GOVERNMENT MODEL ENGINEERING COLLEGE,

THRIKKAKARA, ERNAKULAM

FOURTH SEMESTER ELECTRONICS & COMMUNICATION ENGINEERING

LIST OF EXPERIMENTS FOR EC 230 LOGIC CIRCUIT DESIGN LAB

Experiment No.	Experiment
1	Study of basic Logic Gates and Universal Gates
2	Realization of functions using basic & universal gates (SoP & PoS forms)
3	Design and realization of half / full adder and half / full subtractor using basic gates and universal gates
4	Study of IC 7483 and realization of BCD adder using 7483
5	Code Converters (Binary to Gray and Gray to Binary converters)
6	Multiplexers and De-multiplexers using gates and ICs (74150, 74154 etc.)
7	Realization of combinational circuits using Mux & Demux
8	Study of flip-flops : SR, D, T, JK and Master Slave JK using NAND gates and Study of flip-flop ICs
9	Ring Counter and Johnson Counter
10	Asynchronous and Synchronous Counters
11	Study of Counter ICs : 7490, 7492, 7493, 74192
12	Random Sequence Generator
13	BCD-to-7 segment Decoder/Driver to drive LED display
14	TTL / CMOS Characteristics

The Breadboard

The breadboard consists of two terminal strips and two bus strips (often broken in the centre). Each bus strip has two rows of contacts. Each of the two rows of contacts are a node. That is, each contact along a row on a bus strip is connected together (inside the breadboard). Bus strips are used primarily for power supply connections, but are also used for any node requiring a large number of connections. Each terminal strip has 60 rows and 5 columns of contacts on each side of the centre gap. Each row of 5 contacts is a node.

You will build your circuits on the terminal strips by inserting the leads of circuit components into the contact receptacles and making connections with 22-26 gauge wire. There are wire cutter/strippers and a spool of wire in the lab. It is a good practice to wire +5V and 0V power supply connections to separate bus strips.

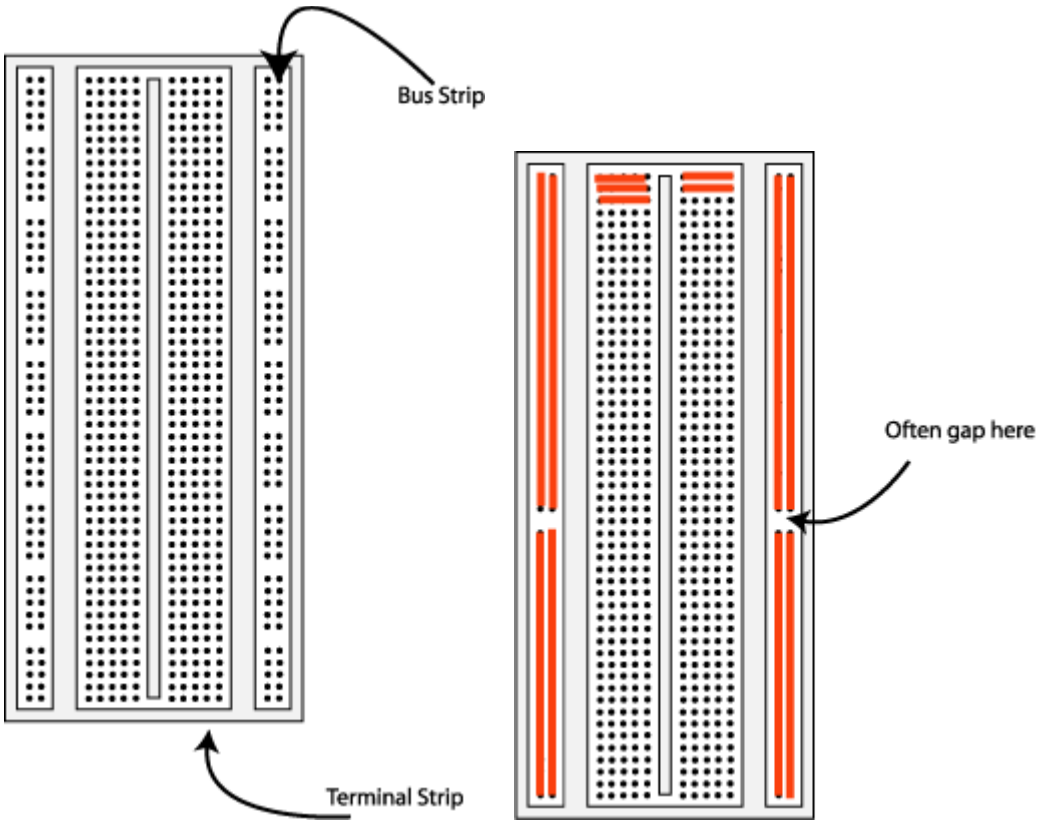


Fig1. The breadboard. The lines indicate connected holes.

The 5V supply MUST NOT BE EXCEEDED since this will damage the ICs (Integrated circuits) used during the experiments. Incorrect connection of power to the ICs could result in them exploding or becoming very hot - with the possible serious injury occurring to the people working on the experiment! Ensure that the power supply polarity and all components and connections are correct before switching on power.

Building the Circuit

Throughout these experiments we will use TTL chips to build circuits. The steps for wiring a circuit should be completed in the order described below:

1. Turn the power (Trainer Kit) off before you build anything!
2. Make sure the power is off before you build anything!
3. Connect the +5V and ground (GND) leads of the power supply to the power and ground bus strips on your breadboard.
4. Plug the chips you will be using into the breadboard. Point all the chips in the same direction with pin 1 at the upper-left corner. (Pin 1 is often identified by a dot or a notch next to it on the chip package)
5. Connect +5V and GND pins of each chip to the power and ground bus strips on the breadboard.
6. Select a connection on your schematic and place a piece of hook-up wire between corresponding pins of the chips on your breadboard. It is better to make the short connections before the longer ones. Mark each connection on your schematic as you go, so as not to try to make the same connection again at a later stage.
7. Get one of your group members to check the connections, **before you turn the power on**.
8. If an error is made and is not spotted before you turn the power on. Turn the power off immediately before you begin to rewire the circuit.
9. At the end of the laboratory session, collect you hook-up wires, chips and all equipment and return them to the demonstrator.
10. Tidy the area that you were working in and leave it in the same condition as it was before you started.

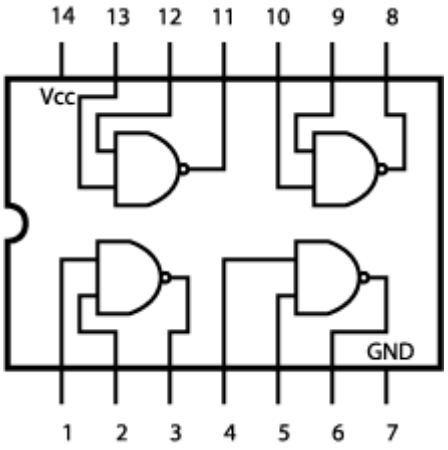
Common Causes of Problems

1. Not connecting the ground and/or power pins for all chips.
2. Not turning on the power supply before checking the operation of the circuit.
3. Leaving out wires.
4. Plugging wires into the wrong holes.
5. Driving a single gate input with the outputs of two or more gates
6. Modifying the circuit with the power on.

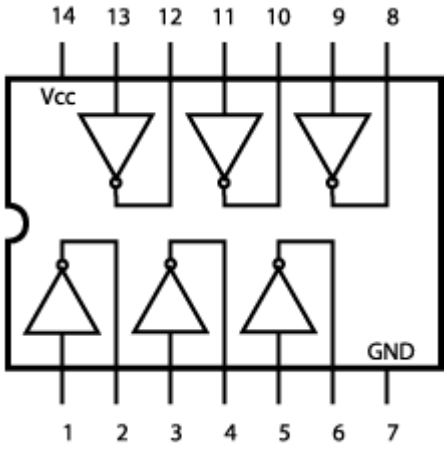
In all experiments, you will be expected to obtain all instruments, leads, components at the start of the experiment and return them to their proper place after you have finished the experiment. Please inform the demonstrator or technician if you locate faulty equipment. If you damage a chip, inform a demonstrator, don't put it back in the box of chips for somebody else to use.

Example Implementation of a Logic Circuit

Build a circuit to implement the Boolean function $F = \bar{A}.\bar{B}$



Quad 2 Input 7400



Hex 7404 Inverter

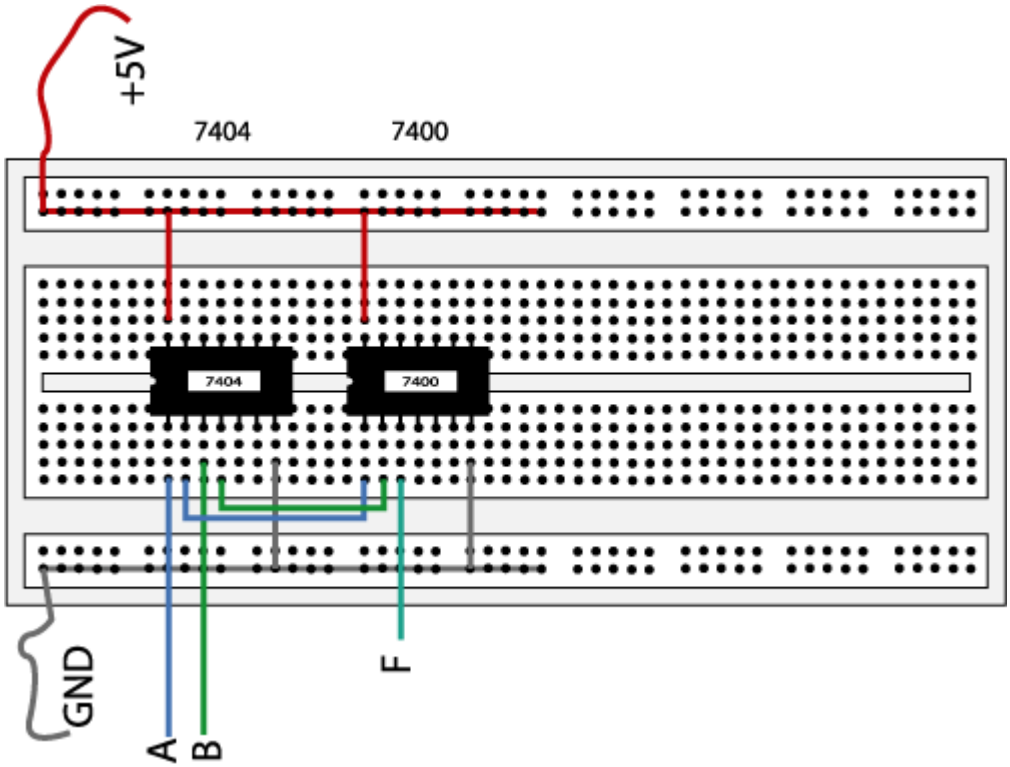
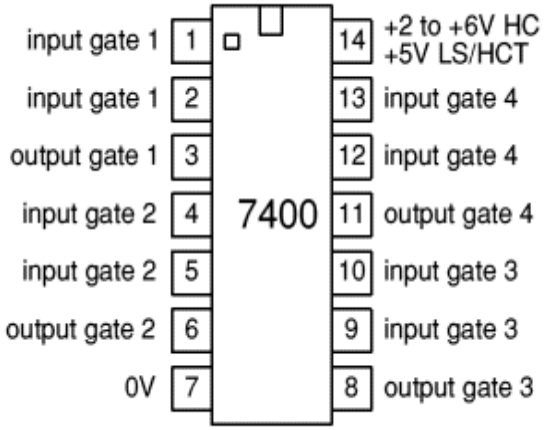


Fig2. The complete designed and connected circuit

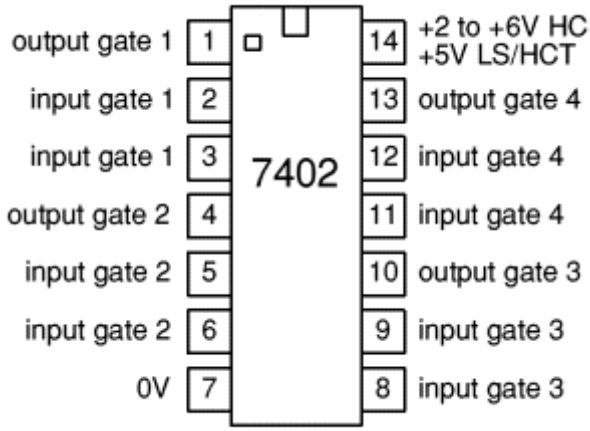
Sometimes the chip manufacturer may denote the first pin by a small indented circle above the first pin of the chip. Place your chips in the same direction, to save confusion at a later stage. Remember that you must connect power to the chips to get them to work.

Useful IC Pin details

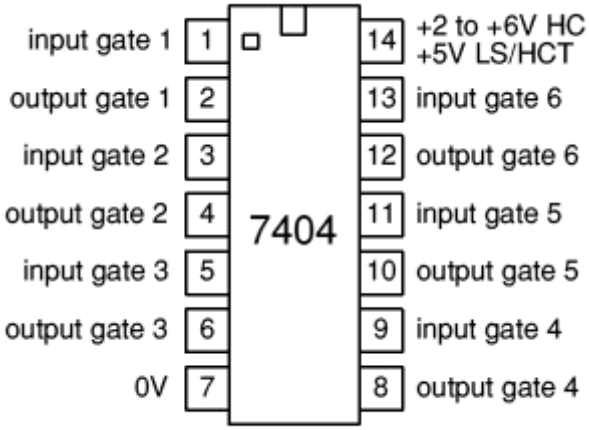
7400(NAND)



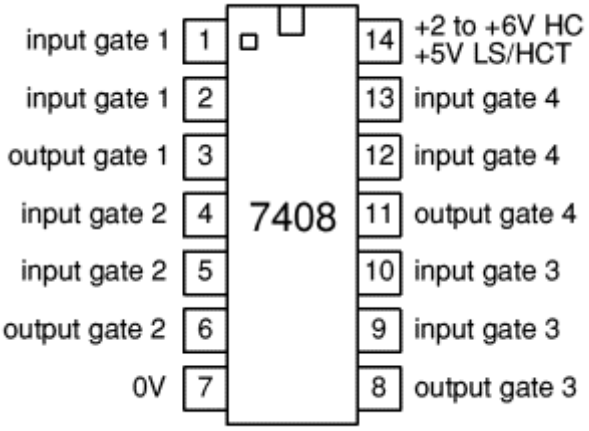
7402(NOR)



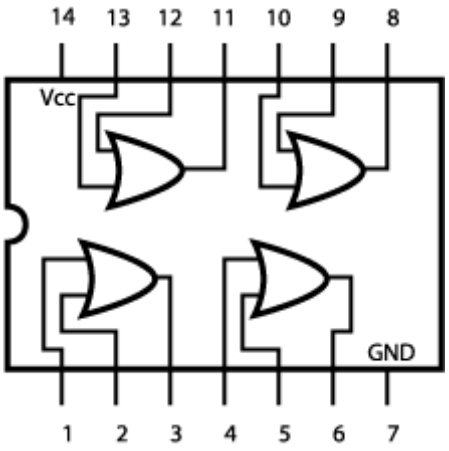
7404(NOT)



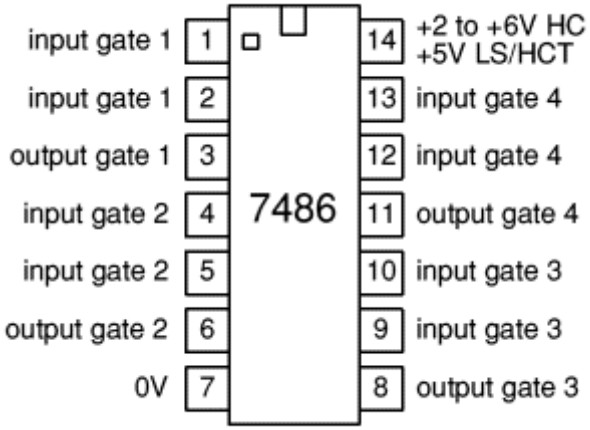
7408(AND)



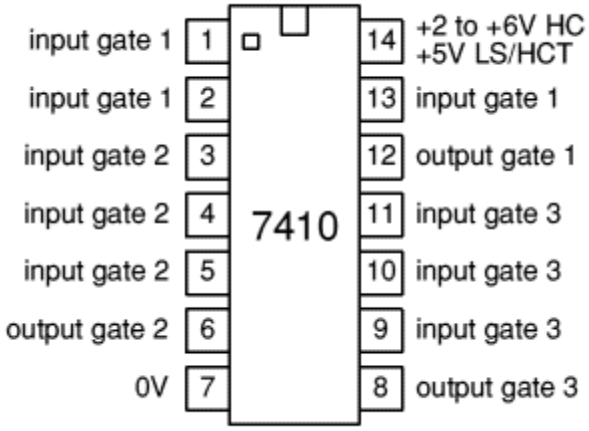
7432(OR)



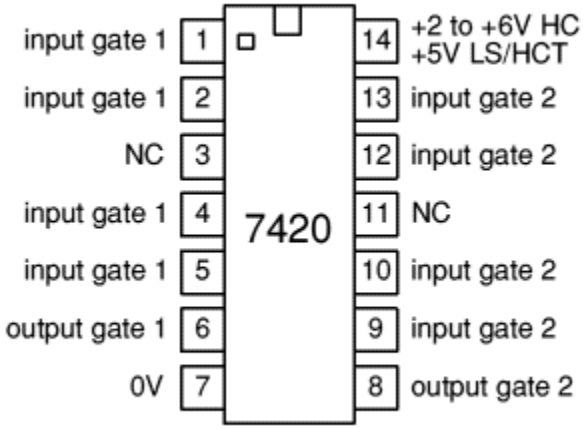
7486(EX-OR)



7410(3-i/p NAND)



7420(4-i/p NAND)



EXPERIMENT NO: 1

REALIZATION OF FUNCTIONS USING BASIC AND UNIVERSAL GATES (SOP AND POS FORMS)

AIM:

To study and verify the truth table of logic gates.
To simplify the given expression and to realize it using Basic gates and Universal gates.

