BATCH No.

Rashtreeya Sikshana Samithi Trust

R.V. College of Engineering®

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science & Engineering

Bengaluru -560059

Applied Logic Design & Computer Organization (22CS34)

III Semester B.E

LABORATORY RECORD
2023 – 2024

Name of the Student:_____

U S N :_____

Semester : _____ **Section :**_____

R.V. College of Engineering®

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science & Engineering

APPLIED LOGIC DESIGN & COMPUTER ORGANIZATION

(22CS34)

Laboratory Record (Autonomous Scheme 2022)

R.V. College of Engineering®

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science & Engineering

Laboratory Certificate

This is to certify that Mr. / Ms _____ _____has satisfactorily completed the course of Experiments in Practical_____ Prescribed by the Department of Computer Science and Engineering during the year _____

Name of the Candidate: _____

USN No.: _____ Semester:_____

Marks	
Maximum	**Obtained**

Signature of the staff in-charge Head of the Department

Date:

Department of Computer Science and Engineering
R V College of Engineering, Bengaluru

Vision

To achieve leadership in the field of Computer Science & Engineering by strengthening fundamentals and facilitating interdisciplinary sustainable research to meet the ever-growing needs of the society.

Mission

- To evolve continually as a center of excellence in quality education in computers and allied fields.
- To develop state-of-the-art infrastructure and create environment capable for interdisciplinary research and skill enhancement.
- To collaborate with industries and institutions at national and international levels to enhance research in emerging areas.
- To develop professionals having social concern to become leaders in top-notch industries and/or become entrepreneurs with good ethics.

Program Educational Objectives (PEOs):

PEO1: Develop Graduates capable of applying the principles of mathematics, science, core engineering and Computer Science to solve real-world problems in interdisciplinary domains.

PEO2: To develop the ability among graduates to analyze and understand current pedagogical techniques, industry accepted computing practices and state-of-art technology.

PEO3: To develop graduates who will exhibit cultural awareness, teamwork with professional ethics, effective communication skills and appropriately apply knowledge of societal impacts of computing technology.

PEO4: To prepare graduates with a capability to successfully get employed in the right role and achieve higher career goals or take-up higher education in pursuit of lifelong learning.

Program Outcomes (PO's):

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

PO2: Problem analysis: Identify, formulate, 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 public health and safety, and cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

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 the 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.

Course Outcomes (COs):

At the end of the course student will be able to

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply design requirements for digital systems and Computer organization
CO 2	Analyse the models used for designing various combinational and sequential circuits
CO3	Design optimized modern processors and memories for given specifications
CO 4	Develop applications of synchronous sequential networks using flip flop and registers.
CO5	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools
CO6	Indulge in developing novel interdisciplinary projects with effective oral and written communication skills

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2										
CO2	1	3			3							1
CO3	2	1	3		3							2
CO4	2		3						3			2
CO5	2	1	2	1	3	2	1	1	3	2	1	3
CO6	3	2	3	1	3	2	2	2	3	3	2	3

High-3: Medium-2 : Low-1

Course - PO Mapping												
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
Course	**H**	**M**	**H**	**L**	**H**	**L**	**L**	**L**	**M**	**M**	**L**	**M**

Do’s and Don’ts in the Laboratory

Do’s…….

1. Come Prepared to the lab.
2. Use the computers and Logic controller kit for academic purposes only.
3. Follow the lab exercise cycles as instructed by the department.
4. Keep the chairs back to its position before you leave.
5. Treat the trainer-kit with care.
6. Keep your lab clean.

Don’ts…….

1. Do not handle any equipment before reading the Instructions/Instruction manuals.
2. Read carefully the power ratings of the equipment before it is switched on whether ratings 230 V/50Hz or 115V/60 Hz. For Indian equipment’s, the power ratings are normally 230V/50Hz. If you have equipment with 115/60 Hz ratings, do not insert power plug, as our normal supply is 230V/50 Hz, which will damage the equipment.
3. Observe type of sockets of equipment power to avoid mechanical damage.
4. Do not forcefully place connectors to avoid the damage.
5. Strictly observe the instructions given by the Teacher/Lab Instructor.

List of Experiments

PART- A: Experiments

Ex. No.	Description	
1	Realization of Excess-3 Code converter with Parallel Adder and Subtractor using 4-bit adder, using the IC – 74283.	
2.	Realization of Full Adder and Full Subtract or using Multiplexers, using IC 74153.	
3	Design and realization One Bit and Two-Bit Magnitude Comparator using logic Gates.	
4	a)	Realization of Binary to Gray Code Converter using decoders, using the IC 74139.
	b)	Realization of single digit Seven segment display using the BCD to seven segment decoders, using the IC–7447 and Realization of Priority Encoder using IC–74147.
5.	Design and Realization of Master-Slave JK Flip Flop using only NAND Gates.	
6	a)	Realization of Synchronous Up-Down programmable counter using IC 74192.
	b)	Realization of Asynchronous decade counter and its variations using IC 7490
7	a)	Design and realization of sequence generator using IC 7495.
	b)	Realization of Ring counter and Johnson counter using IC 7495.
8	Design of Mod-N Synchronous Up counters using IC 74112 / 7476 / Simulation	

PART- B: Innovative Experiments (IE) / Open Ended Experiments

Design a 4/8-bit CPU using the LOGISIM simulator, for the following specifications.

1) Program Counter (Assume 256 program/code memory)
2) Instruction Register (Assume instruction size as 16 bit)
3) General Purpose Registers (RISC type-R0-R7): Use Harvard & Multiple Bus Architecture
4) ALU (to support 4-bit integer arithmetic operations & 4-bit logical operations)
5) Memory – 1024 ROM (to store instructions of size 16 bit) and 256 RAM (to store 4-bit data)
6) Implement the following instructions namely: MOV, ADD, SUB, LOAD, STORE, AND, XOR, NOT, BRANCH, BRANCH ON CONDITION.
7) Result to be displayed on 7-segment displays / reg tab of LOGISIM

Rubrics for Foundation of Computer Systems Designs Lab

PART A

Each experiment is evaluated for 10 marks.

Write-up and Execution rubrics (Max: 6 marks)				
		Excellent	**Good**	**Poor**
a.	**Knowledge / Understanding (2 Marks)**	Demonstrates thorough ability in describing and illustrating the function of the circuit (2-1.5)	Demonstrates considerable ability in describing and illustrating the function of the circuit (1-0.5)	Unable to demonstrate, describe and illustrate the function of the Circuit (0)
b.	**Circuit Construction (2 Marks)**	Excellent circuit functional, construction (2-1.5)	Fair circuit functional, construction (1-0.5)	Poor circuit functional, construction (0)
c.	**Circuit Data Recording (2 Marks)**	Correct data input and excellent presentation (2-1.5)	Correct data input and fair presentation (1-0.5)	Correct data input and poor presentation (0)
Viva rubrics (Max: 4 marks)				
		Excellent	**Good**	**Poor**
a.	**Circuit Data Analysis (2 Marks)**	Answers to data analysis questions, show complete understanding of how the circuit works (2-1.5)	Answers to data analysis questions, show solid understanding of how the circuit works (1)	Answers to data analysis questions, show no understanding of how the circuit works (0)
b.	**Applications (1 Mark)**	Demonstrates a high ability when wiring logic gates to construct logical circuits (1)	Demonstrates considerable ability when wiring logic gates to construct logical circuits(0.5)	Demonstrates very limited ability when wiring logic gates to construct logic circuits (0)
c.	**Communication of Ideas (1 Mark)**	Communicates all ideas clearly (1)	Communicates limited ideas (0.5)	Unable to communicate ideas(0)

PART B

Innovative Experiment is to be evaluated for 10 marks.

Sl.no	Criteria	Measuring Method	Excellent	Good	Poor
a.	**Innovation and Design (6 Marks)**	Innovativeness Design and realization	Demonstrates thorough ability in describing and illustrating the experiment (2-1.5)	Demonstrates considerable ability in describing and illustrating the experiment (1-0.5)	Unable to demonstrate, describe and illustrate the experiment (0)
b.	**Viva Voce (2 Marks)**	Communicatio of the design	Excellent understanding of the design (2-1.5)	Fair understanding of the design (1-0.5)	Poor understanding of the design (0)
c.	**Report and presentation (2 Marks)**	Documentation and Presentation skills	Correct data input and excellent presentation (2-1.5)	Correct data input and fair presentation (1-0.5)	Correct data input and poor presentation (0)

INDEX

Sl. No	Contents	Date	Page No.	Marks (10)			Staff Sign
				Record (06)	Viva (04)	Total (10)	
PART A							
1	a. Realization of Excess-3 Code converter with Parallel Adder and Subtractor using IC – 74283.						
2	Realization of Full Adder and Full Subtractor using IC 74153.						
3	Design and realization One Bit and Two-Bit Magnitude Comparator using Basic Gates.						
4	a. Realization of Binary to Gray Code Converter and vice-versa using IC 74139. b. Realization of single digit Seven segment display using the BCD to seven segment decoders, using the IC–7447 and Realization of Priority Encoder using IC -74147						
5	Design and Realization of Master-Slave JK Flip Flop using JK FFs & NAND Gates only.						
6	a. Realization of Up-Down programmable counter using IC 74192. b. Realization of decade counter and its variations using IC 7490.						
7	a. Realization of different Modes of operation of Universal Shift Register with IC 7495, Realization of Ring counter and Johnson counter using IC 7495 b. Design and realization of sequence generator using IC 7495.						
8	Design of Mod-N Synchronous Up counter using IC 74112 / 7476 / Simulation						
					Total Marks		
Reduced Part A Marks = (Total Marks*30)/ 80 =							

PART B							
Designing a 4bit CPU by interfacing registers, an ALU and a memory chip incorporating the relevant features							
Instructions Implemented	Date	Page No	Maximum Marks (10)				Staff Sign
			Design (6)	Viva (2)	Report (2)	Total (10)	

LAB INTERNALS		
RECORD	**Max – 30**	
PART B	**Max - 10**	
TEST	**Max - 10**	
TOTAL	**Max – 50**	
Signature of the faculty		

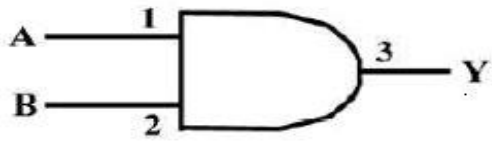
Introduction to Logic gates

This chapter is intended to introduce the student to some of the basic gates that will be used in the foundations of computer systems design laboratory experiments. The content of this chapter includes the Pin diagram of basic gates. Following are the basic gates that are dealt in this chapter.

1. IC 7408: 2- input AND gate
2. IC 7400: 2-input NAND gate
3. IC 7486: X-OR gate
4. IC 7432: 2-input OR gate
5. IC 7404: NOT gate
6. IC 7402: 2-input NOR gate
7. IC 7410: 3-input NAND gate
8. IC 7411: 3-input AND gate
9. IC 7420: dual 4-input NAND gate
10. IC 7421: dual 4-input AND gate

All the above-mentioned IC`s are known as small-scale integrated circuits which belongs to a logical family known as TTL.

