BHARTIYA INSTITUTE OF ENGINEERING & TECHNOLOGY, SIKAR

LAB MANUAL

IV SEMESTER

DIGITAL ELECTRONICS LAB

Subject Code: 4EE4-23

Prepared By:

Dr. SUMAN NEHRA
Head of Department
(EE Dept.)

Department of Electrical Engineering

BIET, Sikar

Website: www.bietsikar.ac.in

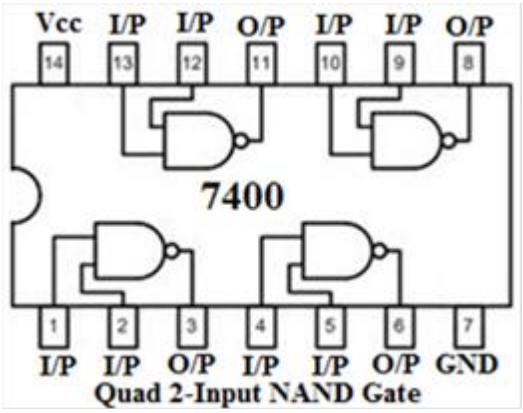
CONTENTS

S.No.	Experiments	Page No.
1.	To verify the truth tables of basic logic gates: AND, OR, NOR, NAND, NOR. Also to verify the truth table of Ex-OR, Ex-NOR (For 2, 3, & 4 inputs using gates with 2, 3, & 4 inputs).	
2.	To verify the truth table of OR, AND, NOR, Ex-OR, Ex-NOR realized using NAND & NOR gates.	
3.	To realize an SOP and POS expression.	
4.	To realize Half adder/ Subtractor & Full Adder/ Subtractor using NAND & NOR gates and to verify their truth tables.	
5.	To design and implement 4-bit adder and subtractor using IC 7483.	
6.	To verify the truth table of 4-to-1 multiplexer and 1-to-4 demultiplexer. Realize the multiplexer using basic gates only. Also to construct and 8- to-1 multiplexer and 1-to-8 demultiplexer using blocks of 4-to-1 multiplexer and 1-to-4 demultiplexer.	
7.	To set up and test a 7-segment static display system to display numbers 0 to 9.	
8.	Using basic logic gates, realize the R-S, J-K and D-flip flops with and without clock signal and verify their truth table.	
9.	To design and implement the 2-bit up counter using D Flip-Flop.	
10.	Perform input/output operations on parallel in/Parallel out and Serial in/Serial out registers using clock. Also exercise loading only one of multiple values into the register using multiplexer.	

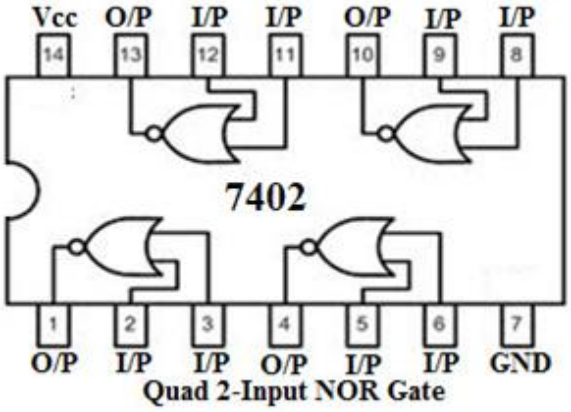
Experiment No.1

AIM: - To verify the truth tables of basic logic gates: AND, OR, NOR, NAND, NOR. Also to verify the truth table of Ex-OR, Ex-NOR

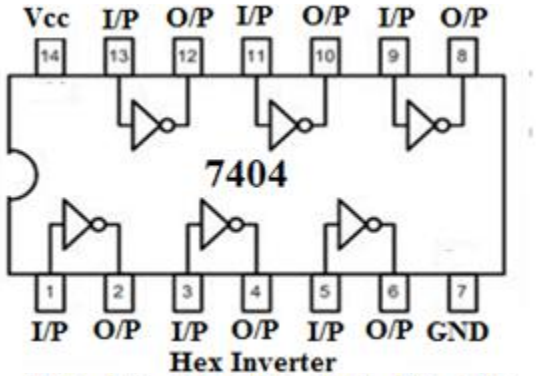
APPARATUS REQUIRED: - IC 7400, IC 7402, IC 7404, IC 7408, IC 7432, IC 7486.



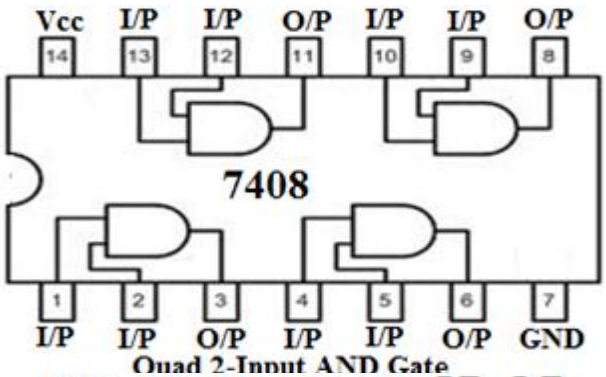
Quad 2-Input NAND Gate



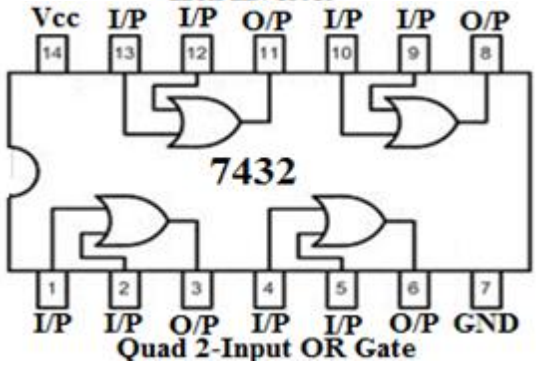
Quad 2-Input NOR Gate



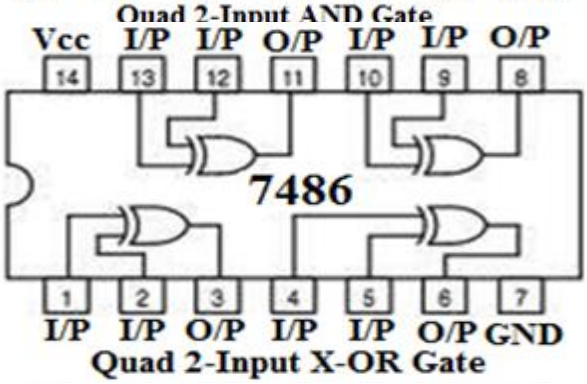
Hex Inverter



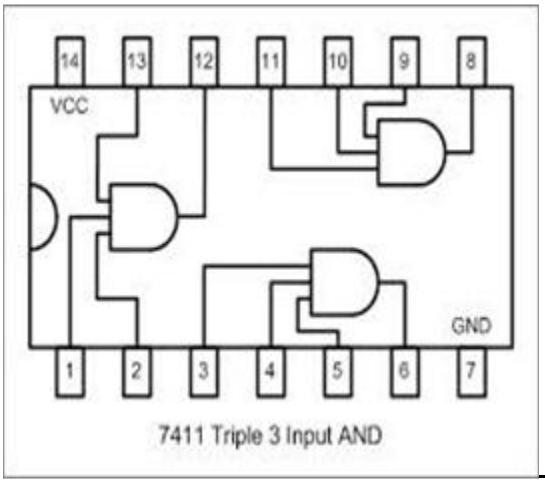
Quad 2-Input AND Gate



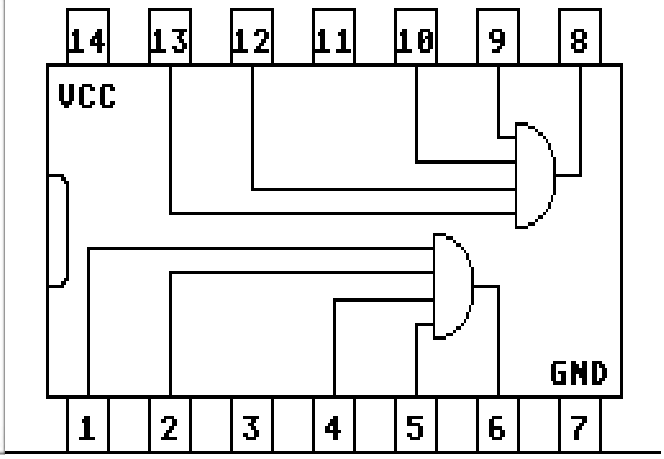
Quad 2-Input OR Gate

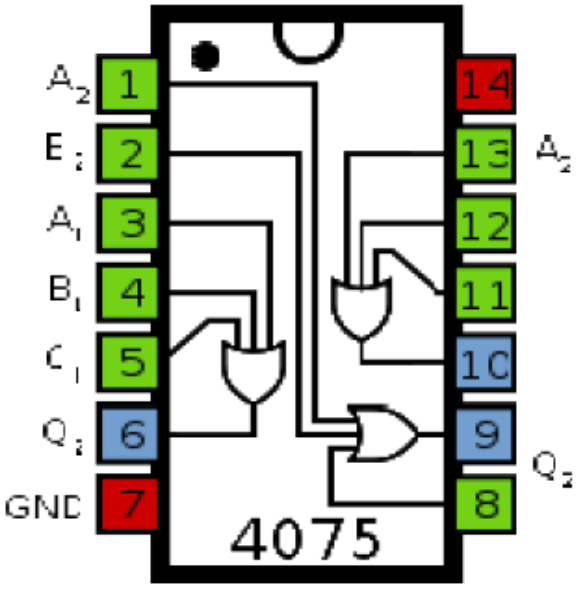


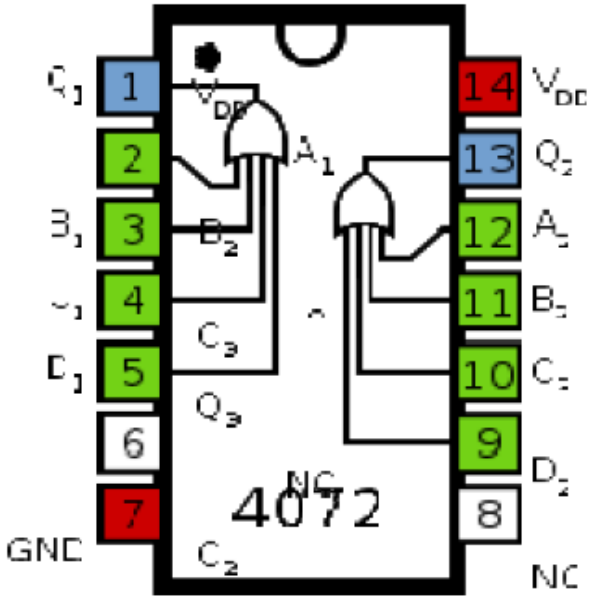
Quad 2-Input X-OR Gate



7421 Dual 4-Input AND Gate







THEORY:- A logic gate is an electronic circuit/device which makes the logical decisions. Digital systems are said to be constructed by using three basic logic gates (AND, OR & NOT)

Universal Gates:- A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

NAND Gate

Logic Equation: Y= $\overline{\quad}$A.B

TTL IC: 7400(Quad 2-input NAND gates)

NAND Gate

Logic symbol

The NAND gate represents the complement of the AND operation. Its name is an abbreviation of NOT AND. The graphic symbol for the NAND gate consists of an AND symbol with a bubble on the output, denoting that a complement operation is performed on the output of the AND gate.

TRUTH TABLE:

Input A	Input B	Output Q
0	0	1
0	1	1
1	0	1
1	1	0

The output of NAND gate is high when at least one is low.

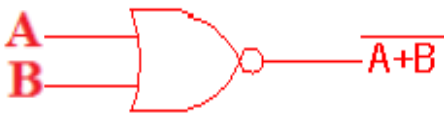
OR

The output of NAND gate is low when all inputs are high.

