Lab manual – DIGITAL ELECTRONICS LAB (3048)

Lab manual – DIGITAL ELECTRONICS LAB (3048)

Department of Electronics & Communication Engineering
GOVERNMENT POLYTECHNIC COLLEGE, PERUMBAVOOR
Koovappady P O, Perumbavoor, Ernakulam – 683544
www.gptcperumbavoor.ac.in

This page is intentionally kept blank to preserve page format.

Index

General Remarks: (For office use only)						
Test 1:	Test 2:	Test 3:	Assign 1:	Assign 2:	Assign 3:	

Exp No. | | Date: | | | / | | | / | | |

VISION AND MISSION

Government Polytechnic College, Perumbavoor Vision and Mission
Vision Excel as a centre of skill education moulding professionals who sincerely strive for the betterment of society.
Mission • To impart state of the art knowledge and skill to the graduate and moulding them to be competent, committed and responsible for the well being of society. • To apply technology in the traditional skills, thereby enhancing the living standard of the community

Department of Electronics & Communication Engineering
Vision Excel as a centre of skill education in Electronics and Communication Engineering, moulding professionals who sincerely strive for the betterment of themselves and society.
Mission • To impart state of the art knowledge, skill and attitude to the students and contributing to their sustainable development. • To merge technologies in the field of Electronics and Communication Engineering with occupational skills, thereby improving quality of living.

Exp No. Date: / /

PEO, PO and PSOs of the Program

Program Educational Outcome (PEOs)

PEO1: Secure successful careers in manufacturing, testing, maintenance, development and marketing in Electronics and Communication Engineering.

PEO2: Acquire knowledge and competency in the domain to develop innovative, cost effective and socially acceptable solutions to engineering problems in a multi disciplinary work environment.

PEO3: Develop strong fundamental knowledge that prepares them for professional careers/higher studies with attitude for lifelong learning.

PEO4: Instill the attitude to be sensitive to ethical, societal and environmental issues while pursuing their professional duties.

PEO5: Possess leadership qualities and be effective communicator to work efficiently with diverse teams, promote and practice appropriate ethical practices.

Program Outcomes (POs)

1. Ability to communicate effectively with customers and officials, both in writing and orally.

2. An ability to apply knowledge of Electronics, communication, computing, mathematical foundations and theory in the implementation and design of electronics based systems.

3. An ability to interpret data sheets and schematics.

4. An ability to understand and solve real time problems in electronics, electronics and communication domain.

5. An ability to develop real time applications based on communication programming skills.

6. Ability to configure, troubleshoot and maintain electronic, communication and digital circuits.

7. Ability to apply entrepreneurial skills in the awake of incubation centers.

8. Ability to use and design communication systems.

9. An ability to recognize the importance of professional development by pursuing undergraduate studies or face competitive examinations that offer challenging and rewarding career in Electronics & Communication.

10. Understanding professional ethical legal security and social issues and responsibilities further to function effectively in a multi disciplinary environment.

Program Specific Outcome (PSO):

PSO1: Specialization knowledge: Apply concepts and knowledge in the field of semiconductor devices, communication and networking technologies, embedded systems.

PSO2: Professional growth: Generate ideas from the knowledge of engineering specialization leading to professional growth.

PSO3: Entrepreneurship: Apply knowledge and understanding of engineering principles to initiate entrepreneurship ventures.

Exp No. 1 Date: / /

SAFETY PROCEDURES

Problem Statement:

The safety instructions are presented to the attention of the students as a mean of preventing accidents while performing experiments and activities in the communication lab of the department .The purpose is to draw attention to the risks involved in lab activities to prevent human suffering and damage to equipment.

Safety in the laboratory:

Working in the lab is not allowed without following electricity precautions displayed.

No individual work is allowed in the lab.

Laboratory in charge is responsible for the arrangements of your lab activities; Listen carefully to his/her instructions and follow them.

To do and not to do:
Inform the lab in charge about dangerous conditions and faults in the lab or nearby environment.

Do not do any action that may harm people or equipments in the lab.

Do not misuse any of the tools or instruments belong to the lab.

Strict discipline should be maintained in the laboratory.

Turn off cell phones before entering the lab.

At the end and beginning of laboratory, follow 5S procedures and leave the work table clean and tidy.

Electrical Safety:

Consult Electrical Engineering section available in the campus for electrical safety queries.

The lab equipment is powered from electrical sockets installed on the tables. Do not use equipment that is powered from a damaged socket.

Do not use equipment that is powered from flexible cable with damaged insulation or if it's plug is not assembled properly.

Do not repair or disassemble electrical equipment including replacement of fuses installed in the equipment.

Do not open the main fuse box, unless it is an emergency and you need to switch off main circuit breaker.

Emergency Switches:

The laboratory has circuit breakers, which is located in the main panel. Identify the place.
In an emergency condition, switch off circuit breakers immediately.

Result

Familiarization of safety precautions performed.

	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 2 Date: / /

HANDLING ELECTROSTATIC DISCHARGE (ESD)

Problem Statement:

Familiarize ESD handling procedures in the laboratory

Theory

In handling electronic devices, datasheets cautions about ESD (Electrostatic Discharge) precautions. These devices are prone to damage because of electrostatic charges made by human body. These charges may be up to 4000 volts and cause damage without being noticed. It is recommended to follow ESD precautions on handling of these devices.

Points for the elimination of ESD damage to electronic components

1. Make sure you have a reliable ground point available near the table.
2. Do not wear clothing which generates static electric charges every time you move.
3. Do not handle static generating objects while working on electronics.
4. Store all chips and other components in appropriate anti-static containers.
5. Keep all ESD sensitive components and spares in anti-static envelopes for storage.
6. Be sure to turn off the power and remove the power plug from all equipment before working repairing or assembling.
7. Do not plug in or remove equipments while the power is on.

Result

Familiarization of ESD protection procedures performed.

	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 3 Date: / /

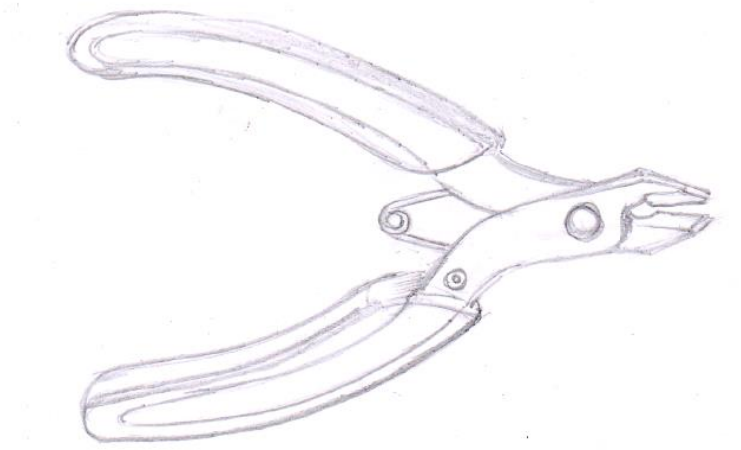
FAMILIARIZATION OF TOOLS AND PROCEDURES

Problem Statement:

To familiarize proper usage of tools used for handling componenets

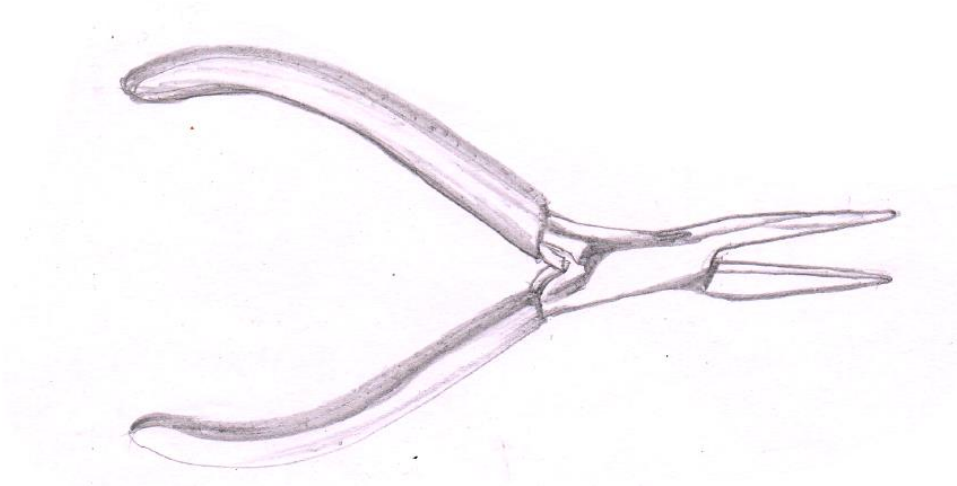
Nipper

These electronic nippers are for cutting wires and component leads. They have a cushioned handle which makes them much more comfortable to use.



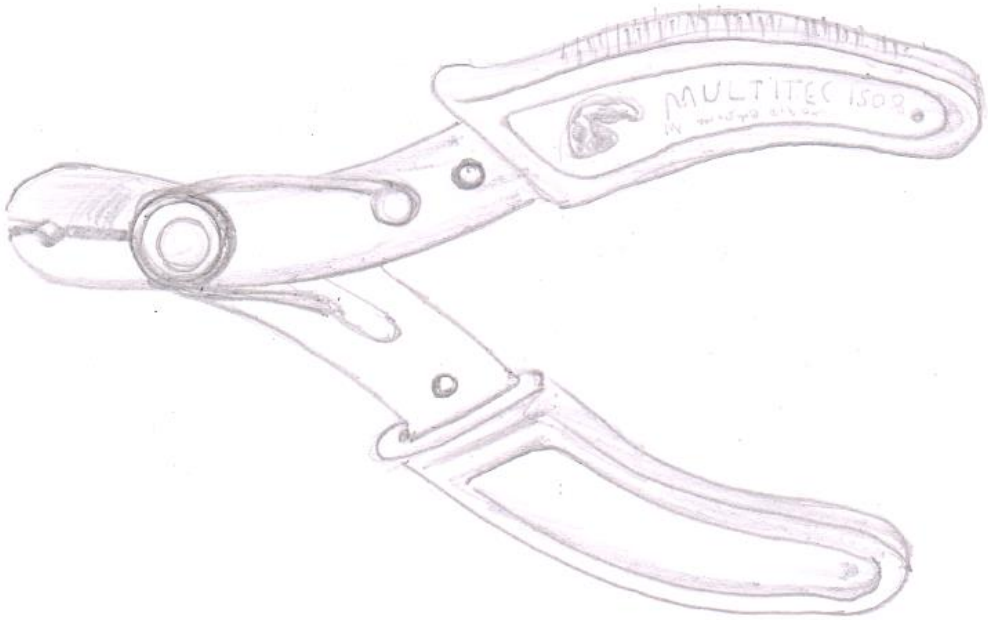
Long nose pliers

Needle-nose pliers are both cutting and holding pliers used to bend, re-position and snip wire. Because of their long shape they are useful for reaching into small areas where cables or other materials have become stuck or unreachable with fingers or other means.



Wire stripper

A wire stripper is a small, hand-held device used to strip the electrical insulation from electric wires.



Result

Study of handling components and their fixing performed.

	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 4 Date: / /

STUDY OF LOGIC GATES & UNIVERSAL GATES

AIM:

To familiarize

i. TTL and CMOS Logic ICs for AND, OR, NOT, NAND, NOR and XOR by verification of truth tables.

ii. Universal gates for implementing other logic functions.

OBJECTIVES:

- Students will be familiar with the basic logic gate ICs commonly used in digital electronics.
- They will get an idea about both TTL and CMOS logic family.
- They will understand the concept of Truth table verification.
- They will understand the universal property of NAND and NOR gates.

COMPONENTS AND EQUIPMENTS REQUIRED:

SL No.	COMPONENTS	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, CD 4011	1 each
5.	NOR GATE	IC 7402, CD 4001	1 each
6.	X-OR GATE	IC 7486	1
7.	IC TRAINER KIT	-	1
8.	Connecting Wires	As required	

PRINCIPLE:

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, 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. Output Y= A.B where A and B are the inputs.

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. Output Y= A+B where A and B are the inputs.

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. Output Y= NOT(A) where A is the input.

NAND GATE:

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

NOR GATE:

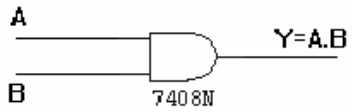
The NOR gate is a combination of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.

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.

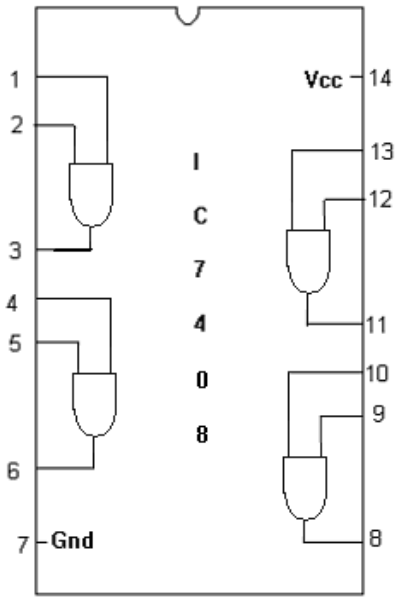
SYMBOL: **PIN DIAGRAM:**

AND GATE



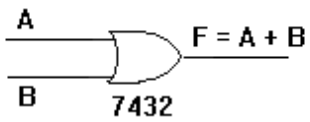
TRUTH TABLE

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

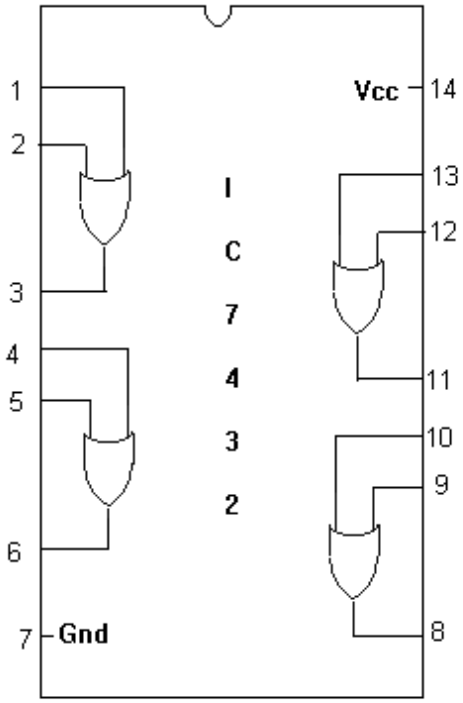


OR GATE:

SYMBOL :



PIN DIAGRAM :

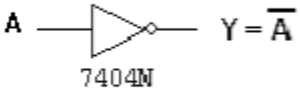


TRUTH TABLE

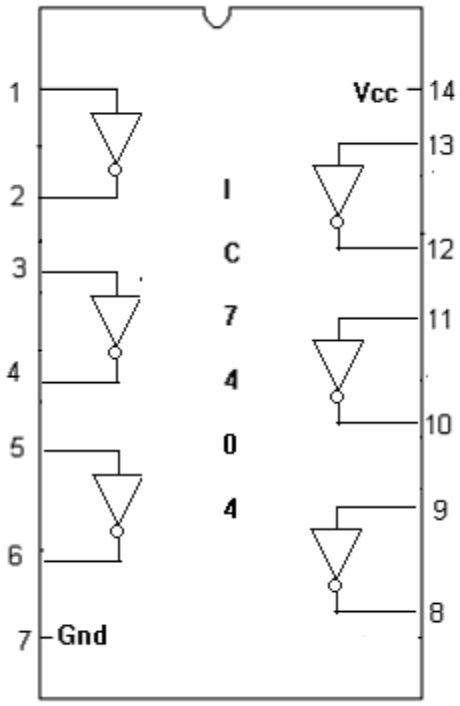
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE:

SYMBOL:



PIN DIAGRAM:

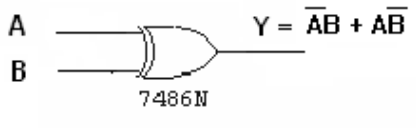


TRUTH TABLE :

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

X-OR GATE :

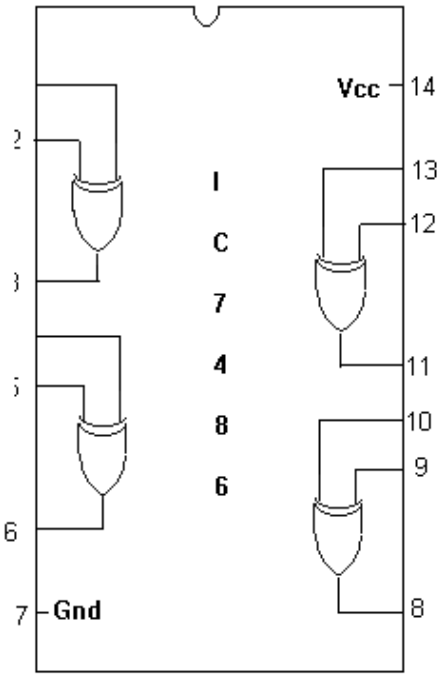
SYMBOL :



TRUTH TABLE :

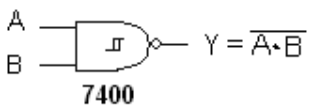
A	B	$\overline{A}B + A\overline{B}$
0	0	0
0	1	1
1	0	1
1	1	0

PIN DIAGRAM :



2-INPUT NAND GATE:

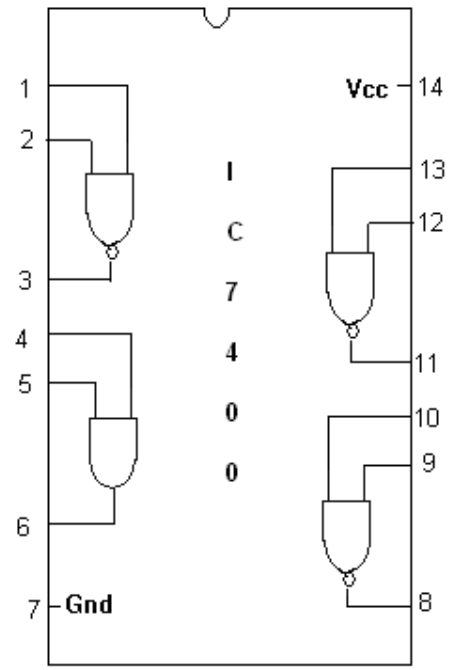
SYMBOL:



TRUTH TABLE

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

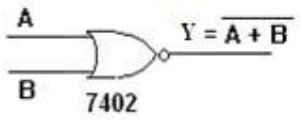
PIN DIAGRAM:



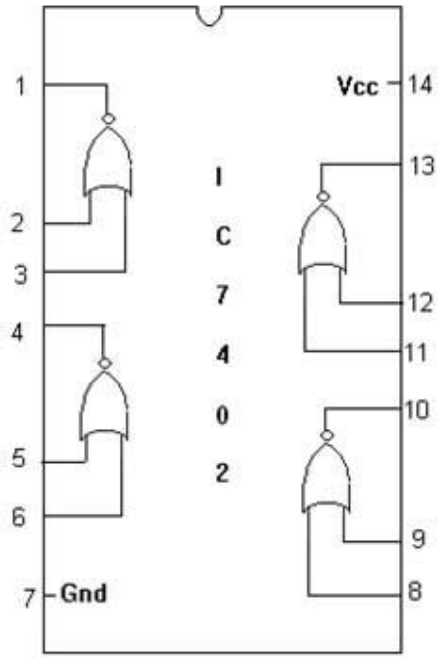
NOR GATE:

NOR GATE:

SYMBOL :



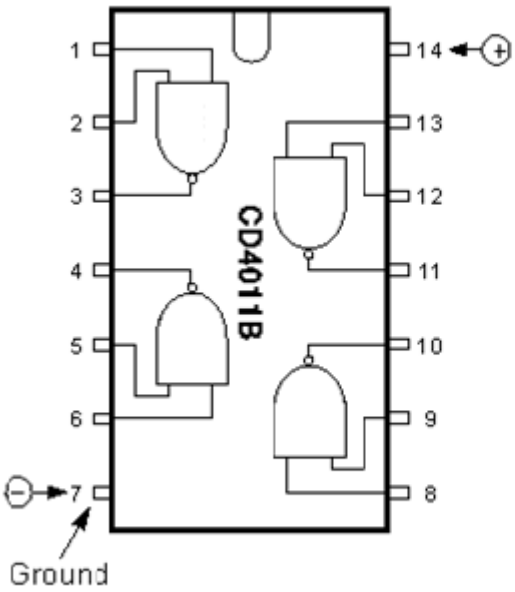
PIN DIAGRAM :



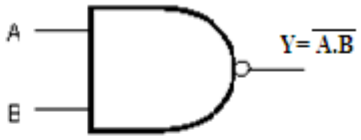
TRUTH TABLE

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

CMOS IC CD4011 -2 Input NAND Gate



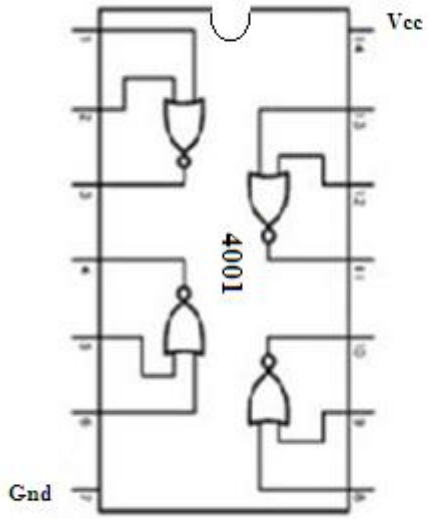
PIN DIAGRAM

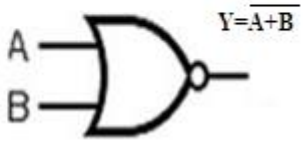


A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

TRUTH TABLE

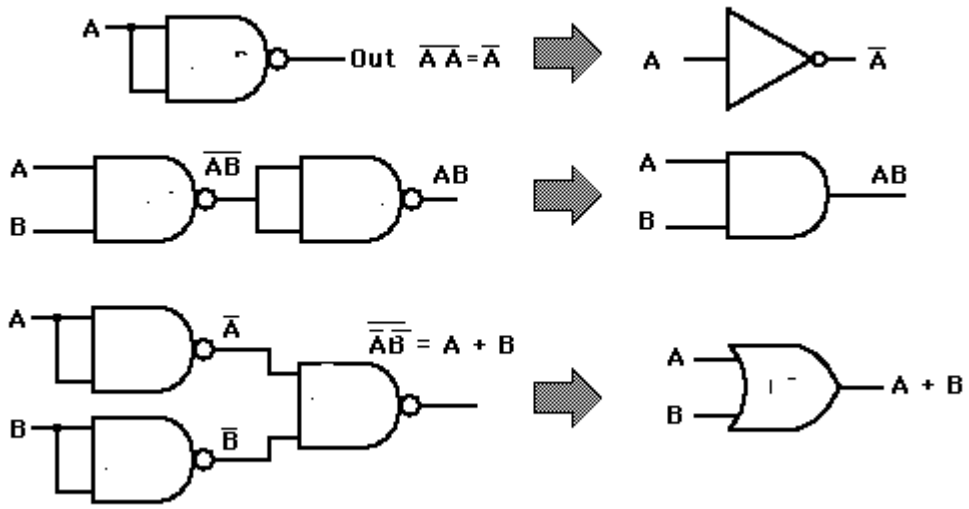
CMOS IC CD4001 -2 Input NOR Gate





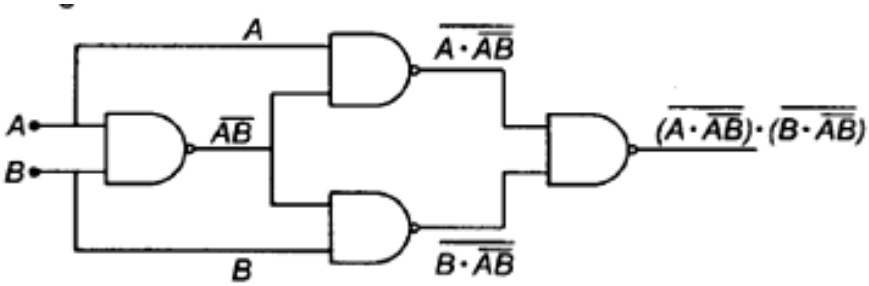
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

UNIVERSAL PROPERTY OF NAND GATE

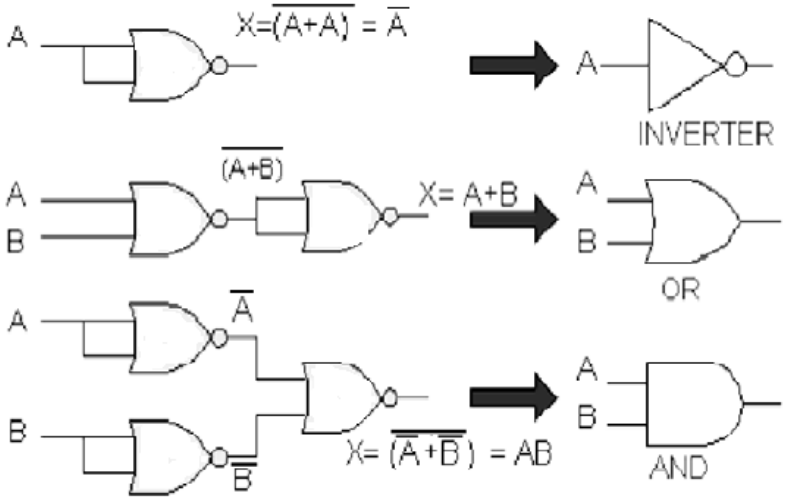


NAND as EXOR gate

$$A \oplus B = \bar{A}B + A\bar{B}$$
$$= \bar{A}B + A\bar{B} + A\bar{A} + B\bar{B}$$
$$= A.(\bar{A} + \bar{B}) + B.(\bar{A} + \bar{B})$$
$$= A.\overline{(A.B)} + B.\overline{(A.B)}$$
$$= A.\overline{(A.B)} \cdot B.\overline{(A.B)}$$

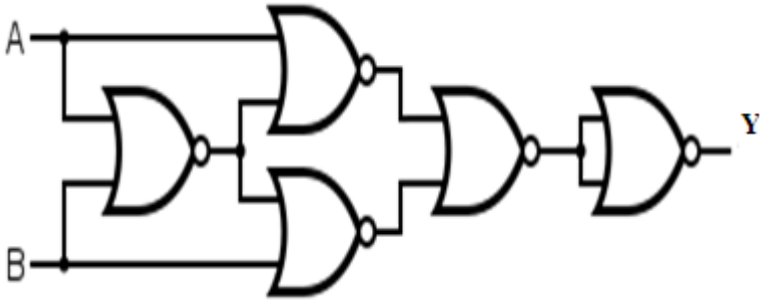


UNIVERSAL PROPERTY OF NOR GATE



NOR as EXOR gate

(using minimal no of NOR gates)



$$A \oplus B = (0 + \overline{A}B) + (A\overline{B} + 0)$$

$$= (A\overline{A} + A\overline{B}) + (\overline{A}B + B\overline{B})$$

$$= \overline{A}(A + B) + \overline{B}(A + B)$$

$$= \overline{\overline{A} \cdot (A+B)} + \overline{\overline{B} \cdot (A+B)}$$

$$= \overline{A+\overline{A+B}} + \overline{B+\overline{A+B}}$$

$$= \overline{A+\overline{A+B}} + \overline{B+\overline{A+B}}$$

PROCEDURE:

1. Check the all the required components.
2. Connect the circuit as per the circuit diagram.
3. Apply Vcc and Ground.
4. Logical inputs are given as per truth table using switches provided.
5. Observe the output on LEDs and verify the truth table.

RESULT:

i. Familiarized with basic TTL and CMOS logic gate ICs and verified their truth tables.

ii. Constructed basic logic gates and X-OR using Universal gates.

INFERENCE:

1. By using NAND and NOR gates we can construct any logic circuits.
2. ……

Exp No. 5 Date: / /

HALF AND FULL ADDERS

AIM:

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

OBJECTIVES:

- Students will study how to interconnect different gates in a circuit.
- They can differentiate Half adder from Full adder.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	OR GATE	IC 7432	1
4.	IC TRAINER KIT	-	1
5.	Connecting wires	-	as required

PRINCIPLE:

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. The Sum is obtained from the X-OR Gate and the Carry out from the AND gate.

FULL ADDER:

A full adder is a combinational circuit that forms the arithmetic sum of three inputs and generates two outputs. A full adder is useful to add three bits at a time but a half adder cannot do so.

Design

HALF ADDER

TRUTH TABLE:

A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

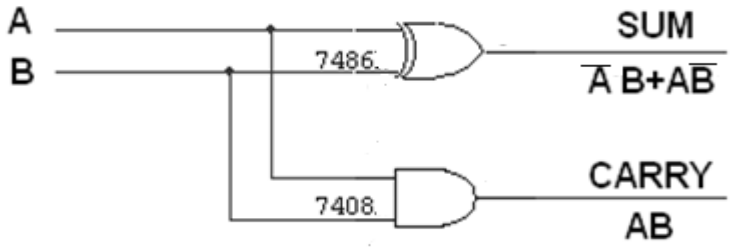
From above truth table, the expressions for carry and sum can be derived using the sum product terms.

$$\text{Sum} = \overline{A}.B + A.\overline{B}$$

$$= A \oplus B$$

Carry= AB

LOGIC DIAGRAM:



Design of Half adder Using NAND ONLY

Sum

$$A \oplus B = \bar{A}B + A\bar{B}$$
$$= \bar{A}B + A\bar{B} + A\bar{A} + B\bar{B}$$
$$= A.(\bar{A} + \bar{B}) + B.(\bar{A} + \bar{B})$$
$$= A.\overline{(A.B)} + B.\overline{(A.B)}$$
$$= \overline{A.\overline{(A.B)}} \cdot \overline{B.\overline{(A.B)}}$$

Carry = A.B

$$= \overline{\overline{A.B}}$$

FULL ADDER

TRUTH TABLE:

A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

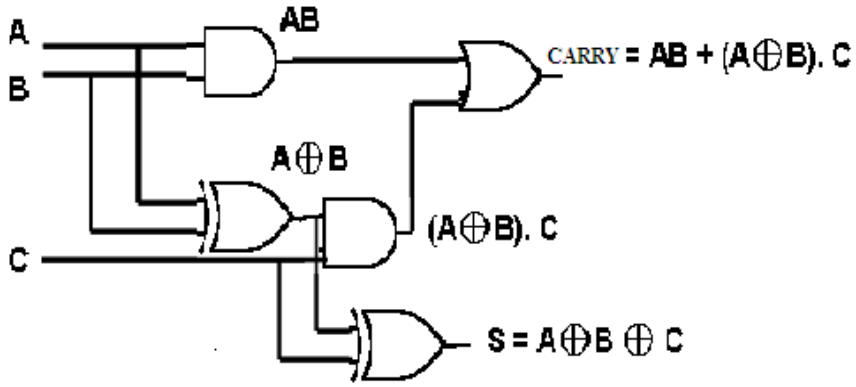
DESIGN:

$$\text{Sum} = \bar{A}\,\bar{B}\,C + \bar{A}\,B\bar{C} + A\bar{B}\,\bar{C} + ABC$$
$$= (\bar{A}\,\bar{B} + AB).C + (\bar{A}\,B + A\bar{B})\,\bar{C}$$
$$= \overline{(A \oplus B)}\,C + (A \oplus B)\,\bar{C}$$
$$= (A \oplus B) \odot C$$
$$= (A \oplus B) \oplus C$$

$$CARRY = \overline{A}\,BC + A\overline{B}\,C + AB\overline{C} + ABC$$

$$= (\overline{A}\,B + A\overline{B})\,.C + AB\,(\overline{C} + C)$$

$$= (A \oplus B).C + AB \qquad \text{(we know that } \overline{C} + C = 1)$$



Using NAND gates only.

Design

We know that,

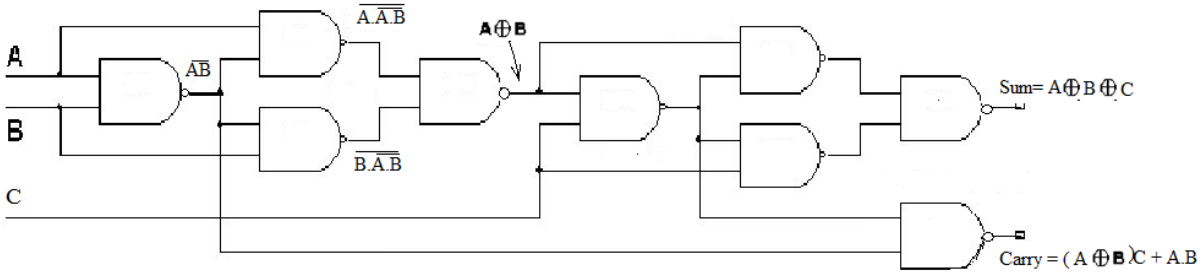
$$A \oplus B = \overline{\overline{A.\overline{A.B}} \;.\; \overline{B.\overline{A.B}}}$$

Then sum $S = A \oplus B \oplus C$

$$= [A \oplus B] \oplus C$$

$$= \overline{\overline{(A \oplus B).\overline{(A \oplus B).C}} \;.\; \overline{C.\overline{(A \oplus B).C}}}$$

$$Carry = \overline{\overline{(A \oplus B).C} \;.\; \overline{A.B}}$$



PROCEDURE:

1. Check the required components
2. Connections are given as per circuit diagram.
3. Logical inputs are given as per truth table.
4. Observe the output and verify the truth table.

RESULT:

Designed and constructed half adder and full adder, circuits and verified the truth table using logic gates.

Inference :

1. A Full adder can handle additional carry input compared to a Half adder.
2. Students get a basic idea about the addition circuits used in calculators and ALU.

Exp No. 6 Date: / /

HALF AND FULL SUBTRACTORS

AIM:

To construct half subtractor and full subtractor circuits and verify the truth table using logic gates.

OBJECTIVES:

- Students will study how to interconnect different gates in a circuit.
- They can differentiate Half subtractor from Full subtractor.

COMPONENTS AND EQUIPMENTS 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.	IC TRAINER KIT	-	1
6.	Connecting wires	-	as required

THEORY:

HALF SUBTRACTOR:

The half subtractor is constructed using X-OR, NOT 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. A full subtractor has three inputs while half subtractor has only two. This additional carry input facility makes them to subtract any two numbers. Two half subtractors can be connected together to give a full subtractor.

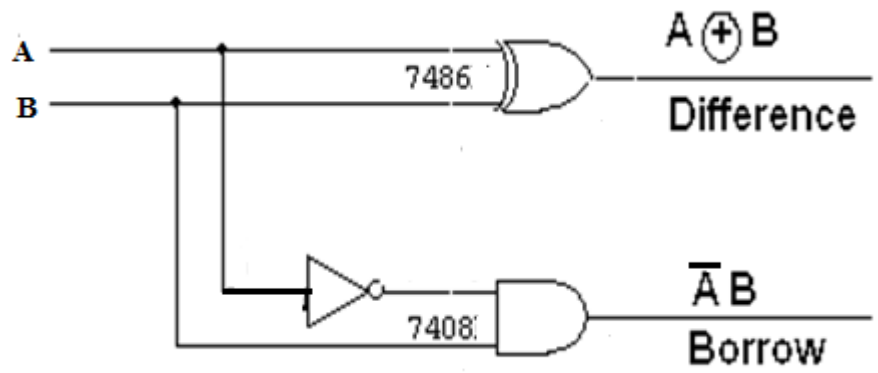
TRUTH TABLE FOR HALF SUBTRACTOR

A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

DIFFERENCE = $\overline{A}B + A\overline{B}$ = A $\oplus$B

BORROW = $\overline{A}B$

HALF SUBTRACTOR LOGIC DIAGRAM:



FULL SUBTRACTOR

TRUTH TABLE:

A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

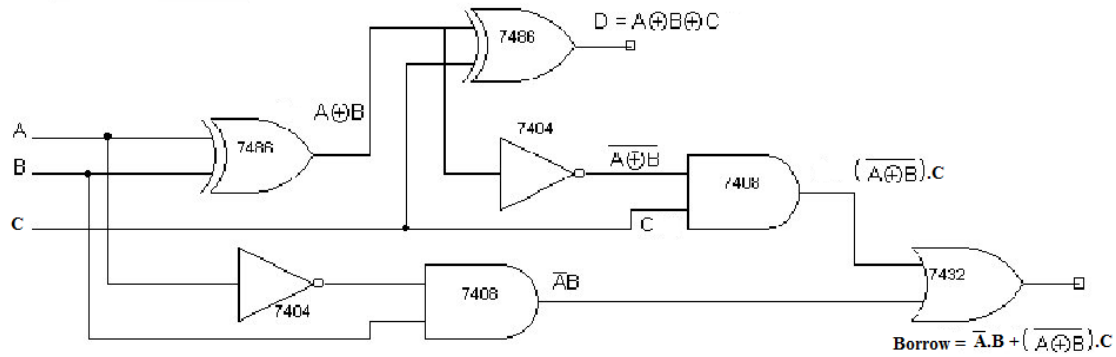
DESIGN:

From the truth table, taking the sum of product terms, we get

$$\text{Difference} = \bar{A}\,\bar{B}\,C + \bar{A}\,B\bar{C} + A\bar{B}\,\bar{C} + ABC$$
$$= (\bar{A}\,\bar{B} + AB).C + (\bar{A}\,B + A\bar{B})\,\bar{C}$$
$$= \overline{(A \oplus B)}\,C + (A \oplus B)\,\bar{C}$$
$$= (A \oplus B) \oplus C$$

$$\text{Borrow} = \bar{A}\,\bar{B}\,C + \bar{A}\,B\,\bar{C} + \bar{A}\,B\,C + A\,B\,C$$
$$= (\bar{A}\,\bar{B} + AB).\,C + \bar{A}B\,(\bar{C} + C)$$
$$= \overline{(A \oplus B)}\;C + \bar{A}\,B$$

FULL SUBTRACTOR LOGIC DIAGRAM:

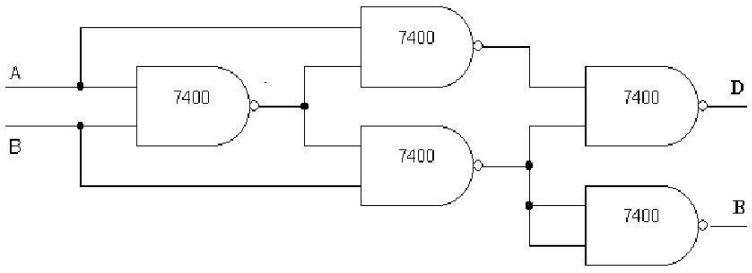


DESIGN:

Design for implementing Half subtractor aand Full subtractor using NAND gates **should also be done by students.**

LOGIC DIAGRAM

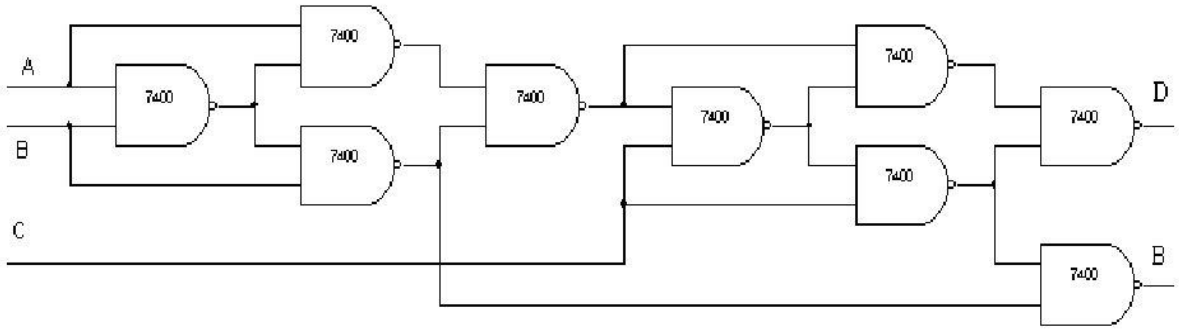
(a)Half subtractor using NAND only.



(b) Full subtractor using NAND only

Diff= (A ⊕B) ⊕ C

Borrow = $\overline{(A \oplus B)}$ C + $\bar{A}$ B



PROCEDURE:

1. Check the required components
2. Connections are given as per circuit diagram.
3. Logical inputs are given as per truth table.
4. Observe the output and verify the truth table.

RESULT:

Designed and constructed half subtractor and full subtractor circuits and verified their truth tables.

Inference :

1. A full subtractor is always required to handle three inputs.
2. ----------

Exp No. 7 Date: / /

BINARY TO GRAY AND GRAY TO BINARY CONVERTER

AIM:

To construct and verify the truth tables of

(i) Binary to gray code converter

(ii) Gray to binary code converter

OBJECTIVE:

- Students will study how to construct a Binary to Gray and Gray to binary code converters.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	IC TRAINER KIT	-	1
3.	Connecting wires	-	as required

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.

Here we take 4 bits for Gray to binary and vice versa conversions. Thus there are four inputs and four outputs. Gray code is a non-weighted code and its hamming distance is always 1.

The input variables for a Binary to Gray converter are designated as B3, B2, B1, and B0 and the output variables are designated as G3, G2, G1,and G0. From the truth table, combinational circuit is designed through K-Map.

Meanwhile the inputs for a Gray to Binary converter is G3, G2, G1,and G0 and its output generates B3, B2, B1, and B0.

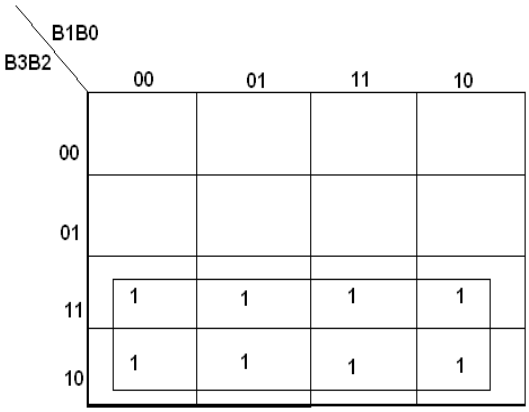
DESIGN

BINARY TO GRAY CODE CONVERTOR

TRUTH TABLE:

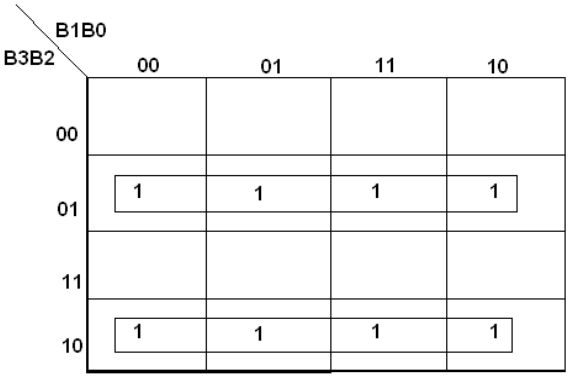
Binary inputs				Gray outputs			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

K-Map for G_3:



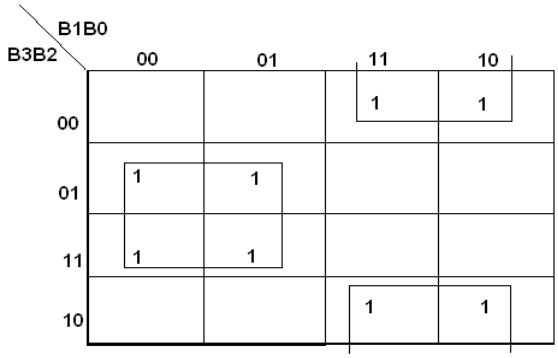
$G_3 = B_3$

K-Map for G_2:



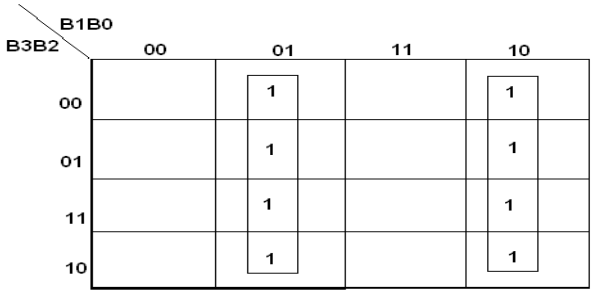
$$G2 = \overline{B3}B2 + B3\overline{B2}$$
$$= B3 \oplus B2$$

K-Map for G_1:



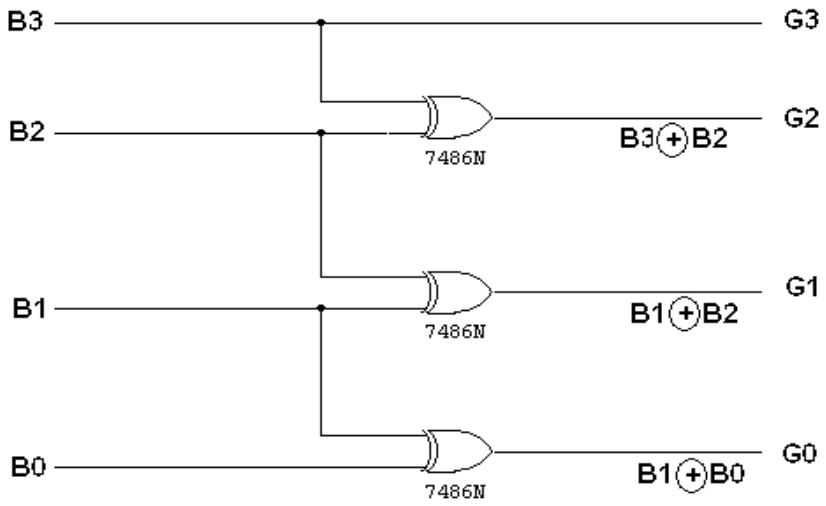
$$G1 = \overline{B1}B2 + B1\overline{B2}$$
$$= B1 \oplus B2$$

K-Map for G_0:



$$G0 = \overline{B1}B0 + B1\overline{B0}$$
$$= B1 \oplus B0$$

LOGIC DIAGRAM:-

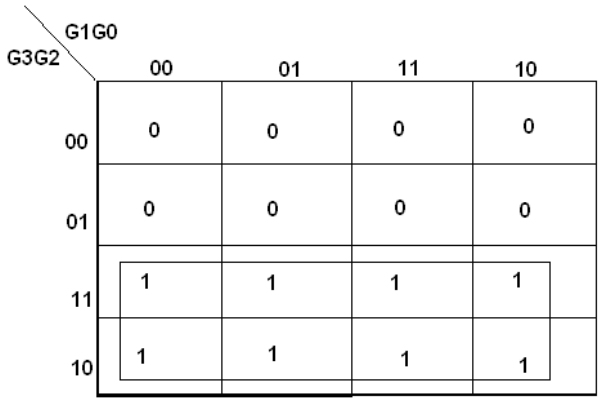


GRAY CODE TO BINARY CONVERTOR

TRUTH TABLE:

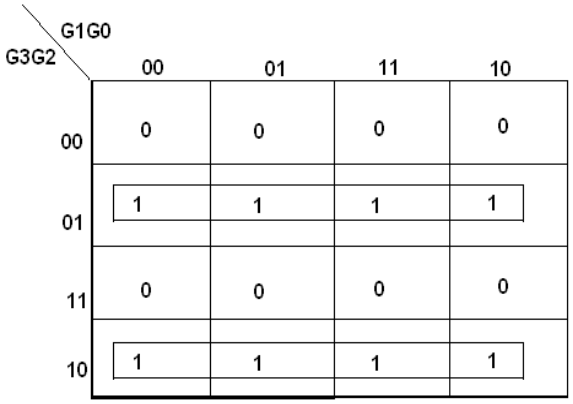
Gray Code				Binary Code			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

K-Map for B_3:



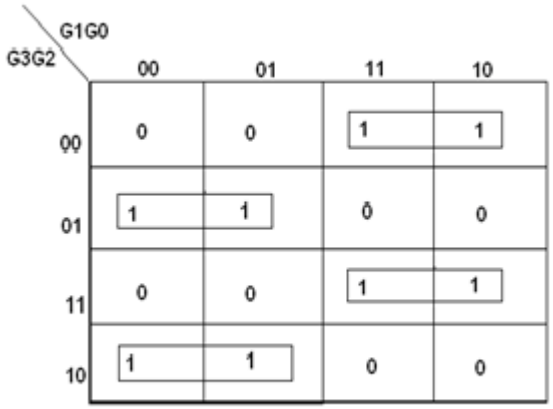
B3 = G3

K-Map for B_2:



$$B2 = \overline{G3}G2 + G3\overline{G2}$$
$$= G3 \oplus G2$$

K-Map for B_1:



$$B1 = \overline{G3}G2\overline{G1} + G3\overline{G2}\overline{G1} + \overline{G3}\overline{G2}G1 + G3G2G1$$
$$= \overline{G1}(\overline{G3}G2 + G3\overline{G2}) + G1(\overline{G3}\overline{G2} + G3G2)$$
$$= \overline{G1}.(G3 \oplus G2) + G1.\overline{(G3 \oplus G2)}$$
$$= G1 \oplus G2 \oplus G3$$
$$= G1 \oplus B2$$

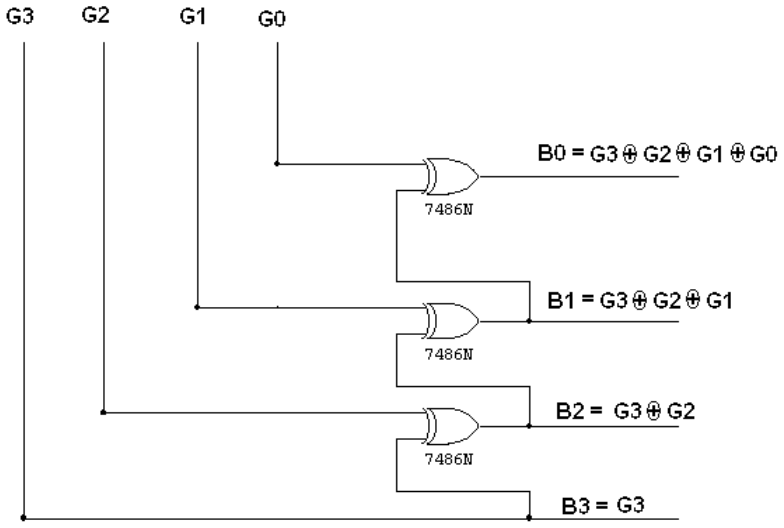
K-Map for B_0:

G3G2 \ G1G0	00	01	11	10
00	0	①	0	①
01	①	0	①	0
11	0	①	0	①
10	①	0	①	0

B0 = ----------
= ----------
= ----------
= G3⊕G2⊕G1⊕G0
= B1 ⊕ G0

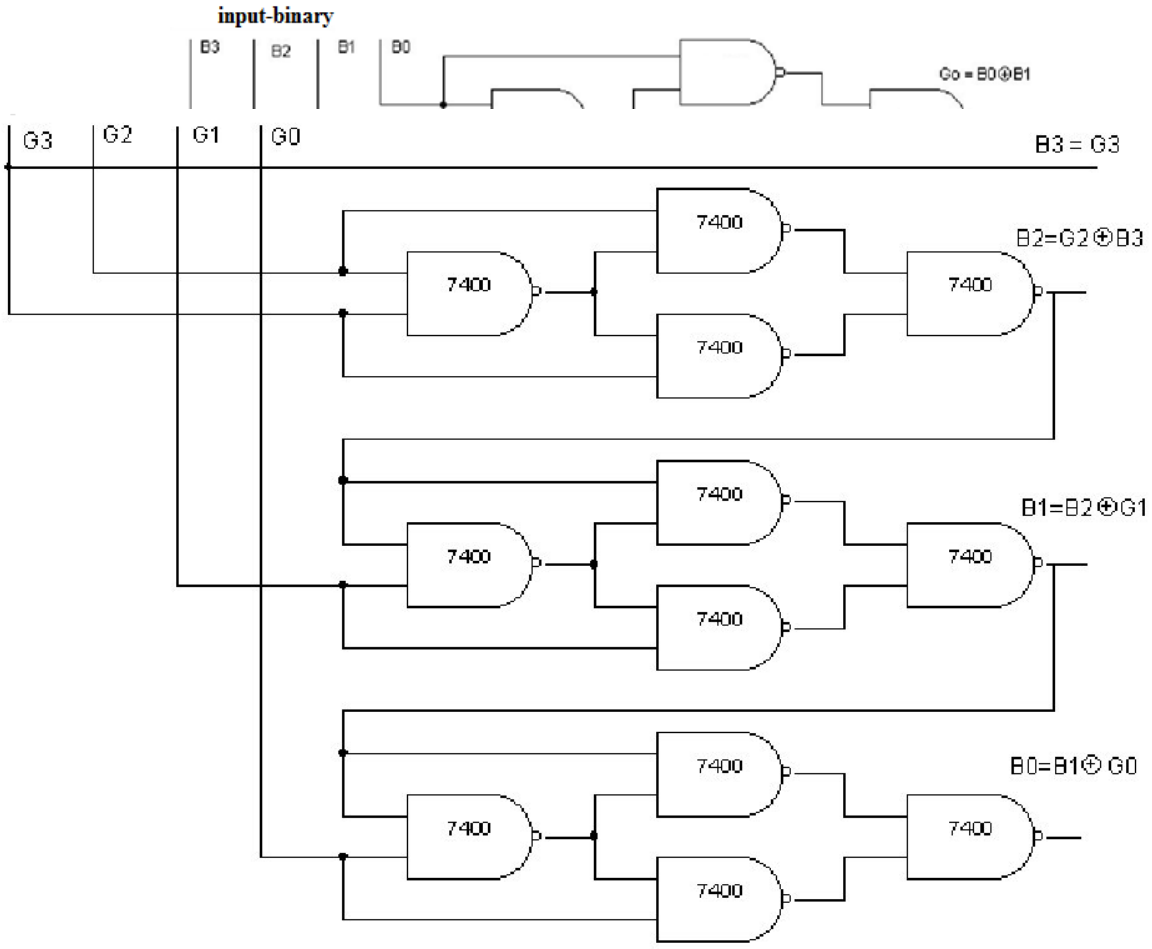
(Solution should be completed by students)

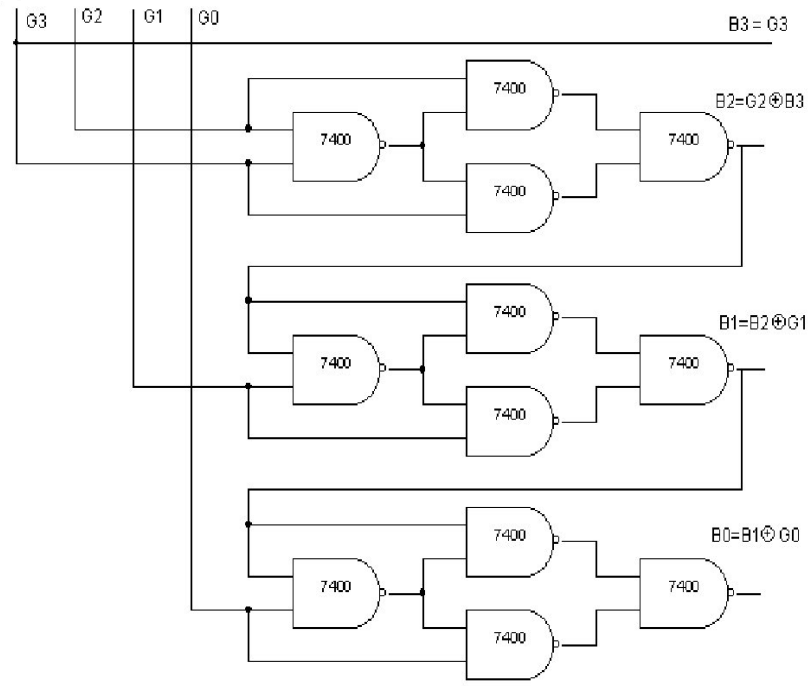
LOGIC DIAGRAM OF GRAY TO BINARY CODE CONVERTER



Using NAND Gates Only: -

Binary To Gray





PROCEDURE:

1. The circuit connections are made as shown in fig.
2. Pin (14) is connected to +Vcc and Pin (7) to ground.
3. In the case of binary to gray conversion, the inputs B0, B1, B2 and B3 are given at respective pins and outputs G0, G1, G2, G3 are taken for all the 16 combinations of the input.
4. In the case of gray to binary conversion, the inputs G0, G1, G2 and G3 are given at respective pins and outputs B0, B1, B2, and B3 are taken for all the 16 combinations of inputs.
5. The values of the outputs are tabulated.

RESULT:

Designed and implemented

a. Binary to gray code converter
b. Gray to binary code converter

Inference :

1.By using a single IC we can do both Gray to Binary and Binary to Gray converter.

2. Students get an idea to construct circuit for any code converter.

3._____

Exp No. 8 Date: / /

COMBINATIONAL LOGIC CIRCUITS FROM BOOLEAN EQUATIONS

AIM:

To implement combinational logic circuits from Boolean Equations.

OBJECTIVES:

After completing the experiment students will learn

- To study how to find which are the input and output variables of the problem.
- To study how to derive the truth table
- To study how to implement the circuit.

COMPONENTS AND EQUIPMENTS:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4	X-OR GATE	IC 7486	1
5	AND GATE 3 I/P	IC 7411	1
6	IC TRAINER KIT	-	1
7.	CONNECTING WIRES	As required -	

THEORY:

The term combinational comes from mathematics. In mathematics a combination is an unordered set. The combinational circuit produces the same output regardless of the order the inputs are changed. A combinational circuit is one where the output at any time depends only on the present combination of inputs at that point of time. The logic gate is the most basic building block of combinational logic. The logical function performed by a combinational circuit is fully defined by a set of Boolean expressions. The other category of logic circuits, called sequential logic circuits, comprises both logic gates and memory elements such as flip-flops. Owing to the presence of memory elements, the

output in a sequential circuit depends upon not only the present but also the past state of inputs.

For converting Boolean expressions to logic diagrams the order of precedence of operations is given below:

1. First, perform all inversions of single terms in the given expression.

2. Perform all operations with parentheses.

3. Perform an AND operation before an OR operation unless parentheses indicate otherwise.

4. If an expression has a bar over it, perform the operations inside the expression first and then invert the result.

PROCEDURE:

1. Prepare the truth table and draw the logic circuit diagram.
2. Check the required components.
3. Make connections as shown in the circuit diagram.
4. Switch ON the trainer kit and provide the input data via the input switches.
5. Observe the output on output LEDs and verify it with truth table.

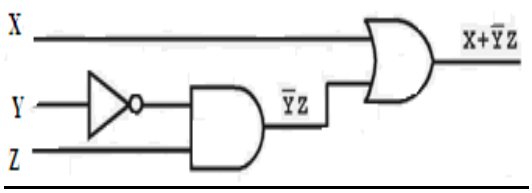
CIRCUIT DIAGRAM:

Note: Students are advised to construct circuit for different equations .

Example1. Implements the Boolean equation

$F(X,Y,Z)=X+\overline{Y}Z$

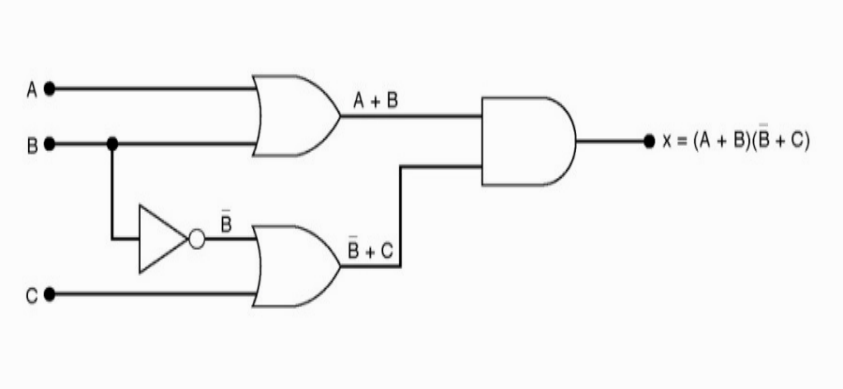
Inputs			$\overline{YZ}$	$\overline{X+YZ}$
X	Y	Z		
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1



Example.2. Draw the logic circuit diagram to implement $x = (A+B)(\overline{B}+C)$
Derive the truth table from equation and verify it.

TRUTH TABLE

Inputs					
A	B	C	(A+B)	$(\overline{B}+C)$	X
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1



Q1.Draw the circuit diagram that implements the equation below using gates having no more than three inputs.

$$x = \overline{A}BC(\overline{A+D})$$

Q2.

.

.

.

RESULT:

Implemented the combinational logic circuits, from the given Boolean expressions.

INFERENCE:

1. Students get basic idea about drawing circuit of any Boolean equation.

2.----------

Exp No. 9 Date: / /

4 BIT ADDER/SUBTRACTOR USING IC 7483

AIM:

To familiarise the 4 bit adder/subtractor using IC 7483.

OBJECTIVES:

To design and set up the following circuit using IC 7483.

- A 4-bit binary parallel adder.
- A 4-bit binary parallel subtractor.

.**COMPONENTS AND EQUIPMENTS:**

SL No.	COMPONENT	SPECIFICATION	QTY
1.	4 BIT ADDER IC	IC 7483	1
2.	NOT GATE	IC 7404	1
3.	X-OR GATE	IC 7486	1
4.	IC TRAINER KIT	-	1
5.	CONNECTING WIRES	AS REQUIRED	

THEORY:

A binary adder/subtractor is a digital circuit that produces the arithmetic sum or difference of two binary numbers. A binary adder can be constructed with full adders connected in cascade with the output carry from each adder connected to the next full adder in the chain. Subtraction of two numbers (A-B) can be performed by taking 2's complement of B and adding it with A. Thus binary adder can be used to perform subtraction of two binary numbers if few changes are made. In case of binary adder, input carry is set to be 0 and all the input bits are given as it is. Whereas for binary subtractor, in order to give 2's complement of B, consider input carry equal to **1** and the bits of B is complemented ie 1'scomplement of B is taken and it is added with 1 to give its 2's complement value. The block diagram of 4-bit binary adder/subtractor is as shown in figure. The circuit performs addition when M = 0 & subtraction when M = 1.

IC 7483 is an integrated circuit which performs addition of two 4-bit binary numbers.

4 BIT BINARY SUBTRACTOR**:**

The circuit for subtracting (A-B) consists of an adder with inverters, placed between each data input 'B' and the corresponding input of full adder. The input carry C_0 must be equal to 1 when performing subtraction.

4 BIT BINARY ADDER/SUBTRACTOR**:**

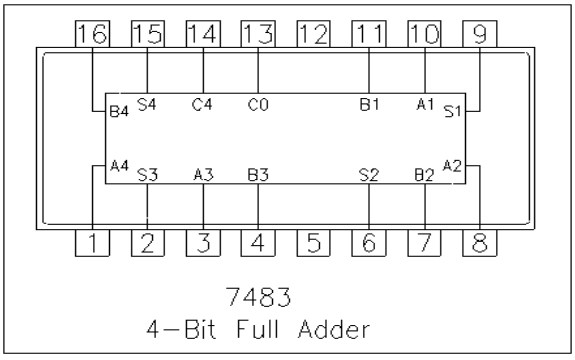
The addition and subtraction operation can be combined into one circuit with one common binary adder. The **mode input** M controls the operation. When M=0, the circuit is adder circuit. When M=1, it becomes subtractor.

PROCEDURE:

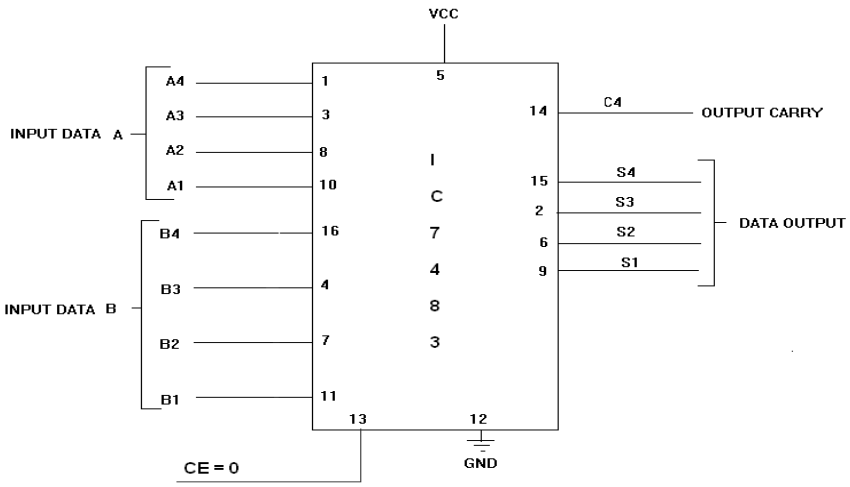
1. Test all the ICs manually/ using IC tester.
2. Connections are made as in the logic circuit diagram and give Vcc and the ground.
3. Connect the appropriate pins to the input switches and output LEDs.
4. Give various combinations of the inputs and observe the output in adder circuit.
5. Similarly verify the subtractor circuit with truth table.
6. Then setup the adder/subtractor circuit. Make M=0,and verify whether it works as a nibble adder.
7. Give M=1 to function as a subtractor and verify the output of adder/subtractor circuit by giving various combinations of inputs

CIRCUIT DIAGRAM:

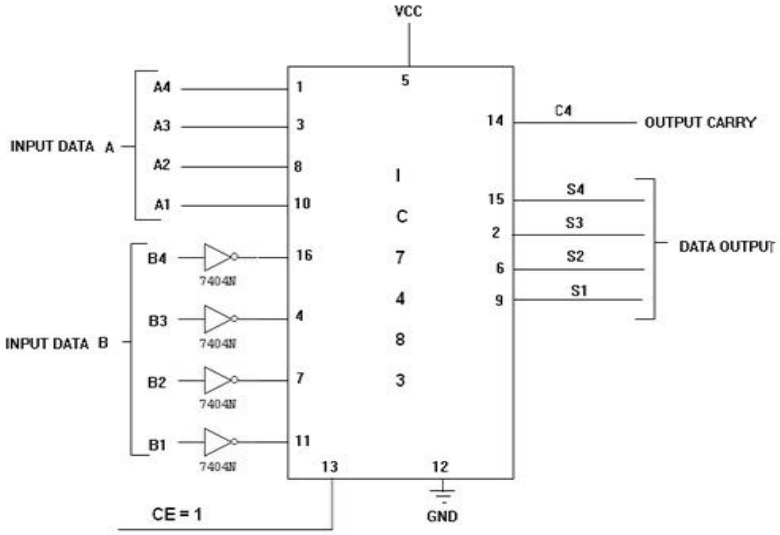
PIN DIAGRAM FOR IC 7483:



4-BIT BINARY ADDER



4-BIT BINARY SUBTRACTOR



4BIT BINARY ADDER/SUBTRACTOR

FUNCTION TABLE

Mode i/p	Input Data A				Input Data B				OUTPUT				
M	A4	A3	A2	A1	B4	B3	B2	B1	C1	S4	S3	S2	S1
0	1	0	0	0	0	0	1	0	0	1	0	1	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0	0	1	0	1	0
0	0	0	0	1	0	1	1	1	0	1	0	0	0
0	0	1	1	1	1	0	0	0	?	?	?	?	?
0	1	1	0	0	0	1	1	1					
0	1	1	1	1	1	1	1	1					
1	1	1	0	0	0	0	1	1	1	1	0	0	1
1	1	0	1	0	0	1	0	1	1	0	1	0	1
1	1	0	0	0	0	1	0	1	?	?	?	?	?
1	1	1	1	1	1	0	1	1					
1	1	1	1	0	1	0	0	0					
1	1	1	0	0	1	0	1	1					

Note:

Subtraction performed in the above circuits is 2’s complement method. Hence carry output is discarded.

When A < B ,and M=1,result will be in the 2’s complement form(ie. result is negative).

RESULT:
Studied 4 bit binary adder/subtractor circuit using IC 7483.

INFERENCE:
1. Students get an idea about adder IC s and its working.
2. IC7483 can be used for addition of BCD numbers, Octal numbers.

Exp No. 10 Date: / /

MULTIPLEXER AND DEMULTIPLEXER

AIM:

To setup a 4 :1 MUX and a 1: 4 DEMUX using (i) basic gates (ii) NAND gates.

OBJECTIVES: After completion of experiment

- To learn about various applications of multiplexer and de-multiplexer
- To learn and understand the working of MUX and DEMUX

.

COMPONENTS AND EQUIPMENTS:

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.	2 I/P NAND GATE	IC 7410	2
5	3 I/P NAND GATE	IC 7400	2
2.	IC TRAINER KIT	-	1
3.	CONNECTING WIRES	As required	

PRINCIPLE:

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

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

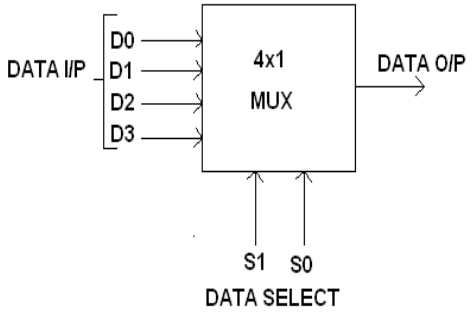
select/control inputs for the selection of output line into which the data is to be transmitted

PROCEDURE:

1. Test all the ICs manually/ using IC tester.
2. Connections are made as in the logic circuit diagram and give Vcc and the ground.
3. Connect the appropriate pins to the input switches and output LEDs.
4. Give various combinations of the inputs and observe the output and verify the truth table.
5. Similarly verify the demux circuit with truth table.

CIRCUIT DIAGRAM:

BLOCK DIAGRAM FOR 4:1 MULTIPLEXER:



DESIGN: 4:1 MUX FUNCTION TABLE:

SELECT INPUTS		DATA INPUTS				min term	OUTPUT
S1	S0	I0	I1	I2	I3		Y
0	0	1	x	x	x	$I0\bar{S1}\bar{S0}$	1
0	1	x	1	x	x	$I1\bar{S1}S0$	1
1	0	x	x	1	x	$I2S1\bar{S0}$	1
1	1	x	x	x	1	**I3S1S0**	1

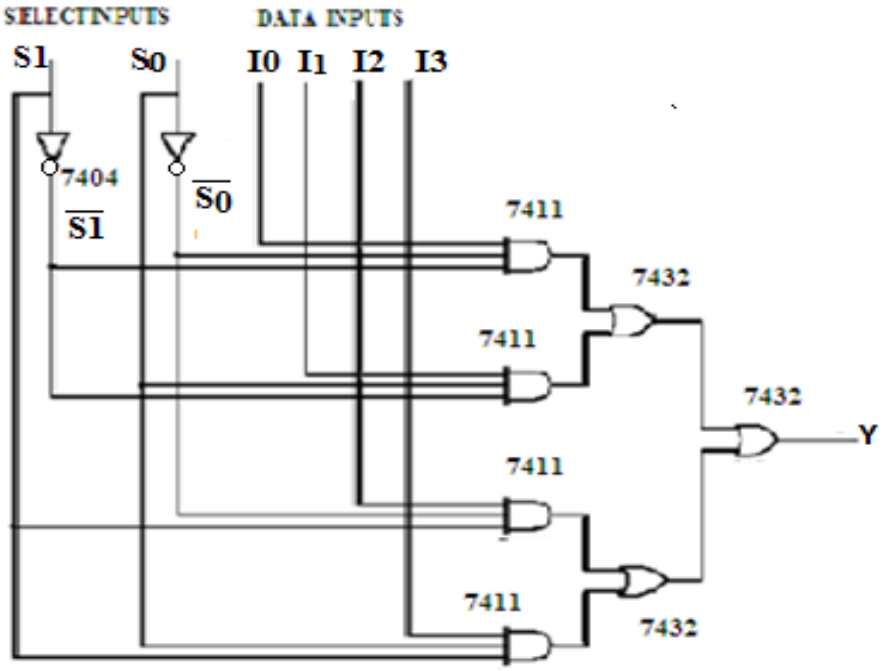
SOP of output $Y = I0\,\bar{S1}\,\bar{S0} + I1\,\bar{S1}S0 + I2\,S1\,\bar{S0} + I3\,S1\,S0$

4:1 MUX USING NAND GATES

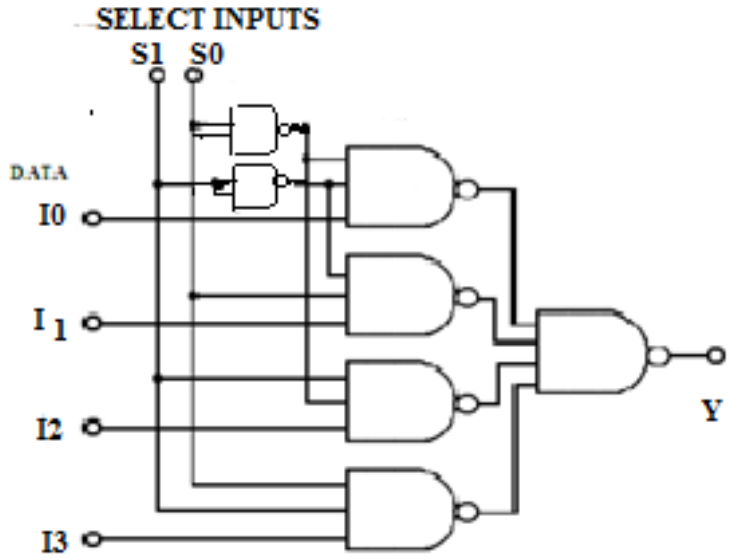
To implement using NAND gates, apply double inversion rule.

Then , $Y=\overline{I0\bar{S}1\bar{S}0+I1\bar{S}1S0+I2S1\bar{S}0+I3S1S0} = \overline{I0\bar{S}1\bar{S}0}\,.\,\overline{I1\bar{S}1S0}\,.\,\overline{I2S1\bar{S}0}\,.\,\overline{I3S1S0}$

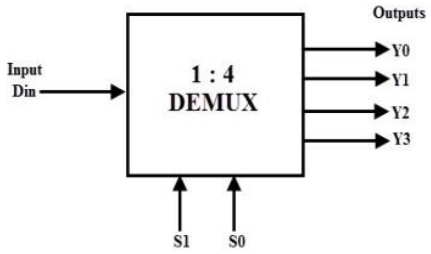
CIRCUIT DIAGRAMS: 4:1 MULTIPLEXER USING BASIC GATES:



4:1 MUX USING NAND GATES



BLOCK DIAGRAM FOR 1:4 DEMULTIPLEXER:



DESIGN

1:4 DEMUX FUNCTION TABLE

DATA INPUT	SELECT INPUTS		OUTPUTS			
	S1	S0	Y0	Y1	Y2	Y3
D	0	0	**D**	0	0	0
D	0	1	0	**D**	0	0
D	0	1	0	0	**D**	0
D	1	1	0	0	0	**D**

From the truth table, when S1=0 and S0= 0, the data input is connected to output Y0 and when S1= 0 and S0=1, then the data input is connected to output Y1. Similarly, other outputs are connected to the input for other two combinations of select lines.

From the table, the output logic can be expressed as min terms and are given below.

$Y0 = \overline{S1}\ \overline{S0}\ D$

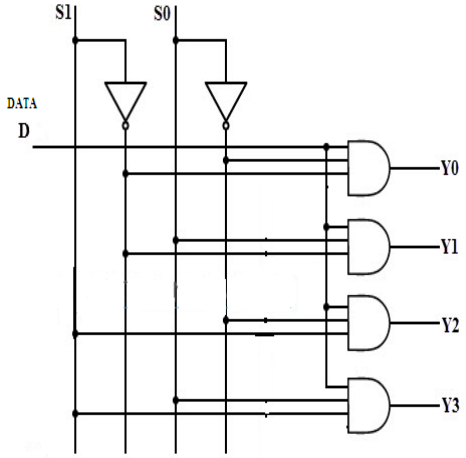
$Y1 = \overline{S1}\ S0\ D$

$Y2 = S1\ \overline{S0}\ D$

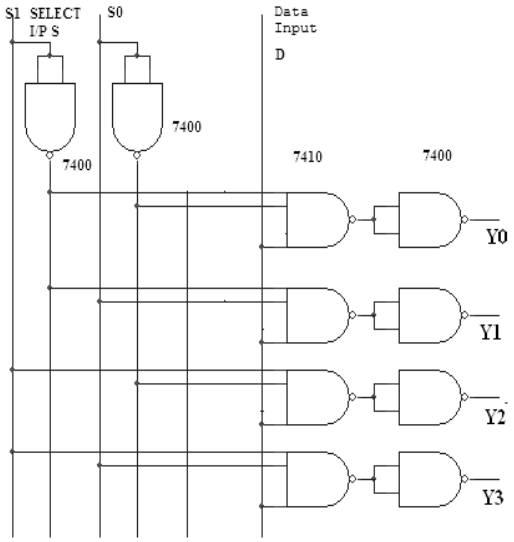
$Y3 = S1\ S0\ D$ Where D is the input data, Y0 to Y3 are output lines and S0 & S1 are select lines.

By applying double inversion rule to each output above, demux using NAND gates can be realized.

4:1 DEMUX USING BASIC GATES



4:1 DEMUX USING NAND GATES



RESULT:

Realized 4:1 Multiplexer and 1:4 Demultiplexer using basic gates and NAND gates and verified the truth table.

INFERENCE:

1. Many input data lines can be switched to a single line using multiplexer.

2. Data is directed to one of many output lines in a DEMUX circuit.

3. Selecting different IO devices for data transfer can be done using MUX.

4. These circuits are used for data routing.

Exp No. 11 Date: / /

STUDY OF MULTPLEXER IC 74151

AIM:

To study the multiplexer IC 74151.

OBJECTIVE:

- Students will study about multiplexer IC 74151.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	8 X 1 MUX	IC 74151	1
2.	IC TRAINER KIT	-	1
3.	Connecting wires	-	as required

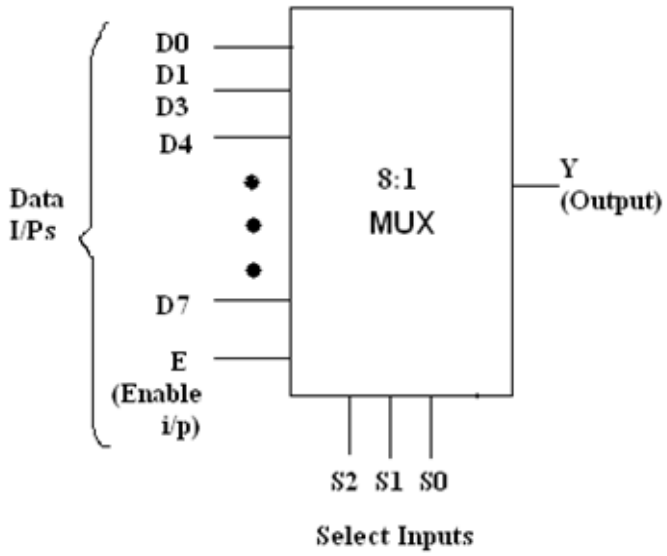
PRINCIPLE:

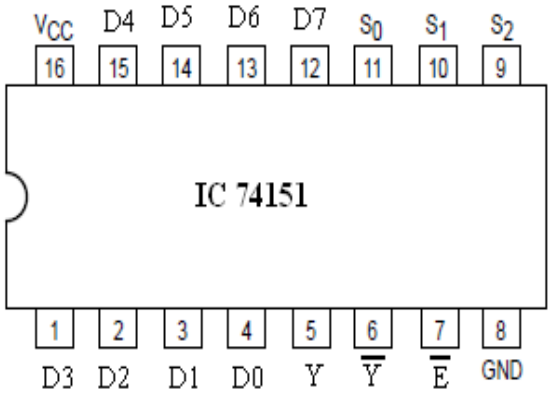
In most of the electronic systems, the digital data is available on more than one line. It is necessary to route this data over a single line. Under such circumstances we require a circuit which selects one of the many inputs at a time. Multiplexer improves the reliability of the digital system because it reduces the number of external wired connections.

An 8:1 Multiplexer has eight data inputs D0 to D7, three select inputs S0 to S2, an enable input and one output.Depending upon the digital code applied at the select inputs one out of n data input is selected & transmitted to a single o/p channel. Normally strobe (G) input is incorporated which is generally active low which enables the multiplexer when it is LOW. Strobe i/p helps in cascading.

IC 74151A is an 8: 1 multiplexer which provides two complementary outputs Y & $\overline{Y}$ The o/p Y is same as the selected i/p & $\overline{Y}$ is its complement.

CIRCUIT DIAGRAM:





[Note: Pin No.7 Enable pin is ACTIVE LOW input. So should be 0 to enable the IC].

ENABLE	SELECT INPUTS			OUTPUT
E (ACTIVE LOW)	S2	S1	S0	Y
1	X	X	X	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7

PROCEDURE:

1. Test the IC74151.
2. Connections are made as per circuit diagram.
3. Keep Enable pin HIGH to see that the output is always 0 for all input combinations.
4. Keep Enable pin LOW and verify the truth table for various combinations of inputs.

RESULT:

Studied about 8x1 MUX 74151 and verified its truth table.

Inference :

1. 74151 can be used to switch any of its 8 inputs to a single output line.
2.

Exp No. 12 Date: / /

STUDY OF RS, D AND T FLIPFLOPS

AIM:

Using NAND gates, construct and verify the truth tables of following Flip-Flops:

(i) RS-Type (ii) D- Type (iii) T- Type. (iv)JK-Type

OBJECTIVE:

- Students will study how to construct flip flops using NAND gates.
- To convert RS to D flipflop and JK to T Flipflop.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	NAND GATE	IC 7400	2
2	NOT GATE	IC 7404	1
2.	3 Input NAND GATE	IC 7410	1
3.	IC TRAINER KIT	-	1
4.	Connecting wires	-	as required

PRINCIPLE:

An SR Flip-Flop can be considered as a basic one-bit memory device that has two inputs, one which will "SET" the device and another which will "RESET" the device and an output Q that will be either at a logic level "1" or logic "0" depending upon this Set/Reset condition. A basic NAND Gate SR flip flop circuit provides feedback from its outputs to its inputs and is commonly

used in memory circuits to store data bits. The term "Flip-flop" relates to the actual operation of the device, as it can be "Flipped" into one logic state or "Flopped" back into another. Flipflop is basically a Bistable Multivibrator.

The symbol for a JK Flip-flop is similar to that of an SR flipflop. Both the S and the R inputs of the previous SR bistable have been replaced by two inputs called the J and K inputs, respectively. The two 2-input NAND gates of the gated SR bistable have now been replaced by two 3-input AND gates with the third input of each gate connected to the outputs Q and $\overline{Q}$.

This cross coupling of the SR Flip-flop allows the previously invalid condition of S = "1" and R = "1" state to be usefully used to turn it into a "Toggle action" as the two inputs are now interlocked. If the circuit is "Set" the J input is inhibited by the "0" status of the Q through the lower AND gate. If the circuit is "Reset" the K input is inhibited by the "0" status of Q through the upper AND gate. When both inputs J and K are equal to logic "1", the JK flip-flop changes state and the truth table for this is given below.

Although this circuit is an improvement on the clocked SR flip-flop it still suffers from timing problems called “race” .Consider clock=1 and J=K=1. This will cause the output to complement again and again. This complement operation continues until the Clock pulse goes back to 0. This undesirable behavior can be eliminated by Edge triggering of JK flip-flop or by using Master slave JK Flip-flops. To avoid this the timing pulse period must be kept as short as possible (high frequency).

T Flip-Flop

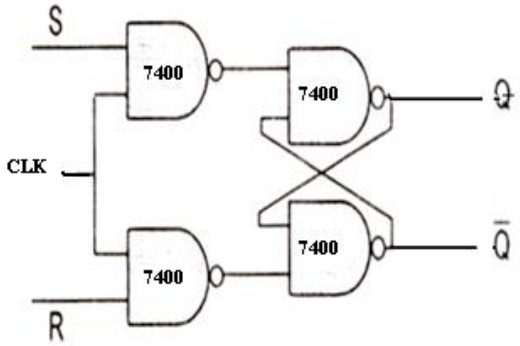
T flip-flops are similar to JK flip-flops. T flip-flops are single input version of JK flip-flops. This modified form of JK flip-flop is obtained by connecting both inputs J and K together. This flip-flop has only one input along with Clock pulse. These flip-flops are called T flip-flops because of their ability to complement its state (i.e.) Toggle. So they are called as Toggle flip-flop.
When T=1 and CP=1, the flip-flop complements its output, regardless of the present state of the Flip-flop. In this case the next state is the complement of the present state.

When T=0, there is no change in the state of the flip-flop (i.e.) the next state is same as the present state of the flip-flop. From the characteristic table and

characteristic equation it is quite evident that when T=0, the next sate is same as the present state.

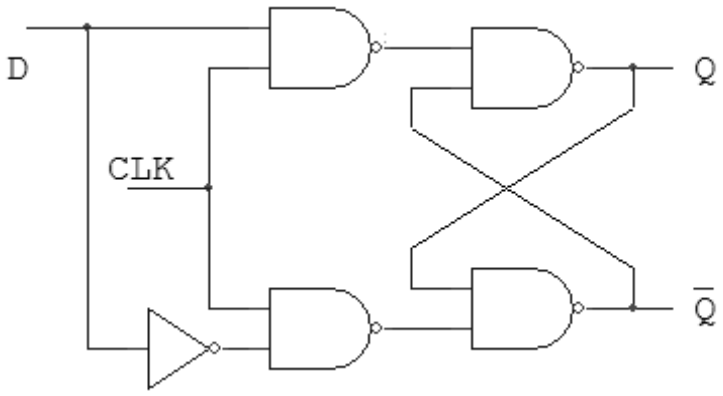
LOGIC DIAGRAM& TRUTH TABLE:

i) SR FLIP-FLOP



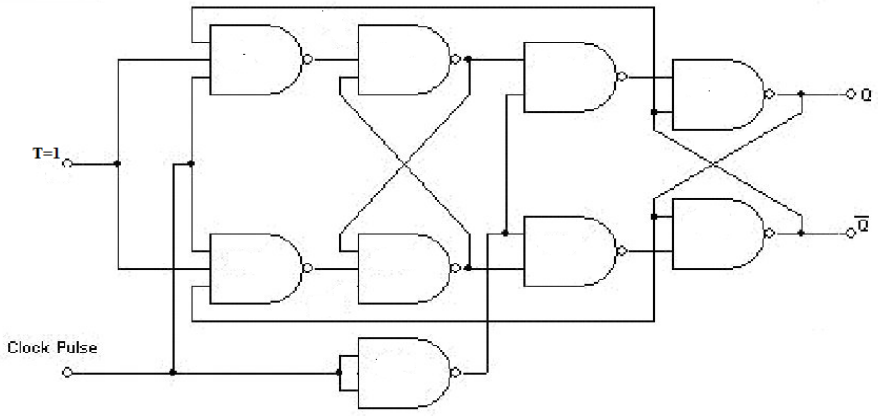
CLK	S	R	Qn+1
0	X	X	Qn (No Change)
1	0	0	No Change
1	0	1	0
1	1	0	1
1	1	1	Invalid

ii) D FLIP-FLOP



CLK	D	Qn+1
0	X	Qn (No Change)
1	0	0
1	1	1

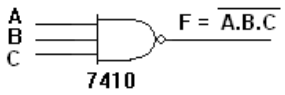
iii) T FLIP-FLOP



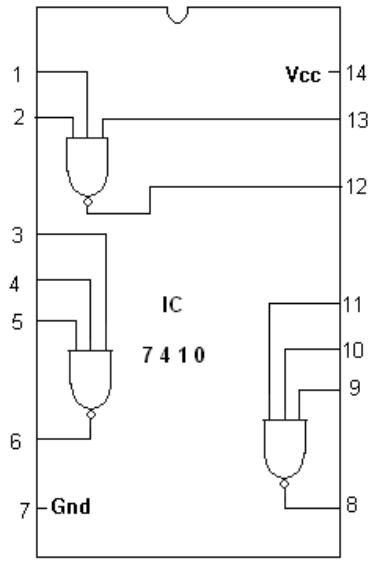
CLK	T	Qn+1	REMARKS
0	X	Qn	(No Change)
1	0	Qn	(No Change)
1	1	$\overline{Qn}$	Toggle

7410 Details:

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

A	B	C	$\overline{A.B.C}$
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

PROCEDURE:

1. Connections are made as per circuit diagram.
2. Verify the truth table for various combinations of inputs.

RESULT

Constructed SR, D, JK and T flip flops and verified the truth tables.

Inference :

1. RS flip flop can be converted to D Flip flop just by connecting R through a NOT gate to S.
2. There exists race around condition in JK flip flop and T flip flop.
3. _____

Exp No. 13 Date: / /

MOD-10 RIPPLE (ASYNCHRONOUS) COUNTER

AIM:

To construct asynchronous (ripple) mod-10 counter using flip-flops.

OBJECTIVE:

- Students will study how to construct a MOD 10 ripple counter.

COMPONENTS AND EQUIPMENTS REQUIRED:

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

PRINCIPLE:

In a digital circuit, counters are used to do 3 main functions: timing, sequencing and counting. Counters are generally made up of flip-flops and logic gates. Digital counters are classified as sequential circuits. The main types of flip-flops used are J-K flip-flops or T flip-flops, which are J-K flip-flops with both J and K inputs tied together.

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 $\overline{Q}$ output of previous stage ie 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.

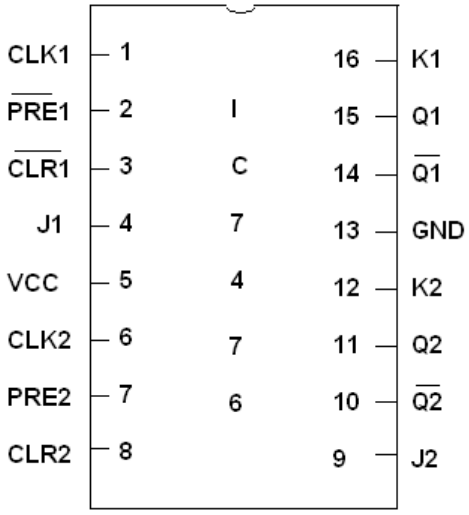
The modulus is the number of unique states through which the counter will sequence. The maximum possible number of states of a counter is $\mathbf{2^n}$ where '**n'** is the number of flip-flops. Counters can be designed to have a number of states in their sequence that is less than the maximum of 2^n . This type of sequence is called a **truncated sequence**. One common modulus for counters with truncated sequences is 10 (Modules10). A decade counter with a count sequence of zero (0000) through 9 (1001) is a BCD decade counter because its 10-state sequence produces the BCD code. To obtain a truncated sequence, it is necessary to force the counter to recycle before going through all of its possible states. A decade counter requires 4 flip-flops. If we take the modulo-16 ripple counter and modified it with additional logic gates it can be made to give a decade (divide-by-10) counter output for use in standard decimal counting and arithmetic circuits. Such counters are generally referred to as Decade Counters. A decade counter requires resetting to zero when the output count reaches the decimal value of 10, ie. when DCBA = 1010 . One way to make the counter recycle after the count of 9 (1001) is to decode the count '10' (1010) with a NAND gate and connect the output of the NAND gate to the clear (CLR) inputs of the flip-flops.

This type of asynchronous counter counts upwards on each leading edge of the input clock signal starting from "0000" until it reaches an output "1010" (decimal 10). Both outputs Q1 and Q3 are now equal to logic "1" and the output from the NAND gate changes state from logic "1" to a logic "0" level and whose output is also connected to the CLEAR (CLR) inputs of all the J-K Flip-flops. This causes all of the Q outputs to be reset back to binary "0000" on the count of 10. Once QB and QD are both equal to logic "0" the output of the NAND gate returns back to a logic level "1" and the counter restarts again from "0000". We now have a decade or Modulo-10 counter.

Similarly we can construct a MOD-N ripple counter by clearing the selected flip flops when it counts N. Students are advised to construct a MOD-12 or MOD-6 counter as an extended experiment.

PIN DIAGRAM:

IC 7476:(Negative edge triggered Dual JK flpflop with active low PRESET and CLEAR inputs)



TRUTH TABLE:

CLK	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

DESIGN:

Design procedure of Mod N asynchronous counter

Usually the count N may not be a power of 2.The general procedure for designing a ripple counter is as follows.

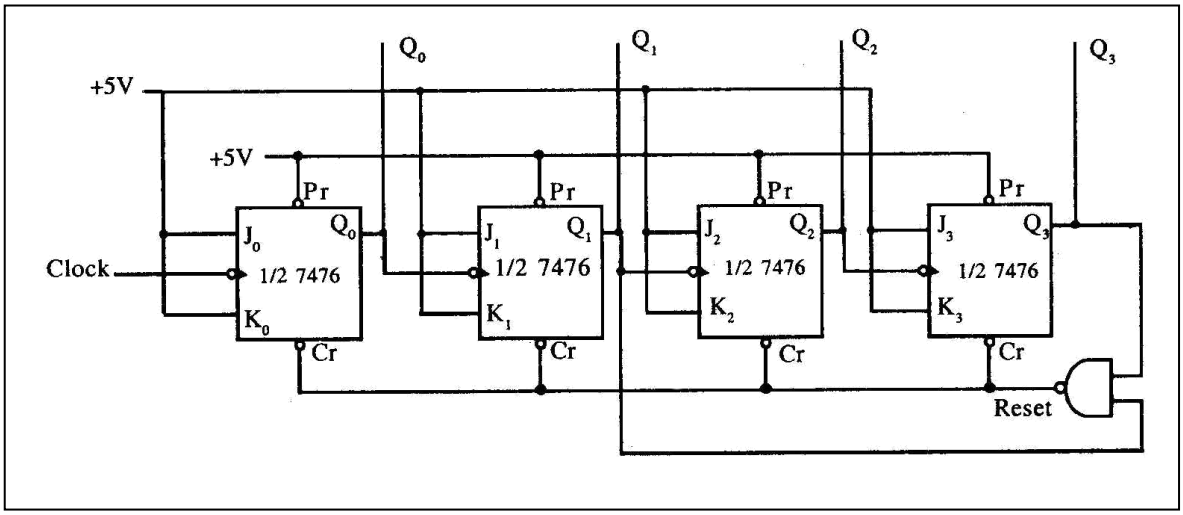
- Find the number of flip flop 'n' such that $\mathbf{2^n > N}$
- Connect the **n** flipflops as a ripple counter.
- Find the binary representation of N
- Put a NAND gate. Each input to the NAND gate is a flip flop output Q which **becomes 1 at the count N.**
- The output of the NAND gate s connected to the clear pin of all the flip flops(because clear input is active LOW in IC 7476)

Example. Design a Mod 10 ripple counter

Binary of decimal 10 is '1010' ie output Q3 and Q1 are '1'.

Hence they are connected to the inputs of a NAND gate and the output of NAND is connected to clear input of all flip flops.

LOGIC DIAGRAM FOR MOD - 10 RIPPLE COUNTER:



PROCEDURE:

1. Connections are given as per circuit diagram.
2. Logical inputs are given as per truth table.
3. Observe the output and verify the truth table.

RESULT:

A MOD 10 RIPPLE/ASYNCHRONOUS counter is constructed and verified the truth table.

Inference :

1. Using Clear input we can reset a counter.
2. We can construct any MOD-N counter by clearing the appropriate flip flops.
3. Can construct Mod-N counter where N cannot be written as power of 2 eg Mod-3,Mod-5,Mod-7,Mod-9,Mod-11,Mod-13 etc
4. Student can construct a single circuit for Mod-N counter

Exp No. 14 Date: / /

SYNCHRONOUS COUNTERS

AIM:

To construct synchronous mod-8 counters using JK flipflops.

OBJECTIVES:

Students will get an idea about

- Understanding the operation and characteristics of synchronous counters
- Analyze counter circuits
- Determine the sequence of a counter
- Determine the modulus of a counter sequences
- Understand different applications of flip flops.
- Realize the frequency division of signals.

COMPONENTS AND EQUIPMENTS:

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

THEORY:

A synchronous counter, in contrast to an asynchronous counter, is one whose output bits change state simultaneously, in synchronous with the clock signal with no ripple. The" clock" pulses are applied to all the flip-flops in counter simultaneously. Depending on the way in which the counting progresses ,the synchronous and asynchronous counters are classified into (i)up counters (ii)down counters(iii) up/down counters

A synchronous binary counter counts from 0 to 2^n-1, where 'n' is the number of bits/flip-flops in the counter. Each flip-flop is used to represent one bit.

Modulus (MOD) is the number of states it counts in a complete cycle before it goes back to the initial state.

Up Counter:

The binary output is taken from the Q outputs of the flip-flops. In FF0 the J and K inputs are permanently wired to logic 1, so Q_0 will change state (toggle) on each clock pulse. This provides the 'ones' count for the least significant bit (LSB).

On FF1 the J1 and K1 inputs are both connected to Q_0 so that FF1 output will only be in toggle mode when Q_0 is also at logic 1. As this only happens on alternate clock pulses, Q_1 will only toggle on even numbered clock pulses giving a 'twos' count on the Q_1 output.

FF2 is put into toggle mode by making J2 and K2 logic 1,only whenQ_0 and Q_1 are at logic 1.

In the first flip-flop, J = K = 1 .In subsequent flipflops, J = K, but the logical value is determined by an AND gate. Thus,J1=K1=Q0 and J2=K2 = Q1·Q0

Down Counter

As every Q output on the JK flip-flops has its complement on $\overline{Q}$, to convert the up counter into the down counter, take the JK inputs for FF1 from the $\overline{Q}$ output of FF0 instead of the Q output. The AND gate takes its inputs from the $\overline{Q}$ outputs of FF0 and FF1.

PROCEDURE:

1. Test all the ICs manually/ using IC tester.
2. Connections are made as in the logic circuit diagram and give V_{CC} and the ground.
3. Connect the clock signal (<16Hz) input and output to LEDs.
4. Observe the count and verify the truth table.

DESIGN

A simple way of implementing the logic for each bit of an ascending counter is for each bit to toggle when all of the less significant bits are at a logic high state. For example, bit 1 toggles when bit 0 is logic high; bit 2 toggles when both bit 1 and bit 0 are logic high; bit 3 toggles when bit 2, bit 1 and bit 0 are all high; and so on

A summary of steps used in the design of this counter follows. In general, these steps can be applied to any sequential circuit:

1. Specify the counter sequence and draw a state diagram.
2. Derive a next-state table from the state diagram.
3. Develop a transition table showing the flip-flop inputs required for each transition, The transition table is always the same for a given type of flip-flop.

4. Transfer the J & K states from the transition table to Karnaugh maps. There is a Karnaugh map for each input of each flip-flop.
5. Group the Karnaugh map cells to generate and derive the logic expression for each flipflop input.
6. Implement the expressions with combinational logic, and combine with the flip-flops to create the counter

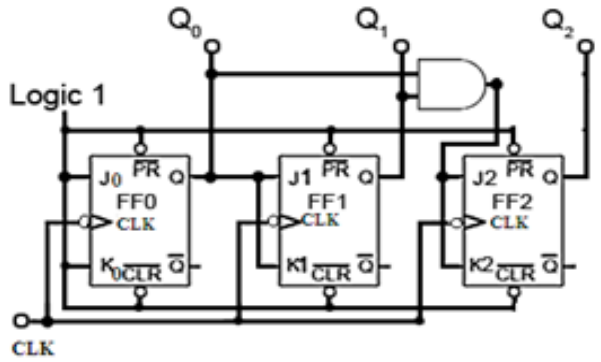
FUNCTION TABLE

3 bit (mod -8) synchronous up counter using JK flipflop

CLOCK	Q2	Q1	Q0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8(recycles)	0	0	0

CIRCUIT DIAGRAM:

3 bit (mod -8) synchronous **up** counter using JK flipflop

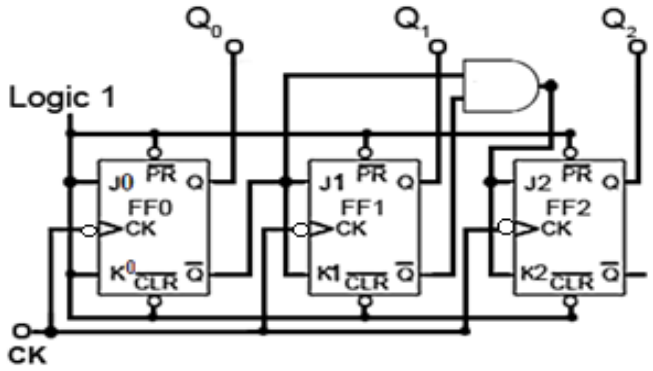


FUNCTION TABLE

3 bit (mod -8) synchronous **down** counter using JK flipflop

CLOCK pulse	Q2	Q1	Q0
Initially	**1**	**1**	**1**
1	**1**	**1**	**0**
2	**1**	**0**	**1**
3	**1**	**0**	**0**
4	**0**	**1**	**1**
5	**0**	**1**	**0**
6	**0**	**0**	**1**
7	**0**	**0**	**0**
8(recycles)	**1**	**1**	**1**

3 bit (mod -8) synchronous **down** counter using JK flipflop



RESULT:

1. Studied 3 bit synchronous up counter .
2. Studied 3 bit synchronous down counter.

INFERENCE:

After studying this section, students should be able to:

1. Understand the operation of synchronous counters. The count sequence is controlled using logic gates.

2. Describe common control features used in synchronous counters such as Preset and Clear.

3. To count the number of times that a certain event takes place; the occurrence of event to be counted is represented by the input signal to the counter
4. To control a fixed sequence of actions in a digital system
5. To generate timing signals
6. To generate clocks of different frequencies

Exp No. 15 Date: / /

STUDY OF COUNTER ICs 7490 & 7492.

AIM:

To familiarize with counter ICs 7490 and 7492.

OBJECTIVE:

- Students will understand about commonly used counter ICs 7490 and 7492.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	COUNTER IC	IC 7490	1
2.	COUNTER IC	IC 7492	1
3.	IC TRAINER KIT	-	1
4.	Connecting wires	-	as required

PRINCIPLE:

This 749x counters from the TTL-series or HC-series of (small-scale) integrated circuits. The most popular counter ICs from the 74xx series are the following, each of which is included in the applet:

- 7490, a 1:10 counter which can be split into 1:2 and 1:5
- 7492, a 1:12 counter which can be split into 1:2 and 1:6
- 7493, a 1:16 counter which can be split into 1:2 and 1:8

All three ICs are based on JK flipflops and feature asynchronous reset inputs. The first counter stage uses separate input and output pins, and can be used as a standalone 1:2 counter stage. The remaining three stages are connected internally, so that the carry and feedback paths are not accessible from the outside.

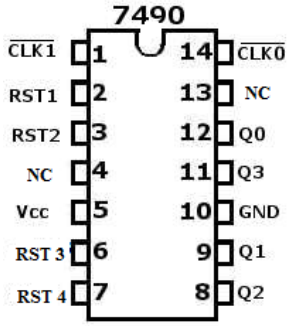
IC 7490 is a decade counter which drives input by 10 and provides BCD outputs 0 to 9, this is also called as decimal counter. This counter comprises of a divide-by 2 and divide-by 5 counters. To use as decade counter we have to cascade divide-by 2 and divide-by 5. Outputs Q0 to Q3 are BCD outputs, inputs A and B are clock inputs to the, divide-by 2 and divide-by 5 counters respectively. R01 and R02 are the reset inputs, when these are activated

counter output goes to 0000. If two 7490 are connected in a manner that input of one becomes the output of other, the second IC will receive a pulse on every tenth count and will reset at every hundredth count.

IC 7492 is an asynchronous divide by 12 counter. If the clock is applied at input B, (pin 1) and outputs are taken from QD,QC and QB, it will function as a MOD-6 counter.

If QA is connected to B input (Pin 12 to Pin 1), and clock is applied at A input, it will function as a MOD-12 counter.

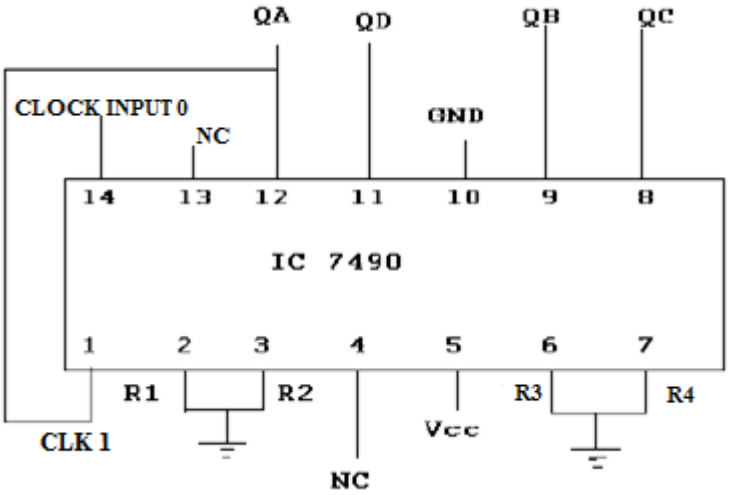
PIN DIAGRAM:



IC 7490 Pin Description:

Pin No	Function	Name
1	Clock input 1	$\overline{CLK1}$
2	Reset1	R1
3	Reset2	R2
4	Not connected	NC
5	Supply voltage; 5V (4.75V – 5.25V)	Vcc
6	Reset3	R3
7	Reset4	R4
8	Output 3, BCD Output bit 2	Q_C
9	Output 2, BCD Output bit 1	Q_B
10	Ground (0V)	Ground
11	Output 4, BCD Output bit 3	Q_D
12	Output 1, BCD Output bit 0	Q_A
13	Not connected	NC
14	Clock input 0	$\overline{CLK0}$

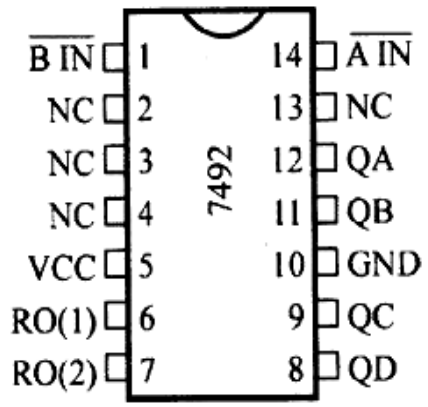
CIRCUIT DIAGRAM

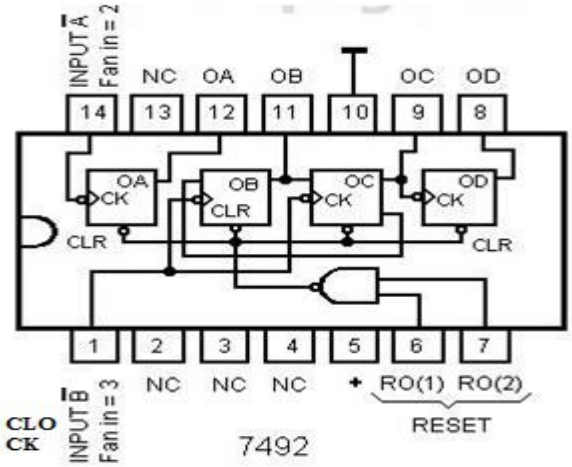


Truth Table of Decade counter USING 7490:

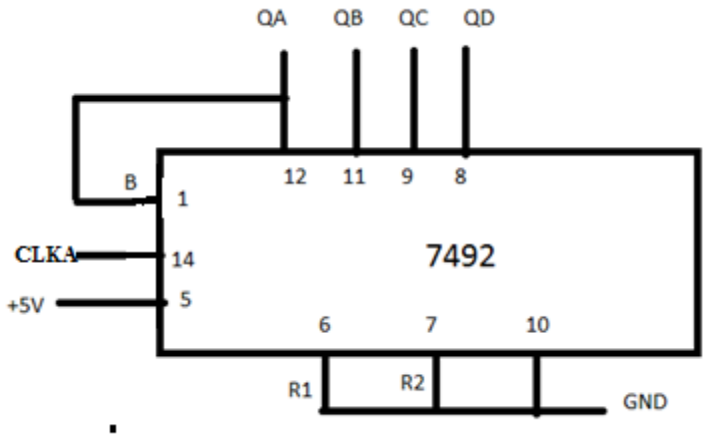
Clock	QD	QC	QB	QA
0	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**1**
2	**0**	**0**	**1**	**0**
3	**0**	**0**	**1**	**1**
4	**0**	**1**	**0**	**0**
5	**0**	**1**	**0**	**1**
6	**0**	**1**	**1**	**0**
7	**0**	**1**	**1**	**1**
8	**1**	**0**	**0**	**0**
9	**1**	**0**	**0**	**1**

Pin Diagram and internal diagram of IC 7492





IC 7492 as Divide by 12 Counter:



Truth Table of 7492 as Divide by 12 counter:

Clock	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0

3	**0**	**0**	**1**	**1**
4	**0**	**1**	**0**	**0**
5	**0**	**1**	**0**	**1**
6	**0**	**1**	**1**	**0**
7	**0**	**1**	**1**	**1**
8	**1**	**0**	**0**	**0**
9	**1**	**0**	**0**	**1**
10	**1**	**0**	**1**	**0**
11	**1**	**0**	**1**	**1**

PROCEDURE:

1. Check the ICs.
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 output.
5. Vary the clock frequency to see the difference in the output.
6. Constructed different types of 4 bit shift registers and verified the truth tables.
7. Constructed different types of 4 bit shift registers and verified the truth tables.

RESULT:

Familiarized with counter ICs 7490 and 7492.

Exp No. 16 Date: / /

4 BIT SHIFT REGISTER

AIM:

To construct following 4 bit shift registers.

(i) Serial in serial out
(ii) Serial in parallel out
(iii) Parallel in serial out
(iv) Parallel in parallel out

OBJECTIVE:

- Students will study how to construct a different shift registers.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	D FLIP FLOP	IC 7474	2
2.	OR GATE	IC 7432	1
3.	IC TRAINER KIT	-	1
4.	Connecting wires	-	as required

PRINCIPLE:

The Shift Register is another type of sequential logic circuit that can be used for the storage or the transfer of data in the form of binary numbers. This sequential device loads the data present on its inputs and then moves or "shifts" it to its output once every clock cycle, hence the name "shift register".

A shift register basically consists of several single bit "D-Type Data Latches", one for each data bit, either a logic "0" or a "1", connected together

in a serial type daisy-chain arrangement so that the output from one data latch becomes the input of the next latch and so on.

Data bits may be fed in or out of a shift register serially, that is one after the other from either the left or the right direction, or all together at the same time in a parallel configuration.

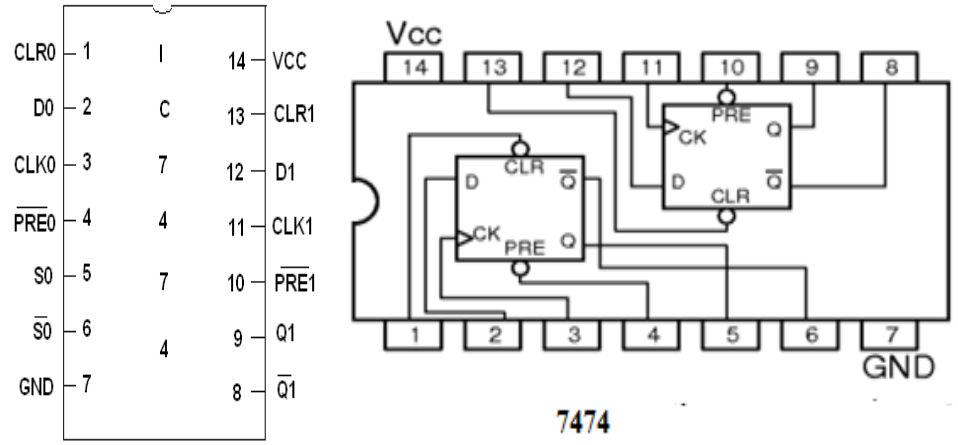
The number of individual data latches required to make up a single **Shift Register** device is usually determined by the number of bits to be stored with the most common being 8-bits (one byte) wide constructed from eight individual data latches.

Shift Registers are used for data storage or for the movement of data and are therefore commonly used inside calculators or computers to store data such as two binary numbers before they are added together, or to convert the data from either a serial to parallel or parallel to serial format. The individual data latches that make up a single shift register are all driven by a common clock (Clk) signal making them synchronous devices.

Shift register IC's are generally provided with a *clear* or *reset* connection so that they can be "SET" or "RESET" as required. Generally, shift registers operate in one of four different modes with the basic movement of data through a shift register being:

- Serial-in to Serial-out (SISO) - the data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.
- Serial-in to Parallel-out (SIPO) - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.
- Parallel-in to Serial-out (PISO) - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.
- Parallel-in to Parallel-out (PIPO) - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

PIN DIAGRAM:

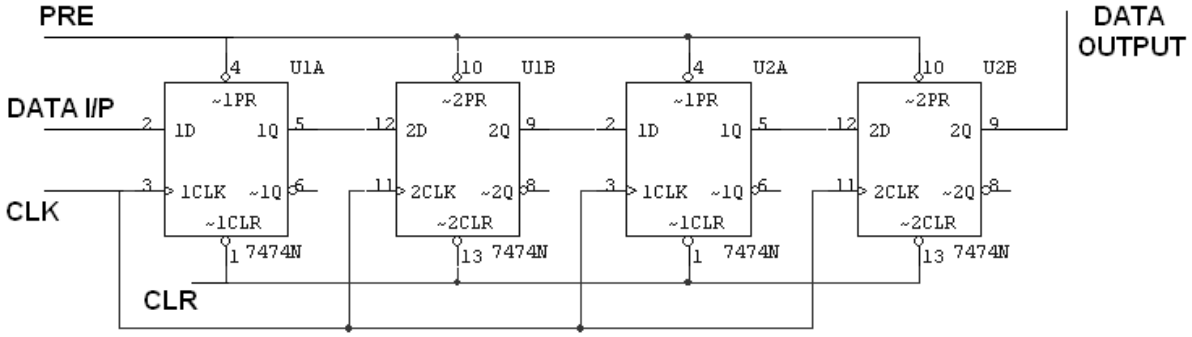


SERIAL IN SERIAL OUT:

TRUTH TABLE:

CLK	Serial in	Serial out
1	1	0
2	0	0
3	0	0
4	1	1
5	X	0
6	X	0
7	X	1

LOGIC DIAGRAM:



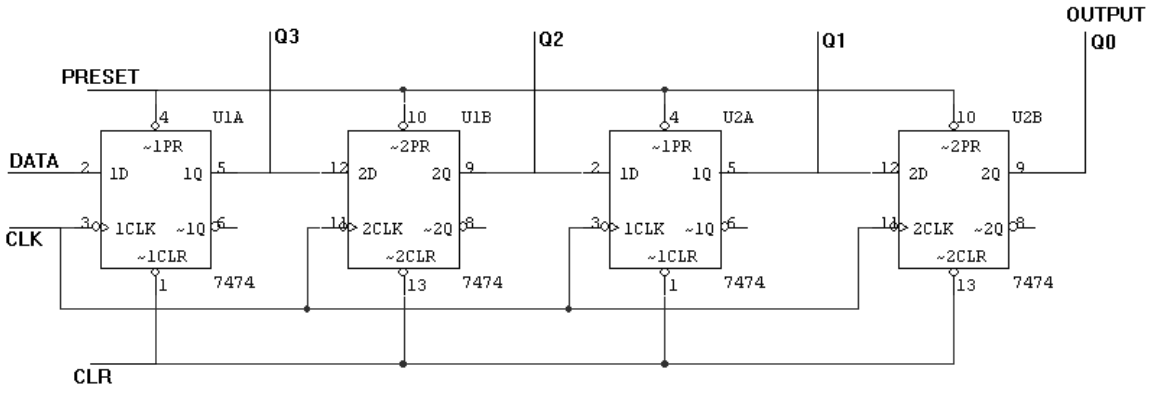
On each clock pulse, the input bit will shift towards right. Thus it requires 4 clock pulse to reach the 4th flip flop's output.

SERIAL IN PARALLEL OUT:

TRUTH TABLE:

CLK	DATA	OUTPUT			
		Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

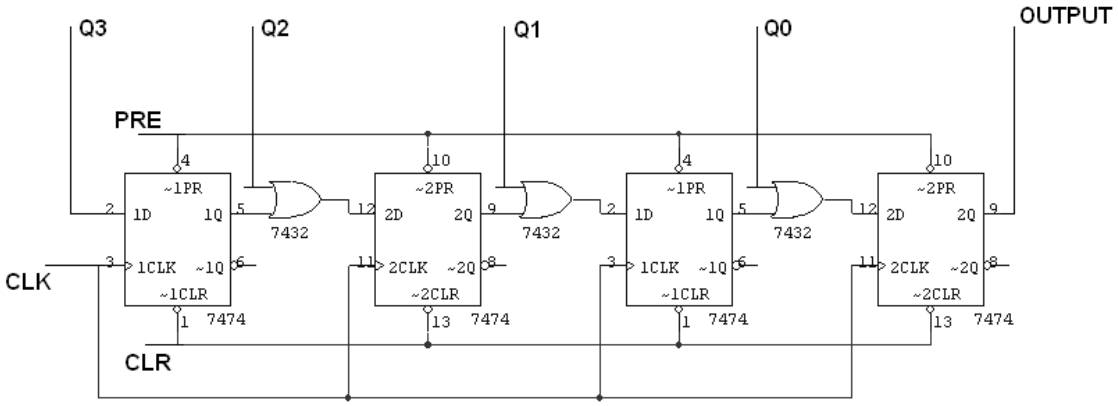
We can see that, the working condition is the same as SISO, but the output is taken from each flip flop's output.



PARALLEL IN SERIAL OUT:

TRUTH TABLE:

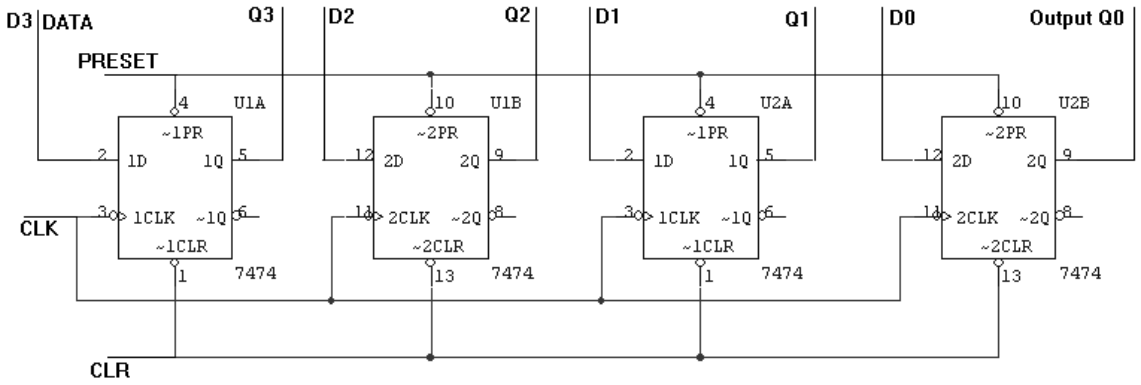
CLK	Q3	Q2	Q1	Q0	O/P
0	1	0	0	1	1
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1



PARALLEL IN PARALLEL OUT:

TRUTH TABLE:

CLK	DATA INPUT				OUTPUT			
	D_A	D_B	D_C	D_D	Q_A	Q_B	Q_C	Q_D
1	1	0	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0



PROCEDURE:

(i) Connections are given as per circuit diagram.
(ii) For each shift register, inputs and outputs are connected with at most care.
(iii) The input bits to the shift registers are given.
(iv) Observe the output and verify the truth table.
(v) Continue this for different inputs.

RESULT:

Constructed different types of 4 bit shift registers and verified the truth tables.

Inference :

1. Shift registers are the important modules for Data transfer in Digital communication
2. These are used for storage and transfer of data inside a microprocessor, microcontroller etc.

Exp No. 17 Date: / /

SHIFT REGISTER ICs

AIM:

To study the operation of shift register ICs in all its modes i.e.SIPO/SISO, PISO/PIPO.

OBJECTIVES:

- To verify the different functions of IC7495 such as clearing the register, serial input, parallel input, serial/parallel output etc.

.

COMPONENTS AND EQUIPMENTS:

SL No.	COMPONENT	IC NUMBER	QTY
1.	4 BIT SHIFT REGISTER IC	IC 7495	1
2	IC trainer kit		1
3	Connecting wires		

THEORY:

The 7495 is a 4-Bit Shift Register with serial and parallel synchronous operating modes. The serial shift right and parallel load are activated by separate clock inputs which are selected by a mode control input.

It has a Serial and four Parallel (A–D) Data inputs and four Parallel Data outputs (QA–QD). The serial or parallel mode of operation is controlled by a Mode Control input and two Clock Inputs (CLK1) and (CLK2). The serial (right-shift) or parallel data transfers occur synchronous with the HIGH to LOW transition of the selected clock input.

When the Mode Control input is HIGH, CLK2 is enabled. A HIGH to LOW transition on enabled CLK2 transfers parallel data from the A–D inputs to the QA–QD outputs.

When the Mode Control input is LOW, CLK1 is enabled. A HIGH to LOW transition on enabled CLK1 transfers the data from Serial input to QA and shifts the data in QA to QB, QB to QC, and QC to QD respectively (right-shift). A left-shift is accomplished by externally connecting QD to C, QC to B, and QB to A, and operating the 7495 in the parallel mode (Mode Control input = HIGH).

For normal operation, Mode Control input should only change states when both Clock inputs are LOW. However, changing Mode Control input from LOW to HIGH while CLK2 is HIGH, or changing Mode Control input from HIGH to LOW while CLK1 is HIGH and CLK2 is LOW will not cause any changes on the register outputs.

PROCEDURE:

Serial In Parallel Out (SIPO):-

1. Connections are made as per circuit diagram.
2. Apply the data at serial i/p
3. Apply one clock pulse at clock 1 (Right Shift) observe this data at QA.
4. Apply the next data at serial i/p.
5. Apply one clock pulse at clock 2, observe that the data on QA will shift to QB and the new data applied will appear at QA.
6. Repeat steps 2 and 3 till all the 4 bits data are entered one by one into the shift register.

Serial In Serial Out(SISO):-

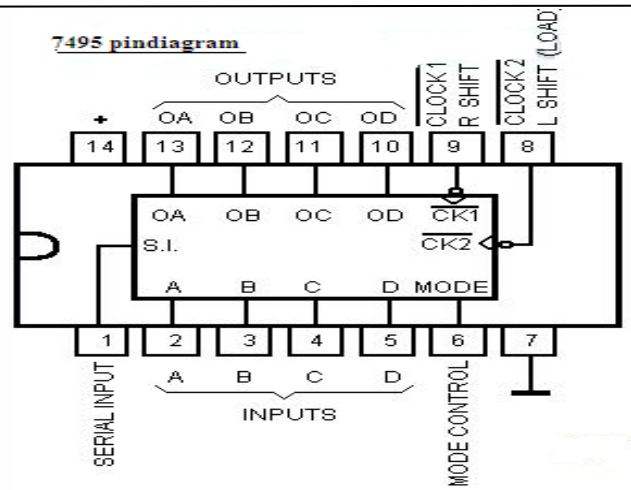
1. Connections are made as per circuit diagram.
2. Load the shift register with 4 bits of data one by one serially.
3. At the end of 4th clock pulse the first data ‘d0’ appears at QD.
4. Apply another clock pulse; the second data ‘d1’ appears at QD.
5. Apply another clock pulse; the third data appears at QD.
6. Application of next clock pulse will enable the 4th data ‘d3’ to appear at QD. Thus the data applied serially at the input comes out serially at QD

Parallel In Serial Out (PISO):-

1. Connections are made as per circuit diagram.
2. Apply the desired 4 bit data at A, B, C and D.
3. Keeping the mode control M=1 apply one clock pulse. The data applied at A, B, C and D will appear at QA, QB, QC and QD respectively.
4. Now mode control M=0. Apply clock pulses one by one and observe the Data coming out serially at QD

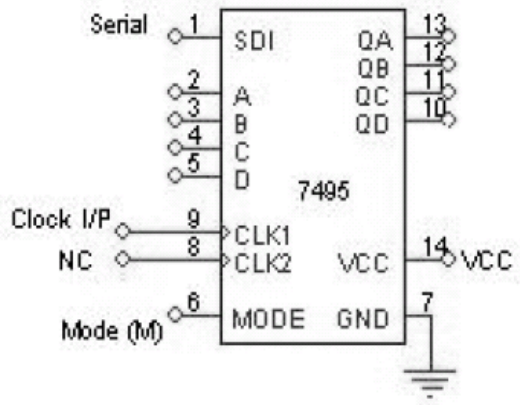
Parallel In Parallel Out (PIPO):-

1. Connections are made as per circuit diagram.
2. Apply the 4 bit data at A, B, C and D.
3. Apply one clock pulse at Clock 2 (Note: Mode control M=1).
4. The 4 bit data at A, B, C and D appears at QA, QB, QC and QD respectively.



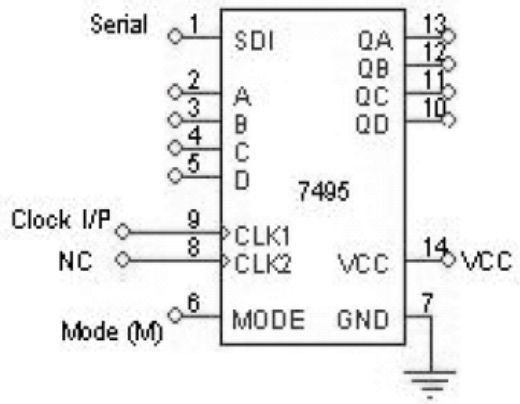
LOGIC DIAGRAM:

SIPO (Right Shift):-



Clock	Serial i/p	QA	QB	QC	QD
1	0	0	X	X	X
2	1	1	0	X	X
3	1	1	1	0	X
4	1	1	1	1	0

SISO:-



Clock	Serial i/p	QA	QB	QC	QD
1	do=0	0	X	X	X
2	d1=1	1	0	X	X
3	d2=1	1	1	0	X
4	d3=1	1	1	1	0=do
5	X	X	1	1	1=d1
6	X	X	X	1	1=d2
7	X	X	X	X	1=d3

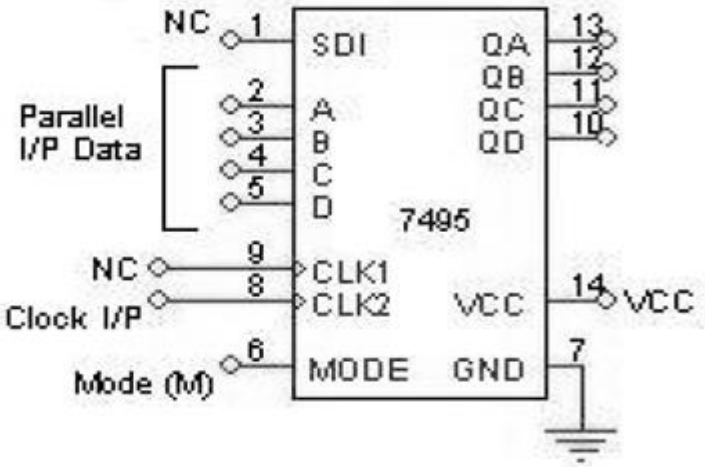
LOGIC SYMBOL

Note:

To work as a PIPO, keep the mode control M=1 and apply one clock pulse. The data applied at A, B, C and D will appear at QA, QB, QC and QD respectively.(PIPO)

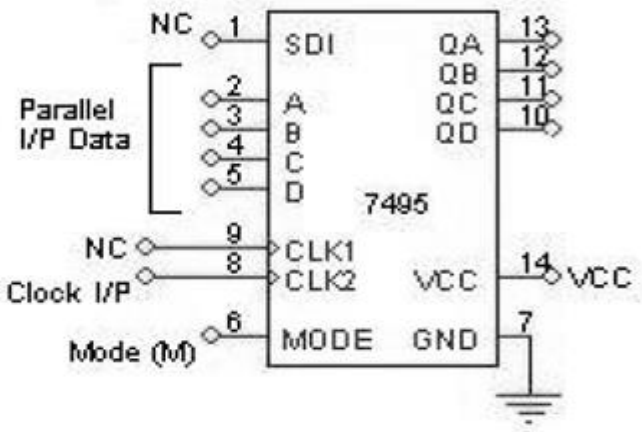
To get PISO ,change the mode control M=0. Apply clock pulses one by one and observe the data coming out serially at

PISO



Mode	Clock	Parallel i/p				Parallel o/p			
		A	B	C	D	QA	QB	QC	QD
1	1	1	0	1	1	1	0	1	1
0	2	X	X	X	X	X	1	0	1
0	3	X	X	X	X	X	X	1	0
0	4	X	X	X	X	X	X	X	1

PIPO:-



Clock	Parallel i/p				Parallel o/p			
	A	B	C	D	QA	QB	QC	QD
1	1	0	1	1	1	0	1	1

QD

RESULT:

Shift registers using IC 7495 in all its modes i.e.SIPO/SISO, PISO/PIPO are verified.

INFERENCE:

1. 4-bit Parallel-Access Shift Register IC s 4 bit parallel in /serial out7494, 74195
2. 4 bit bidirectional 74194, can also be studied.

Exp No. 18 Date: / /

JOHNSON COUNTER

AIM:

To construct a Johnson Counter and verify the truth table.

OBJECTIVE:

- To construct a Johnson counter.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	D FLIP FLOP	IC 7474	2
2.	IC TRAINER KIT	-	1
3.	Connecting wires	-	as required

PRINCIPLE:

A **shift register counter** is basically a shift register with the serial output connected back to the serial input to produce special sequences. These devices are often classified as counters because they exhibit a specified sequence of states. Two of the most common types of shift register counters, the Johnson counter and the ring counter.

JohnsonCounter

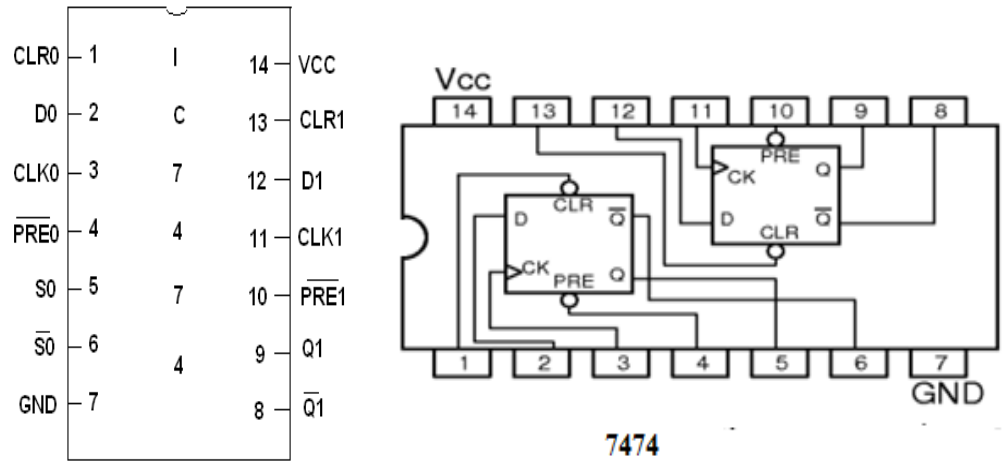
The Johnson digital counter or Twisted Ring Counter is a synchronous shift register with feedback from the inverted output ($\overline{Q}$) of the last flip-flop. $\overline{Q}$ of the last flip flop is connected back to the input D of the first flip-flop. This inversion of Q before it is fed back to input D causes the counter to "count" in a special way. The main benefit of this type of counter is that it only needs half the number of flip flops compared to that of standard ring counter to represent many states. So an n-stage Johnson counter gives a sequence

of 2n different states and can therefore be treated as a “Mod 2n counter” whereas an n-stage ring counter has only n states that is “Mod n counter”.

It can be implemented using D-type flip-flops (or JK-type flip-flops). The output of each flip flop is connected to the input of the next. The complementary output $\overline{Q}$ of the last flipflop is connected to the first input.

In order to make Q1 high to begin, we should keep $\overline{PRE\ 1}$ = 0 for a time period which is less than the clock duration. After that keep $\overline{PRE\ 1}$ = 1 for proper working.

PIN DIAGRAM AND INTERNAL LOGIC DIAGRAM OF IC 7474

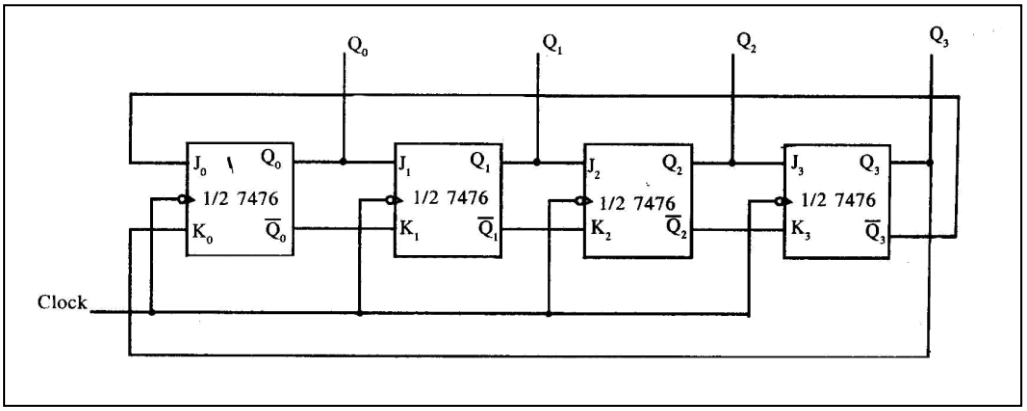


TRUTH TABLE:

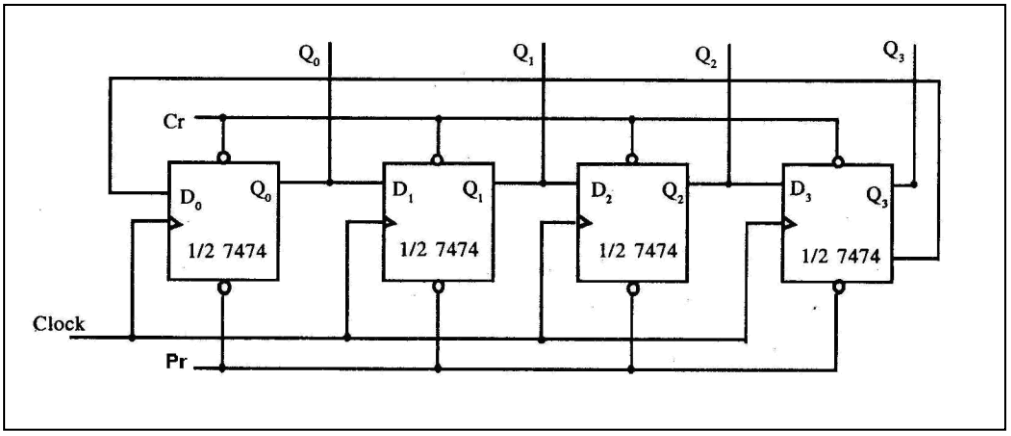
CLK	Q0	Q1	Q2	Q3
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

LOGIC DIAGRAM FOR JOHNSON COUNTER

Johnson Counter using JK flip flops



Johnson Counter using D flip flops



PROCEDURE:

Connections are given as per circuit diagram.

(i) In the beginning, clear all flip flops using CLR input.

(ii) In order to make Q1 high to begin, we should keep $\overline{\text{PRE 1}}$ = 0 for a time period which is less than the clock duration. After that keep $\overline{\text{PRE 1}}$ = 1 fo.r proper working

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

RESULT:

Constructed a Johnson counter and the truth table is verified.

Inference :

1. Many beautiful patterns can be generated using Johnson counter.
2.

Exp No. 19 Date: / /

RING COUNTER

AIM:

To construct a Ring Counter and verify the truth table.

OBJECTIVE:

- To construct a Ring counter.

COMPONENTS AND EQUIPMENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	D FLIP FLOP	IC 7474	2
2.	IC TRAINER KIT	-	1
3.	Connecting wires	-	as required

PRINCIPLE:

A ring counter is formed by feeding the output of a shift register to its own input. Here the last output ie. Q_D in a shift register is connected back to the serial input. The data pattern enclosed within the shift register will re-circulate with respect to the clock pulse. Ring counter is one of the shift register applications. A ring counter has N states where 'N' is the number of flip-flops.

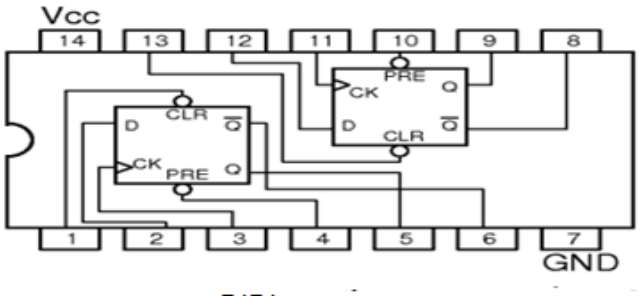
The synchronous **Ring Counter** is preset so that exactly one data bit in the register is set to logic "1" with all the other bits reset to "0". To achieve this, a "CLEAR" signal is firstly applied to all the flip-flops together in order to "RESET" their outputs to a logic "0" level and then a "PRESET" pulse is applied to the input of the first flip-flop (FFA) before the clock pulses are applied. This then places a single logic "1" value into the circuit of the ring counter.

So on each successive clock pulse, the counter circulates the same data bit between the four flip-flops over and over again around the "ring" every

fourth clock cycle. But in order to cycle the data correctly around the counter we must first “load” the counter with a suitable data pattern as all logic “0’s” or all logic “1’s” outputted at each clock cycle would make the ring counter invalid.

PIN DIAGRAM

IC 7474(positive edge triggered dual D flipflop with active LOW, PRESET and CLEAR inputs)

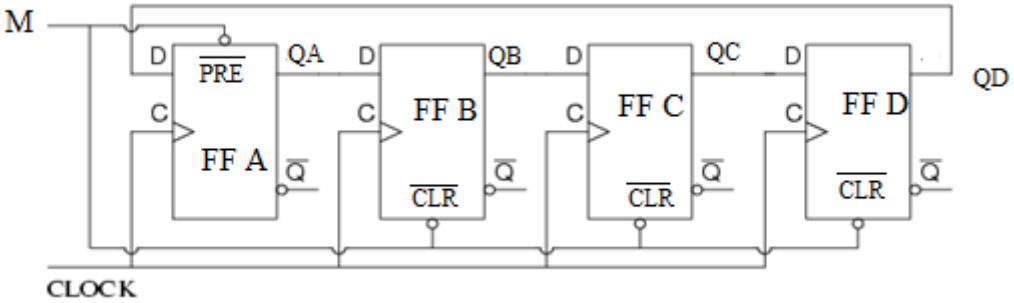


7474

TRUTH TABLE:

Clock	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0

LOGIC DIAGRAM



PROCEDURE:

1. Connections are made as per the circuit diagram.

2. Apply the data 1000 at flipflop A, B, C and D respectively.

(i) 3. In the beginning, clear all flip flops using $\overline{CLR}$ input by applying '0' to all clear inputs. After that, keep it at logic state '1'. In order to make Q_A high to begin, we should keep $\overline{PRE}$ of flipflop A at " 0 " for a time period which is less than the clock duration. After that keep $\overline{PRE}$ = 1 for proper working

For this,keep the mode input M = 1, apply one clock pulse. (<16Hz).

4. Now the mode M is made 0 and clock pulses are applied one by one and

the truth table is verified.

5.Observe the output and verify the truth table.

RESULT:

Constructed a Ring counter and the truth table is verified.

Inference :

1. Many beautiful patterns can be generated using Ring counter.
2.

This page is intentionally kept blank to preserve page format.

GOVERNMENT POLYTECHNIC COLLEGE, PERUMBAVOOR
Koovappady P O, Perumbavoor, Ernakulam – 683544
www.gptcperumbavoor.ac.in