IC7408: Two Input AND Gate



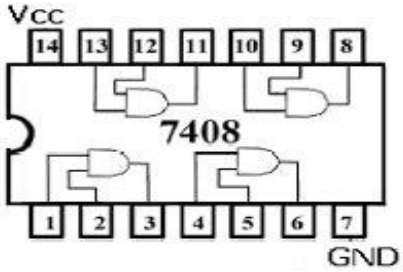
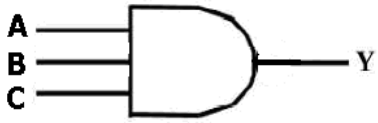


Fig.1.1 (a) AND Gate Logic Symbol **(b) Pin Diagram for the 7408 Quad AND**

IC7411: Three Input AND Gate



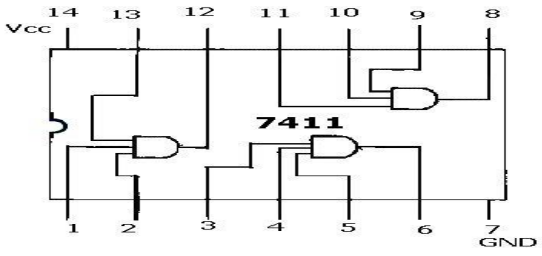
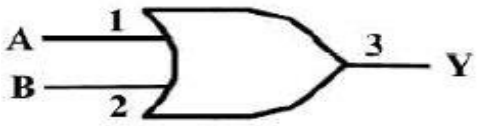


Fig.1.2 (a) Logic Symbol for three input AND gate **(b) Pin Diagram for the 7411 Triple 3 input AND gate**

IC7432: Two Input OR Gate



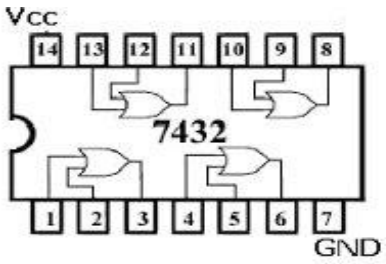
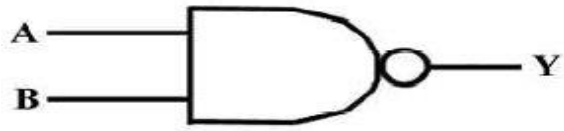


Fig.1.3 (a) OR Gate Logic Symbol **(b) Pin Diagram for the 7432 Quad OR**

IC 7400: Two Input NAND Gate



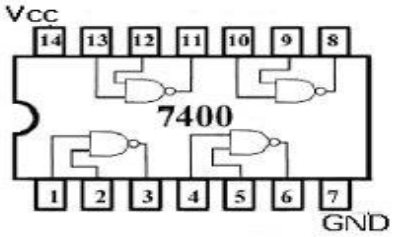


Fig.1.4 (a) NAND Gate Logic Symbol **(b) Pin Diagram for the 7400 Quad NAND**

IC 7410: Three Input NAND Gate

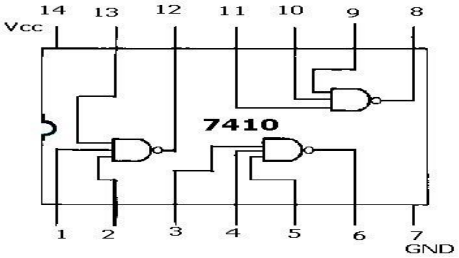
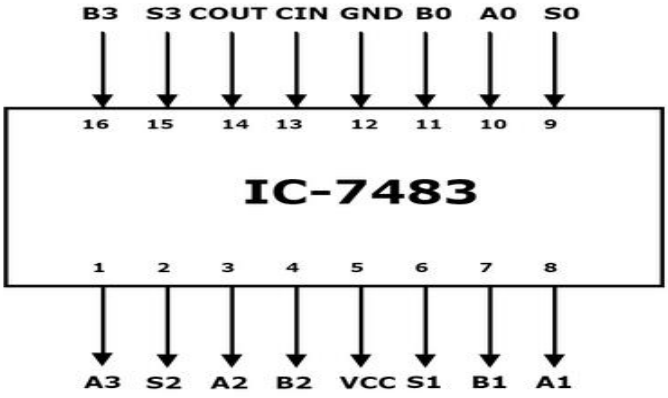


Fig.1.5 (a) Logic Symbol Of three input NAND **(b) Pin Diagram for the 7410 Triple 3 input NAND gate**

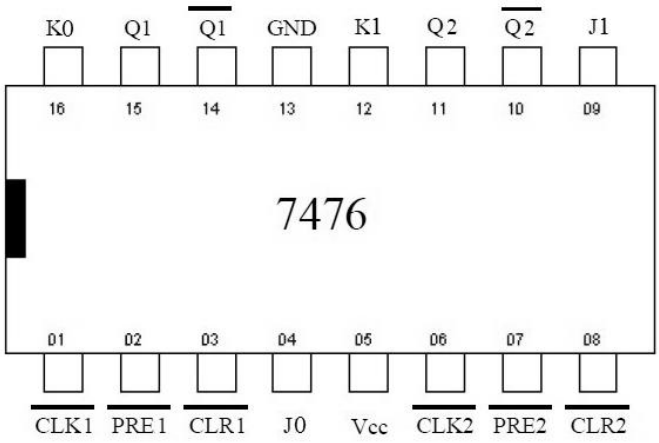
IC 74283 Parallel Adder



IC 7476: J K Flip-flop

IC 7476 : The 74LS76 is a Dual JK flip-flop IC, which contains two individual JK type bi-stable within a single chip enabling single or master-slave toggle flip-flops to be made. This is a negative-edge triggered flip-flop, with both active low preset and clear inputs.

.



IC 7420: Four Input NAND Gate

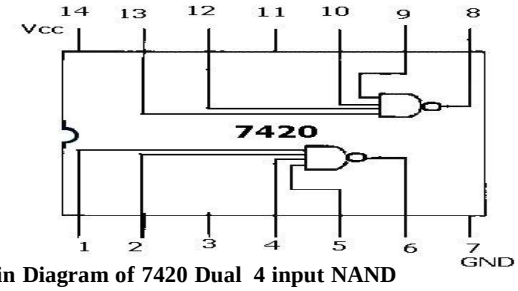


Fig.1.6 (a) Logic Symbol Of four input NAND gate gate **(b) Pin Diagram of 7420 Dual 4 input NAND**

IC 7402: Two Input NOR Gate

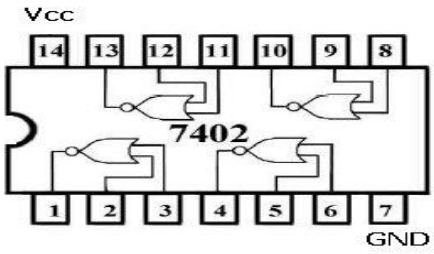
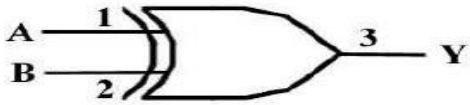


Fig.1.7 (a) NOR Gate Logic Symbol **(b) Pin Diagram for the 7402 Quad NOR**

IC7486: Two Input EX-OR Gate



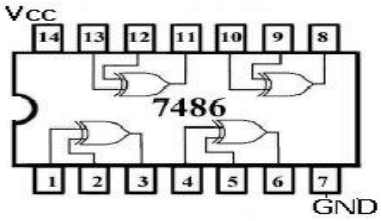
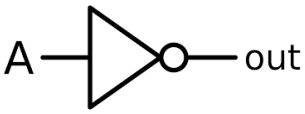


Fig.1.8 (a) X-OR Gate Logic Symbol **(b) Pin Diagram for the 7486 Quad X-OR**

IC7404: Two Input NOT Gate



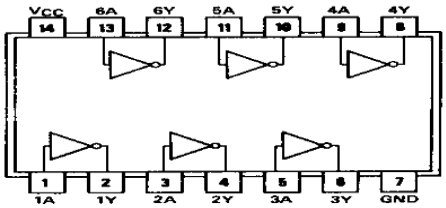


Fig: 1.9 (a) Pin Diagram for the 7404 NOT Gate **(b) Pin Diagram for the 7404 Inverter**

Experiment 01

1. Realization of Excess-3 Code converter with Parallel Adder and Subtractor using IC – 74283.

Aim: To design Excess-3 Code converter using IC 74283 parallel adder and verify the truth table

Components Required:

Sl. No.	Components	Specification	Quantity
1	Parallel Adder	IC -74283	1 No
2	2 Input EX-OR	IC -7486	1 No
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

Procedure: -

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Connect to VCC and apply various combinations of input according to the truth table.
4. Note down the output readings for different combinations of inputs.

Theory:

The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code.

Parallel adders are digital circuits that compute the addition of variable binary strings of equivalent or different size in parallel. The schematic diagram of a parallel adder is shown below, to add or subtract binary numbers. The circuit is laid out from right to left, similar to the way we add binary numbers. Therefore, the least significant column is on the right, and the most significant column is on the left. The boxes labeled FA are full adders. The Carry out from each full adder is the Carry-in to the next higher full adder. The numbers being processed are A4 A3 A2 A1and B4 B3 B2 B1, while the answer is S4 S3 S2 S1.

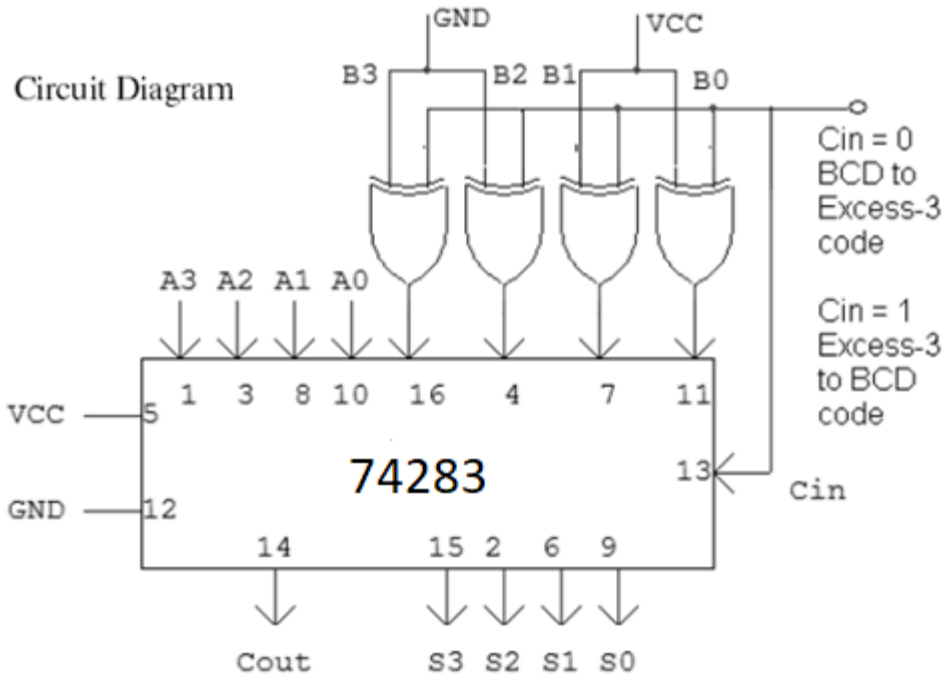
Binary Coded Decimal:

Binary Coded Decimal is a method of using binary digits to represent the decimal digits 0 through 9. It is possible to assign weights to the binary bits according to their positions. The weights in the BCD code are 8, 4, 2 and 1. Ex: $(137)_{10}$ - BCD equivalent $(0001\ 0011\ 0111)_{2.}$

Excess-3 Code:
This is an un-weighted code. Its code assignment is obtained from the corresponding value of BCD after the addition of $(0011)_2$.