NOR Gate:-

Logic Equation: Y= $\overline{A+B}$

TTL IC: 7402(Quad 2-input NOR gates)



Logic symbol

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The symbol is an OR gate with a small circle on the output. The small circle represents inversion

TRUTH TABLE:

Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	0

The output of NOR gate is low when at least one input is high.
OR
The output of NOR gate is high when all inputs are low.

NOT Gate:-

Logic Equation: Y= $\overline{A}$

TTL IC: 7404(Hex Inverter)

Logic symbol

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top, as shown at the outputs.

TRUTH TABLE:

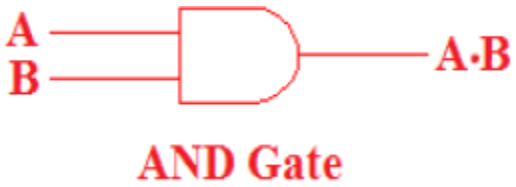
Input A	Output Q
0	1
1	0

The output Q is high when the input A is low.
OR
The output Q is low when the input A is high.

AND Gate: -

Logic Equation:Y=A.B

TTL IC: 7408(Quad 2-input AND gates)



Logic symbol

OR Gate: -

Logic Equation:Y=A+B

TTL IC: 7432(Quad 2-input OR gates)

Logic symbol

An OR gate can have two or more inputs, A plus (+) is used to describe the OR operations.

TRUTH TABLE:

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

The output Q is high if at least one input is high

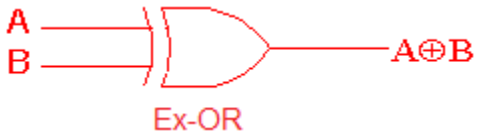
OR

The output Q is low if all inputs are low

Ex-OR Gate: -

Logic Equation Y= $A \oplus B = \bar{A}.B + A.\bar{B}$

TTL IC: 7486(Quad 2-input Ex-OR gates)



Logic symbol

An encircled plus sign (⊕) is used to show the Ex-OR operation.

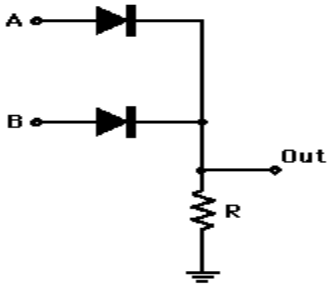
TRUTH TABLE:

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	0

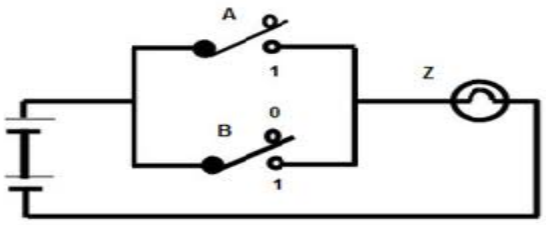
The output Q is high if algebraic sum of 1(high) is odd

OR

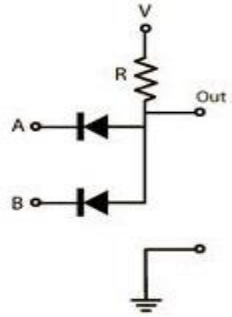
The output Q is low if algebraic sum of 1(high) is even.



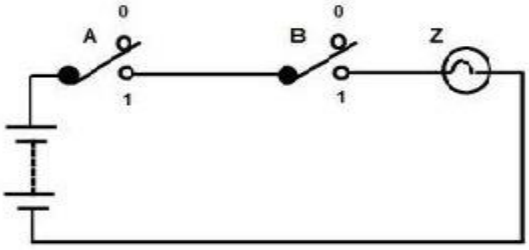
2 Input OR Gate Using Diode



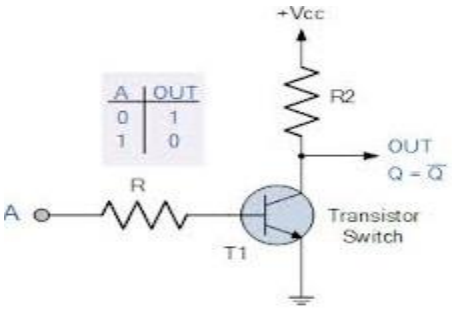
2 Input OR Gate Using Switch



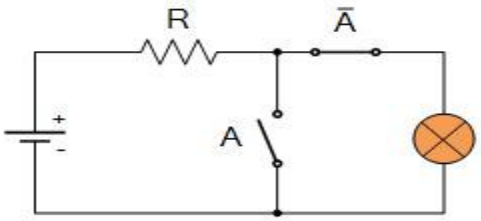
2 Input AND Gate Using Diode



2 Input AND Gate Using Switch



NOT Gate Using Transistor



Switch A – Open = "0", Lamp – ON = "1"
Switch A – Closed = "1", Lamp – OFF = "0"

NOT Gate Using Switch

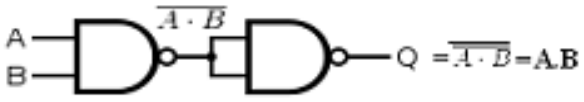
Experiment No.2

AIM:- To verify the truth table of OR, AND, NOR, Ex-OR, Ex-NOR realized using NAND & NOR gates.

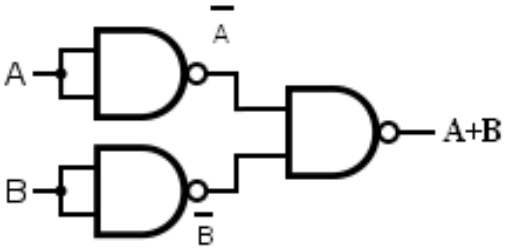
APPARATUS REQUIRED:-Digital Trainer Kit ,IC 7400,IC 7402, Connecting Lead

THEORY:- AND, OR, NOT are called basic gates as their logical operation Cannot be simplified further. NAND and NOR are called universal gates as using only NAND or only NOR any logic function can be implemented. Using NAND and NOR gates and De Morgans Theorems different basic gates & EX-OR gates are realized.

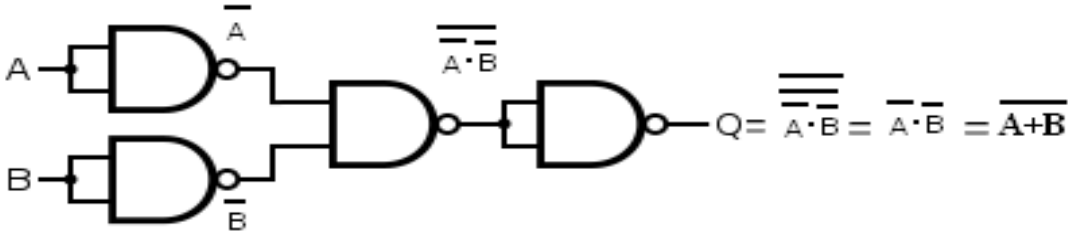
NOT Gate using NAND Gate



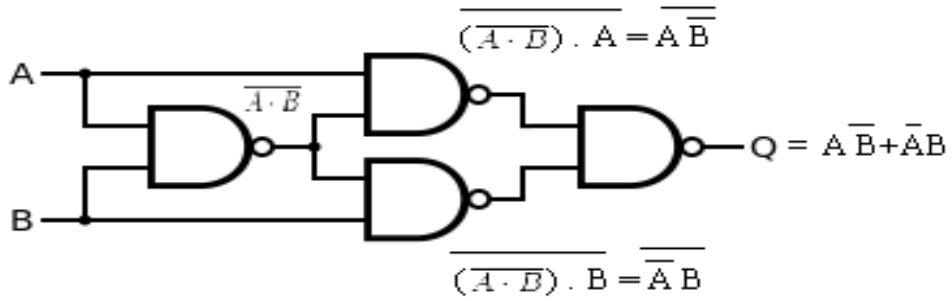
AND Gate using NAND Gate



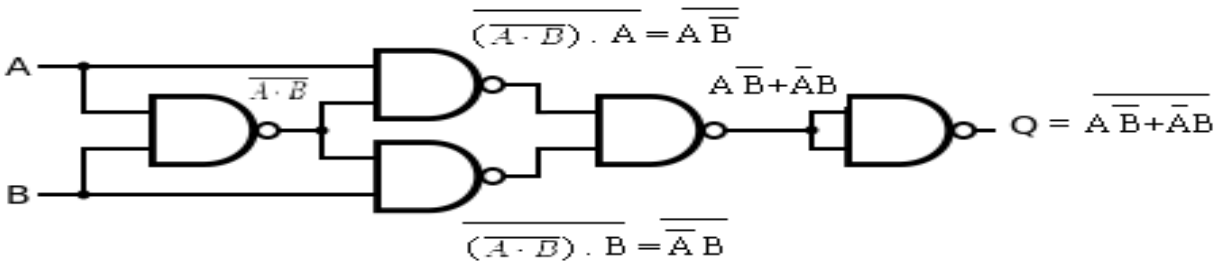
OR Gate using NAND Gate



NOR Gate using NAND Gate



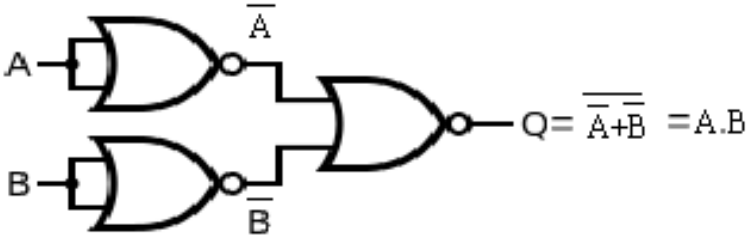
Ex-OR Gate using NAND Gate



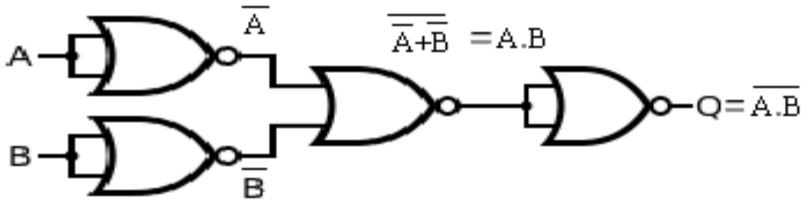
Ex-NOR Gate using NAND Gate

NOT Gate using NOR Gate

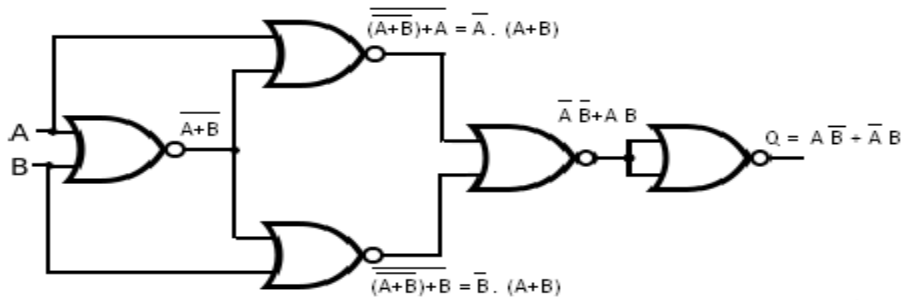
OR Gate using NOR Gate



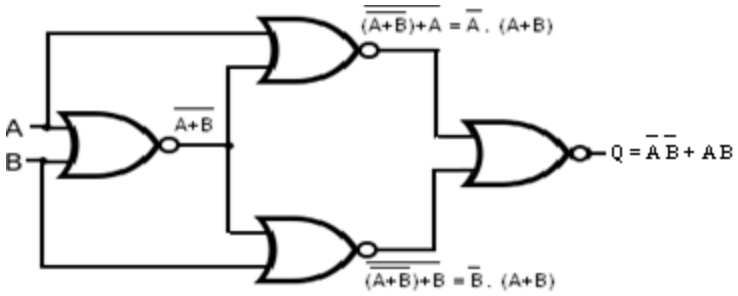
AND Gate using NOR Gate



NAND Gate using NOR Gate



Ex-OR Gate using NOR Gate



Ex-NOR Gate using NOR Gate

Experiment No. 3

Aim: To realize an SOP and POS expression.

