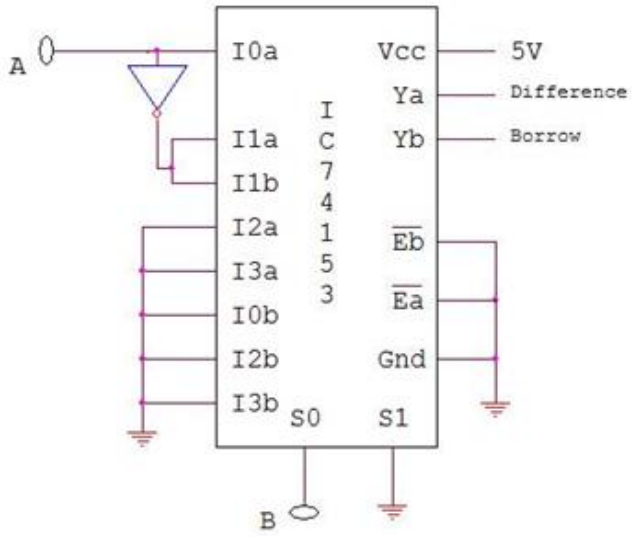


Fig: 7.7

Truth Table:

Inputs		Outputs	
A	**B**	**D**	**Br**
0	0		
0	1		
1	0		
1	1		

Table: 7.7

Lab Task 4: Full Subtractor using MUX:

Design:

DIFFERENCE

I0	I1	I2	I3
0	1	2	3
4	5	6	7
A	A'	A'	A

BORROW

I0	I1	I2	I3
0	1	2	3
4	5	6	7
0	A'	A'	1

Circuit Diagram:

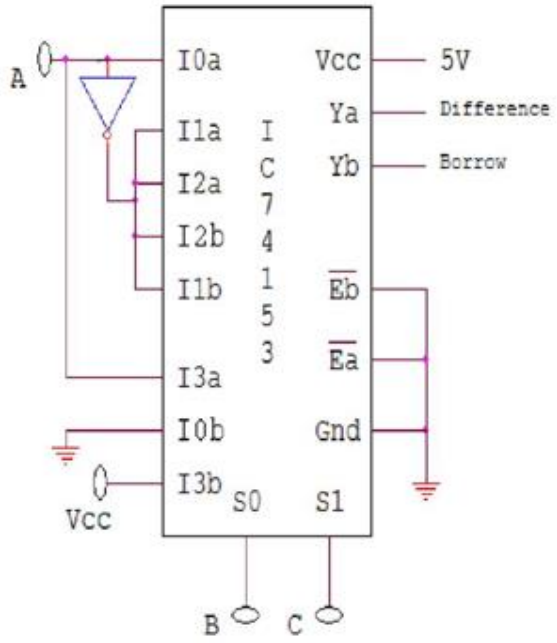


Fig: 7.8

Truth Table:

Inputs			Outputs	
A	B	C	D	Br
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table: 7.8

Post Lab:

Design Half Adder and full Adder using MUX without using IC74153 (using basic gates only).

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 8: To Show the response of Binary Comparator(s) using hardware and software tools and to Reproduce the Comparator using dedicated IC

Objectives

- To compare two input signals using logic gates
- To analyze the working of 4-bit magnitude comparator's IC 7485

Pre Lab

Part 1 -Familiarize yourself with Comparators

Magnitude Comparator is a logical circuit, which compares two signals A and B and generates three logical outputs, whether A > B, A = B, or A < B. IC 7485 is a high speed 4-bit magnitude comparator, which compares two 4-bit words. The A = B Input must be held high for proper compare operation.

In Lab:

Lab Tasks-Part-1

Lab Task 1: 1-Bit Comparator

Design:

$$A>B = A\,\bar{B}$$

$$A<B = \bar{A}\,B$$

$$A=B = \bar{A}\,\bar{B} + AB$$

Circuit Diagram:

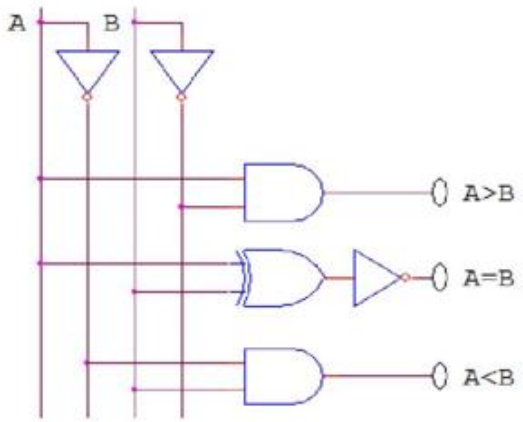


Fig: 8.1

Truth Table:

Inputs		Outputs		
A	B	A>B	A=B	A<B
0	0			
0	1			
1	0			
1	1			

Table: 8.1

Lab Task 2: 2-Bit Comparator

Design:

$$(A>B) = A1\,\bar{B}1 + A0\,B1\,\bar{B}\,0 + \bar{B}\,0A1A0$$

$$(A=B) = (A0 \oplus B0)\,(A1 \oplus B1)$$

$$(A<B) = B1\,\bar{A}1 + B0\,\bar{A}1\,\bar{A}\,0 + \bar{A}\,0B1B0$$

Circuit Diagram:

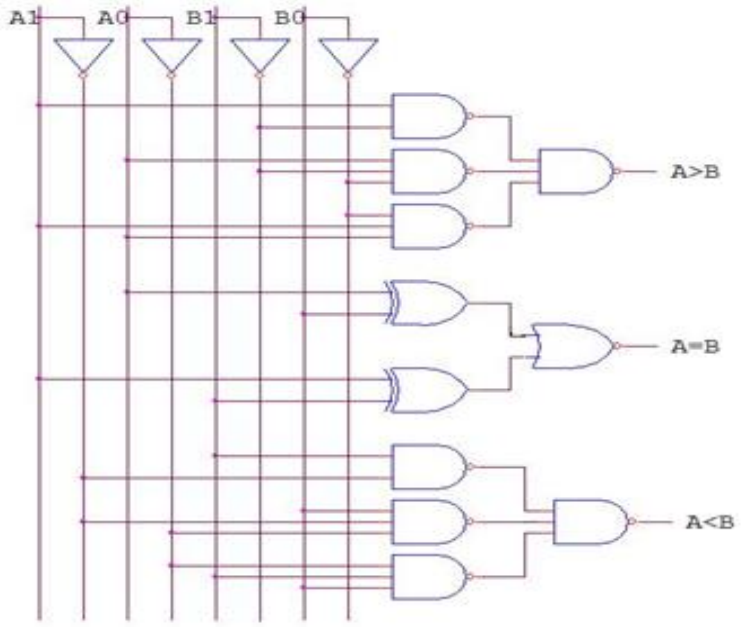


Fig: 8.2

Truth Table:

Inputs				Outputs		
A1	A0	B1	B0	A>B	A=B	A<B
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Table: 8.2

Lab Tasks-Part-2

Lab Task 1: 4-Bit Comparator

Circuit Diagram:

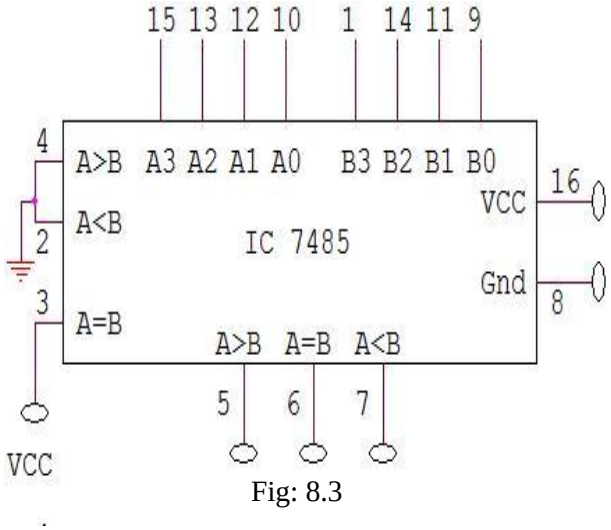


Fig: 8.3

Truth Table:

A				B				Result
A3	A2	A1	A0	B3	B2	B1	B0	
0	0	0	1	0	0	0	0	
0	0	0	1	0	0	0	1	
0	0	0	0	0	0	0	1	

Table: 8.3

Post Lab:

Design 4-Bit comparator using basic gates.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 9: To Show the response of an Encoder/Decoder and to Reproduce the binary converters using basic logic gates

Objectives

Part 1

- To explain the basic functionality and working of Encoder using logic gates
- To explain the basic functionality and working of decoder using logic gates
- To display the results of various applications of encoder using hardware and software platforms
- To display the results of various applications of decoder using hardware and software platforms

Pre Lab

Part 1 - To realize a decoder circuit using basic gates and to verify IC 74LS139

Decoder

A decoder is a combinational circuit that connects the binary information from 'n' input lines to a maximum of 2^n unique output lines. Decoder is also called a min-term generator/max-term generator. A min-term generator is constructed using AND and NOT gates. The appropriate output is indicated by logic 1 (positive logic). Max-term generator is constructed using NAND gates. The appropriate output is indicated by logic 0 (Negative logic).

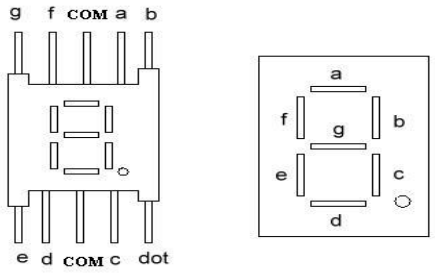
The IC 74139 accepts two binary inputs and when enable provides 4 individual active low outputs. The device has 2 enable inputs (Two active low).

Part 2 - To set up and test a 7-segment static display system to display numbers 0 to 9

The Light Emitting Diode (LED) finds its place in many applications in these modern electronic fields. One of them is the Seven Segment Display. Seven-segment displays contain the arrangement of the LEDs in "Eight" (8) passion, and a Dot (.) with a common electrode, lead (Anode or Cathode). The purpose of arranging it in that passion is that we can make any number out of that by switching ON and OFF the particular LED's. Here is the block diagram of the Seven Segment LED arrangement.

The Light Emitting Diode (LED), finds its place in many applications in this modern electronic fields. One of them is the Seven Segment Display. Seven-segment displays contain the

arrangement of the LEDs in “Eight” (8) passion, and a Dot (.) with a common electrode, lead (Anode or Cathode). The purpose of arranging it in that passion is that we can make any number out of that by switching ON and OFF the particular LED's. Here is the block diagram of the Seven Segment LED arrangement.



Seven-Segment Display

LED’s are basically of two types- Common Cathode (CC) -All the 8 anode legs uses only one cathode, which is common. Common Anode (CA)-The common leg for the entire cathode is of Anode type.

A decoder is a combinational circuit that connects the binary information from ‘n’ input lines to a maximum of 2^n unique output lines. The IC7447 is a BCD to 7-segment pattern converter. The IC7447 takes the Binary Coded Decimal (BCD) as the input and outputs the relevant 7 segment code.

Part 3 - To set up a circuit of Decimal-to-BCD Encoder using IC 74147 and to design and set up a circuit of Hexadecimal-to-Binary Encoder

Encoder

An encoder performs a function that is the opposite of decoder. It receives one or more signals in an encoded format and output a code that can be processed by another logic circuit. One of the advantages of encoding data, or more often data addresses in computers, is that it reduces the number of required bits to represent data or addresses. For example, if a memory has 16 different locations, in order to access these 16 different locations, 16 lines (bits) are required if the addressing signals are in 1 out of *n* format. However, if we code the 16 different addresses into a binary format, then only 4 lines (bits) are required. Such a reduction improves the speed of information processing in digital systems.

In Lab:

Lab Tasks-Part-1

2:4 DECODER (MIN TERM GENERATOR):

Truth Table:

INPUT		OUTPUT			
A	B	Y0	Y1	Y2	Y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Table: 9.1

Boolean Expressions:

$$Y0 = \overline{A}\overline{B} \qquad Y1 = \overline{A}B$$

$$Y2 = A\overline{B} \qquad Y3 = AB$$

Circuit Diagram:

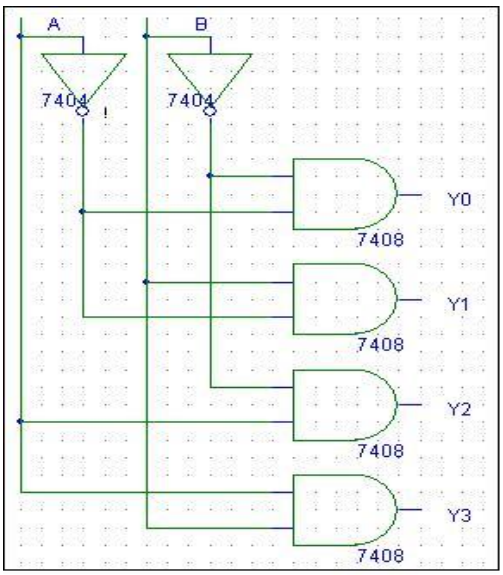


Fig: 9.1

2:4 DECODER (MAX TERM GENERATOR):

TRUTH TABLE:

INPUT		OUTPUT			
A	B	Y0	Y1	Y2	Y3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Table: 9.2

CIRCUIT DIAGRAM:

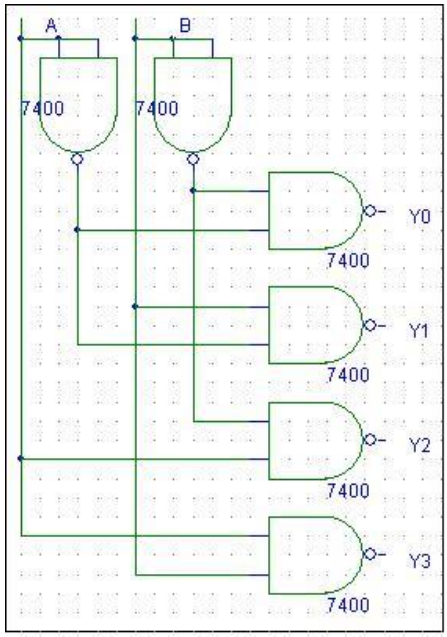


Fig: 9.2

Lab Tasks-Part-2:

CIRCUIT DIAGRAM:

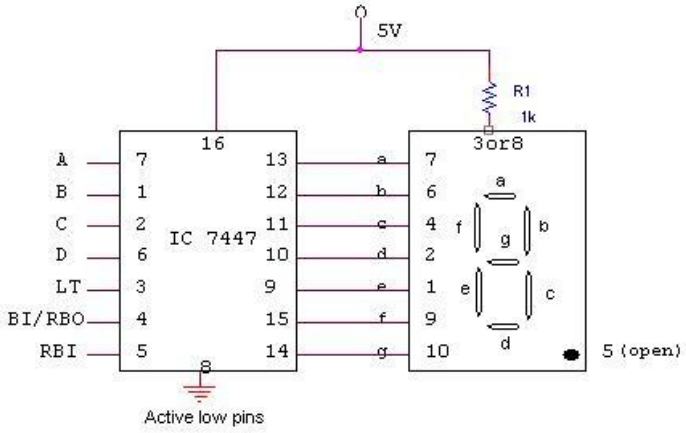


Fig: 9.3

TRUTH TABLE:

BCD Inputs				Output Logic Levels from IC 7447 to 7-segments							Decimal number display
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	
0	0	0	1	1	0	0	1	1	1	1	
0	0	1	0	0	0	1	0	0	1	0	
0	0	1	1	0	0	0	0	1	1	0	
0	1	0	0	1	0	0	1	1	0	0	
0	1	0	1	0	1	0	0	1	0	0	
0	1	1	0	1	1	0	0	0	0	0	
0	1	1	1	0	0	0	1	1	1	1	
1	0	0	0	0	0	0	0	0	0	0	
1	0	0	1	0	0	0	1	1	0	0	

Table: 9.3

Lab Tasks-Part-3:

DECIMAL-TO BCD ENCODER USING IC 74147.

TRUTH TABLE

INPUTS									OUTPUTS			
I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	A_3	A_2	A_1	A_0
1	1	1	1	1	1	1	1	0				
X	X	X	X	X	X	X	0	1				
X	X	X	X	X	X	0	1	1				
X	X	X	X	X	0	1	1	1				
X	X	X	X	0	1	1	1	1				
X	X	X	0	1	1	1	1	1				
X	X	0	1	1	1	1	1	1				
X	0	1	1	1	1	1	1	1				
0	1	1	1	1	1	1	1	1				
1	1	1	1	1	1	1	1	1				

Table: 9.4

CIRCUIT DIAGRAM:

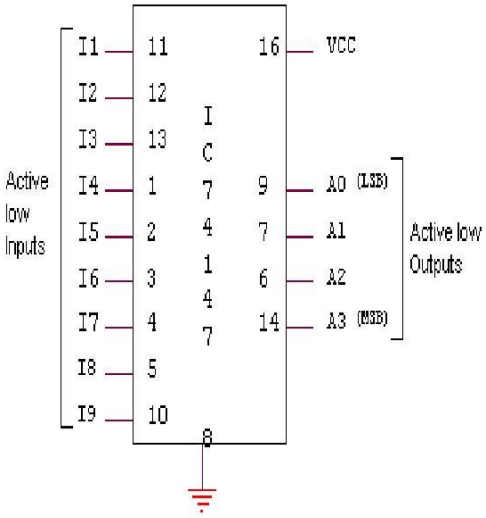


Fig: 9.4

Octal to Binary Converter:

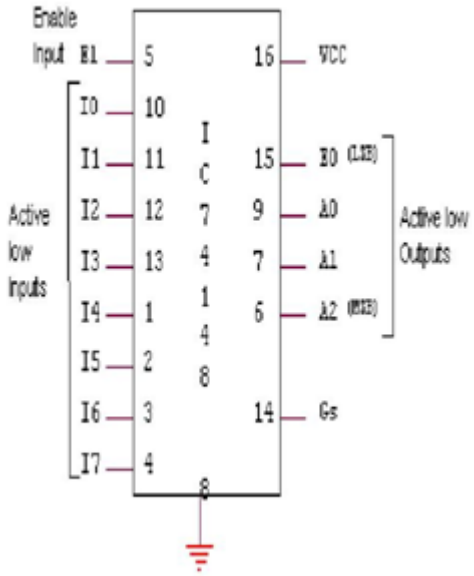


Fig: 9.5

Truth Table:

Inputs									Outputs				
E_1	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A_2	A_1	A_0	G_S	E_0
1	X	X	X	X	X	X	X	X					
0	1	1	1	1	1	1	1	1					
0	X	X	X	X	X	X	X	0					
0	X	X	X	X	X	X	0	1					
0	X	X	X	X	X	0	1	1					
0	X	X	X	X	0	1	1	1					
0	X	X	X	0	1	1	1	1					
0	X	X	0	1	1	1	1	1					
0	X	0	1	1	1	1	1	1					
0	0	1	1	1	1	1	1	1					