BCD to Excess-3 (or) Excess-3 to BCD: Since each code uses four bits to represent a decimal digit, there must be four inputs and four output variables. The input variable are designated as B3, B2, B1, B0 and the output variables are designated as E3 , E2, E1, E0 in the truth table.

Circuit Diagram:



Truth Table:

i) BCD - EXCESS-3 CODE

BCD				EXCESS-3			
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				

ii) Excess-3 – BCD CODE

EXCESS-3				BCD			
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				

Inference:

Experiment 02

Realization of Full Adder and Full Subtractor using IC 74153.

A. **Aim:** To Realize combinational circuit Full Adder using IC74153

Components Required:

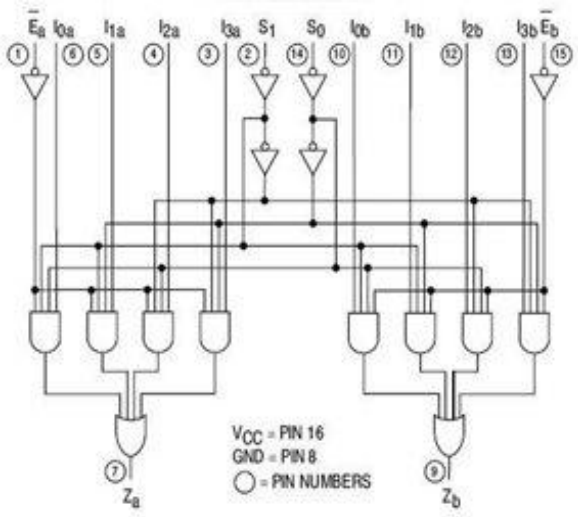
Sl. No.	Components	Specification	Quantity
1	Multiplexer	IC -74153	1 No
2	NOT GATE	IC -7404	1No.
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

Theory

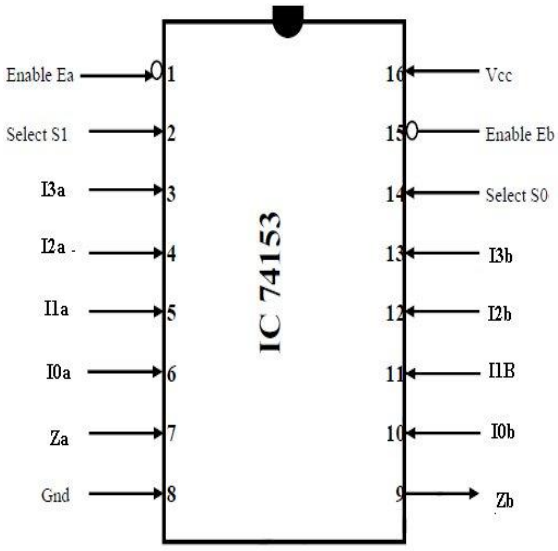
IC-74153 is a dual 4-to-1 multiplexer have 2 sets of 4:1 MUX name MUX A and MUX B, and the IC has 2 select lines S0 and S1 for each of the MUX but these selection line are internally shorted together as shown in the logic diagram below and hence the entire IC has only two selection pins. Each 4:1 MUX is associated with an active low enable pin namely E_a and E_b. With appropriated logic level on enable pins and on selection pins, logic levels on the input pin of any one or both the MUX can be transferred to the output. Z_a and Z_b are the output of MUX A and MUX B respectively.

74LS153 Dual 4-Input Multiplexer

Logical Diagram



Pin Diagram IC 74153



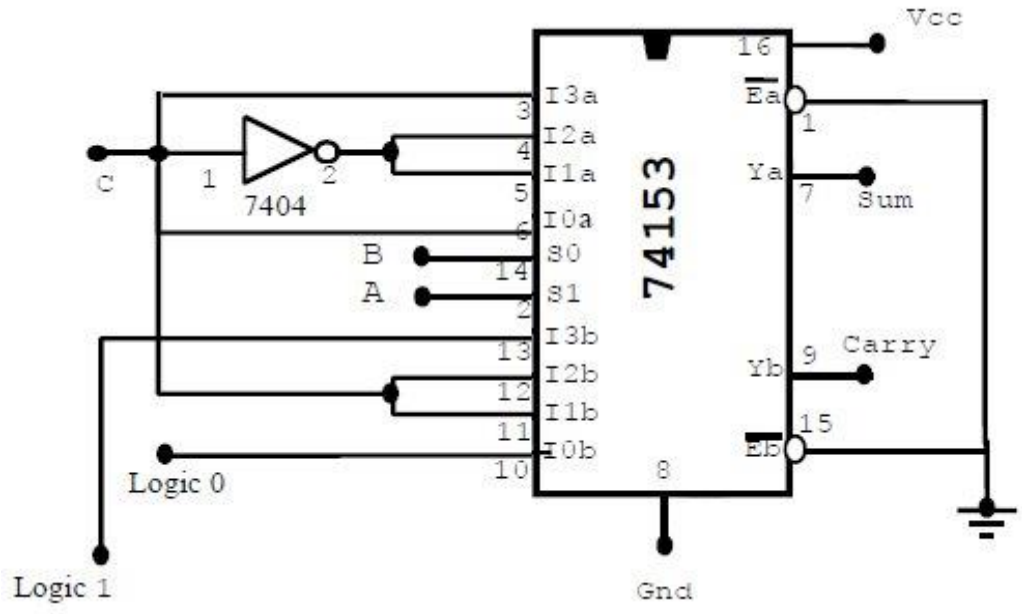
To realize any Boolean function with ' n' input variable, MUX require n-1 or n-2 selection pins. We use either Tabular method or Floyd method to generate the expressions. Of these two methods tabular method is been used in this experiment to realize Adder and Subtractor using 4:1 MUX.

In this experiment**,** Sum is realized on Mux A, Carry realized on Mux B, both enable pins of the Mux are connecting to logic 0, Connect A,B inputs to S1, S0 lines of Mux respectively. According to input A and B different data inputs lines (I0 to I3) are selected. Connect the data inputs to logic 0 or logic 1 according to the required outputs. Expression for sum and carry in terms input variable C, as shown below diagrams.

Truth table of Full adder:

Inputs			**Outputs**	
A	**B**	**C**	Sum	Carry
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Logic Diagram:



For Sum:

C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	1	1	0	Required Output
1	1	0	0	1	
	I0a=C	I1a=C'	I2a=C'	I3a=C	Input to selected lines

For Carry:

C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	0	0	1	Required Output
1	0	1	1	1	
	I0b=0	I1b=C	I2b=C	I3b=1	Input to selected lines

Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connects VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

B. Aim: To Realize a Full Subtractor using IC74153

Components Required:

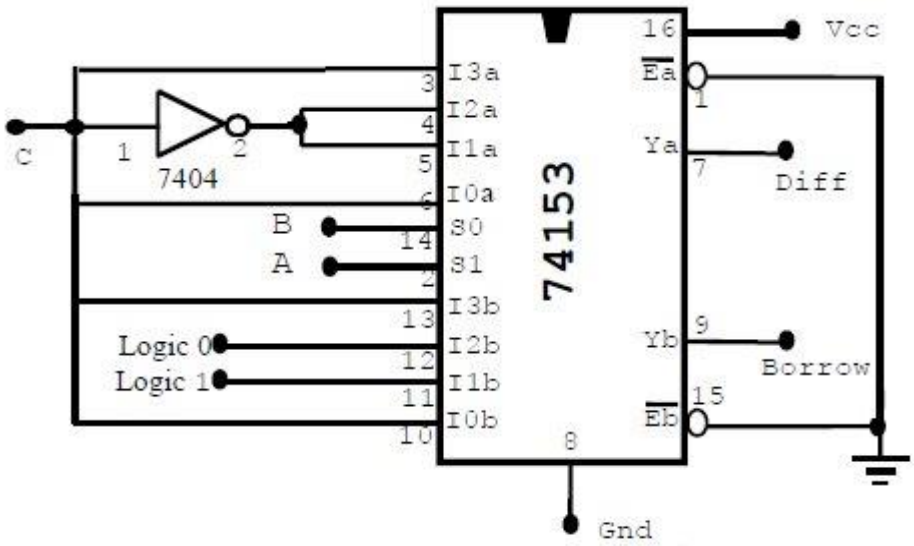
Sl. No.	Components	Specification	Quantity
1	Multiplexer	IC -74153	1 No
2	NOT GATE	IC -7404	1No.
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

Full Subtractor using 74153

Truth table of Full subtractor:

Inputs			Outputs	
A	**B**	**C**	Diff	Borrow
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Logic Diagram:



For Difference

C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	1	1	0	Required Output
1	1	1	0	1	
	I0b=C	I1a=C'	I2a=C'	I3a=C	Input to selected lines

For Borrow:

C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	1	0	0	Required Output
1	1	1	0	1	
	I0b=C	I1b=1	I2b=0	I3b=C	Input to selected lines

Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connects VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

Assigment 3: Realize 8:1 Mux using 2:1 & 4:1 Mux, 16:1 using 8:1 & 4:1

Experiment 03

Design and Realization of One and Two Bit Magnitude Comparator using Basic Gates

A. Aim: To design one bit comparator, two bit comparator using logic gates

Components Required:

Sl. No.	Components	Specification	Quantity
1	AND GATE	IC -7408	1 No
2	NAND GATE	IC-7400	1 No.
3	3 Input NAND GATE	IC-7410	1 No.
4	NOT GATE	IC -7404	1 No.
5	XOR GATE	IC-7486	1 No.
6	Digital trainer kit & patch cables	–	1 No, 20 patch cables

Theory

One Bit Comparator

A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitude. The outcome of the comparison is specified by 3 binary variables that indicate whether **A = B** or A< **B** or **A>B**.

Truth table for one bit Comparator

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

Boolean Expression (A<B) =

Boolean Expression (A=B) =

Boolean Expression (A>B) =

K Map (A<B)

A \ B	0	
0		
1		

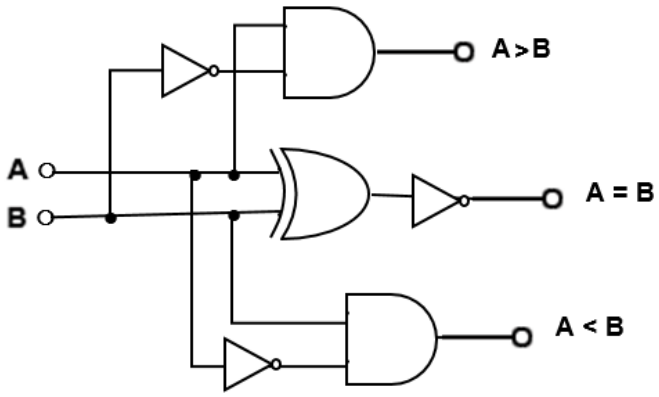
K Map (A=B)

A \ B	0	1
0		
1		

K Map (A>B)

A \ B	0	1
0		
1		

Logic Diagram of One Bit Comparator:



Two Bit Comparator

A magnitude comparator is a combinational circuit that compares four digital Bits A1, A0 and B1, B0 and determines their relative magnitude. The outcome of the comparison is specified by 3 binary variables that indicate whether A = B or A < B or A >B.