Components Required: IC 7408 (AND), IC 7404 (NOT), IC 7432 (OR), IC 7400 (NAND), IC 7402 (NOR), IC 7486 (EX-OR), IC Trainer Kit, Patch cards

Theory:

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 two minterm.

Sum of minterms : **S**um **O**f **P**roduct **(SOP)**

Product of maxterms : **P**roduct **O**f **S**um **(POS)**

Procedure:

1. Verify that the gates are working.
2. Construct a truth table for the given problem.
3. Draw a Karnaugh Map corresponding to the given truth table.
4. Simplify the given Boolean expression manually using the Karnaugh Map.

A. Implementation Using Logic Gates:

5. Realize the simplified expression using logic gates.
6. Connect VCC and ground as shown in the pin diagram.
7. Make connections as per the logic gate diagram.
8. Apply the different combinations of input according to the truth tables. Verify that the results are correct.

B. Implementation Using Universal Gates:

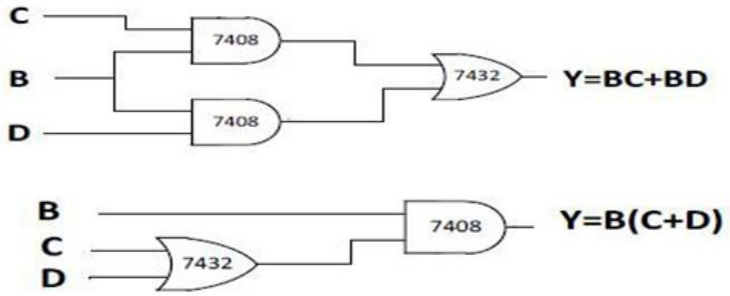
1. Convert the AND-OR logic into NAND-NAND and NOR-NOR logic.
2. Realize the simplified Boolean expressions using only NAND gates, and then using only NOR gates.
3. Connect the circuits according to the circuit diagrams, apply inputs according to the truth table and verify the results.

Simplified Boolean expression:

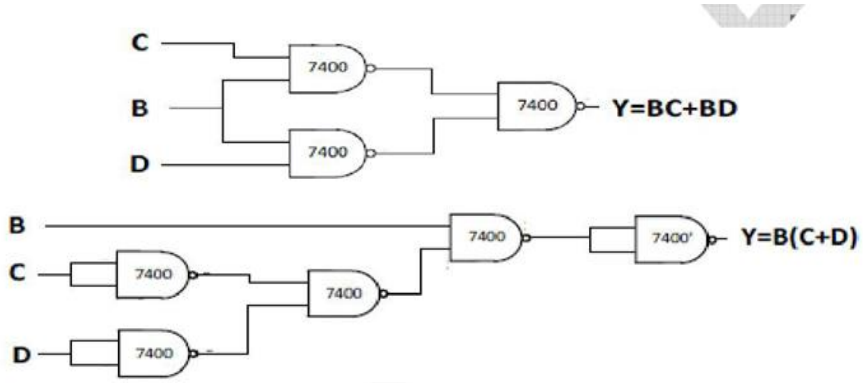
SOP form $Y=f(A,B,C,D)=BC+BD$

POS form $Y=f(A,B,C,D)=B(C+D)$

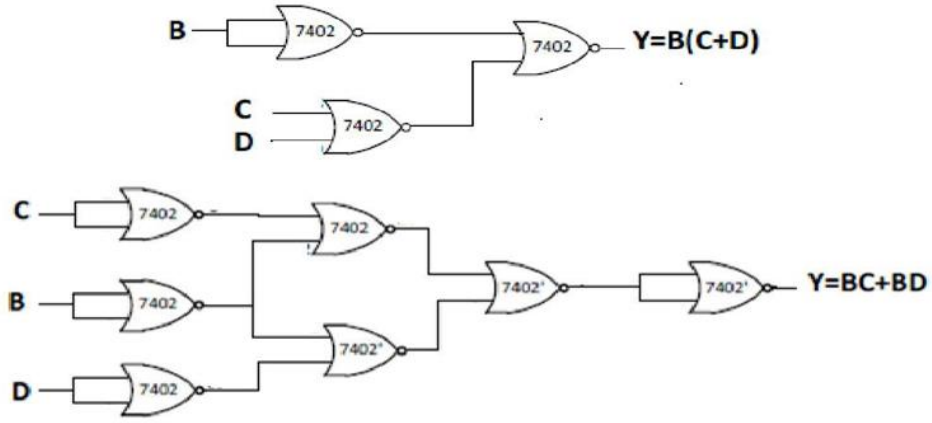
Simplification using Basic Gates:



Simplification using NAND gate



Simplification Using NOR gate



Experiment No.4

AIM:- To realize Half adder/ Subtractor & Full Adder/ Subtractor using NAND & NOR gates and to verify their truth tables.

APPARATUS REQUIRED:-Digital Trainer Kit ,IC 7404, IC 7408, IC7486, Connecting Lead.

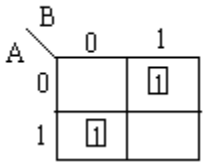
THEORY:-

Half Adder:-A half adder is a combinational circuit that performs the sum of two binary digits (A, B) to give a maximum of two binary outputs namely the sum(S) and the carry(C). Carry is the higher order bit and the lower order bit of output.

Input		Output	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

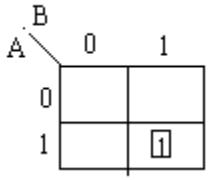
Truth table

K-MaP for Sum:-



S =A ⊕ B

K-Map for Carry:-

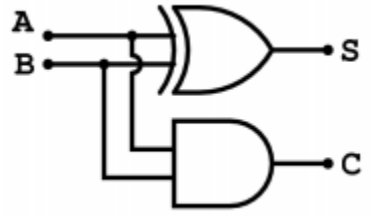


C = A.B

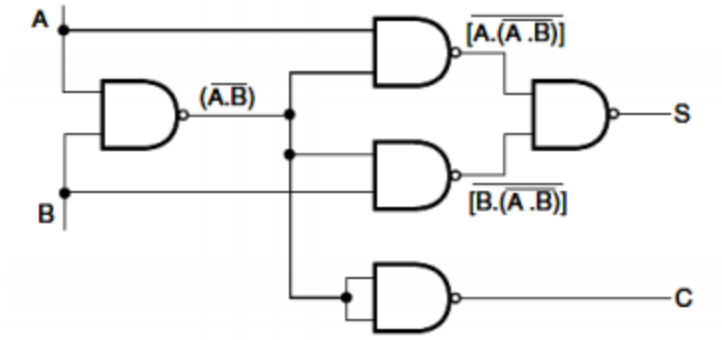
Boolean Expression:-

S=A⊕B

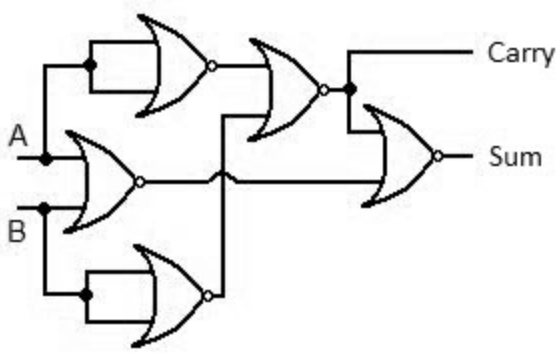
C = A.B



Logic Diagram of Half Adder



Half adder using NAND gate



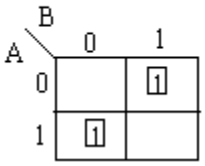
Half Adder using NOR gate

Half Subtractor:- A half subtractor is a combinational circuit that performs the difference between two binary digits (A, B) to give a maximum of two binary outputs namely the Difference and the Borrow . The Borrow output here specifies whether a '1' has been borrowed to perform the subtraction.

TRUTH TABLE:-

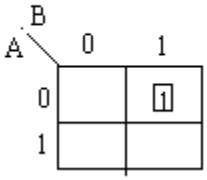
INPUT		OUTPUT	
A	B	D	B0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-Map for Difference:-



D =A ⊕ B

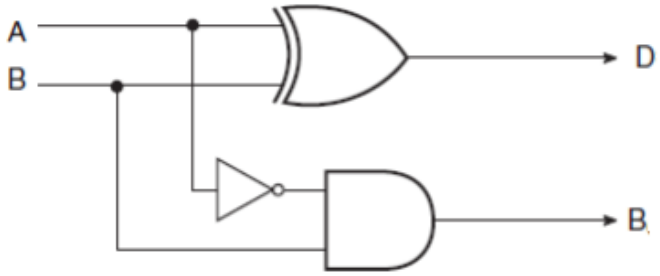
K-Map for Borrow:-



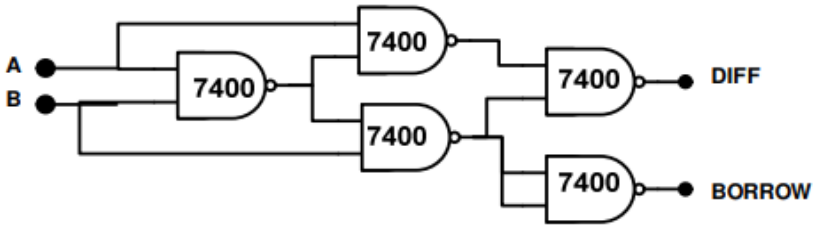
B = $\bar{A}$.B

Boolean Expression:-

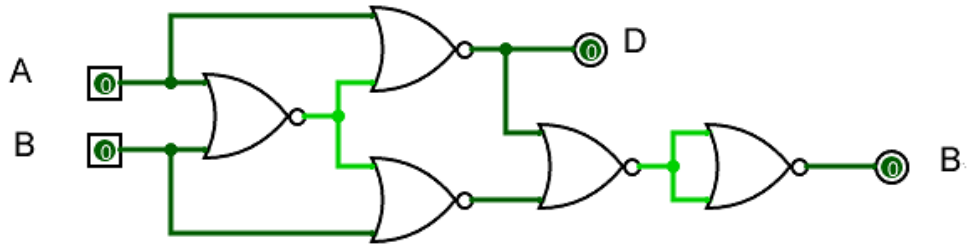
D =A ⊕B
B = $\bar{A}$.B



Logic Diagram of Half Subtractor



Half Subtractor Using NAND gate

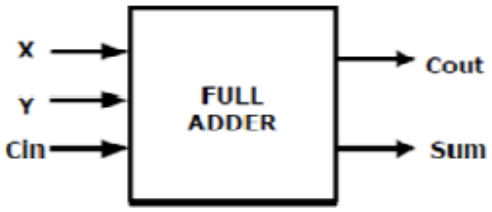


Half Subtractor Using NOR gate

Full Adder:- A full adder is a combinational circuit that performs the sum of three binary digits (A, B,Cin) to give a maximum of two binary outputs namely the sum(S) and the carry- out (Cout).The full adder becomes necessary when a carry input must be added to the two binary digits to obtain the correct sum. A half adder has no input for carries from previous circuits.