COMPONENTS REQUIRED:

- Logic gates (IC) trainer kit.
- Connecting patch chords.
- IC 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486 Patch Cords & IC Trainer Kit.

THEORY:

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

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

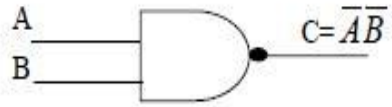
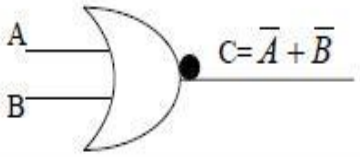
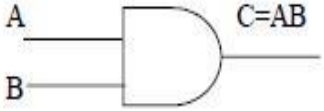
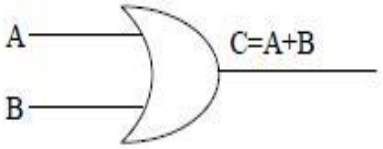
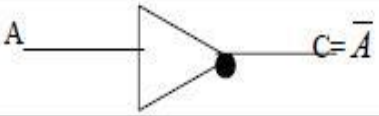
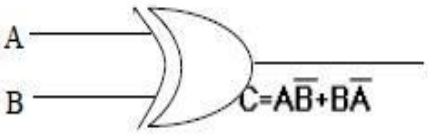
TTL Transistor-transistor logic
ECL Emitter-coupled logic
MOS Metal-oxide semiconductor
CMOS Complementary metal-oxide semiconductor

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

power consumption. CMOS logic consumes far less power than MOS logic. There are various commercial integrated circuit chips available. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

PROCEDURE:

1. Check the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Provide the input data via the input switches and observe the output on output LEDs

S.NO	GATE	SYMBOL	INPUTS		OUTPUT
			A	B	C
1.	NAND IC 7400		0	0	1
			0	1	1
			1	0	1
			1	1	0
2.	NOR IC 7402		0	0	1
			0	1	0
			1	0	0
			1	1	0
3.	AND IC 7408		0	0	0
			0	1	0
			1	0	0
			1	1	1
4.	OR IC 7432		0	0	0
			0	1	1
			1	0	1
			1	1	1
5.	NOT IC 7404		1	-	0
			0	-	1
6.	EX-OR IC 7486		0	0	0
			0	1	1
			1	0	1
			1	1	0

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?

REALIZATION OF BOOLEAN EXPRESSION

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

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

Karnaugh Maps

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

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

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

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

For a function of four variables: f(w, x, y, z)

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

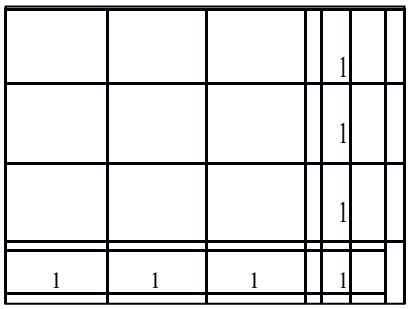
Realization of Boolean expression:

1) Y= $A\bar{B}\bar{C}D+\bar{A}B\bar{C}D+\bar{A}BCD+A\bar{B}C\bar{D}+A\bar{B}C\bar{D}+\bar{A}\bar{B}CD+\bar{A}BCD$

TRUTH TABLE

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

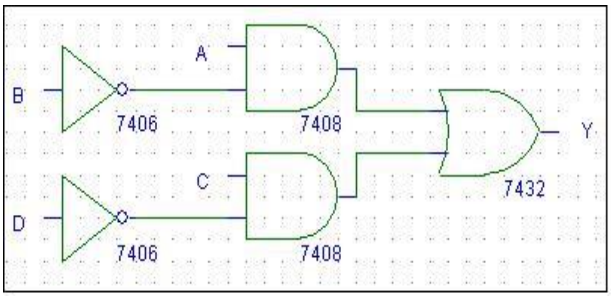
K Map Realization



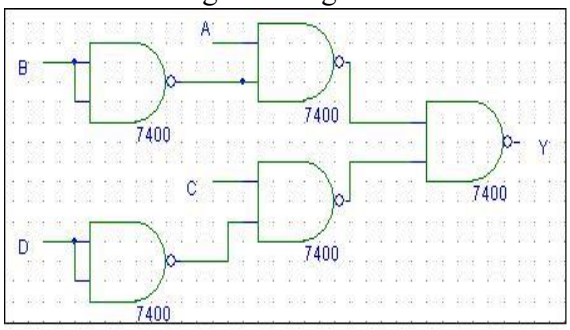
After simplifying using K-Map method we get

Y =A $\bar{B}$ + C $\bar{D}$

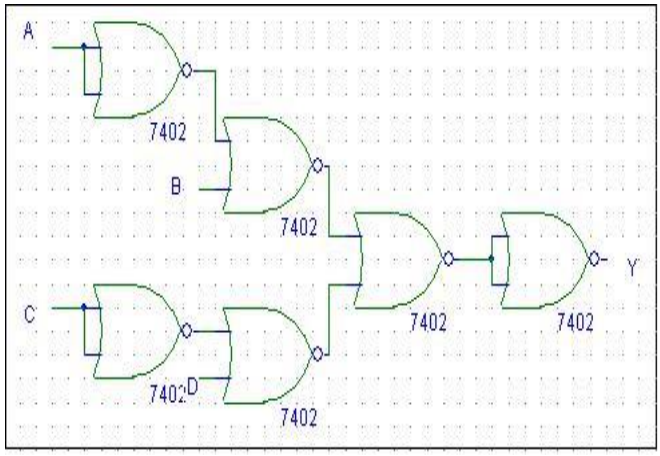
Realization using Basic gates



Realization using NAND gates



Realization using NOR gate



2) For the given Truth Table, realize a logical circuit using basic gates and NAND gates

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

PROCEDURE:

1. Check the components for their proper working. Insert the appropriate IC to the IC base.
2. Make connections as shown in the circuit diagram.
3. Provide the input data via the input switches and observe the output on output LEDs Verify the Truth Table

RESULT:

Simplified and verified the Boolean function using basic gates and universal gates

VIVA QUESTIONS:

1. What are the different methods to obtain minimal expression?
2. What is a Min term and Max term
3. State the difference between SOP and POS.
4. What is meant by canonical representation?
5. What is K-map? Why is it used?
6. What are universal gates

EXPERIMENT NO: 2

DESIGN AND REALIZATION OF HALF/FULL ADDER AND SUBTRACTOR USING BASIC GATES AND UNIVERSAL GATES

AIM:

To realize

i) Half Adder and Full Adder

ii) Half Subtractor and Full Subtractor by using Basic gates and NAND gates

COMPONENTS REQUIRED:

IC 7400, IC 7408, IC 7486, IC 7432, Patch Cords & IC Trainer Kit.

THEORY:

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

$S = A \oplus B$ $\qquad$ $C = A\,B$

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

$S = (x \oplus y) \oplus Cin$ $\qquad$ $C = xy + Cin\,(x \oplus y)$

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

$S = A \oplus B$ $\qquad$ $C = \bar{A}\,B$

Full Subtractor: Subtracting two single-bit binary values, B, Cin from a single-bit value Aproduces a difference bit D and a borrow out Br bit. This is called full subtraction. The Boolean functions describing the full-subtracter are:

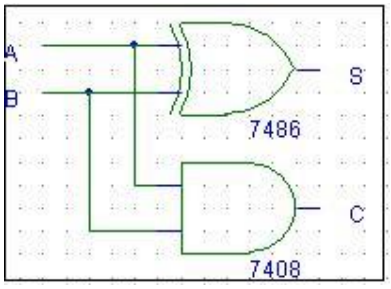
$D = (x \oplus y) \oplus Cin$ $\qquad$ $Br = \bar{A}\,B + \bar{A}\,(Cin) + B(Cin)$

I. TO REALIZE HALF ADDER

TRUTH TABLE

INPUTS		OUTPUTS	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

i) Basic Gates

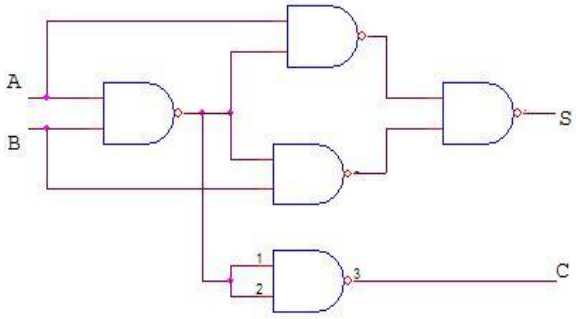


BOOLEAN EXPRESSIONS:

$S = A \oplus B$

$C = A\,B$

ii) NAND Gates



II. FULL ADDER

TRUTH TABLE

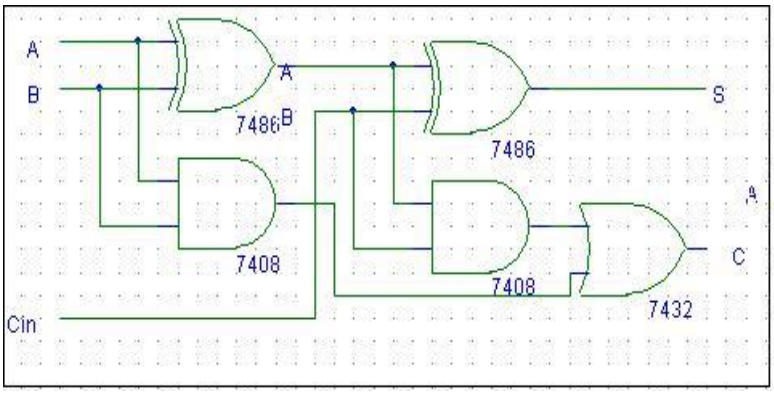
INPUTS			OUTPUTS	
A	B	Cin	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

BOOLEAN EXPRESSIONS:

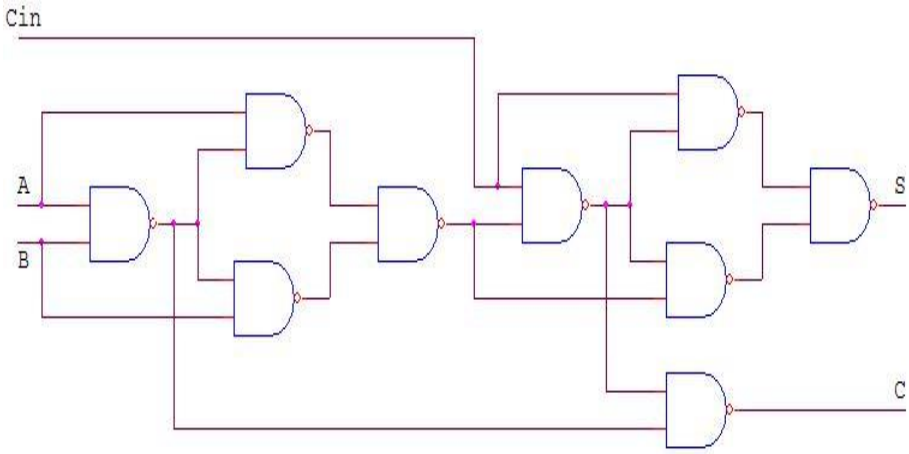
$S = A \oplus B \oplus C$

$C = A\,B + B\,Cin + A\,Cin$

i) BASIC GATES



ii) NAND GATES



III. HALF SUBTRACTOR

TRUTH TABLE

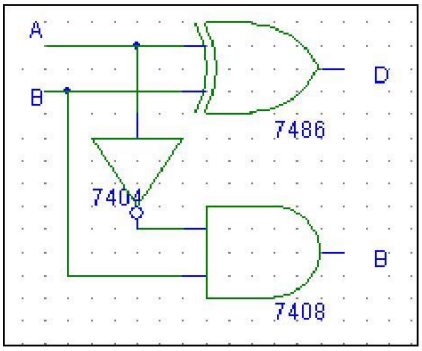
INPUTS		OUTPUTS	
A	B	D	Br
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

BOOLEAN EXPRESSIONS:

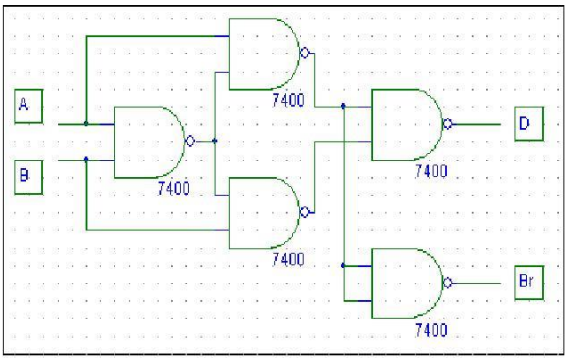
$$D = A \oplus B$$

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

i)BASIC GATES



ii) NAND Gates



IV. FULL SUBTRACTOR

TRUTH TABLE

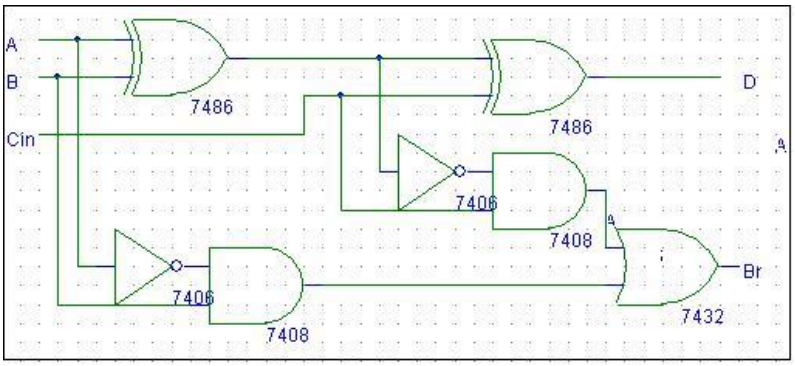
INPUTS			OUTPUTS	
A	B	Cin	D	Br
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

BOOLEAN EXPRESSIONS:

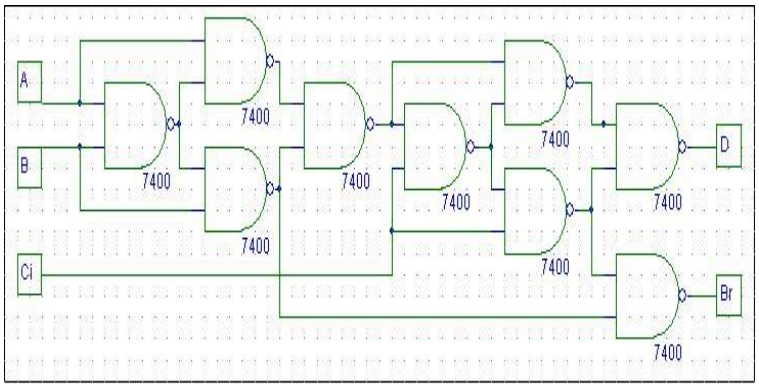
$$D = A \oplus B \oplus C$$

$$Br = \bar{A}\,B + B\,Cin + \bar{A}\,Cin$$

i) BASIC GATES



ii) To Realize the Full subtractor using NAND Gates only



PROCEDURE:

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

RESULT:

The truth table of the above circuits is verified.

VIVA QUESTIONS:

1) What is a half adder?
2) What is a full adder?
3) What are the applications of adders?
4) What is a half subtractor?
5) What is a full subtractor?
6) What are the applications of subtractors?
7) Obtain the minimal expression for above circuits.
8) Realize a full adder using two half adders
9) Realize a full subtractors using two half subtractors

EXPERIMENT NO: 3

4-BIT ADDERS/SUBTRACTORS AND BCD ADDER USING 7483

AIM:

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

i) A 4-bit binary parallel adder.
ii) A 4-bit binary parallel subtractor.

LEARNING OBJECTIVE**:**

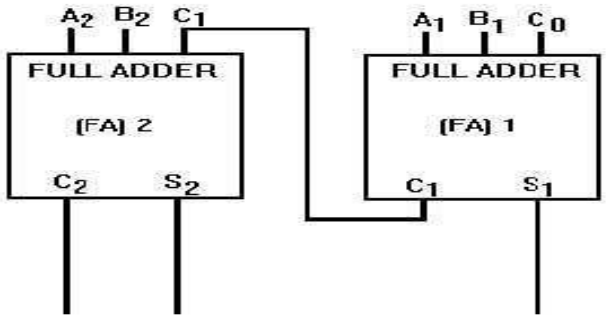
To learn about IC 7483 and its internal structure. To realize a subtractor using adder IC 7483.

COMPONENTS REQUIRED:

IC 7483, IC 7486, Patch Cords & IC Trainer Kit.

THEORY:

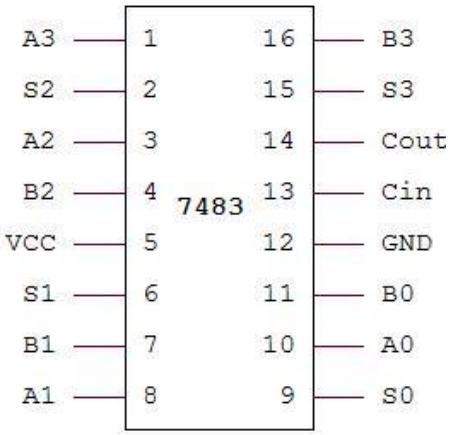
The Full adder can add single-digit binary numbers and carries. The largest sum that can be obtained using a full adder is 11_2. Parallel adders can add multiple-digit numbers. If full adders are placed in parallel, we can add two- or four-digit numbers or any other size desired. Figure below uses STANDARD SYMBOLS to show a parallel adder capable of adding two, two-digit binary numbers The addend would be on A inputs, and the augend on the B inputs. For this explanation we will assume there is no input to C_0 (carry from a previous circuit)



To add 10_2 (addend) and 01_2 (augend), the addend inputs will be 1 on A2 and 0 on A1. The augend inputs will be 0 on B2 and 1 on B1. Working from right to left, as we do in normal addition, let's calculate the outputs of each full adder. With A1 at 0 and B1 at 1, the output of adder1 will be a sum (S_1) of 1 with no carry (C_1). Since A2 is 1 and B2 is 0, we have a sum (S_2) of 1 with no carry (C_2) from adder1. To determine the sum, read the

outputs (C2, S2, and S1) from left to right. In this case, C2 = 0, S2 = 1, and S1 = 1. The sum, then, of 10_2 and 01_2 is 011_2. To add four bits we require four full adders arranged in parallel. IC 7483 is a 4- bit parallel adder whose pin diagram is shown.

	MSB				**LSB**
INPUTS					Cin
		A_3	A_2	A_1	A_0
		B_3	B_2	B_1	B_0
OUTPUT	Cout	S_3	S_2	S_1	S_0



IC 7483 pin diagram

1. 4-Bit Binary Adder

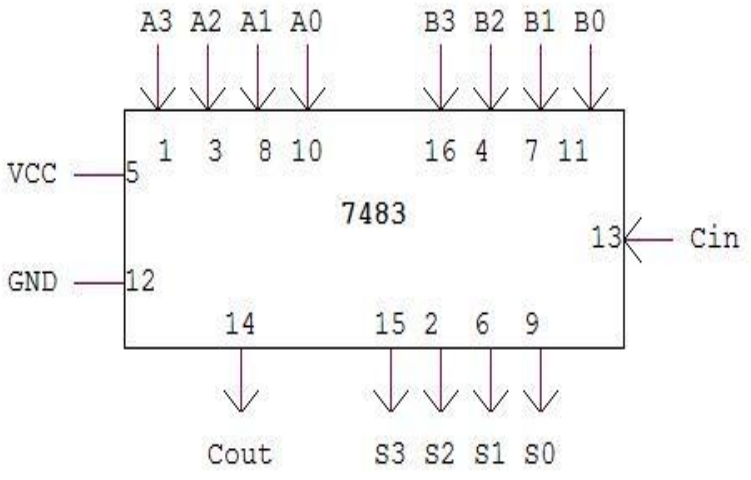
An Example: 7+2=11 (1001)

- 7 is realized at $A_3\ A_2\ A_1\ A_0 = 0111$

□2 is realized at $B_3\ B_2\ B_1\ B_0 = \underline{0010}$

Sum = <u>1001</u>

ADDER CIRCUIT:



PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply augend and addend bits on A and B and cin=0.
- Verify the results and observe the outputs.

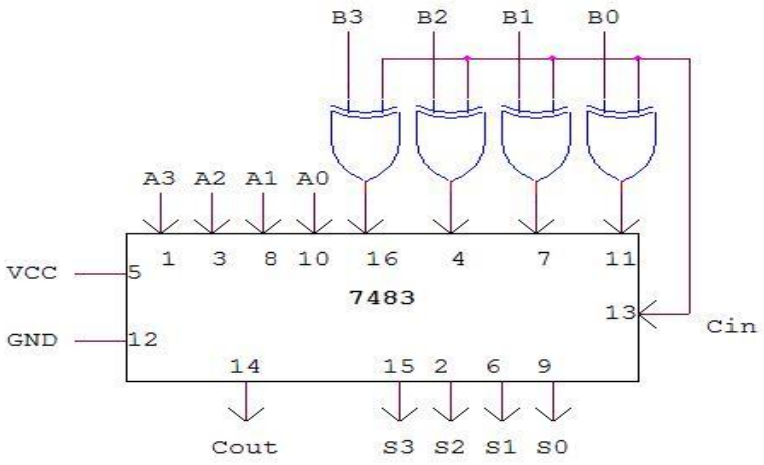
ii) 4-BIT BINARY SUBTRACTOR.

Subtraction is carried out by adding 2's complement of the subtrahend.
Example: 8 – 3 = 5 (0101)

- 8 is realized at $A_3\ A_2\ A_1\ A_0$ = 1000
- 3 is realized at $B_3B_2B_1B_0$ through X-OR gates = 0011
- Output of X-OR gate is 1's complement of 3 = 1100
- 2's Complement can be obtained by adding Cin =1

Therefore

Cin = 1
$A_3\ A_2\ A_1\ A_0$ = 1 0 0 0
$B_3\ B_2\ B_1\ B_0$ = 1 1 0 0
S_3 S2 S1 S_0 = 0 1 0 1
Cout = 1 (Ignored)



PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply Minuend and subtrahend bits on A and B and cin=1.
- Verify the results and observe the outputs.

RESULTS:

Verified the working of IC 7483 as adder and subtractor.

EXPERIMENT NO: 4

2/3 BITBINARYCOMPARATORS

AIM:

To realize One & Two Bit Comparator and to study 7485 magnitude comparator.

LEARNING OBJECTIVE:

- To learn about various applications of comparator
- To learn and understand the working of IC 7485 magnitude comparator
- To learn to realize 8-bit comparator using 4-bit comparator

THEORY:

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

COMPONENTS REQUIRED:

IC 7400, IC 7410, IC 7420, IC 7432, IC 7486, IC 7402, IC 7408, IC 7404, IC 7485, Patch Cords & IC Trainer Kit.

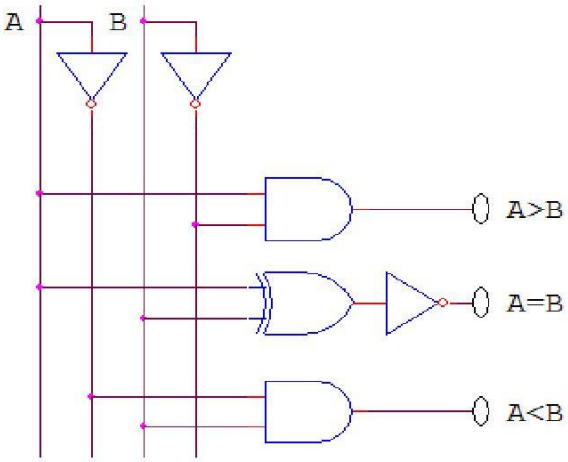
1) 1- BIT COMPARATOR

TRUTH TABLE

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

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

$A=B = A\,B + AB$



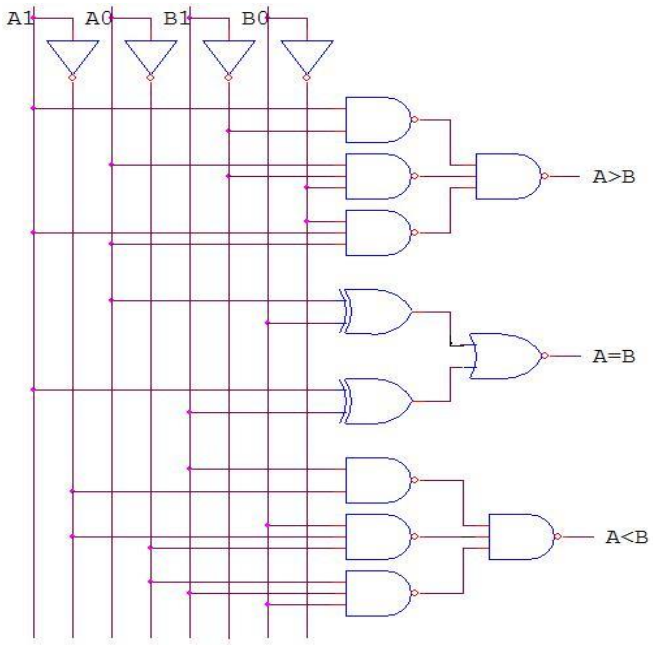
INPUTS		OUTPUTS		
A	**B**	**A > B**	**A = B**	**A < B**
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

2) 2- BIT COMPARATOR

(A>B)= A1 $\bar{B}$1+A0B1$\bar{B}$0+B0A1$\bar{A}$0

(A=B) = (A0 $\odot$ B0) (A1 $\odot$ B1)

(A<B) = B1 A1 +B0A1A0+A0B1B0

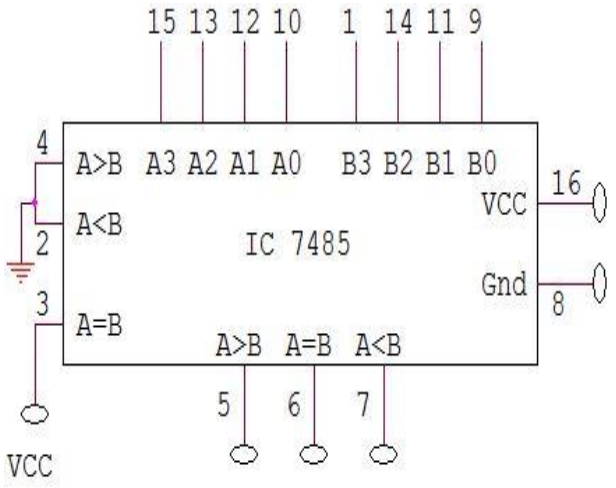


2-bit comparator circuit diagram

TRUTH TABLE

INPUTS				OUTPUTS		
A_1	A_0	B_1	B_0	A > B	A = B	A < B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

3) TO COMPARE THE GIVEN DATA USING 7485 CHIP.



.

A				B				Result
A3	**A2**	**A1**	**A0**	**B3**	**B2**	**B1**	**B0**	
0	0	0	1	0	0	0	0	**A > B**
0	0	0	1	0	0	0	1	**A = B**
0	0	0	0	0	0	0	1	**A < B**

PROCEDURE:

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

RESULT:

One bit, two bit and four bit comparators are verified using basic gates and magnitude comparator IC7485

VIVA QUESTIONS:

1) What is a comparator?
2) What are the applications of comparator?
3) Derive the Boolean expressions of one bit comparator and two bit comparators.
4) How do you realize a higher magnitude comparator using lower bit comparator
5) Design a 2 bit comparator using a single Logic gates?
6) Design an 8 bit comparator using a two numbers of IC 7485?

EXPERIMENT NO: 5

BINARY TO GRAY & GRAY TO BINARY CODE CONVERTER

AIM: To realize Binary to Gray code converter and vice versa.

LEARNING OBJECTIVE:

- To learn the importance of non-weighted code
- To learn to generate gray code

COMPONENTS REQUIRED:
IC 7400, IC 7486, and IC 7408, Patch Cords & IC Trainer Kit

BINARY TO GRAY CONVERSION

Binary				Gray			
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

BOOLEAN EXPRESSIONS:

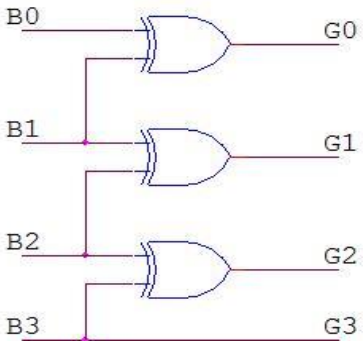
G3=B3

G2=B3 ⊕ B2

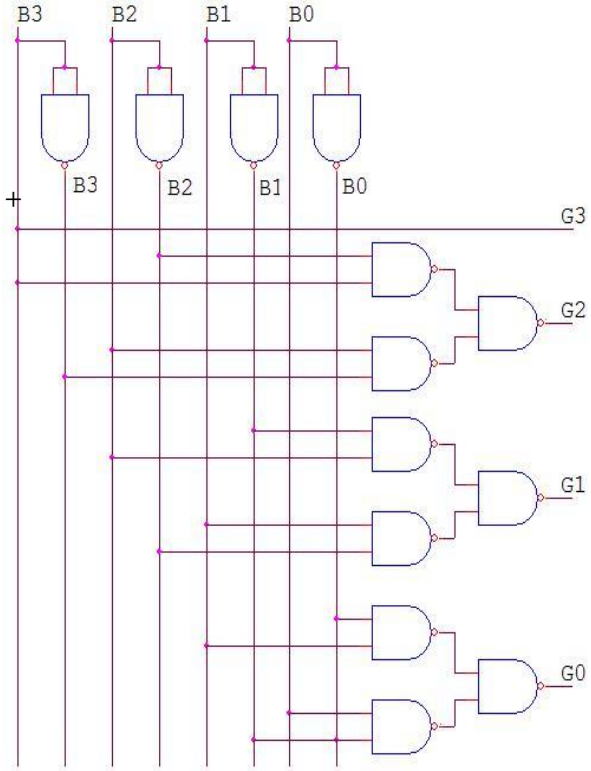
G1=B1 ⊕ B2

G0=B1 ⊕ B0

BINARY TO GRAY CODE USING EX-OR GATES



REALIZATION USING NAND GATES:



I) GRAY TO BINARY CONVERSION

0	**0**	**1**	**1**	
0	**0**	**1**	**1**	
0	**0**	**1**	**1**	
0	**0**	**1**	**1**	B3 = G3

0	**1**	**0**	1	
0	**1**	**0**	1	B2=G3 ⊕ G2
0	**1**	**0**	1	
0	**1**	**0**	1	

0	1	**0**	1	
0	1	**0**	1	B1=G3 ⊕ G2 ⊕ G1
1	**0**	1	**0**	
1	**0**	1	**0**	

0	**1**	**0**	1	
1	**0**	**1**	**0**	
0	**1**	**0**	1	
1	**0**	**1**	**0**	B0=G3⊕G2⊕G1⊕G0

GRAY TO BINARY CONVERSION

Gray				Binary			
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

BOOLEAN EXPRESSIONS:

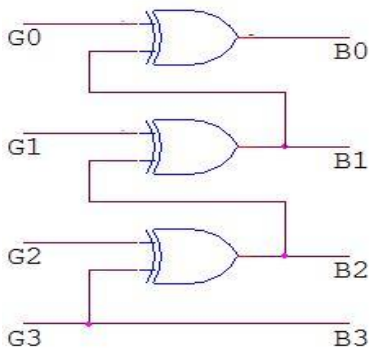
B3=G3

B2=G3 ⊕ G2

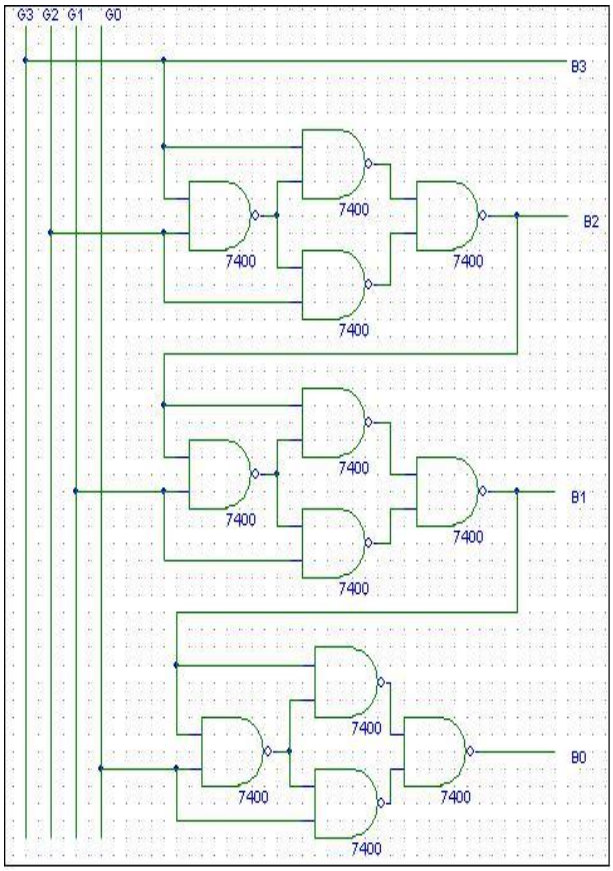
B1=G3 ⊕ G2 ⊕ G1

B0=G3 ⊕ G2 ⊕ G1 ⊕ G0

GRAY TO BINARY CODE CONVERSION USING EX-OR GATES



REALIZATION USING NAND GATES:



PROCEDURE:

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

RESULT:

Binary to gray code conversion and vice versa is realized using EX-OR gates and NAND gates.

VIVA QUESTIONS:

1) What are code converters?
2) What is the necessity of code conversions?
3) What is gray code?
4) Realize the Boolean expressions for
   a) Binary to gray code conversion
   b) Gray to binary code conversion

EXPERIMENT NO: 6

STUDY OF FLIP FLOPS

AIM:

Truth Table verification of

1) RS Flip Flop
2) T type Flip Flop.
3) D type Flip Flop.
4) JK Flip Flop.
5) JK Master Slave Flip Flop.

LEARNING OBJECTIVE:

To learn about various Flip-Flops
To learn and understand the working of Master slave FF
To learn about applications of FFs
Conversions of one type of Flip flop to another

COMPONENTS REQUIRED:

IC 7408, IC 7404, IC 7402, IC 7400, Patch Cords & IC Trainer Kit.

THEORY:

Logic circuits that incorporate memory cells are called *sequential logic circuits*; their output depends not only upon the present value of the input but also upon the previous values. Sequential logic circuits often require a timing generator (a clock) for their operation.

The latch (flip-flop) is a basic bi-stable memory element widely used in sequential logic circuits. Usually there are two outputs, Q and its complementary value.

Some of the most widely used latches are listed below.

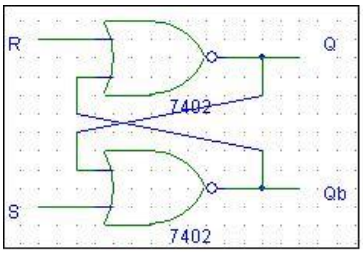
SR LATCH:

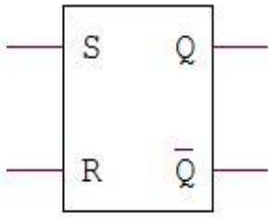
An S-R latch consists of two cross-coupled NOR gates. An S-R flip-flop can also be design using cross-coupled NAND gates as shown. The truth tables of the circuits are shown below.

A clocked S-R flip-flop has an additional clock input so that the S and R inputs are active only when the clock is high. When the clock goes low, the state of flip-flop is latched and cannot change until the clock goes high again. Therefore, the clocked S-R flip-flop is also called “enabled” S-R flip-flop.

A D latch combines the S and R inputs of an S-R latch into one input by adding an inverter. When the clock is high, the output follows the D input, and when the clock goes low, the state is latched. A S-R flip-flop can be converted to T-flip flop by connecting S input to Qb and R to Q.

1) S-R LATCH:



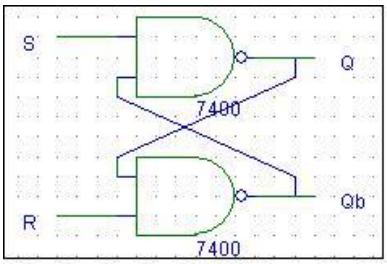


(A) LOGIC DIAGRAM **(B) SYMBOL**

TRUTH TABLE

S	R	Q+	$\overline{Q}$ b+
0	0	Q	$\overline{Q}$b
0	1	0	1
1	0	1	0
1	1	0*	0*

$\overline{S}\,\overline{R}$ LATCH:

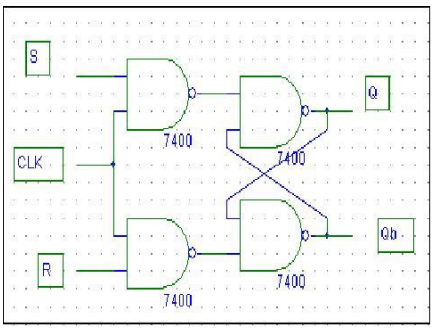


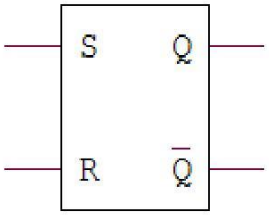
TRUTH TABLE

S	R	Q+	$\overline{Q}$ b+
0	0	1*	1*
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$ b

2) SR FLIP FLOP:

CIRCUIT DIAGRAM:





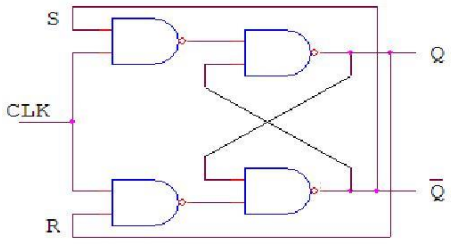
(A) LOGIC DIAGRAM **(B) SYMBOL**

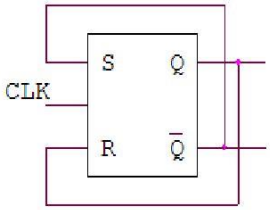
TRUTH TABLE

S	R	Q+	$\overline{Q}$ b+
0	0	Q	$\overline{Q}$b
0	1	0	1
1	0	1	0
1	1	0*	0*

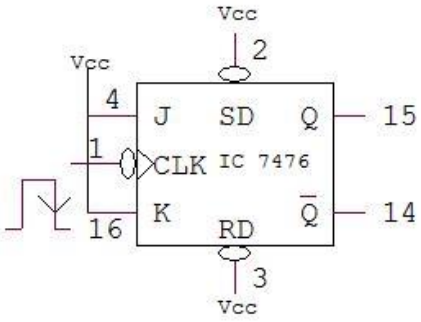
3) CONVERSION OF SR-FLIP FLOP TO T-FLIP FLOP (Toggle)

LOGIC DIAGRAMSYMBOL





T FLIP FLOP USING IC 7476

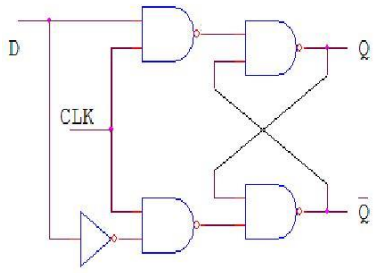


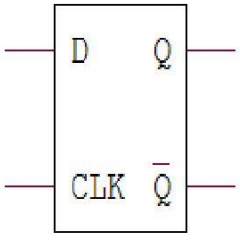
TRUTH TABLE

T	Qn + 1
0	Qn
1	Qn

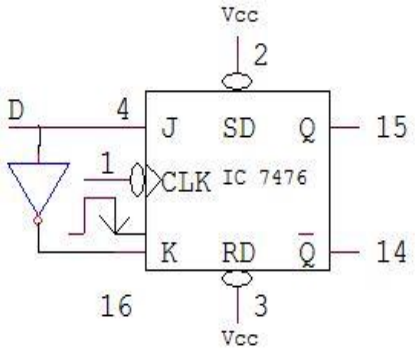
4) CONVERSION OF SR-FLIP FLOP TO D-FLIP FLOP :

LOGIC DIAGRAMSYMBOL





D FLIP FLOP USING IC 7476

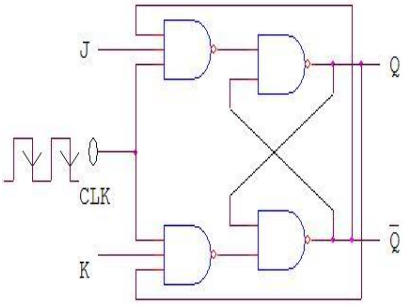


TRUTH TABLE

CLOCK	D	Q+	Q+
0	X	Q	$\overline{Q}$
1	0	0	1
1	1	1	0

5. CONVERSION OF SR-FLIP FLOP TO JK-FLIP FLOP

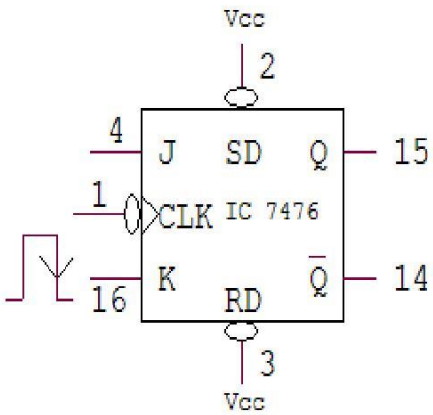
LOGIC DIAGRAM



TRUTH TABLE

Clock	J	K	Q+	Q'+	Comment
1	0	0	Q	Q'	No Change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	Q'	Q	Toggle

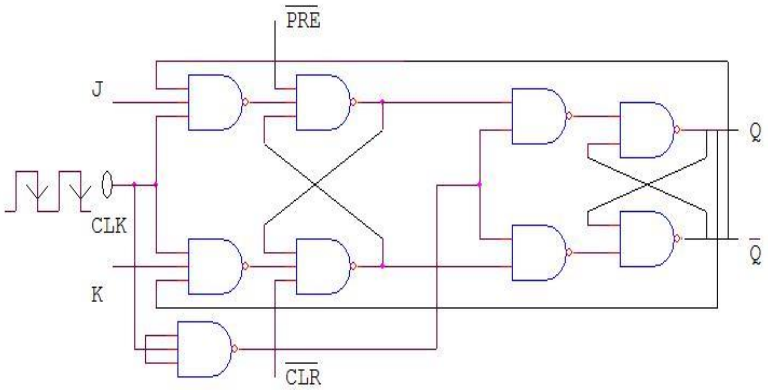
LOGIC DIAGRAM



TRUTH TABLE

SD	RD	Clock	J	K	Q	Q'	Comment
0	0	Not Allowed					
0	1	X	X	X	1	0	Set
1	0	X	X	X	0	1	Reset
1	1	1	0	0	NC	NC	Memory
1	1	1	0	1	0	1	Reset
1	1	1	1	0	1	0	Set
1	1	1	1	1	Q'	Q	Toggle

6. JK MASTER SLAVE FLIP FLOP



TRUTH TABLE

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

Clock	J	K	Q+	Q'+	Comment
1	0	0	Q	Q'	No Change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1		Race Around	

PROCEDURE:

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

VIVA QUESTIONS:

1. What is the difference between Flip-Flop & latch?
2. Give examples for synchronous & asynchronous inputs?
3. What are the applications of different Flip-Flops?
4. What is the advantage of Edge triggering over level triggering?
5. What is the relation between propagation delay & clock frequency of flip-flop?
6. What is race around in flip-flop & how to overcome it?
7. Convert the J K Flip-Flop into D flip-flop and T flip-flop?
8. List the functions of asynchronous inputs?

EXPERIMENT NO: 7

ASYNCHRONOUS COUNTER: REALIZATION OF Mod-N COUNTERS

AIM:

To realize Mod-N asynchronous counter

LEARNING OBJECTIVE:

To learn about Asynchronous Counter and its application
To learn the design of asynchronous up counter and down counter

COMPONENTS REQUIRED:

IC 7476, Patch Cords & IC Trainer Kit

THEORY:

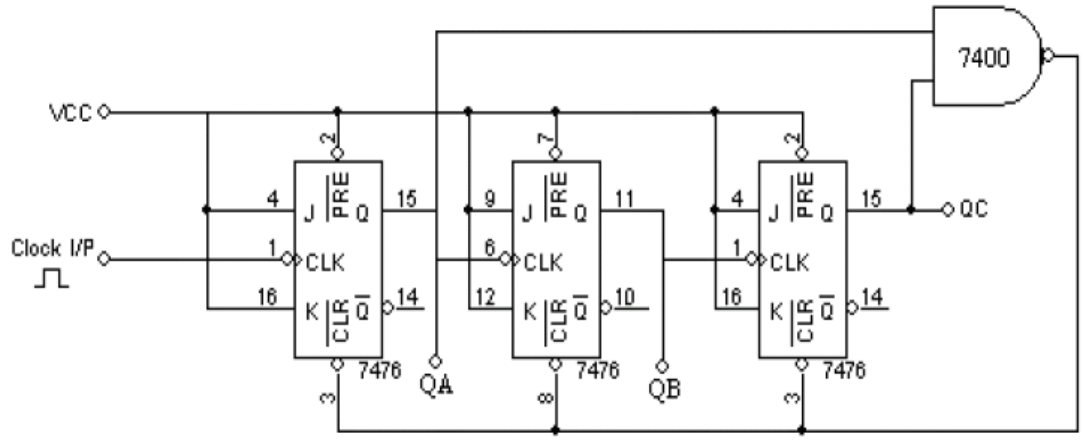
A counter in which each flip-flop is triggered by the output goes to previous flip-flop. As all the flip-flops do not change state simultaneously spike occur at the output. To avoid this, strobe pulse is required. Because of the propagation delay the operating speed of asynchronous counter is low. Asynchronous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

PROCEDURE:

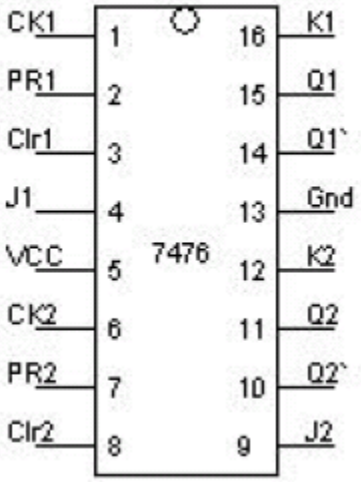
- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Verify the Truth Table and observe the outputs.

Circuit Diagram:

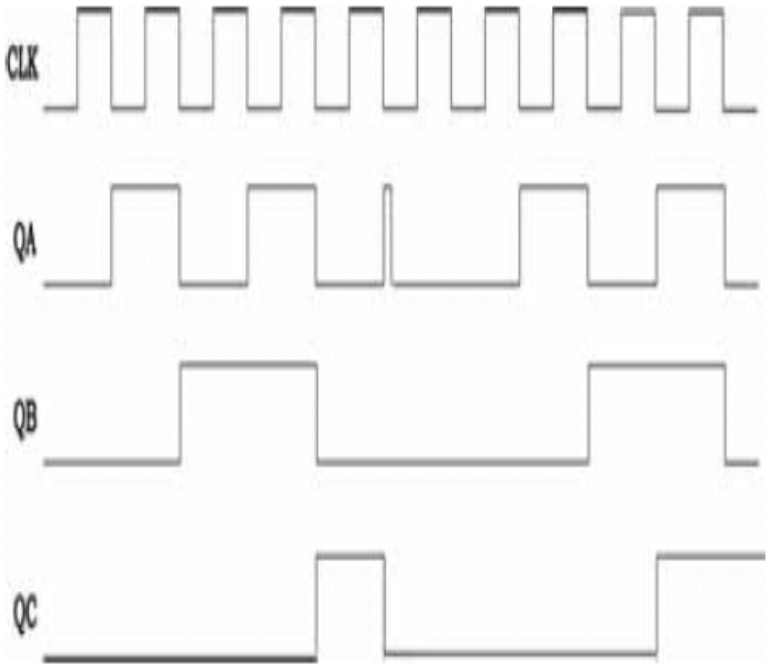
Mod 5 Asynchronous Counter:-



Pin Details: -

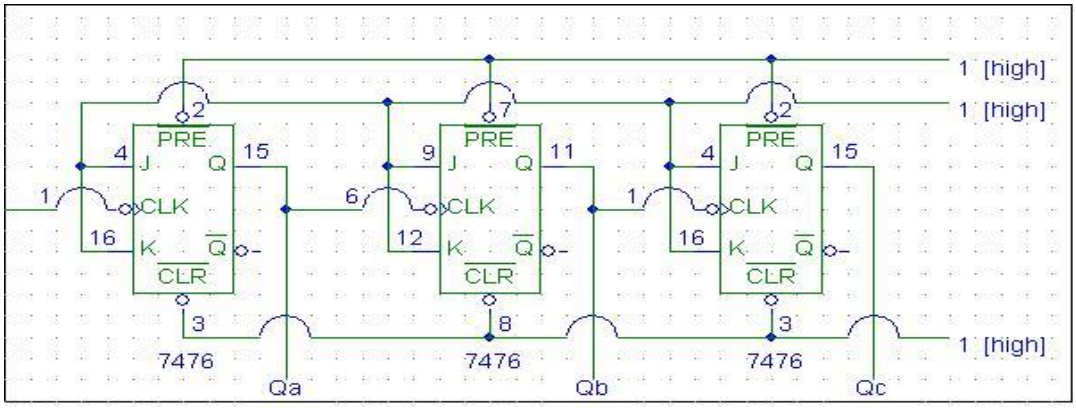


Mod 5 Asynchronous counter			
Clock	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	0	0	0



MOD-8 UP COUNTER

CIRCUIT DIAGRAM:

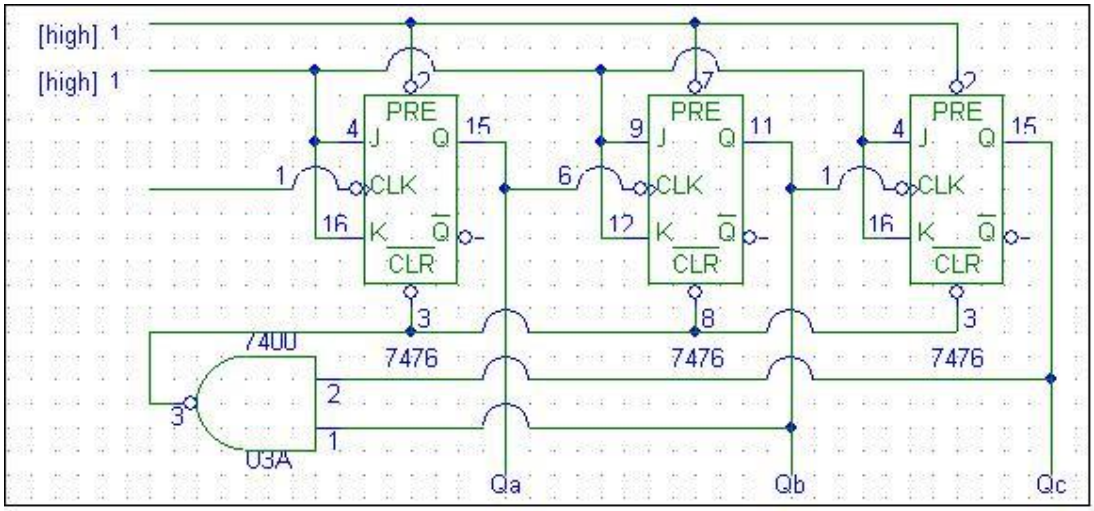


TRUTH TABLE

CLK	Q_C	Q_B	Q_A
0	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	0	0	0

MOD_6 UP COUNTER

CIRCUIT DIAGRAM

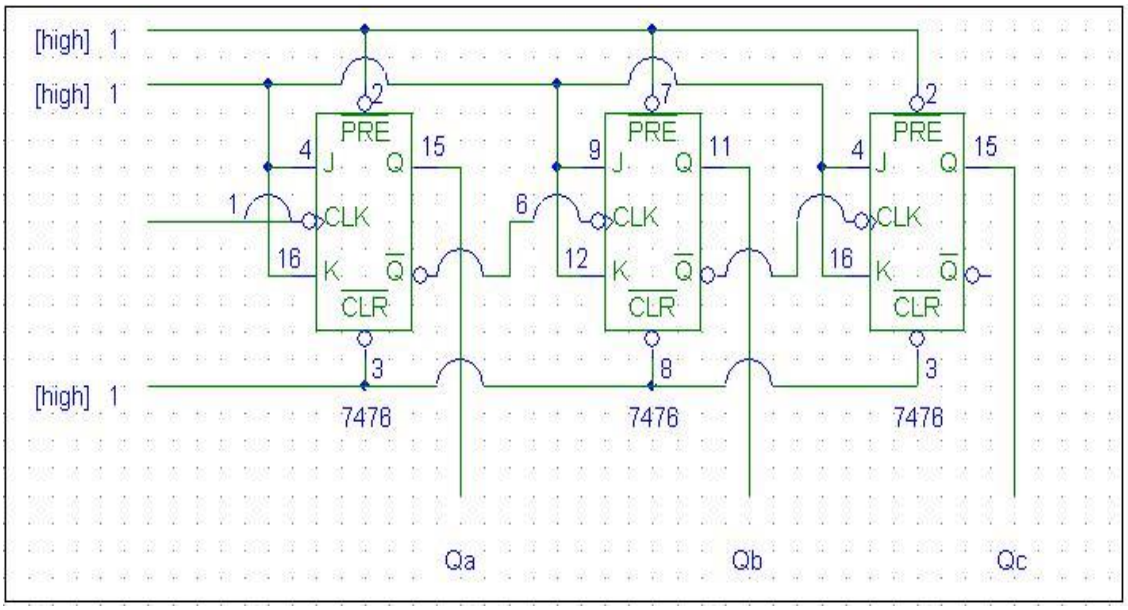


TRUTH TABLE

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

MOD-8 DOWN COUNTER

CIRCUIT DIAGRAM:

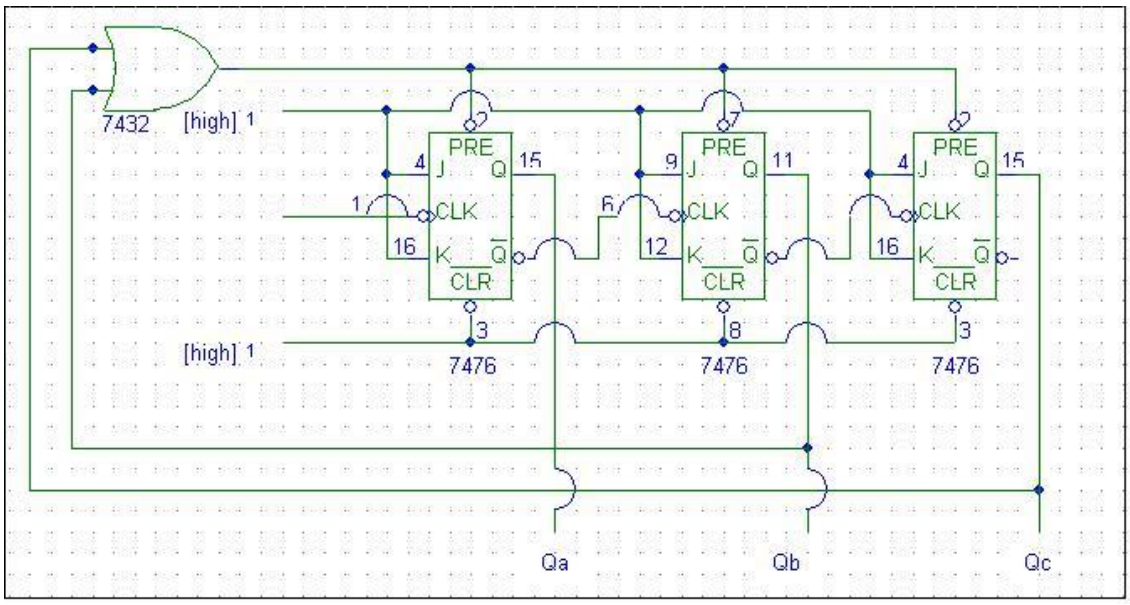


TRUTH TABLE

CLK	Q_C	Q_B	Q_A
0	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	1	1	1

MOD-6 DOWN COUNTER

CIRCUIT DIAGRAM:



TRUTH TABLE

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

RESULT: The working of Mod-N Asynchronous counters is verified

VIVA QUESTIONS:

1. What is an asynchronous counter?
2. How is it different from a synchronous counter?
3. Realize asynchronous counter using T flip-flop

EXPERIMENT NO: 8

ASYNCHRONOUS COUNTER: 3-BIT UP/DOWN COUNTER

Aim: - To realize the 3-bit Asynchronous up/down counter

Apparatus Required: -

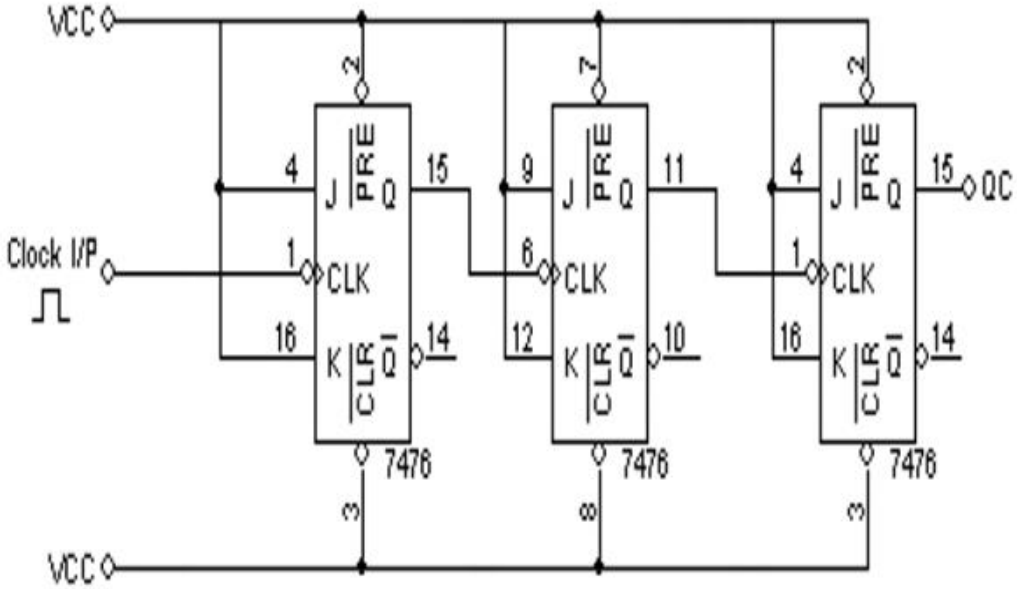
IC 7408, IC 7476, IC 7400, IC 7432 etc.

Procedure**:** -

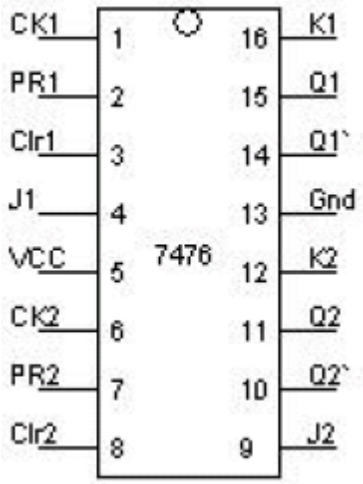
1. Connections are made as per circuit diagram.
2. Clock pulses are applied one by one at the clock I/P and the O/P is observed at QA, QB & QC for IC 7476.
3. Verify the Truth table .

CIRCUIT DIAGRAM

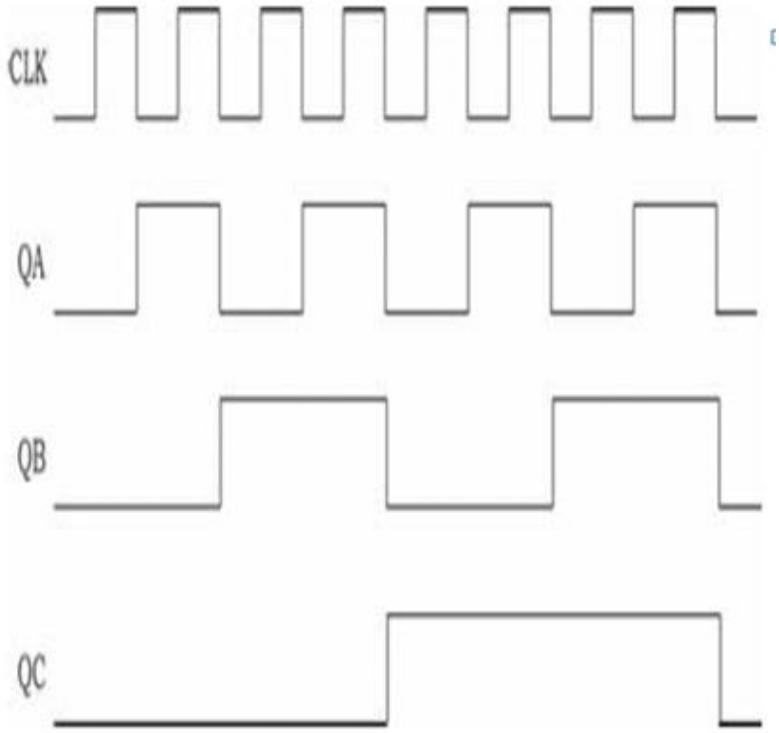
Circuit Diagram: - 3-Bit Asynchronous Up Counter



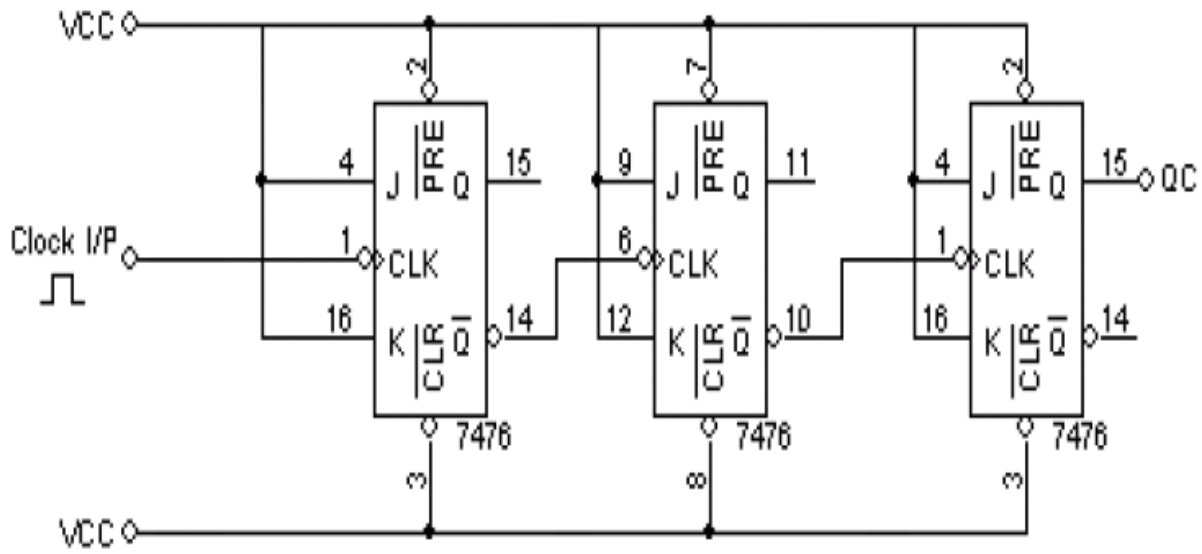
Pin Details: -



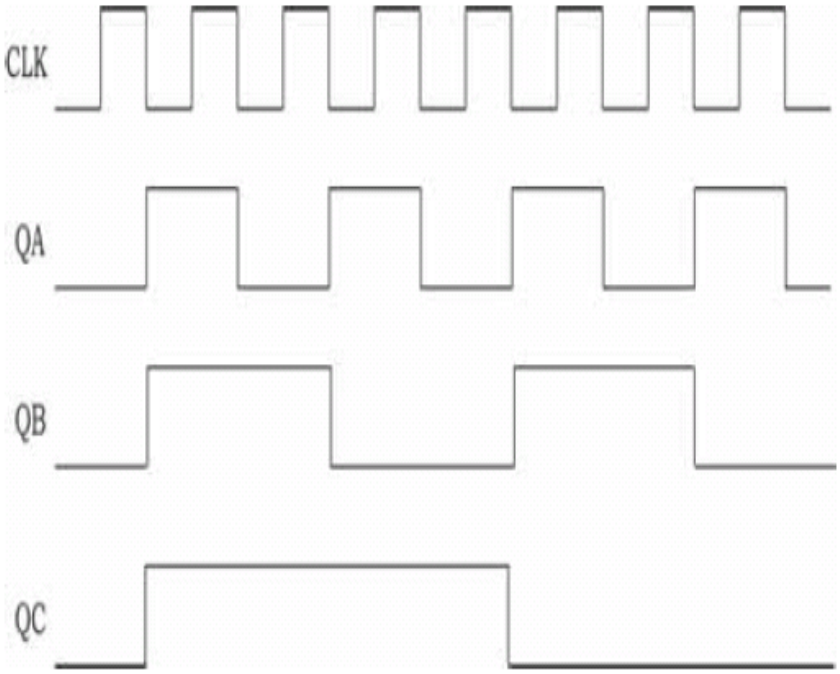
3-bit Asynchronous up counter			
Clock	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0



Circuit Diagram: - 3-Bit Asynchronous Down Counter



3-bit Asynchronous down counter			
Clock	QC	QB	QA
0	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	1	1	1
9	1	1	0



RESULT: The working of 3-bit up/down asynchronous counter is verified

EXPERIMENT NO: 9

SYNCHRONOUS COUNTER: REALIZATION OF 4-BIT BINARY UP/DOWN COUNTER

AIM:

To design IC 74193 as a up/down counter

LEARNING OBJECTIVE:

To learn about pre-settable Counter and its application

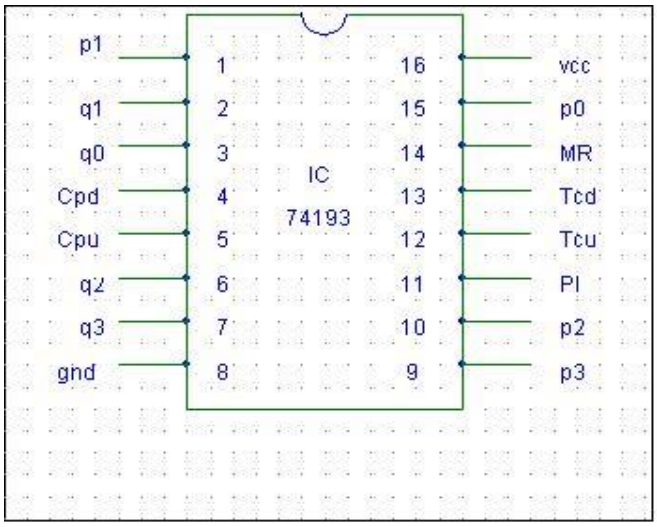
COMPONENTS REQUIRED:

IC 74193, Patch Cords & IC Trainer Kit

PROCEDURE:

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

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

Up counter

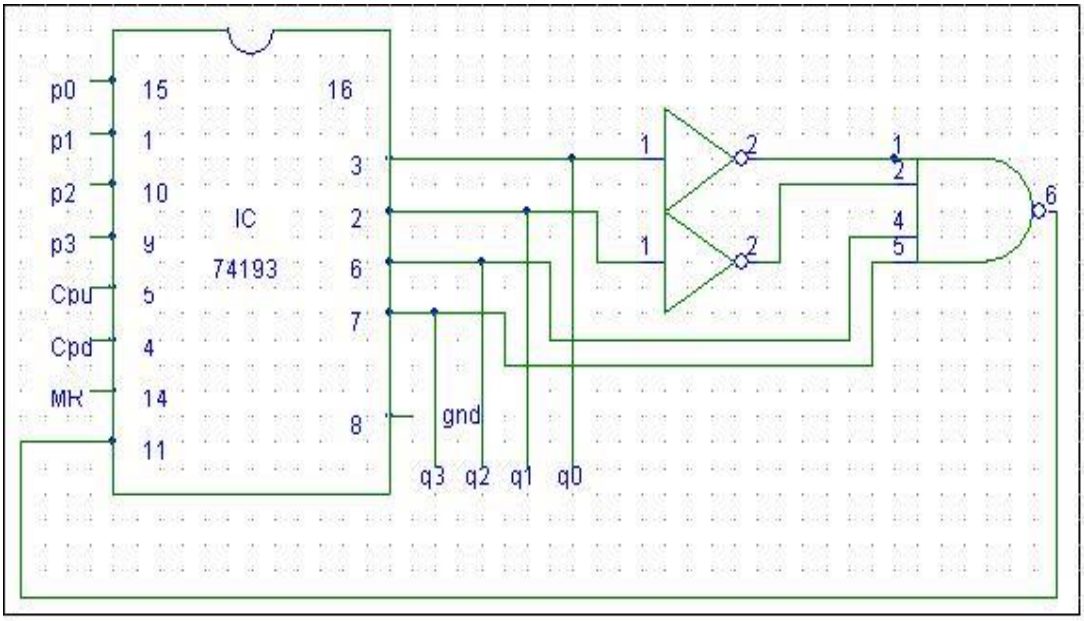
CLK	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Down counter

CLK	Q_D	Q_C	Q_B	Q_A
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	1	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	0	1	1	1
9	0	1	1	0
10	0	1	0	1
11	0	1	0	0
12	0	0	1	1
13	0	0	1	0
14	0	0	0	1
15	0	0	0	0
16	1	1	1	1

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

CIRCUIT DIAGRAM

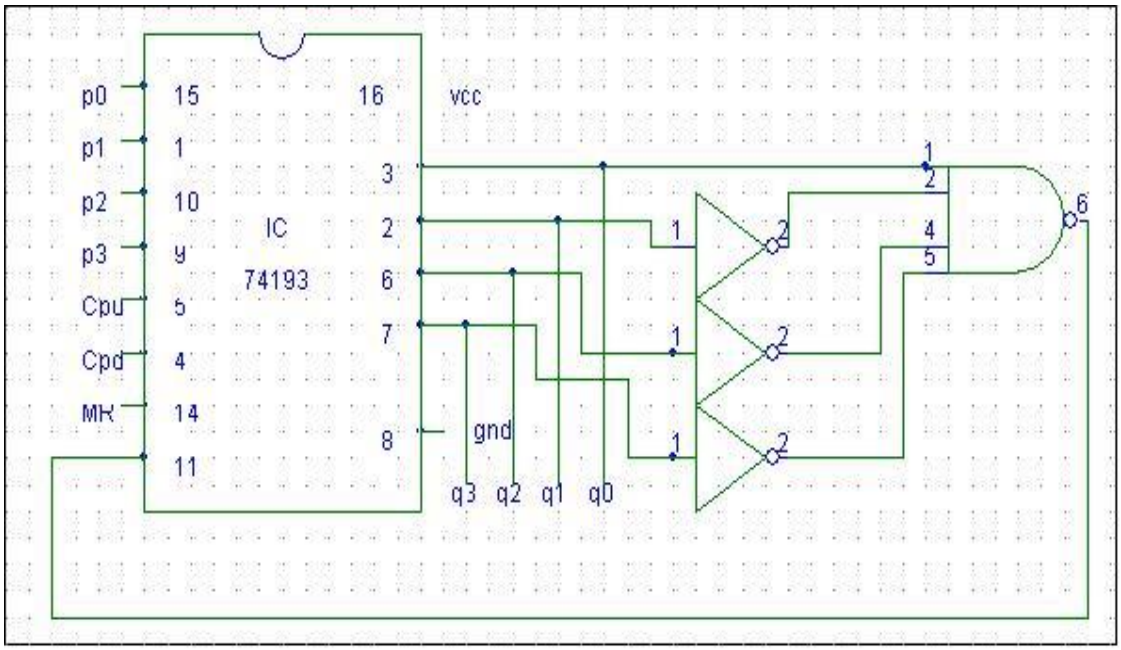


TRUTH TABLE

CLK	QD	QC	QB	QA
1	0	0	1	0
2	0	0	1	1
3	0	1	0	0
4	0	1	0	1
5	0	1	1	0
6	0	1	1	1
7	1	0	0	0
8	1	0	0	1
9	1	0	1	0
10	1	0	1	1
11	1	1	0	0
12	0	0	1	0

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

CIRCUIT DIAGRAM



TRUTH TABLE

CLK	Q_D	Q_C	Q_B	Q_A
1	1	0	1	1
2	1	0	1	0
3	1	0	0	1
4	1	0	0	0
5	0	1	1	1
6	0	1	1	0
7	0	1	0	1
8	0	1	0	0
9	0	0	1	1
10	0	0	1	0
11	0	0	0	1
12	1	0	1	1

RESULT: The working of IC 74193 as an up/down pre-settable counter is verified

VIVA QUESTIONS:

1) What is a pre-settable counter?
2) What are the applications of pre-settable counters?
3) Explain the working of IC 74193
4) Write the circuit for preset value of 0100 and N=5 (up counter)

EXPERIMENT NO: 10

SYNCHRONOUS COUNTERS:REALIZATION OF Mod-N COUNTERS

AIM:

To design and test 3-bit binary synchronous counter using flip-flop IC 7476 for the given sequence.

LEARNING OBJECTIVE:

To learn about synchronous Counter and its application
To learn the design of synchronous counter counter

COMPONENTS REQUIRED:

IC 7476, Patch Cords & IC Trainer Kit

THEORY:

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

PROCEDURE:

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

MOD 5 COUNTER:

TRUTH TABLE:

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

Present count			Next count		
Q_C	Q_B	Q_A	Q_C	Q_B	Q_A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0

· JK flip flop
excitation table:

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

DESIGN:

1	X	X	1
0	X	X	X

$J_A = Q_C$

X	1	X	X
X	X	X	X

$K_A = 1$

0	1	X	X
0	X	X	X

$J_B = Q_A$

X	X	1	0
X	X	X	X

$K_B = Q_A$

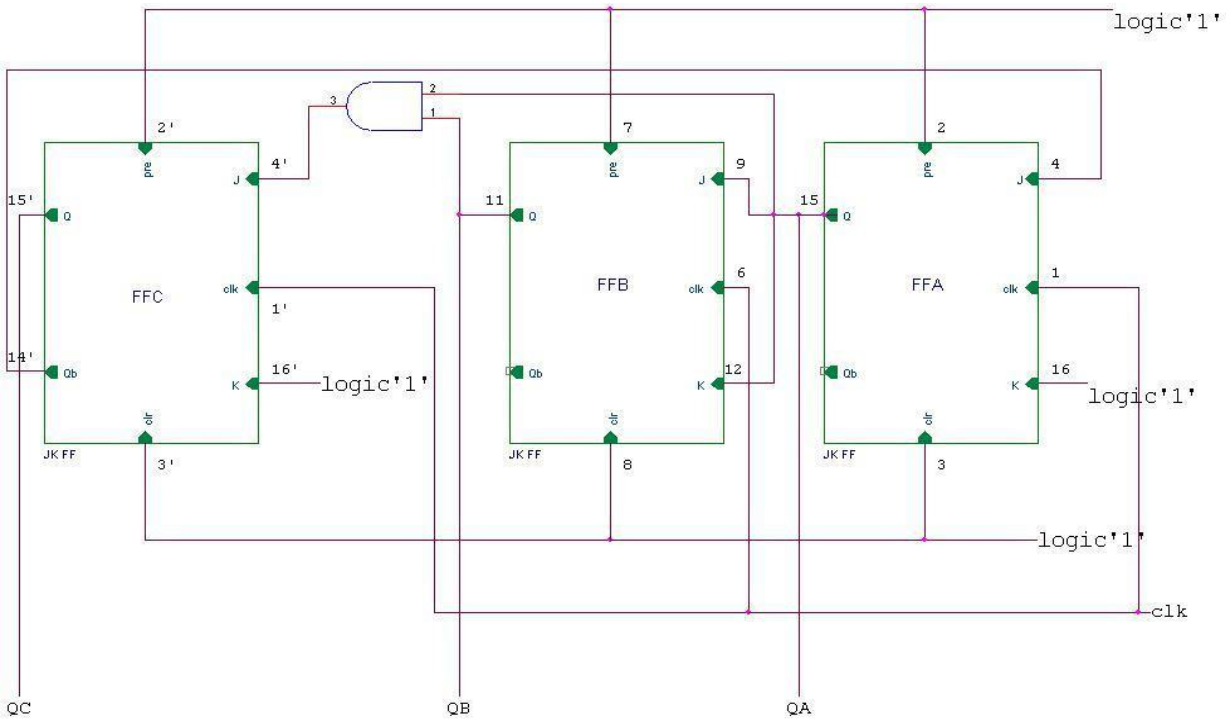
0	0	1	0
X	X	X	X

$J_C = Q_B Q_A$

X	X	X	X
1	X	X	X

$K_c = 1$

CIRCUIT DIAGRAM



MOD 8 COUNTER:

TRUTH TABLE:

QC	QB	QA
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

JK FF excitation table:

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

DESIGN:

1	X	X	1
1	X	X	1

$J_A = 1$

X	1	1	X
X	1	1	X

$K_A = 1$

0	0	1	0
X	X	X	X

0	1	X	X
X	1	X	X

JB=QA

$J_C = Q_B Q_A$

X	X	1	0
X	X	1	0

KB=QA

X	X	X	X
0	0	1	0

$K_C = Q_B Q_A$

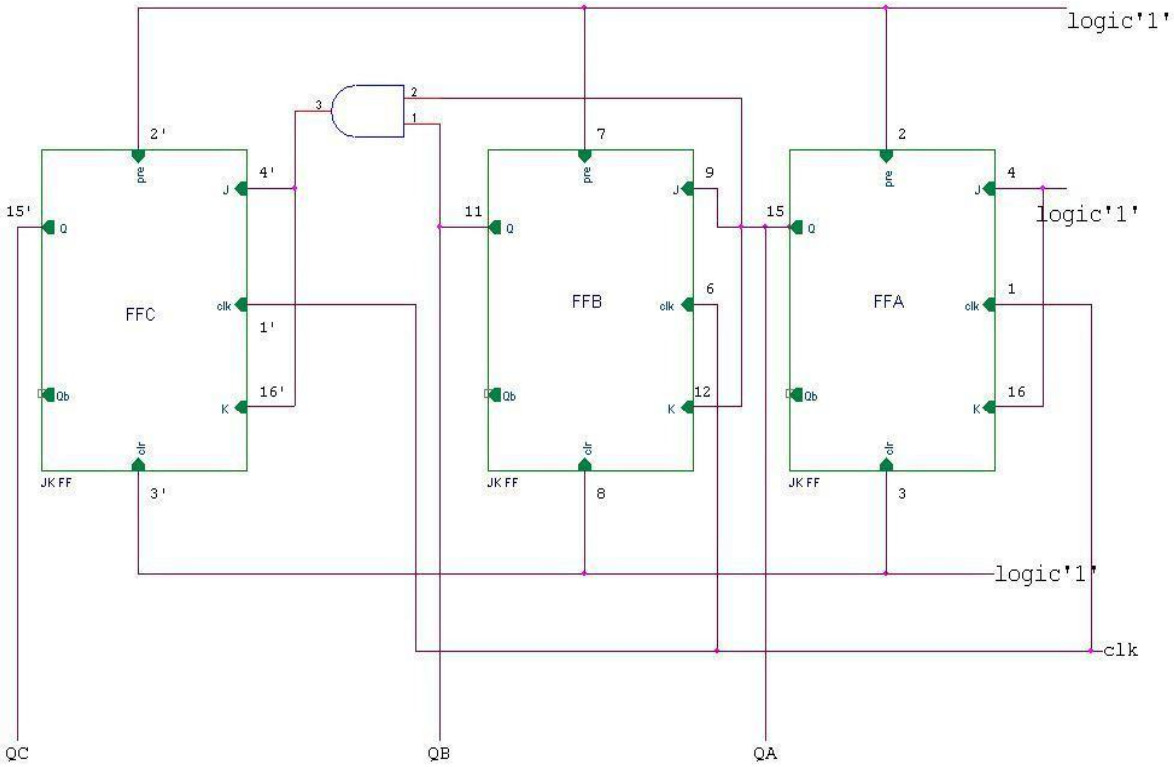
Present count

QC	QB	QA
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Next count

QC	QB	QA
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

CIRCUIT DIAGRAM



RESULT:

The working of synchronous Mod-N counters is verified.

VIVA QUESTIONS:

1. What are synchronous counters?
2. What are the advantages of synchronous counters?
3. What is an excitation table?
4. Write the excitation table for D, T FF
5. Design mod-5 synchronous counter using T FF

EXPERIMENT NO: 11

SHIFT REGISTER:STUDY OF SHIFT RIGHT,SIPO,SISO,PIPO,PISO(Using FF & 7495)

AIM:

To realize and study of Shift Register.

1) SISO (Serial in Serial out)
2) SIPO (Serial in Parallel out)
3) PIPO (Parallel in Parallel out)
4) PISO (Parallel in Serial out)

COMPONENTS REQUIRED: IC 7495, Patch Cords & IC Trainer Kit.

PROCEDURE:

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

2) SERIAL IN SERIAL OUT (SISO) (Right Shift)

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

2) SERIAL IN PARALLEL OUT (SIPO)

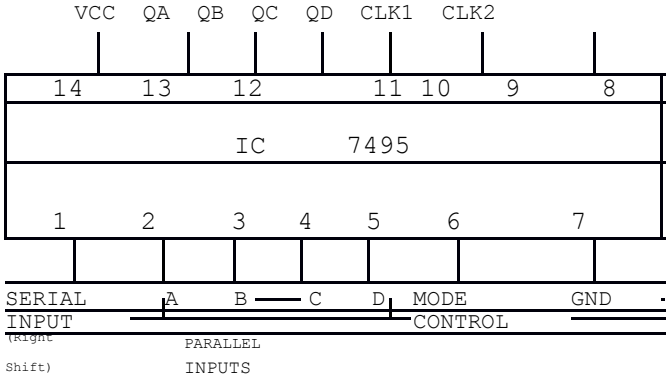
Serial i/p data	Shift Pulses	Q_A	Q_B	Q_C	Q_D
-	-	X	X	X	X
0	t1	0	X	X	X
1	t2	1	0	X	X
0	t3	0	1	0	X
1	t4	**1**	**0**	**1**	**0**

3) PARALLEL IN PARALLEL OUT (PIPO)

Clock Input Terminal	Shift Pulses	Q_A	Q_B	Q_C	Q_D
-	-	X	X	X	X
CLK_2	t1	1	0	1	0

4) PARALLEL IN SERIAL OUT (PISO)

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



RESULT: The various operations of a shift register is verified

EXPERIMENT NO: 12

RING COUNTER AND JOHNSON COUNTER(Using FF & 7495)

AIM:

To realize and study Ring Counter and Johnson counter using FF & 7495

LEARNING OBJECTIVE:

To learn about Ring Counter and its application

To learn about Johnson Counter and its application

COMPONENTS REQUIRED:

IC 7495, IC 7404, Patch Cords & IC Trainer Kit.

THEORY:

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

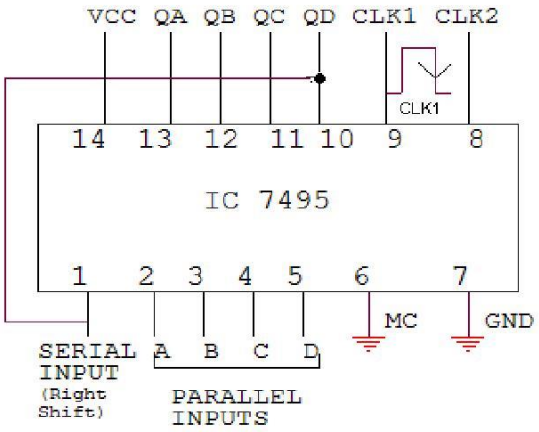
A basic ring counter can be slightly modified to produce another type of shift register counter called Johnson counter. Here complement of last output is connected back to the not gate input and not gate output is connected back to serial input. A four bit Johnson counter gives 8 state output.

PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply clock to pin number 9 and observe the output

CIRCUIT DIAGRAM:

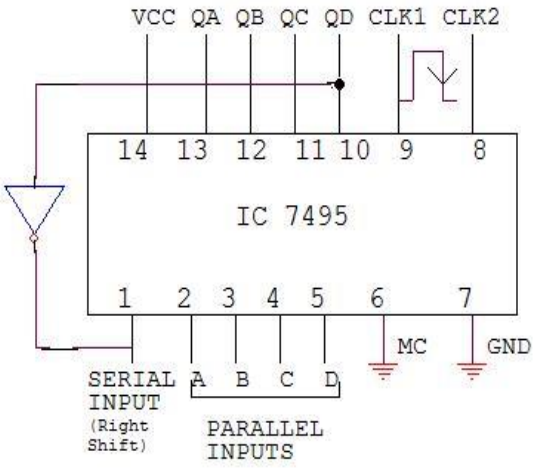
1) RING COUNTER



TRUTH TABLE

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

2) JOHNSON COUNTER TRUTH TABLE



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

RESULT: The truth table & working of Ring and Johnson counters is verified.

EXPERIMENT NO: 13

REALIZATION OF COUNTERS USING IC's (7490, 7492, 7493)

AIM: To design IC 7490 as a decade counter with BCD count sequence
To design IC 7492 as a divide by 12 counter
To design IC 7493 as a divide by 16 counter

LEARNING OBJECTIVE:

To learn about decade Counter
To use it as a divide by N counter [N<=10 ,say N=7,N=5]

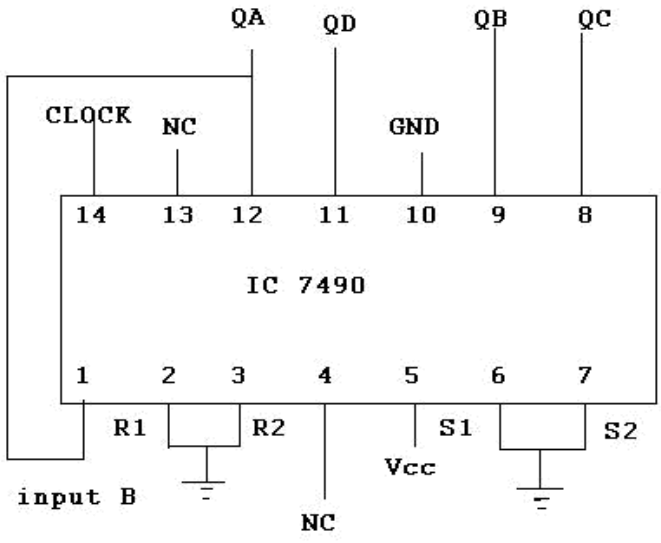
COMPONENTS REQUIRED:

IC 7490, Patch Cords & IC Trainer Kit

PROCEDURE:

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

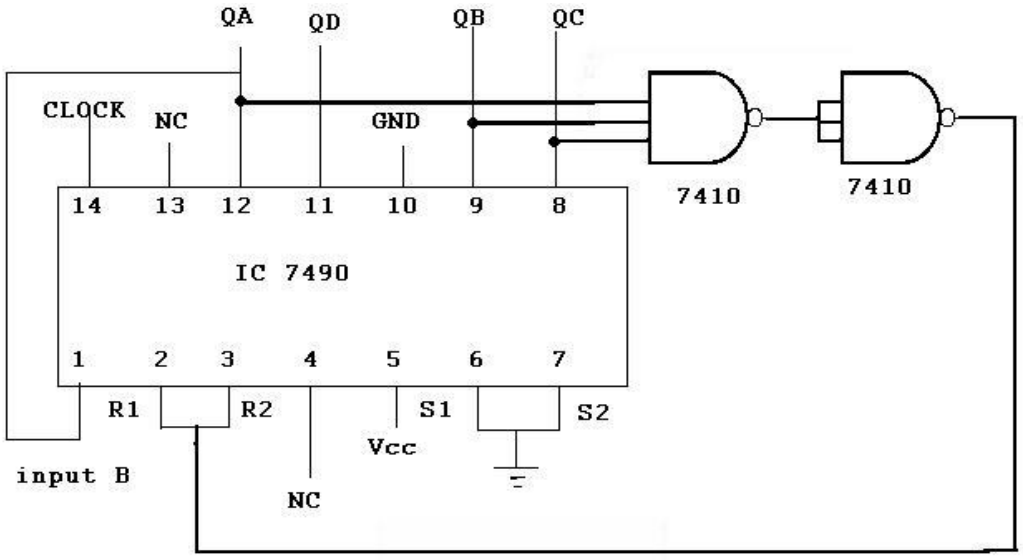
DECADE COUNTER:



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

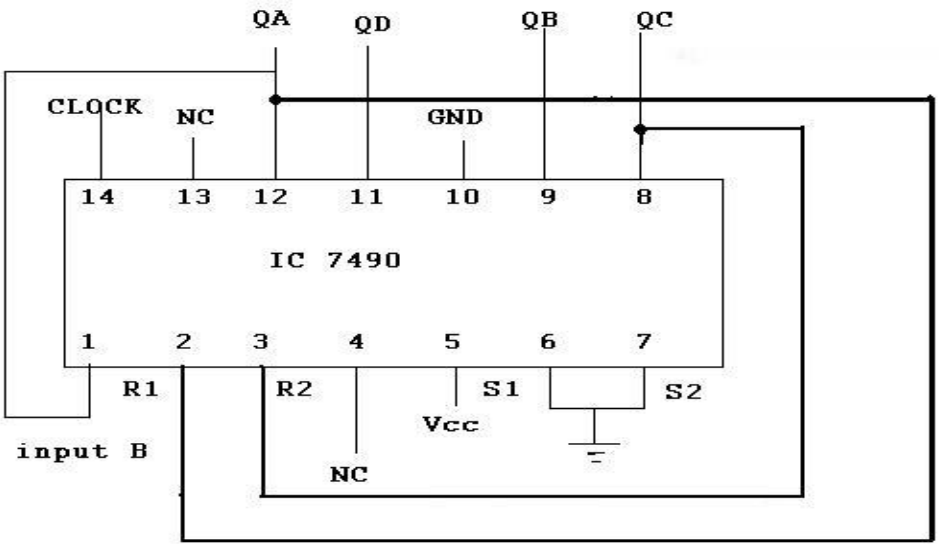
7490 AS A DIVIDE BY N COUNTER (N=7):



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	0	0	0

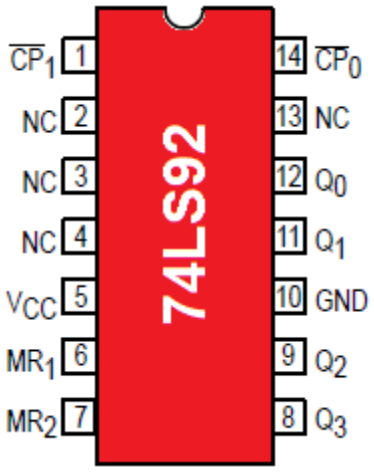
DIVIDE BY 5 COUNTER:



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	0	0	0

7492 IC (DIVIDE BY 12 COUNTER)



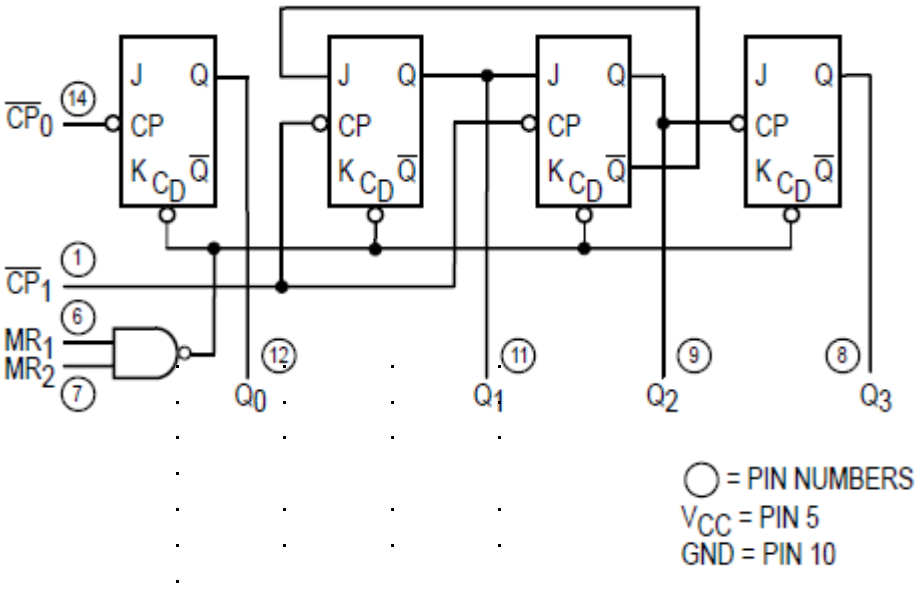
The 7492 / 74LS92 counter IC is a 4-Bit ripple counter (4 cascaded counting elements). The

individual counting circuits inside are partitioned into two blocks, one is a divide by two counter and the other capable of divide by 6, which when combined together effectively implements a divide by 12, perfect for the hour tracking register for a digital clock.

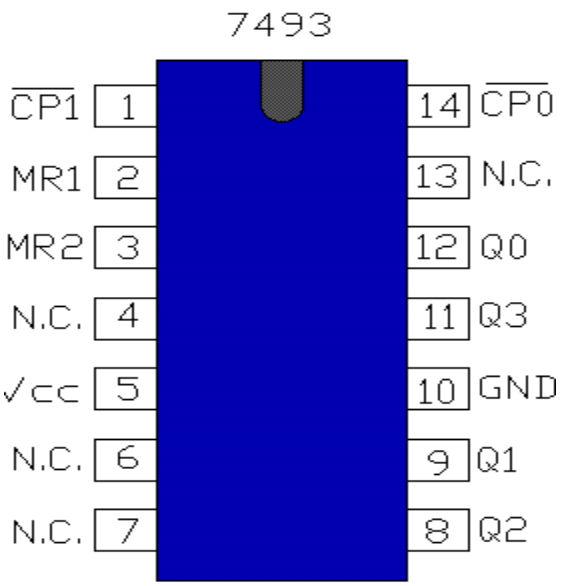
For the decade counter version refer to 74LS90.
For the hexadecimal counter (divide by 16 / 4-bit counter) refer to 74LS93.

The counter contains gated master clear circuitry.

74LS92 (Divide by 12 Counter)



7493 -4 BIT-BINARY RIPPLE COUNTER WITH ASYNCHRONUS RESET



Features:

· Various counting modes
· Asynchronous master reset
· Output capability: standard
· ICC category: MSI

A ripple counter, properly known as an asynchronous counter, is a cascade of digital registering units. Each one of which relies on the output of the one before it, except, of course, the first one in the chain. Since this chip is a 4 bit counter, it consists of 4 independent latch circuits which are daisy-chained together. While fine in ordinary digital clocks, ripple counters can cause glitchy results when used in precision instruments to the delay in updating the full chain of registers.

The 7493 can be wired for various modes of operations, including the full four bit mode (0-15 in decimal) and also shorter counts such as 0-7 or 0-3 etc... This configurability allows the device to be suited in a wide variety of different areas such as frequency dividers and special control applications. Multiple 7493s can be configured together to provide greater counting lengths and more division possibilities

RESULT: The working of IC 7490 ,7492,7493 are verified

VIVA QUESTIONS:

What is a decade counter?
What do you mean by a ripple counter?
Explain the design of Modulo-N counter ($N \leq 9$) using IC 7490

EXPERIMENT NO: 14

MULTIPLEXERS AND DEMULTIPLEXERS USING GATES AND ICs(74150,74154)

AIM:

To design and set up a 4:1 Multiplexer (MUX) using only NAND gates.
To design and set up a 1:4 Demultiplexer(DE-MUX) using only NAND gates.
To verify the various functions of IC 74153(MUX) and IC 74139(DEMUX).

LEARNING OBJECTIVE:

To learn about various applications of multiplexer and de-multiplexer To learn and understand the working of IC 74153 and IC 74139
To learn to realize any function using Multiplexer

THEORY:

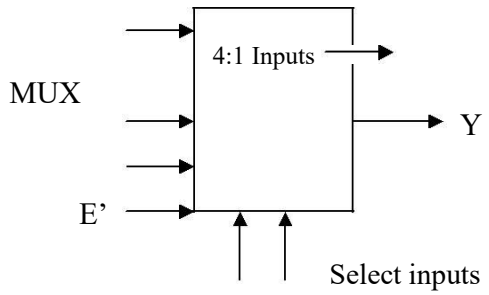
Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output.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

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

COMPONENTS REQUIRED:

IC 7400, IC 7410, IC 7420, IC 7404, IC 74153, IC 74139, Patch Cords & IC Trainer Kit.

i) 4:1 MULTIPLEXER

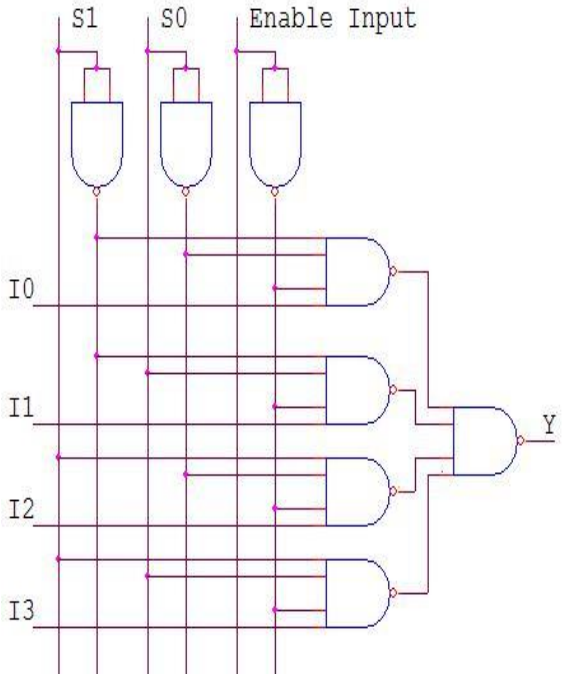


Output $Y = E'S_1'S_0'I_0 + E'S_1'S_0I_1 + E'S_1S_0'I_2 + E'S_1S_0I_3$

ii) DE-MUX USING NAND GATES

Enable Inputs	Data Input	Select Inputs		Outputs			
E	**D**	**S_1**	**S_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
1	0	X	X	X	X	X	X
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	0	0	0

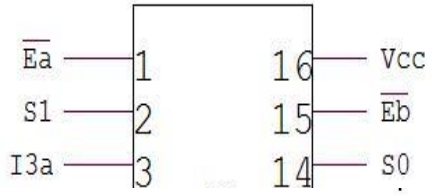
REALIZATION USING NAND GATES

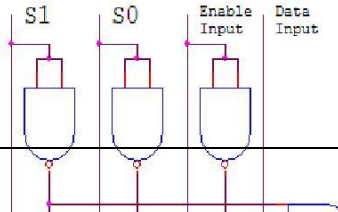


TRUTH TABLE

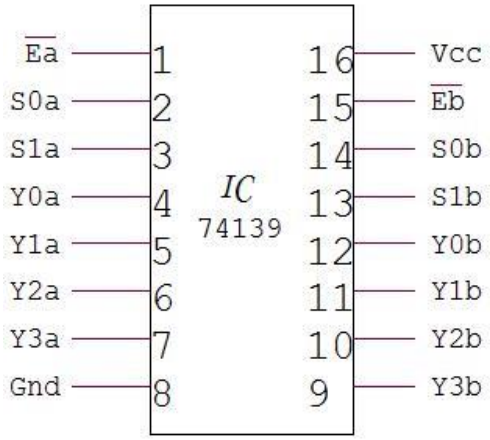
Select Inputs		Enable Input	Inputs				Out puts
S_1	**S_0**	**E**	**I_0**	**I_1**	**I_2**	**I_3**	**Y**
X	X	1	X	X	X	X	0
0	0	0	0	X	X	X	0
0	0	0	1	X	X	X	1
0	1	0	X	0	X	X	0
0	1	0	X	1	X	X	1
1	0	0	X	X	0	X	0
1	0	0	X	X	1	X	1
1	1	0	X	X	X	0	0
1	1	0	X	X	X	1	1

VERIFY IC 74153 MUX (DUAL 4:1 MULTIPLEXER)





VERIFICATION OF IC 74139 (DEMUX)



TRUTH TABLE

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

PROCEDURE:

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

RESULT: Studied multiplexers and demultiplexersusing gates and IC.

EXPERIMENT NO: 15

REALIZATION OF COMBINATIONAL CIRCUITS USING MUX & DEMUX

AIM:

To design and set up a half adder/full adder using MUX

COMPONENTS REQUIRED:

IC 7400, IC 7410, IC 7420, IC 7404, IC 74153, IC 74139, Patch Cords & IC Trainer Kit.

THEORY:

Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer. Some of the characteristics of combinational circuits are following −

- The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
- The combinational circuit does not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have an n number of inputs and m number of outputs

HALF ADDER USING MUX:

DESIGN:

SUM

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

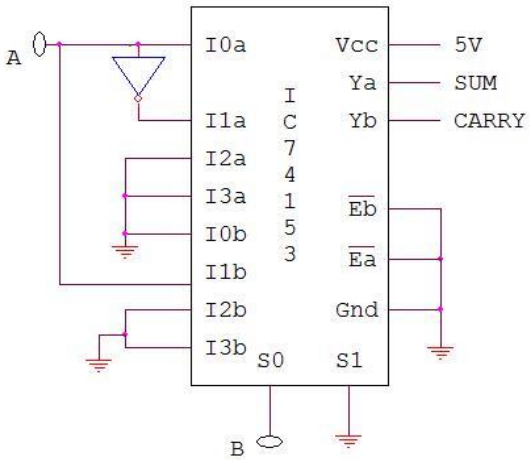
CARRY

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

TRUTH TABLE

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

CIRCUIT:



FULL ADDER USING MUX:

DESIGN:

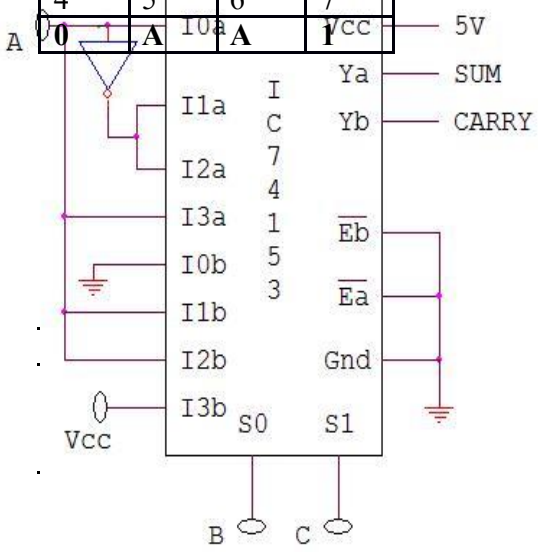
I0	I1	I3	I3
0	1	2	3
4	5	6	7
A	**A'**	**A'**	**A**

TRUTH TABLE

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

FULL ADDER CIRCUIT

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



HALF SUBTRACTOR USING MUX:

DESIGN:

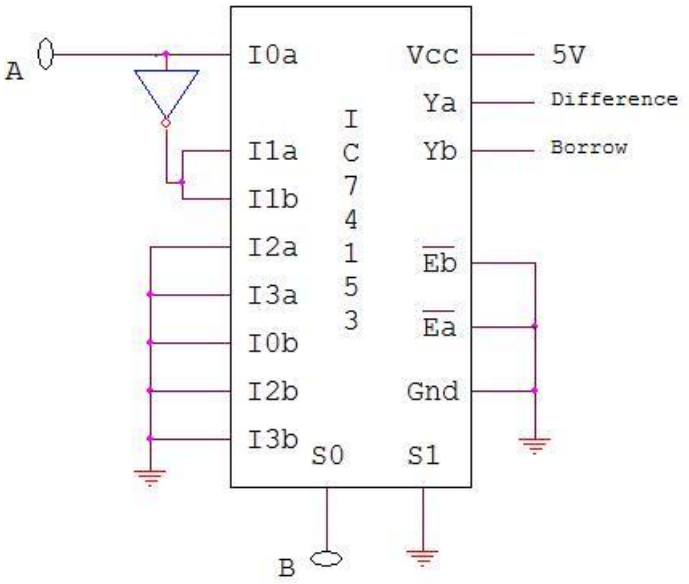
DIFFERENCE

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

BORROW

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

CIRCUIT:



TRUTH TABLE

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

FULL SUBTRACTOR USING MUX:

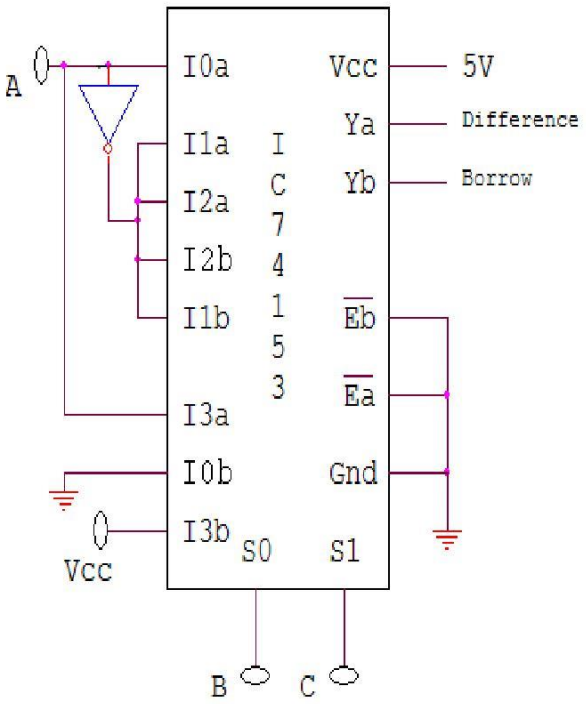
DESIGN:

DIFFERENCE

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

BORROW

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



TRUTH TABLE

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

PROCEDURE:

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

RESULT: Adder and subtractor circuits are realized using multiplexer IC 74153.

VIVA QUESTIONS:

1) What is a multiplexer?
2) What is a de-multiplexer?
3) What are the applications of multiplexer and de-multiplexer?
4) Derive the Boolean expression for multiplexer and de-multiplexer.
5) How do you realize a given function using multiplexer.
6) What is the difference between multiplexer & demultiplexer?
7) In 2n to 1 multiplexer how many selection lines are there?
8) How to get higher order multiplexers?
9) Implement an 8:1 mux using 4:1 mux.

EXPERIMENT NO: 16

RANDOM SEQUENCE GENERATORS

AIM: To design and set up a Sequence Generator using IC 7495.

COMPONENTS REQUIRED: IC 7495, IC 7486, Patch Cords & IC Trainer Kit.

PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- By Keeping mode=1. Load the input A,B,C,D as in Truth Table 1st Row and give a clock pulse
- For count mode make mode = 0.
- Verify the Truth Table and observe the outputs.

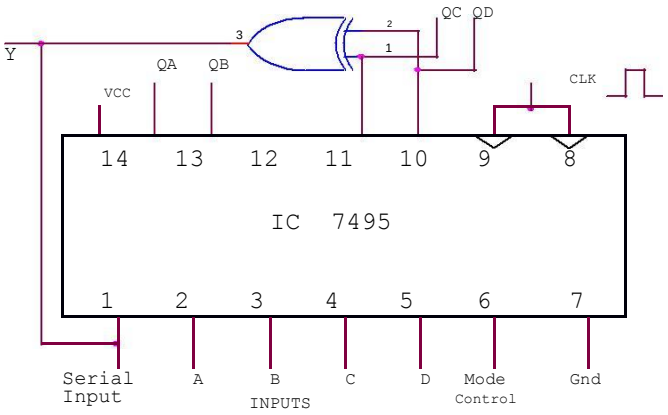
DESIGN 1:
Sequence = 100010011010111
Sequence length S = 15

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

1	1	1	0	1
	1	1	1	
		1	1	
			1	

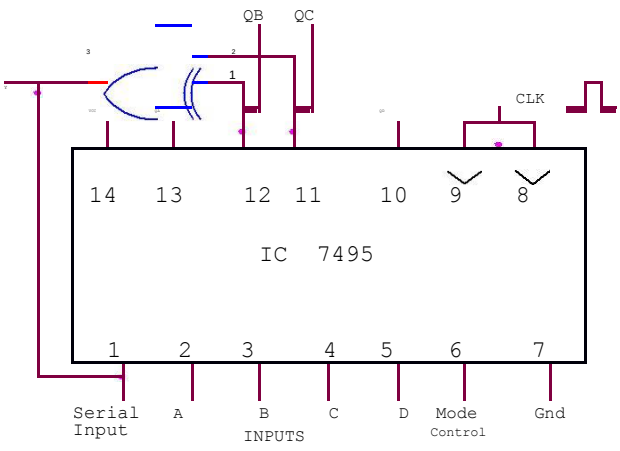
$Y = Q_C (+) Q_D$

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



DESIGN 2:
Sequence = 1001011
Sequence length S = 7

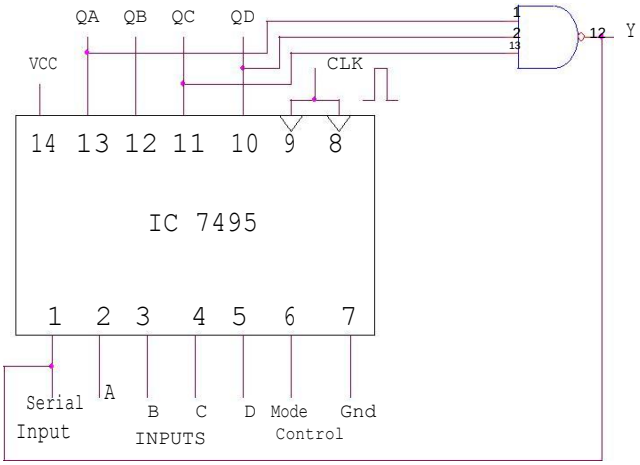
$Y = Q_B \oplus Q_C$



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

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

DESIGN 3:
Sequence = 1101011
Sequence length S = 7



$$Y = Q_A + Q_C + Q_D$$

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

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

VIVA QUESTIONS:

1) What is the necessity for sequence generation?
2) What are PISO, SIPO, and SISO with respect to shift register?
3) Differentiate between serial data & parallel data.
4) What is the significance of Mode control bit?
5) What is a ring counter?
6) What is a Johnson counter?
7) How many Flip-flops are present in IC 7495?

EXPERIMENT NO: 17

BCD TO 7-SEGMENT DECODER/DRIVER CHIP TO DRIVE LED DISPLAY

AIM:

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

LEARNING OBJECTIVE:

To learn about various applications of decoder
To learn and understand the working of IC 7447
To learn about types of seven-segment display

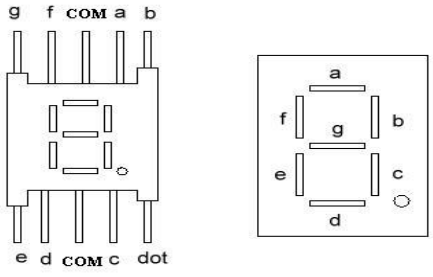
COMPONENTS REQUIRED:

IC7447, 7-Segment display (common anode), Patch chords, resistor (1K) & IC Trainer Kit

THEORY:

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

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



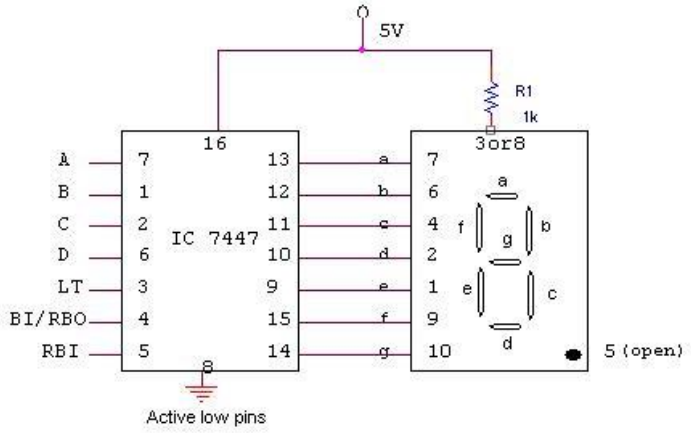
Seven-Segment Display

LED’s are basically of two types-

Common Cathode (CC) -All the 8 anode legs uses only one cathode, which is common. Common Anode (CA)-The common leg for all the cathode is of Anode type.

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

CIRCUIT DIAGRAM:



TRUTH TABLE:

BCD Inputs				**Output Logic Levels from IC 7447 to 7-segments**							**Decimal number display**
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	1	1	0	0	9

PROCEDURE:

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

VIVA QUESTIONS:

1. What are the different types of LEDs?
2. Draw the internal circuit diagram of an LED.
3. What are the applications of LEDs?