Truth table for Two bit Comparator

Observations:

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			

K-Map for (A<B)

Boolean Expression

(A<B)

A1 A0 \ B1 B0	00	01	11	10
00				
01				
11				
10				

Boolean Expression (A=B)

K-Map for (A=B)

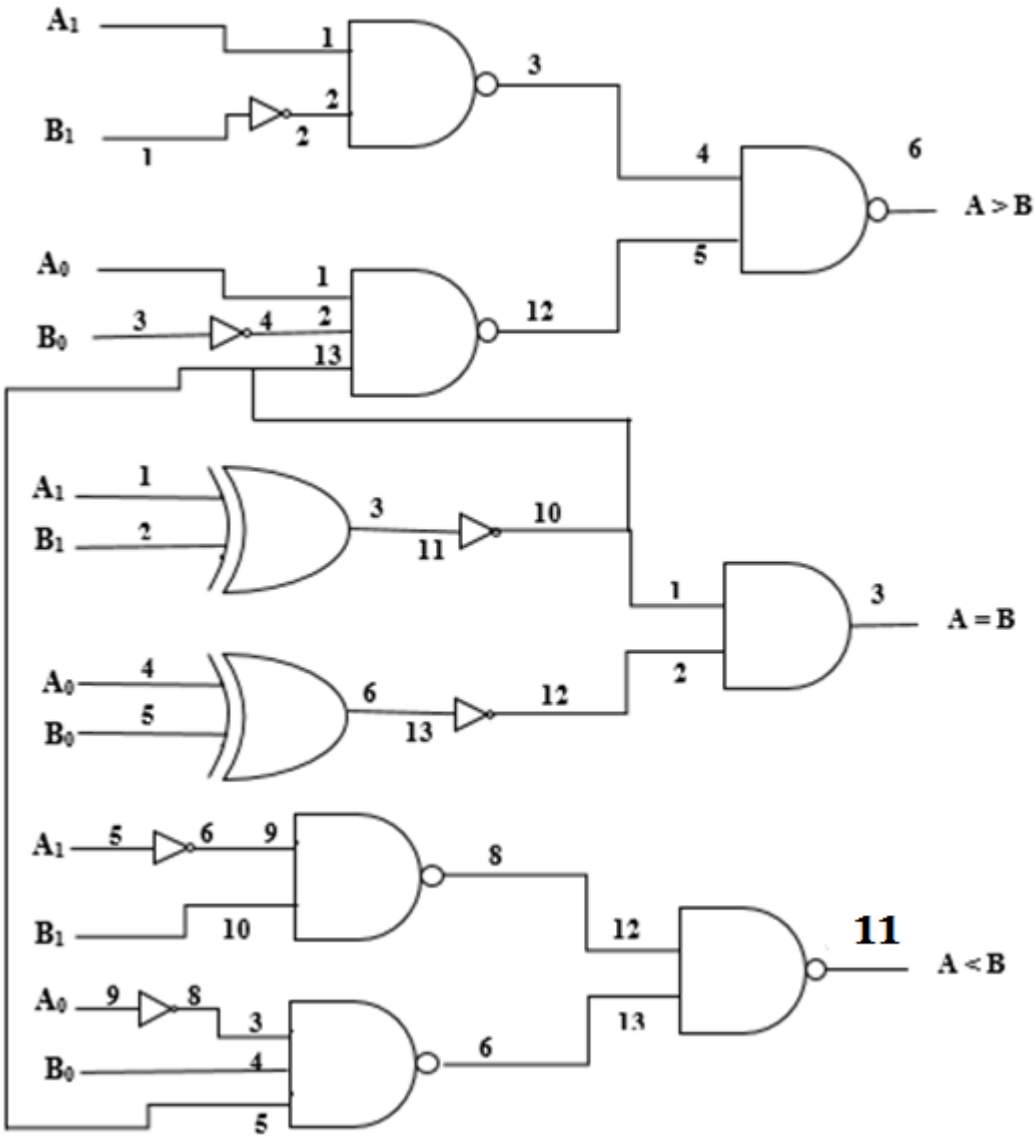
A1 A0 \ B1 B0	00	01	11	10
00				
01				
11				
10				

Boolean Expression (A>B)

K-Map for (A>B)

A1 A0 \ B1 B0	00	01	11	10
00				
01				
11				
10				

Logic diagram of two-bit comparator:



Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connect VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

Experiment 04

A. Realization of Binary to Gray Code Converter and vice-versa using IC 74139
B. Realization of Decoder using IC-7447 and Realization of Encoder using IC-74147

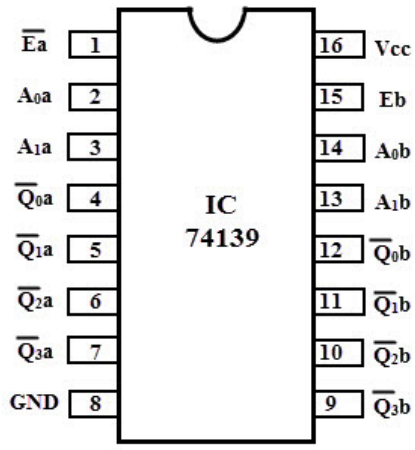
A. Aim : Realization of Binary to Gray Code Converter and vice-versa using IC 74139

Components Required:

Sl. No.	Components	Specification	Quantity
1	2-to-4 Line Decoder	74139	1 No
2	4 Input NAND gate	7420	1 No
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

Theory

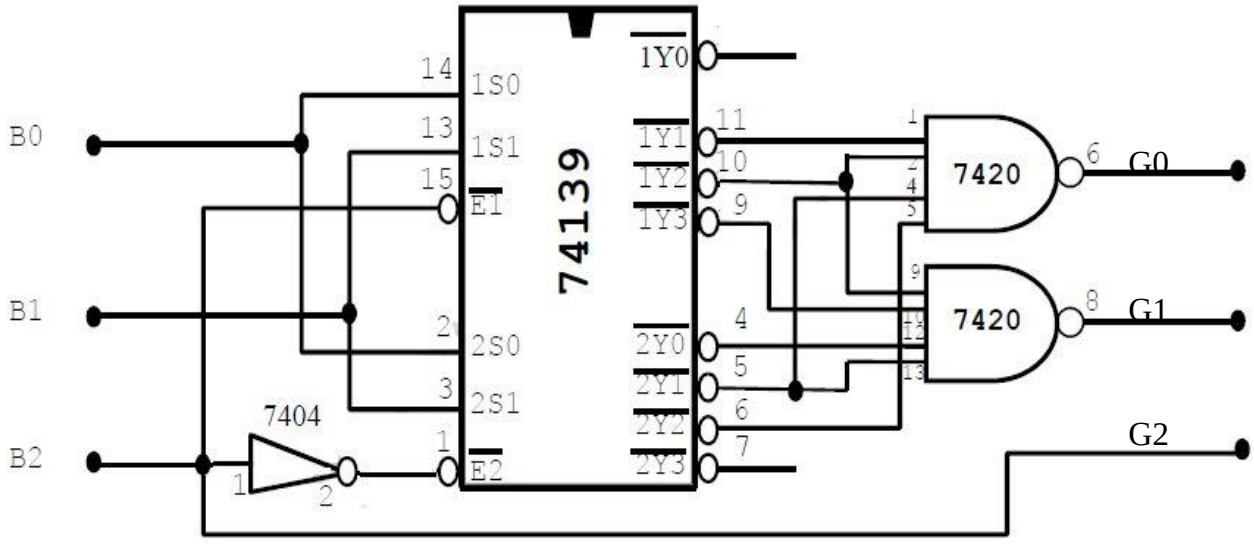
IC 74139: This IC is a dual 2-to-4-line decoder in the 7400 series. Each side of the chip is a decoder with an active low enable from a 2-bit address to four active low signals. The two decoders can be wired independently, or a /Y from one decoder can be wired to the /E on the other to calculate more complex combinational logic functions. This IC can function as a dual 2 x 4 DEMUX also. It has active low enable pin, 2 input/select lines and 4 output lines in each decoder.



Procedure: -

1. Verify IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connect pin 16 to VCC and pin 8 to GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Circuit Diagram:



Truth Table: 3 Bit binary code conversion using IC74139

Inputs			Outputs		
B2	**B1**	**B0**	**G2**	**G1**	**G0**
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

G0 = B1⊕B0

G1 = B2 ⊕B1

G2 = B2

G0 = ∑ m( )

G1 = ∑ m( )

G2 = ∑ m( )

Inference:

B. Realization of Decoder using IC-7447 and Realization of Encoder using IC-74147

1. Aim: To realize a Decoder using IC-7447 and Encoder using IC-74147

Components Required:

Sl. No.	Components	Specification	Quantity
1	Decoder	IC -7447	1 No
2	7 Segment display LED		1 No.
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

Theory:

74LS47 is a BCD to 7-segment decoder/driver IC. It accepts a binary coded decimal as input and converts it into a pattern to drive a seven-segment for displaying digits 0 to 9. Binary coded decimal (BCD) is an encoding in which each digit of a number is represented by its own binary sequence (usually of four bits).
For example 239 in BCD is represented as 0010 0011 1001.

74LS47 IC accepts four lines of BCD (8421) input data and generates their complements internally. The data is decoded with seven AND/OR gates to drive indicator LEDs of the seven segment directly. The outputs correspond to Common anode (CA) configuration of seven segment. The 74LS47 chip is used to drive 7 segment display. You must use the 74LS47 with a **common anode 7-segment display** (e.g. King bright part number SA03). The input to the 74LS47 is a binary number **DCBA** where D is 8s, C is 4s, B is 2s and A is 1s. The inputs DCBA often come from a binary counter. The display is only sensible if the binary number is between DCBA=0000 (0) and DCBA=1001 (9); this is called Binary Coded Decimal or BCD for short.

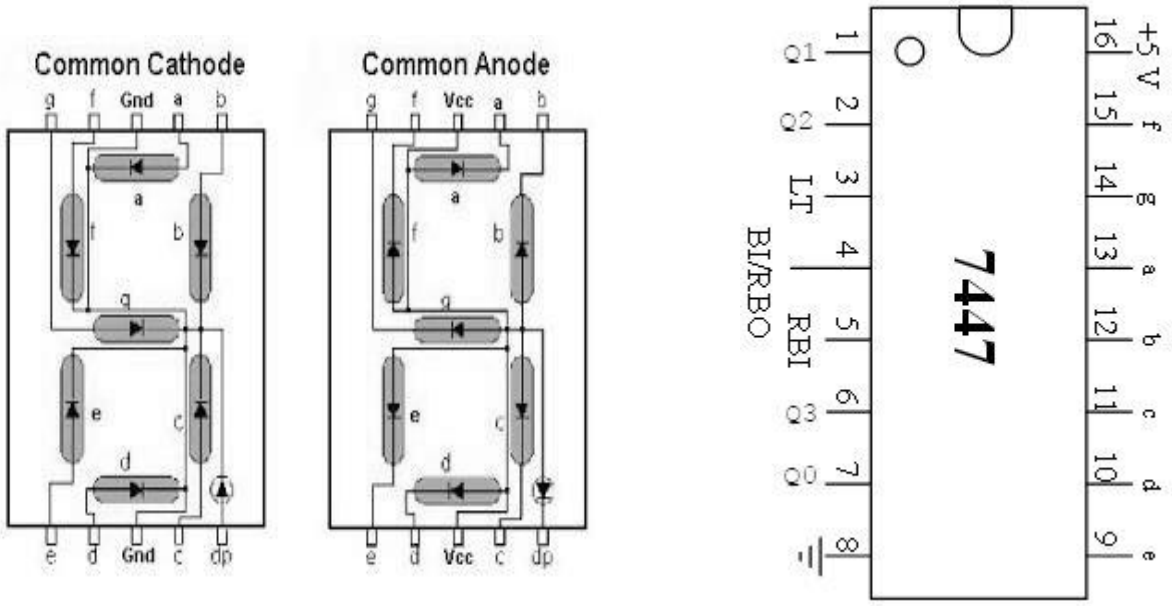
The inputs / $\overline{BI}$ $\overline{RBO}$, $\overline{RBI}$ $\overline{LT}$ and (LAMP TEST)are usually connected to +5v.