Full Adder

Logic Diagram

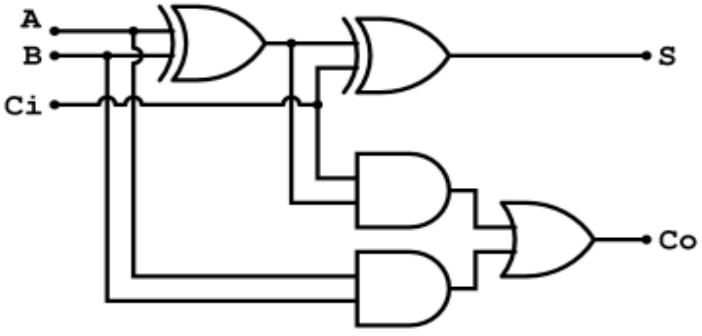


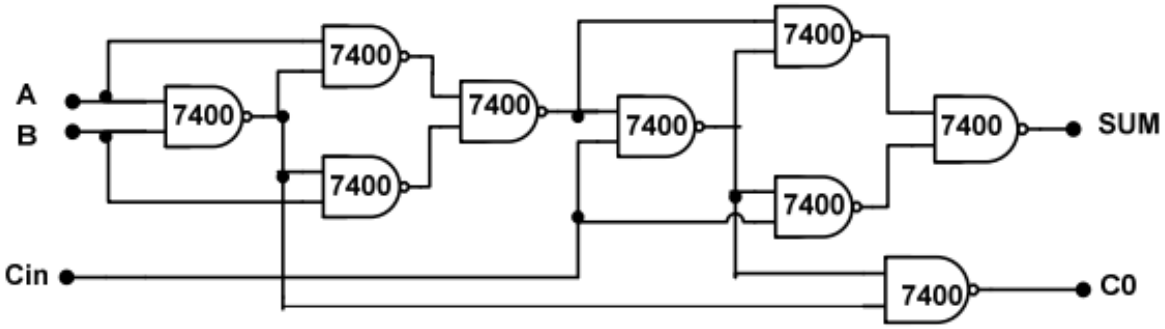
Truth Table

C_i	A	B	Sum (S)	Carry (C_o)
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

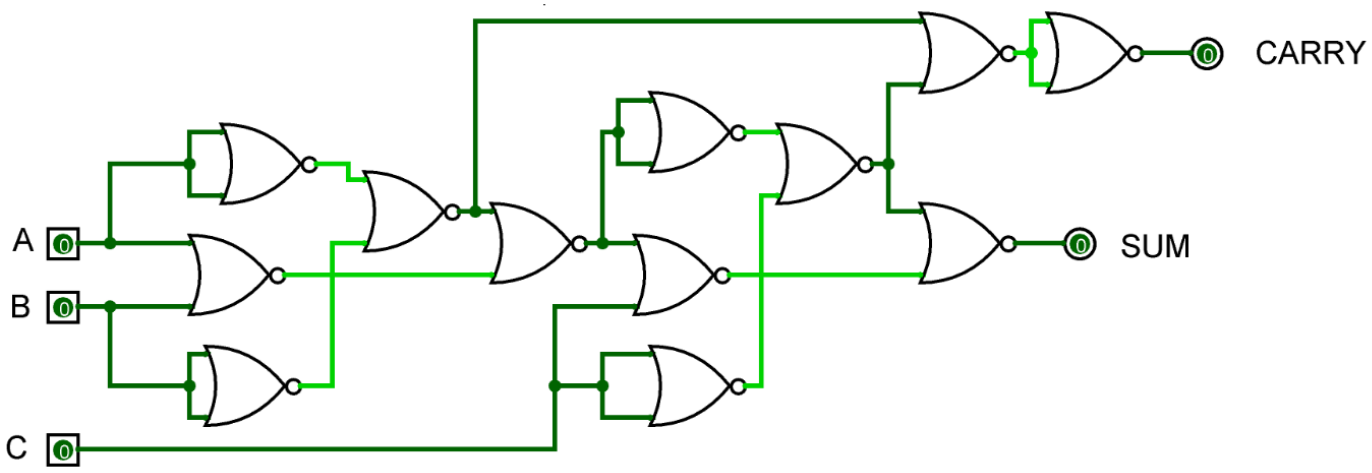
Circuit Diagram

USING BASIC AND XOR GATES





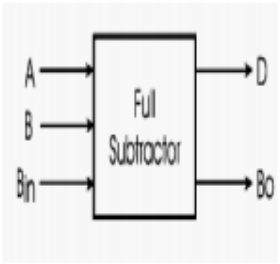
Full Adder Using NAND gate



Full adder using NOR gate

Full Subtractor:- A full subtractor is a combinational circuit that performs the three bit subtraction (A, B, Bin) to give a maximum of two binary outputs namely the difference (D) and the borrow (Bout). Full subtractor takes into consideration whether a '1' has already been borrowed by the previous adjacent lower minuend or not.

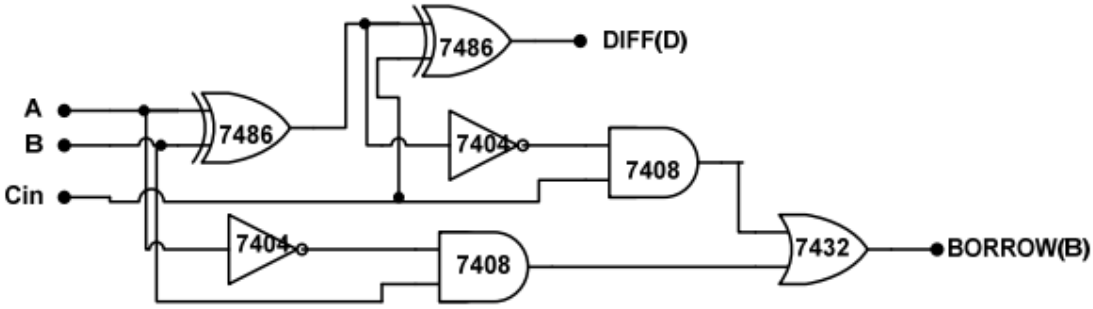
Logic Diagram:

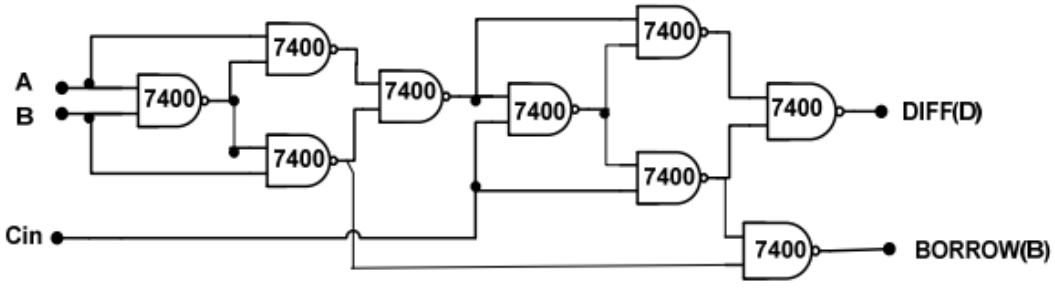


Truth Table

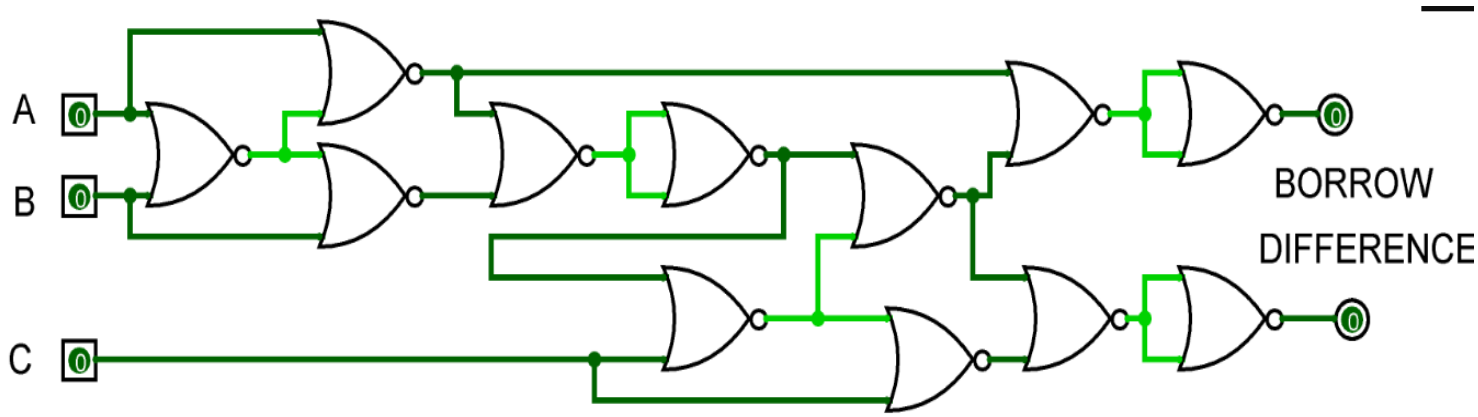
Minuend (A)	Subtrahend (B)	Borrow In (B_{in})	Difference (D)	Borrow Out (B_0)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Circuit Diagram
(USING BASIC AND XOR GATES)





Full Subtractor Using NAND gate



Full Subtractor using NOR gate

Experiment No. 5

AIM: To design and implement 4-bit adder and subtractor using IC 7483.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	IC	IC 7483	1
2.	EX-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	40

THEORY: 4 BIT BINARY ADDER: A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of next full adder in chain. The augends bits of 'A' and the addend bits of 'B' are designated by subscript numbers from right to left, with subscript 0 denoting the least significant bits. The carries are connected in chain through the full adder. The input carry to the adder is C0 and it ripples through the full adder to the output carry C4.

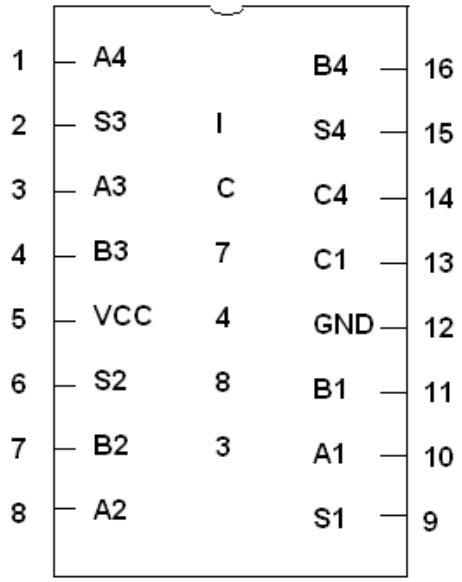
4 BIT BINARY SUBTRACTOR: The circuit for subtracting A-B consists of an adder with inverters, placed between each data input 'B' and the corresponding input of full adder. The input carry C0 must be equal to 1 when performing subtraction.

4 BIT BINARY ADDER/SUBTRACTOR: The addition and subtraction operation can be combined into one circuit with one common binary adder. The mode input M controls the operation. When M=0, the circuit is adder circuit. When M=1, it becomes subtractor.

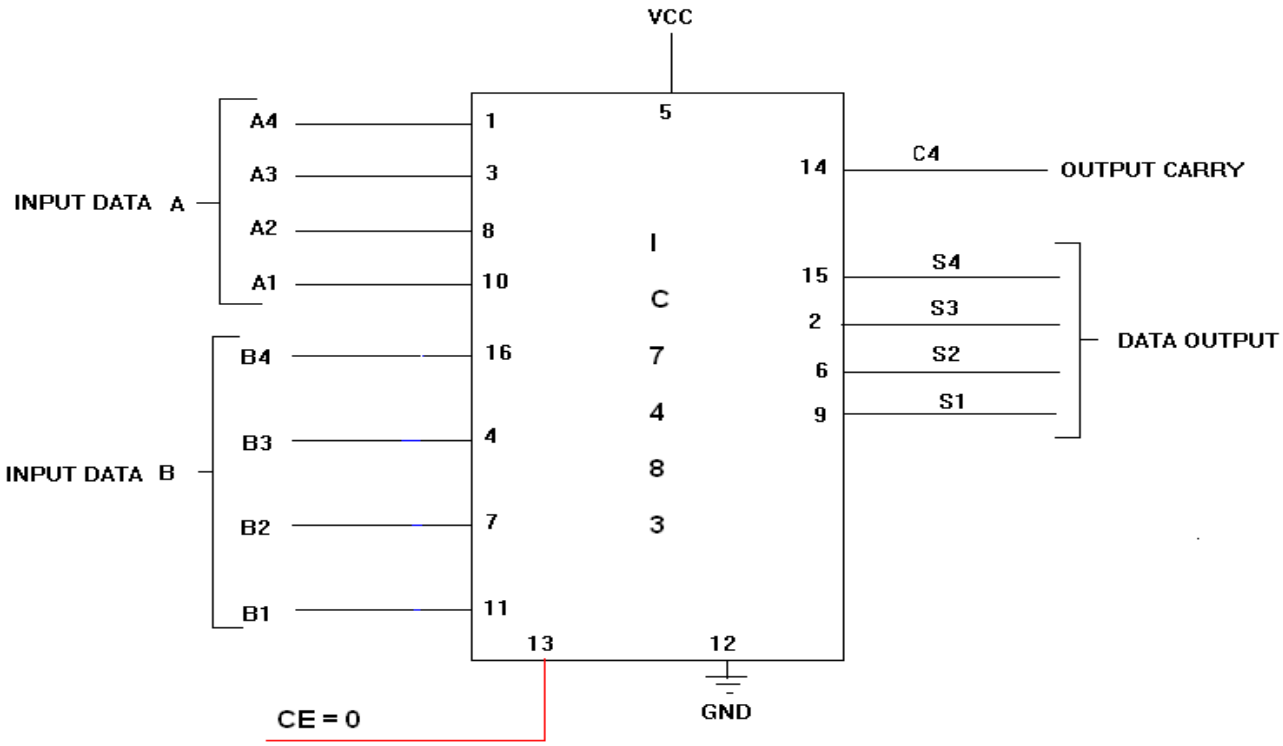
4 BIT BCD ADDER: Consider the arithmetic addition of two decimal digits in BCD, together with an input carry from a previous stage. Since each input digit does not exceed 9, the output sum cannot be greater than 19, the 1 in the sum being an input carry. The output of two decimal digits must be represented in BCD and should appear in the form listed in the columns.

