.

VISWAJYOTHI COLLEGE OF  **ENGINEERING AND TECHNOLOGY**

VISION

Moulding Engineers par Excellence with integrity, fairness and human values

MISSION

- We commit to develop the institution as a Centre of Excellence of International Standards.
- We guide our students in the attainment of intellectual and professional competence for successfully coping with the rapid advancements in technologies and the ever changing world of business, industry and services.
- We help each and every student in their personal growth into mature and responsible individuals.
- We strive to cultivate a sense of social and civic responsibility in our students, thus empowering them to serve the humanity.
- We promise to ensure a free environment where quest for the truth is encouraged.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

Moulding socially responsible and professionally competent Computer Engineers to adapt to the dynamic technological landscape.

MISSION

- Foster the principles and practices of computer science to empower life-long learning and build careers in software and hardware development.
- Impart value education to elevate students to be successful, ethical and effective problem-solvers to serve the needs of the industry, government, society and the scientific community.
- Promote industry interaction to pursue new technologies in Computer Science and provide excellent infrastructure to engage faculty and students in scholarly research activities.

.

PROGRAM EDUCATIONAL OBJECTIVES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Our Graduates

- Shall have creative and critical reasoning skills to solve technical problems ethically and responsibly to serve the society.
- Shall have competency to collaborate as a team member and team leader to address social, technical and engineering challenges.
- Shall have ability to contribute to the development of the next generation of information technology either through innovative research or through practice in a corporate setting.
- Shall have potential to build start-up companies with the foundations, knowledge and experience they acquired from undergraduate education.

PROGRAM OUTCOMES

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **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.
3. **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.
4. **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.
5. **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.

.

6. **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.
7. **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.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to ones own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **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

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

- Ability to integrate theory and practice to construct software systems of varying complexity.
- Able to Apply Computer Science skills, tools and mathematical techniques to analyze, design and model complex systems.
- Ability to design and manage small-scale projects to develop a career in a related industry.

Syllabus

.

CS234 : DIGITAL SYSTEMS LAB

Objectives

- To familiarize students with digital ICs, the building blocks of digital circuits.
- To provide students the opportunity to set up different types of digital circuits and study their behavior.

List of Exercises/Experiments

1. Familiarizations and verification of the truth tables of basic gates and universal gates.
2. Verification of Demorgan's laws for two variables.
3. Implementation of half adder and full adder circuits using logic gates.
4. Implementation of half subtractor and full subtractor circuits using logic gates.
5. Implementation of parallel adder circuit.
6. Realization of 4 bit adder/subtractor and BCD adder circuits using IC 7483.
7. Implementation of a 2 bit magnitude comparator circuit using logic gates.
8. Design and implementation of code convertor circuits
9. a) BCD to excess 3 code b) binary to gray code
10. Implementation of multiplexer and demultiplexer circuits using logic gates. Familiarization with various multiplexer and demultiplexer ICs.
11. Realization of combinational circuits using multiplexer/demultiplexer ICs.
12. Implementation of SR, D, JK, JK master slave and T flip flops using logic gates. Familiarization with IC 7474 and IC 7476.
13. Implementation of shift registers using flip flop Integrated Circuits.
14. Implementation of ring counter and Johnson counter using flip flop Integrated Circuits.
15. Realization of asynchronous counters using flip flop ICs.
16. Realization of synchronous counters using flip flop ICs. Familiarization with various counter Integrated Circuits.
17. Implementation of a BCD to 7 segment decoder and display.
18. Simulation of Half adder, Full adder using VHDL.

INDEX

.

.

DESIGN EXPERIMENTS		
1	STUDY OF 7490 BCD COUNTER	36
2	8:1 MUX USING TWO 4:1 MUX	37
OPEN ENDED EXPERIMENTS		
1	PRESETTABLE 4-BIT BINARY UP/DOWN COUNTER TO DESIGN IC 74193 AS A UP/DOWN COUNTER.	39
2	16: 1 MULTIPLEXER TO DESIGN AND SET UP 16:1 MUX USING TWO 8:1 MUX AND ONE OR GATE	40

.

EXPERIMENT NO: I

1. FAMILIARIZATIONS AND VERIFICATION OF THE TRUTH TABLES OF BASIC GATES AND UNIVERSAL GATES.

AIM:

To familiarize with digital IC trainer kit and to familiarize with logic gate IC packages and verify the truth tables of logic gates.

COMPONENTS REQUIRED:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	NAND GATE 2 I/P	IC 7400	1
5.	NOR GATE	IC 7402	1
6.	X-OR GATE	IC 7486	1
7.	NAND GATE 3 I/P	IC 7410	1
8.	AND GATE 3 I/P	IC 7411	1
9.	IC TRAINER KIT	-	1

THEORY:

Circuit that takes the logical decision and the process are called logic gates. Each gate has one or more input and only one output.OR, AND and NOT are basic gates. NAND and NOR are known as universal gates.

AND GATE:

.

The AND gate performs a logical multiplication commonly known as AND function. The output is high when both the inputs are high. The output is low level when any one of the inputs is low.

OR GATE:

The OR gate performs a logical addition commonly known as OR function. The output is high when any one of the inputs is high. The output is low level when both the inputs are low.

NOT GATE

:The NOT gate is called an inverter. The output is high when the input is low. The output is low when the input is high.

NAND GATE:

The NAND gate is a contraction of AND-NOT. The output is high when both inputs are low and any one of the input is low .The output is low level when both inputs are high.

NOR GATE:

The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.

X-OR GATE:

The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

LOGICAL SYMBOL, TRUTH TABLE & IC PINOUTS

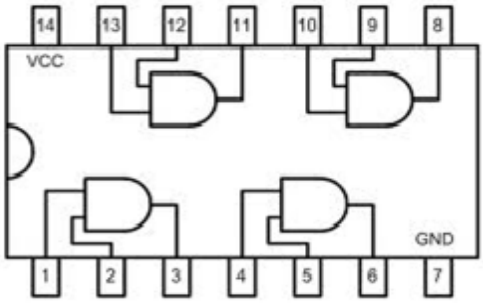
7408 QUAD 2 INPUT AND GATE:

LOGIC SYMBOL **PIN DIAGRAM**

.

TRUTH TABLE

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1



7408 Quad 2 Input AND

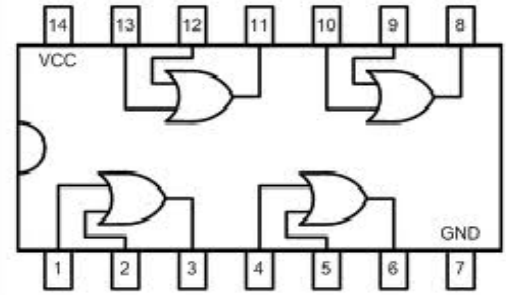
7432 QUAD 2 INPUT OR GATE:

LOGIC SYMBOL

TRUTH TABLE

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

PIN DIAGRAM



7432 Quad 2 Input OR

7404-HEX INVERTER GATES

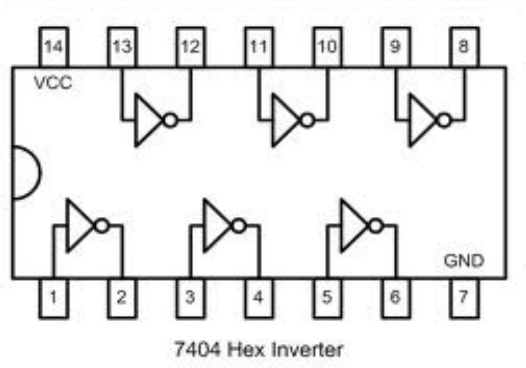
LOGIC SYMBOL

PIN DIAGRAM

.