Note: A. BI/RBO serves as blanking input/ripple blanking output. BI and RBI must be open or held high. Input may be high or low.
B. When a low level is applied to the blanking input all segment outputs got a high level regardless of other inputs.
C. When ripple blanking input and other inputs A, B, C and D are at low level with the lamp test input at high level all segment outputs go to a high level and RBO goes to low level (response condition).
D. When BI/RBO is open or held at high and a low is applied to LT all segment outputs go to low.

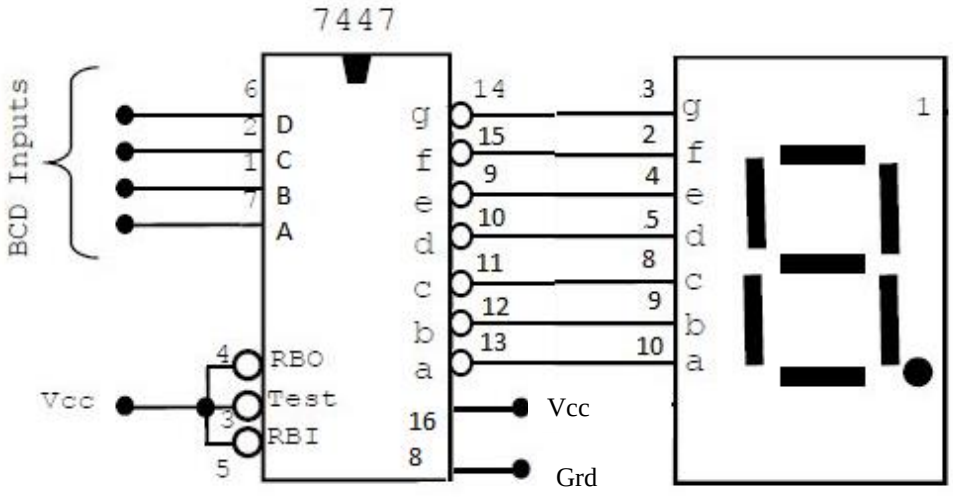
Procedure

1. Connect V_{CC} [pin 16], $\overline{LT}$ $\overline{BI}$ $\overline{RBO}$ $\overline{RBI}$ [pin 3], / [pin 4] and [pin 5] to 5v.
2. Connect GND [pin 8] to 0v.
3. Connect DCBA [pins 1, 2, 6 and 7] to DCBA on your counter.
4. Connect [pins 9-15] to abcdefg on the **common anode** 7-segment display.
5. Apply BCD input on DCBA lines using toggle switches and observe the output on7-segment LED display unit

Pin Diagram IC 7447



Using 7447 (Decoder) and seven segment decoder Logic diagram



Truth Table:

Binary Inputs	Decoder Outputs							7 Segment Display Outputs
D C B A	a	b	c	d	e	f	g	
0 0 0 0	1	1	1	1	1	1	0	0
0 0 0 1	0	1	1	0	0	0	0	1
0 0 1 0	1	1	0	1	1	0	1	2
0 0 1 1	1	1	1	1	0	0	1	3
0 1 0 0	0	1	1	0	0	1	1	4
0 1 0 1	1	0	1	1	0	1	1	5
0 1 1 0	1	0	1	1	1	1	1	6
0 1 1 1	1	1	1	0	0	0	0	7
1 0 0 0	1	1	1	1	1	1	1	8
1 0 0 1	1	1	1	1	0	1	1	9

Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connects VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

B. Aim: To realize an Encoder using IC 74147

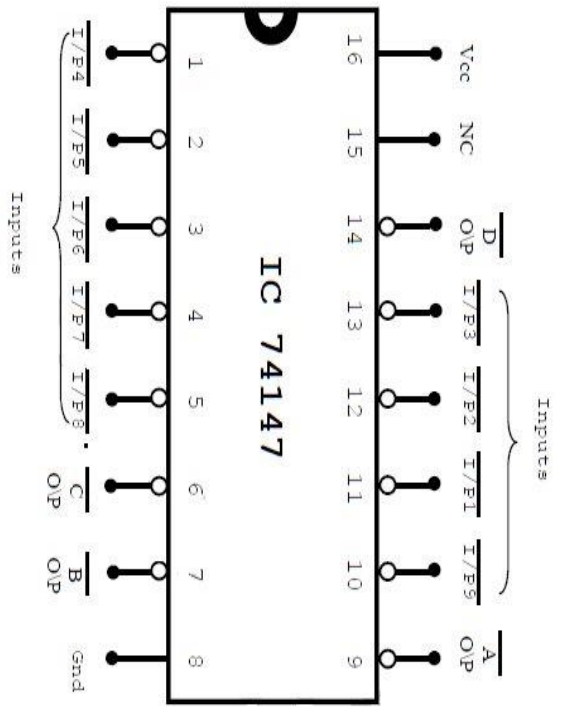
Components Required:

Sl. No.	Components	Specification	Quantity
1	Encoder	IC -74147	1 No
2	Digital trainer kit & patch cables	–	1 No, 20 patch cables

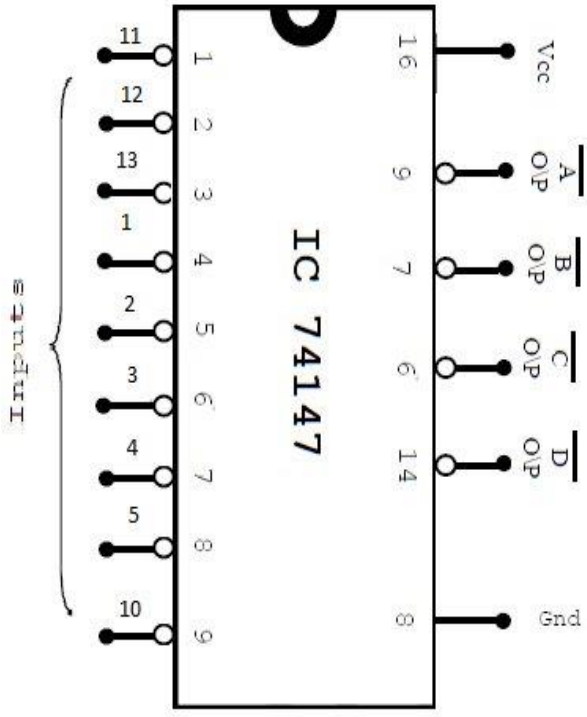
Theory:
IC 74147

The 74HC/HCT147 is 9-input priority encoders, accept data from nine active LOW inputs A 0 to A8 and provide a binary representation on the four active LOW outputs Y 0 - Y3. A priority is assigned to each input so that when two or more inputs are simultaneously active, the input with the highest priority is represented on the output, with input line A8 having the highest priority. The device provide the 9 to 4-line priority encoding function by use of the implied decimal "zero". The "zero" is encoded when all nine data inputs are HIGH, forcing all four outputs HIGH.

Pin Diagram



Logic Diagram



Truth Table:

Active – Low Decimal Inputs									Active Low BCD Output			
1	2	3	4	5	6	7	8	9	D	C	B	A
1	1	1	1	1	1	1	1	1				
0	1	1	1	1	1	1	1	1				
X	0	1	1	1	1	1	1	1				
X	X	0	1	1	1	1	1	1				
X	X	X	0	1	1	1	1	1				
X	X	X	X	0	1	1	1	1				
X	X	X	X	X	0	1	1	1				
X	X	X	X	X	X	0	1	1				
X	X	X	X	X	X	X	0	1				
X	X	X	X	X	X	X	X	0				
X	X	X	X	X	X	X	X	X				

Procedure:

1. Verify IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connect VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

Experiment 05

Realization of Master-Slave J K Flip-Flop,
i) Using 74LS76 - Dual JK flip-flop IC
ii) Using NAND gates.

Aim: To realize Master-Slave JK Flip-Flops using NAND Gates and IC 7476

Components Required:

Sl. No.	Components	Specification	Quantity
1	Flip-Lop	IC -7476	1 No
2	NAND GATE	IC-7400	1 No
3	3 INPUT NAND GATE	IC-7410	2 No
4	Digital trainer kit & patch cables	–	1 No, 35 patch cables

Theory:

Flip-Flop

A flip-flop circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit) until directed by an input signal to switch states. A flip-flop has two useful states, when Q=1 and Q' = 0, it is in the set state (or 1-state). When Q = 0 and Q' = 1, it is in the clear state (or 0-state). The outputs Q and Q' are complements of each other and are referred to as the normal and complemented outputs, respectively. The binary state of the flip-flop is taken to be the value of the normal output.

The NAND basic flip-flop circuit operates with both inputs normally at 1 unless the state of the flip-flop has to be changed. The application of a momentary 0 to the set input causes output Q to go to 1 and Q' to go to 0, thus putting the flip-flop into the set state. After the set input returns to 1, a momentary 0 to the reset input causes a transition to the clear state. When both inputs go to 0, both outputs go to 1 – a condition avoided in normal flip-flop operation.

JK Flip-Flop

A JK flip-flop is a refinement of the RS flip-lop in that the indeterminate state of the RS type is defined in the JK type. Inputs J and K behave like inputs S and R to set and clear the flip-flop (note that in a JK flip-flop, the letter J is for set and the letter K is for clear). When inputs are applied to both J and K simultaneously, the flip-flop switches to its complement state, that is, if Q=1, it switches to Q=0, and vice versa.

Master – Slave Flip-flop:

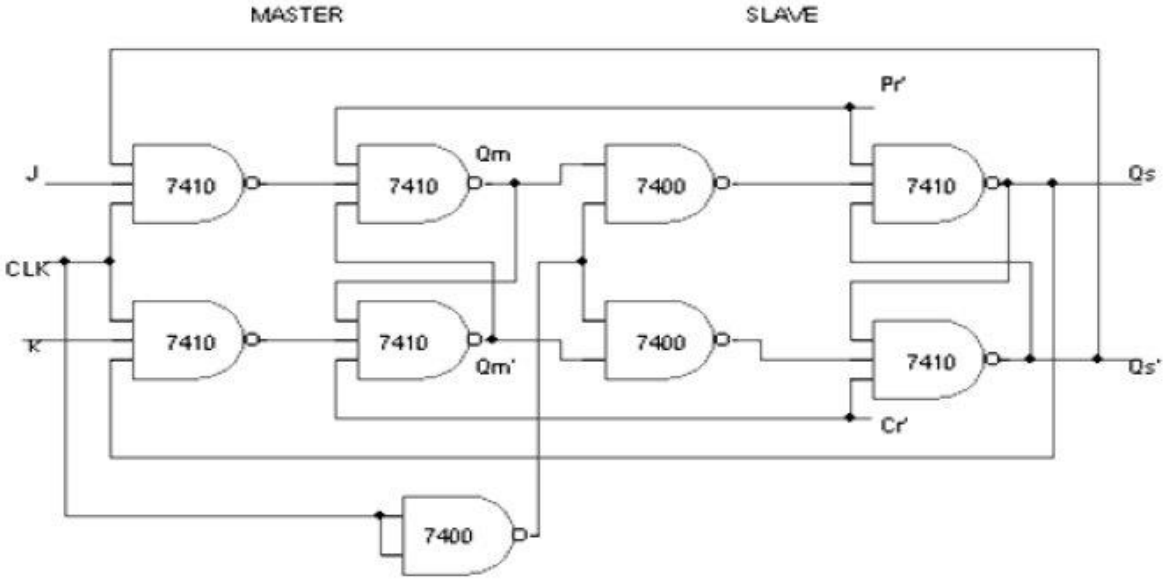
A master – slave flip-flop is constructed from two separate flip-flops. One circuit serves as a master and the other as a slave, and the overall circuit is referred to as a master-slave flip -flop. It consists of a master flip-flop and a slave flip-flop. When clock pulse CP is 0, the output of the inverter is 1. Since the clock input of the slave is 1, the flip-flop is enabled and output Q is equal to Y, while Q' is equal to Y'. The master flip flop is disabled because CP = 0. When the pulse becomes 1, the information then at the external R and S inputs is

transmitted to the master flip-flop. The slave flip-flop, however, is isolated as long as the pulse is at its 1 level, because the output of the inverter is 0. When the pulse returns to 0, the master flip-flop is isolated, which prevents the external inputs from affecting it. The slave flip-flop then goes to the same state as the master flip-flop.

Procedure:

1. Rig up the circuit as shown in the figure
2. Keep both preset and reset/clear at logic high
3. Verify the truth table for JK Master-Slave

Logic Diagram of JK Master Slave using NAND Gate



Truth Table:

CLK	J	K	Qn+1(next state)
H	0	0	
H	0	1	
H	1	0	
H	1	1	

Inference:
Assignment: Realize D and T flip-flops using SR/JK flip-flops.

Experiment 06

A. Realization of Up and Down Programmable Counter using IC74192 & IC74193

Aim: To Realization of counter using IC74192 & IC74193

(i) Up / Down counter
(ii) Presettable Up counter
(iii) Presettable Down counter

Components Required:

Sl. No.	Components	Specification	Quantity
1	COUNTER	IC -74192	1 No
2	NOT GATE	IC-7404	1 No
3	4 INPUT NAND GATE	IC-7420	1 No
4	Digital trainer kit & patch cables	–	1 No, 20 patch cables

IC74192:

74HC192: 74HCT192 is a 4-bit synchronous decade binary up/down counter.
Separate up/down clocks, ClkU and ClkD respectively, simplify operation. The outputs change state synchronously with the LOW-to-HIGH transition of either clock input. If the CLKU clock is pulsed while CLKD is held HIGH, the device will count up. If the CLKD clock is pulsed while CLKU is held HIGH, the device will count down.

Only one clock input can be held HIGH at any time to guarantee predictable behavior. The device can be cleared at any time by the asynchronous master reset input (Clr); it may also be loaded in parallel by activating the asynchronous parallel load input.

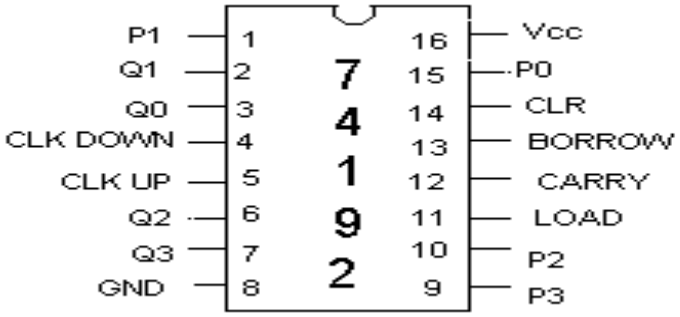
The terminal count up (Carry) and terminal countdown (Borrow) outputs are normally HIGH. When the circuit has reached the maximum count state of 09, the next HIGH-to-LOW transition of CLKU will cause carry to go LOW. Carry will stay LOW until CLKU goes HIGH again, duplicating the count up clock. Likewise, the Borrow output will go LOW when the circuit is in the zero state and the CLKD goes LOW.

The terminal count outputs can be used as the clock input signals to the next higher order circuit in a multistage counter, since they duplicate the clock waveforms. Multistage counters will not be fully synchronous, since there is a slight delay time difference added for each stage that is added.

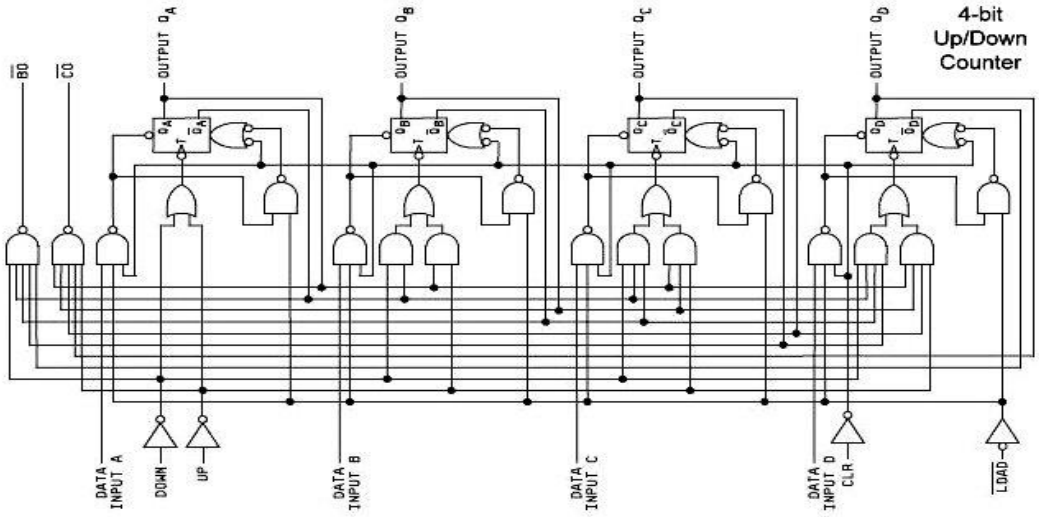
The counter may be preset by the asynchronous parallel load capability of the circuit. Information present on the parallel data inputs (P0 to P3) is loaded into the counter and appears on the outputs (Q0 to Q3) regardless of the conditions of the clock inputs when the parallel load input is LOW.

A HIGH level on the master reset input will disable the parallel load gates, override both clock inputs and set all outputs (Q0 to Q3) LOW. If one of the clock inputs is LOW during and after a reset or load operation, the next LOW-to-HIGH transition of that clock will be interpreted as a legitimate signal and will be counted. Inputs include clamp diodes. This enables the use of current limiting resistors to interface inputs to voltages in excess of VCC.

Pin Diagram



Pin Configuration



Truth Table

Load	Clear	Clk–up	Clk–down	Mode
X	1	X	X	Preset to zero
1	0	↑	1	Up – count
1	0	1	↑	Down count
0	0	X	X	Preset
1	0	1	1	Stop count

Procedure Mod 10 Up & Down Counter: -

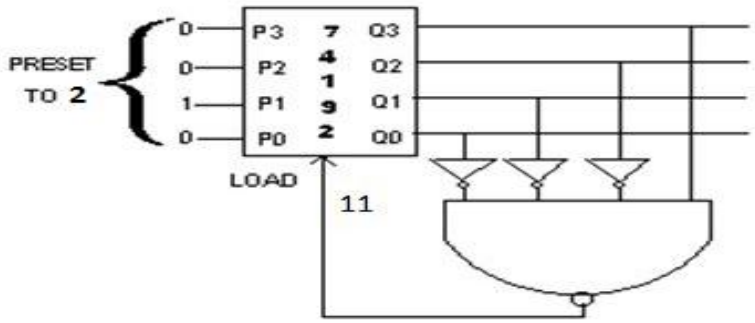
For up count clear the counter.

1. Clk-down=logic-1
2. Keep clear = logic-0 to clear the output, load can be maintained at either logic-0 or logic-1.
3. Then, first make Load=logic-1 and then Clear=logic-0, to start the counter.
4. Mono-pulses are applied at Clk-up.
5. Observe the count sequence at Q_3 Q_2 Q_1 Q_0.

For down count, clear the counter.

1. Clk-up=logic-1
2. Keep clear = 0, load = 0 to set the output.
3. Keep clear = 0 change load to 1, to start the counter.
4. Mono-pulses are applied at Clk-down.
5. Observe the count sequence at Q3 Q_2 Q_1 Q_0.

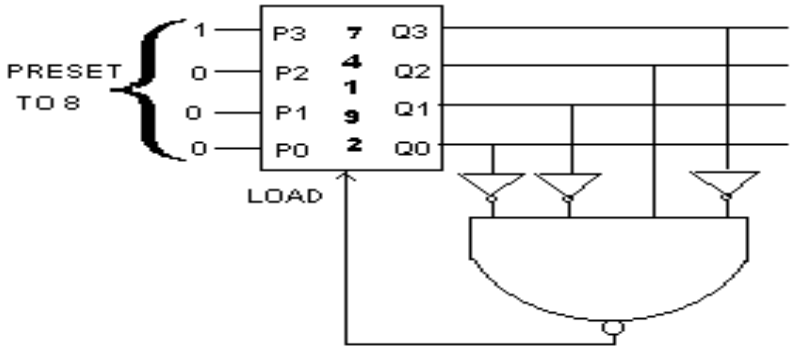
Preset table Mod 6 Up Counter
Circuit Diagram:



Procedure: -
To count from 2 to 7

1) Preset the data input to 0010.
2) Set Clk-down=logic-1
3) Set Clear=logic-0
4) Connect the load to the output of the NAND gate
5) Mono-pulses are applied at Clk-up
6) The o/p count should changes from 2 to 7, 2 to 7 & so on.

Preset table Mod 4 Down Counters:
Circuit Diagram:



Procedure for down counter

1) Preset the data input to 1000.
2) Set Clk-up=logic-1
3) Set Load=logic-0 and Clear=logic-0
4) Connect the load to the output of the NAND gate
5) Mono-pulses are applied at Clk-down
6) The o/p count should changes from 8 to 5, 8 to 5 & so on.

Note: Students can also design Up – Down Synchronous Counter using Ic74193 74193 is a synchronous 4 bit binary up – counter which can be cleared and preset to any count

1) Pin details and functional table of 74192
2) The procedure for both up and down counting as same as in 74192 except that 74193 counts 0, 1, 2 ----- 15 in up counting mode and 15, 14, 13 ------2, 1, 0 in down – counting mode
3) In up – counting mode carry goes low when Q_3 Q_2 Q_1 Q_0 changes from 1111 to 0000
4) In down – counting mode, borrow goes low when Q_3 Q_2 Q_1 Q_0 changes from 0000 to 1111

Inference:

Assignment: Design of Multi-stage Up/Down counters using 74192

B. Realization of decade counter and its variations using IC 7490.

Aim: To design and realize a decade counter using IC 7490.

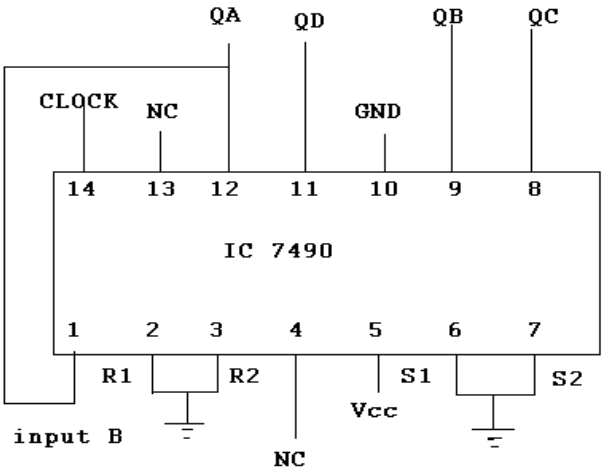
Components Required:

Sl. No.	Components	Specification	Quantity
1	BCD Counter	IC 7490	1 No
2	3-input t NAND gate	IC 7410	1 No
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

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.

Circuit Diagram:



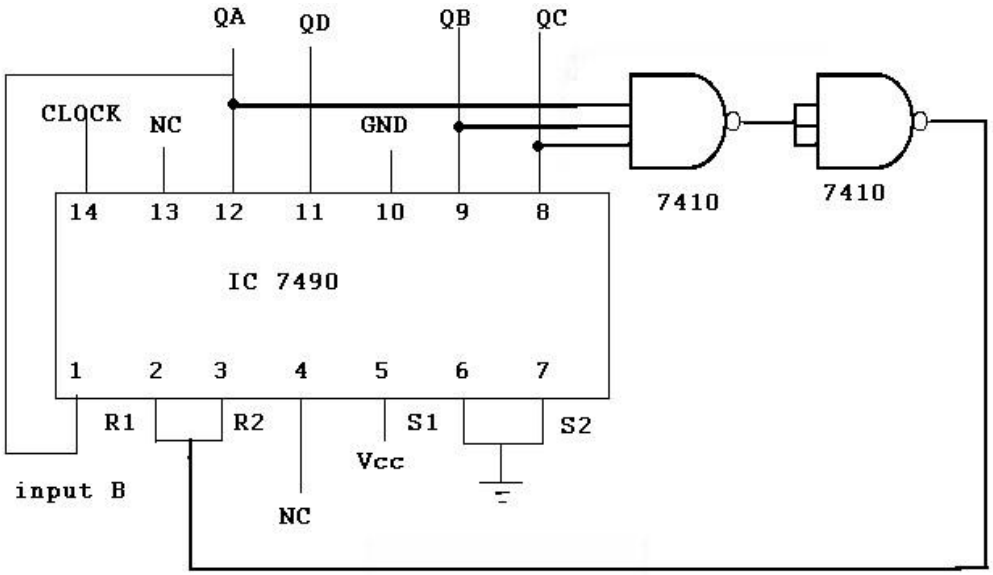
Truth Table:

Q_D	Q_C	Q_B	Q_A
0	0	0	0

Example 2:

7490 AS A DIVIDE BY N COUNTER (N=7):

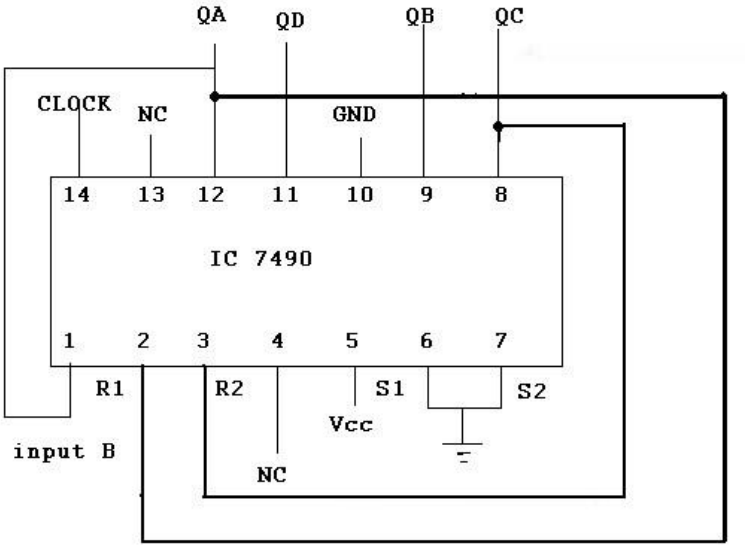
Circuit Diagram:



Truth Table:

Q_D	Q_C	Q_B	Q_A
0	0	0	0

Example 3: DIVIDE BY 5 COUNTERS:



Truth Table:

Q_D	Q_C	Q_B	Q_A
0	0	0	0

Inference:

Experiment 07

A. Realization of sequence generator using IC 7495

Aim: To design and realize a sequence generator using IC 7495

Components Required:

Sl. No.	Components	Specification	Quantity
1	Shift Register	IC -7495	1 No
2	Ex-OOR gate	IC-7486	1 No
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

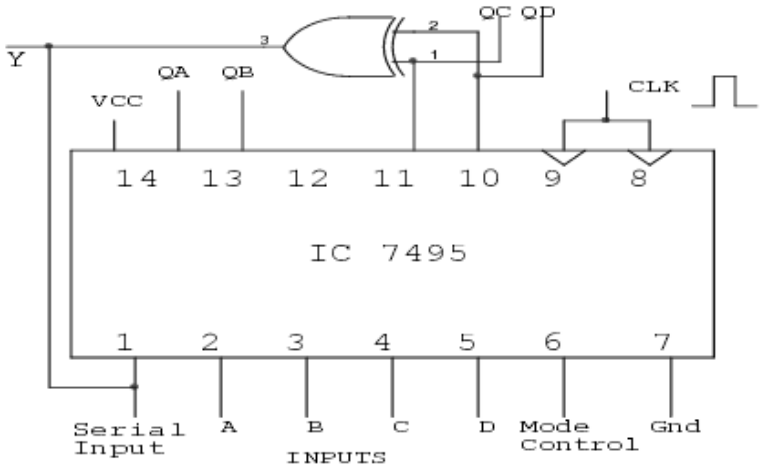
Theory

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. By Keeping mode=1. Load the input A,B,C,D as in Truth
5. Table 1st Row and give a clock pulse
6. For count mode make mode = 0.
7. Verify the Truth Table and observe the outputs.

Example 1:
Sequence = 100010011010111
Sequence length S = 15

Circuit Diagram:

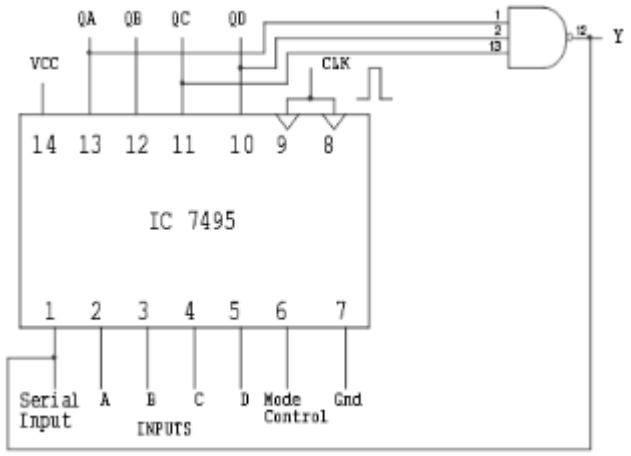


Truth Table:

Q_A	Q_B	Q_C	Q_D	$Y = Q_C$ xor Q_D
1	1	1	1	0
0	1	1	1	0
0	0	1	1	0
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0
0	0	1	0	1
1	0	0	1	1
1	1	0	0	0
0	1	1	0	1
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1
	1	1	1	
		1	1	
			1	

DESIGN 3:
Sequence = 1101011
Sequence length S = 7

Y = QA'+ QB' + QC'



Q_A	Q_B	Q_C	Q_D	Y
1	1	1	0	1
1	1	1	1	0
0	1	1	1	1
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	1
	1	1	0	
		1	1	
			1	

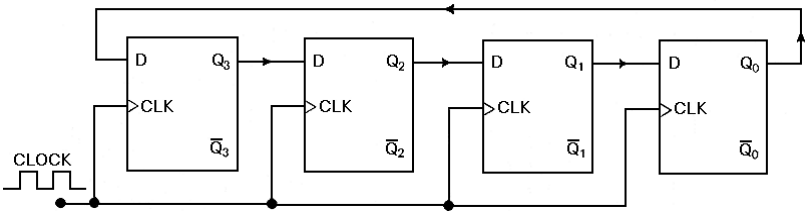
B. Realization of Ring and Johnson Counter using IC-7495

Components Required:

Sl. No.	Components	Specification	Quantity
1	4 BIT SHIT REGISTER	IC -7495	1 No
2	NOT GATE	IC-7404	1 No
3	Digital trainer kit & patch cables	–	1 No, 20 patch cables

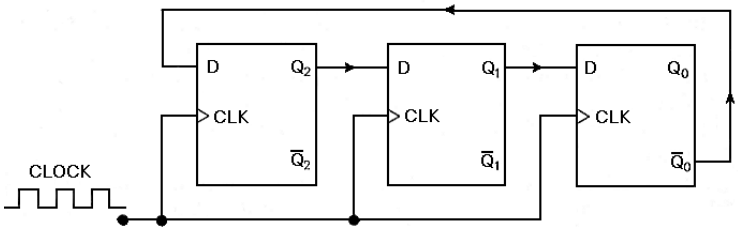
Theory

a. Ring counter : A ring counter is a type of counter composed of a type circular shift register. The output of the last shift register is fed to the input of the first register. Ring counters are used in hardware logic design (e.g. ASIC & FPGA) to create simple finite state machines. A binary counter will require an adder circuit which is substantially more complex than a ring counter. Additionally, the worst case propagation delay on an adder circuit will be proportional to the number of bits in the code (due to the carry propagation). The propagation delay of a ring counter will be a constant regardless of the number of bits in the code. The complex combinational logic of an adder can create timing errors which may result in erratic hardware performance. Also, ring counters with Hamming distance 2(or more) allow the detection of single bit upsets that can occur in hazardous environments.



The disadvantage of ring counters is that they are lower density codes. A binary counter can represent 2^N states, where N is the number of bits in the code, whereas an Overbank counter can represent only N states and a Johnson counter can represent only 2N states. This may be an important consideration in hardware implementations where registers are more expensive than combinational logic.

b. **JOHNSON COUNTER :** The Johnson Ring Counter or "Twisted Ring Counters", is another shift register with feedback exactly the same as the standard Ring Counter above, except that this time the inverted output Q of the last flip-flop is now connected back to the input D of the first flip-flop as shown below.

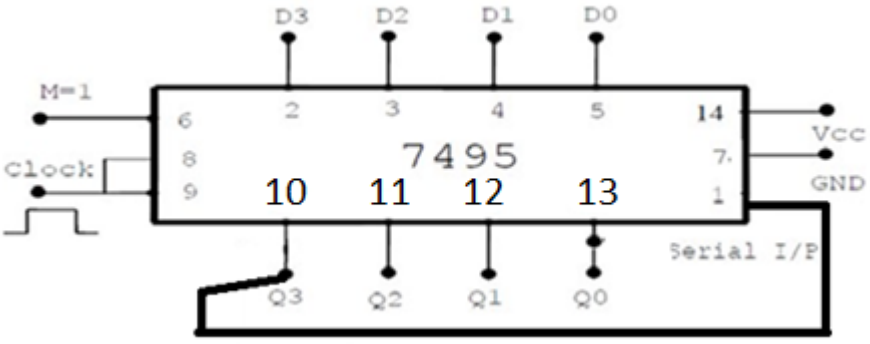


The main advantage of this type of ring counter is that it only needs half the number of flip-flops compared to the standard ring counter then it's modulo number is halved. So a "n-stage" Johnson counter will circulate a single data bit giving sequence of 2n different states and can therefore be considered as a "mod-2n counter".

Truth Table:

Mode M	Clock	Inputs				Output			
		D3	D2	D1	D0	Q3	Q2	Q1	Q0
1	0	1	0	0	0				
0	1								
0	2								
0	3								
0	4								
0	5								

Logic Diagram for Ring Counter:



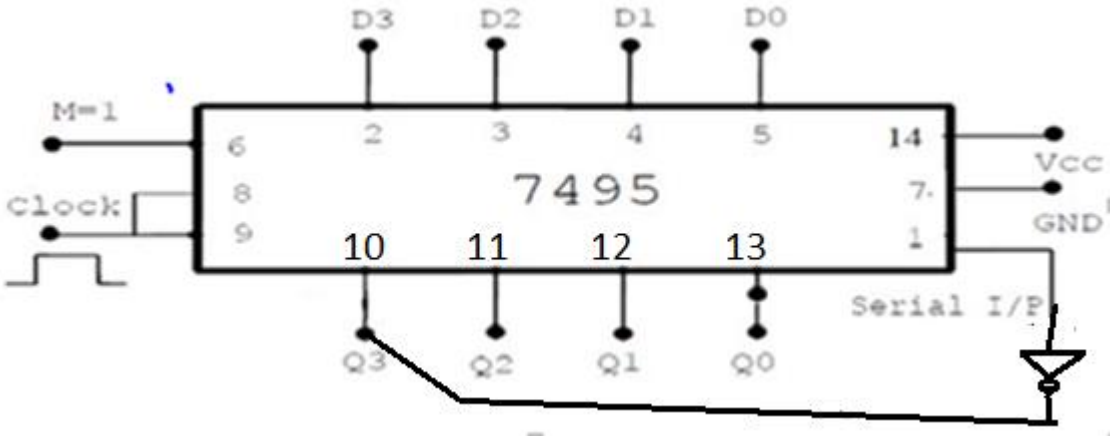
Procedures for Ring counter:

1. Connection is made as shown in the logic diagram.
2. 4 bit Data is loaded into shift registers **D3, D2, D1 & D0.**
3. Keeping Mode control M=1, clock pulse are applied, the data present in **D3, D2, D1 & D0** will appear at **Q3 , Q2, Q1 & Q0 respectively.**
4. Keeping Mode control M=0, clock pulse are applied one by one and data arriving out serially at Q0 is observed.
5. Connect the Q3 to Serial i/p and keep applying clock pulses to realize ring counter.

Truth Table:

Mode M	Clock	Inputs				Output			
		D3	D2	D1	D0	Q3	Q2	Q1	Q0
1	0	0	0	0	0				
0	1								
0	2								
0	3								
0	4								
0	5								
0	6								
0	7								
0	8								

Logic Diagram for Johnson Counter



Procedure for Johnson Counter:

1. Connection is made as shown in the logic diagram.
2. 4 bit Data is loaded into shift registers **D3, D2, D1 & D0.**
3. Keeping Mode control M=1, clock pulse are applied, the data present in **D3, D2, D1 & D0** will appear at **Q3 , Q2, Q1 & Q0 respectively.**
4. Keeping Mode control M=0, clock pulse are applied one by one and data arriving out serially at Q0 to Q3 is observed.
5. Connect the output Q3 to Serial input through a NOT GATE and keep applying clock pulses to realize Johnson counter.

Experiment 8

Mod-N Synchronous Up Counter

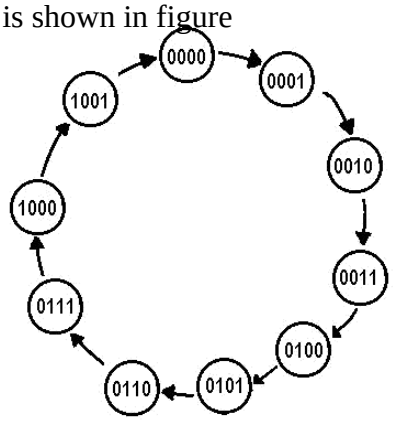
Aim: To design and test Mod-N (5, 6, 9, 10/decade) Synchronous Up counter using JK Flip Flop using trainer kit / simulator

Components Required:

Sl. No.	Components	Specification	Quantity
1	AND GATE	IC 7408	1 No
2	OR GATE	IC 7432	1 No
3	J K Flip Flop	IC -7476 N	2 No
4	Digital trainer kit & patch cables	–	1 No, 30 patch cables

Example: Decade Counter

A synchronous decade counter will count from zero to nine and repeat the sequence. The state diagram of this counter is shown in figure



State diagram of synchronous decade counter

Since there are ten states, four JK flip-flops are required. The truth tables of present and next state for the decade counter are shown in figure. Using the excitation table of JK flip-flop and the outputs of J and K are filled.

Procedure:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.

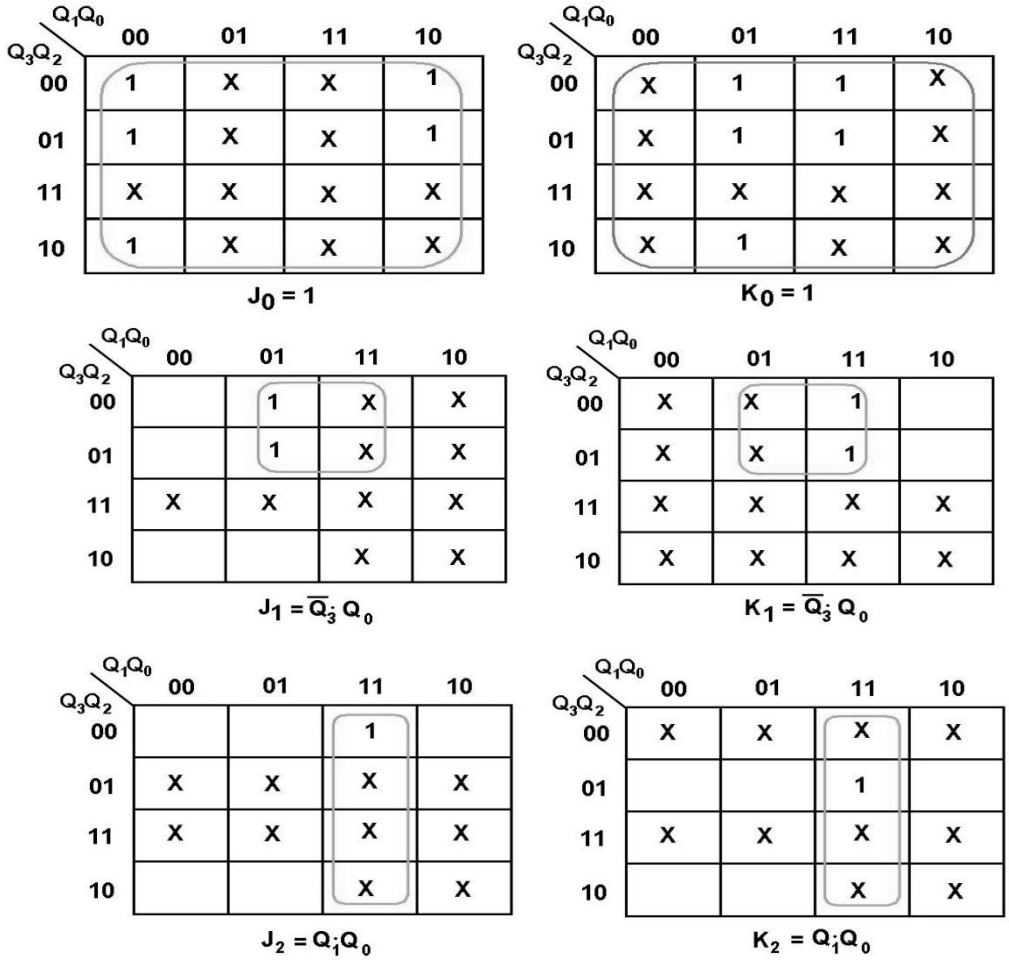
3. Make connections as shown in the circuit diagram.

4. Verify the Truth Table and observe the outputs.

Truth Table:

Present State				Next State				Output							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	0	0	0	0	X	1	0	X	0	X	X	1

The Karnaugh maps of the output J_0, K_0, J_1, K_1, J_2, K_2, J_3, and K_3 are shown below:



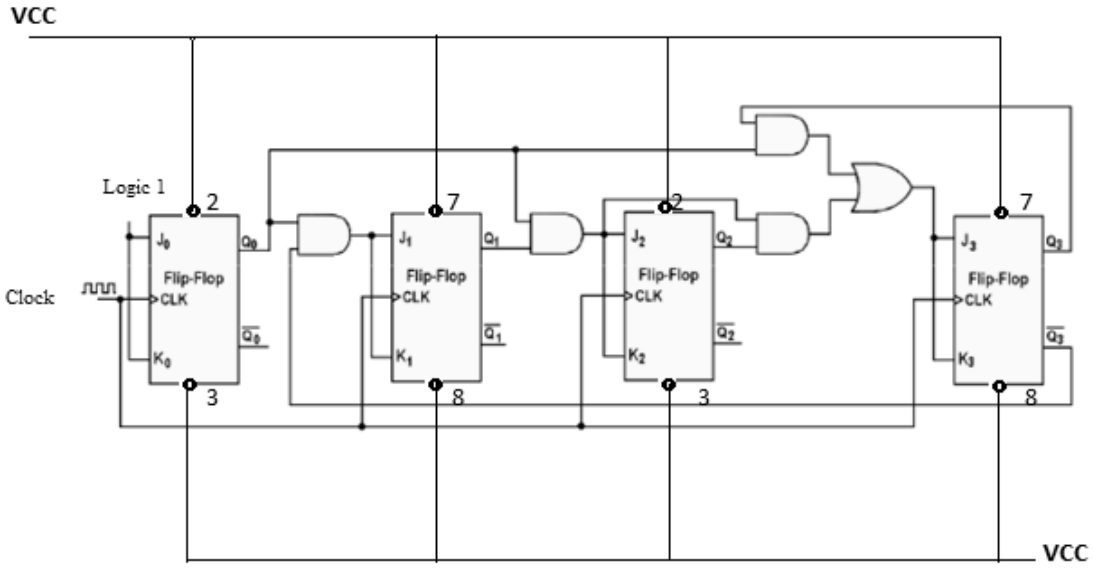
Q_3Q_2 \ Q_1Q_0	00	01	11	10
00				
01			1	
11	X	X	X	X
10	X	X	X	X

$J_3 = Q_3Q_0 + Q_2Q_1Q_0$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	X	X
10		1	X	X

$K_3 = Q_3Q_0 + Q_2Q_1Q_0$

Logic Diagram:



Inference:

PART B – Innovative Experiments (IE) / Open Ended Experiments

Design a 4/8-bit CPU using the LOGISIM simulator, for the following specifications.

1. Program Counter (Assume 256 program/code memory)
2. Instruction Register (Assume instruction size as 16 bit)
3. General Purpose Registers (RISC type-R0-R7): Use Harvard & Multiple Bus Architecture
4. ALU (to support 4-bit integer arithmetic operations & 4-bit logical operations)
5. Memory – 1024 ROM (to store instructions of size 16 bit) and 256 RAM (to store 4-bit data)
6. Implement the following instructions namely: MOV, ADD, SUB, LOAD, STORE, AND, XOR, NOT, BRANCH, BRANCH ON CONDITION.
7. Result to be displayed on 7-segment displays / reg tab of LOGISIM

1. Title:
2. Aim:
3. Team Members with USN:

4. Components Used in Design and their specifications:

5. Block Diagram

6. Description

7. Conclusion and Inferences