ABCD adder that adds 2 BCD digits and produce a sum digit in BCD. The 2 decimal digits, together with the input carry, are first added in the top 4 bit adder to produce the binary sum.

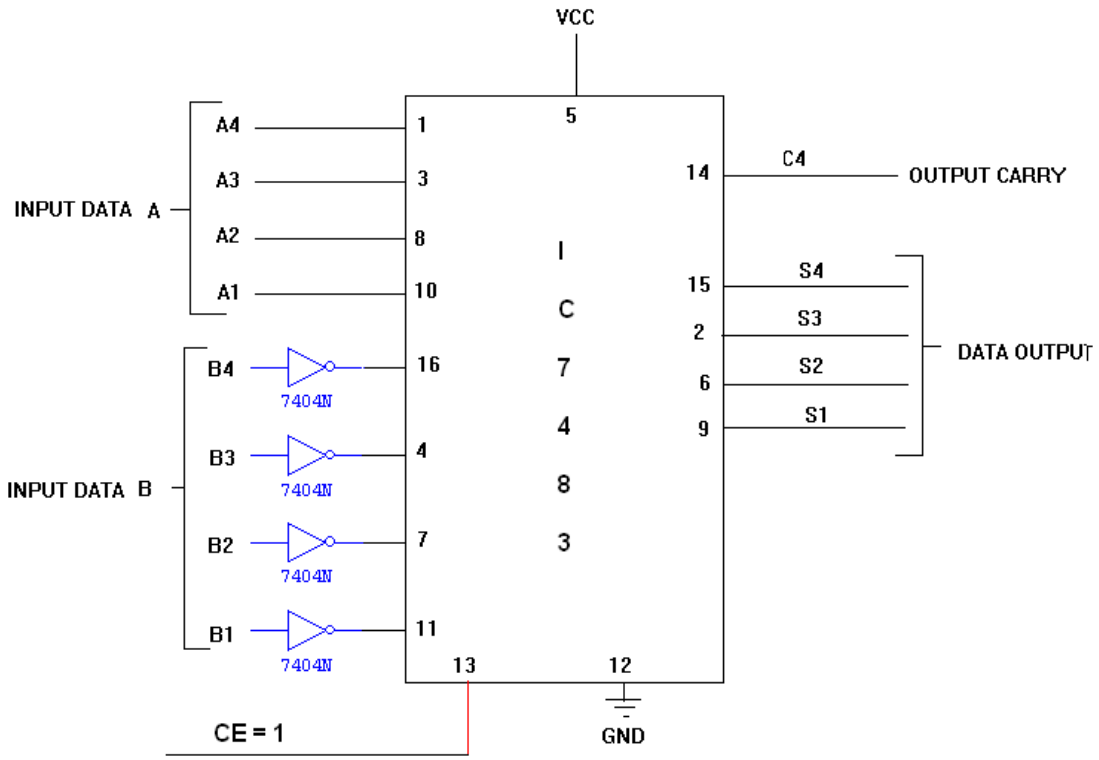
PIN DIAGRAM FOR IC 7483:



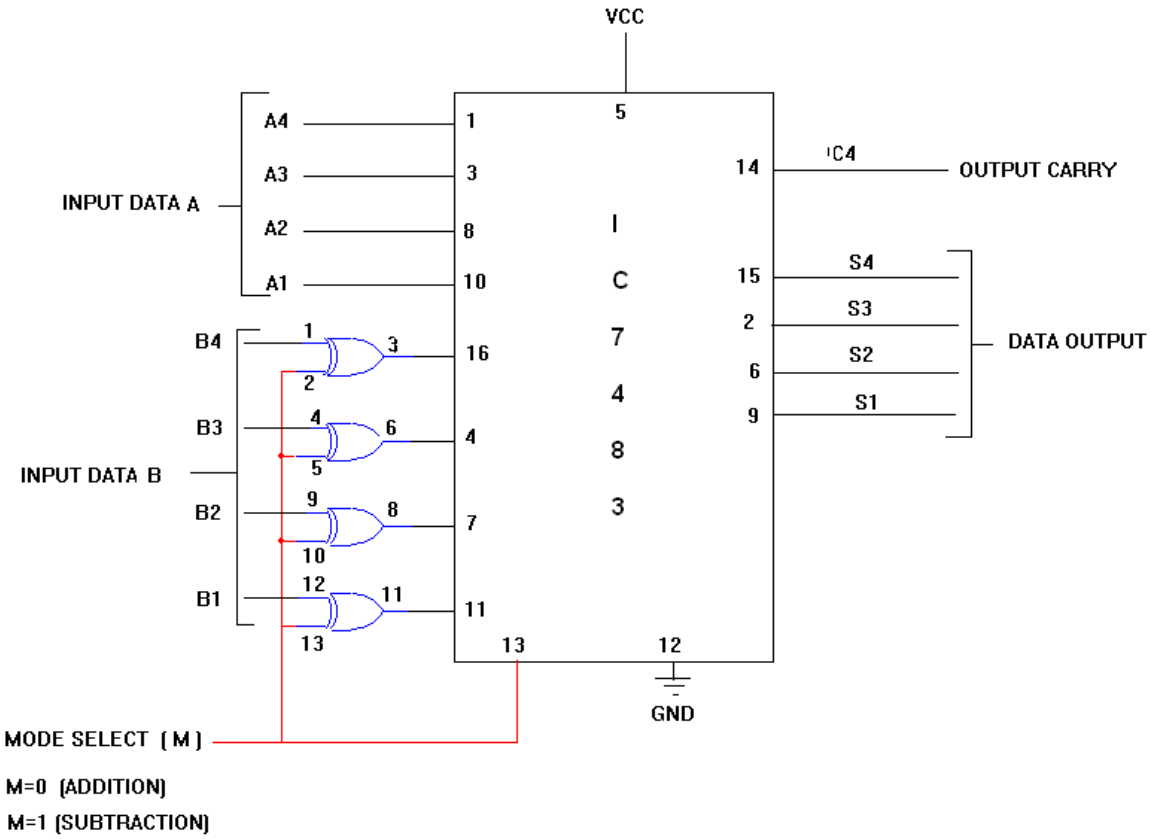
LOGIC DIAGRAM: 4-BIT BINARY ADDER



LOGIC DIAGRAM: 4-BIT BINARY SUBTRACTOR



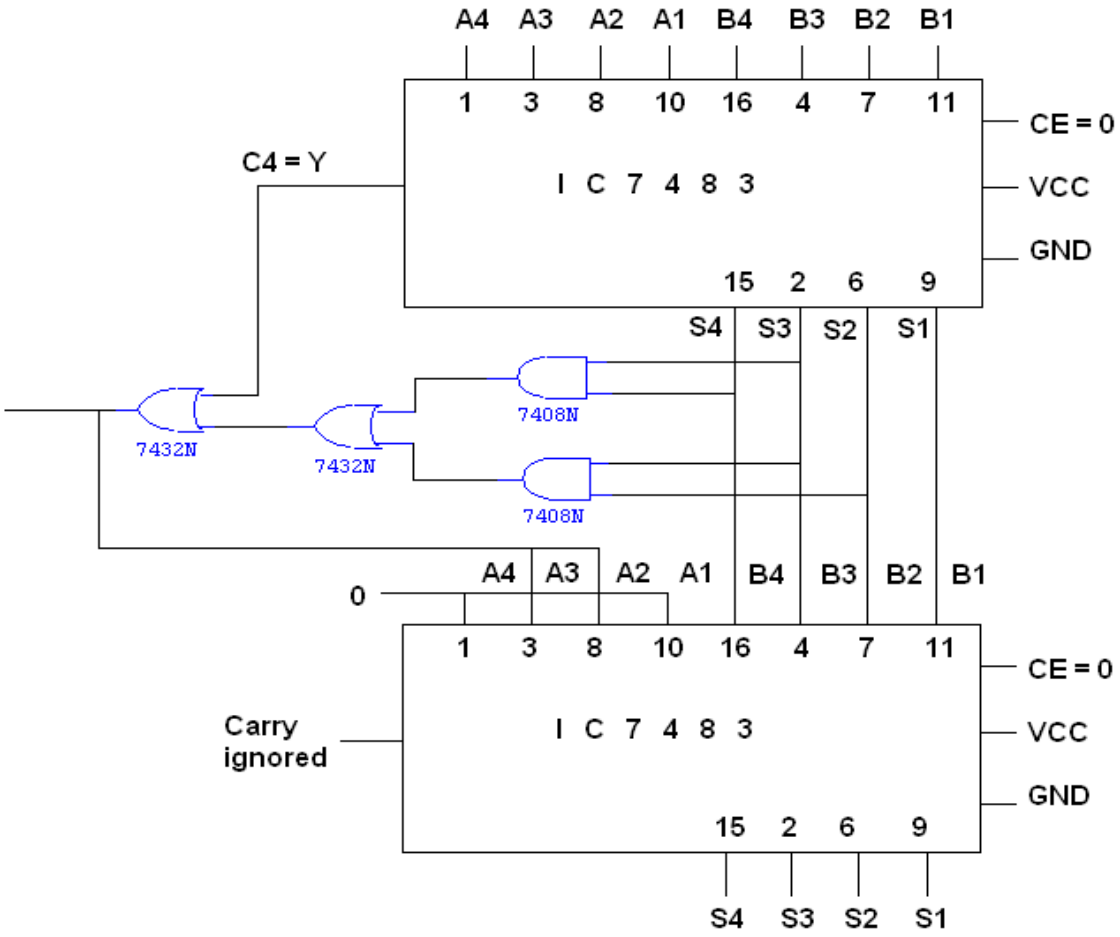
LOGIC DIAGRAM: 4-BIT BINARY ADDER/SUBTRACTOR

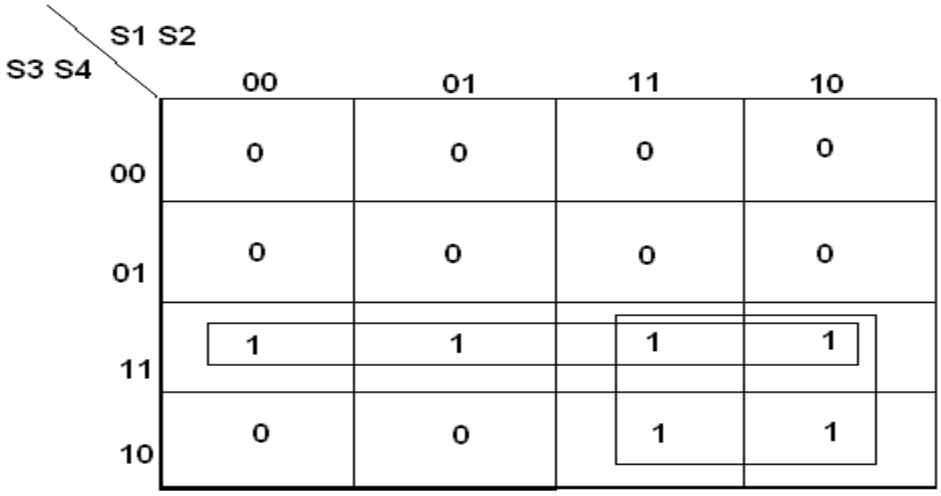


TRUTH TABLE:

Input Data A				Input Data B				Addition					Subtraction				
A4	A3	A2	A1	B4	B3	B2	B1	C	S4	S3	S2	S1	B	D4	D3	D2	D1
1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	0	1	0	0	1	1	1	1
1	1	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1
1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1

LOGIC DIAGRAM: BCD ADDER





KMAP : **Y = S4 (S3 + S2)**

TRUTH TABLE:

BCD SUM			**CARRY**	
S4	**S3**	**S2**	**S1**	**C**
0	**0**	**0**	**0**	**0**
0	**0**	**0**	**1**	**0**
0	**0**	**1**	**0**	**0**
0	**0**	**1**	**1**	**0**
0	**1**	**0**	**0**	**0**
0	**1**	**0**	**1**	**0**
0	**1**	**1**	**0**	**0**
0	**1**	**1**	**1**	**0**
1	**0**	**0**	**0**	**0**
1	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**1**
1	**0**	**1**	**1**	**1**
1	**1**	**0**	**0**	**1**
1	**1**	**0**	**1**	**1**
1	**1**	**1**	**0**	**1**
1	1	1	1	1

PROCEDURE:

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

RESULT:

Experiment No. 6

AIM: To verify the truth table of 4-to-1 multiplexer and 1-to-4 demultiplexer. Realize the multiplexer using basic gates only. Also to construct and 8- to-1 multiplexer and 1-to-8 demultiplexer using blocks of 4-to-1 multiplexer and 1-to-4 demultiplexer.

THEORY:

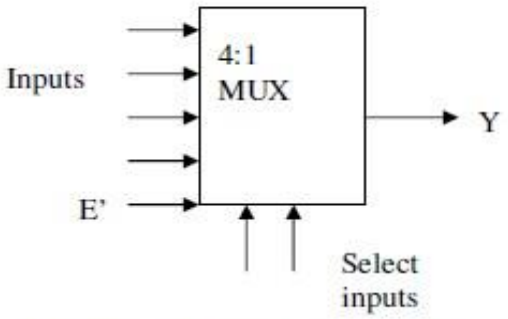
Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output. 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:

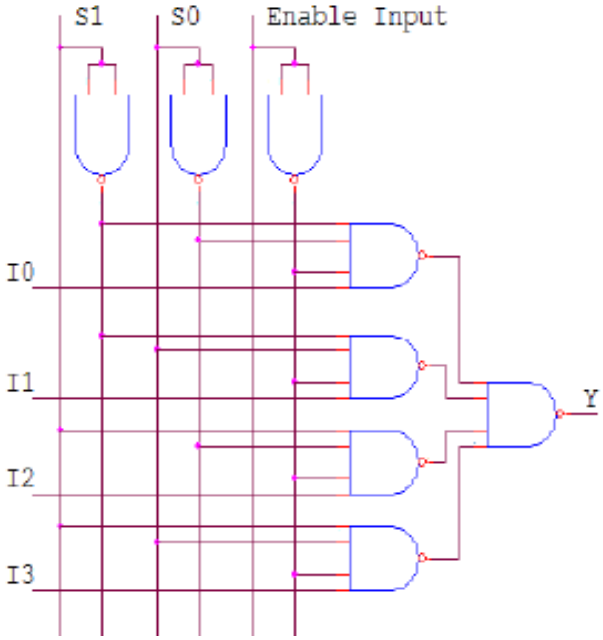
IC 7400, IC 7410, IC 7420, IC 7404, IC 74153, IC 74139, Patch Cords & IC Trainer Kit.

i) 4:1 MULTIPLEXER



Output Y= E'S1'S0'I0 + E'S1'S0I1 + E'S1S0'I2 + E'S1S0I3

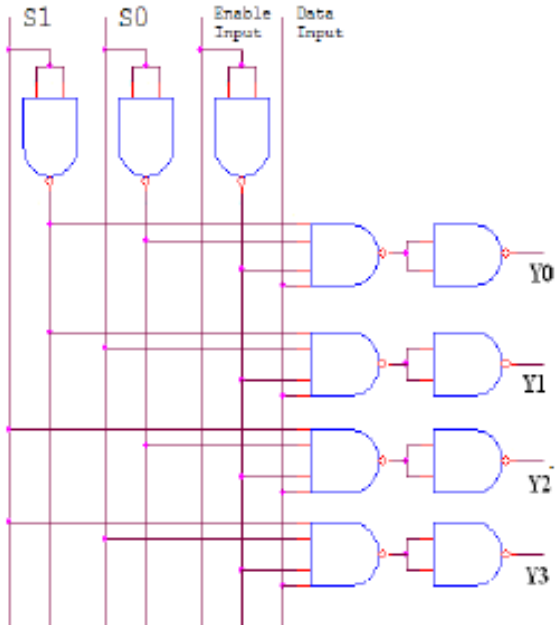
REALIZATION USING NAND GATES



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

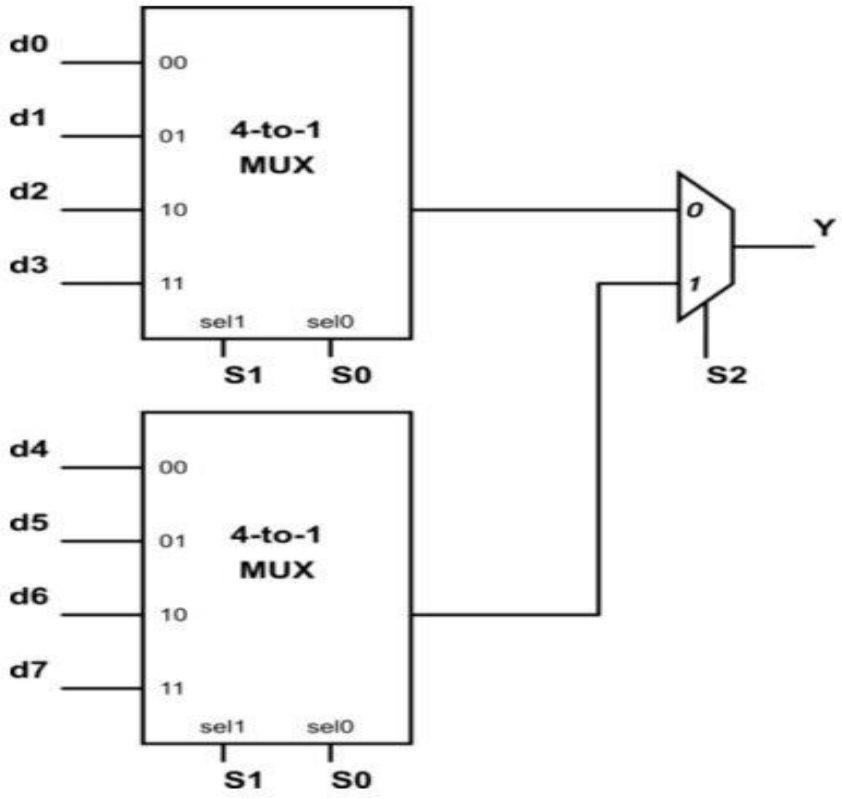
ii) DE-MUX USING NAND GATES



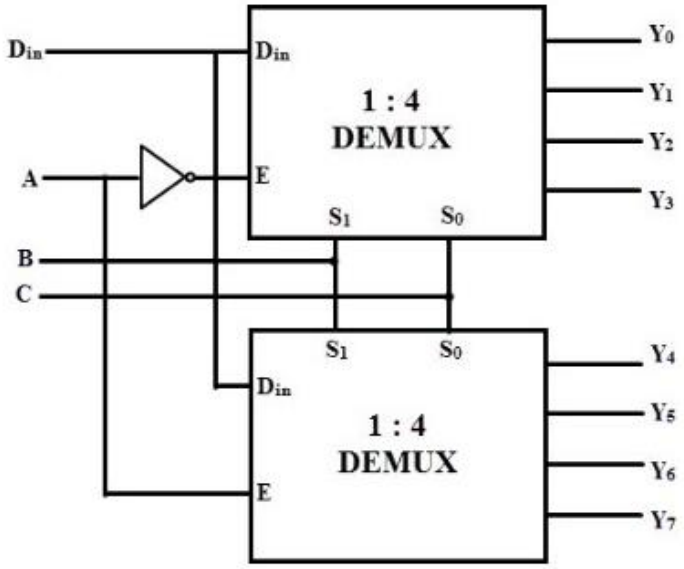
Enable Inputs	Data Input	Select Inputs		Outputs			
E	D	S_1	S_0	Y_3	Y_2	Y_1	Y_0
1	0	X	X	X	X	X	X
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	0	0	0

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.



8 to 1 Multiplexer using 4 to 1 Multiplexer



1 to 8 Demultiplexer using 1 to 4 Demultiplexer

Experiment No. 7

AIM:- Design and build BCD-to-7 segment decoder

APPARATUS REQUIRED:-Digital Trainer Kit, IC 7447, Seven Segment Display, Resistor 220 ohm, Connecting Lead

THEORY:- A Decoder IC, is a device which converts one digital format into another and the most commonly used device for doing this is the Binary Coded Decimal (BCD) to 7-Segment Display Decoder.

7-segment LED (Light Emitting Diode) or LCD (Liquid Crystal) displays, provide a very convenient way of displaying information or digital data in the form of numbers, letters or even alpha-numerical characters and they consist of 7 individual LED's (the segments), within one single display package.

In order to produce the required numbers or HEX characters from 0 to 9 and A to F respectively, on the display the correct combination of LED segments need to be illuminated and BCD to 7-segment Display Decoders such as the 74LS47 do just that.

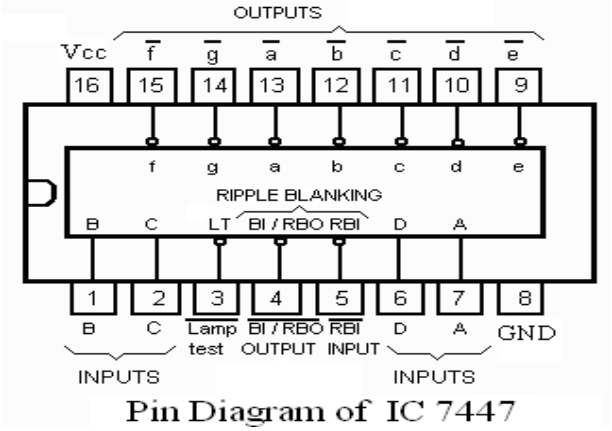
A standard 7-segment LED display generally has 8 input connections, one for each LED segment and one that acts as a common terminal or connection for all the internal segments. Some single displays have an additional input pin for the decimal point in their lower right or left hand corner.

Types of 7-segment LED digital display

Common Cathode Display (CCD) - In the common cathode display, all the cathode connections of the LED's are joined together to logic "0" and the individual segments are illuminated by application of a "HIGH", logic "1" signal to the individual Anode terminals.

Common Anode Display (CAD) - In the common anode display, all the anode connections of the LED's are joined together to logic "1" and the individual segments are illuminated by connecting the individual Cathode terminals to a "LOW", logic "0" signal.