Table: 9.5

HEXADECIMAL TO BINARY ENCODER

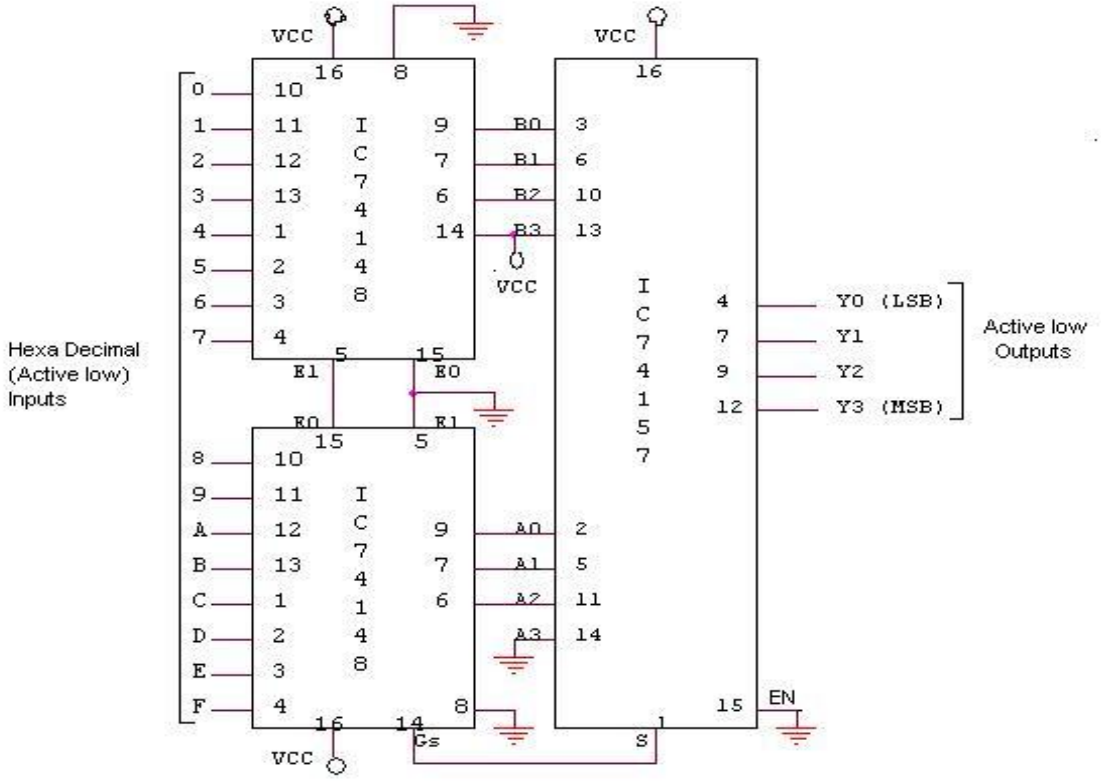


Fig: 9.6

TRUTH TABLE

INPUTS																OUTPUTS			
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	Y_3	Y_2	Y_1	Y_0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1				
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1				
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1				
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1				
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1				
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1				
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1				
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1				
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1				
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1				
1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1				
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1				
1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Table: 9.6

Post Lab:

Design the encoders and decoders used in Lab task-1 and Lab task-3 using NAND and NOR Gates.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 10: To Identify the Response of Sequential Circuit(s) using Hardware and Software Platforms

Objectives

- To describe the basic functions of sequential circuits
- To differentiate between various types of Flip Flops
- To identify various ICs used to perform basic functionality of various Flip Flops

Pre Lab

Truth Table verification of FF and conversion of one FF to another

Logic circuits that incorporate memory cells are called *sequential logic circuits*; their output depends not only upon the present value of the input but also upon the previous values. Sequential logic circuits often require a timing generator (a clock) for their operation. The latch (flip-flop) is a basic bi-stable memory element widely used in sequential logic circuits. Usually there are two outputs, Q and its complementary value. Some of the most widely used latches are listed below.

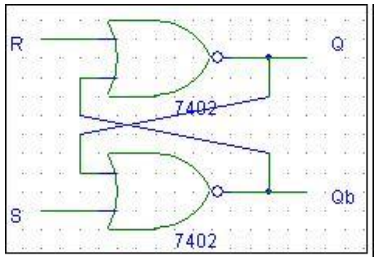
SR LATCH

S-R latch consists of two cross-coupled NOR gates. An S-R flip-flop can also be design using cross-coupled NAND gates as shown. The truth tables of the circuits are shown below. A clocked S-R flip-flop has an additional clock input so that the S and R inputs are active only when the clock is high. When the clock goes low, the state of flip-flop is latched and cannot change until the clock goes high again. Therefore, the clocked S-R flip-flop is also called "enabled" S-R flip-flop. A D latch combines the S and R inputs of an S-R latch into one input by adding an inverter. When the clock is high, the output follows the D input, and when the clock goes low, the state is latched.

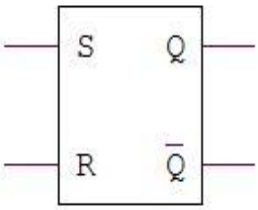
In Lab

Lab Task 1:

S-R LATCH:



LOGIC DIAGRAM



SYMBOL

Fig: 10.1

- **TRUTH TABLE**

Inputs		Observed Output	
S	R	Q+	$\overline{Q}$ b+
0	0		
0	1		
1	0		
1	1		

Table: 10.1

Lab Task 2:
SR FLIP FLOP:

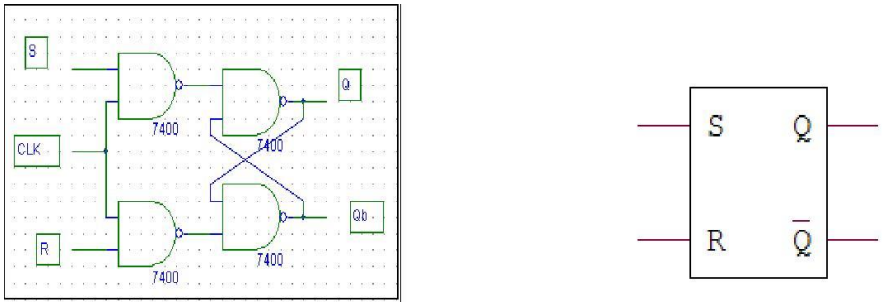


Fig: 10.2

- **TRUTH TABLE**

Inputs		Observed Output	
S	R	Q+	Q b+
0	0		
0	1		
1	0		
1	1		

Table: 10.2

Lab Task 3:
CONVERSION OF SR-FLIP FLOP TO T-FLIP FLOP (Toggle)

LOGIC DIAGRAM **SYMBOL**

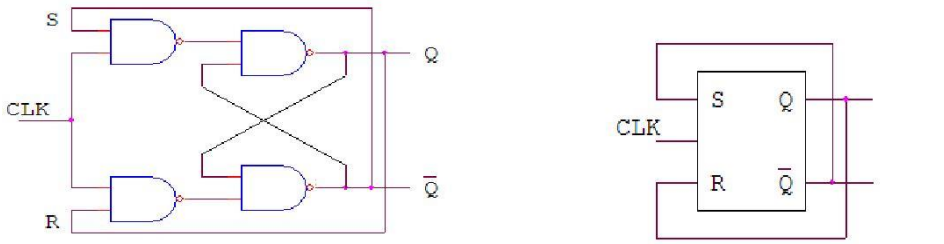
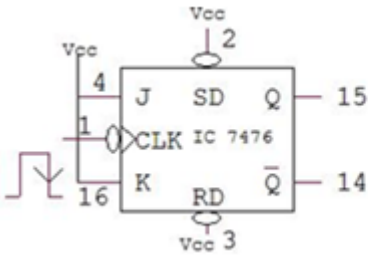


Fig: 10.3

T FLIP FLOP USING IC 7476



TRUTH TABLE

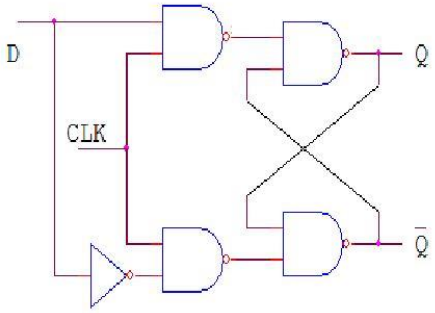
T	Observed Qn + 1
0	
1	

Table: 10.3

Lab Task 4:

CONVERSION OF SR-FLIP FLOP TO D-FLIP FLOP

LOGIC DIAGRAM



SYMBOL

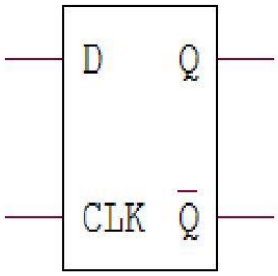
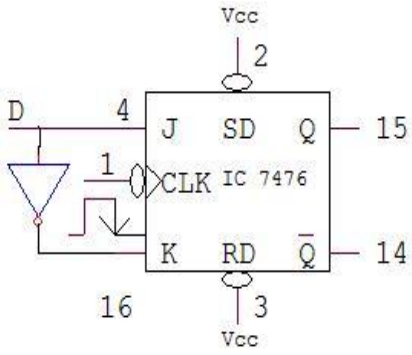


Fig:10.4

D FLIP FLOP USING IC 7476



TRUTH TABLE

CLOCK	D	Q+	Q'
0	X		
1	0		
1	1		

Table: 10.4

Lab Task 5:

CONVERSION OF SR-FLIP FLOP TO JK-FLIP FLOP

Logic Diagram

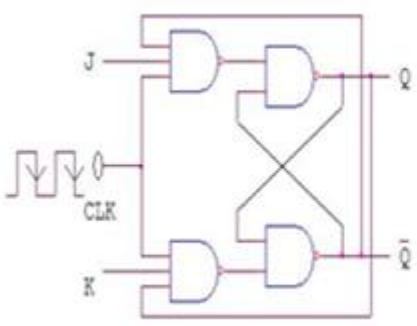


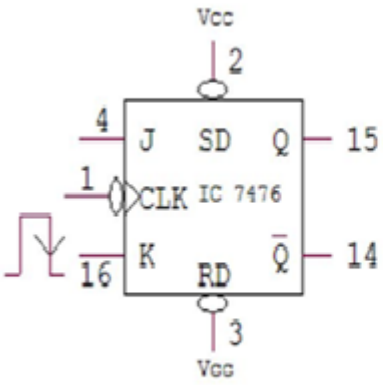
Fig: 10.5

Truth Table

Clock	J	K	Q+	Q'+	Comment
1	0	0	Q	Q'	
1	0	1	0	1	
1	1	0	1	0	
1	1	1	Q'	Q	

Table: 10.5

Logic Diagram



Truth Table

SD	RD	Clock	J	K	Q	Q'	Comment
0	0	Not Allowed					
0	1	X	X	X	1	0	
1	0	X	X	X	0	1	
1	1	1	0	0	NC	NC	
1	1	1	0	1	0	1	
1	1	1	1	0	1	0	
1	1	1	1	1	Q'	Q	

Table: 10.6

Lab Task 6:

JK Master Slave FLIP FLOP

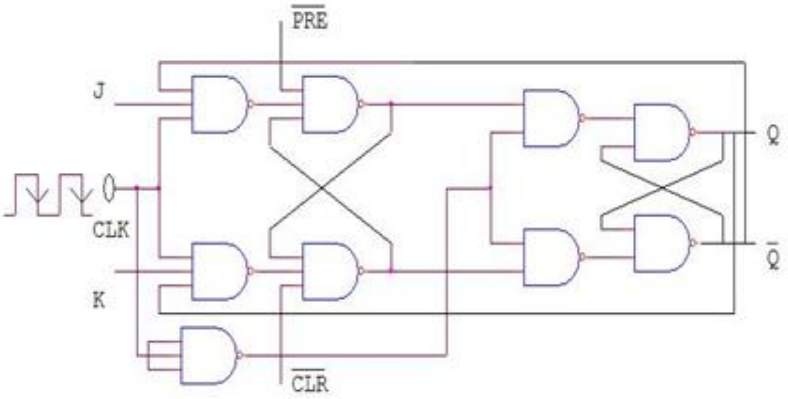


Fig: 10.6

Truth Table

$\overline{PRE} = \overline{CLR} = 1$

Clock	J	K	Q+	Q'+	Comment
1	0	0	Q	Q'	
1	0	1	0	1	
1	1	0	1	0	
1	1	1	Race Around		

Table: 10.7

Post Lab:

Differentiate between different types of FLIP FLOPS.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 11: To show the Response of Shift Registers using 7495 IC and Reproduce the shift registers using D-Flip-Flops

Objectives

1. To explain the working of a shift register using 7495 IC
2. To differentiate between the different types of shift register

Pre Lab

In digital circuits, a shift register is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the "data" input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the "bit array" stored in it, shifting in the data present at its input and shifting out the last bit in the array, at each transition of the clock input.

More generally, a shift register may be multidimensional, such that it's "data in" and stage outputs are themselves bit arrays: this is implemented simply by running several shift registers of the same bit-length in parallel.

Shift registers can have both parallel and serial inputs and outputs. These are often configured as 'serial-in, parallel-out' (SIPO) or as 'parallel-in, serial-out' (PISO). There are also types that have both serial and parallel input and types with serial and parallel output. There are also 'bidirectional' shift registers which allow shifting in both directions: L→R or R→L. The serial input and last output of a shift register can also be connected to create a 'circular shift register'.

In Lab

1. Check all the components for their working
2. Insert the appropriate IC into the IC base
3. Make connections as shown in the circuit diagram
4. Verify the Truth Table and observe the outputs

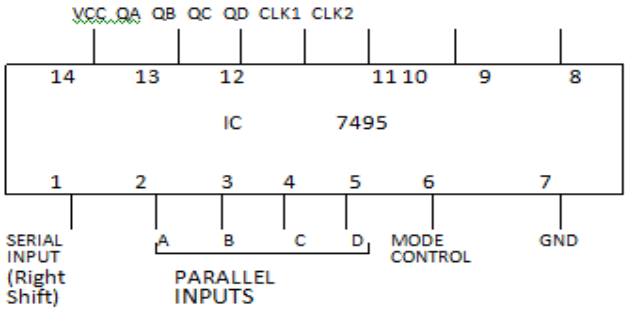


Fig: 11.1

Lab Task 1:
SERIAL IN SERIAL OUT (SISO) (Right Shift)

Serial i/p Data	Shift Pulses	Q_A	Q_B	Q_C	Q_D
-	-				
0	t1				
1	t2				
0	t3				
1	t4				
X	t5				
X	t6				
X	t7				
X	t8				

Table: 11.1

Lab Task 2:
SERIAL IN PARALLEL OUT (SIPO)

Serial i/p data	Shift Pulses	Q_A	Q_B	Q_C	Q_D
-	-				
0	t1				
1	t2				
0	t3				
1	t4				

Table: 11.2

Lab Task 3:
PARALLEL IN PARALLEL OUT (PIPO)

Clock Input Terminal	Shift Pulses	Q_A	Q_B	Q_C	Q_D
-	-				
CLK_2	t1				

Table: 11.3

Lab Task 4
PARALLEL IN SERIAL OUT (PISO)

Clock Input Terminal	Shift Pulses	Q_A	Q_B	Q_C	Q_D
-	-				
CLK_2	t1				
CLK_2	t2				
0	t3				
1	t4				
X	t5				

Table: 11.4

Post Lab:

1. Design shift registers using basic gates.
2. Differentiate between different types of shift registers.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 12: To Reproduce a Synchronous Sequence Detector using hardware and software tools

Objectives

To analyze working principle of a sequence detector

Pre Lab

A sequence detector is a sequential logic circuit that can be used to detect whether a given sequence of bits has been received or not by a receiver. Consider, for example, a single-input single-output sequence detector. Let the sequence to be detected be 101. Repetition is allowed in this sequence. This means that, the circuit will give an output of I when it detects first the sequence 101 in a series of incoming bits. Let the next two bits after the first 101 be 01. We find that the circuit will read this as 101 and output another 1. This is because we have given permission for repetition, and the circuit will consider the last 1 in the first 101 as a valid first 1 in the next sequence of 101.

In Lab

Procedure

1. Draw the state diagram of the state machine below and show it to the lab instructor.
2. Fill the state table.
3. Assign State numbers
4. Find simplified Expressions (*State Equations*) for the flip-flops
5. Draw the circuit diagram using **NAND GATES ONLY** for the state machine

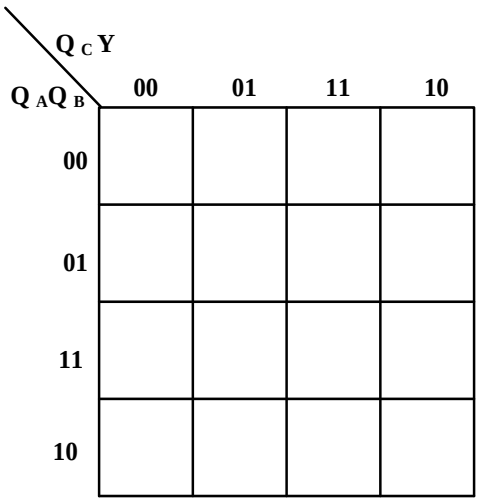
State Diagram

State Table

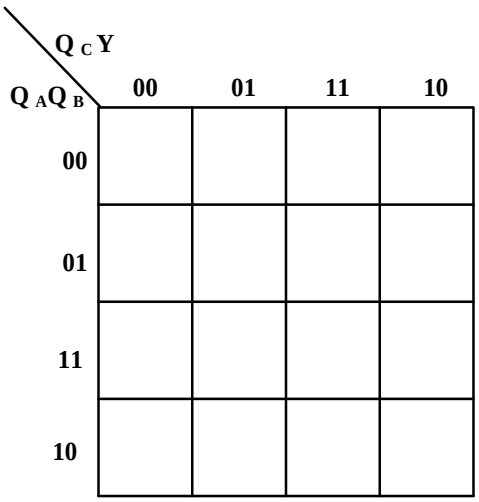
Present State				Next State			
Q_A	Q_B	Q_C	Y	D_A	D_B	D_C	
0	0	0	0				0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7
1	0	0	0				8
1	0	0	1				9
1	0	1	0				10
1	0	1	1				11
1	1	0	0				12
1	1	0	1				13
1	1	1	0				14
1	1	1	1				15

Table: 12.1

For _____



For _____



For _____

$Q_C Y$ / $Q_A Q_B$	00	01	11	10
00				
01				
11				
10				

Table: 12.2

State Equations

$D_A =$ $D_B =$

$D_C =$

Circuit Diagram

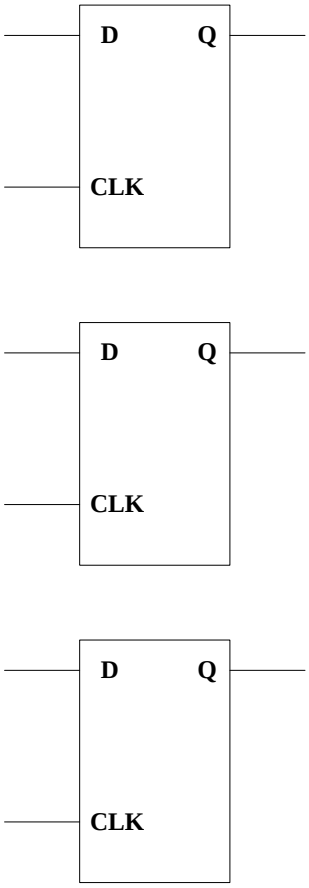


Fig: 12.1

Implementation

Connect the circuit according to your design and check for the following input sequence.

Input	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**
State																			
Output																			
Input (Cont')	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**
State																			
Output																			

Table: 12.3

Post Lab:

Design a sequence detector which to detect 110101.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 13: To Show the Response of Ring and Johnson Counter using 7495 IC

Objectives

- To explain the working of ring and Johnson counter using flip flops
- To display the results of ring and Johnson counter using hardware and software platforms

Pre Lab

Part 1 -Familiarize yourself with Ring Counter:

Ring counter is a basic register with direct feedback such that the contents of the register simply circulate around the register when the clock is running. Here the last output that is QD in a shift register is connected back to the serial input.

Part 2 -Familiarize yourself with Johnson Counter:

A basic ring counter can be slightly modified to produce another type of shift register counter called Johnson counter. Here complement of last output is connected back to the not gate input and not gate output is connected back to serial input. A four bit Johnson counter gives 8 state output.

In Lab

Lab Task-1

Circuit Diagram:

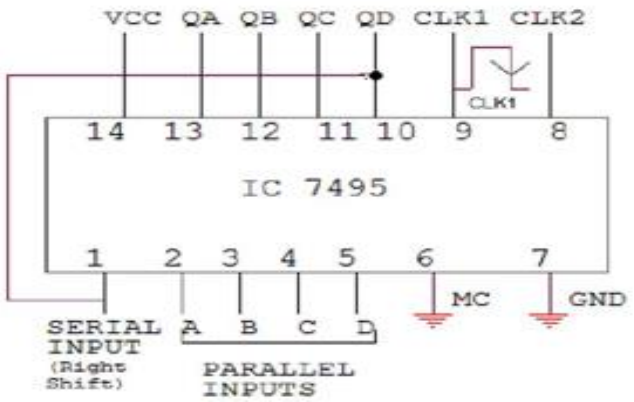


Fig: 13.1

Truth Table:

Clock pulses	Q_A	Q_B	Q_C	Q_D
0				
1				
2				
3				
4				
5				
6				
7				
8				

Table: 13.1

Lab Task-2:
Circuit Diagram :

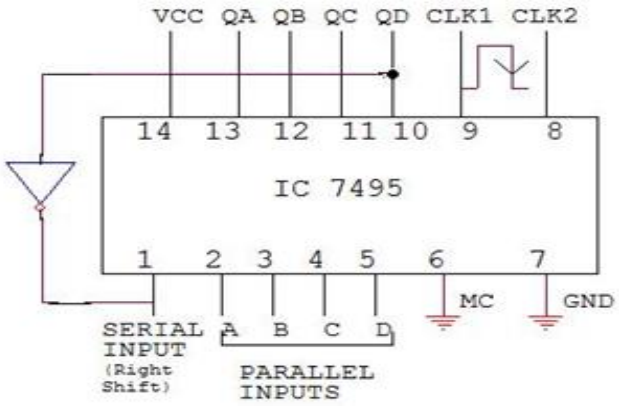


Fig: 13.1

Truth Table :

Clock Pulses	Q_A	Q_B	Q_C	Q_D
0				
1				
2				
3				
4				
5				
6				
7				
8				

Table: 13.1

Post Lab:

1. Differentiate between Ring couner and Johnson Counter.
2. Design ring counter using basic gates

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____

LAB # 14: To Reproduce the Asynchronous Counters using hardware and software tools

Objectives

- To analyze the working principle of a Synchronous counter using flip flops
- To analyze the working principle of asynchronous counter using flip flops
- To design synchronous and asynchronous counter using flip flops

Pre Lab

Part 1 -Familiarize yourself with Asynchronous Counter:

A counter in which each flip-flop is triggered by the output goes to previous flip-flop. As all the flip-flops do not change state simultaneously spike occur at the output. To avoid this, strobe pulse is required. Because of the propagation delay the operating speed of asynchronous counter is low. Asynchronous counter are easy and simple to construct.

Part 2 -Familiarize yourself with Synchronous Counter:

A counter in which each flip-flop is triggered by the output goes to previous flip-flop. As all the flip-flops do not change states simultaneously in asynchronous counter, spike occur at the output. To avoid this, strobe pulse is required. Because of the propagation delay the operating speed of asynchronous counter is low. This problem can be solved by triggering all the flip-flops in synchronous with the clock signal and such counters are called synchronous counters.

In Lab:

Mod 8 up counter

Circuit Diagram

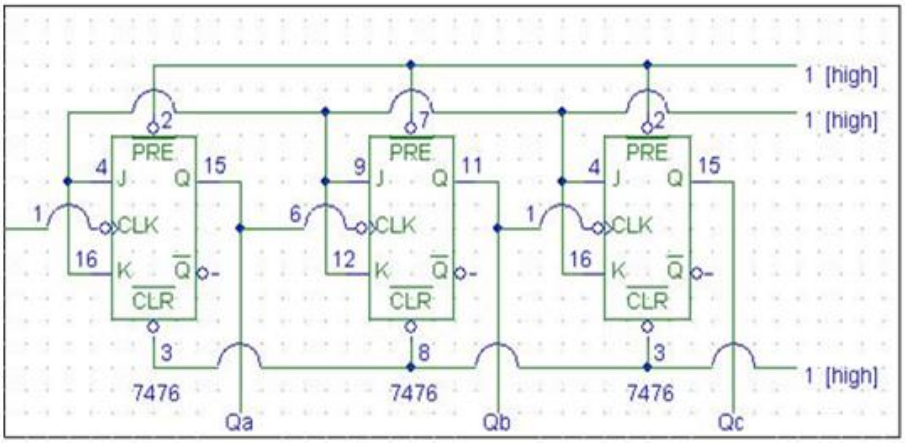


Fig: 14.1

TRUTH TABLE

CLK	Q_C	Q_B	Q_A
0			
1			
2			
3			
4			
5			
6			
7			
8			

Table: 14.1

Mod_ 6 up counter

Circuit Diagram:

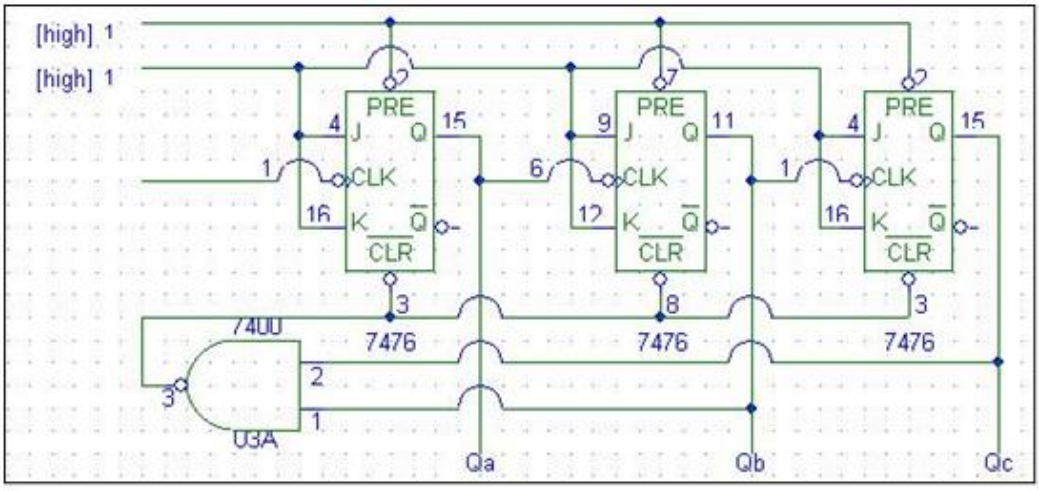


Fig: 14.2

TRUTH TABLE

CLK	Q_C	Q_B	Q_A
0			
1			
2			
3			
4			
5			
6			

Table: 14.2

Mod_ 8 down counter

Circuit Diagram:

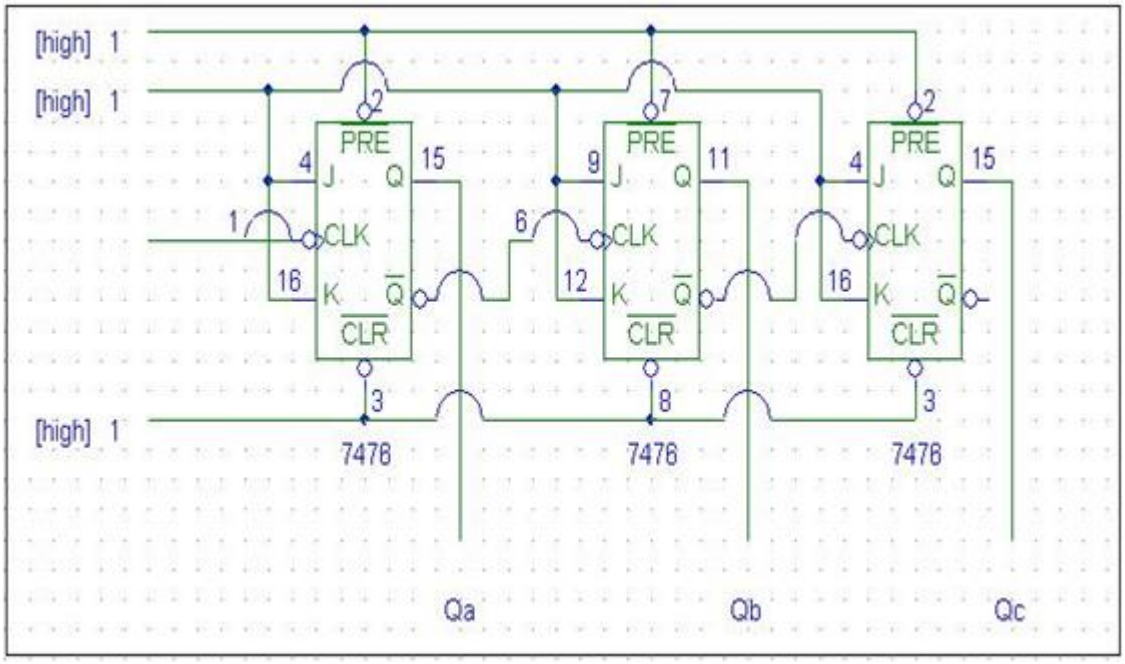


Fig: 14.3

TRUTH TABLE

CLK	Q_C	Q_B	Q_A
0			
1			
2			
3			
4			
5			
6			
7			
8			

Table: 14.3

Mod_ 6 down counter

Circuit Diagram:

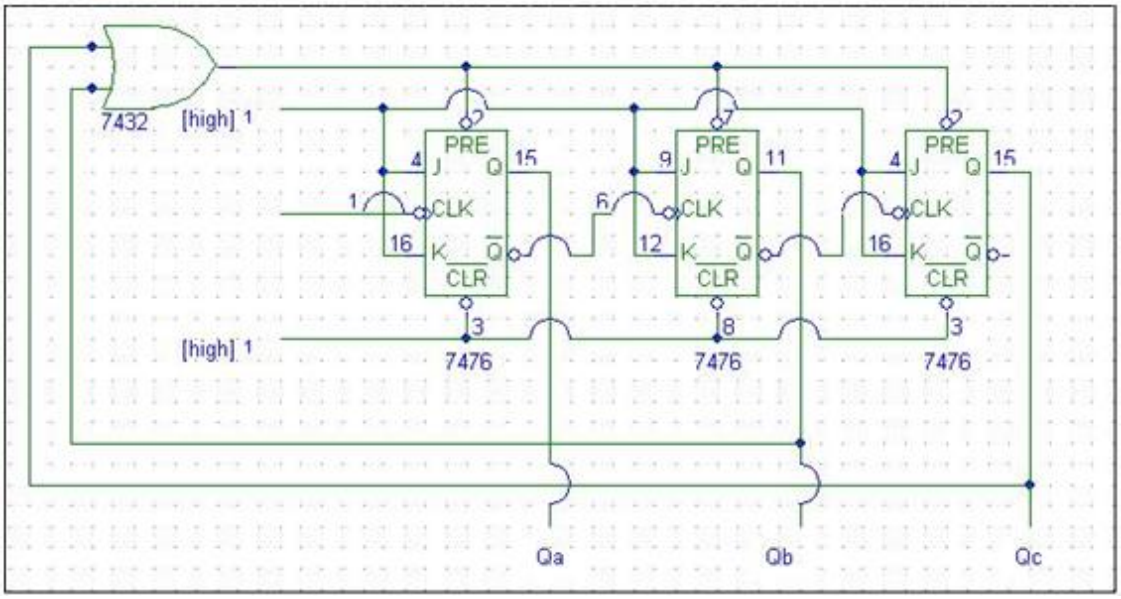


Fig: 14.4

TRUTH TABLE

CLK	Q_C	Q_B	Q_A
0			
1			
2			
3			
4			
5			
6			

Table: 14.4

PROCEDURE:

1) Check all the components for their working.
2) Insert the appropriate IC into the IC base.
3) Make connections as shown in the circuit diagram.
4) Verify the Truth Table and observe the outputs.

Mod_ 5 counter

Truth Table :

Q_C	Q_B	Q_A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
0	0	0

Present count

Q_C	Q_B	Q_A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

Next Count

Q_C	Q_B	Q_A
0	0	1
0	1	0
0	1	1
1	0	0
0	0	0

Table: 14.5

JK FF excitation table:

Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table: 14.6

Design:

1	X	X	1
0	X	X	X

$J_A = \overline{Q_C}$

X	1	X	X
X	X	X	X

$K_A = 1$

0	1	X	X
0	X	X	X

$J_B = Q_A$

X	X	1	0
X	X	X	X

$K_B = Q_A$

0	0	1	0
X	X	X	X

$J_C = Q_B Q_A$

X	X	X	X
1	X	X	X

$K_C = 1$

Table: 14.7

Circuit Diagram:

Mod_ 8 counter:

Truth Table :

Q_C	Q_B	Q_A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

Present count

Q_C	Q_B	Q_A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Next Count

Q_C	Q_B	Q_A
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

Table: 14.8

JK FF excitation table:

Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table: 14.8

Design:

1	X	X	1
1	X	X	1

$J_A = 1$

X	1	1	X
X	1	1	X

$K_A = 1$

0	1	X	X
X	1	X	X

$J_B = Q_A$

X	X	1	0
X	X	1	0

$K_B = Q_A$

0	0	1	0
X	X	X	X

$J_C = Q_B Q_A$

X	X	X	X
0	0	1	0

$K_C = Q_B Q_A$

Table: 14.9

Circuit Diagram:

Post Lab:

Differentiate between synchronous counter and asynchronous Counter.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date: _____