MAHA BARATHI ENGINEERING COLLEGE

NH-79, SALEM-CHENNAI HIGHWAY, A.VASUDEVANUR, CHINNASALEM (TK), KALLAKURICHI (DT) 606 201.
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
Accredited by NAAC and Recognized under section 2(f) & 12(B) status of UGC, New Delhi

www.mbec.ac.in | Ph: 04151-256333, 257333 | E-mail: mbec123@gmail.com

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

EC3352 DIGITAL SYSTEMS DESIGN LABORATORY

II Year/ III Semester B.E ECE

Regulation 2021

(As Per Anna University, Chennai syllabus)

Prepared By	Verified By	Approved By
Staff I/C	(HoD/ECE)	(Principal)

EC3352 DIGITAL SYSTEMS DESIGN LABORATORY

LIST OF EXPERIMENTS:

1. Study of Logic Gates
2. Design and implementation of Half/Full Adder and Subtractor using logic gates.
3. Design and implementation of code converters using logic gates
    i. Binary to gray code
    ii. Gray to Binary code
4. Design and implementation Code Converters using logic gates
    i. BCD to Excess 3
    ii. Excess 3 to BCD
5. Design and implementation of Multiplexer and De-multiplexer using logic gates and study of IC74150 and IC 74154.
6. Design and implementation of encoder and decoder using logic gates and study of IC7445 and IC74147.
7. Design and implementation of Magnitude Comparators using logic gates and study of IC 7485.
8. Construction and verification of 4 bit ripple counter and Mod-10 / Mod-12 Ripple counters.
9. Design and implementation of 3-bit synchronous up/down counter.
10. Design and implementation of shift registers SISO, SIPO.
11. Design and implementation of shift registers PISO, PIPO.

INDEX

Expt.No.:1	STUDY OF LOGIC GATES
Date:	

AIM:

To study about logic gates and verify their truth tables.

APPARATUS REQUIRED:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	NAND GATE 2 I/P	IC 7400	1
5.	NOR GATE	IC 7402	1
6.	X-OR GATE	IC 7486	1
7.	NAND GATE 3 I/P	IC 7410	1
8.	IC TRAINER KIT	-	1
9.	PATCH CORD	-	14

THEORY:

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

AND GATE:

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

OR GATE:

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

NOT GATE:

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

NAND GATE:

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

NOR GATE:

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

X-OR GATE:

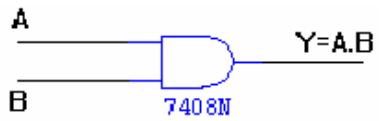
The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

PROCEDURE:

1. Place the IC on IC Trainer Kit.
2. Connect VCC and ground to respective pins of IC Trainer Kit.
3. Connections are given as per circuit diagram.
4. Logical inputs are given as per circuit diagram.
5. Observe the output and verify the truth table.

AND GATE:

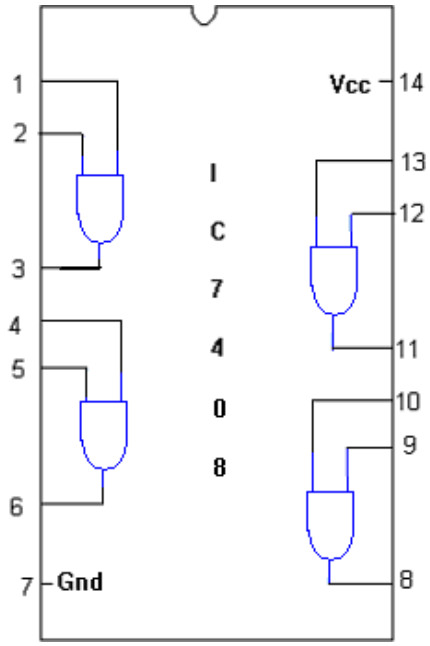
SYMBOL:



TRUTH TABLE

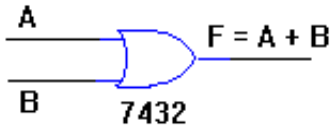
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

PIN DIAGRAM:



OR GATE:

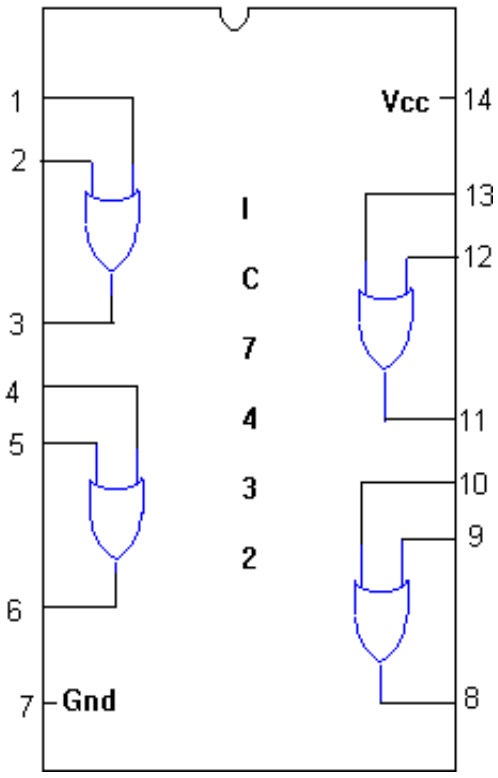
SYMBOL :



TRUTH TABLE

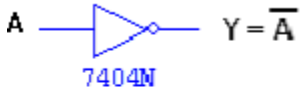
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

PIN DIAGRAM :

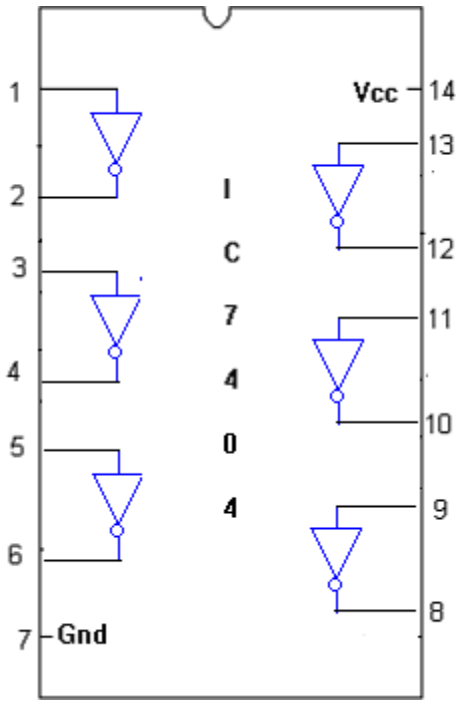


NOT GATE:

SYMBOL:



PIN DIAGRAM:

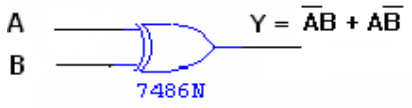


TRUTH TABLE :

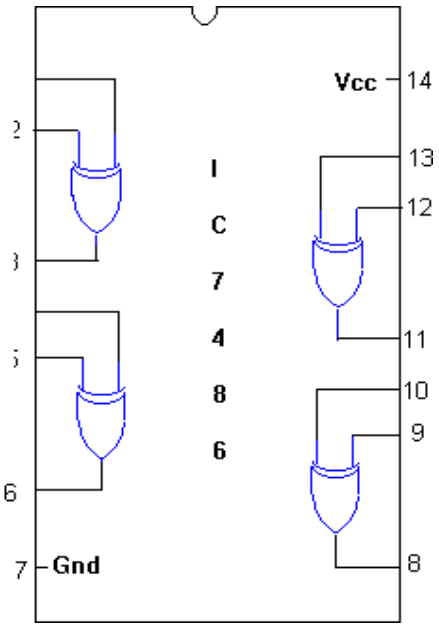
A	$\overline{A}$
0	1
1	0

X- OR GATE :

SYMBOL :



PIN DIAGRAM :

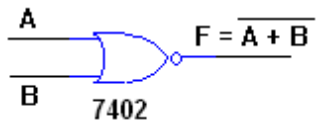


TRUTH TABLE :

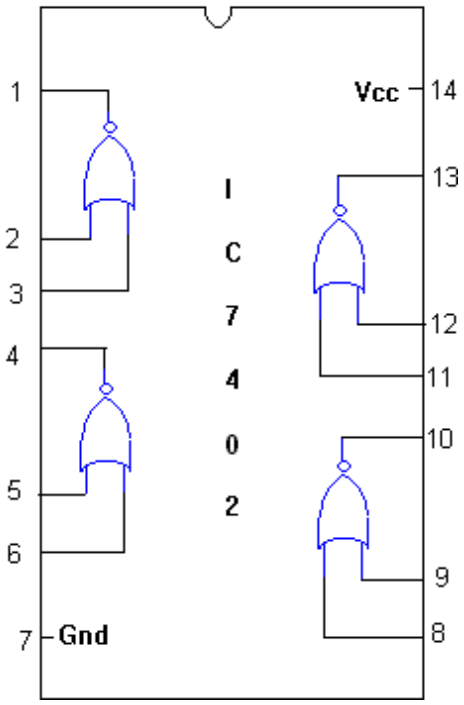
A	B	$\overline{A}B + A\overline{B}$
0	0	0
0	1	1
1	0	1
1	1	0

NOR GATE

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

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

RESULT

The function of logic gates has been verified.

VIVA QUESTIONS

1. What are logic gates?
2. What is a logic circuit?
3. What is logic design?
4. What is a truth table?
5. Why AND, OR and NOT gates are called basic gates?
6. What is mean by universal gates?

Expt.No.:2	DESIGN AND IMPLEMENTATION OF HALF/FULL ADDER AND SUBTRACTOR USING LOGIC GATES
Date:	

AIM:

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

APPARATUS 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
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	23

THEORY:

HALF ADDER:

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

FULL ADDER:

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

HALF SUBTRACTOR:

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

FULL SUBTRACTOR:

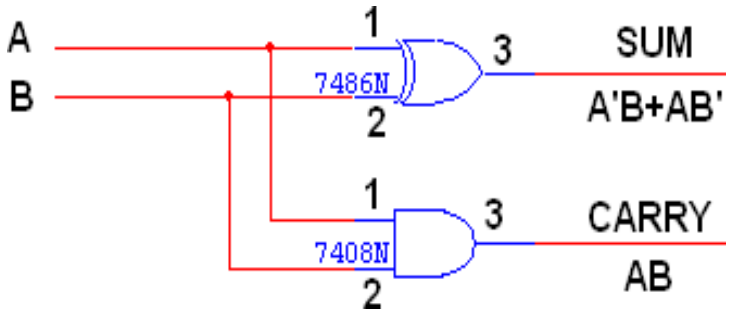
The full subtractor is a combination of X-OR, AND, OR, NOT Gates. In a full subtractor the logic circuit should have three inputs and two outputs.The two half subtractor put together gives a full subtractor .The first half subtractor will be C and A B. The output will be difference output of full subtractor.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

PROCEDURE:

(i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

LOGIC DIAGRAM

HALF ADDER



TRUTH TABLE:

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

K-Map for SUM:

A \ B	00	01
00		1
01	1	

SUM = A'B + AB'

K-Map for CARRY:

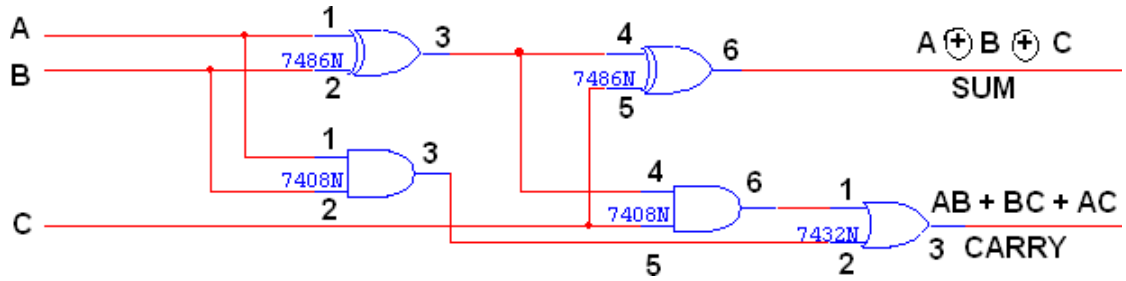
A \ B	00	01
00		
01		1

CARRY = AB

LOGIC DIAGRAM:

FULL ADDER

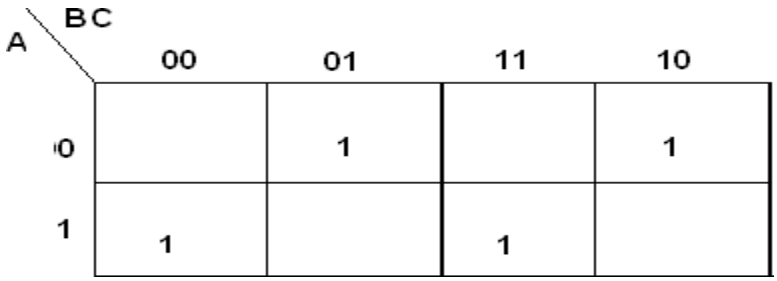
FULL ADDER USING TWO HALF ADDER



TRUTH TABLE:

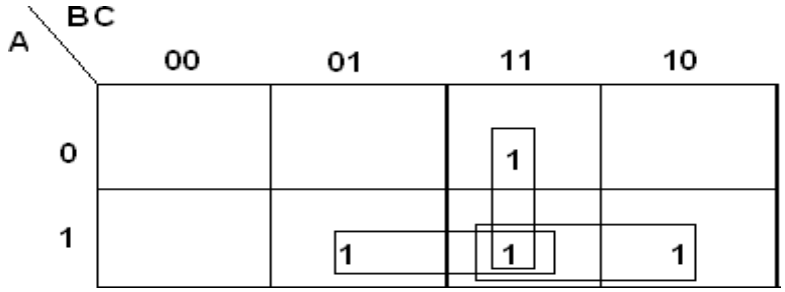
A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

K-Map for SUM:



SUM = A'B'C + A'BC' + ABC' + ABC

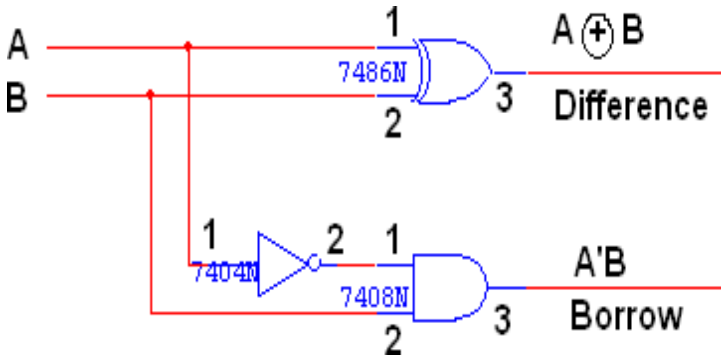
K-Map for CARRY:



CARRY = AB + BC + AC

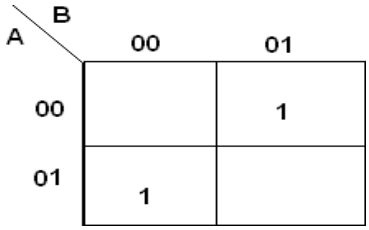
LOGIC DIAGRAM:

HALF SUBTRACTOR



TRUTH TABLE:

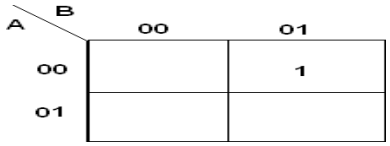
A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0



K-Map for DIFFERENCE:

DIFFERENCE = A'B + AB'

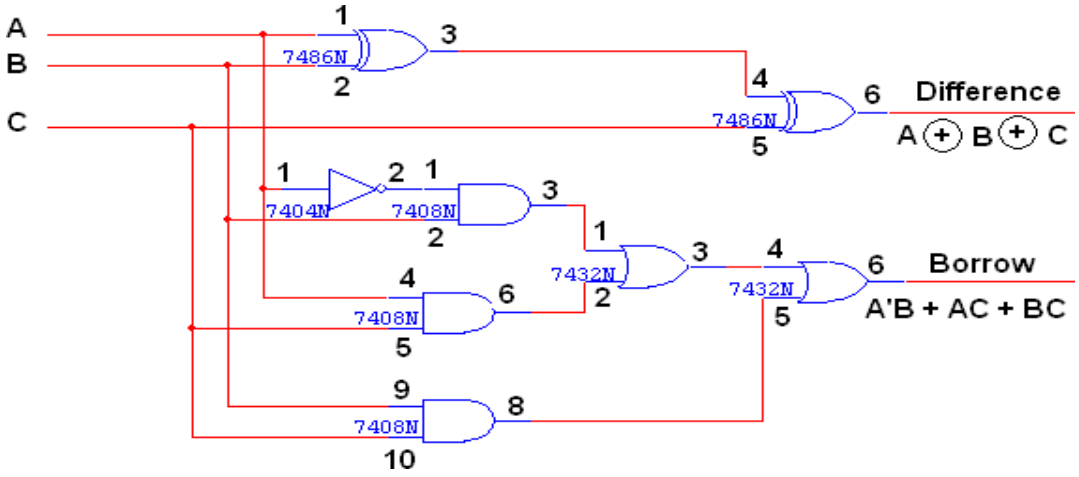
K-Map for BORROW:



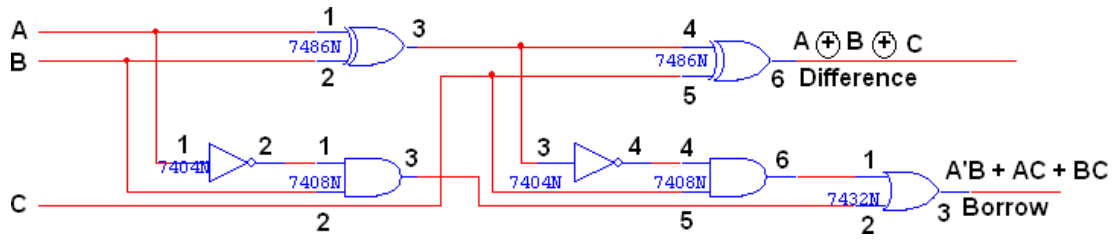
BORROW = A'B

LOGIC DIAGRAM:

FULL SUBTRACTOR



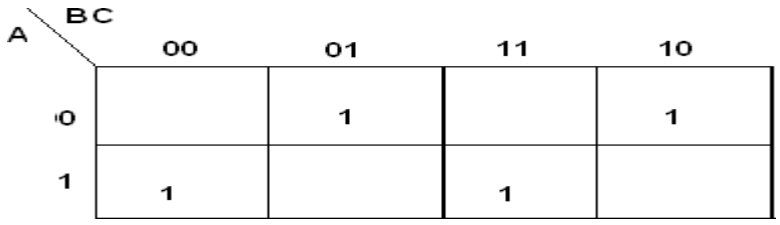
FULL SUBTRACTOR USING TWO HALF SUBTRACTOR:



TRUTH TABLE:

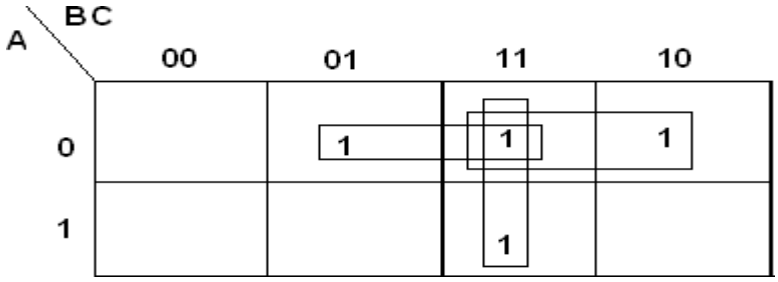
A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-Map for Difference:



Difference = A'B'C + A'BC' + AB'C' + ABC

K-Map for Borrow:



Borrow = A'B + BC + A'C

RESULT:

Thus, the half adder, full adder, half subtractor and full subtractor circuits are designed, constructed and verified the truth table using logic gates.

VIVA QUESTIONS:

1) What is a half adder?

2) What is a full adder?

3) What are the applications of adders?

4.What is half subtractor?

5. What is full subtractor?

Expt.No.:3	DESIGN AND IMPLEMENTATION OF CODE CONVERTERS
Date:	

AIM:

To design and implement 4-bit

(i) Binary to gray code converter
(ii) Gray to binary code converter

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	XOR Gate	IC 7486	1
2.	AND Gate	IC 7408	1
3.	OR Gate	IC 7432	1
4.	NOT Gate	IC 7404	1
5.	IC TRAINER KIT	-	1
6.	PATCH CORDS	-	35

THEORY:

Code is a symbolic representation of discrete information. Codes are of different types. 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.

Gray Code is one of the most important codes. It is a non-weighted code which belongs to a class of codes called minimum change codes. In this codes while traversing from one step to another step, only one bit in the code group changes. In case of Gray Code two adjacent code numbers differs from each other by only one bit.

The input variable are designated as B3, B2, B1, B0 and the output variables are designated as G3, G2, G1, G0. From the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable. A two-level logic diagram may be obtained directly from the Boolean expressions derived by the k - map.

PROCEDURE:

i. Connections were given as per circuit diagram.
ii. Logical inputs are given through switches as per truth table
iii. Logical output values are observed in the LED outputs and verified with the truth tables.

BINARY TO GRAY CODE CONVERTER:

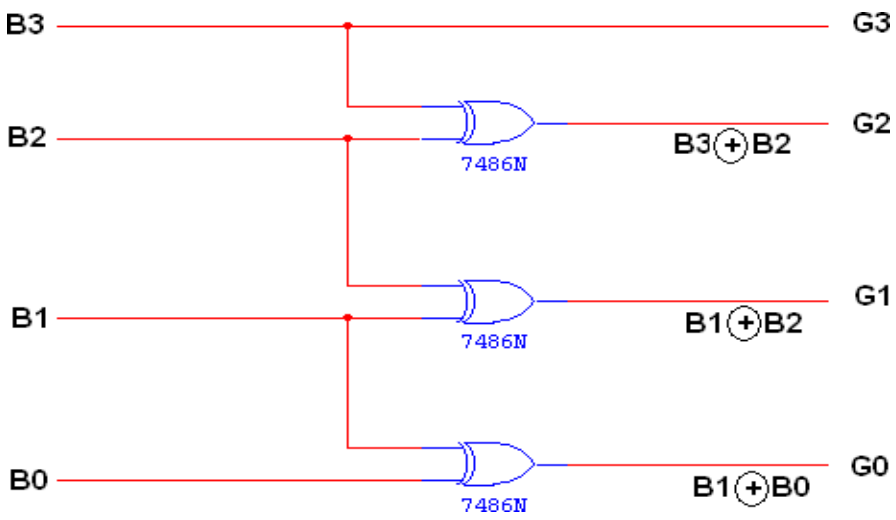
TRUTH TABLE:

Binary Input					**Gray code output**			
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**

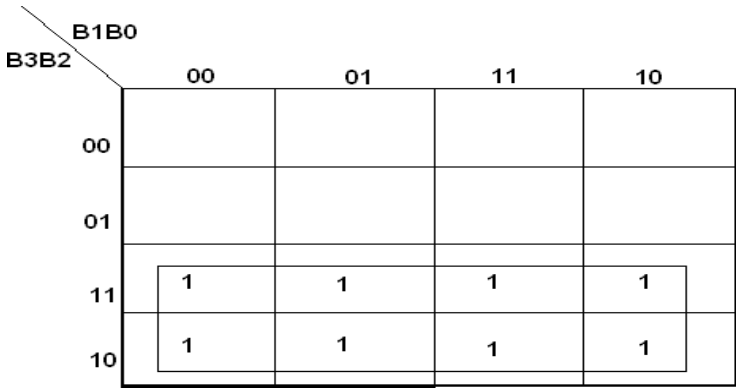
1	1	0	1		1	0	1	1
1	1	1	0		1	0	0	1
1	1	1	1		1	0	0	0

LOGIC DIAGRAM:

BINARY TO GRAY CODE CONVERTOR

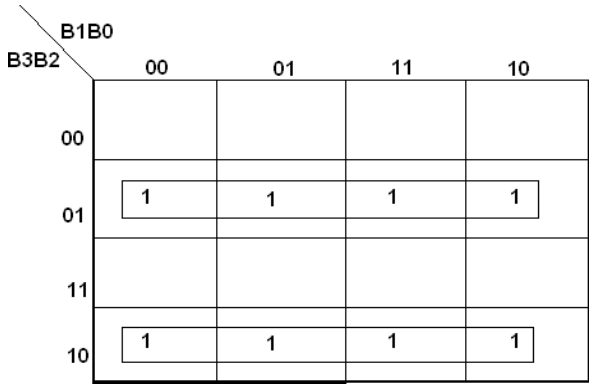


K-Map for G_3:



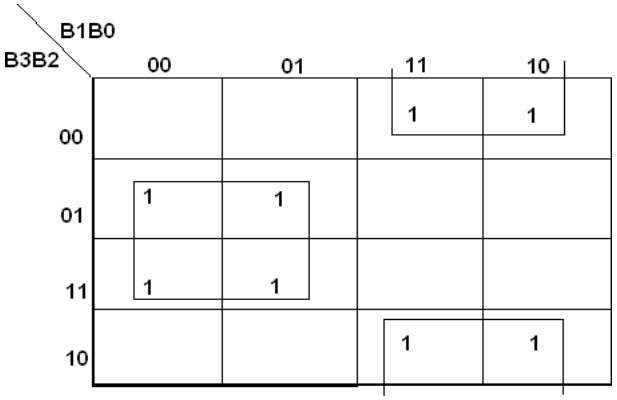
$G_3 = B_3$

K-Map for G_2:



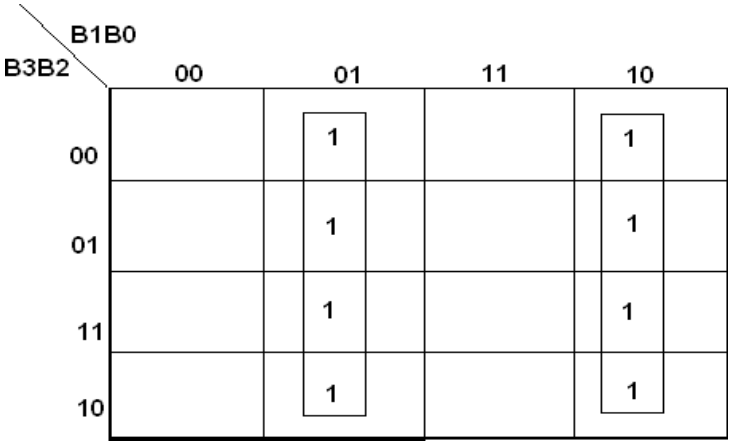
$G2 = B3 \oplus B2$

K-Map for G_1:



$G1 = B1 \oplus B2$

K-Map for G_0:



$G0 = B1 \oplus B0$

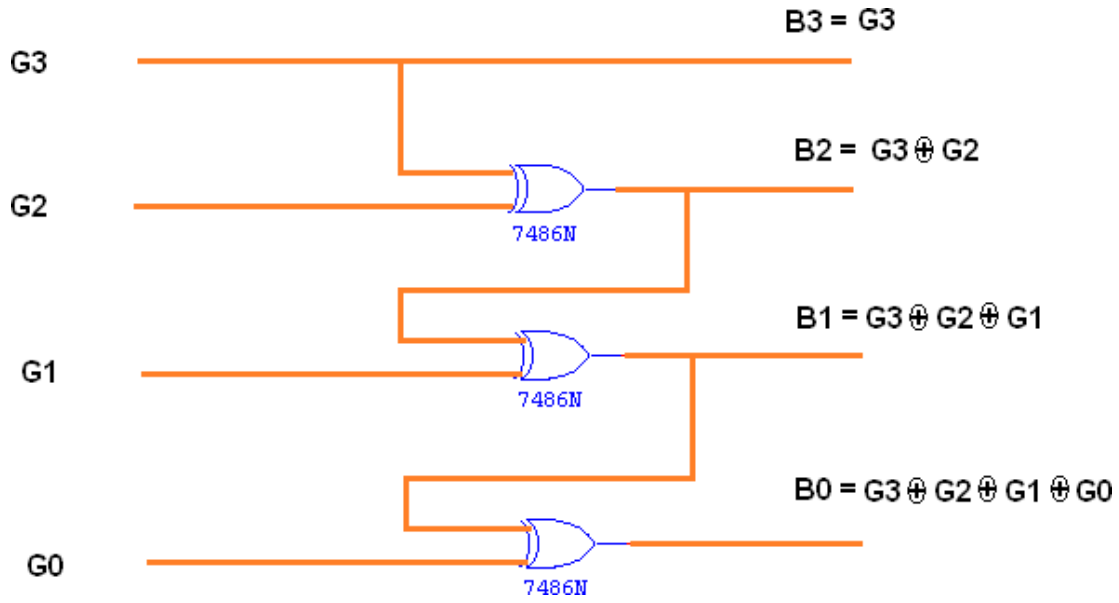
Gray Code to Binary Code Converter:

TRUTH TABLE:

Gray code input					Binary output			
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**	**0**		**0**	**0**	**1**	**1**
0	**0**	**1**	**1**		**0**	**0**	**1**	**0**
0	**1**	**0**	**0**		**0**	**1**	**1**	**1**
0	**1**	**0**	**1**		**0**	**1**	**1**	**0**
0	**1**	**1**	**0**		**0**	**1**	**0**	**0**
0	**1**	**1**	**1**		**0**	**1**	**0**	**1**
1	**0**	**0**	**0**		**1**	**1**	**1**	**1**
1	**0**	**0**	**1**		**1**	**1**	**1**	**0**
1	**0**	**1**	**0**		**1**	**1**	**0**	**0**
1	**0**	**1**	**1**		**1**	**1**	**0**	**1**
1	**1**	**0**	**0**		**1**	**0**	**0**	**0**
1	**1**	**0**	**1**		**1**	**0**	**0**	**1**
1	**1**	**1**	**0**		**1**	**0**	**1**	**1**
1	**1**	**1**	**1**		**1**	**0**	**1**	**0**

LOGIC DIAGRAM:

GRAY CODE TO BINARY CONVERTOR

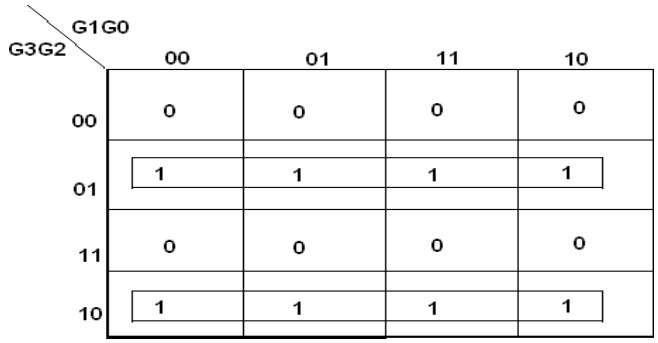


K-Map for B_3:

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

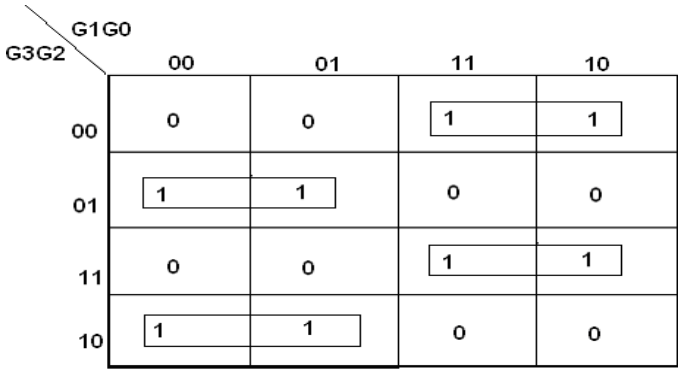
B3 = G3

K-Map for B_2:



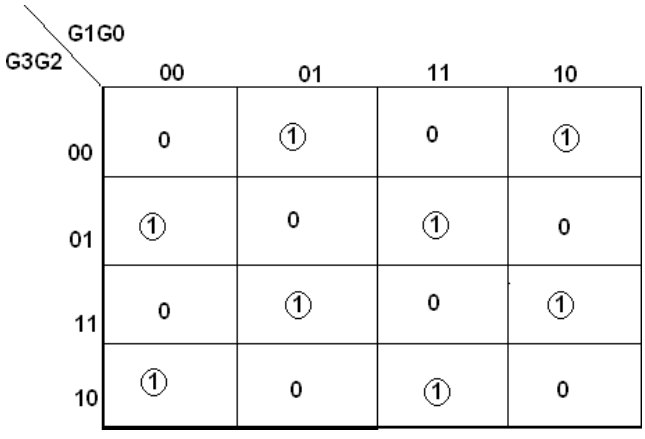
$B2 = G3 \oplus G2$

K-Map for B_1:



$B1 = G3 \oplus G2 \oplus G1$

K-Map for B_0:



$B0 = G3 \oplus G2 \oplus G1 \oplus G0$

RESULT:

Thus the code convertors circuits were designed using logic gates and their truth table were verified.

VIVA QUESTIONS

1. What is BCD?
2. What is the base for Binary?
3. How will you convert Binary to Gray?
4. What is Gray Code?
5. What are the logic gates used in converting Gray code to Binary?
6. What is the need of code converters?
7. How will you design Code converters?
8. What is the difference between Positive and Negative logic?

Expt.No.:4	DESIGN AND IMPLEMENTATION CODE CONVERTERS
Date:	

AIM:

To design and implement 4-bit

i) BCD to excess-3 code converter
ii)Excess-3 to BCD code converter

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	XOR Gate	IC 7486	1
2.	AND Gate	IC 7408	1
3.	OR Gate	IC 7432	1
4.	NOT Gate	IC 7404	1
5.	IC TRAINER KIT	-	1
6.	PATCH CORDS	-	35

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.The bit combination assigned to binary code to gray code. Since each code uses four bits to represent a decimal digit. There are four inputs and four outputs. Gray code is a non-weighted code.The input variable are designated as B3, B2, B1, B0 and the output variables are designated as C3, C2, C1, Co. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable.A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. To convert from binary codeto Excess-3 code, the input lines must

supply the bit combination ofelements as specified by code an the output line generate the corresponding bit combination of code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables. A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps. These are various other possibilities for a logic diagram that implements this circuit. Now the OR gate whose output isC+D has been used to implement partially each of three outputs.

PROCEDURE:

(i) Connections were given as per circuit diagram.

(ii) Logical inputs are given through switches as per truth table

(iii) Logical output values are observed in the LED outputs and verified with the truth tables.

BCD to Excess-3 Code Converter:

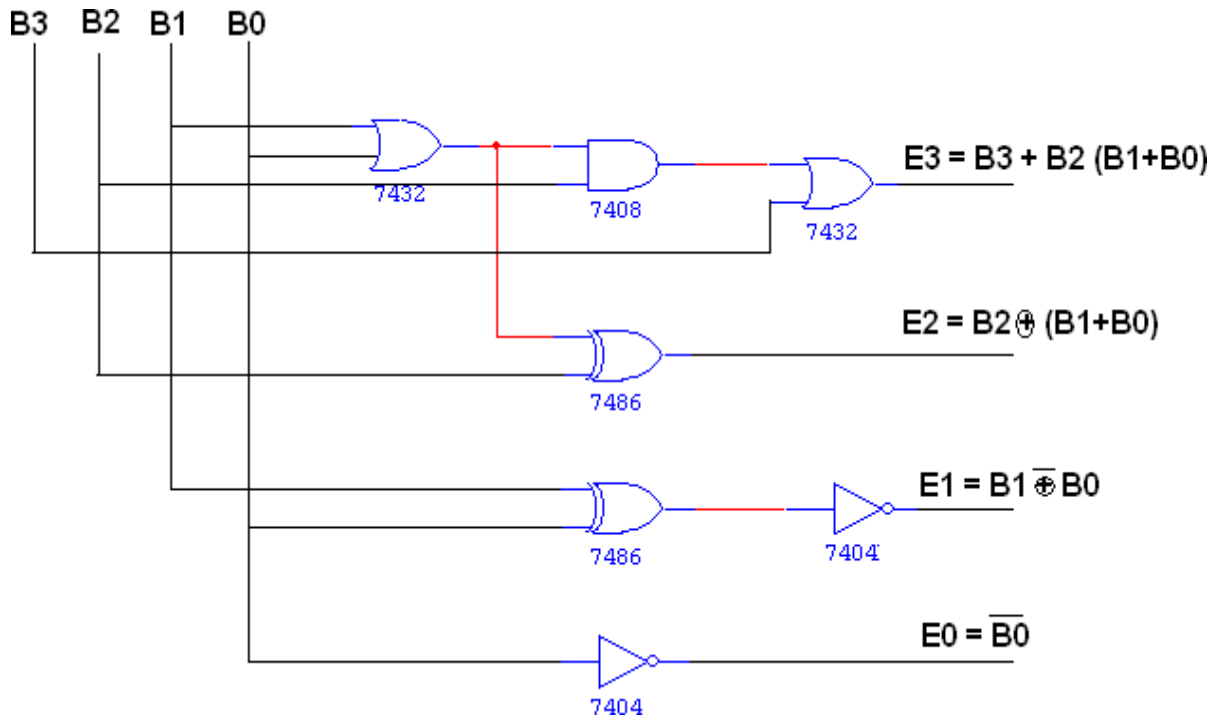
TRUTH TABLE:

BCD input					**Excess - 3 output**			
B3	**B2**	**B1**	**B0**		**E3**	**E2**	**E1**	**E0**
0	**0**	**0**	**0**		**0**	**0**	**1**	**1**
0	**0**	**0**	**1**		**0**	**1**	**0**	**0**
0	**0**	**1**	**0**		**0**	**1**	**0**	**1**
0	**0**	**1**	**1**		**0**	**1**	**1**	**0**
0	**1**	**0**	**0**		**0**	**1**	**1**	**1**
0	**1**	**0**	**1**		**1**	**0**	**0**	**0**
0	**1**	**1**	**0**		**1**	**0**	**0**	**1**
0	**1**	**1**	**1**		**1**	**0**	**1**	**0**
1	**0**	**0**	**0**		**1**	**0**	**1**	**1**
1	**0**	**0**	**1**		**1**	**1**	**0**	**0**
1	**0**	**1**	**0**		**x**	**x**	**x**	**x**
1	**0**	**1**	**1**		**x**	**x**	**x**	**x**

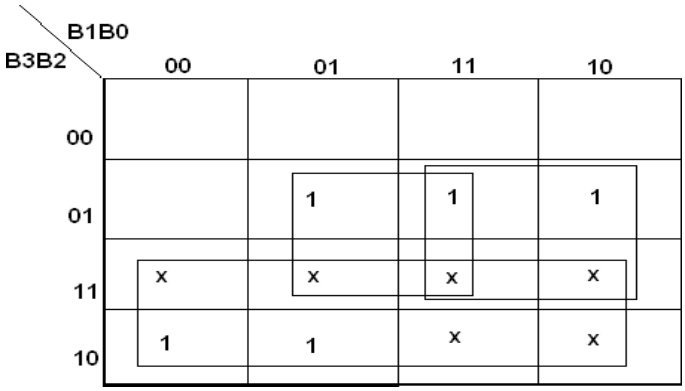
1	1	0	0		x	x	x	x
1	1	0	1		x	x	x	x
1	1	1	0		x	x	x	x
1	1	1	1		x	x	x	x

LOGIC DIAGRAM:

BCD TO EXCESS-3 CONVERTOR

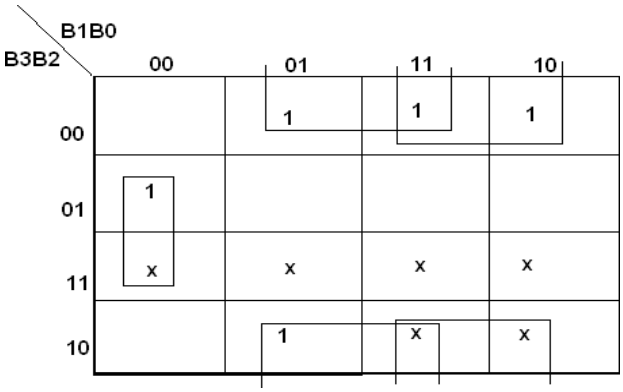


K-Map for E_3:



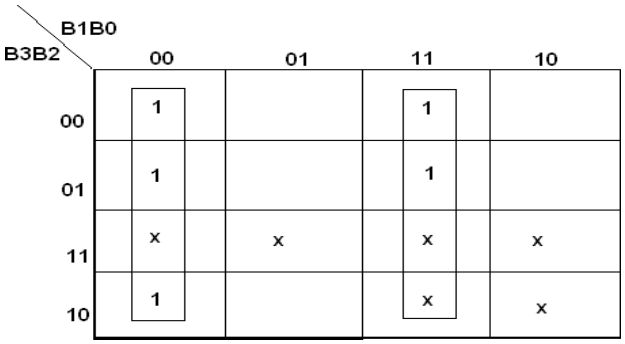
E3 = B3 + B2 (B0 + B1)

K-Map for E_2:



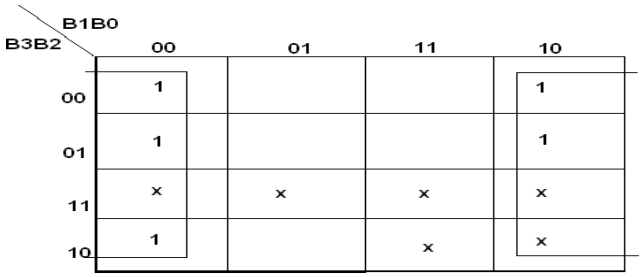
$E2 = B2 \oplus (B1 + B0)$

K-Map for E_1:



$E1 = B1 \bar{\oplus} B0$

K-Map for E_0:



$$E0 = \overline{B0}$$

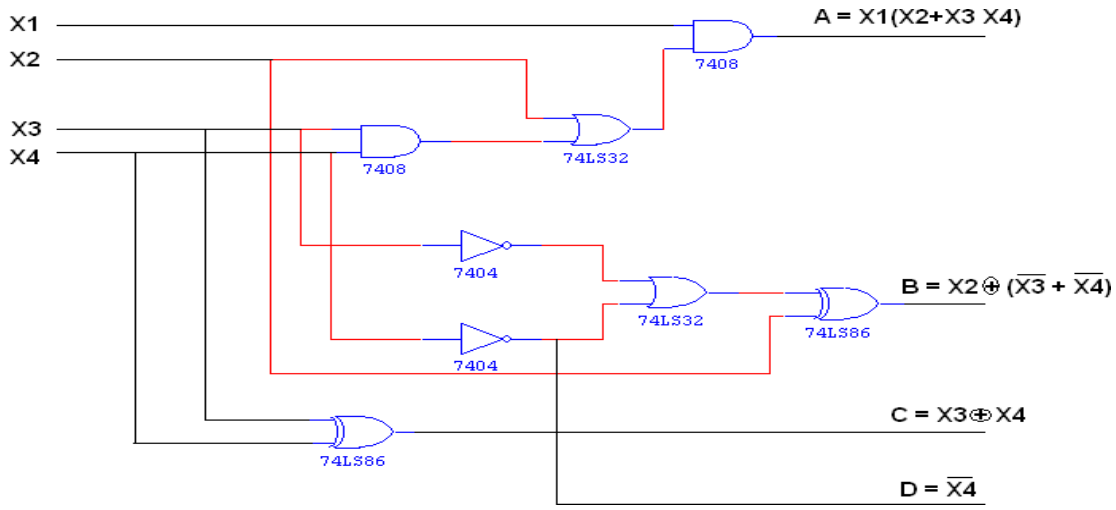
Excess – 3 to BCD Code Converter:

TRUTH TABLE:

Excess - 3 input					BCD output			
X4	**X3**	**X2**	**X1**		**A**	**B**	**C**	**D**
0	**0**	**0**	**0**		**x**	**x**	**x**	**x**
0	**0**	**0**	**1**		**x**	**x**	**x**	**x**
0	**0**	**1**	**0**		**x**	**x**	**x**	**x**
0	**0**	**1**	**1**		**0**	**0**	**0**	**0**
0	**1**	**0**	**0**		**0**	**0**	**0**	**1**
0	**1**	**0**	**1**		**0**	**0**	**1**	**0**
0	**1**	**1**	**0**		**0**	**0**	**1**	**1**
0	**1**	**1**	**1**		**0**	**1**	**0**	**0**
1	**0**	**0**	**0**		**0**	**1**	**0**	**1**
1	**0**	**0**	**1**		**0**	**1**	**1**	**0**
1	**0**	**1**	**0**		**0**	**1**	**1**	**1**
1	**0**	**1**	**1**		**1**	**0**	**0**	**0**
1	**1**	**0**	**0**		**1**	**0**	**0**	**1**
1	**1**	**0**	**1**		**x**	**x**	**x**	**x**
1	**1**	**1**	**0**		**x**	**x**	**x**	**x**
1	**1**	**1**	**1**		**x**	**x**	**x**	**x**

LOGIC DIAGRAM:

EXCESS-3 TO BCD CONVERTOR



K-Map for A:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	0	0	0
11	1	X	X	X
10	0	0	1	0

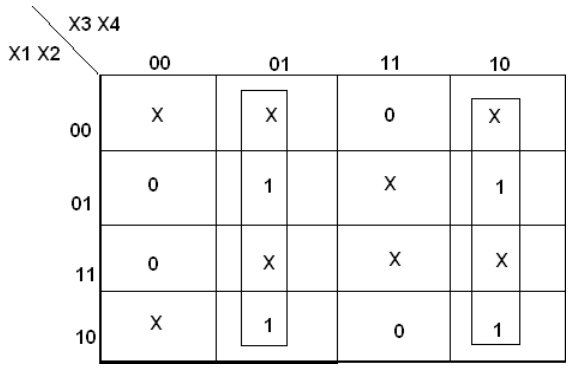
$$A = X1\ X2 + X3\ X4\ X1$$

K-Map for B:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	0	1	0
11	0	X	X	X
10	1	1	0	1

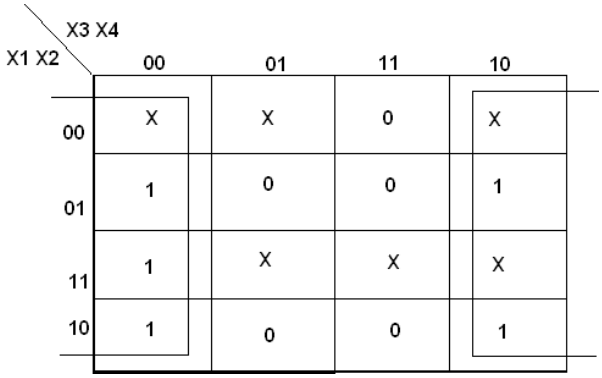
$$B = X2 \oplus (\overline{X3} + \overline{X4})$$

K-Map for C:



$$C = X3 \oplus X4$$

K-Map for D:



$$D = \overline{X4}$$

RESULT:

Thus the code convertors circuits were designed using logic gates and their truth table were verified.

VIVA QUESTIONS

1. Differentiate BCD & Excess 3 code
2. What is BCD?
3. What is the base for Binary?
4. How will you convert Binary to Gray?
5. What is Gray Code?
6. Give the conversion process of Excess 3 to BCD
7. What are the logic gates used in converting Gray code to Binary?
8. What is the need of code converters?
9. How will you design Code converters?
10. What is the difference between Positive and Negative logic?

Expt.No.:5	DESIGN AND IMPLEMENTATION OF MULTIPLEXER AND DE-MULTIPLEXER
Date:	

AIM:

To design and implement 4-to-1 multiplexer and 1-to-4 demultiplexer using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	3 I/P AND GATE	IC 7411	2
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
2.	IC TRAINER KIT	-	1
3.	PATCH CORDS	-	32

THEORY:

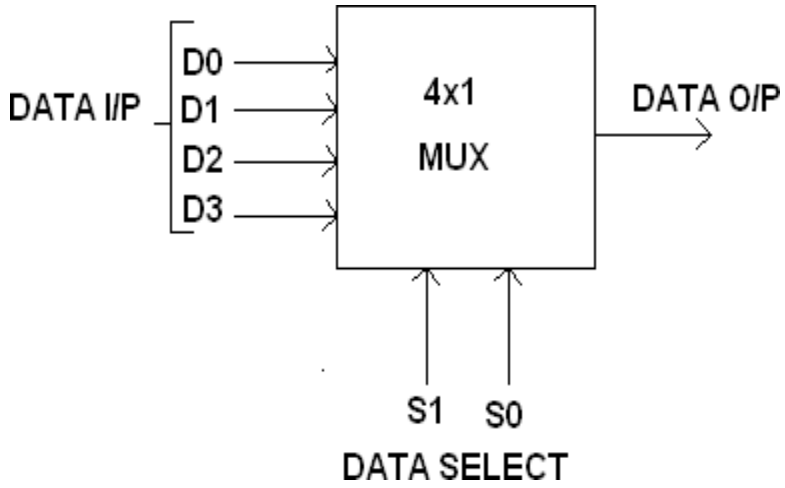
MULTIPLEXER:

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

DEMULTIPLEXER:

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

BLOCK DIAGRAM FOR 4:1 MULTIPLEXER:

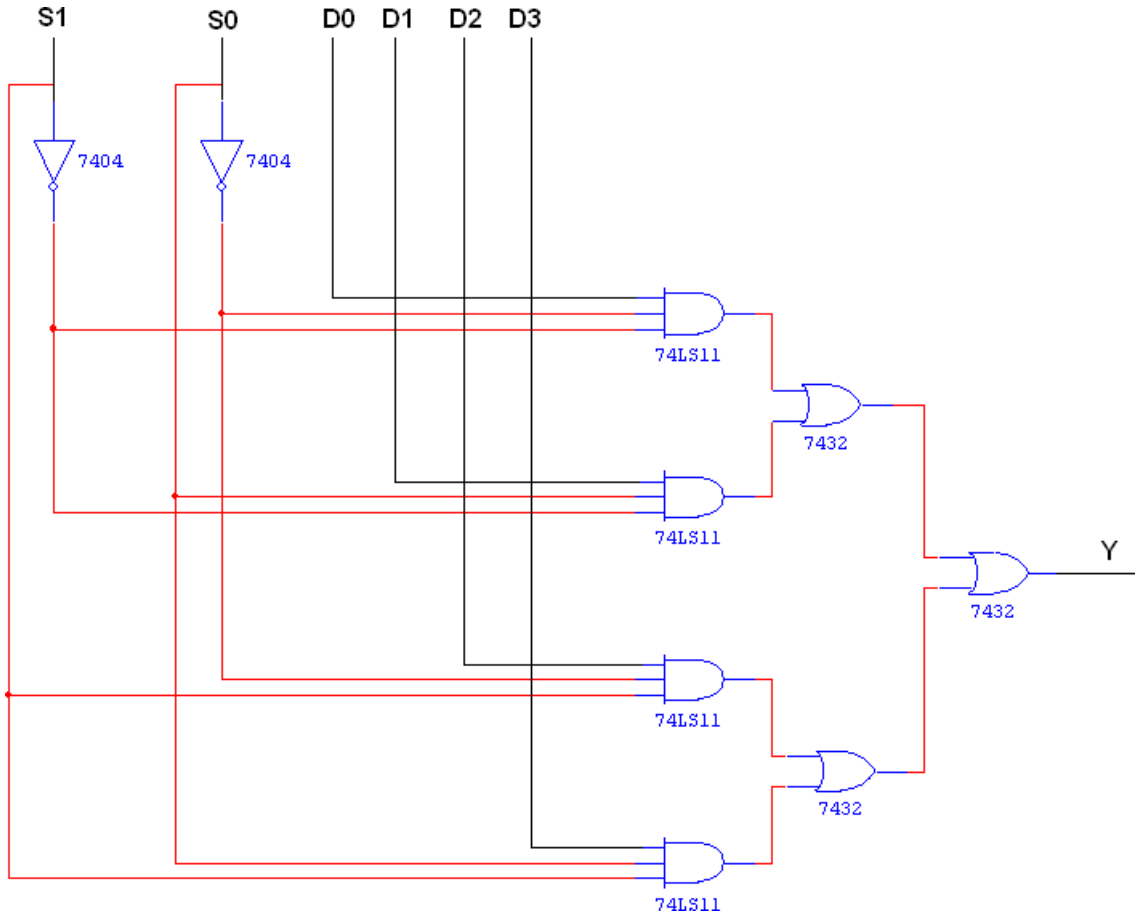


FUNCTION TABLE:

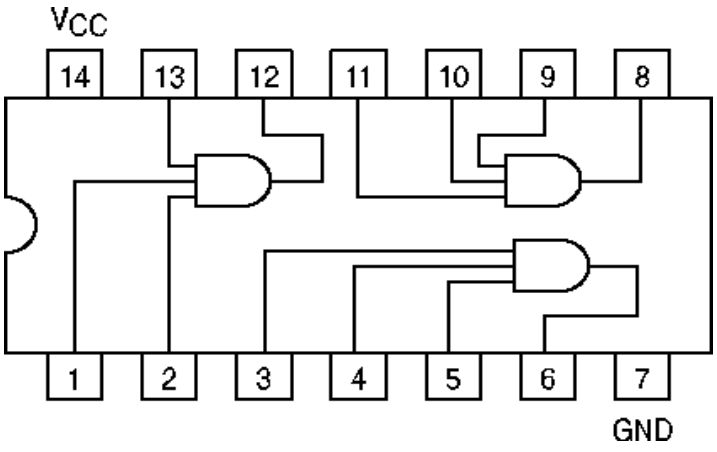
S1	S0	INPUTS Y
0	0	D0 → D0 S1' S0'
0	1	D1 → D1 S1' S0
1	0	D2 → D2 S1 S0'
1	1	D3 → D3 S1 S0

Y = D0 S1' S0' + D1 S1' S0 + D2 S1 S0' + D3 S1 S0

CIRCUIT DIAGRAM FOR MULTIPLEXER:



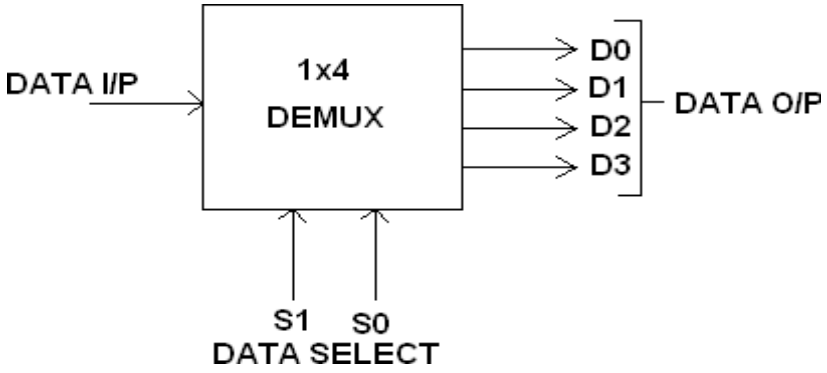
PIN DIAGRAM OF IC7411:



TRUTH TABLE:

S1	S0	Y = OUTPUT
0	0	D0
0	1	D1
1	0	D2
1	1	D3

BLOCK DIAGRAM FOR 1:4 DEMULTIPLEXER:

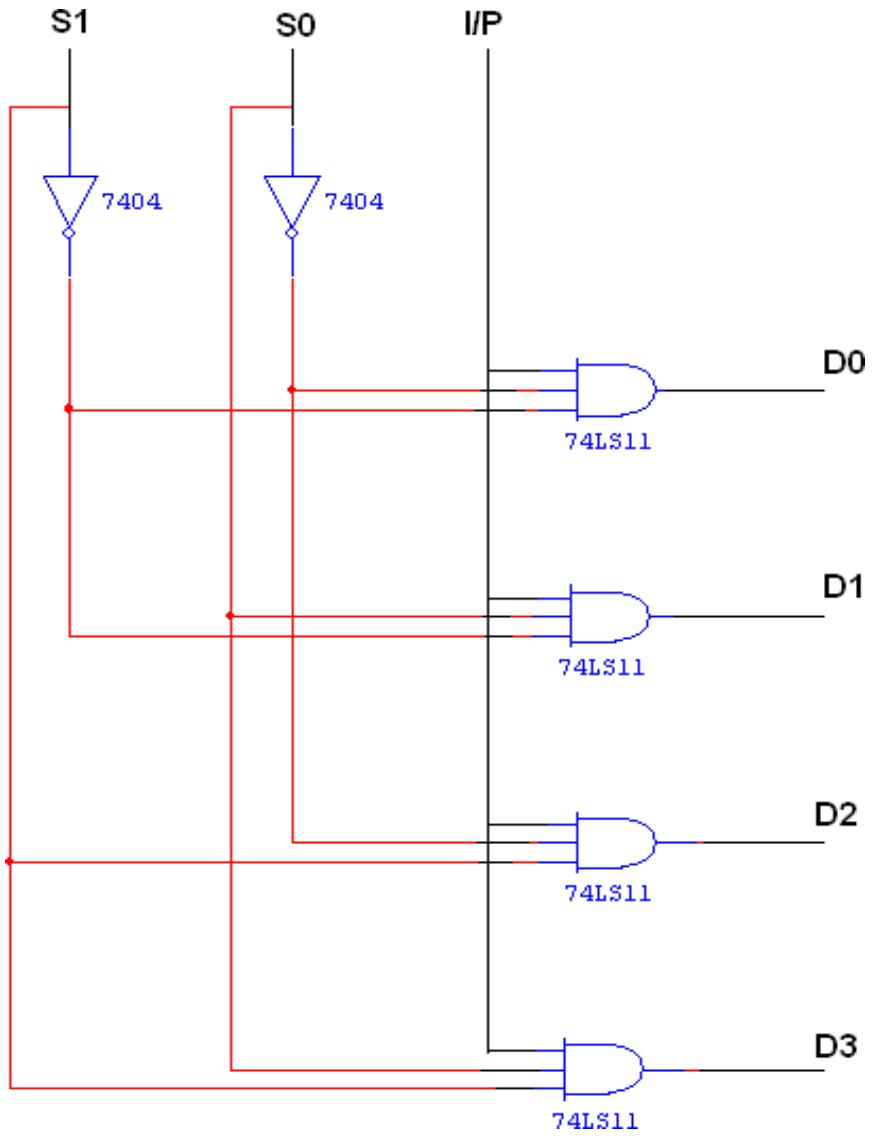


FUNCTION TABLE:

S1	S0	INPUT
0	0	X → D0 = X S1' S0'
0	1	X → D1 = X S1' S0
1	0	X → D2 = X S1 S0'
1	1	X → D3 = X S1 S0

Y = X S1' S0' + X S1' S0 + X S1 S0' + X S1 S0

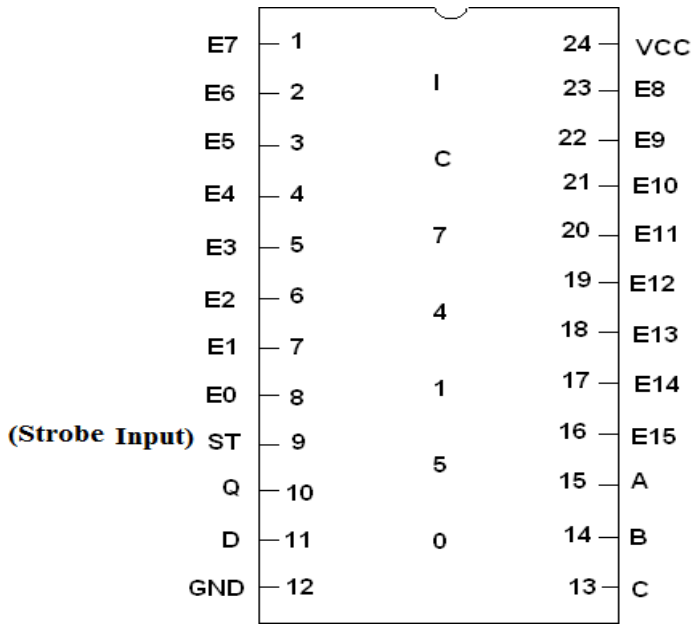
LOGIC DIAGRAM FOR DEMULTIPLEXER:



TRUTH TABLE:

INPUT			OUTPUT			
S1	**S0**	**I/P**	**D0**	**D1**	**D2**	**D3**
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

PIN DIAGRAM FOR IC 74150: 16-to-1 Multiplexer



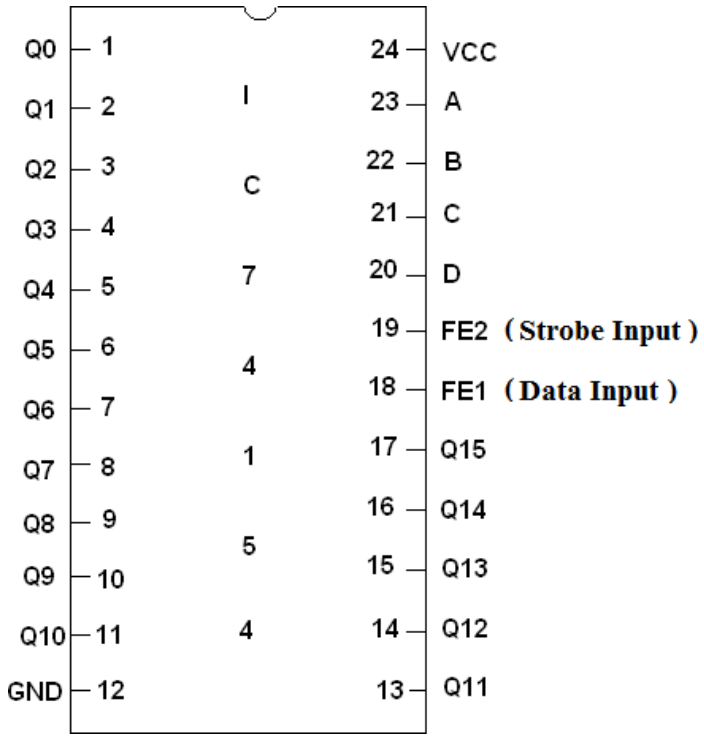
Note : Strobe is an Active Low Enable Input

E - Data Input

A, B, C, D - Select Input

Q - Data output

PIN DIAGRAM FOR IC 74154: 1-to-16 Demultiplexer



PROCEDURE:

(i) Connections are given as per circuit diagram.

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

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

RESULT:

Thus the multiplexer and de-multiplexer circuit was designed using logic gates and their truth table was verified.

VIVA QUESTIONS

1. What is Mux?
2. What is the use of select lines?
3. What is a Demultiplexer?
4. Differentiate 2:1, 4:1 Multiplexer
5. What is called as Distributor? Why?

6. What is the use of IC 74150?
7. Why Mux is called as selector?
8. What is the use of IC 74154?
9. Will you design 8:1 Multiplexor using logic gates?
10. Why 74154 is used rather than logic gates?

Expt.No.:6	DESIGN AND IMPLEMENTATION OF ENCODER AND DECODER
Date:	

AIM:

To design and implement encoder and decoder using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	3 I/P NAND GATE	IC 7411	2
2.	OR GATE	IC 7432	3
3.	NOT GATE	IC 7404	1
2.	IC TRAINER KIT	-	1
3.	PATCH CORDS	-	27

THEORY:

ENCODER

An encoder is a digital circuit that performs inverse operation of a decoder. An encoder has 2^n input lines and n output lines. In encoder the output lines generates the binary code corresponding to the input value. In octal to binary encoder it has eight inputs, one for each octal digit and three output that generate the corresponding binary code. In encoder it is assumed that only one input has a value of one at any given time otherwise the circuit is meaningless. It has an ambiguila that when all inputs are zero the outputs are zero. The zero outputs can also be generated when D0 = 1.

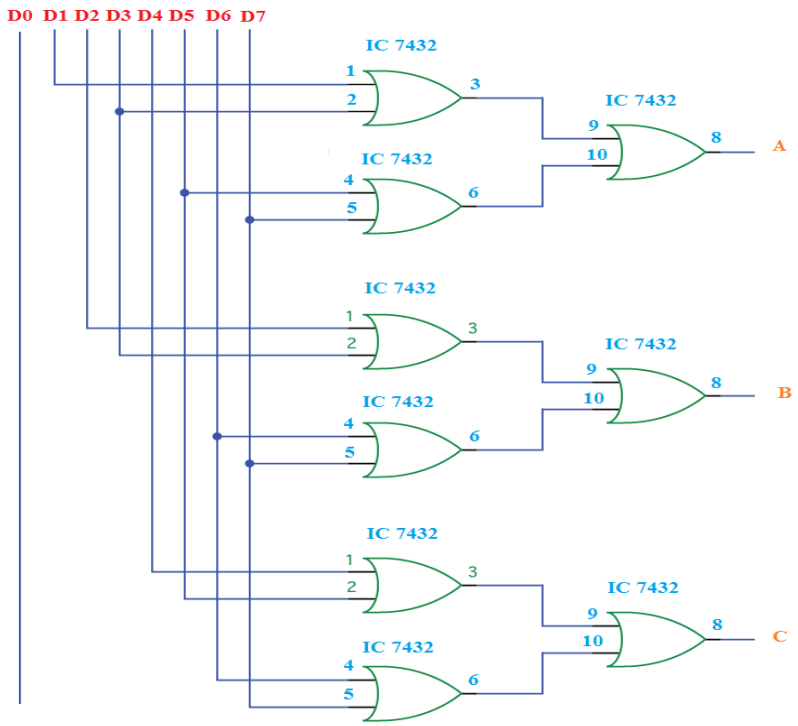
DECODER:

A decoder is a multiple input multiple output logic circuit which converts coded input into coded output where input and output codes are different. The input code generally has fewer bits than the output code. Each input code word produces a different output code word i.e., there is one to one mapping can be expressed in truth table. In the block diagram of decoder circuit the encoded information is present as n input producing 2^n possible outputs. 2^n output values are from 0 through out $2^n - 1$.

TRUTH TABLE: OCTAL TO BINARY ENCODER

INPUT (OCTAL)								OUTPUT		
D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**A**	**B**	**C**
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

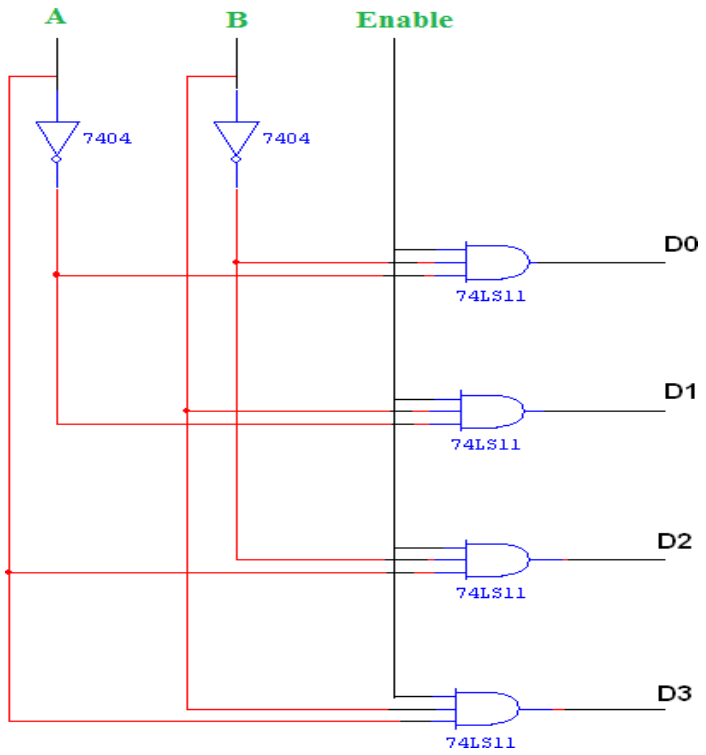
LOGIC DIAGRAM FOR ENCODER:



TRUTH TABLE: 2 TO 4 DECODER WITH ENABLE

INPUT			OUTPUT			
EN	A	B	D0	D1	D2	D3
0	X	X	0	0	0	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	0	1

LOGIC DIAGRAM: 2 TO 4 DECODER



PROCEDURE:

(i) Connections are given as per circuit diagram.

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

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

RESULT:

Thus the encoder and decoder circuit was designed using logic gates and their truth table was verified.

VIVA QUESTIONS

1. What is decoder?
2. What do you mean by encoder?
3. Define priority encoder.
4. What is the difference between encoder and demultiplexer?
5. List out the application of decoders?
6. What is cascade decoder?
7. Distinguish between encoder and decoder?
8. What is the use of input variable in cascade decoder?

Expt.No.:7	DESIGN AND IMPLEMENTATION OF MAGNITUDE COMPARATORS
Date:	

AIM

To design and implement

(i) 2 – bit magnitude comparator using basic gates.

(ii) 8 – bit magnitude comparator using IC 7485.

APPARATUS REQUIRED

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	2
2.	X-OR GATE	IC 7486	1
3.	OR GATE	IC 7432	1
4.	NOT GATE	IC 7404	1
5.	4-BIT MAGNITUDE COMPARATOR	IC 7485	2
6.	IC TRAINER KIT	-	1
7.	PATCH CORDS	-	30

THEORY

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

$A = A_3\ A_2\ A_1\ A_0 B = B_3\ B_2\ B_1\ B_0$

The equality of the two numbers and B is displayed in a combinational circuit designated by the symbol (A=B).

This indicates A greater than B, then inspect the relative magnitude of pairs of significant digits starting from most significant position. A is 0 and that of B is 0.We have A<B, the sequential comparison can be expanded as

$A>B = A_3B_3^1 + X_3 A_2 B_2^1 + X_3 X_2 A_1 B_1^1 + X_3 X_2 X_1 A_0 B_0^1$

$A<B = A_3^1B_3 + X_3A_2^1B_2 + X_3X2A_1^1B_1 + X_3X_2X_1A_0^1B_0$

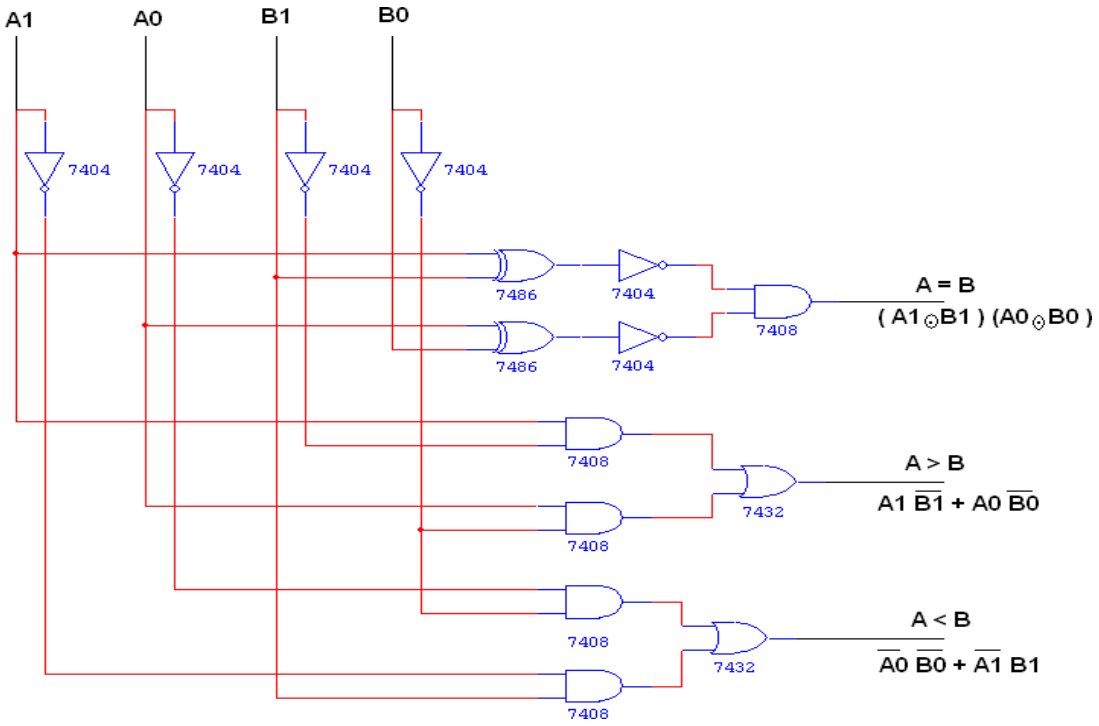
The same circuit can be used to compare the relative magnitude of two BCD digits.

Where, A = B is expanded as,

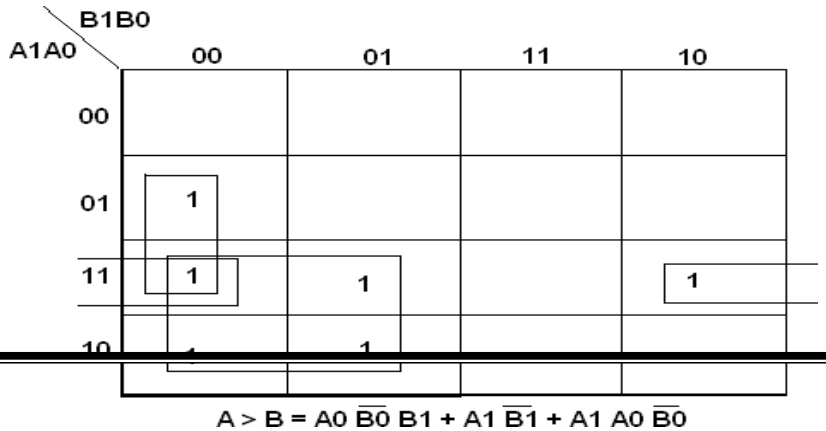
$$A = B = (A_3 + B_3)\,(A_2 + B_2)\,(A_1 + B_1)\qquad (A_0 + B_0)$$

LOGIC DIAGRAM:

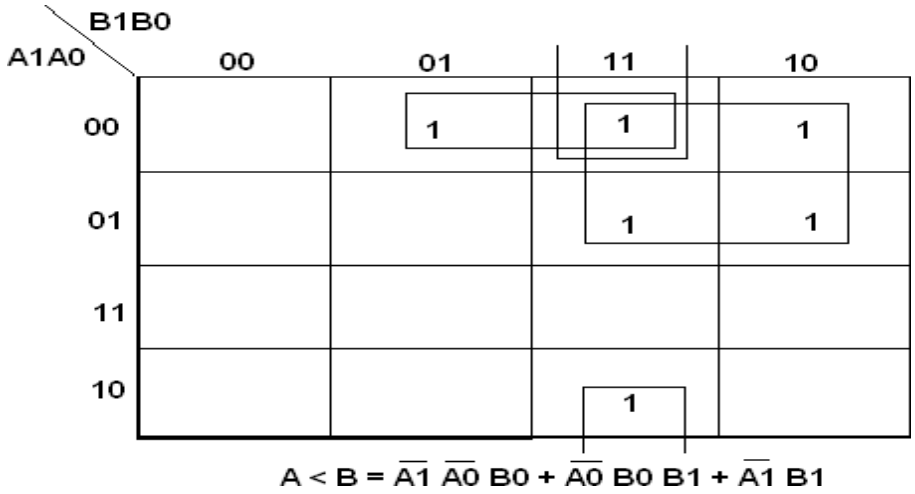
2 BIT MAGNITUDE COMPARATOR

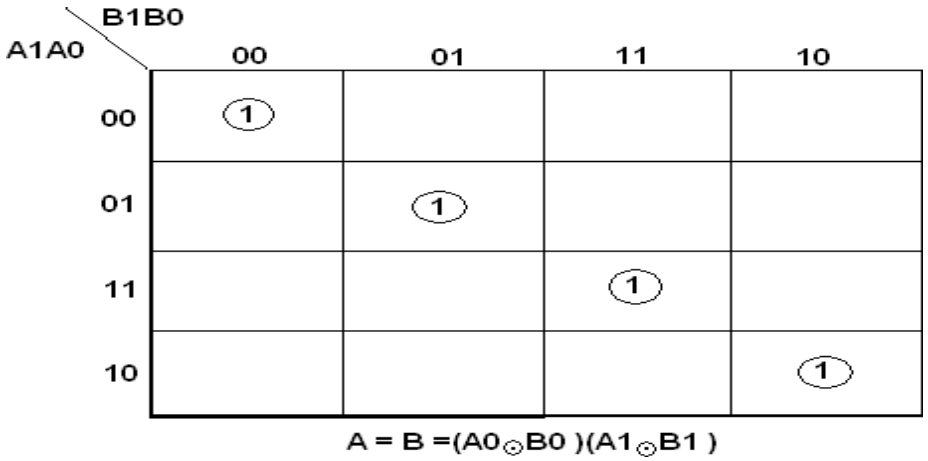


K MAP



A > B = A0 $\overline{B0}$ B1 + A1 $\overline{B1}$ + A1 A0 $\overline{B0}$



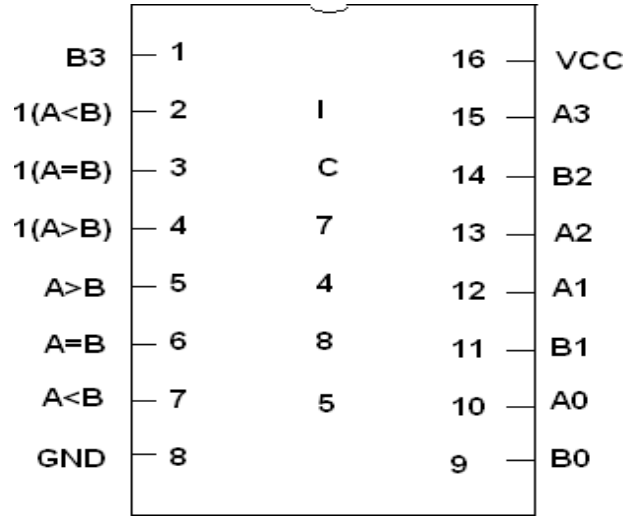


TRUTH TABLE

A1	A0	B1	B0	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

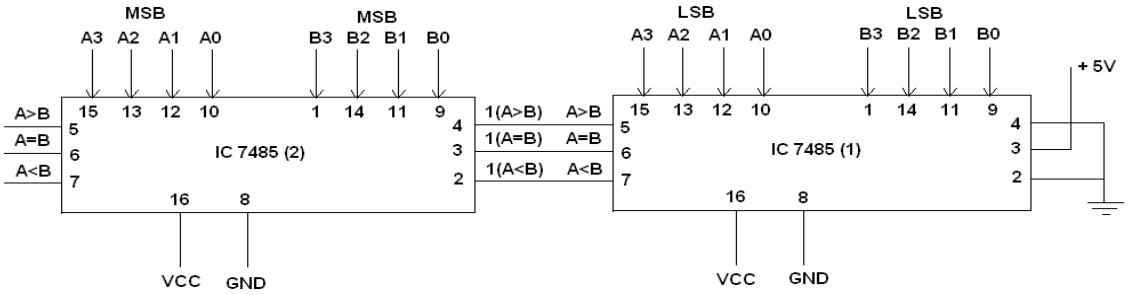
1	**0**	**0**	**1**	**1**	**0**	**0**
1	**0**	**1**	**0**	**0**	**1**	**0**
1	**0**	**1**	**1**	**0**	**0**	**1**
1	**1**	**0**	**0**	**1**	**0**	**0**
1	**1**	**0**	**1**	**1**	**0**	**0**
1	**1**	**1**	**0**	**1**	**0**	**0**
1	**1**	**1**	**1**	**0**	**1**	**0**

PIN DIAGRAM FOR IC 7485:



LOGIC DIAGRAM:

8 BIT MAGNITUDE COMPARATOR



TRUTH TABLE:

A	B	A>B	A=B	A<B
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0	1	0
0 0 0 1 0 0 0 1	0 0 0 0 0 0 0 0	1	0	0
0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 1	0	0	1

PROCEDURE:

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

RESULT:

Thus the magnetitude comparator circuit was designed using logic gates and their truth table was verified.

Viva Question:

1. what is magnitude comparator?
2. what is most significant bit?
3. explain truth table of a comparator?
4. what is equality?
5. Explain magnitude comparator7485 IC
6. what is IC?

Expt.No.:8	CONSTRUCTION AND VERIFICATION OF 4 BIT RIPPLE COUNTER AND MOD-10 / MOD-12 RIPPLE COUNTERS
Date:	

AIM:

To design and verify 4 bit ripple counter mod 10/ mod 12 ripple counter.

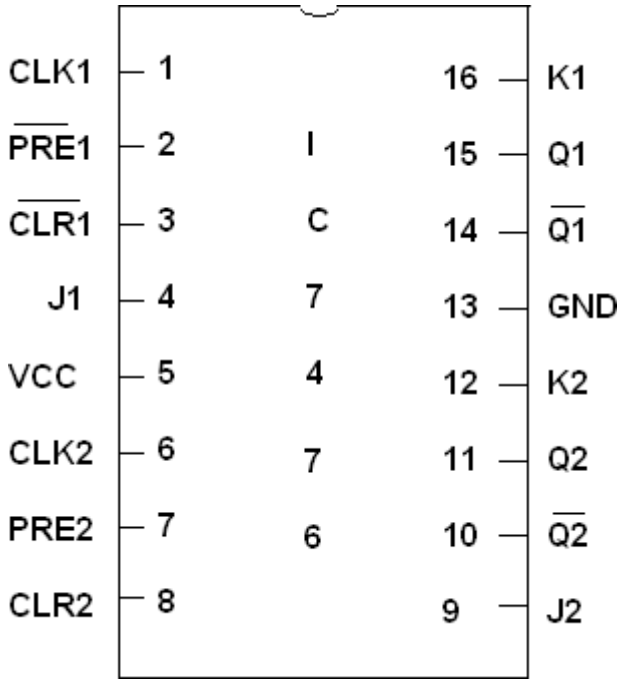
APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	JK FLIP FLOP	IC 7476	2
2.	NAND GATE	IC 7400	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	30

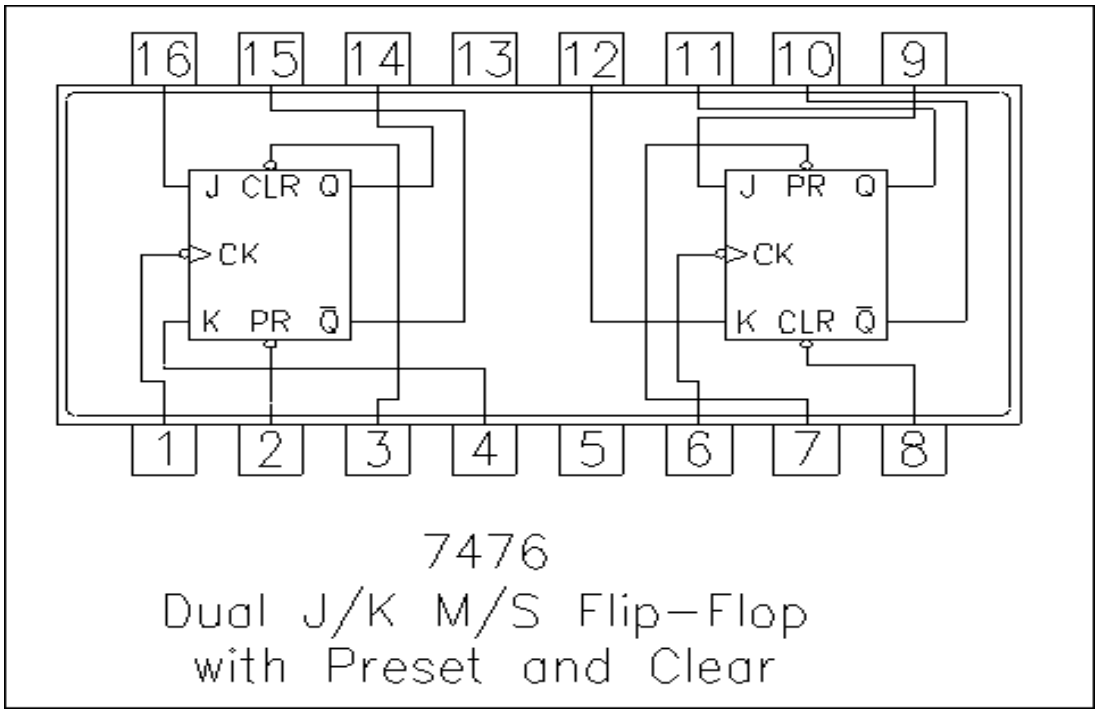
THEORY:

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

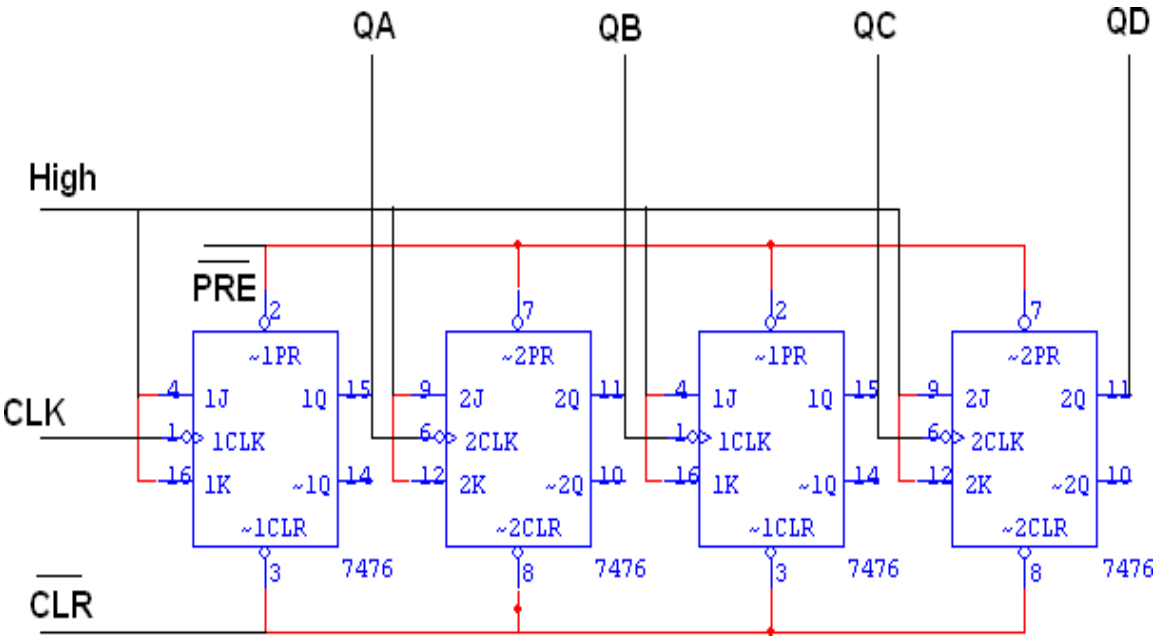
PIN DIAGRAM FOR IC 7476:



Equivalent Diagram:



LOGIC DIAGRAM FOR 4 BIT RIPPLE COUNTER:

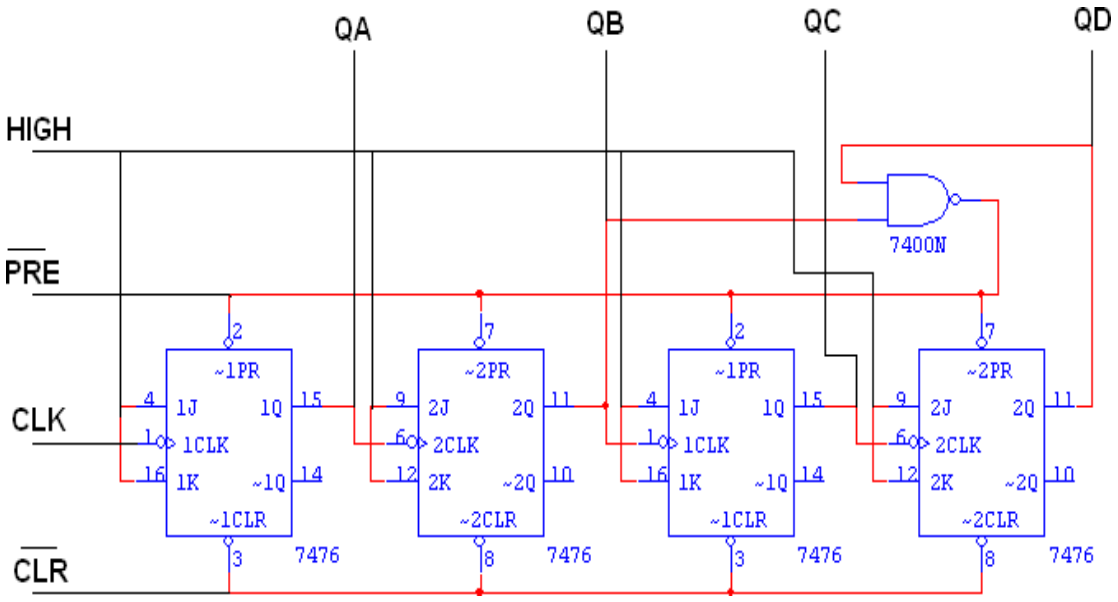


TRUTH TABLE:

CLK	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1

9	**1**	**0**	**0**	**1**
10	**0**	**1**	**0**	**1**
11	**1**	**1**	**0**	**1**
12	**0**	**0**	**1**	**1**
13	**1**	**0**	**1**	**1**
14	**0**	**1**	**1**	**1**
15	**1**	**1**	**1**	**1**

LOGIC DIAGRAM FOR MOD - 10 RIPPLE COUNTER:

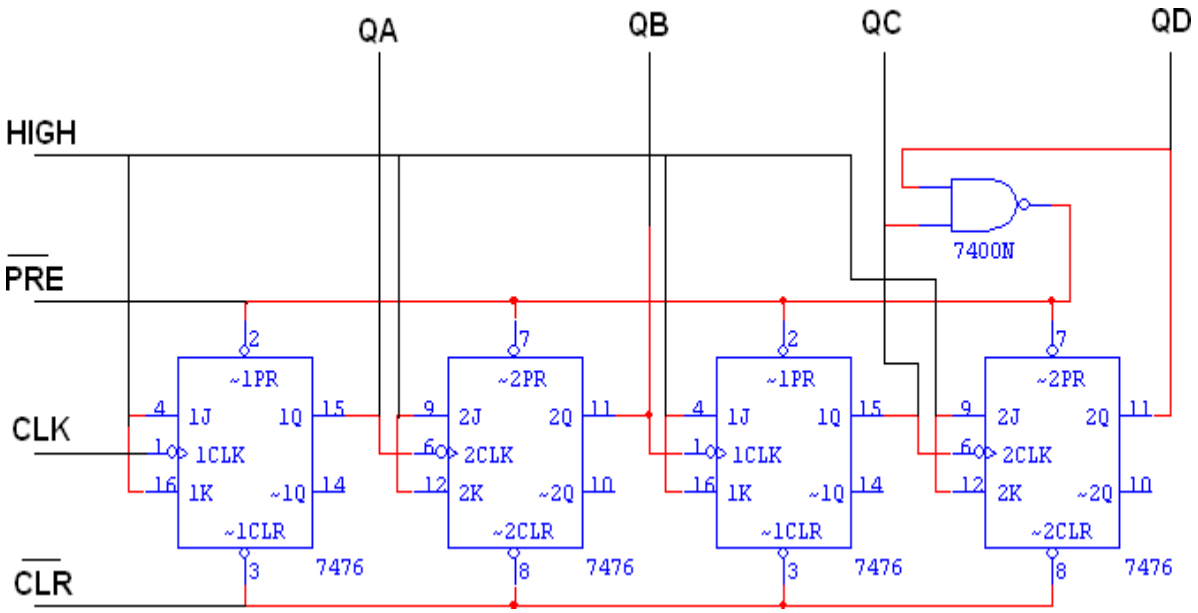


TRUTH TABLE:

CLK	QA	QB	QC	QD
0	**0**	**0**	**0**	**0**
1	**1**	**0**	**0**	**0**
2	**0**	**1**	**0**	**0**

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

LOGIC DIAGRAM FOR MOD - 12 RIPPLE COUNTER:



TRUTH TABLE:

CLK	QA	QB	QC	QD
0	**0**	**0**	**0**	**0**
1	**1**	**0**	**0**	**0**

2	**0**	**1**	**0**	**0**
3	**1**	**1**	**0**	**0**
4	**0**	**0**	**1**	**0**
5	**1**	**0**	**1**	**0**
6	**0**	**1**	**1**	**0**
7	**1**	**1**	**1**	**0**
8	**0**	**0**	**0**	**1**
9	**1**	**0**	**0**	**1**
10	**0**	**1**	**0**	**1**
11	**1**	**1**	**0**	**1**
12	**0**	**0**	**0**	**0**

PROCEDURE:

(i) Connections are given as per circuit diagram.

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

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

RESULT:

Thus the 4 bit ripple counter (Modulus 16), mod -10 ripple counter and mod- 12 ripple counter circuits were designed using JK flip-flops and their output was verified.

Viva Questions

1. What is a counter?
2. What are the types of counter?
3. Which flip-flop is most commonly used for counter applications?
4. Why asynchronous counter is called ripple counter?

5. What is Up/Down counter?
6. What are the applications of Flip-flops?
7. What is the use of a Clock signal?
8. What are the types of Triggering methods in F/Fs

Expt.No.:9	DESIGN AND IMPLEMENTATION OF 3-BIT SYNCHRONOUS UP/DOWN COUNTER
Date:	

AIM:

To design and implement 3 bit synchronous up / down counter using JK flip-flops and logic gates.

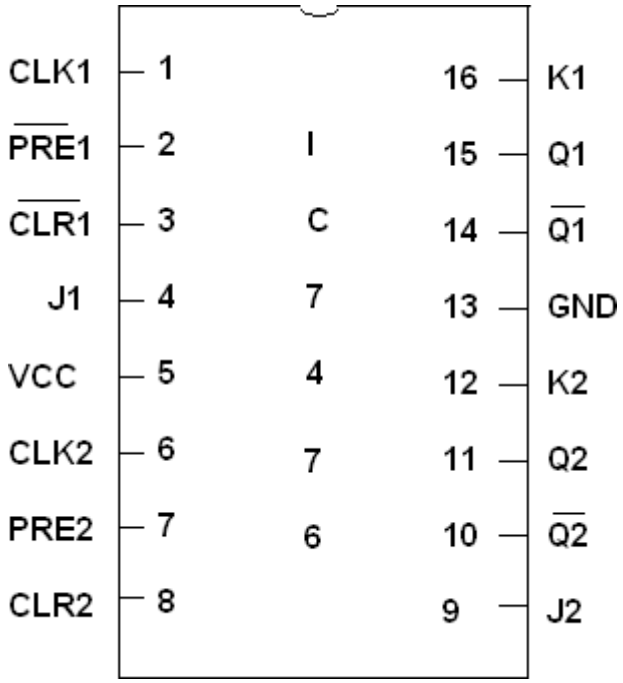
APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	JK FLIP FLOP	IC 7476	2
2.	3 I/P AND GATE	IC 7411	1
3.	OR GATE	IC 7432	1
4.	XOR GATE	IC 7486	1
5.	NOT GATE	IC 7404	1
6.	IC TRAINER KIT	-	1
7.	PATCH CORDS	-	35

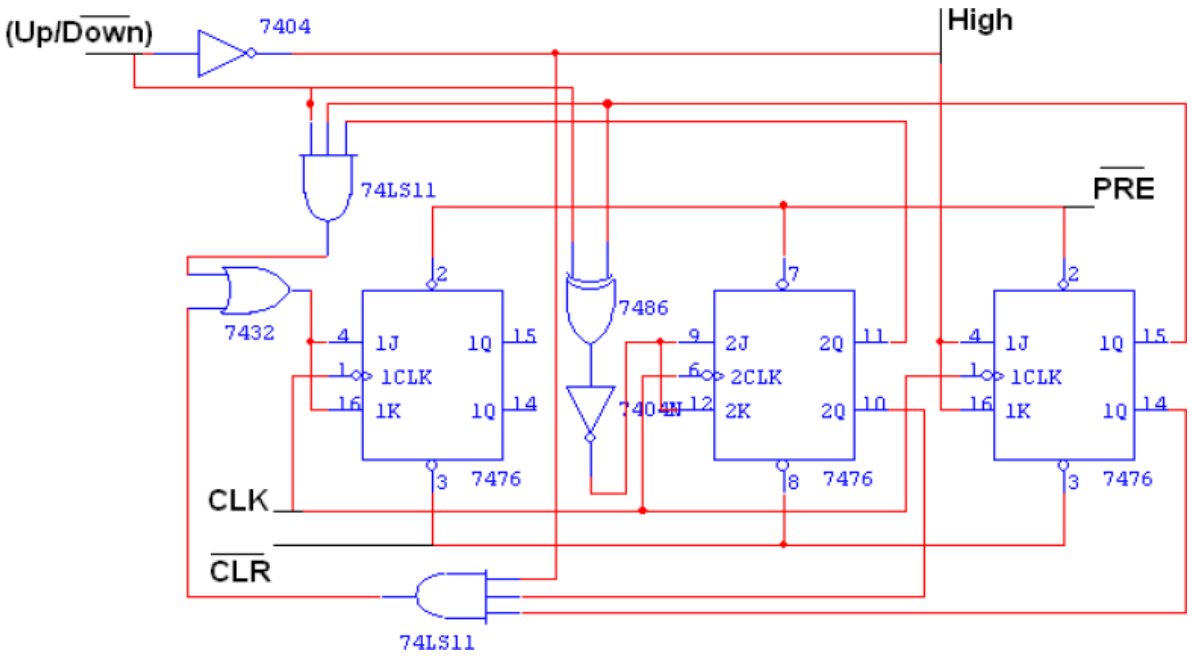
THEORY:

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. In synchronous counters, the clock inputs of all the flip-flops are connected together and are triggered by the input pulses. Thus, all the flip-flops change state simultaneously (in parallel). An up/down counter is one that is capable of progressing in increasing order or decreasing order through a certain sequence. An up/down counter is also called bidirectional counter. Usually up/down operation of the counter is controlled by up/down signal. When this signal is high counter goes through up sequence and when up/down signal is low counter follows reverse sequence.

PIN DIAGRAM FOR IC 7476:



LOGIC DIAGRAM: 3 – BIT SYNCHRONOUS UP / DOWN COUNTER



TRUTH TABLE:

	Up Counter			Down Counter		
CLK	**Q_A**	**Q_B**	**Q_C**	**Q_A**	**Q_B**	**Q_C**
0	0	0	0	1	1	1
1	0	0	1	1	1	0
2	0	1	0	1	0	1
3	0	1	1	1	0	0
4	1	0	0	0	1	1
5	1	0	1	0	1	0
6	1	1	0	0	0	1
7	1	1	1	0	0	0
8	0	0	0	1	1	1

PROCEDURE:

(i) Connections are given as per circuit diagram.

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

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

RESULT:

Thus Up counter and Down counter was designed successfully using IC 7476 and its truth table was verified successfully.

VIVA QUESTIONS

1. Compare Synchronous, Asynchronous counters?
2. What is State diagram?
3. What is the disadvantage of Synchronous counter?
4. How will you design a synchronous counter?
5. What are the types of Flip-flops

Expt.No.:10	DESIGN AND IMPLEMENTATION OF SHIFT REGISTERS SISO, SIPO
Date:	

AIM:

To design and implement the following types of shift registers using D flip-flops.

(i) Serial in serial out
(ii) Serial in parallel out

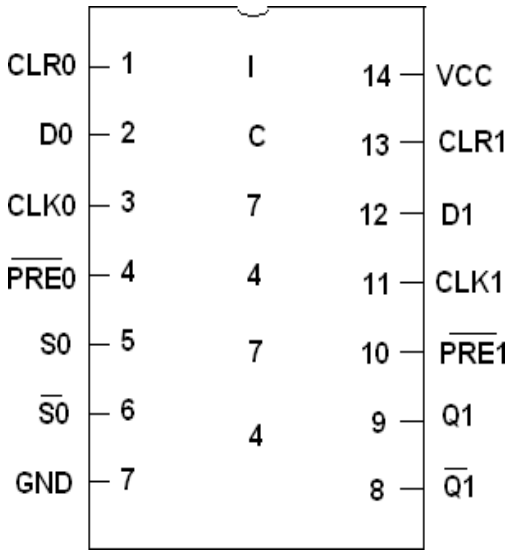
APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	D FLIP FLOP	IC 7474	2
2.	OR GATE	IC 7432	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	35

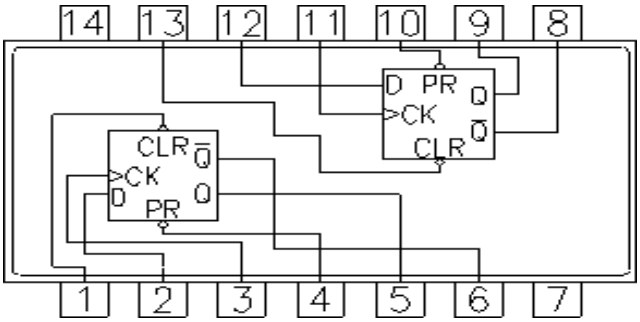
THEORY:

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

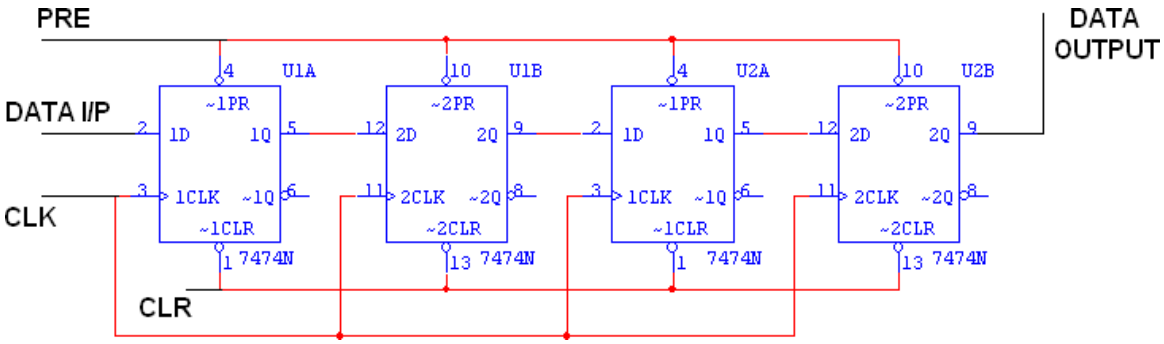
PIN DIAGRAM: IC7474 Dual D Flip-flop with Preset and Clear



EQUIVALENT DIAGRAM: IC7474 Dual D Flip-flop



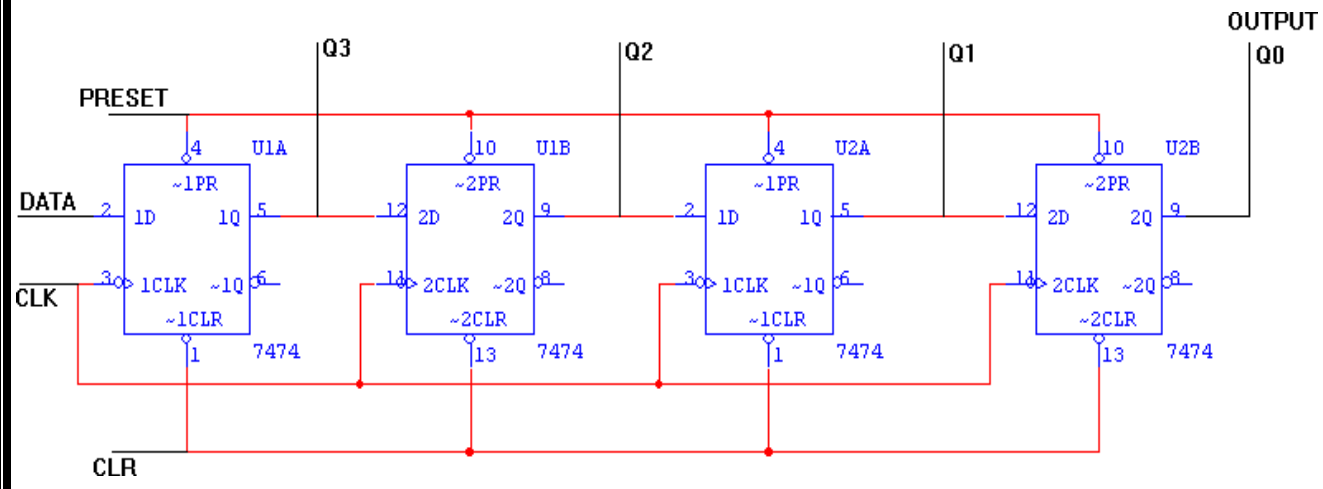
LOGIC DIAGRAM: SERIAL IN SERIAL OUT



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: 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**

PROCEDURE:

(i) Connections are given as per circuit diagram.

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

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

RESULT:

Thus the various shift registers were designed successfully using flip-flops and their truth tables were verified successfully.

VIVA QUESTIONS

1. What is a Register?
2. What is a Shift Register?
3. What is the basic device used in a Shift register?
4. What is the of Shift registers?
5. Give one application of shift register
6. What is SISO shift register? What is the IC used for it?
7. What is a Ring Counter?
8. What is a Bi-directional Shift register?
9. What is a PISO shift register?
10. Which is faster?

Expt.No.:11	**DESIGN AND IMPLEMENTATION OF SHIFT REGISTERS PISO, PIPO**
Date:	

AIM:

To design and implement the following types of shift registers using D flip-flops.

i. Parallel in serial out
ii. Parallel in parallel out

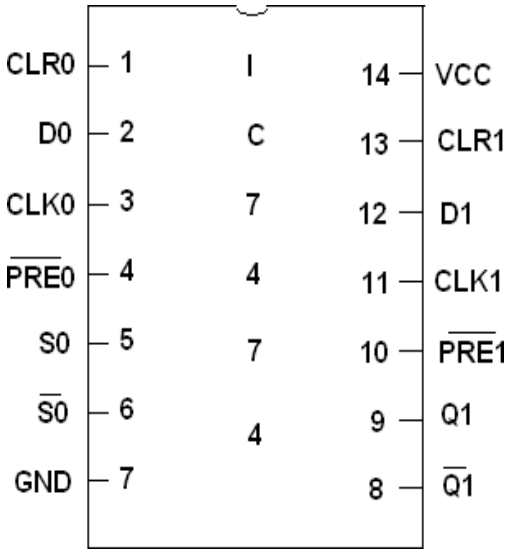
APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	D FLIP FLOP	IC 7474	2
2.	OR GATE	IC 7432	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	35

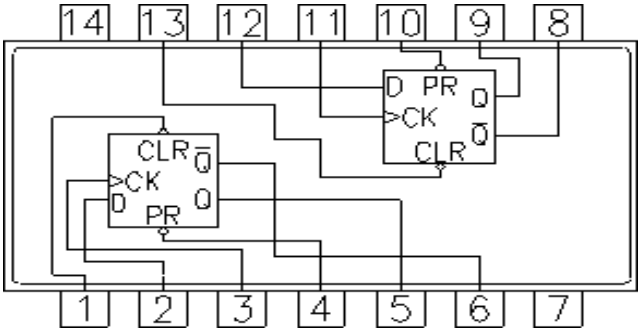
THEORY:

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

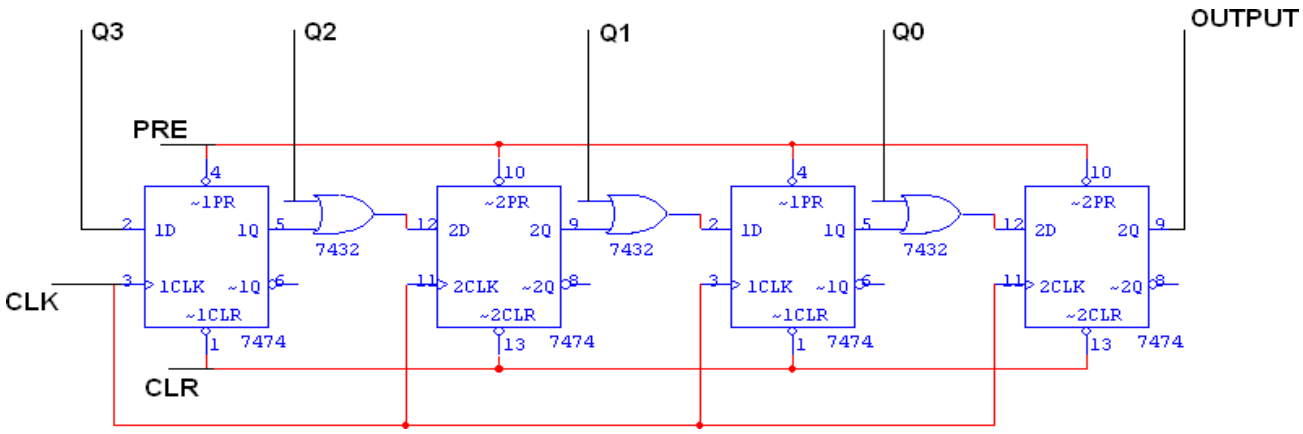
PIN DIAGRAM: IC7474 Dual D Flip-flop with Preset and Clear



EQUIVALENT DIAGRAM: IC7474 Dual D Flip-flop



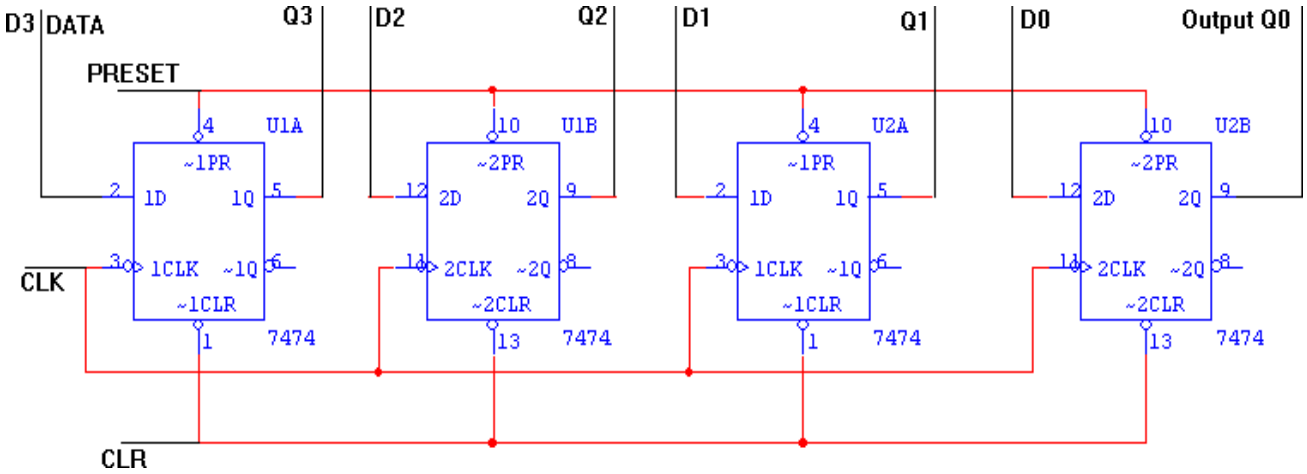
LOGIC DIAGRAM: PARALLEL IN SERIAL OUT



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

LOGIC DIAGRAM: 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. Logical inputs are given as per circuit diagram.

iii. Observe the output and verify the truth table.

RESULT:

Thus the various shift registers were designed successfully using flip-flops and their truth tables were verified successfully.

VIVA QUESTIONS

1. What is a Register?
2. What is a Shift Register?
3. What is the basic device used in a Shift register?
4. What is the of Shift registers?
5. Give one application of shift register
6. What is SISO shift register? What is the IC used for it?
7. What is a Ring Counter?
8. What is a Bi-directional Shift register?
9. What is a PISO shift register?
10. Which is faster?