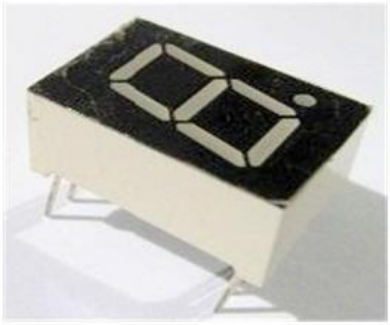
Lamp Test ($\overline{LT}$) turns all segments on so you can verify at once that all display segments are working, or identify display units that need to be replaced.



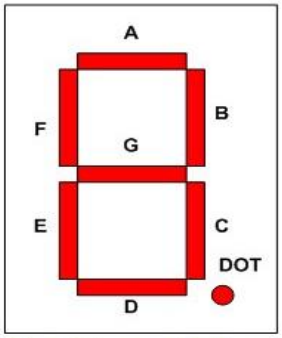
Pin Diagram of IC 7447

The Blanking input (BT) is just the reverse; it forces the entire display off. This is used in many cases to blank out leading or trailing zeros from a long display. $\overline{LT}$' will override BT so you can test even blanked-out display digits.

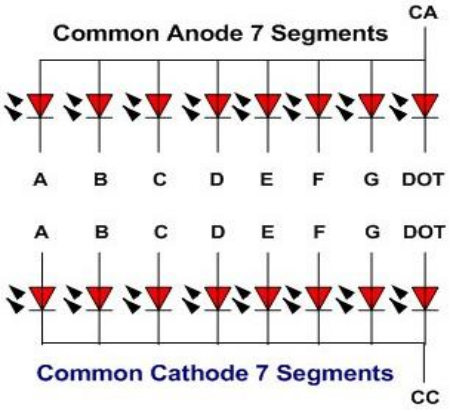
RBI stands for Ripple Blanking Input. When RBI is low and DCBA=0000 the display is blank otherwise the number is displayed on the display. This is used to remove leading zeroes from a number (e.g. display 89 instead of 089). To use with more than one display connect RBI (Ripple Blanking Output) from most significant 74xx47 to the RBI of the next 74xx47.



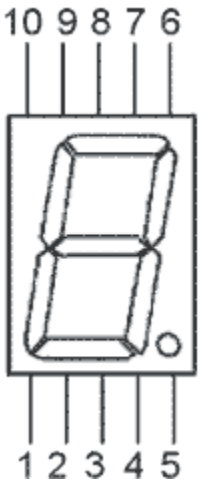
Typical 7 Segments Display



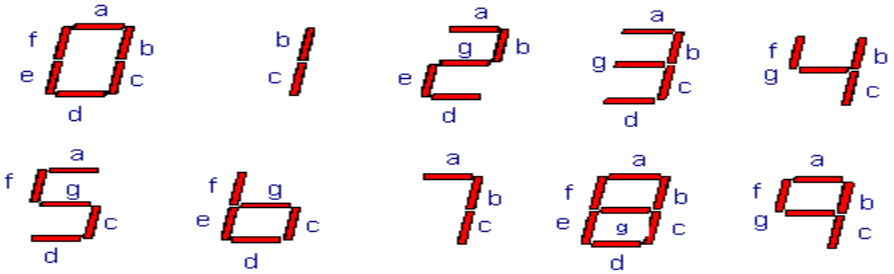
The 7 Segment's Name and the DOT



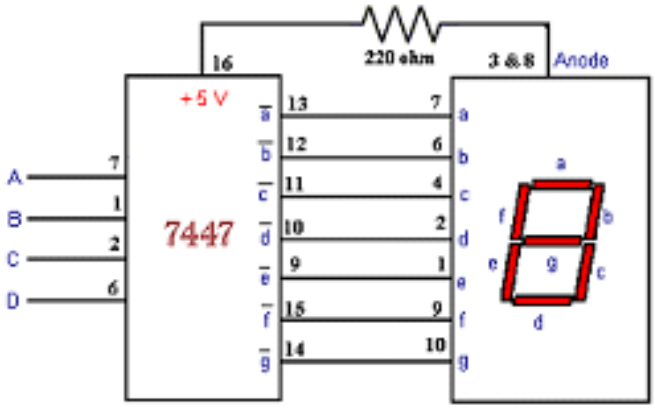
The Seven Segments Display



Pin No.	Detail
1	Cathode **e**
2	Cathode **d**
3	Com. Anode
4	Cathode **c**
5	Cathode D.P
6	Cathode **b**
7	Cathode **a**
8	Com. Anode
9	Cathode **f**
10	Cathode **g**



Inputs				Outputs						
D	C	B	A		$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0



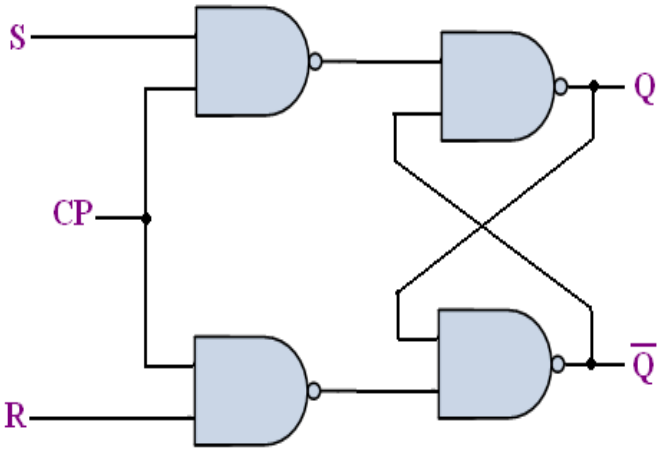
Experiment No.8

AIM:- Using basic logic gates, realize the R-S, J-K and D-flip flops with and without clock signal and verify their truth table.

APPARATUS REQUIRED:- Digital Trainer Kit, IC 7400,IC 7402,IC 7410, Connecting Lead.

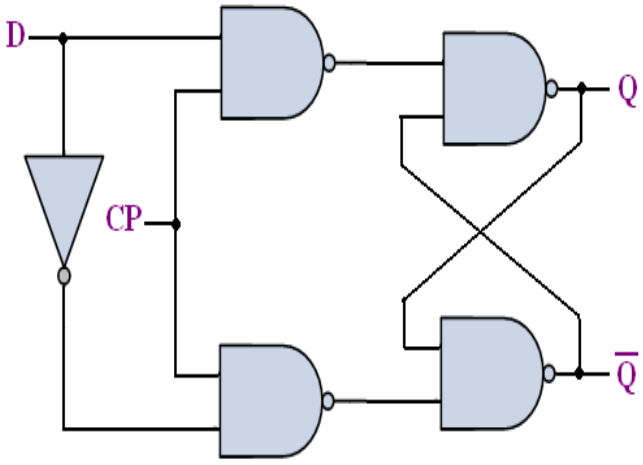
THEORY:-

Logic Diagram of S-R Flip Flop



Truth Table:-

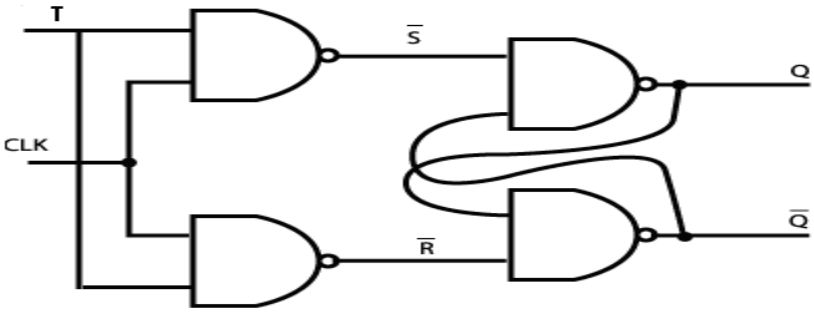
Inputs			Outputs		State
CP	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	X	X	Q_n	$\overline{Q}_n$	No change
1	0	0	Q_n	$\overline{Q}_n$	No change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	Indeterminate		Avoid this condition



Logic Diagram of D Flip Flop

Truth Table:-

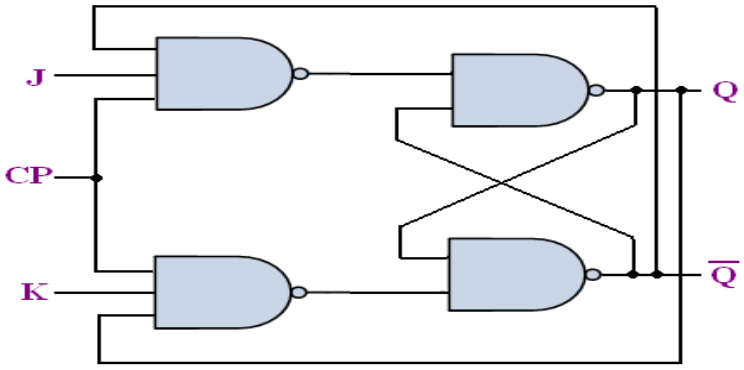
Inputs		Outputs		State
CP	D	Q_{n+1}	$\overline{Q}_{n+1}$	
0	X	Q_n	$\overline{Q}_n$	No change
1	0	0	1	Reset
1	1	1	0	Set



Logic Diagram of T Flip Flop

Truth Table:-

Input	Outputs	
	Present State	Next State
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0



Logic Diagram of JK Flip Flop

Truth Table:-

Inputs			Outputs		State
CP	J	K	Q_{n+1}	$\overline{Q}_{n+1}$	
0	X	X	Q_n	$\overline{Q}_n$	No change
1	0	0	Q_n	$\overline{Q}_n$	No change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	$\overline{Q}_n$	Q_n	Toggle

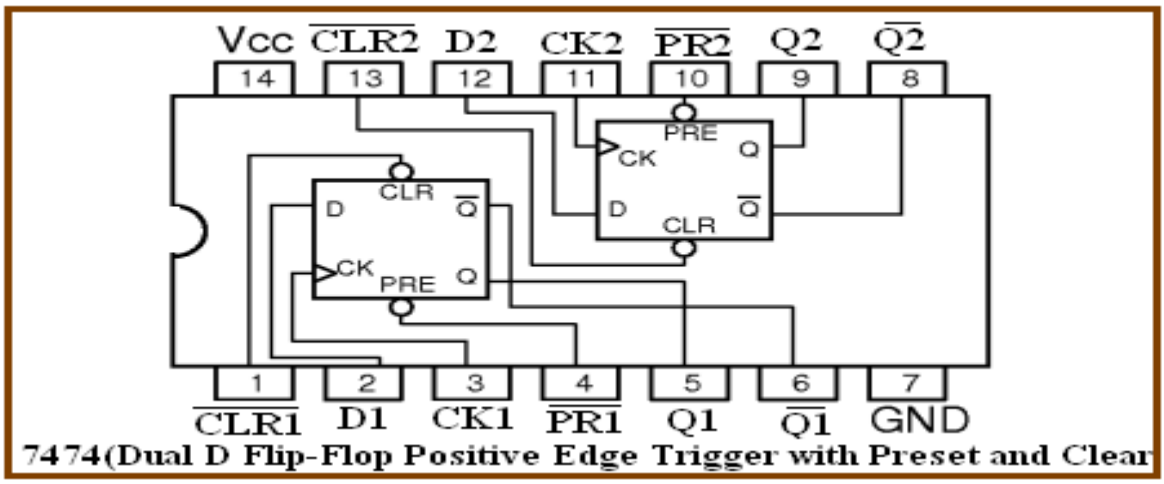
Experiment No.9

AIM:- To design and implement the 2-bit up counter using D Flip-Flop

APPARATUS REQUIRED:-Digital Trainer Kit, IC 7474, Connecting Lead.

