JSS MAHAVIDYAPEETHA

LABORATORY MANUAL

Subject Name: Computer Organization Lab
Subject Code: KCS 352

COURSE : B.Tech SEMESTER : III SEM
Name

Roll No.

Section-Batch

Department of Computer Science and Engineering
JSS ACADEMY OF TECHNICAL EDUCATION
C-20/1, SECTOR-62, NOIDA

VISION OF THE INSTITUTE
JSS Academy of Technical Education Noida aims to become an Institution of excellence in imparting quality Outcome Based Education that empowers the young generation with Knowledge, Skills,Research, Aptitude and Ethical values to solve Contemporary Challenging Problems.

MISSION OF THE INSTITUTE
• Develop a platform for achieving globally acceptable level of intellectual acumen and technological competence.
• Create an inspiring ambience that raises the motivation level for conducting quality research.

• Provide an environment for acquiring ethical values and positive attitude.

VISION OF THE DEPARTMENT

To spark the imagination of the Computer Science Engineers with values, skills and creativity to solve the real world problems.

MISSION OF THE DEPARTMENT

• To inculcate creative thinking and problem solving skills through effective teaching, learning and research.

• To empower professionals with core competency in the field of Computer Science and Engineering.

• To foster independent and life long learning with ethical and social responsibilities.

PROGRAM EDUCATIONAL OUTCOMES (PEOs)

PEO1: To empower students with effective computational and problem solving skills.

PEO2: To enable students with core skills for employment and entrepreneurship.

PEO3: To imbibe students with ethical values and leadership qualities.

PEO4: To foster students with research oriented ability which helps them in analyzing and solving real life problems and motivate them for pursuing higher studies.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the

consequent responsibilities relevant to the professional engineering practice.
PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to

comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)
PSO1: An ability to apply foundation of Computer Science and Engineering, algorithmic
principles and theory in designing and modeling computation based systems.
PSO2: The ability to demonstrate software development skills.

COURSE OUTCOMES (COs)
C 228.1 Implementation of various combinational circuits using IC's
C 228.2 Implementation of various code convertors, multiplexers, encoders and decoder
using IC's
C 228.3 Designing and verification of Flip Flops
C 228.4 Designing of I/O using Registers, ALU and Control Unit and demonstrating the usage of Register Transfer Language(RTL)
C 228.5 Implementation of Matrix Multiplication on multi -processor system and study of

scalability of multiprocessor systems.

CO-PO MAPPING

PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12
C 228.1
3 3 3 1 0 1 1 0 2 - 2 2
C 228.2
3 3 3 1 3 1 1 0 2 - 1 2
C 228.3
3 3 3 1 3 1 1 0 2 - 1 1
C 228.4
3 3 3 1 0 1 1 1 2 - 1 1
C 228.5
3 3 3 1 0 1 1 1 2 - 1 2

CO-PSO MAPPING

PSO1 PSO2
C 228.1
1
-
C 228.2
1
-
C 228.3
1
-
C 228.4
1
-
C 228.5
1
-

LIST OF EXPERIMENTS
Sr.
No.
Title of experiment Corresponding
CO

1.
Verification of Logic Gates C 228.1
2.
Implementing HALF ADDER, FULL ADDER using basic logic gates.
C 228.1
3.
Implementing Binary -to -Gray, Gray -to -Binary code conversions.
C 228.2
4.
Implementing 3-8 line DECODER C 228.2
5.
Implementing 4x1 and 8x1 MULTIPLEXERS. C228.2
6.
Verify the excitation tables of various FLIP-FLOPS. C 228.3
7.
Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
C 228.4
8.
Design of an 8- bit ARITHEMATIC LOGIC UNIT. C 228.4
9.
Write an algorithm and program to perform matrix multiplication of two n * n matrices on the 2-D mesh SIMD model, Hypercube SIMD Model or multiprocessor system.

C228.5
10.
Study of Scalability for Single board Multi-board, multi-core, multiprocessor using Simulator.
C228.5
Content Beyond Syllabus
11.
Implementing HALF SUBTRACTOR and FULL SUBTRACTOR using basic logic gates
C 228.1
12.
Design and Implement SISO and SIPO Shift Registers C 228.4
13.
Design and Implement PISO and PIPO Shift Registers C 228.4

INTRODUCTION

Computer Organization Lab comprises of certain experiments that are performed in Logicly Simulator. This simulator provides an interactive environment for creating and conducting simulated experiments on computer organization and architecture. It supports gate level design to CPU design.

The main features of the simulators are as follows:

• Logic: The simulator supports 5 valued logic. So the simulator supports wired AND for bus based design. These 5 states along with their corresponding wire values are - True (T) (wire color: blue), False (F) (wire color: black), High impedence (Z) (wire color: green), Unknown (X) (wire color: maroon), Invalid (I) (wire color: orange)

• Graphical organization of the simulator: The simulator contains a pallete on the right hand side. This pallete contains all the components and tools . Tools are used to act up on the components. A toolbar on the top which contains several buttons. These buttons are - save/open, simulate (after creating a circuit, this button has to be pressed to simulate the circuit and to get output), plot graph (to plot input-output wave form), undo/redo, delete, zoom in/zoom out, increment/decrement LED (for digital LED which can also be used as input and display), start/stop clock pulse, to check the name or pin configuration of a component, changing connection types, checking the user identification. A canvas in the middle where the circuits will be designed. A toolbar on the left side which contains the following buttons - Set Port to set the number of input and output ports for a circuit, Set Label and set name to set the label contents and the name of different components, Load Memory to load the memory content to the inbuilt memory(4 bit address and 12 bit data) for performing the computer design experiment. Data can be load either from file or through form.

• Components: Components have been catagorized according to their functionality and put into different drawers in the pallete. The area under every drawer is scrallable, if you are unable to see all the components in a particular drawer just click on the area and scroll. Different drawers - Circuits- contains 8 and 16 terminal circuits and flow container which can hold other circuit components, Logic gates- contains all kinds of basic logic gates with 2 and 3 inputs, Display and inputs- contains all kinds of component needed to give input to the circuit along with free running clock and displaying outputs of the circuit, Adders- contains different types of adder circuits,

Sequential ckt- contains basic flipflops, registers for designing sequential circuits, Other Components- contains different kinds of components like decoders, multiplexers, arithmetic logic units(ALU), memory elements(RAM cell), cache memory(without any replacement policy)required to design combinational circuits, Control Unit- contains a controller whose state table (Moore m/c) can be loaded from the interface, Computer Design- contains a single instruction CPU and a Memory (4
bit address and 12 bit data, can be loaded by user).

PREFACE

This laboratory manual is designed to have students experience on how to implement
Computer Organization design using Logic Simulator. The idea of writing this lab manual is
to make the undergraduate students aware of the new Computer Organization tools and how
to use them properly & efficiently. This manual serves as a guide for learning and implementing the designs through the simulator. The manual contains procedures, and pre-
experiment questions to help students prepare for experiments.

This practical manual will be helpful for students of Computer Science & Engineering for
understanding the course from the point of view of applied aspects. Though all the efforts
have been made to make this manual error free, yet some errors might have crept in
inadvertently. Suggestions from the readers for the improvement of the manual are most
welcomed.

Dr. Rachna Jain, Assistant Professor, Dept. of CSE

DO'S AND DONT'S

DO's

1. Conform to the academic discipline of the department.
2. Enter your credentials in the laboratory attendance register.
3. Read and understand how to carry out an activity thoroughly before coming to the laboratory.
4. Ensure the uniqueness with respect to the methodology adopted for carrying out the experiments.
5. Shut down the machine once you are done using it.

DONT'S

1. Eatables are not allowed in the laboratory.
2. Usage of mobile phones is strictly prohibited.
3. Do not open the system unit casing.
4. Do not remove anything from the computer laboratory without permission.
5. Do not touch, connect or disconnect any plug or cable without your faculty/laboratory technician's permission.

GENERAL SAFETY INSTRUCTIONS

1. Know the location of the fire extinguisher and the first aid box and how to use them in case of an emergency.

2. Report fires or accidents to your faculty /laboratory technician immediately.

3. Report any broken plugs or exposed electrical wires to your faculty/laboratory technician immediately.

4. Do not plug in external devices without scanning them for computer viruses.

DETAILS OF THE EXPERIMENTS CONDUCTED
(TO BE USED BY THE STUDENTS IN THEIR RECORDS)

S.No	DATE OF CONDUCTION	EXPT. No	TITLE OF THE EXPERIMENT	PAGE No.	MARKS AWARDED (20)	FACULTY

SIGNATURE
WITH REMARK
1
2
3
4
5
6
7
8
9
10

GUIDELINES FOR LABORTORY RECORD PREPARATION

While preparing the lab records, the student is required to adhere to the following guidelines:

Contents to be included in Lab Records:

1. Cover page
2. Vision
3. Mission
4. PEOs
5. POs
6. PSOs
7. COs
8. CO-PO-PSO mapping
9. Index
10. Experiments

'¢ Aim
'¢ Equipments and Components Required
'¢ Theory
'¢ Procedure
'¢ Result and Conclusion

A separate copy needs to be maintained for pre-lab written work

The student is required to make the Lab File as per the format given on the next two pages.

JSS ACADEMY OF TECHNICAL EDUCATION

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMPUTER ORGANIZATION LAB FILE (RCS 352)

Name

Roll No.

Section- Batch

JSS Academy of Technical Education – NOIDA
Department of Computer Science & Engineering

INDEX

Experiment No.	Experiment Name	Date of Conduction	Date of Submission	Faculty Signature

GUIDELINES FOR ASSESSMENT

Students are provided with the details of the experiment (Aim, pre-experimental questions, procedure etc.) to be conducted in next lab and are expected to come prepared for each lab class.

Faculty ensures that students have completed the required pre-experiment questions and they complete the in-lab programming assignment(s) before the end of class. Given that the lab programs are meant to be formative in nature, students can ask faculty for help before and during the lab class.

Students' performance will be assessed in each lab based on the following Lab Assessment Components:

AC1: Written Work (Max. marks = 4)
AC2: Fundamental Knowledge to conduct Experiment (Max. marks = 4)
AC3: Experiment Completed Successfully (Max. marks = 4)
AC4: Questions Answered (Max. marks = 4)
AC5: Punctuality (Max. marks = 4)

In each lab class, students will be awarded marks out of 4 under each component head, making it total out of 20 marks.

EXPERIMENT 1

Aim: Verification of logic gates

Equipment Required & Component Required:

S.No.	Equipments	Specification	Quantity
1	Digital IC Trainer kit	-	1
2	Digital Multimeter		1

S.No.	Components	Specification	Quantity
1	Digital ICs	7400, 7402, 7404, 7408, 7432, 7486.	1 each
2	Patch cords	-	6

Theory:
• Details of IC used and pin configurations.
• Working of logic gates.

1. OR GATE:

PIN CONFIGURATION OF 74LS32

TRUTH TABLE

OBSERVATION TABLE

INPUT A INPUT B OUTPUT Y
0 0 0
0 1 1
1 0 1
1 1 1
INPUTS
OUTPUT USING
VOLTMETER
USING LED
A B
- - -
- - -
- - -
- - -
14 13 12 11 10 9 8
5 6 7 1 2 3 4
14 13 12 11 10 9 8
5 6 7 1 2 3 4
V
CC

GND

2. AND GATE:

PIN CONFIGURATION OF 74LS08

TRUTH TABLE OBSERVATION TABLE

3. NOT GATE:

GND

PIN CONFIGURATION OF 74LS04

TRUTH TABLE OBSERVATION TABLE

INPUTS
OUTPUT USING
VOLTMETER
USING
LED
A B
- - -

- - -

- - -

- - -

INPUT A INPUT B OUTPUT Y
0 0 0
0 1 0
1 0 0
1 1 1
INPUTS
OUTPUT USING
VOLTMETER
USING LED
A B
- - -
- - -
INPUT A
OUTPUT
Y
0 1
1 0
GND
14 13 12 11 10 9 8
 5 6 7 1 2 3 4
14 13 12 11 10 9 8
 5
 6
 7 1 2 3 4
V
CC

14 13 12 11 10 9 8
 5 6 7 1 2 3 4
14 13 12 11 10 9 8
 5

6
7 1 2 3 4
V
CC

14 13 12 11 10 9 8
5 6 7 1 2 3 4
V
CC

4. NOR GATE

TRUTH TABLE OBSERVATION TABLE

5. NAND GATE:

PIN CONFIGURATION OF 74LS00

TRUTH TABLE OBSERVATION TABLE

INPUTS
OUTPUT
USING
VOLTMETER
USING LED
A B

- - -

- - -

- - -

- - -
INPUT
A
INPUT
B
OUTPUT
Y
0 0 1
0 1 0
1 0 0
1 1 0
INPUT
A
INPUT
B
OUTPUT
Y
0 0 1
0 1 1
1 0 1
1 1 0
INPUTS
OUTPUT USING
VOLTMETER
USING LED
A B
- - -

- - -

- - -

- - -

6. EX-OR GATE:

PIN CONFIGURATION OF 7486

TRUTH TABLE OBSERVATION TABLE

Pre-Experiment Questions:-
Q.1 What are the different types of logic gates?
Q.2 Give the truth table of all the basic gates.
Q.3 Which gates are known as universal Gates? And why?
Q.4 Differentiate between NAND and AND Gate.

Procedure:
1. Identify the pin no's of the given IC.
2. From the IC No. Find out the type of gate.
3. Check for the proper working of the gate.
4. Connect the circuit as per circuit diagram.
5. For all combination of input condition. Tabulate the output voltage by connecting a
voltmeter at the output end.
6. Verify it with truth Table.
7. Repeat the above procedure for all gates.

Result & Conclusion: All Logic Gates are verified.
INPUTS
OUTPUT
USING
VOLTMETER
USING LED
A B
- - -
- - -
- - -
- - -
INPUT
A
INPUT

B
OUTPUT
Y
0 0 0
0 1 1
1 0 1
1 1 0

Post Experiment Questions: -
Q.1 How do you implement the gates using diodes?
Q.2 Implement the basic gates using universal gates.
Q.3 What do you understand the word IC?
Q.4 What is a chip?

EXPERIMENT 2

Aim: Design and implementation of HALF ADDER, FULL ADDER using basic logic gates

Equipment & Components Required:

S.No.	Equipments	Specification	Quantity
1	Digital IC Trainer kit	-	1
2	Digital Multimeter		1

S.No.	Components	Specification	Quantity
1	Digital ICs	7400, 7402, 7404, 7408, 7432, 7486.	1 each

2

Patch cords
- 6

Theory:

a) To design and implement half adder using logic gates

HALF ADDER

CIRCUIT DIAGRAM TRUTH TABLE

INPUT A INPUT B
OUTPUTS
S C
0 0 0 0
0 1 1 0
1 0 1 0
1 1 0 1

b) To design and implement full adder using logic gates

FULL ADDER

CIRCUIT DIAGRAM

TRUTH TABLE

Pre-Experiment Questions:
Q.1 Explain the truth table of half adder.
Q.2 How many Ex-or and or or gate can be used to make a half adder?
Q.3 How do we convert half adder to full adder?

Q.4 Explain the characteristics of half adder.

Procedure:
• Identify the pins.
• Connect the circuit as per circuit diagram.
• Obtain outputs with various input combinations.
• Verify it with the Boolean function using truth table

Result & Conclusion: All logical circuits have been implemented & verified through truth table.

Post-Experiment Question:
Q.1 What are the applications of half adder?
Q.2 What are the applications of full adder?
Q.3 Explain the advantages of half adder?
Q.4 Ellaborate the advantages of full adder over half adder.

EXPERIMENT 3

Aim: Design and implementation of Binary to Gray, Gray to Binary Code conversions

Equipment & Components Required:

S.No. Equipments Specification Quantity
1 Digital IC Trainer kit - 1
2 Digital Multimeter 1

S.No. Components Specification Quantity

1

Digital ICs
7400, 7402,
7404,
7408, 7432,
7486.
1 each

2

Patch cords
- 6

Theory:

a) To design and implement Binary to Gray Code conversions

Pin diagram of Binary to gray code converter using 7486 Ic(Exor Gate)

Circuit Diagram of

Binary to Gray Code Converter

Truth Table

b) To design and implement Binary to Gray Code conversions

Pin diagram of Gray to Binary code converter using 7486 Ic(Exor Gate)

INPUTS OUTPUTS
A

B C D G
4
G
3
G
2
G
1

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

Circuit Diagram for Gray to Binary Code
Converter

T Truth Table

Pre-Experiment Questions:

Q.1 What is a code converter?

Q.2 Differentiate between translator and code converter.

Q.3 Explain the primary usage of grey code.

Q.4 Illustrate the reasons for using grey code.

Procedure:

• Collect the components necessary to accomplish this experiment.
• Plug the IC chip into the breadboard.
• Connect the supply voltage and ground lines to the chips. PIN7 = Ground and PIN14 = +5V.
• Make connections as shown in the respective circuit diagram.
• Connect the inputs of the gate to the input switches of the LED.
• Connect the output of the gate to the output LEDs.
• Once all connections have been done, turn on the power switch of the breadboard

INPUTS OUTPUTS

A

B C D B

3

B

2

B

1

B

0

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

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

• Operate the switches and fill in the truth table (Write "1" if LED is ON and "0" if L1 is OFF Apply the various combination of inputs according to the truth table and observe the condition of Output LEDs.

Result & Conclusion: Binary to gray and gray to binary code converter has been designed
using EXOR gate and its truth table verified.

Post-Experiment Question:
Q.1 What are the advantages of code converter?
Q.2 What are the properties of gray code?
Q.3 Describe a way by which we can convert BCD to binary using hardware approach.
Q.4 List the process to generate n bit gray codes.

EXPERIMENT 4

Aim: Design and implementation of 3-8 line DECODER
Equipment's & Components Required:

S.No.	Equipment's	Specification	Quantity
1	Digital IC Trainer kit	-	1
2	Digital Multimeter		1

S.No.	Components	Specification	Quantity
1	Digital ICs	7400, 7402, 7404, 7408, 7432, 7486.	1 each
2	Patch cords	-	6

Theory:
a) 4 to 2 encoder using logic gates:
Truth Table Logic Diagram:

I
3
I
2
I
1
I
0
O
1
O
0

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

b) 3 to 8 decoder using logic gates:

Symbol: Truth table:

Logic Diagram of 3 to 8 decoder

Pre-Experiment Questions:

Q.1 Difference between Encoder and Decoder.

Q.2 Design a 4 to 16 decoder using two 2 to 4 decoder.

Procedure:

- Collect the components necessary to accomplish this experiment.
- Plug the IC chip into the breadboard.
- Connect the supply voltage and ground lines to the chips. PIN7 = Ground
- and PIN14 = +5V.
- Make connections as shown in the respective circuit diagram.
- Connect the inputs of the gate to the input switches of the LED.
- Connect the output of the gate to the output LEDs.
- Once all connections have been done, turn on the power switch of the breadboard

- Operate the switches and fill in the truth table (Write "1" if LED is ON and "0" if L1 is OFF Apply the various combination of inputs according to the truth tabe and obseve the condition of Output LEDs.

Result & Conclusion: 3-8 line decoder has been implemented & verified through truth table.

Post Experiment Questions:

Q.1 Design a 5 to 32 decoder using one 2 to 4 and four 3 to 8 decoder IC’S.

Q.2 Write a note on BCD to decimal decoder.

EXPERIMENT 5

Aim: Design & implement a 4x1 MUX and 8x1 MULTIPLEXERS

Equipment's & Components Required:

S.No.	Equipment's	Specification	Quantity
1	Digital IC Trainer kit	-	1
2	Digital Multimeter		1

S.No.	Components	Specification	Quantity
1	Digital ICs	7400, 7402, 7404, 7408, 7432, 7486.	1 each
2	Patch cords	-	6

Theory:

a) 4 to 1 Multiplexer:

Symbol: Truth table:

Addressing Input		Selected
b	a	
0	0	A
0	1	B
1	0	C
1	1	D

Logic Diagram:

b) 8x1 Multiplexer

Pin diagram of 8:1 Mux using two 4:1 Mux

Circuit of 8:1 Mux using dual 4:1 Mux
Truth Table of 8:1 Mux Using Dual 4:1 Mux

Select Lines			Inputs								Output	
E a	S 0	S 1	I o	I 1	I 2	I 3	I 4	I 5	I 6	I 7	Z a	Z b
Y												
0	0	0	0	×	×	×	×	×	×	×	0	×
0	0	0	1	×	×	×	×	×	×	×	1	×
0	0	1	×	0	×	×	×	×	×	×	0	×

0 0 1 × 1 × × × × × × 1 × 1
0 1 0 × × 0 × × × × × 0 × 0
0 1 0 × × 1 × × × × × 1 × 1
0 1 1 × × × 0 × × × × 0 × 0
0 1 1 × × × 1 × × × × 1 × 1
1 0 0 × × × 0 × × × × 0 0
1 0 0 × × × × 1 × × × × 1 1
1 0 1 × × × × × 0 × × × 0 0
1 0 1 × × × × × 1 × × × 1 1
1 1 0 × × × × × × 0 × × 0 0
1 1 0 × × × × × × 1 × × 1 1
1 1 1 × × × × × × × 0 × 0 0
1 1 1 × × × × × × × 1 × 1 1

Pre -Experiment Questions:
Q.1 Explain the need of multiplexer.
Q.3 Which is the major functioning responsibility of the multiplexing combinational circuit?
Q.4 How many NOT gates are required for the construction of a 4-to-1 multiplexer?

Procedure:
• Collect the components necessary to accomplish this experiment.
• Plug the IC chip into the breadboard.
• Connect the supply voltage and ground lines to the chips. PIN7 = Ground
• and PIN14 = +5V.
• Make connections as shown in the respective circuit diagram.
• Connect the inputs of the gate to the input switches of the LED.
• Connect the output of the gate to the output LEDs.
• Once all connections have been done, turn on the power switch of the breadboard
• Operate the switches and fill in the truth table (Write "1" if LED is ON and "0" if L1 is OFF Apply the various combination of inputs according to the truth tabe and obseve the condition of Output LEDs.
Result & Conclusion: 4 to 1 & 8 to 1 multiplexer has been implemented & verified through
truth table.
Post Experiment Questions
Q.3 In 1-to-4 demultiplexer, how many select lines are required?
Q.4 Which IC is used for the implementation of 1-to-16 DEMUX?

EXPERIMENT 6

Aim: Verify the excitation table of various FLIP-FLOPS

Equipments & Components Required:

S.No.	Equipments	Specification	Quantity
1	Digital IC Trainer kit	-	1
2	Digital Multimeter		1

3. Components Required:

S.No.	Components	Specification	Quantity
1	Digital ICs	7400, 7402, 7404, 7408, 7432, 7486.	1 each
2	Patch cords	-	6

Theory:

Flip-flops are synchronous bistable devices. The term synchronous means the output changes state only when the clock input is triggered. That is, changes in the output occur in synchronization with the clock. A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the stored bit. Since memory elements in sequential circuits are usually flip-flops, it is worth summarizing the behavior of various flip-flop types before proceeding further. All flip -flops can be divided into four basic types: SR, JK, D and T. They differ in the number of inputs and in the response invoked by different value of input signals. The four types of flip -flops are defined in the Table below.

Circuit Diagram

Pre-Experiment Questions:
Q.1 Difference between Latch and Flip Flop.
Q.2 Differentiate between combinational and sequential circuits.
Q.3 The truth table for an S-R flip-flop has how many VALID entries?
Q.4 What is a trigger pulse?

Procedure:
• Collect the components necessary to accomplish this experiment.
• Plug the IC chip into the breadboard.
• Connect the supply voltage and ground lines to the chips. PIN7 = Ground
• and PIN14 = +5V.
• Make connections as shown in the respective circuit diagram.
• Connect the inputs of the gate to the input switches of the LED.
• Connect the output of the gate to the output LEDs.
• Once all connections have been done, turn on the power switch of the breadboard

• Operate the switches and fill in the truth table (Write "1" if LED is ON and "0" if L1 is OFF Apply the various combination of inputs according to the truth table and observe the condition of Output LEDs.

Result & Conclusion: Verified excitation table of various flip flops.

Post Experiment Questions:

Q.1 How is a JK Flip Flop made to toggle?
Q.2 How many stable states does a Flip Flop has?
Q.3 What is the significance of the J and K terminals on the J-K flip-flop?
Q.4 Determine the output frequency for a frequency division circuit that contains 12 flip-flops
with an input clock frequency of 20.48 MHz.

EXPERIMENT 7

Aim: Design and implement 8-bit Input/Output System with four 8-bit internal registers

Equipment's & Components Required:

S.No. Equipments Specification Quantity
1 Logic Simulator - 1

S.No. COMPONENT SPECIFICATION QTY.
1. D FLIP FLOP IC 7474 2
2. OR GATE IC 7432 1
3. IC TRAINER KIT - 1
4. PATCH CORDS - 15

Theory:
A register is capable of shifting its binary information in one or both directions is known as
shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with
output of one flip flop connected to input of next flip flop. All flip flops receive common
clock pulses which causes the shift in the output of the flip flop.The simplest possible shift
register is one that uses only flip flop. The output of a given flip flop is connected to the input
of next flip flop of the register. Each clock pulse shifts the content of register one bit position
to right.

LOGIC DIAGRAM:
8-bit Input/Output System with four 8-bit internal register

Pre-Experiment Questions:
Q.1 What are the functions of a bus?
Q.2 State the features of multiplexers.
Q.3 What is the differnence between register and counter?
Q.4 Explain serial shifting method.

Procedure:
- Connections are given as per circuit diagram.
- Logical inputs are given as per circuit diagram.
- Observe the output and verify the truth table.

Result & Conclusion: Verified 8-bit Input/Output System with four 8-bit internal registers on simulator.

Post-Experiment Questions:
Q.1 What are the advantages of using bus interface?
Q.2 Define an Internal register?
Q.3 How many types of registers are there?
Q.4 What is a binary register?

EXPERIMENT 8

Aim: Design of an 8- bit ARITHMETIC LOGIC UNIT.

Equipment's & Components Required:

S.No.	Equipments	Specification	Quantity
1	Logic Simulator	-	1

Theory:
ALU or Arithmetic Logical Unit is a digital circuit to do arithmetic operations like addition,
subtraction, division, multiplication and logical operations like and, or, xor, nand, nor etc. A
simple block diagram of a 4 bit ALU for operations and, or, xor and Add is shown in the

Logic diagram.

LOGIC DIAGRAM:
Block diagram of a 4 bit ALU

Design Issues :
The circuit functionality of a 1 bit ALU is shown here, depending upon the control signal S1
and S0 the circuit operates as follows:
for Control signal S1 = 0 , S0 = 0, the output is A And B,
for Control signal S1 = 0 , S0 = 1, the output is A Or B,
for Control signal S1 = 1 , S0 = 0, the output is A Xor B,
for Control signal S1 = 1 , S0 = 1, the output is A Add B.
The truth table for 16-bit ALU with capabilities similar to 74181 is shown here:

Required functionality of ALU (inputs and outputs are active high)
Mode Select F
n
for active HIGH operands
Inputs Logic Arithmetic (note 2)
S3 S2 S1 S0 (M = H) (M = L) (C
n
=L)
L L L L A' A
L L L H A'+B' A+B
L L H L A'B A+B'
L L H H Logic 0 minus 1
L H L L (AB)' A plus AB'
L H L H B' (A + B) plus AB'
L H H L A "• B A minus B minus 1
L H H H AB' AB minus 1
H L L L A'+B A plus AB
H L L H (A "• B)' A plus B
H L H L B (A + B') plus AB
H L H H AB AB minus 1
H H L L Logic 1 A plus A (Note 1)
H H L H A+B' (A + B) plus A
H H H L A+B (A + B') plus A
H H H H A A minus 1
L denotes the logic low and H denotes logic high.

Pre-Experiment Questions:
Q.1 What are the functions of a an ALU?
Q.2 How does an ALU work?
Q.3 Describe the components of ALU.
Q.4 What are the basic operations of I/O unit?

Procedure:
- Connections are given as per circuit diagram.
- Logical inputs are given as per circuit diagram.
- Observe the output and verify the truth table.

Result & Conclusion: Verified the design of an 8 bit ALU.

Post-Experiment Questions:
Q.1 What are the functions of a CPU?
Q.2 What are the components of CPU and how are they interconnected?
Q.3 What are the basic operations of memory unit?
Q.4 How many ALU's a computer can have?

EXPERIMENT 9

Aim: To write an algorithm and program to perform matrix multiplication of two n * n matrices on the 2-D mesh SIMD model, Hypercube SIMD Model or multiprocessor
system.

Equipment's & Components Required:

S.No. Equipment's Specification Quantity
1 Logic Simulator - 1

Theory:
Mesh Network: A set of nodes arranged in the form of a p dimensional lattice is called a mesh

network. In a mesh network only neighboring nodes can communicate with each other. Therefore, interior nodes can communicate with 2p other nodes.

ALGORITHM:
Algorithm
Procedure MATRIXMULT
begin
for k = 1 to n-1 step 1do
begin
for all Pi,j where i and j ranges from 1 to n do
if i is greater than k then
rotate a in the east direction
end if
if j is greater than k then
rotate b in the south direction
end if
end
for all Pi,j where i and j lies between 1 and n do
compute the product of a and b and store it in c
for k= 1 to n-1 step 1 do

for all Pi,j where i and j ranges from 1 to n do rotate a in the east rotate b in the south c=c+aXb end

HYPERCUBE ALGORITHM

Pre-Experiment Questions:
Q.1 Define Multicore clusters?
Q.2 What is a Mesh network?
Q.3 Explain Hypecube.
Q.4 What are the advantages of multiprocessor?

Procedure:
• Write an algorithm on Mesh networks

Result & Conclusion: Designed an algorithm and program to perform matrix multiplication of
two n * n matrices on the 2-D mesh SIMD model, Hypercube SIMD Model or multiprocessor
system

Post-Experiment Questions:
Q.1 What is an MPI interface?
Q.2 Define SIMD Parallel system.
Q.3 What is a parallel processing system?
Q.4 Explain the advantages of SIMD processors.

EXPERIMENT 10

Aim: To study the Scalability for Single board, Multi-board, multi-core, multiprocessor
using Simulator

Equipments & Components Required:

SL.No.	Equipments	Specification	Quantity
1	Logic Simulator	-	1

Theory:
A multi-core processor is a single computing component with two or more independent processing units called cores, which read and execute program instructions. The
instructions are ordinary CPU instructions (such as add, move data, and branch) but the single
processor can run multiple instructions on separate cores at the same time, increasing overall
speed for programs amenable to parallel computing. Manufacturers typically integrate the
cores onto a single integrated circuit die (known as a chip multiprocessor or CMP) or onto
multiple dies in a single chip package.

Diagram of a generic dual-core processor with CPU-local :-

A multi-core processor implements multiprocessing in a single physical package. Designers
may couple cores in a multi-core device tightly or loosely. For example, cores may or may
not share caches, and they may implement message passing or shared-memory inter-core

communication methods. Common network topologies to interconnect cores
include bus, ring, two-dimensional mesh, and crossbar. Homogeneous multi-core systems
include only identical cores; heterogeneous multi-core systems have cores that are not
identical (e.g. big.LITTLE have heterogeneous cores that share the same instruction set,
while AMD Accelerated Processing Units have cores that don't even share the same
instruction set). Just as with single-processor systems, cores in multi-core systems may
implement architectures such as VLIW, superscalar, vector, or multithreading.

Multiprocessor :-
A multiprocessor is a computer system with two or more central processing units (CPUs),
with each one sharing the common main memory as well as the peripherals. This helps in
simultaneous processing of programs.
The key objective of using a multiprocessor is to boost the system’s execution speed, with
other objectives being fault tolerance and application matching.
A good illustration of a multiprocessor is a single central tower attached to two computer
systems. A multiprocessor is regarded as a means to improve computing speeds, performance
and cost-effectiveness, as well as to provide enhanced availability and reliability.

Scalibilty :-
Flynns Classification of multiprocessor machines:

1. SISD = Single Instruction Single Data
2. SIMD = Single Instruction Multiple Data (Array Processors or Data Parallel machines)
3. MISD does not exist.
4. MIMD = Multiple Instruction Multiple Data Control

Pre-Experiment Questions:
Q.1 Define Multiprocessor?
Q.2 How do we measure scalability of Multiprocessor?
Q.3 Differentiate between Symmetric and Assymetric Multiprocessor.
Q.4 What is multiprocessing?
Procedure:
• Study various case study related to multi processors.

Result & Conclusion: Study of case study of Multi processors.

Post-Experiment Questions:
Q.1 What is a multiprocessor architecture?
Q.2 Describe various types of Multi processors.
Q.3 Differentiate between spinning and switching.

EXPERIMENT 11

Aim: Design and implementation of HALF SUBTRACTOR, FULL SUBTRACTOR using basic logic gates.

Equipment & Components Required:

S.No.	Equipments	Specification	Quantity
1	Digital IC Trainer kit	-	1
2	Digital Multimeter		1

S.No.	Components	Specification	Quantity
1			

Digital ICs
7400, 7402,
7404,
7408, 7432,
7486.
1 each

2

Patch cords
- 6

Theory:

a) To design and implement half Subtractor using logic gates

HALF SUBTRACTOR

CIRCUIT DIAGRAM TRUTH TABLE

b) To design and implement full subtractor using logic gates

INPUTS OUTPUTS
A B BIN D BOUT
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

FULL SUBTRACTOR

CIRCUIT DIAGRAM TRUTH TABLE

Pre-Experiment Questions:
Q.1 Explain the truth table of half subtractor.
Q.2 How many Ex-or and or or gate can be used to make a half subtractor?
Q.3 Why XOR gate is called an inverter?
Q.4 How many outputs are required for the implementation of a subtractor?

Procedure:
• Identify the pins.
• Connect the circuit as per circuit diagram.
• Obtain outputs with various input combinations.
• Verify it with the Boolean function using truth table

Result & Conclusion: All logical circuits have been implemented & verified through truth table.

Post-Experiment Question:
Q.1What are the applications of half subtractor?
Q.2 What are the applications of full subtractor?
Q.3 What does minuend and subtrahend denotes in a subtractor?
Q.4 How can a full subtractor be implemented?

INPUT X INPUT Y
OUTPUTS
D B
0 0 0 0
0 1 1 1
1 0 1 0
1 1 0 0

EXPERIMENT 12

Aim: Design and implementation of SISO and SIPO shift registers

Equipments & Components Required:

SL.No. Equipments Specification Quantity
1 Logic Simulator -

Theory:
A register is capable of shifting its binary information in one or both directions is known as
shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with
output of one flip flop connected to input of next flip flop. All flip flops receive common
clock pulses which causes the shift in the output of the flip flop. The simplest possible shift
register is one that uses only flip flop. The output of a given flip flop is connected to the input
of next flip flop of the register. Each clock pulse shifts the content of register one bit position
to right.

LOGIC DIAGRAM:
SERIAL IN SERIAL OUT:

TRUTH TABLE:
Sl.No. COMPONENT SPECIFICATION QTY.
1. D FLIP FLOP IC 7474 2
3. IC TRAINER KIT - 1
4. PATCH CORDS - 15

CLK
Serial in Serial out
1 1 0
2 0 0

3 0 0
4 1 1
5 X 0
6 X 0
7 X 1

SERIAL IN PARALLEL OUT:

TRUTH TABLE:

CLK

DATA
OUTPUT
Q
A
Q
B
Q
C
Q
D

1 1 1 0 0 0
2 0 0 1 0 0
3 0 0 0 1 1
4 1 1 0 0 1

Pre-Experiment Questions:
Q.1 State the features of Shift Registers.
Q.2 What is the main functionality of Shift Registers?
Q.3 How can parallel data be taken out of a shift register simultaneously?

Q.4 What is meant by parallel load of a shift register?

Procedure:
• Connections are given as per circuit diagram.
• Logical inputs are given as per circuit diagram.
• Observe the output and verify the truth table.

Result: All Shift registers have been implemented & verified through truth table.

Post-Experiment Questions:
Q.1 How can we use shift registers in serial communications? Explain.
Q.2 List the ICs which are used as 8 bit SISO, SIPO modes and as a bidirectional shift
register.
Q.3 Assume that a 4-bit serial in/serial out shift register is initially clear. We wish to store the
nibble 1100. What will be the 4-bit pattern after the second clock pulse?
Q.4 List the categories for classification of shift registers.

EXPERIMENT 13

Aim: Design and implementation of PISO and PIPO shift registers

Equipment's & Components Required:

S.No.	Equipment's Specification	Quantity
1	Logic Simulator	-

Theory:
A register is capable of shifting its binary information in one or both directions is known as
shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with
output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses which causes the shift in the output of the flip flop.The simplest possible shift
register is one that uses only flip flop. The output of a given flip flop is connected to the input

of next flip flop of the register. Each clock pulse shifts the content of register one bit position
to right.

LOGIC DIAGRAM:

PARALLEL IN SERIAL OUT:

S.No. COMPONENT SPECIFICATION QTY.
1. D FLIP FLOP IC 7474 2
3. IC TRAINER KIT - 1
4. PATCH CORDS - 15

TRUTH TABLE:
CLK Q3 Q2 Q1 Q0 O/P
0 1 0 0 1 1
1 0 0 0 0 0
2 0 0 0 0 0
3 0 0 0 0 1

PARALLEL IN PARALLEL OUT:

TRUTH TABLE:

CLK
DATA INPUT OUTPUT
D
A
D
B
D
C
D
D
Q
A
Q

B
Q
C
Q
D

1 1 0 0 1 1 0 0 1
2 1 0 1 0 1 0 1 0

Pre-Experiment Questions:
Q.1 State the features of Parallel Shift Registers.
Q.2 Differentiate between serial and parallel Shift Registers.
Q.3 How many clock pulses will be required to completely load serially a 5-bit shift register?
Q.4 What are the three output conditions of a three-state buffer?

Procedure:
• Connections are given as per circuit diagram.
• Logical inputs are given as per circuit diagram.
• Observe the output and verify the truth table.

Result & Conclusion: All shift registers have been implemented & verified through truth table.

Post-Experiment Questions:
Q.1 What are the differences between serial loading and parallel loading?
Q.2 In what type of register do we have access to only left most or right most flip flops
Q.3 How many clock pulses are required to serially enter a byte of data into an 8-bit register?
Q.4 List the ICs which are used as 8 bit PISO, PIPO modes and as a bidirectional shift
register.

APPENDIX

AKTU SYLLABUS

KCS 352: COMPUTER ORGANIZATION LAB

1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER and Implementing 4x1 and 8x1 MULTIPLEXERS.
4. Verify the excitation tables of various FLIP-FLOPS.
5. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
6. Design of an 8-bit ARITHMETIC LOGIC UNIT.
7. Design the data path of a computer from its register transfer language description.
8. Design the control unit of a computer using either hardwiring or microprogramming based
on its register transfer language description.
9. Design the control unit of a computer using either hardwiring or microprogramming based
on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path.