TRUTH TABLE

A	$\overline{A}$
0	1
1	0



7404 Hex Inverter

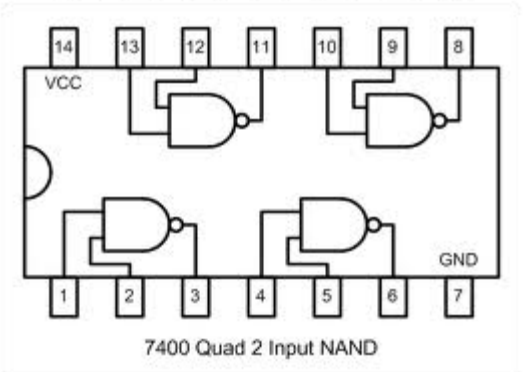
7400 QUAD 2-INPUT NAND GATE:

LOGIC SYMBOL

TRUTH TABLE

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

PIN DIAGRAM

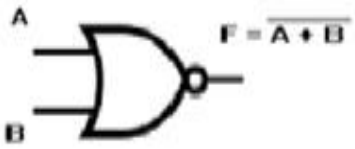


7400 Quad 2 Input NAND

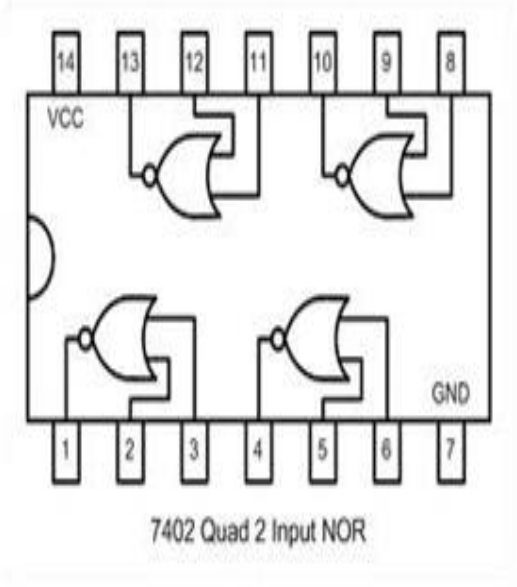
7402 QUAD 2- INPUT NOR GATE:

LOGIC SYMBOL **TRUTH TABLE** **PIN DIAGRAM**

.



INPUTS		OUTPUT
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0



7402 Quad 2 Input NOR

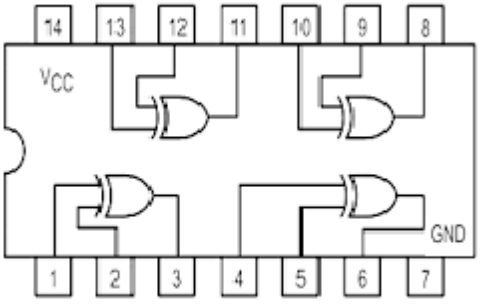
7486 QUAD 2-INPUT X-OR GATE

LOGIC SYMBOL

TRUTH TABLE

A	B	$\overline{A}B + A\overline{B}$
0	0	0
0	1	1
1	0	1
1	1	0

PIN DIAGRAM

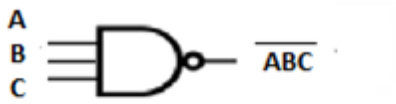


Quad 2 Input XOR gate

.

7410 TRIPLE 3-INPUT NAND GATE

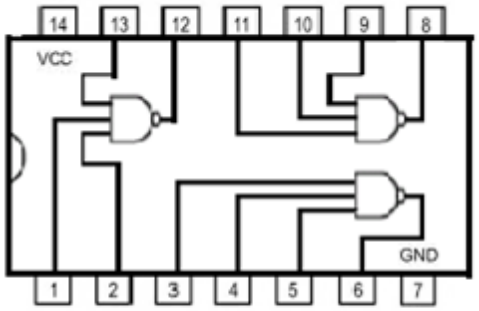
LOGIC SYMBOL



TRUTH TABLE

A	B	C	Z
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	0

PIN DIAGRAM

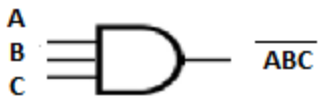


7410 Triple 3 Input NAND Gate

.

7411 TRIPLE 3-INPUT AND GATE :

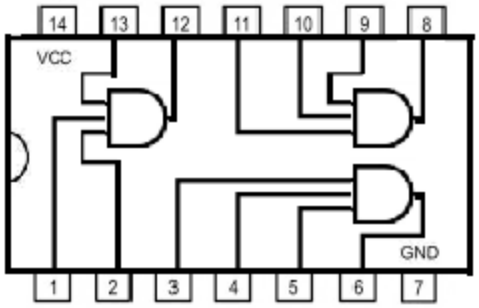
LOGIC SYMBOL



TRUTH TABLE

A	B	C	Z
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

PIN DIAGRAM



7411 Triple 3 Input AND Gate

PROCEDURE:

1. Test the IC using digital IC tester before conducting the experiment
2. Verify the dual in line package pin out the IC before feeding the input
3. Connections are given as per circuit diagram.
4. Logical inputs are given as per circuit diagram.
5. Observe the output and verify the truth table.

.

EXPERIMENT NO: II

IMPLEMENTATION OF HALF ADDER AND FULL ADDER CIRCUITS USING LOGIC GATES.

AIM:

To design and construct half adder, full adder circuits using logic gates and verify the truth table.

COMPONENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
5.	NAND GATE	IC 7400	2
6.	NOR GATE	IC 7402	2
7.	IC TRAINER KIT	-	1

.

THEORY

HALF ADDER:

A half adder has two inputs for the two bits to be added and two outputs one from the sum ‘ S’ and other from the carry ‘ c’ into the higher adder position. Above circuit is called as a carry signal from the addition of the less significant bits sum from the X-OR Gate the carry out from the AND gate.

FULL ADDER:

A full adder is a combinational circuit that forms the arithmetic sum of input; it consists of three inputs and two outputs. A full adder is useful to add three bits at a time but a half adder cannot do so. In full adder sum output will be taken from X-OR Gate, carry output will be taken from OR Gate.

.

PROCEDURE:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. Note down the output readings for half/full adder sum and the carry bit for different combinations of inputs.

EXPERIMENT NO: III

IMPLEMENTATION OF HALF SUBTRACTOR AND FULL SUBTRACTOR CIRCUITS USING LOGIC GATES

AIM:

To design and construct half subtractor, full subtractor circuits using basic logic gates and universal gates and verify using the truth table.

COMPONENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
5.	NAND GATE	IC 7400	2
6.	NOR GATE	IC 7402	2
7.	IC TRAINER KIT	-	1

.

THEORY

HALF SUBTRACTOR:

.

The half subtractor is constructed using X-OR and AND Gate. The half subtractor has two input and two outputs. The outputs are difference and borrow. The difference can be applied using X-OR Gate, borrow output can be implemented using an AND Gate and an inverter.

FULL SUBTRACTOR:

The full subtractor is a combination of X-OR, AND, OR, NOT Gates. In a full subtractor the logic circuit should have three inputs and two outputs. The two half subtractor put together gives a full subtractor .The first half subtractor will be C and A B. The output will be difference output of full subtractor. The expression AB assembles the borrow output of the half subtractor and the second term is the inverted difference output of first X-OR.

PROCEDURE:

1. Verify the gates.
2. Design circuit diagram
3. Make the connections as per the circuit diagram.
4. Switch on VCC and apply various combinations of input according to truth table.
5. Note down the output readings for half/full subtractor difference and the borrow bit for different combinations of inputs.

EXPERIMENT NO: IV
IMPLEMENTATION OF A 2 BIT MAGNITUDE COMPARATOR

CIRCUIT USING LOGIC GATES

AIM:

To design and implement 1bit and 2 bit magnitude comparator using logic gates.

COMPONENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	2
2.	X-OR GATE	IC 7486	1
3.	OR GATE	IC 7432	1
4.	NOT GATE	IC 7404	1

.

5.	IC TRAINER KIT	-	1

THEORY:

The comparison of two numbers is an operator that determine one number is greater than, less than (or) equal to the other number. A magnitude comparator is a combinational circuit that compares two numbers A and B and determine their relative magnitude. The outcome of the comparator is specified by three binary variables that indicate whether A>B, A=B (or) A<B.

One bit comparator:

A comparator used to compare two bits, i.e., two numbers each of single bit is called a single bit comparator. It consists of two inputs for allowing two single bit numbers and three outputs to generate less than, equal and greater than comparison outputs.

Two bit comparator:

A 2-bit comparator compares two binary numbers, each of two bits and produces their relation such as one number is equal or greater than or less than the other. The two-bit comparator has four inputs and three outputs.The first number A is designated as A = A1A0 and the second number is designated as B = B1B0. This comparator produces three outputs as G (G = 1 if A>B), E (E = 1, if A = B) and L (L = 1 if A<B).

PROCEDURE:

i. Design circuit diagram
ii. Give connections as per the circuit diagram.
iii. Give logical inputs as per the circuit diagram.
iv. Observe the output and verify the truth table.

.

EXPERIMENT NO: V

DESIGN AND IMPLEMENTATION OF CODE CONVERTOR CIRCUITS

AIM:

To design and realize the following using logic gates
a) BCD to Excess- 3 Code.
b) Binary to Gray Code.

COMPONENTS REQUIRED:

Sl. No.	Component	Specification	Quantity
1.	IC Trainer kit		
2.	IC's	7486	1
		7408	1
		7432	1
		7404	1
		7411	1
3.	Connecting wires		

THEORY:

BCD to excess 3 code:

Code converter is a combinational circuit that translates the input code word into a new corresponding word. The excess-3 code digit is obtained by adding three to the corresponding BCD digit. To Construct a BCD-to-excess-3-code converter with a 4-bit adder feed BCD code to the 4-bit adder as the first operand and then feed constant 3 as the second operand. The output is the corresponding excess-3 code. To make it work as a excess-3 to BCD converter, we feed excess-3 code as the first operand and then feed 2's complement of 3 as the second operand. The output is the BCD code.

Binary to gray code:

In this code while traversing from one step to another step, only one bit in the code group changes. In case of Gray Code two adjacent code numbers differs from each other by only one bit. The M.S.B. of the gray code will be exactly equal to the first bit of the given binary number. Now the second bit of the code will be exclusive-or of the first and second bit of the given binary number and the third bit of gray code will be equal to the exclusive-or of the second and third bit of the given binary number and so on.

.

PROCEDURE:

i. Check all the components for their working.
ii. Insert the appropriate IC into the IC base.
iii. Design circuit diagram
iv. Make connections as per the circuit diagram.
v. Apply BCD code as first operand(A) and binary 3 as second operand(B) and cin=0 for
    a. Realizing BCD-to-Excess-3-code:
vi. Apply Excess-3-code code as first operand(A) and binary 3 as second operand(B) and
    a. Cin=1 for realizing Excess-3-code to BCD.
vii. Verify the Truth Table and observe the outputs.

EXPERIMENT NO: VI

(A) IMPLEMENTATION OF MULTIPLEXER AND DEMULTIPLEXER CIRCUITS USING LOGIC GATES

AIM:

To design and implement multiplexer and demultiplexer using logic gates.

COMPONENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	3 I/P AND GATE	IC 7411	2
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	IC TRAINER KIT	-	1

THEORY:

MULTIPLEXER:

Multiplexer means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input line and n selection lines whose bit combination determine which input is selected.

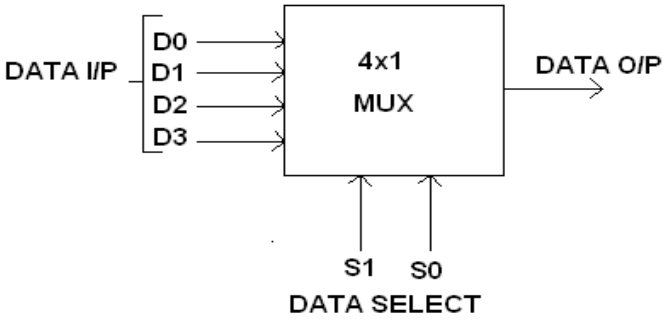
.

DEMULTIPLEXER:

The function of Demultiplexer is in contrast to multiplexer function. It takes information from one line and distributes it to a given number of output lines. For this reason, the demultiplexer is also known as a data distributor. Decoder can also be used as demultiplexer. In the 1: 4 demultiplexer circuit, the data input line goes to all of the AND gates. The data select lines enable only one gate at a time and the data on the data input line will pass through the selected gate to the associated data output line.

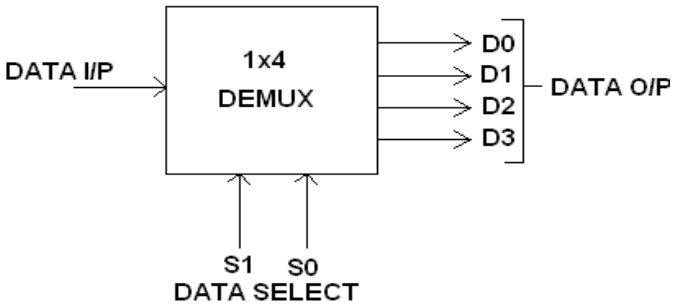
4:1 MULTIPLEXER

BLOCK DIAGRAM



1:4 DEMULTIPLEXER

BLOCK DIAGRAM



.

PROCEDURE:

(i) Design circuit diagram
(ii) Give connections as per the circuit diagram.
(iii) Give logical inputs as per the circuit diagram.
(iv) Observe the output and verify the truth table.

(B) FAMILIARIZATION OF MULTIPLEXER IC.

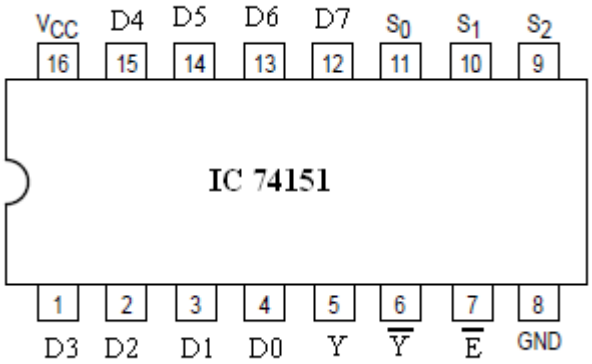
AIM:

- To familiarize with IC 74151 (8X1 MUX)
- **Components Required:**

Sl. No.	Component	Specification	Quantity
1.	IC Trainer kit		
2.	IC's	74151	1
3.	Connecting wires		

-

Pin diagram of IC 74151



.

TRUTH TABLE

Select Data Inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

PROCEDURE:

(i) Check the IC's
(ii) Set up the logic diagram on bread board
(iii) Give the power supply and apply the inputs.
(iv) Verify the truth tables using the outputs obtained

EXPERIMENT NO: VII

REALIZATION OF COMBINATIONAL CIRCUIT USING MULTIPLEXER IC

AIM:

To realize the following function using IC 74151

$F = \sum m (1, 3, 4, 11, 12, 13, 14, 15)$

COMPONENTS REQUIRED:

Sl. No.	Component	Specification	Quantity
1.	IC Trainer kit		
2.	IC's	7486	1
		7408	1
		7432	1
		7404	1
		7411	1
3.	Connecting wires		

.

THEORY:

COMBINATIONAL CIRCUITS USING MULTIPLEXER ICS

Any Boolean function of n-variables can be implemented using a multiplexer with n-1 selection lines. For that, the first n-1 input variables of the function will be connected to the selection lines and the nth input variable is evaluated according to the value of the minterms of the function. These evaluated values are connected to the data input lines.

IMPLEMENTATION OF BOOLEAN FUNCTION USING 8X1 MUX IC 74151

F(A,B,C,D)=∑(1,3,4,11,12,13,14,15)

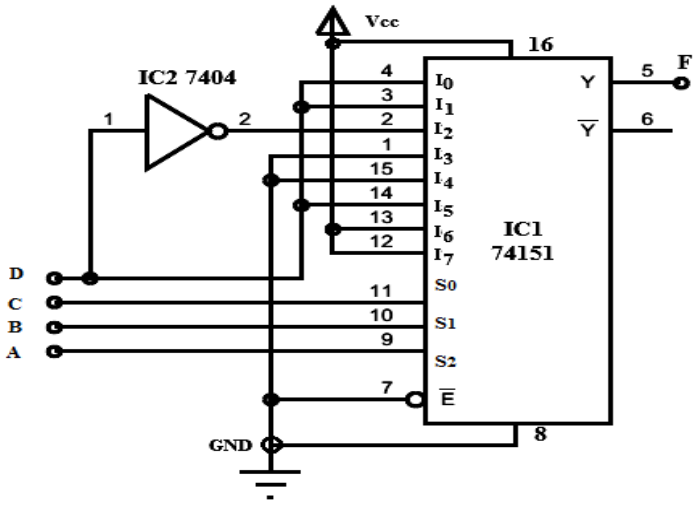
TRUTHTABLE

INPUTS				OUTPUT	
A	B	C	D	F	
0	0	0	0	0	F=D
0	0	0	1	1	
0	0	1	0	0	F=D
0	0	1	1	1	
0	1	0	0	1	F=D’
0	1	0	1	0	
0	1	1	0	0	F=0
0	1	1	1	0	
1	0	0	0	0	F-0
1	0	0	1	0	
1	0	1	0	0	F=D
1	0	1	1	1	
1	1	0	0	1	F=1
1	1	0	1	1	
1	1	1	0	1	F=1
1	1	1	1	1	

A,B,C are given as select lines & D is given as data input

.

CIRCUIT DIAGRAM



PROCEDURE:

(i) Check the IC's
(ii) Set up the logic diagram on bread board
(iii) Give the power supply and apply the inputs.
(iv) Verify the truth tables using the outputs obtained

EXPERIMENT NO: VIII

(1) IMPLEMENTATION OF SR, D, JK, JK MASTER SLAVE AND T FLIP FLOPS USING LOGIC GATES.

(2) FAMILIARIZATION WITH IC 7474 AND IC 7476.

AIM:

1. To implement the following flip flops using NAND gate
   i. SR FF
   ii. D FF
   iii. JK FF
   iv. MS JK FF
   v. T FF
2. To familiarize with IC 7474 and IC 7476

.

COMPONENTS REQUIRED:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	IC's	IC 7404	1
2.		IC 7400	2
3.		IC 7474, IC 7476	1
4.	IC TRAINER KIT	-	1

A

THEORY:

A flip-flop is a device very like a latch in that it is a bistable multivibrator, having two states and a feedback path that allows it to store a bit of information. The difference between a latch and a flip-flop is that a latch is asynchronous, and the outputs can change as soon as the inputs do (or at least after a small propagation delay). A flip-flop, on the other hand, is *edge-triggered* and only changes state when a control signal goes from high to low or low to high. The four main types of flip-flop are : SR, JK, D, and T.

SR FLIPFLOP:

RS flip flop is a basic flip flop where R stands for reset and S stands for set. So S-R flip flop we can call Set-Reset flip flop. A RS flip flop made with a R-S latch and triggering circuit. The outputs change states as per the inputs only on the occurrence of a clock pulse. The clocked flip-flop could be a level-triggered one or an edge-triggered .

D FLIPFLOP:

The D flip-flop is the edge-triggered variant of the transparent latch. On the rising edge of the clock, the output is given the value of the *D* input *at that moment*. The output can only change at the clock edge, and if the input changes at other times, the output will be unaffected.D flip-flops are by far the most common type of flip-flops and some are made entirely from D flip-flops. They are also commonly used for shift-registers and input synchronisation

JK FLIPFLOP:

.

The JK flip-flop is a simple enhancement of the SR flip-flop where the state J=K=1 is not forbidden. It works just like a SR FF where J is serving as set input and K serving as reset. The only difference is that for the formerly "forbidden" combination J=K=1 this flip-flop now performs an action: it inverts its state. As the behavior of the JK flip-flop is completely predictable under all conditions, this is the preferred type of flip-flop for most logic circuit designs. But there is still a problem i.e. both the outputs are same when one tests the circuit practically. This is because of the internal toggling on every propagation elapse completion. The main remeady is going for master-slave jk flip-flop,this ff overrides the self(internal) recurring toggling through the pulsed clocking feature incorporated.

T FLIPFLOP:

A **T flip-flop** is a device which swaps or "toggles" state every time it is triggered if the *T* input is asserted, otherwise it holds the current output.

PROCEDURE:

(i) Design circuits
(ii) Give connections as per the circuit diagram.
(iii) Logical inputs are given as per circuit diagram.
(iv) Observe the output and verify the truth table.

EXPERIMENT NO: IX

SHIFT REGISTERS

AIM

To design and setup

a) Serial IN serial OUT shift register
b) Serial IN parallel OUT shift register
c) Parallel IN serial OUT shift register

COMPONENTS REQUIRED

Sl.No.	COMPONENTS	SPECIFICATION	QUANTITY
1.	Hex inverter gate	IC 7404	1
2.	Dual D flip flop	IC 7474	2
3.	Quad two input NANDgate	IC 7408	2
4.	Quad two input OR gate	IC 7432	1

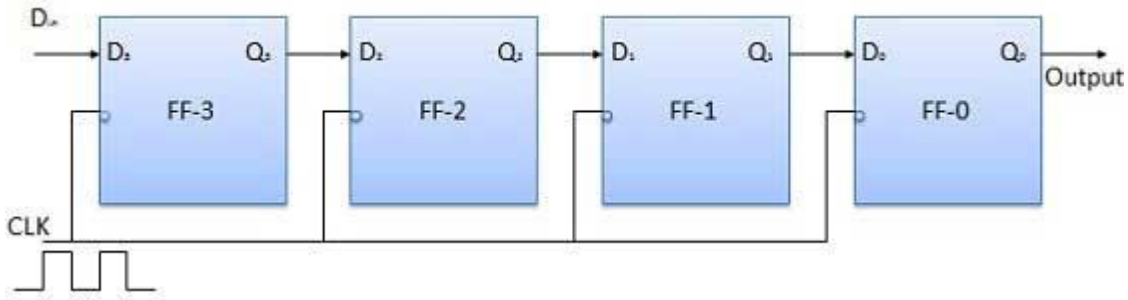
.

5.	IC trainer kit		1

.

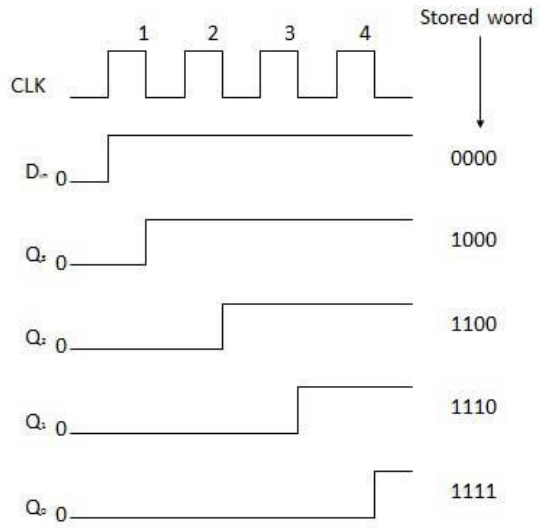
CIRCUIT DIAGRAM

a) Serial IN serial OUT shift register



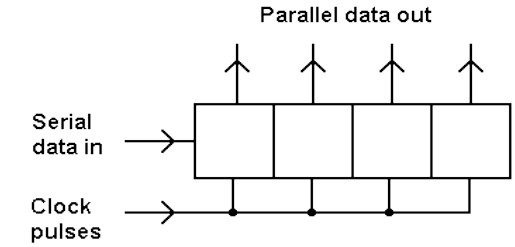
	CLK	$D_3 = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

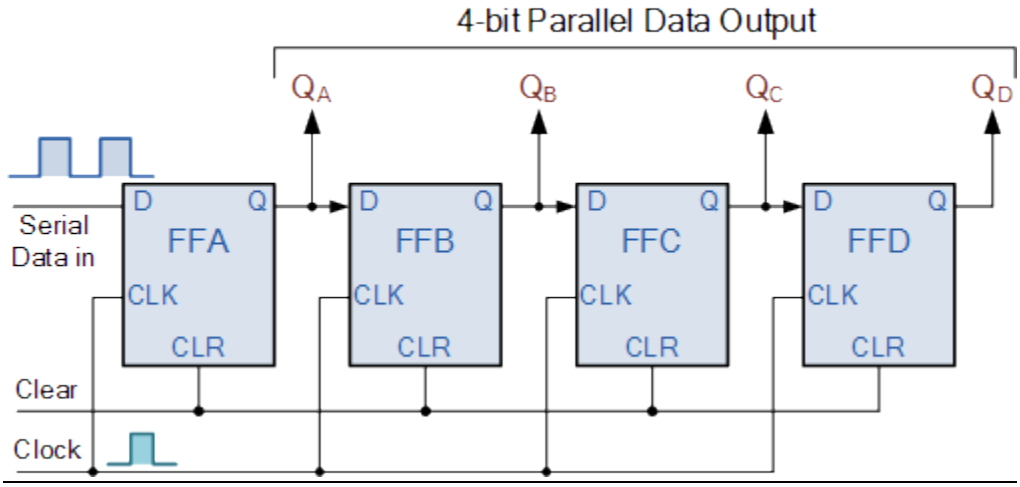
→ Direction of data travel



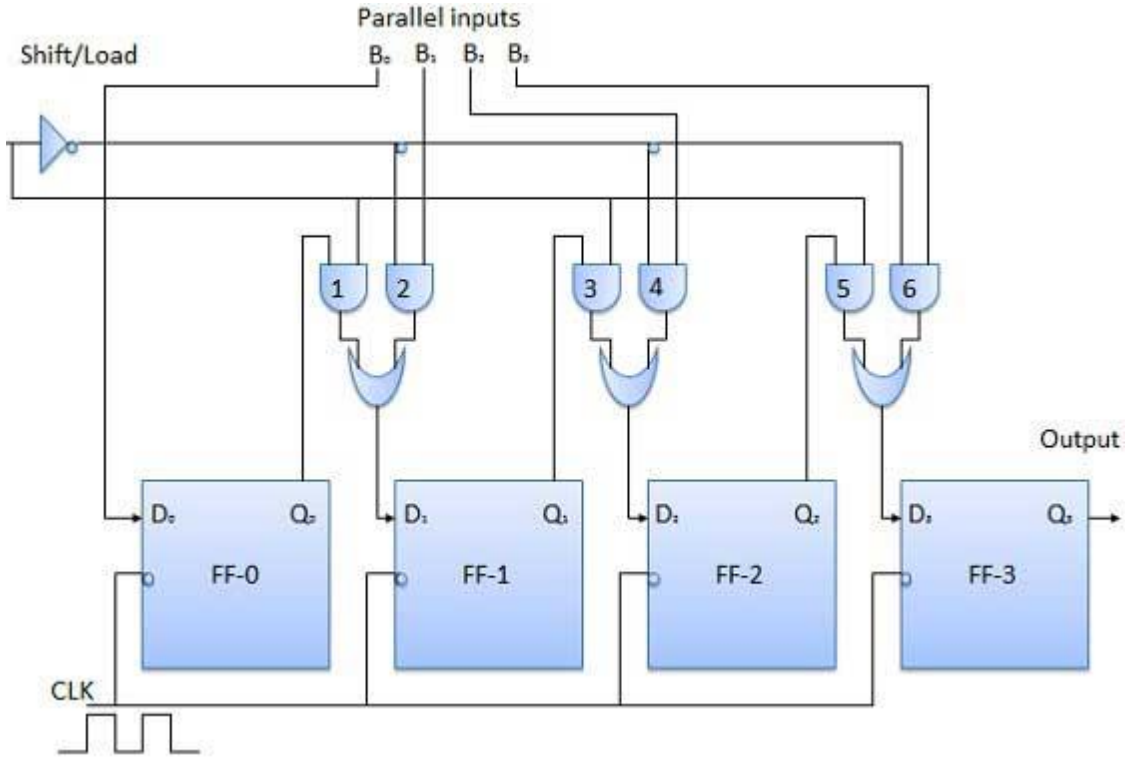
b) Serial IN parallel OUT shift register

.





c) Parallel IN serial OUT shift register



.

THEORY

i. Serial IN parallel OUT shift registers

In this type of register, the data bits are entered into the register serially, but the data stored in the register is shifted out in parallel form. Once the data bits are stored, each bit appears on its respective output line and all bits are available simultaneously, rather than on a bit-by-bit basis as with the serial output.

ii. Serial IN serial OUT shift registers

This type of shift register accepts data serially, i.e. one bit at a time, and also outputs data serially. With four stages, i.e. four FF_s, the register can store upto four bits of data. Serial data is applied at the D input of the first FF. The Q output of the first FF is connected to the D input of the second FF, the Q output of the second FF is connected to the D input of the third FF and the Q output of the third FF is connected to the D input of the fourth FF. The data is outputted from the Q terminal of the FF.

When serial data is transferred into a register, each new bit is clocked into the first FF at the positive-going edge of each clock pulse. The bit that was previously stored by the first FF is transferred to the second FF. The bit that was stored by the second FF is transferred to the third FF, and so on. The bit that was stored by the last FF is shifted out.

iii. Parallel IN serial OUT shift registers

For a parallel-in, serial-out, shift register, the data bits are entered simultaneously into their respective stages on parallel lines, rather than on a bit-by-bit basis on one line as with serial data inputs, but the data bits are transferred out of the register serially , i.e. on a bit-by-bit basis over a single line.

iv. There are four data lines A, B, C and D through which the data is entered into the register in parallel form. The signal Shift/ LOAD allows (a) the data to be entered in parallel form into the register and (b) the data to be shifted out serially from terminal Q_4. The OR gate allows either the normal shifting operation or the parallel data entry depending on which AND gates are enabled by the level on the Shift/LOAD input.

PROCEDURE

1. Check all the ICs using a digital IC tester.
2. Design circuit diagram
3. Connect the components as per the circuit diagram.
4. Provide the inputs for the circuit.
5. Observe the output obtained using LEDs or seven segment display.
6. Repeat the above steps for each case.

.

EXPERIMENT NO: X

RING COUNTER & JOHNSON (TWISTED RING) COUNTER

AIM:

To design and set up 4-bit ring counter and Johnson counter using flipflops.

COMPONENTS REQUIRED:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	JK FLIPFLOP	IC 7476	2
2.	IC TRAINER KIT	-	1

THEORY:

Ring counter and Johnson counter are two important shift counters.

RING COUNTER:

It is a shift counter shifting the sequence in a cyclic nature. A ring counter constructed using JK flip flops by connecting Q and ~Q outputs from one flip flop to the J and K inputs of the next flip-flops respectively. The output of the final flip flop is connected to the input of the first flip-flop. To start the counter, the flip-flops should be initialized using preset and clear inputs. For each clock pulse the number gets shifted like in a ring. Ring counter is called a divide by N counter where N is the number of flip-flops.. Ring counter divides the clock frequency by N. Ring counters are used for switching devices or lamps in a particular order ,rotate the stepper motor etc.

JOHNSON COUNTER:

A ring counter can be converted to a Johnson counter by connecting Q and ~Q outputs of the last flipflop to the J and K inputs of the first flip flop respectively. The mod number of the counter is double of that of the ring counter. Johnson counter also called twisted ring counter or divide by 2n counter.

PROCEDURE:

1. Design circuit diagram

.

2. Set up the ring counter to shift 1000 as per the diagram. Set first flip flop output to logic 1 by applying logic 0 to its preset input. After presetting the output , inactivate the preset. Clear the remaining Q outputs to logic 0 by applying logic 0 to clear pins. After clearing the outputs inactivate clear pins.

2. Apply clock pulse and observe the outputs. Repeat the experiment to shift other binary pattern.

3. Set up the Johnson counter. Clear all Q outputs by connecting clear pins to logic 0.After clearing deactivate clear pins. Apply clock pulses and verify its operations.

EXPERIMENT NO: XI

ASYNCHRONOUS COUNTER

AIM:

To design and verify 3-bit asynchronous up counter and down counter and mod 5aynchronous counter

COMPONENTS REQUIRED:

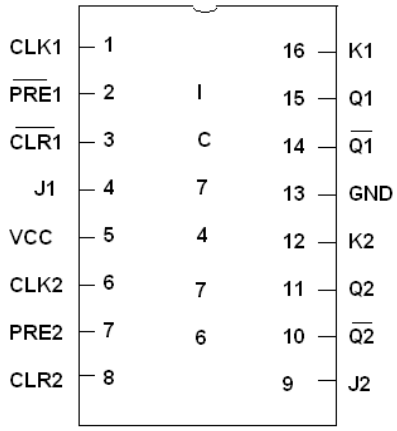
Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	JK FLIP FLOP	IC 7476	2
2.	NAND GATE	IC 7400	1
3.	IC TRAINER KIT	-	1

THEORY:

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appears as counter output. This is the main difference between a register and a counter. There are two types of counter, synchronous and asynchronous. In synchronous common clock is given to all flip flop and in asynchronous first flip flop is clocked by external pulse and then each successive flip flop is clocked by Q or Q output of previous stage. A soon the clock of second stage is triggered by output of first stage. Because of inherent propagation delay time all flip flops are not activated at same time which results in asynchronous operation.

PIN DIAGRAM FOR IC 7476:

.



PROCEDURE:

(i) Design the circuit

(ii) Connections are given as per circuit diagram.

(iii) Logical inputs are given as per circuit diagram.

(iv) Observe the output and verify the truth table.

EXPERIMENT NO: XII

DESIGN AND IMPLEMENTATION OF SYNCHRONOUS UP COUNTER & DOWN COUNTER

AIM:

To design and verify 4-bit synchronous up counter and down counter

COMPONENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	JK FLIP FLOP	IC 7476	2
2.	2 I/P AND GATE	IC 7400	1
3.	IC TRAINER KIT	-	1

THEORY:

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. An up/down counter is one that is

.

capable of progressing in increasing order or decreasing order through a certain sequence. An up/down counter is also called bidirectional counter. Usually up/down operation of the counter is controlled by up/down signal. When this signal is high counter goes through up sequence and when up/down signal is low counter follows reverse sequence.

EXCITATION TABLE OF JK FLIPFLOP

PRESENT STATE	NEXT STATE	J	K
Q	Q_{t+1}		
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

PROCEDURE:

(i) Design the circuit

(ii) Give Connections as per the circuit diagram.

(iii) Logical inputs are given as per circuit diagram.

(iv) Observe the output and verify the truth table.

.

ADVANCED EXPERIMENTS

EXPERIMENT NO: I

Implementation of 4 bit adder/subtractor circuit using IC 7483

AIM:

To implement a 4 bit adder/subtractor circuit using IC 7483

COMPONENTS REQUIRED:

Sl.No	Components	Specification	Quantity
1	IC Trainer Kit	-	1
2	IC Tester	-	1
3	Connecting Wires	-	-
4	IC7486	X-OR GATE	1
6	IC7483	Parallel Adder	1

THEORY:

Design the circuit and set up the connections as per the circuit diagram. To add the nibbles SUB of be made 0. To sub $B_3B_2B_1B_0$ from $A_3A_2A_1A_0$ SUB = 1. XOR gates functions as controlled inverter. When SUB=1 $B_3B_2B_1B_0$ is complemented. Now A3A2A1A0 complemented version of $B_3B_2B_1B_0$ and 1 at Cin pin are added together.

PROCEDURE:

1. Check whether the ICs are working properly or not using the digital IC tester.
2. Design the circuit
3. Make connections as shown in the circuit diagram.
4. Provide Vcc and GND.
5. Verify the output using truth table.

.

EXPERIMENT NO: II

REALIZATION of BCD adder circuit using IC 7483

AIM:

To realize BCD adder using gates.

COMPONENTS REQUIRED:

Sl.No	Components	Specification	Quantity
1	IC Trainer Kit	-	1
2	IC Tester	-	1
3	Connecting Wires	-	-
4	IC7408	Quad 2-Input AND Gate	1
5	IC7432	Quad 2-Input OR Gate	1
6	IC7483	Parallel Adder	3

THEORY:

In computing and electronic systems, binary-coded decimal(BCD) is a class of binary encodings of decimal numbers where each decimal digit is represented by a fixed number of bits, usually four or eight. Special bit patterns are sometimes used for a sign or for other indications (e.g., error or overflow).

A BCD adder add two 4-bit BCD code groups using straight binary addition. If the sum is greater than 9, 6 is added to the result and a carry is generated. It is a 74LS8316 IC. If two BCD code groups $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ are applied to 4 bit parallel adder, the output will be $S_4S_3S_1S_0$, where S_4 is the carry.

PROCEDURE:

a. Check whether the ICs are working properly or not using the digital IC tester.
b. Design the circuit
c. Make connections as shown in the circuit diagram.
d. Provide Vcc and GND.
e. Verify the output using truth table.

.

DESIGN EXPERIMENTS

EXPERIMENT: 1 STUDY OF 7490 BCD COUNTER

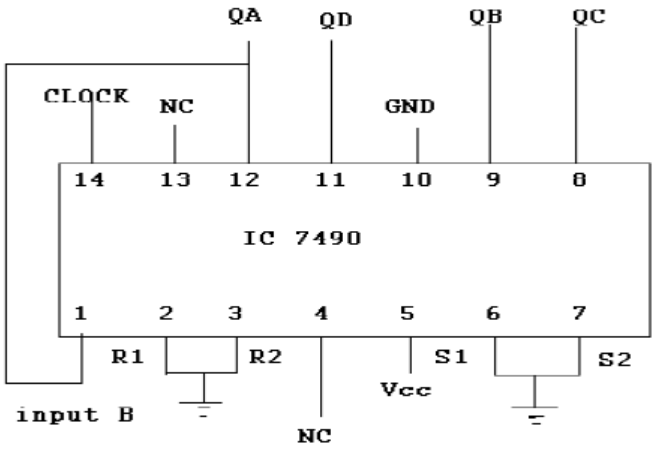
AIM:

To design IC 7490 as a decade counter with BCD count sequence

COMPONENTS REQUIRED:

IC 7490, Patch Cords & IC Trainer Kit

DECADE COUNTER:



TRUTH TABLE:

Q_D	Q_C	Q_B	Q_A
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
0	0	0	0

.

PROCEDURE:

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

EXPERIMENT:2

8:1 MUX using two 4:1 MUX

Aim: To design the 8:1 MUX using two 4:1 MUX .

Objective:

- To get familiar with the concept of multiplexing .
- To get familiar with MSI (medium scale integration) technology.
- To get familiar with the expansion of multiplexing using standard IC packages.

Apparatus Required:

- Prototyping board (breadboard)
- DC Power Supply 5V Battery
- Light Emitting Diode (LED)
- Digital ICs:74153:Dual 4:1 MUX
  7432: Quad 2 input OR gate
  7404 : Hex inverter
- Connecting Wires

Theory:

Multiplexer: A data selector, more commonly called a Multiplexer, shortened to "Mux" or "MPX", is combinational logic switching devices that operate like a very fast acting multiple position rotary switches. They connect or control, multiple input lines called "channels" consisting of either 2, 4, 8 or 16 individual inputs, one at a time to an output. Then the job of a multiplexer is to allow multiple signals to share a single common output. A single multiplexer as Ic is 4:1, i.e., it can handle a maximum of 4 inputs. When the number of inputs is more then 4, a multiplexer tree can be used ,also known as multiplexer stack.

Procedure:

1. Design the circuit
2. Collect the components necessary to accomplish this experiment.
3. Plug the IC chip into the breadboard.
4. Connect the supply voltage and ground lines to the chips. PIN7 = Ground and PIN14 = +5V.

.

5. Make connections as shown in the respective circuit diagram.
6. Connect the inputs of the gate to the input switches of the LED.
7. Connect the output of the gate to the output LEDs.
8. Once all connections have been done, turn on the power switch of the breadboard
9. 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.

.

OPEN-ENDED EXPERIMENTS

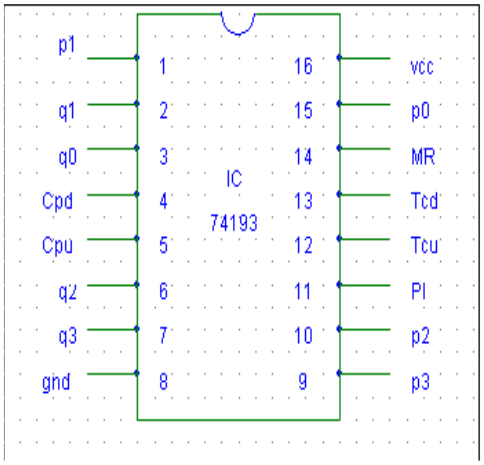
EXPERIMENT: 1

PRESETTABLE 4-BIT BINARY UP/DOWN COUNTER

AIM:
To design IC 74193 as a up/down counter.

COMPONENTS REQUIRED:
IC 74193, Patch Cords & IC Trainer Kit

PIN DETAILS OF IC 74193



1. P1,P2,P3 and P0 are parallel data inputs
2. Q0,Q1,Q2 and Q3 are flip-flop outputs
3. MR: Asynchronous master reset
4. PL: Asynchronous parallel load(active low) input
5. TCd : Terminal count down output
6. TCu : Terminal count up output

a) Design up counter for preset value 0010 and N=10

b) Design of down counter for preset value 1011 and N=10

.

EXPERIMENT: 2

16: 1 MULTIPLEXER

AIM:

To design and set up 16:1 MUX using two 8:1 MUX and one OR gate.

COMPONENTS REQUIRED:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	8:1 MUX	IC 74151	2
2.	OR Gate	IC 7432	1
3.	IC TRAINER KIT	-	1

THEORY

Here, to select one of the 16 inputs, four select line $S_3S_2S_1S_0$ are required. among the four select lines, the least significant three select lines S_2 S_1S_0 are connected with three select inputs of both the multiplexers ICs. The most significant select line is S_3 is connected directly to the Enable input of MUX1 while the same is connected through an inverter to the Enable input of MUX2. When S_3 = 0, MUX1 is selected and the inputs are multiplexed to the output Z and MUX2 is disabled. When S_3 = 1, vice-versa will occur.

.

VIVA QUESTIONS

1. Why NAND & NOR gates are called universal gates?
2. Realize the EX – OR gates using minimum number of NAND gates.
3. Give the truth table for EX-NOR and realize using NAND gates?
4. What are the logic low and High levels of TTL IC's and CMOS IC's?
5. Compare TTL logic family with CMOS family?
6. Which logic family is fastest and which has low power dissipation?
7. What are the different methods to obtain minimal expression?
8. What is a Min term and Max term
9. What is meant by canonical representation?
10. What is meant by canonical representation?
11. What is K-map? Why is it used?
12. What are universal gates?) State the difference between SOP and POS.
13. What is a half adder?
14. What is a full adder?
15. What are the applications of adders?
16. What is a half subtractor?
17. What is a full subtractor?
18. What are the applications of subtractors?
19. Obtain the minimal expression for above circuits.
20. Realize a full adder using two half adders
21. Realize a full subtractors using two half subtractors.
22. What is the internal structure of 7483 IC?
23. What do you mean by code conversion?
24. What are the applications of code conversion?
25. How do you realize a subtractor using full adder?
26. What is a ripple Adder? What are its disadvantages?
27. What is a multiplexer?
28. What is a de-multiplexer?
29. What are the applications of multiplexer and de-multiplexer?
30. Derive the Boolean expression for multiplexer and de-multiplexer.
31. How do you realize a given function using multiplexer
32. What is the difference between multiplexer & demultiplexer?
33. In 2n to 1 multiplexer how many selection lines are there?
34. How to get higher order multiplexers?
35. Implement an 8:1 mux using 4:1 muxes?
36. What is a comparator?
37. What are the applications of comparator?
38. Derive the Boolean expressions of one bit comparator and two bit comparators.
39. How do you realize a higher magnitude comparator using lower bit comparator

.

40. Design a 2 bit comparator using a single Logic gates?
41. What are the applications of decoder?
42. What is a priority encoder?
43. What is the role of an encoder in communication?
44. What is the advantage of using an encoder?
45. What is the difference between decoder & encoder?
46. For n- 2n decoder how many i/p lines & how many o/p lines?
47. What are the different codes & their applications?
48. What are code converters?
49. Using 3:8 decoder and associated logic, implement a full adder?
50. Implement a full subtractor using IC 74138?
51. What is the difference between decoder and de-mux?
52. What are the different types of LEDs?
53. Draw the internal circuit diagram of an LED.
54. What are the applications of LEDs?
55. What is the difference between Flip-Flop & latch?
56. Give examples for synchronous & asynchronous inputs?
57. What are the applications of different Flip-Flops?
58. What is the advantage of Edge triggering over level triggering?
59. What is the relation between propagation delay & clock frequency of flip-flop?
60. What is race around in flip-flop & how to overcome it?
61. Convert the J K Flip-Flop into D flip-flop and T flip-flop?
62. List the functions of asynchronous inputs?
63. What is the necessity for sequence generation?
64. What are PISO, SIPO, and SISO with respect to shift register?
65. Differentiate between serial data & parallel data
66. What is the significance of Mode control bit?
67. What is a ring counter?
68. What is a Johnson counter?
69. How many Flip-flops are present in IC 7495?
70. What is an asynchronous counter?
71. How is it different from a synchronous counter?
72. Realize asynchronous counter using T flip-flop
73. What are synchronous counters?
74. What are the advantages of synchronous counters?
75. What is an excitation table?
76. Write the excitation table for D, T FF
77. Design mod-5 synchronous counter using T FF
78. What is a decade counter?
79. What do you mean by a ripple counter?
80. Explain the design of Modulo-N counter ($N \geq 9$) using IC 7490

.

UNIVERSITY SAMPLE QUESTIONS

1. Realize the given logic function using not more than five 2-input NAND gates
2. Convert the following expression into SOP&POS:- and realize using universal gates
3. Implement the function F (A, B, C) =∑m (0, 1, 4).
4. Design a circuit with four inputs and one output, such that the output goes to '1'whenever two or more of inputs are '1'. For other cases the output remains at '0'.
5. Difference between half adder and a full adder
6. Design full adder cum Subtractor using mode control
7. Design a full adder using two half adder and suitable gate
8. Design a two bit digital comparator
9. Which logic is used in single bit digital comparator
10. Design 2-bit odd parity generator and checker.
11. Design 3-bit binary to gray code converter
12. Design 5-bit gray code to binary
13. Implement the function F (A, B, C) =∑m (0, 1, 3, 4) using multiplexer.
14. Design an 8:1 multiplexer
15. Design an 1 to 16 demultiplexer
16. Design a full adder using multiplexer
17. Design a bidirectional shift register using mode control.
18. 2.Design a ring counter/Johnson counter with mode control using JK flip flop
19. Design an up counter using T flip flop
20. Design an up counter using D flip flop
21. Design an asynchronous UP/DOWN counter using mode control.
22. Design a Modulo 12 (MOD 12) counter
23. Design an asynchronous decade down counter
24. Design BCD up counter
25. Design half adder cum Subtractor using mode control.

.

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY,
VAZHAKULAM P.O, MUVATTUPUZHA, ERNAKULAM - **686670**

COURSE DATA SHEET - DIGITAL SYSTEMS LAB

COURSE OBJECTIVES:

1	To provide an introduction to Logic Systems Design thereby giving a hands on experience on working with digital ICS ,which enable the study Computer System Architecture.

COURSE OUTCOMES:

Number of Course Outcomes expected to be around six.

C216.1	Understand the behaviour of digital IC's, IC trainer kit and IC tester.
C216.2	Design and implementation of combinational logic circuits and compare the output with truth table
C216.3	Examine and implement different Flip flop IC's
C216.4	Design and implement different shift registers using D Flipflop and compare the output with truth table
C216.5	Memorise the fundamentals of counters and design various types of counters

CORELATION BETWEEN COURSE OUTCOMES AND PROGRAMME OUTCOMES

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C216.1	2	-	2	1	3	2	1	-	1	-	1	-
C216.2	1	2	2	1	1	1	1	-	1	-	1	-
C216.3	1	2	2	1	1	1	1	-	1	-	1	-
C216.4	2	2	2	1	1	1	1	-	1	-	1	-
C216.5	1	2	2	1	1	1	1	-	1	-	1	-
C216	1.4	1.6	2	1	1.4	1.2	1	-	1	-	1	-

CORELATION BETWEEN COURSE OUTCOMES AND PROGRAMME SPECIFIC OUTCOMES

SNO	PSO 1	PSO 2	PSO3
C216.1	-	2	-
C216.2	-	1	-
C216.3	-	1	-
C216.4	-	1	-
C216.5	-	1	-
C216	-	1.33	-