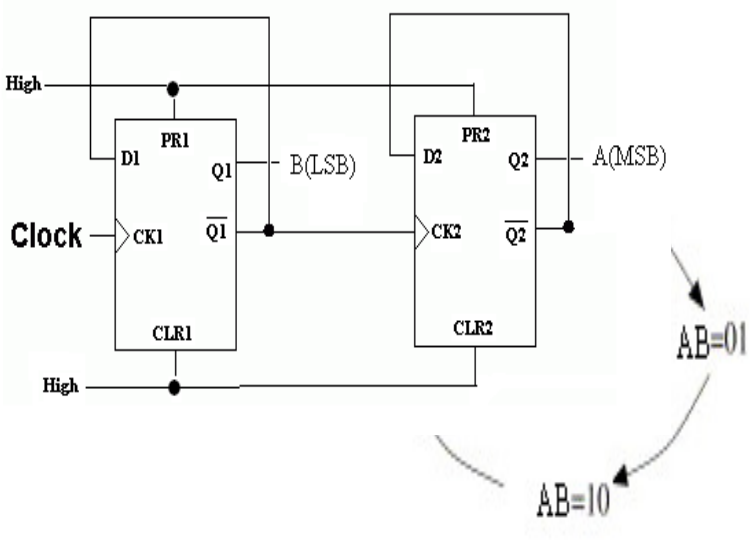
THEORY:- Counters are a specific type of sequential circuit.

- The state or the flip-flop values themselves, serves as the "output."
- The output value increases by one on each clock cycle.
- After the largest value, the output "wraps around" back to 0.



Pin Diagram Of Dual D Flip Flop Positive- Edge Triggered with Preset & Clear

Pre-set (PR) and clear (CLR) inputs are active-low .These inputs are asynchronous; they operate at any time whenever set low .Pre-set sets Q high, and clear sets Q low. Also note that the D flip- flop is rising-edge-triggered



Truth Table:-

Clock	Outputs		Decimal Count
	A	B	
0	0	0	0
1	0	1	1
2	1	0	2
3	1	1	3

Experiment No. 10

AIM: DESIGN AND IMPLEMENTATION OF SHIFT REGISTER

To design and implement

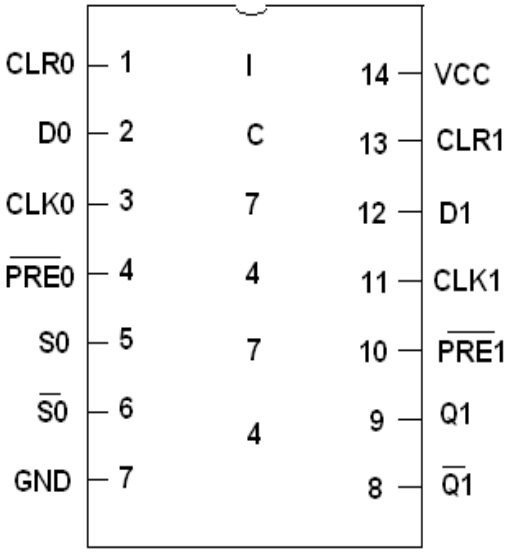
- Serial in serial out
- Serial in parallel out
- Parallel in serial out
- Parallel in parallel out

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

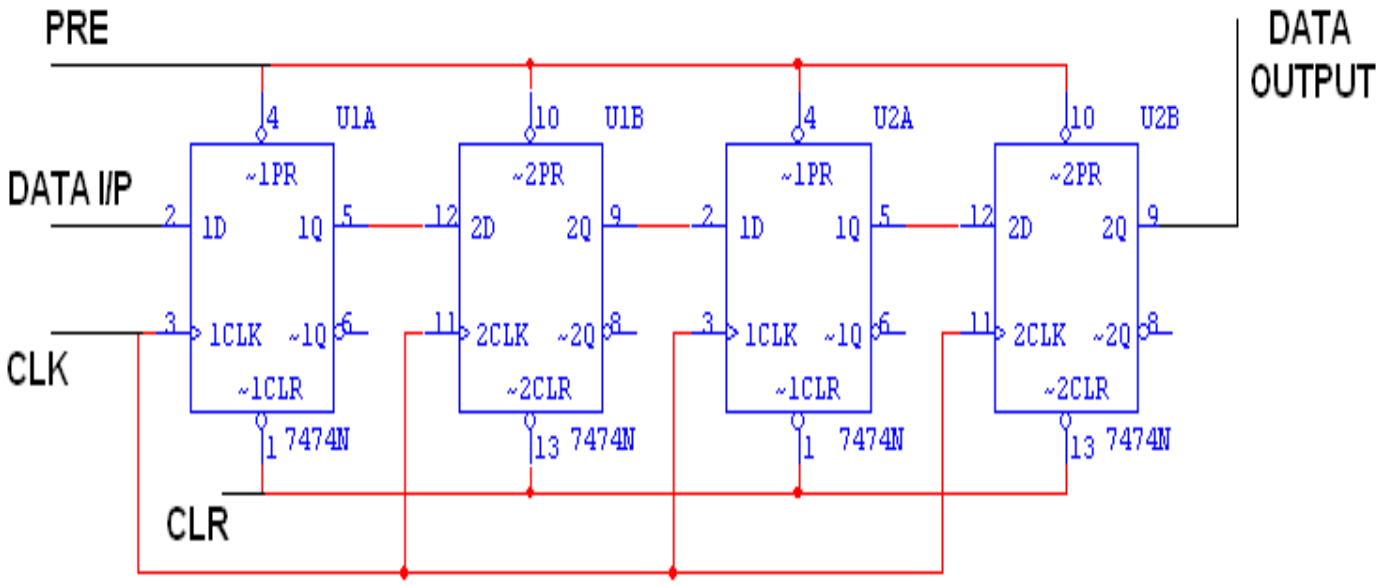
THEORY: A register is capable of shifting its binary information in one or both directions is known as shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses 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:



LOGIC DIAGRAM:

SERIAL IN SERIAL OUT:

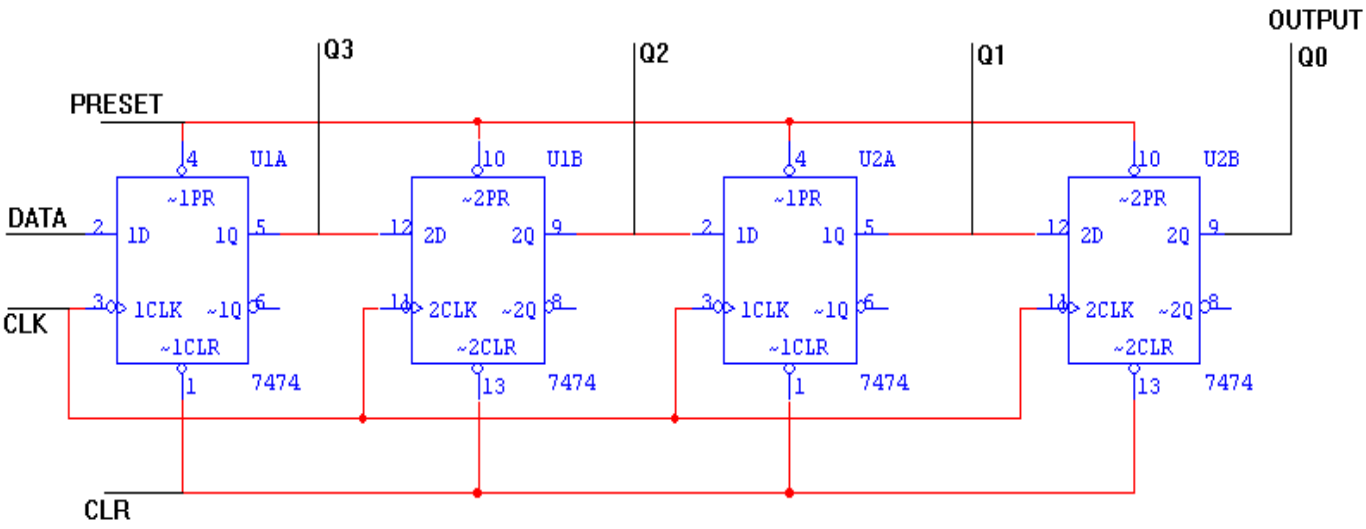


TRUTH TABLE:

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:

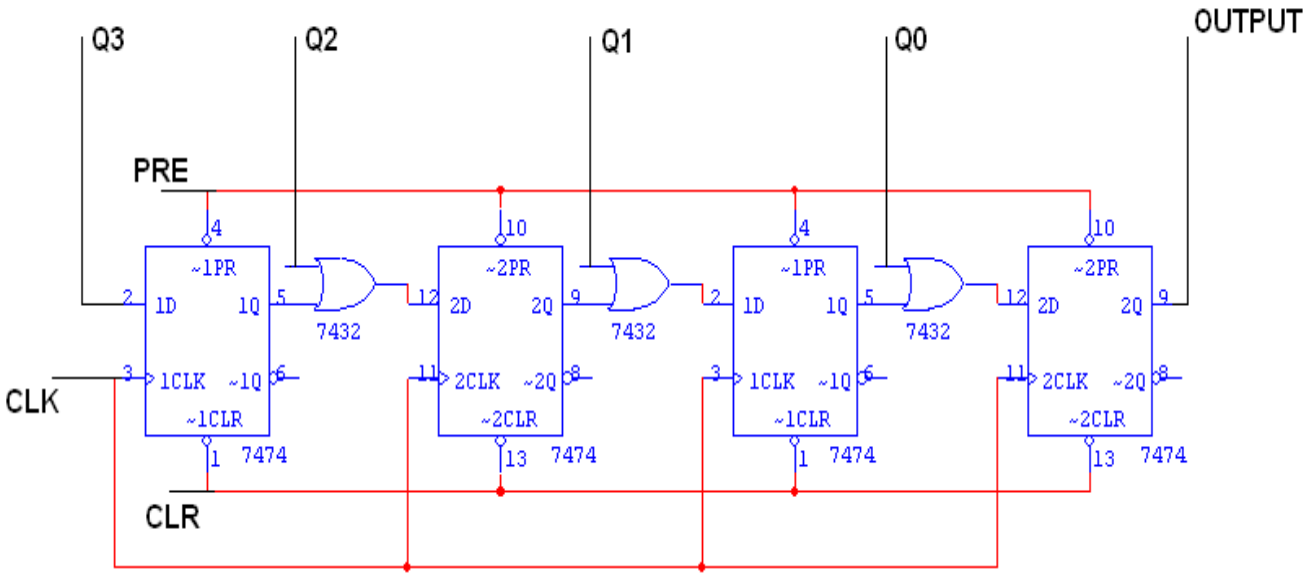


TRUTH TABLE:

CLK	DATA	OUTPUT			
		QA	QB	QC	QD
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

LOGIC DIAGRAM:

PARALLEL IN SERIAL OUT:

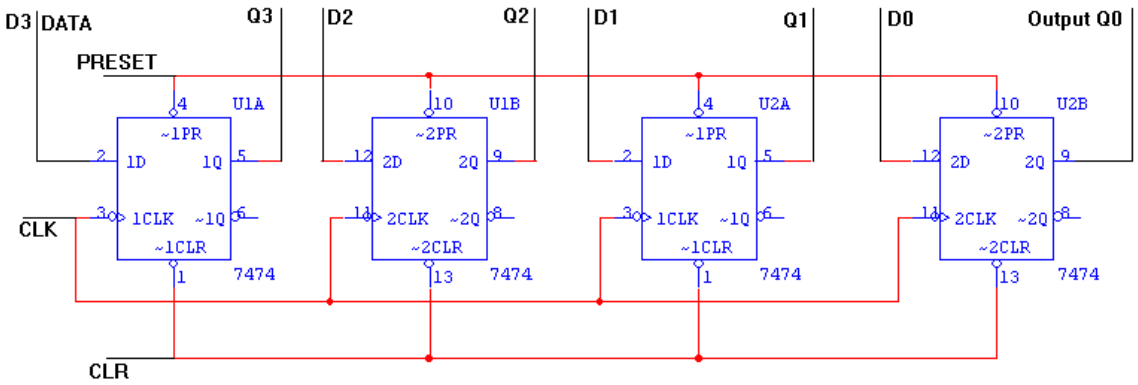


TRUTH TABLE:

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			
	DA	DB	DC	DD	QA	QB	QC	QD
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: