JHARSUGUDA ENGINEERING SCHOOL, JHARSUGUDA

Laboratory Manual

ON

Digital Electronics And Microprocessor

(5th Sem. EE)

Prepared by: JYOTI NAIK

(Lect. in E&TC)

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Sl no	Name of the experiment
1	Verify truth tables of AND, OR, NOT, NOR, NAND, XOR, XNOR gates
2	Implement various gates by using universal properties of NAND and NOR gates and verify truth table
3	Implement half adder and full adder using logic gates
4	Implement half subtractor and full subtractor using the logic gate
5	To study multiplexer and Demultiplexer.
6	Study of flip -flops 1) S-R flip -flop 2) J-K flip -flop 3) D flip -flop 4) T flip –flop.
7	Implement Mode-10 asynchronous counters
8	Study shift registers
9	To write programs to find out 1’s complement and 2’s complement of an 8 bit number
10	To write programs for finding out addition operation and subtraction operation of 8 bit numbers.
11	To Write Program For Finding 1’s And 2’s Complement Of A 16-Bit Number.
12	To write program for finding binary addition and decimal addition of two 8-bit numbers; sum: 16-bits
13	To write program for finding larger of two numbers and largest number in a data array.

Experiment-1

Aim of the experiment: -

Verify truth tables of AND, OR, NOT, NOR, NAND, XOR, XNOR gates.

Apparatus required: -

1)Electrical and Electronics system Trainer.

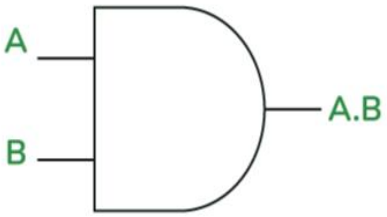
2)Digital logic gate Experiment panel.

3)Connecting wires (As per required)

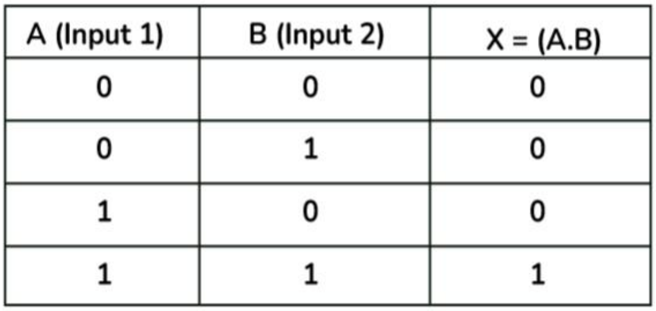
Theory: -

AND Gate

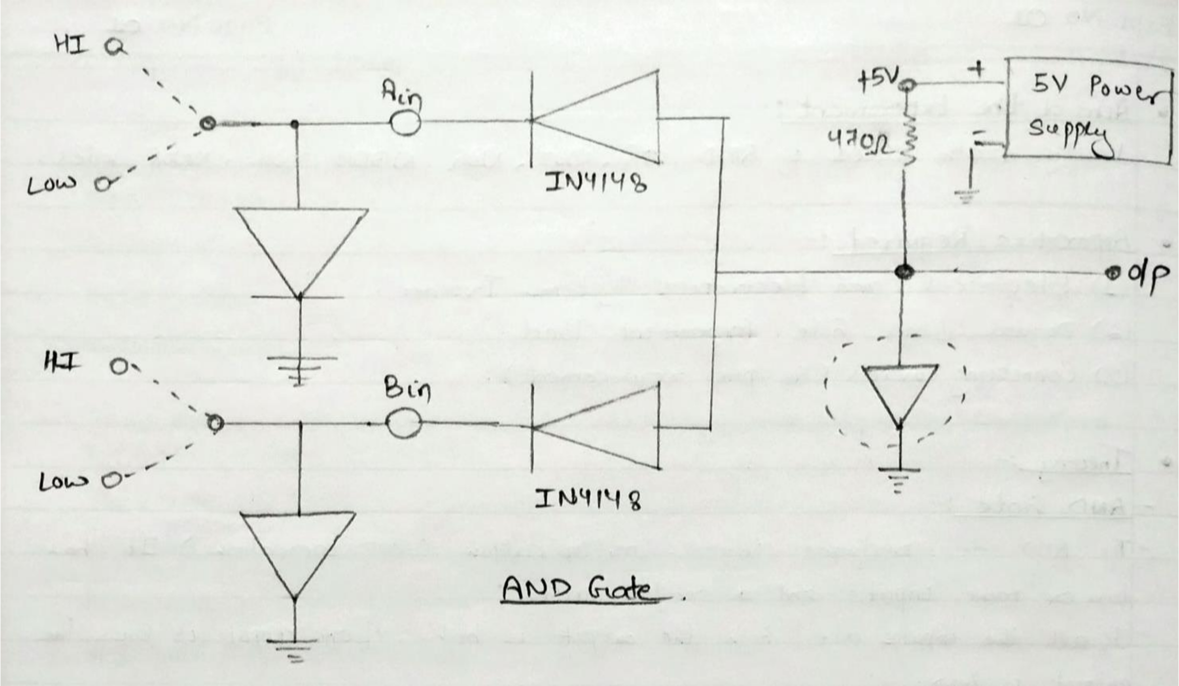
- The AND gate performs logical multiplication (AND operation). It has two or more inputs and a single output.
- If all the inputs are high, the output is high, if any input is low the output is low.
- If A and B are the inputs and Y is the output then Y=A.B



Truth Table

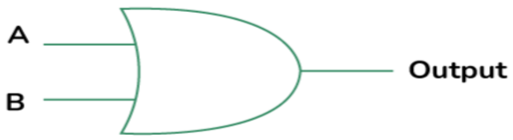


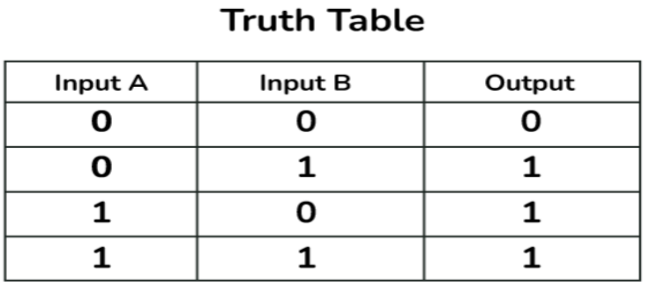
A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1



OR Gate

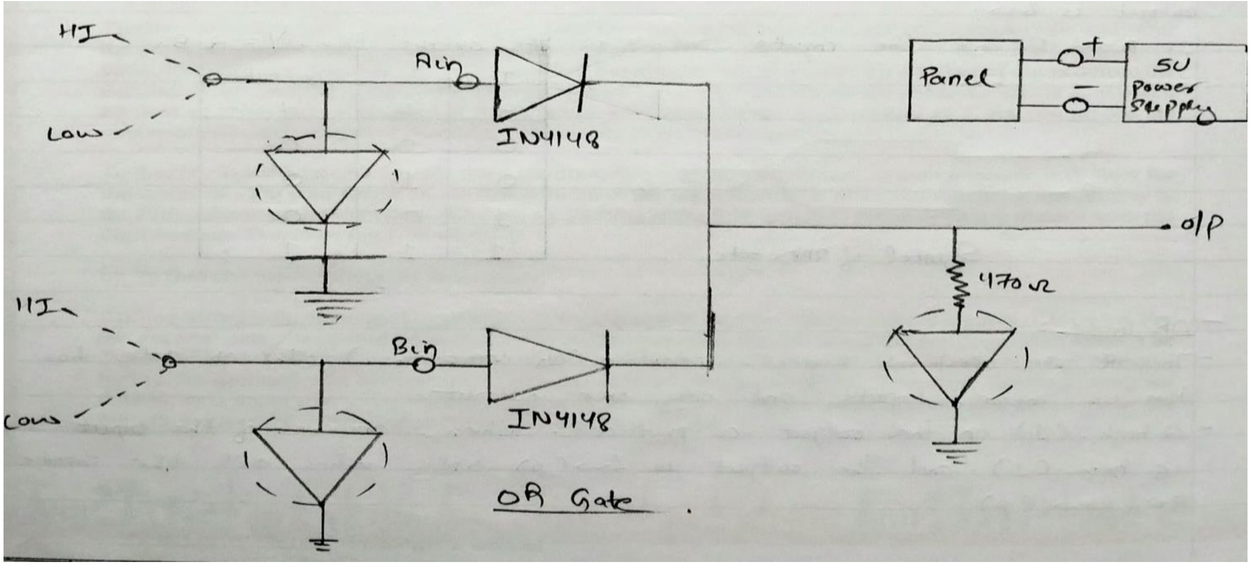
- The OR gate performs logical addition (OR operation). The OR gate has two or more inputs and only one output.
- A high(1) output is produced when anyone of the input is high(1) and the output is low(0) only when all the inputs are low(0)
- If A and B are inputs, then Y output will be Y=A+B.





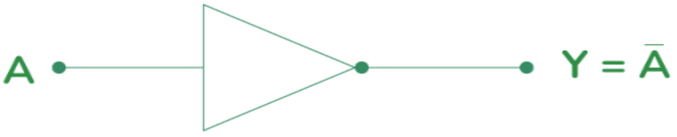
Truth Table

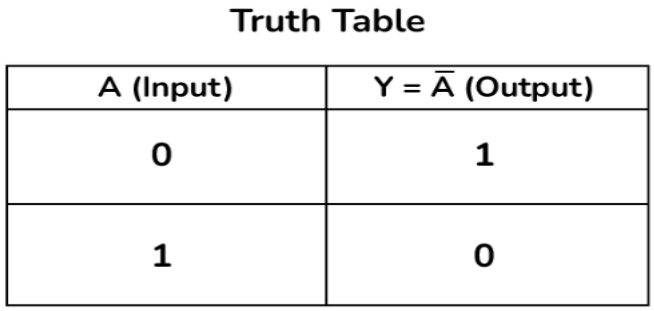
Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1



NOT Gate (inverter)

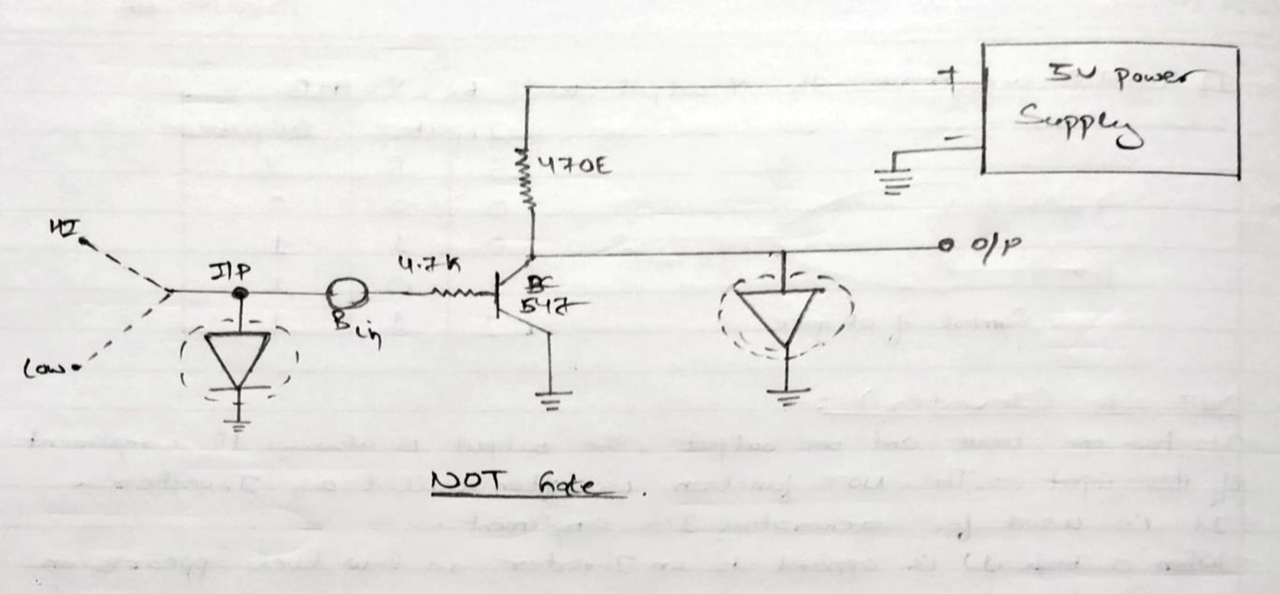
- It has one input and one output the output is always the complement of its input. The NOT function is often called as inverter.
- It is used for generating 1's complement.
- When a high(1) is applied to an inverter, a low level appears on its output and vice versa.





Truth Table

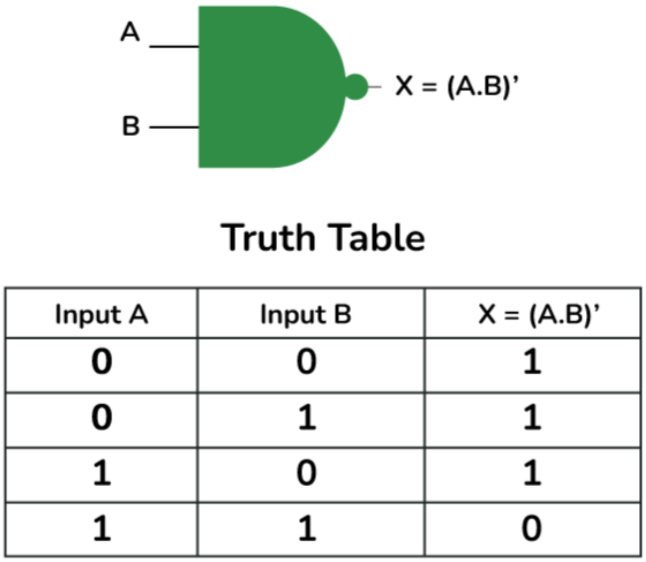
A (Input)	Y = Ā (Output)
0	1
1	0



NOT Gate.

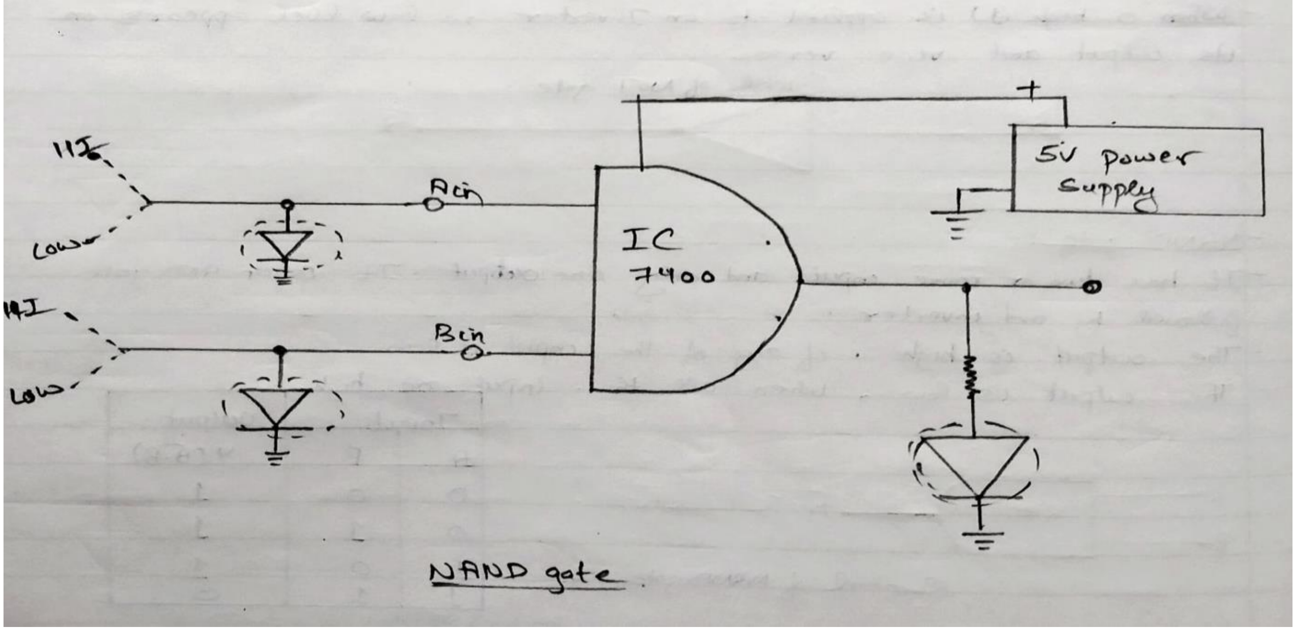
NAND Gate

- It has two or more inputs and only one output. It is an AND gate followed by an inverter.
- The output is high, if any of the input is low.
- The output is low, when all the inputs are high.



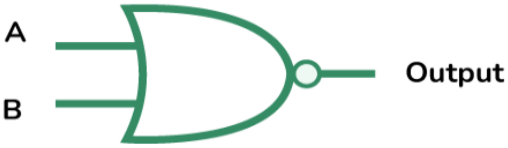
Truth Table

Input A	Input B	X = (A.B)'
0	0	1
0	1	1
1	0	1
1	1	0



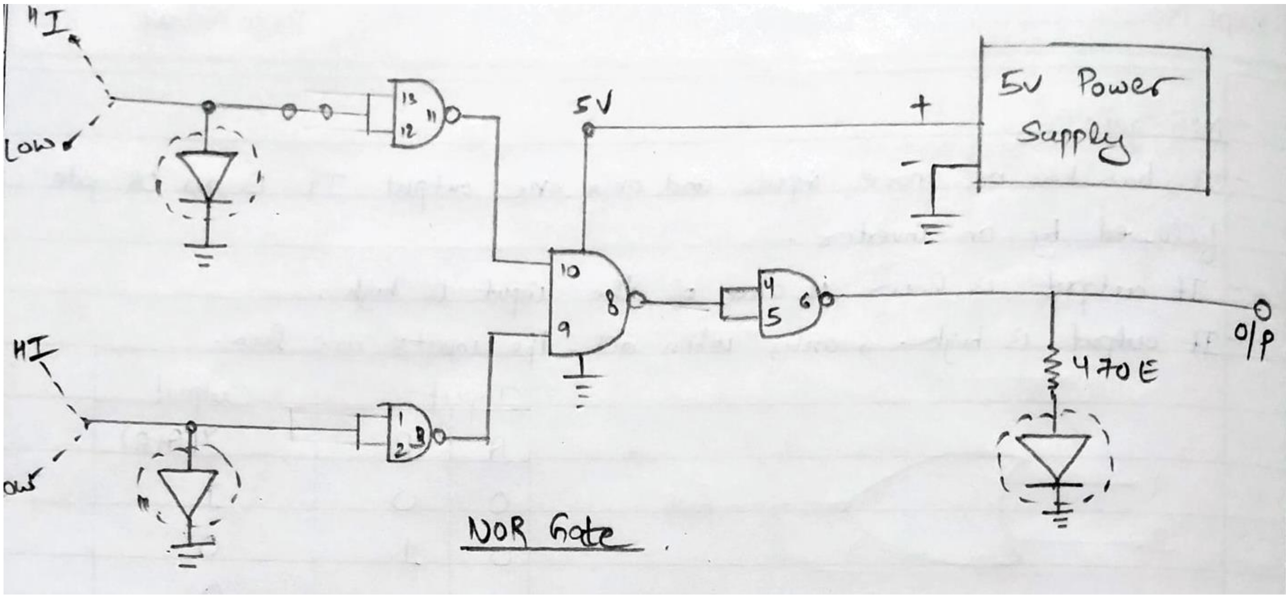
NOR Gate

- It has two or more inputs and only one output. It is an OR gate followed by an inverter.
- It output is low, if any of the input is high.
- It output is high, only when all the inputs are low.



Truth Table

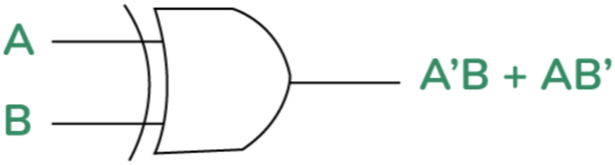
Input A	Input B	0 = (A + B)’
0	0	1
0	1	0
1	0	0
1	1	0



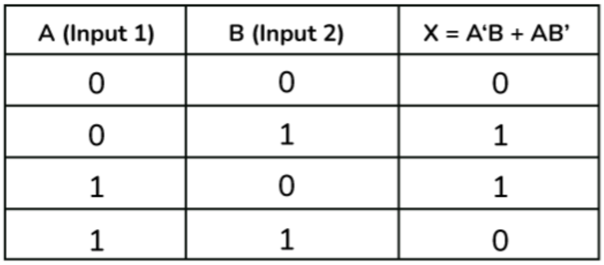
NOR Gate

Exclusive OR gate (EXOR gate)

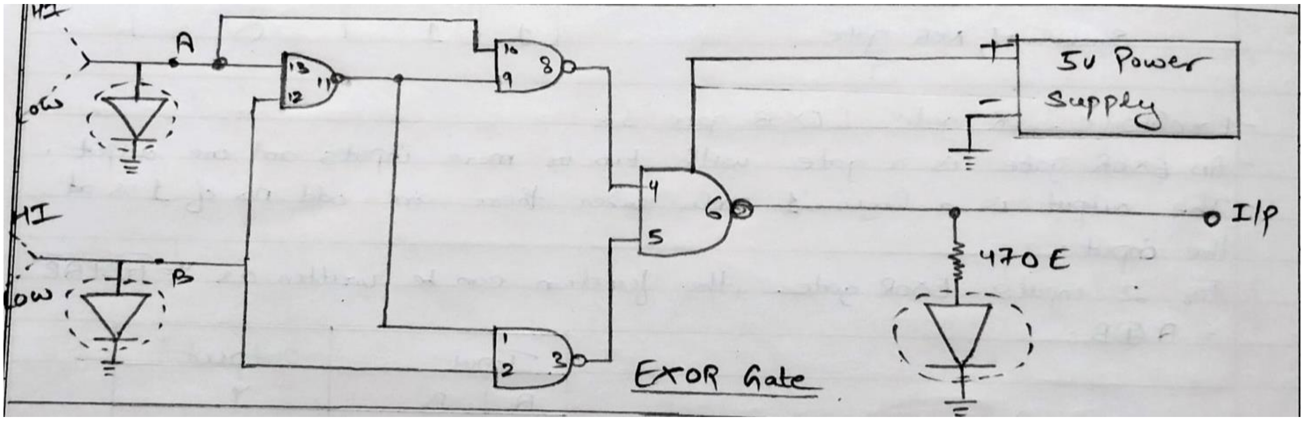
- An EXOR gate is a gate with two or more inputs and one output.
- The output is a logic(1) only when there are odd number of 1's at the input.
- For 2 inputs EXOR gate, the function can be written as Y= (A)'B+A(B)'=A ⊕ B



Truth Table

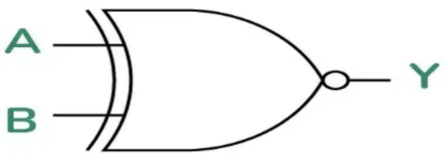


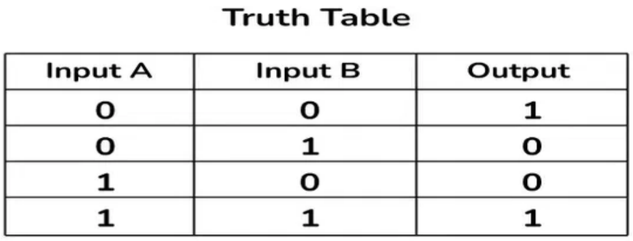
A (Input 1)	B (Input 2)	X = A'B + AB'
0	0	0
0	1	1
1	0	1
1	1	0



Exclusive NOR gate (EXNOR Gate)

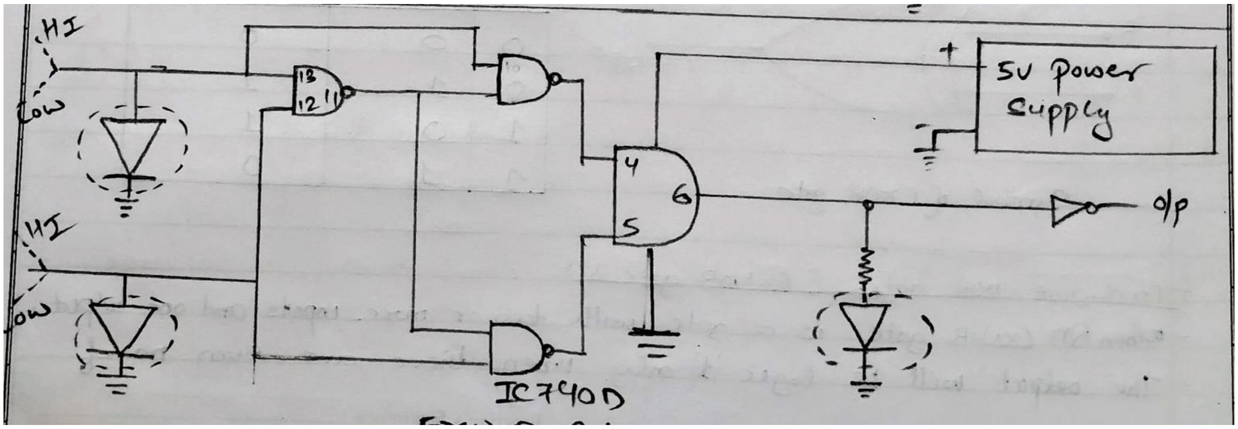
- An EXNOR gate is a gate with two or more inputs and one output.
- The output will be logical(1) only when there are even number of 0's at the input.
- For all other case the output will be zero.
- For 2 input EXNOR gate, the function can be written as $Y=(AB)'+AB= A\odot B$





Truth Table

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



PROCEDURE :-

1) All the logic gates i.e. AND OR NOT NOR NAND EXOR EXNOR gate are identified to the trainer kit.
2) Connections are made as per circuit diagram.
3) The output is observed and the truth table is verified.

Conclusion

AND OR NOT NAND EXOR EXNOR gate is successfully studied on the truth table of the above said gates are verified.

Experiment 2

AIM OF THE EXPERIMENT

Implement various gates by using universal properties of NAND and NOR gates and verify truth table.

APPARATUS REQUIRED

(1) Electrical and Electronics system Trainer.

(2) Digital logic Gate Experiment panel.

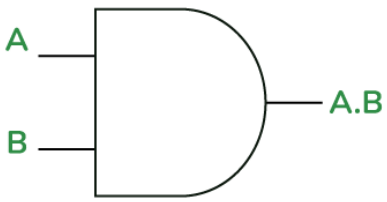
(3) Connecting wires (As per required)

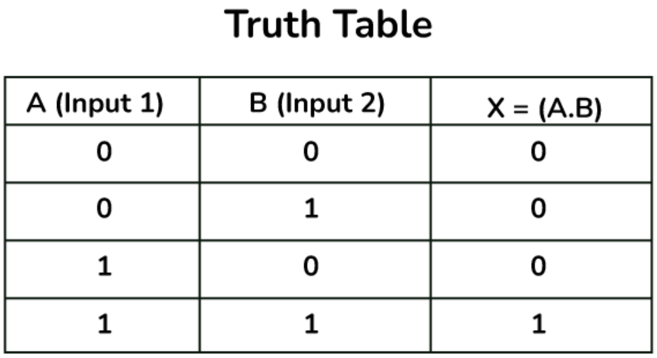
THEORY

AND gate

- The AND gate performs logical multiplication commonly known as AND function.

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





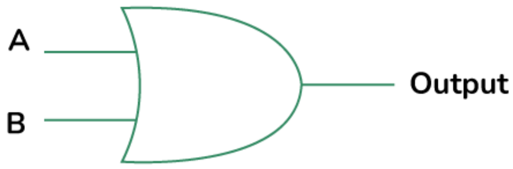
Truth Table

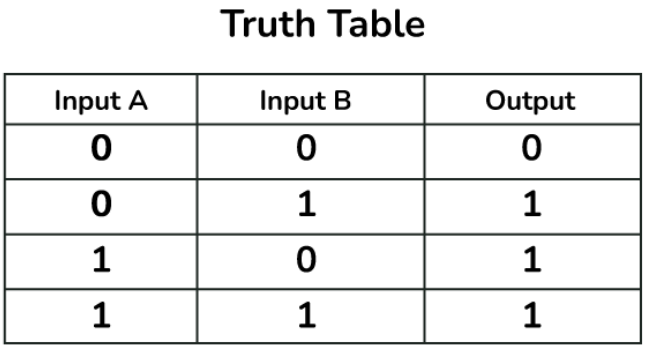
A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

OR gate

-The OR gate perform a logical addition commonly known as OR Function.

- The output is high when anyone of the input is high and the output is low when both the outputs are low.

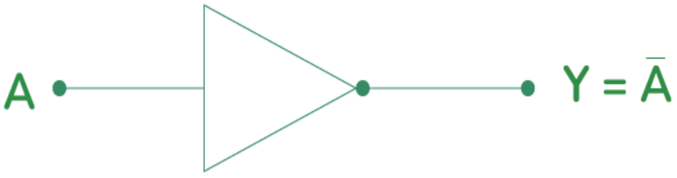


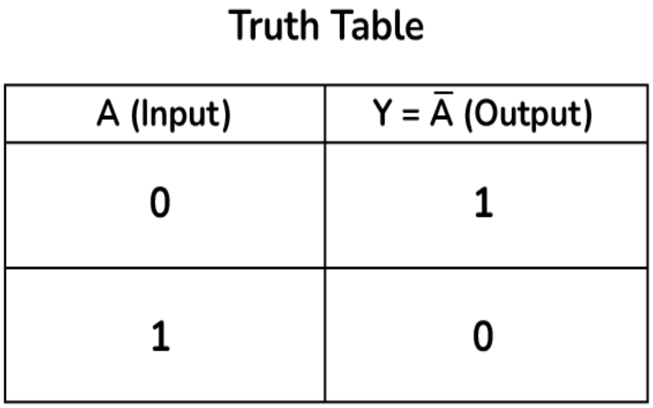

Truth Table

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

NOT gate

- The NOT gate is called on inverter. The output is high when input is low and the output low when the input is high.



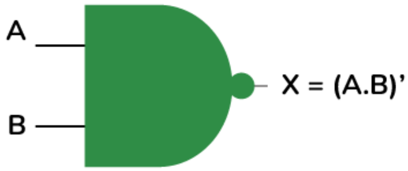

Truth Table

A (Input)	Y = Ā (Output)
0	1
1	0

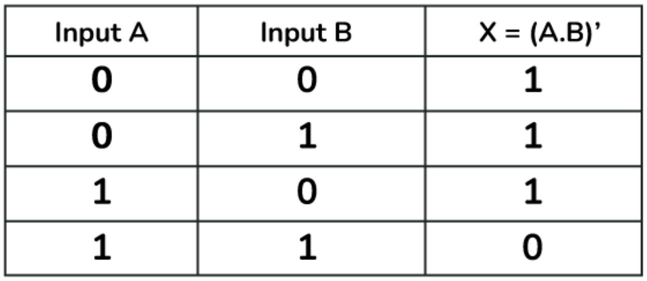
NAND gate

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

- The output is low level when both inputs are high.



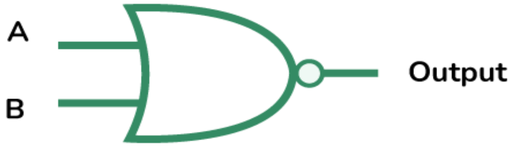
Truth Table



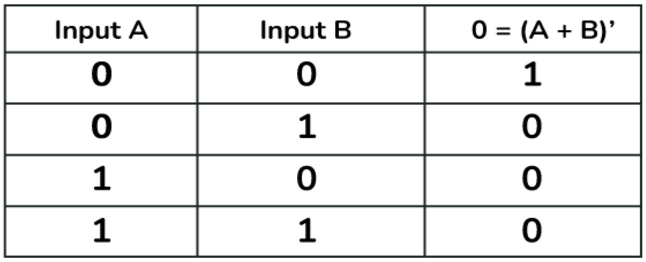
Input A	Input B	X = (A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

NOR GATE

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



Truth Table



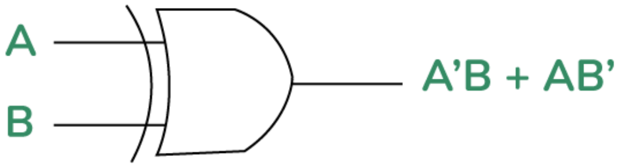
Input A	Input B	0 = (A + B)'
0	0	1
0	1	0
1	0	0
1	1	0

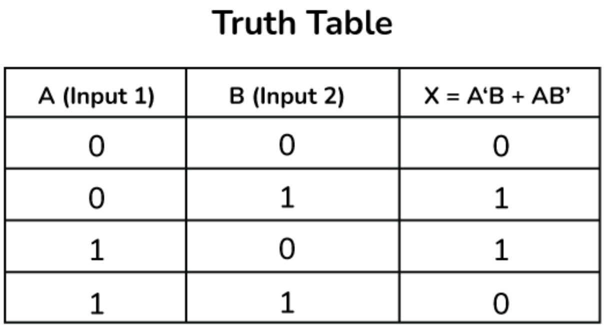
EXCLUSIVE OR gate (EXOR gate)

- An EXOR gate is gate with two or more inputs and one output.

- The output is logic 1 only when there are odd number of 1'S at the input.

- The logic symbol for EXOR gate and truth table is.





Truth Table

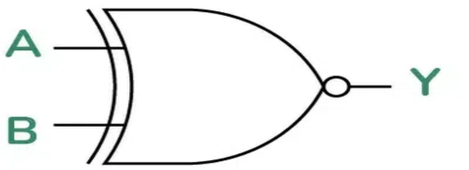
A (Input 1)	B (Input 2)	X = A‘B + AB’
0	0	0
0	1	1
1	0	1
1	1	0

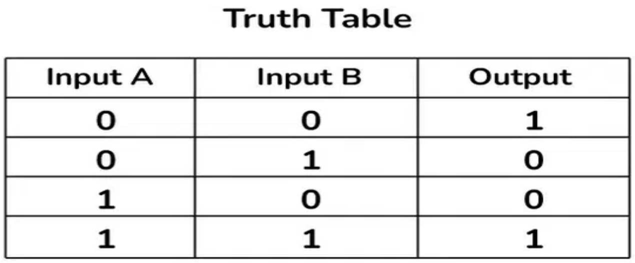
EXCLUSIVE NOR gate (EXNOR gate)

- An EXNOR gate is a gate with two or more inputs and one output.

- The output will be logic '1' only when there are even number of '0' at the input.

- For all the other cases the output will be zero.




Truth Table

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

PROCEDURE

- All the logic gates i.e. AND, OR, NOT, NOR, NAND, XOR, XNOR gates are identified to the trainer ki and were implemented by using Universal NAND and NOR gate.

- Connection is made as per the circuit diagram.

- The output was observed and the truth table was verified.

CONCLUSION

AND, OR, NOT, NAND, NOR, EXOR, EXNOR gates are successfully implemented using NAND and NOR gates and the truth table of the above said gate is verified.

Experiment 3

AIM OF THE EXPERIMENT

Implement half adder and full adder using logic gates.

APPARATUS REQUIRED

1. Electrical and electronics system trainer kit

2. Half adder, full adder experiment panel

3. Connecting wires (As per circuit diagram)

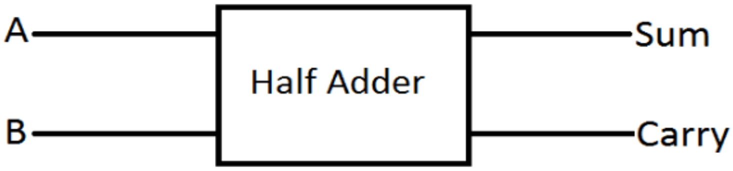
THEORY

HALF ADDER

-It is a combinational and logic circuit having two inputs and two outputs.

- It performs two bit binary addition and produces the sum and carry.

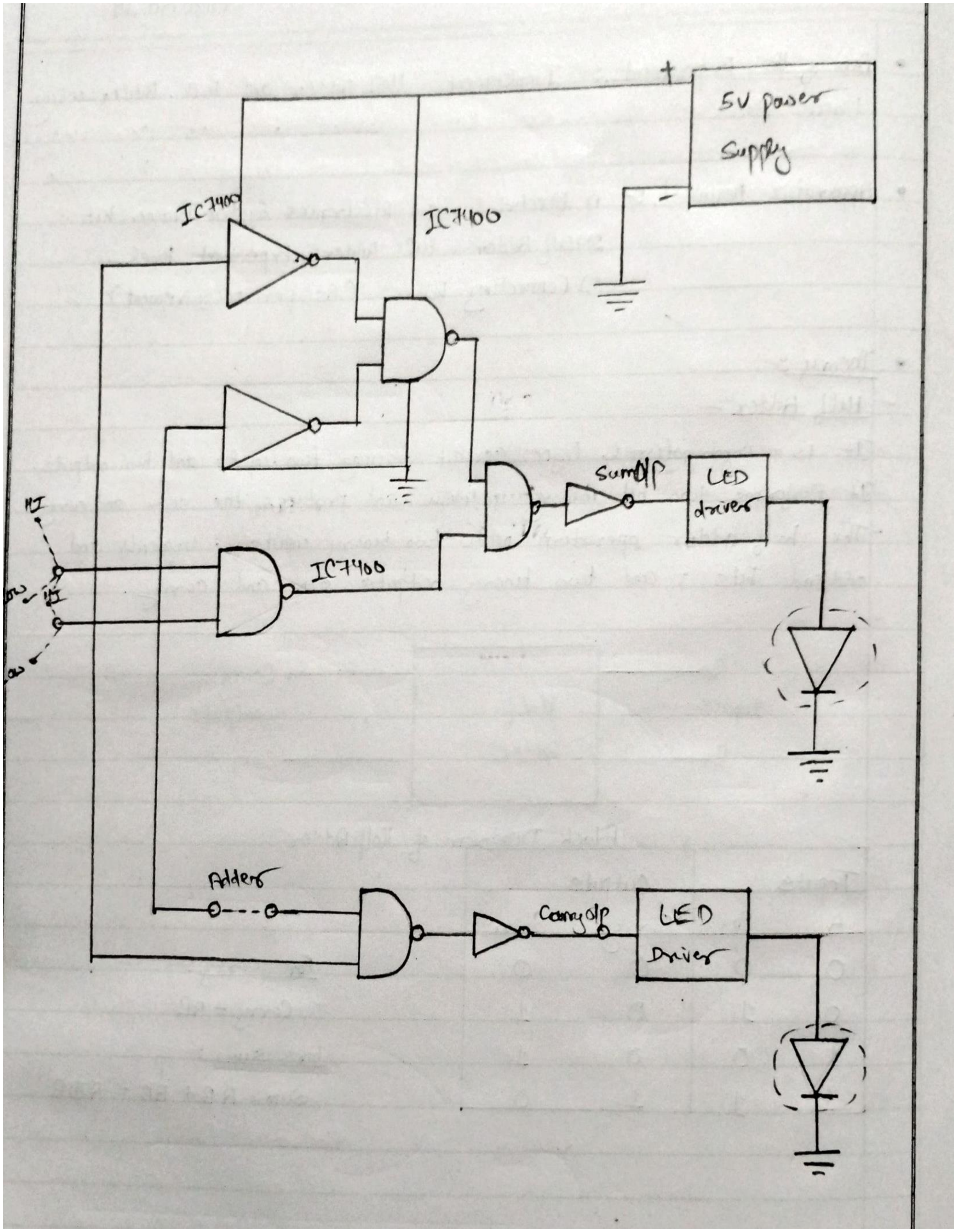
- The half adder operation needs to binary inputs augends and addend bits; and two binary outputs sum and carry.



INPUTS		OUTPUTS	
A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

For carry :- carry=AB

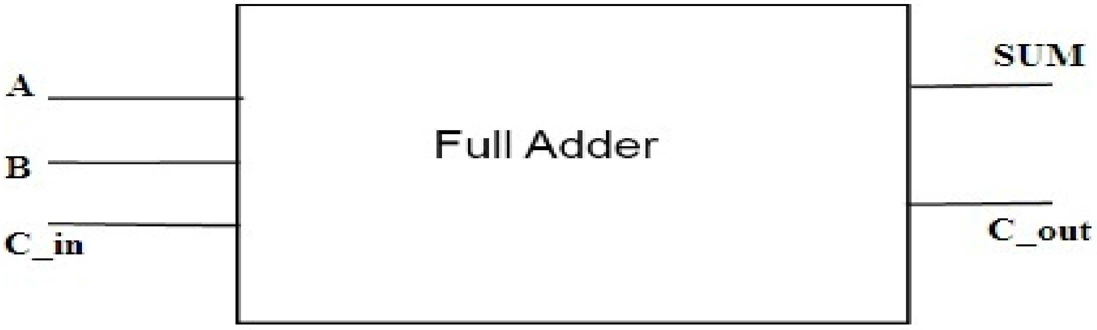
For sum :- sum=A’B+AB’=A ⊕ B



FULL ADDER

- A full adder is a combinational circuit that performs the arithmetic's sum of three inputs bits.

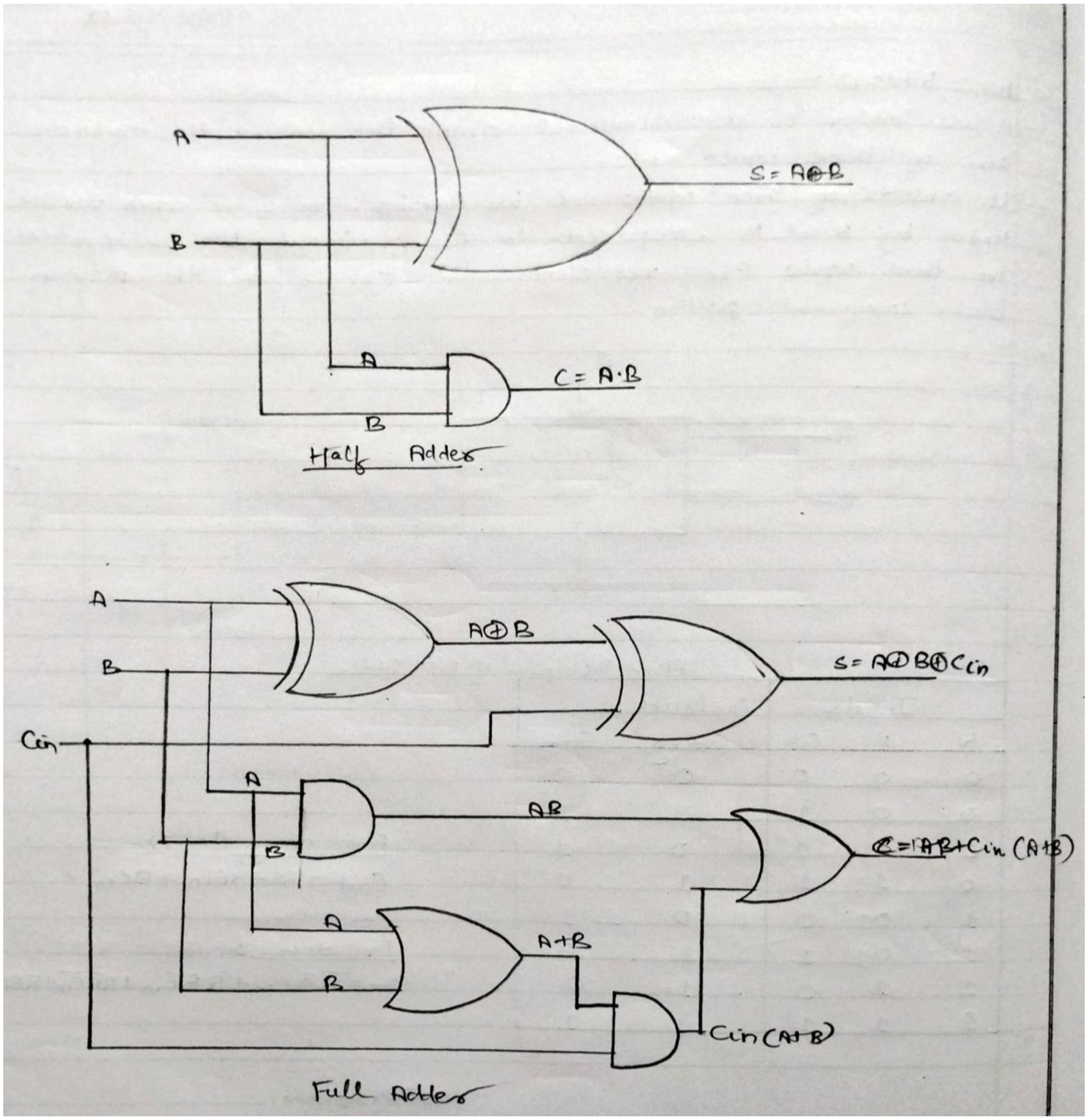
- It consists of three inputs and two outputs. Two of the input variables denoted by A and B, represented the two significant bits to be added. The third inputs Cin represent the carry from the precious lower significant position.



INPUTS			OUTPUTS	
A	B	Cin	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

For carry :- C_{out}=AB+AC_{in}+BC_{in}

For sum :- Sum= A'B'C_{in}+A'BC'$_{in}$+AB'C'$_{in}$+ABC_{in}



PROCEDURE

- Half Adder, Full Adder circuit is identified in the trainer kit.

- Connections are made as per the circuit diagram.

- The supply is switch on and the output for half adder and full adder is observed.

- The truth table of half adder and full adder is verified.

CONCLUSION

Half Adder and Full Adder are constructed and there truth table is verified.

Experiment 4

Aim of the experiment: - Implement half subtractor and full subtractor using the logic gate.

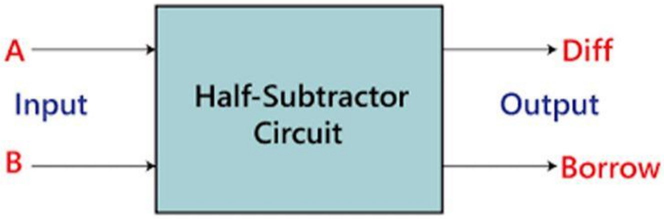
Apparatus required:-

1. Electrical and electronics system trainer kit
2. Half subtractor, Full subtractor experiment panel
3. Connecting wires (as per requirement)

Theory

Half subtractor

- A half subtractor is a combinational circuit that subtracts two binary bits and produces two output, difference (D) and borrow.
- The Boolean functions for two outputs of the half subtractor are D=x'y+xy'=x⊕y



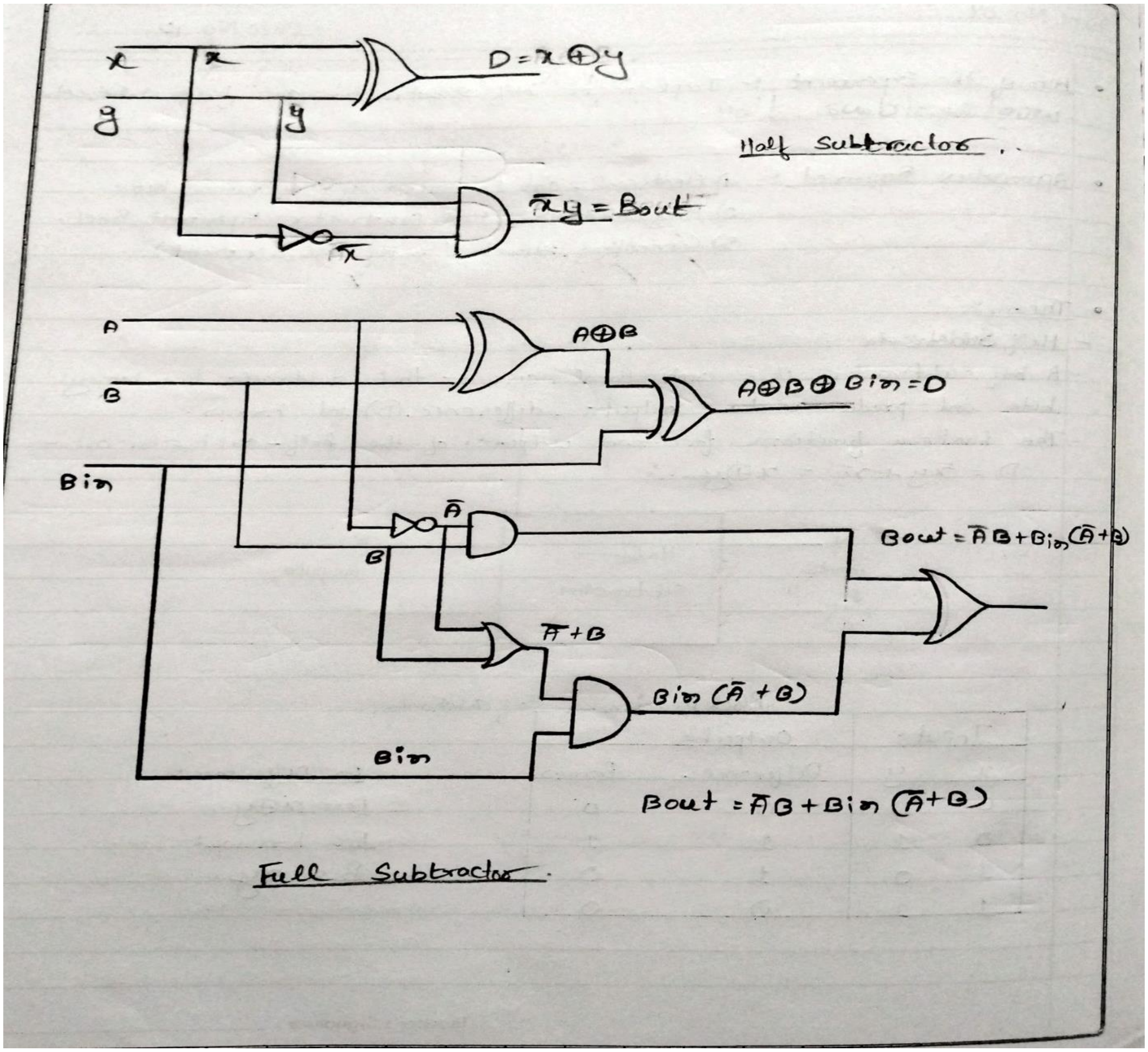
Block diagram of half subtractor

Input		Output	
X	Y	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

For difference:-

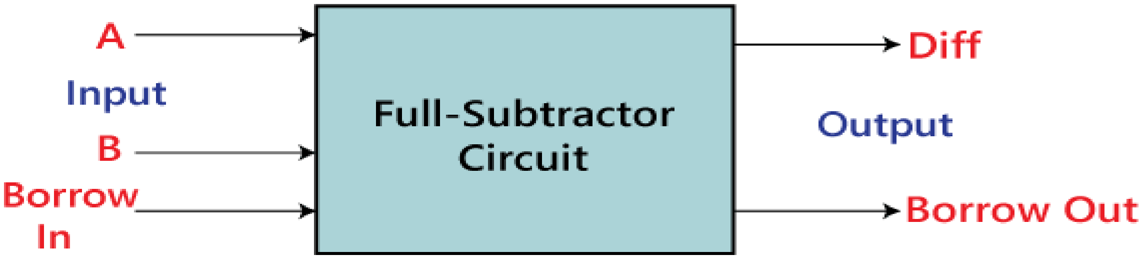
- D= x⊕y

For borrow:-
B=x'y



Full subtractor:-

- A full substroctor is a combinational circuit that performs a subtraction between two bits ,taking into account borrow of the lower significant stage.
- The three input A,B and Bin denotes minuend ,subtrahend and previous borrow responsibly.
- The two output ,D and B out represent the difference and output borrow, respectively .

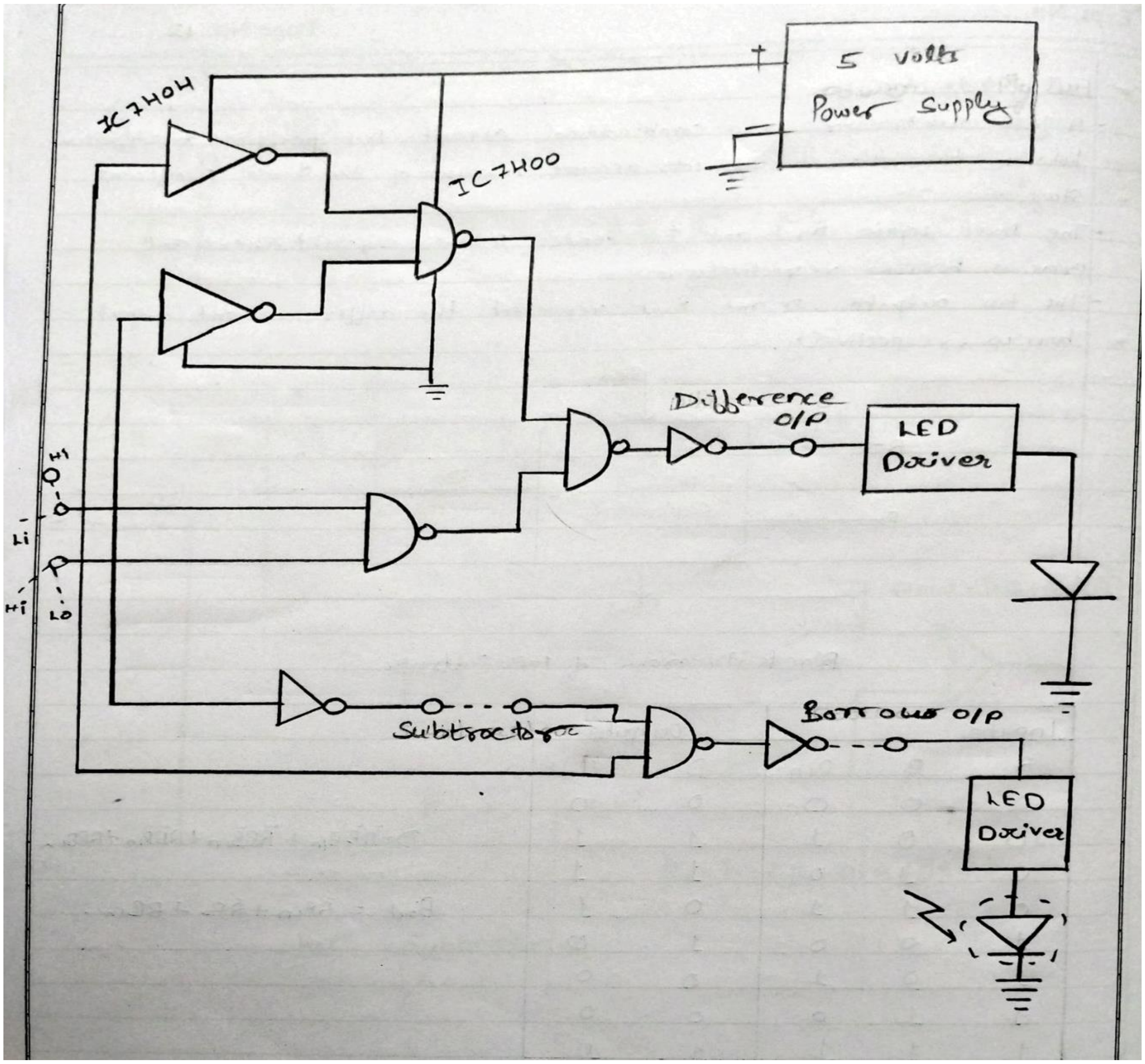


Input	Output

A	B	Bin	D	Bout
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

$D= A'B'B_{in}+A'BB'_{in}+AB'B'_{in}+ABB_{in}$

$B_{out}= A'B_{in}+A'B+BB_{in}$



Procedure:-

1)Half and full subtractor circuit is identified in the trainer kit.
2) connection is made as per circuit diagram.
3) The supply is switched on and the output for half subtractor and full subtractor is observed.
4)The truth table of half subtractor and full subtractor is verified.

Conclusion:-

Half subtractor and full subtractor are constructed and there truth table is verified.

Experiment 5

AIM OF THE EXPERIMENT

To study multiplexer and Demultiplexer.

APPARATUS REQUIRED

(1) Dc power supply -5V

(2) Panel No. P14

(3) Connecting wires (As per required)

Theory

Multiplexer -

-A MUX is a digital switch that has multiple inputs(sources) and a single output (destinations).

- The select lines determine which inputs is connected to the output.

- Multiplexer is many into one and it provides the digital equivalent of an analog selector switch.

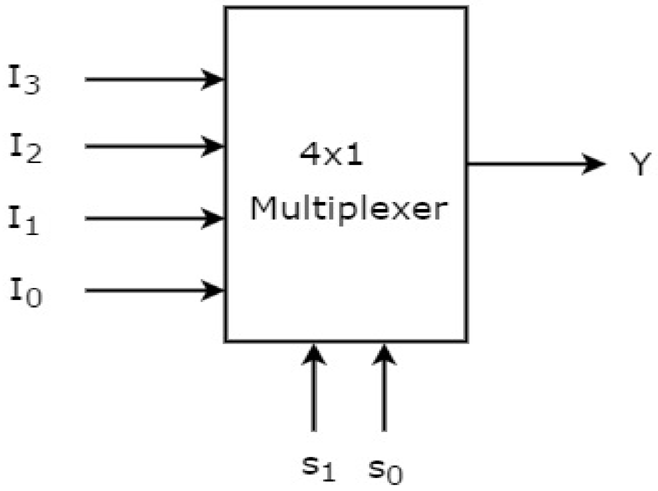
Multiplexer using IC74153-

- It is a dual 4:1 multiplexer IC. It has four inputs in each section and Y_0, Y_1 are the corresponding outputs.

- The G_0 and G_1 are the corresponding active low stable inputs to these sections.

- Select lines S_1 and S_0 are common both sections. IC74352 is a dual 4:1 MUX with output inverted.

- Pin out of IC74352 is same as that of IC74153. IC74153 can be used to move data from a group of registers to a common output bus.



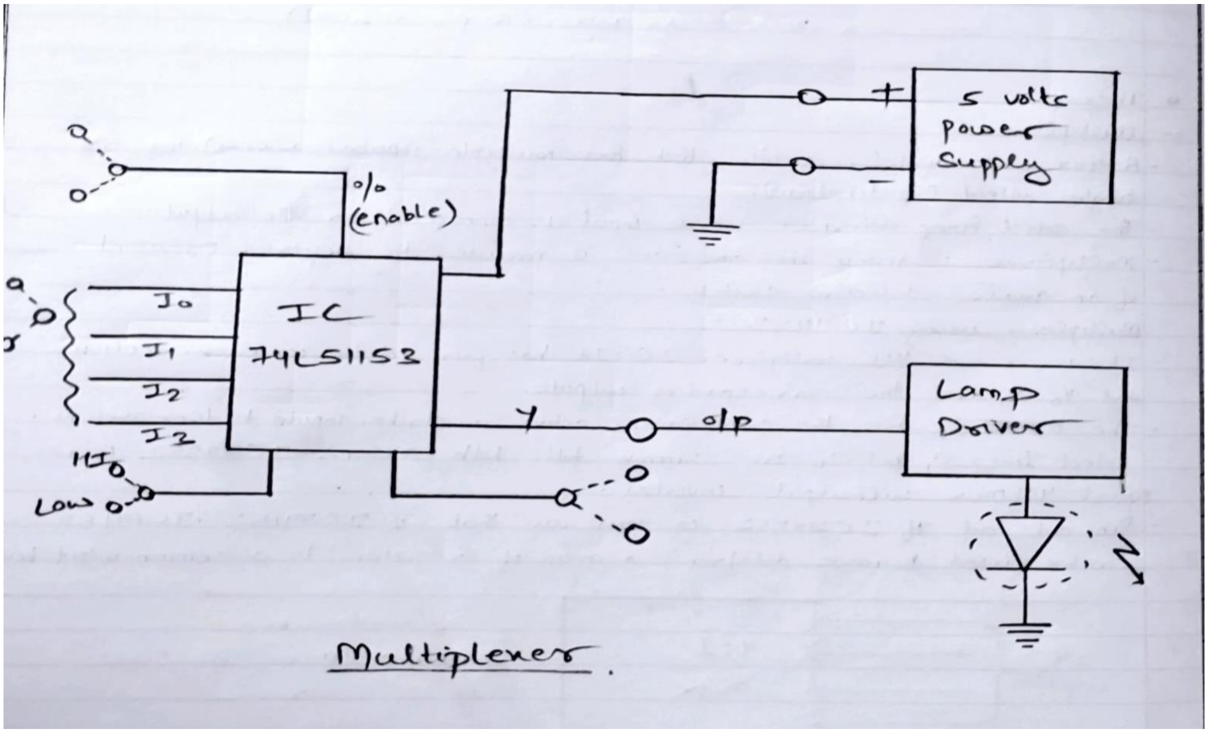
Truth table:-

E’	Selection	Inputs	Outputs
-	S_1	S_0	Y
0	0	0	I_0
0	1	0	I_1
0	0	1	I_2
0	1	1	I_3

Observation Table :-

Selection Inputs			Inputs				Output
S_1	S_0	E_L	I_0	I_1	I_2	I_3	Y
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

L=LOW H=HIGH X= ANY STATE.



PROCEDURE

- IC7415 is used for the experiment. In IC741 is a dual 4 input multiplexer. It has two select inputs, one enable and 4 data inputs.

- Any inputs out of four can be selected by select inputs (S_0 and S_1) . Enable input can be used to store the output.

- In other words any data can be connected to the output using select inputs. S_0 and S1 when enable input is zero.

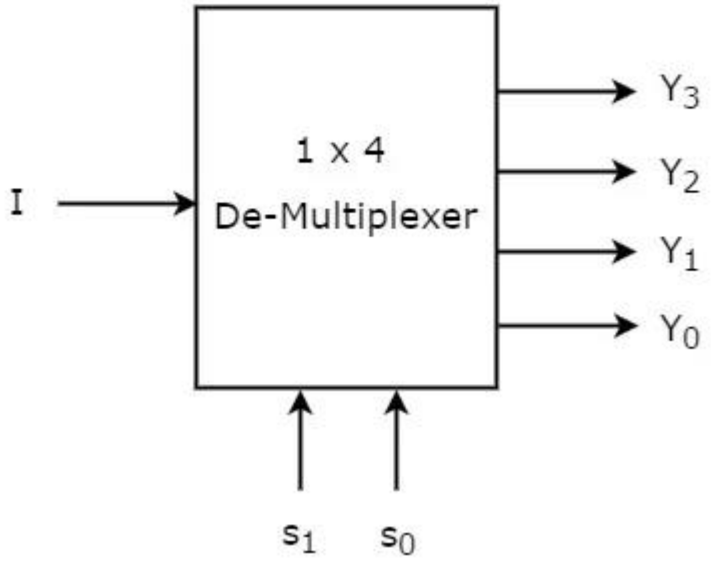
- Connect 5 volt supply to the panel, switch on the supply, Apply various input (select input S_0 S_1 , Data input I_0-I_3 are enable) as given in the table and observe other output in LED. So that LED will indicate the state of output. All inputs may be given through the terminals marked Hi and Lo (4 no. Each) . Verify the truth table.

Demultiplexer:-

- A demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2n possible output lines.

- The selection of specific output lines is controlled by the various values of n selection line.

- The single input variables has a path to all four outputs, but the input information directed to only one of the output lines.

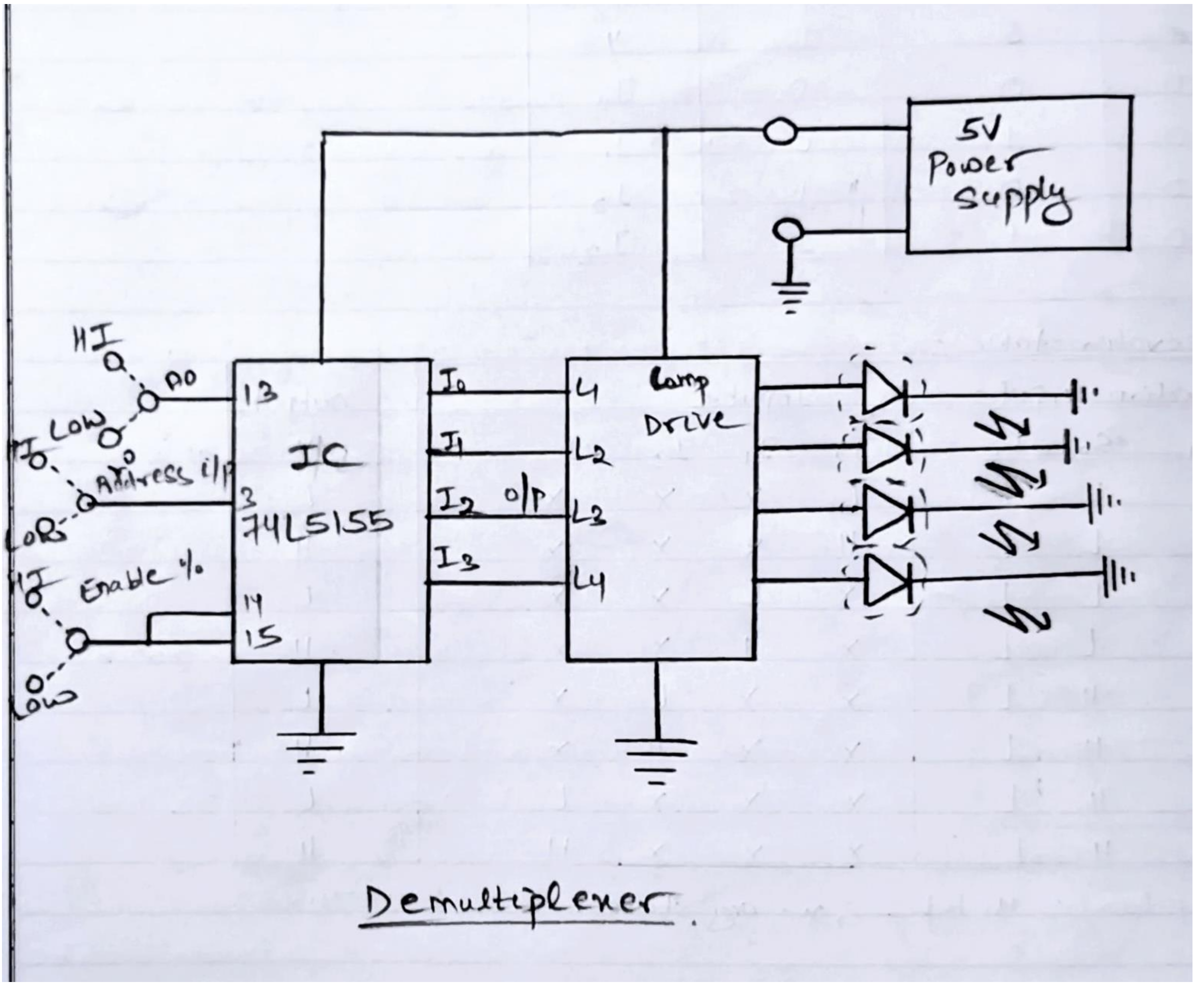


Truth Table :-

S_1	S_0	E’	I_3	I_2	I_1	I_0
0	0	0	0	0	0	D
1	0	0	0	0	D	0
0	1	0	0	D	0	0
1	1	0	D	0	0	0

Observation Table :-

Selection lines		Enable	Output			
S_1	S_0	E’	I_0	I_1	I_2	I_3
L	L	L	L	X	X	X
H	L	L	X	L	X	X
L	H	L	X	X	L	X
H	H	L	X	X	X	L



Demultiplexer

PROCEDURE

- Demultiplexer is a circuit which select the output for its only input i.e. output is selected among many address inputs under enable condition.

- IC used is IC7415155 (Dual 1 of 4 demultiplexer). It has two select lines $_{S0}$ and S_1 which decides the output out of I_0 , I_1 , I_2 and I_3 . Output is low when enable otherwise high.

- Locate the circuit shown on panel no.14, connect 5v supply and apply input conditions as in truth table. Observe output by connecting LEDs and verify the truth table.

CONCLUSION

From the above experiment we concluded that the result of truth table and observation table are same

Experiment 6

Aim of the experiment :-

Study of flip -flops 1) S-R flip -flop 2) J-K flip -flop 3) D flip -flop 4) T flip –flop.

Apparatus required :-

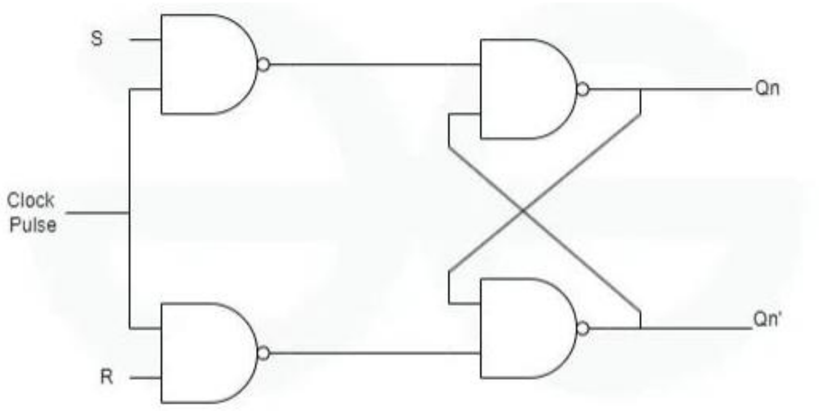
1. External DC power supply +5v
2. Manual pulsar switch
3. Panel no.13

Theory :-

S-R flip -flop -

-The SR latch can be constructed by using either two coupled NAND gates or two cross coupled NOR gates, using NOR gates and active high S-R latch can be constructed and using NAND gates and active low SR can be constructed.

- The name of the latch S-R or SET-RESET designed from the name of its input.



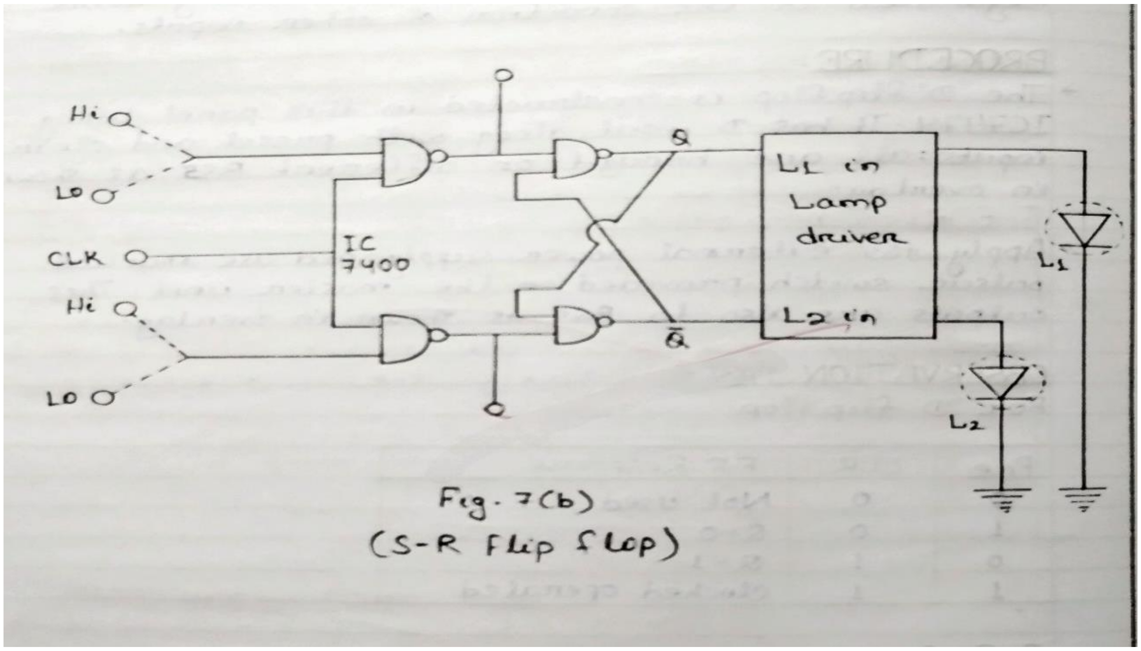
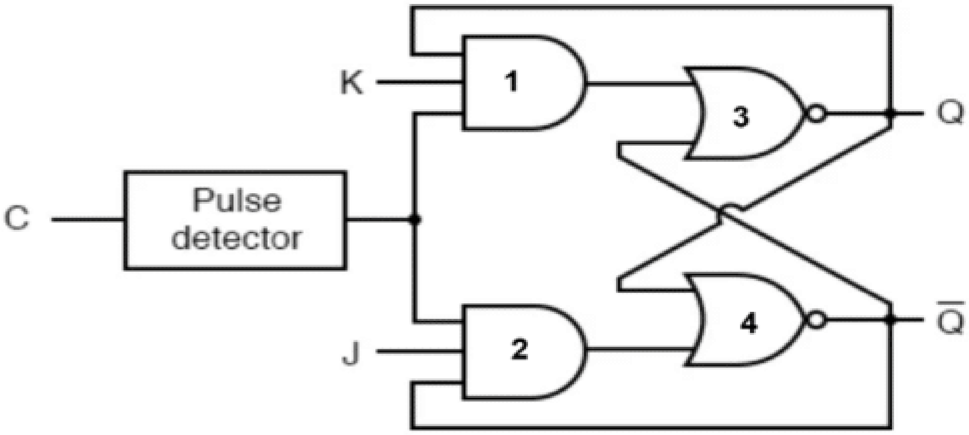


Fig. 7(b)
(S-R Flip flop)

J-K flip -flop -

- The function of J-K flip -flop is identical in that of S-R flip -flop except that, it has no invalid state like that of SR flip -flop.

- The uncertainty in the state of a SR flip -flop when S=R=1 can be eliminated by converting it into a JK flip -flop.



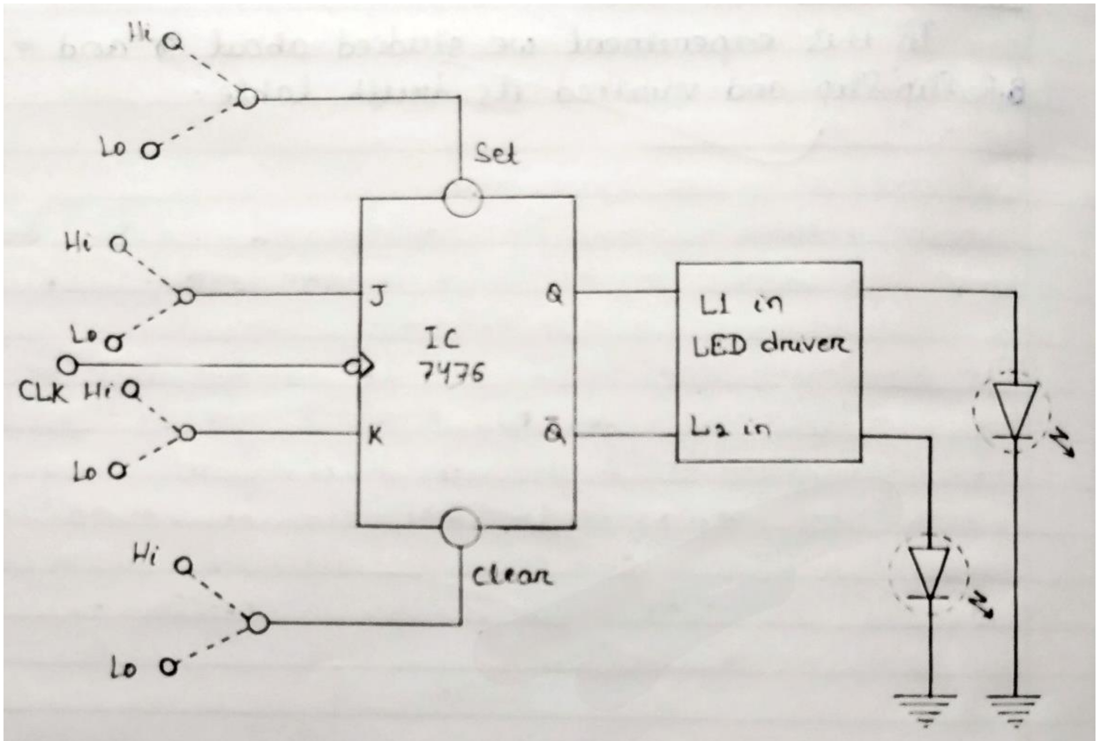
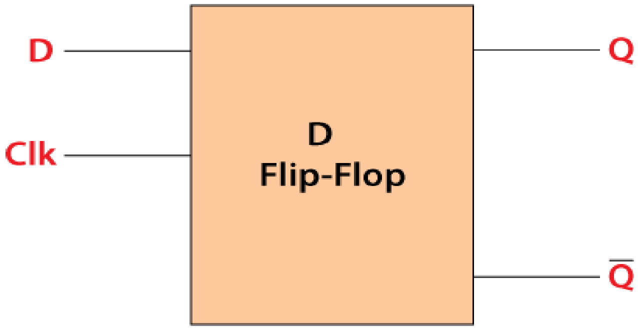


Fig 8(a)
(J-K flipflop)

D flip –flop :-

-The D flip -flop is often called a delay flip -flop. The data (0 or 1) at input D is delayed one clock pulse for getting output Q.

-It has one data input (D) and a clock input (CLK) and the output are labelled as Q and Q'



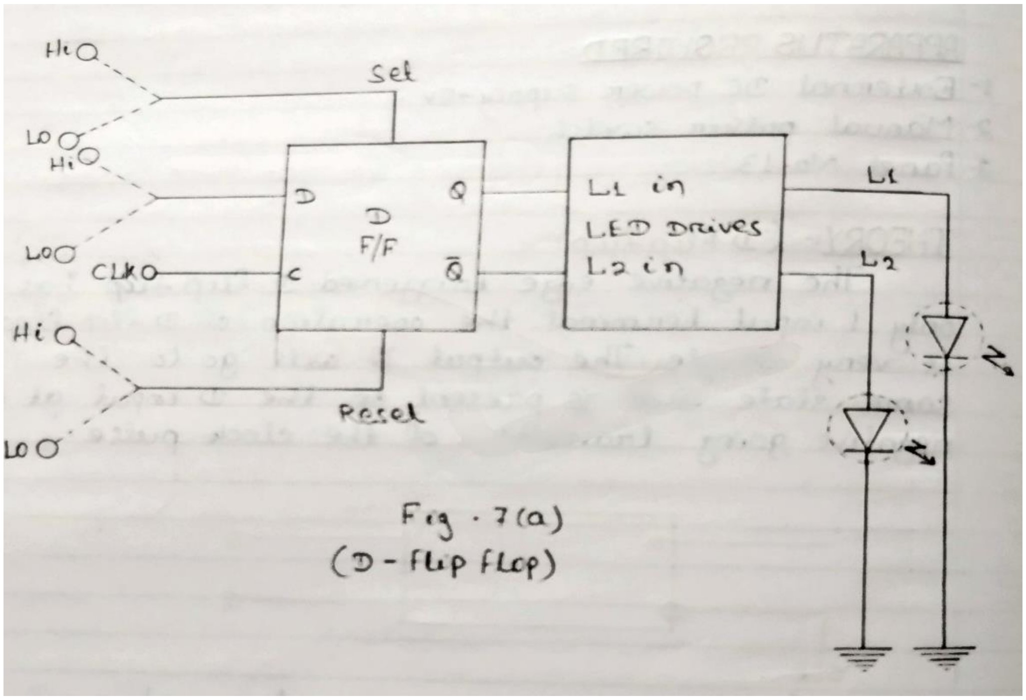
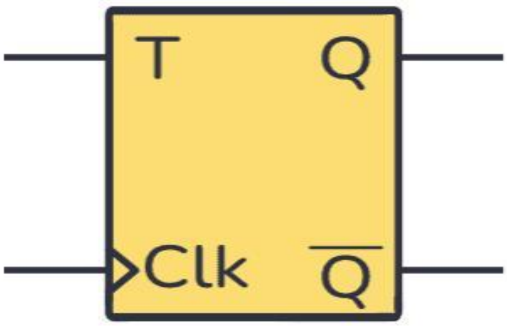


Fig. 7(a)
(D - Flip Flop)

<u>T flip -flop:</u>-

-The T flip -flop has a single control input, labelled T for toggle when T is high, the flip flap toggles on every new clock pulse. When T is low the flip flop remains in whatever state it was before.



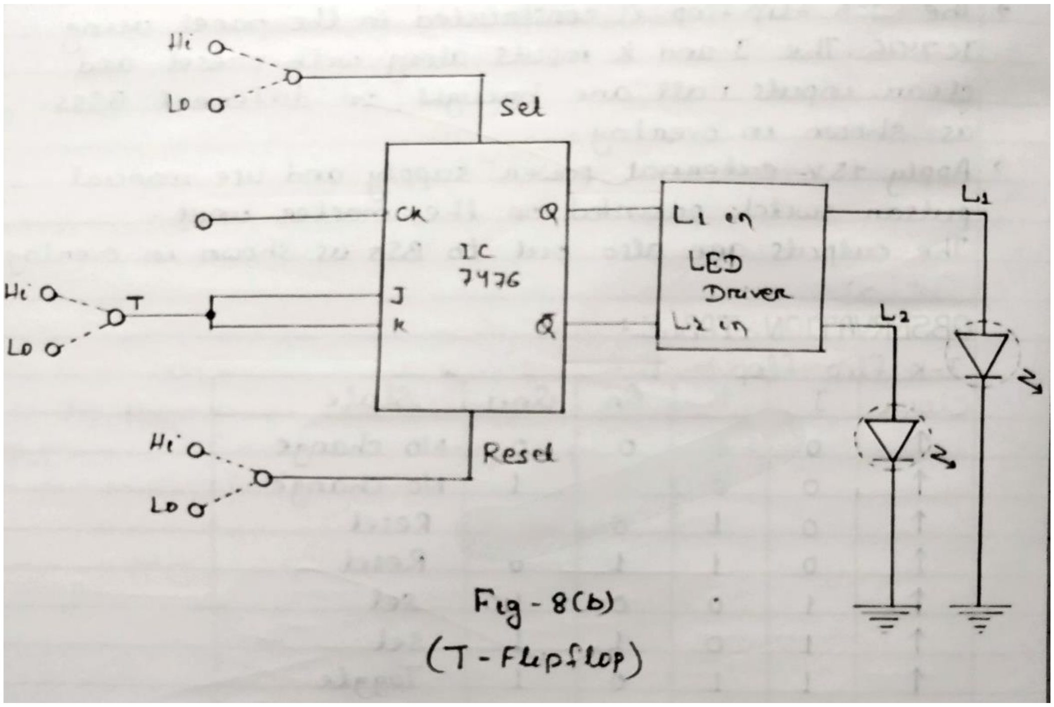


Fig - 8(b)
(T - Flipflop)

Procedure:-

SR flip-flop-

-The S-R flip flop is constructed around IC7400(NAND).

- Make the connection as shown in the circuit diagram.

-Apply +5V external power supply and use manual pulsar switch provided on the master unit.

-The two different input R and S are brought BS5s with clock input at separate BS5.

-The two output are also out to BS5s as shown in overlay.

J-K flip-flop:-

-The JK flip -flop is constructed in the panel using IC7476 .

-The J and K inputs along with present and clear inputs all are brought on different BS5S as shown in overlay.

-Apply +5V external power supply and use manual pulsar switch provided on the master unit.

-The outputs are also out to B35as should in overlay.

D flip -flop:-

-The D flip -flop is constructed in this panel using IC7474 .

-It has D Input along with present and clear inputs, all are brought on different BS5 as shown in overlay.

-Apply +5V external power supply and use manual pulsar switch provided on the master unit. This outputs are also to BS5 as shown in overlay.

T flip flop:-

-The T flip -flop is constructed in this panel using IC7476 JK flip -flop when both of inputs are shorted.

-The he J and K inputs along with present and clear inputs all are brought on different BS5s as shown overlay.

-Apply +5v external power supply and manual pulsar switch provided on the master unit. The outputs are also out to BS5s as shown in overlay.

Observation :-

SR flip flop -

S	**R**	**Qn**	**Qn+1**	**State**
0	0	0	0	No change
0	0	1	1	No change
0	1	0	0	Reset
0	1	1	0	Reset
1	0	0	1	Set
1	0	1	1	Set
1	1	0	X	Invalid
1	1	1	X	Invalid

J-K flip-flop:-

Clock	**J**	**K**	**Q**	**Qn+1**	**State**
↑	0	0	0	0	No change
↑	0	0	1	1	No change
↑	0	1	0	0	Reset
↑	0	1	1	0	Reset
↑	1	0	0	1	Set
↑	1	0	1	1	Set
↑	1	1	0	1	Toggle

↑	1	1	1	0	Toggle
0	X	X	0	0	No change
0	X	X	1	1	No change

D flip - flop:-

Pre	**CLR**	**F.F Response**
0	0	NOT used
1	0	Q=0
0	1	Q=1
1	1	Clock operated

T flip-flop :-

Clock	**T**	**Qn**	**Qn+1**	**State**
↑	0	0	0	No change
↑	0	1	1	No change
↑	1	0	1	Toggle
↑	1	1	0	Toggle
0	X	0	0	No change
0	X	1	1	No change

Conclusion :- In this experiment we studied about SR flip flop, JK flip flop, D flip flop and T flip flop, and verified its truth table.

Experiment 7

Aim of the experiment :- Implement Mode-10 asynchronous counters

Apparatus Required :-

1. Digital trainer kit
2. Decade counter (IC7490)
3. Connecting wires – As required.

Theory:-

- The number of the output states of a counter is called modulus(mod) of the counter and the number of flip flops used in the counter and the way in which they are connected are to determine the number of states.
- The maximum number of flip flops in the counter is N= 2^n , where n is the number of flip flops in the counter.
- The maximum modulus of an n-bit counter is N=2^n , A MOD-n counter counts from 0 to 2^n-1.

Decade Ripple counter:-

- In decade counter the sequence is truncated upto ten states, 0000(0 in decimal) through 1001(9 in decimal). The truncation in the count sequence is achieved by the resetting the counter at particular count instead of going through all of its normal states.
- In case of BCD decade counter, it is to reset back to the 0000 state after the 1001 state, the resetting of counter is done with the help of reset input of each flip-flop.
- These inputs are activated when desired state is reached in case of BCD counter, reset input is activated using NAND gate when 1010 state is reached.

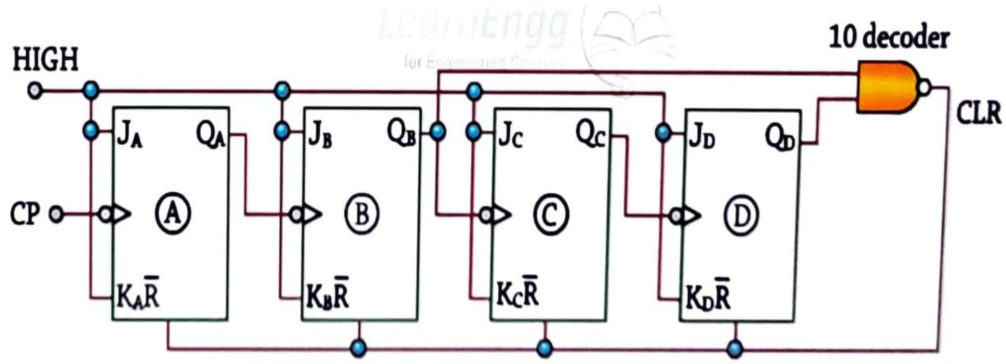


Fig.8: Circuit diagram of decade counter

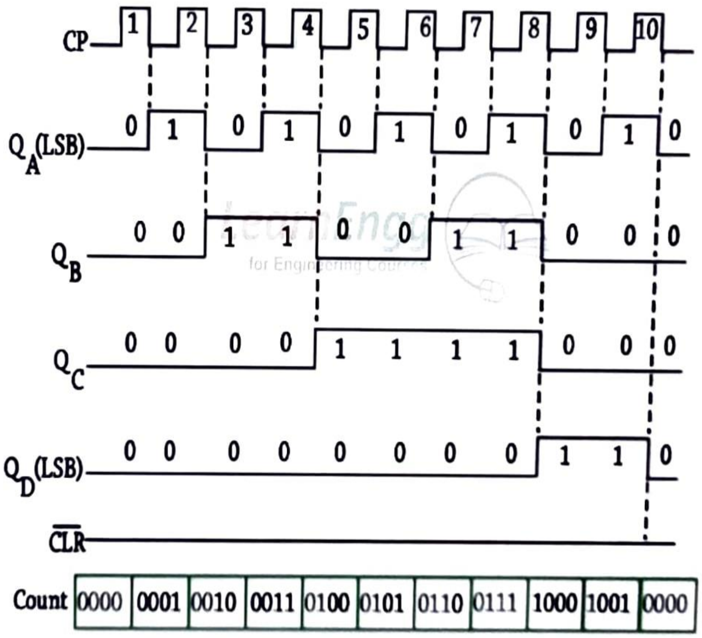


Fig.9: Timing diagram

Design of MOD-10 Ripple counter :-

- The design of mod-10 ripple counter can be explained with the help of timing diagram. From the timing diagram it can be seen that Q_0 is complimented on the negative edge of every clock pulse.
- The external clock is connected to clock input of the first flip-flop and both J & K = 1
- Q_1 is complemented if $Q_3 = 0$ and Q_0 goes from 1 to 0, and Q_1 is to be cleared if $Q_3 = 1$ and Q_0 goes from 1 to 0.
- Hence Q_0 is to be connected as clock input to 2nd flip flop K = 1 and J should become 0 when Q_3 is 1 that is Q_3 is to be connected to J of 2nd flip flop.
- Q_2 must be complimented when Q_1 goes from 1 to 0.
- Hence Q_1 must be connected to clock input of 3rd flip-flop with J=K=1.
- Finally, Q_3 must be complemented when Q_2Q_1 =11 and Q goes from 1 to 0, and must be cleared if either Q_2 or $Q_1 = 0$ and Q_0 goes from 1 to 0.
- Hence 4th flip flop must be clocked with Q_0 with its K=1 and J must be connected from the outpu of the AND gate whose inputs are Q_1 and Q_2.
- By following the above steps, we can design the mod-10 ripple counter without using the asynchronous inputs.

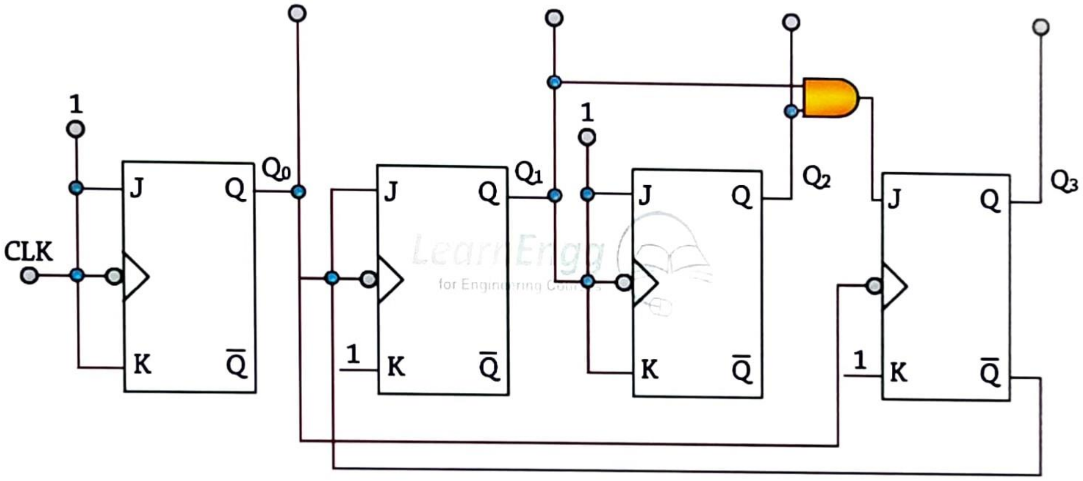
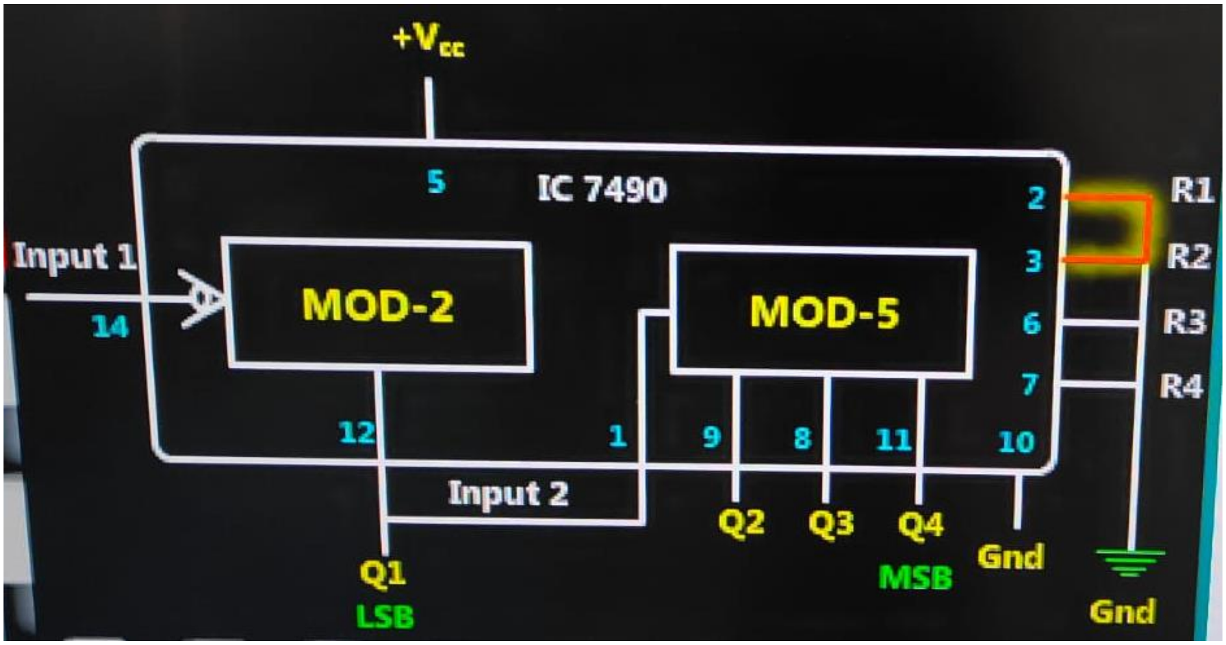


Fig.10: Mod – 10 ripple counter without asynchronous inputs

Procedure :-

1. Make the connections as per the circuit diagram shown in the figure.



Conclusion :-

From the above experiment we implemented Mode-10 asynchronous counters successfully.

Experiment-8

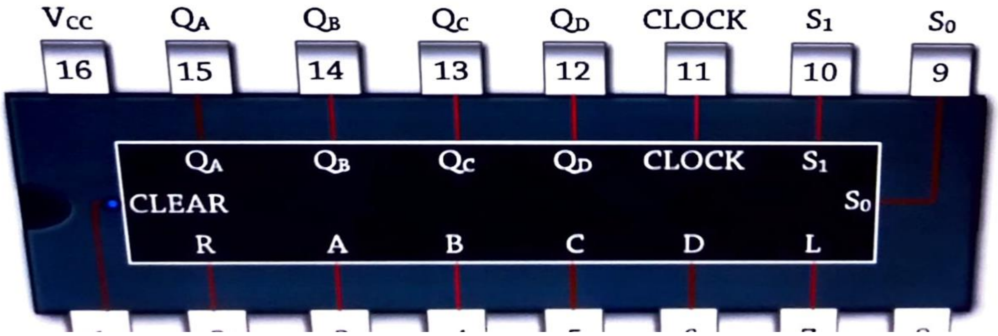
Aim of the experiment:- Study shift registers

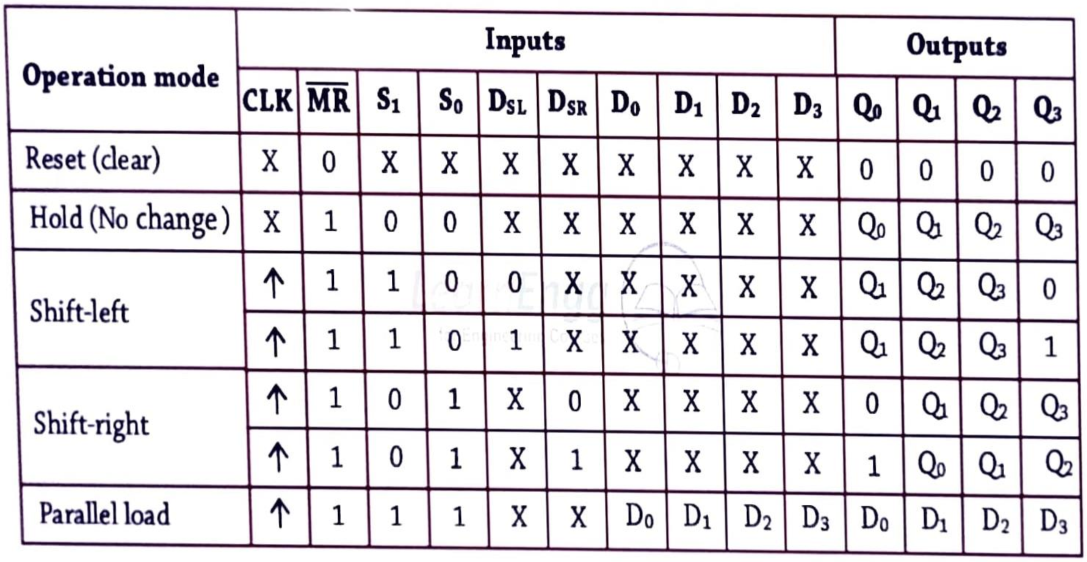
Apparatus Required:-

1. Digital trainer kit
2. IC74194
3. Connecting wires – as required

Theory:-

- The binary information data in a register can be moved from stage to stage within the register o into or out of the register upon application of clock pulse.
- This type of bit movement or shifting is a sensual for certain arithmetic and logical operations used in microprocessors.
- This gave rise to a group of registers called shift registers. They are very important in applications involving the storage and transfer of data in a digital system.
- The bidirectional shift register is designed to incorporate virtually all of the features. A system designer may want in a shift register.
- It features parallel inputs, parallel outputs, right shift and left shift serial inputs, operating mode control inputs, and a direct overriding clear line.
- The register has four distinct mode of operation: parallel (broadside) load, shift right (in the direction Q_A towards Q_D); shift left; inhibit clock (do nothing).
- Synchronous parallel loading is accomplished by applying the four bits of data and taking both mode control imports, S_0 and S_1 high.
- The data are loaded into their respective flip-flops and appear at the output after the positive transition of the clock input.
- During loading, serial data flow is inhibited, shift right is accomplished synchronously with the rising each of the clock pulse when S_0 is high and S_1 is low.
- Serial data for this mode is entered at the shift right data input. When S_0 is low and S_1 is high, Data shifts left synchronously and new data is entered at the shift left serial inputs.
- Clocking off the flip is inhibited when both mode control inputs are low. The mode control inputs should be changed only when the clock inputs is high.



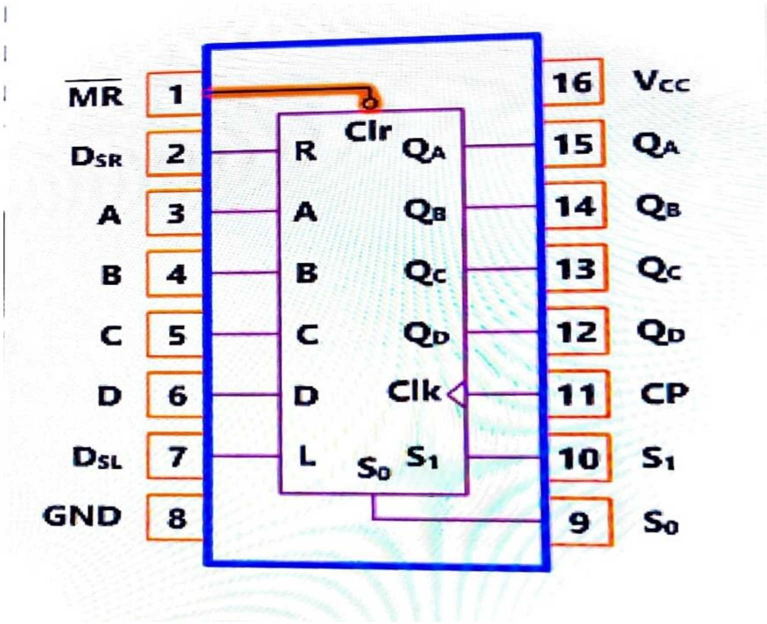


Operation mode	Inputs										Outputs			
	CLK	$\overline{MR}$	S_1	S_0	D_{SL}	D_{SR}	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3
Reset (clear)	X	0	X	X	X	X	X	X	X	X	0	0	0	0
Hold (No change)	X	1	0	0	X	X	X	X	X	X	Q_0	Q_1	Q_2	Q_3
Shift-left	↑	1	1	0	0	X	X	X	X	X	Q_1	Q_2	Q_3	0
	↑	1	1	0	1	X	X	X	X	X	Q_1	Q_2	Q_3	1
Shift-right	↑	1	0	1	X	0	X	X	X	X	0	Q_1	Q_2	Q_3
	↑	1	0	1	X	1	X	X	X	X	1	Q_0	Q_1	Q_2
Parallel load	↑	1	1	1	X	X	D_0	D_1	D_2	D_3	D_0	D_1	D_2	D_3

Fig.2. Truth table for shift register

<u>Procedure</u> :-

1. Make the connections as per the circuit diagram shown in the figure.



<u>Conclusion</u> :- From the above experiment we studied shift register successfully.

Aim of the experiment :- To write programs to find out 1’s complement and 2’s complement of an 8 bit number.

Apparatus required :-

8085 microprocessor kit.

Theory :-

PROGRAM (1’s Complement) -

Memory Address	**Machine Codes**	**Mnemonics**	**Operand**	**Comments**
2000	3A, 01, 35	LDA	2501	Get data in accumulator
2003	2F	CMA		Take it’s 1’s complement
2004	32, 02, 25	STA	2502	Store result in 2502H
2007	76	HLT		Halt/Stop

Program for 1’s complement -

96 H = 10010110
1’s complement = 01101001 = 69 H

Data -
2501 – 96 H

Result -
2502 – 69 H

PROGRAM (2’s Complement) -

Memory Address	**Machine Codes**	**Mnemonics**	**Operand**	**Comments**
2000	3A, 01, 35	LDA	2501H	Get data in accumulator
2003	2F	CMA	-	Take it’s 1’s complement
2004	3C	INR	A	Take its 2’s Complement

2005	32, 02, 25	STA	2502H	Store result in 2502H
2008	76	HLT	-	Halt/Stop

Program for 2’s complement -

96 H = 10010110

1’s complement = 01101001 = 69 H
+00000001
01101011 = 6A

Data -
2501 – 96 H

Result -
2502 – 6 AH

Conclusion :- From the above experiment, we have found the 1’s and 2’s complement of an 8 bit number by using 8085 microprocessor kit.

Experiment 9

Aim of the experiment :- To write programs to find out 1’s complement and 2’s complement of an 8 bit number.

Apparatus required :-

8085 microprocessor kit.

Theory :-

PROGRAM (1’s Complement) -

Memory Address	**Machine Codes**	**Mnemonics**	**Operand**	**Comments**
2000	3A, 01, 35	LDA	2501	Get data in accumulator
2003	2F	CMA		Take it’s 1’s complement
2004	32, 02, 25	STA	2502	Store result in 2502H
2007	76	HLT		Halt/Stop

Program for 1’s complement -

96 H = 10010110
1’s complement = 01101001 = 69 H

Data -
2501 – 96 H

Result -
2502 – 69 H

PROGRAM (2’s Complement) -

Memory Address	**Machine Codes**	**Mnemonics**	**Operand**	**Comments**
2000	3A, 01, 35	LDA	2501H	Get data in accumulator
2003	2F	CMA	-	Take it’s 1’s complement
2004	3C	INR	A	Take its 2’s Complement

2005	32, 02, 25	STA	2502H	Store result in 2502H
2008	76	HLT	-	Halt/Stop

Program for 2’s complement -

96 H = 10010110

1’s complement = 01101001 = 69 H
+00000001
01101011 = 6A

Data -
2501 – 96 H

Result -
2502 – 6 AH

Conclusion :- From the above experiment, we have found the 1’s and 2’s complement of an 8 bit number by using 8085 microprocessor kit.

Experiment-10

Aim of the experiment :- To write programs for finding out addition operation and subtraction operation of 8 bit numbers.

Apparatus required :-

8085 microprocessor kit.

Theory :-

PROGRAM (Addition)-

Memory Address	Machine Codes	Mnemonics	Operand	Comments
2000	21.01.25	LXI	H, 2501H	Get address at 1st no. in HL pair.
2003	7E	MOV	A, M	Move 1st number in accumulator.
2004	23	INX	H	Increment the content of H-L pair
2005	86	ADD	M	Add 1st & 2nd number
2006	32.03.25	STA	2503H	Store the result in 2503H
2009	76	HLT	-	Stop

Data -
2501 H – 49 H

Result -
2503 H – 9 FH

PROGRAM (Subtraction) :-

Memory Address	Machine Codes	Mnemonics	Operand	Comments
2000	21.01.25	LXI	H, 2501H	Get address at 1st no. in HL pair.
2003	7E	MOV	A, M	Move 1st number in accumulator.
2004	23	INX	H	Increment the content of H-L pair
2005	96	SUB	M	Subtract 1st & 2nd number

2006	23	INX	H	Content of H-L pair incremented to 2503 H
2007	77	STA	2503H	Store the result in 2503H
2008	76	HLT		Stop

Data -
2501 – 49 H
2502 – 32 H

Result -
2503 – 17 H

Conclusion :- From the above experiment, we have found addition and subtraction operation of 8 bit numbers.

Experiment -11

Aim of the experiment :- To write program for finding 1’s and 2’s complement of a 16-bit number.

Apparatus required :-

8085 microprocessor kit.

Theory :-

PROGRAM (1’s Complement) -

Find 1’s complement of 5485 H.

The number in the binary form can be represented as follows:

5485 = 0101 0100 1000 0101

1’s complement = 1010 1011 0111 1010 = AB7A H

The number is in the memory locations 2501H and 2502H.

The result is to be stored in the memory location 2503H and 2504H.

Memory Address	Machine Codes	Mnemonics	Operand	Comments
2000	21, 01, 25	LXI	H, 2501 H	Address of LSBs of the number
2003	7E	MOV	A, M	8 LSBs of the number in accumulator
2004	2F	CMA	-	Complement of t8 LSBs of the number
2005	32, 03, 25	STA	2503 H	Store 8 LSBs of the result
2008	23	INX	H	Address of 8 MSBs of the number MSBs
2009	7E	MOV	A, M	8 MSBs of the number in accumulator
200A	2F	CMA	-	Complement of 8 MSBs of the number
200B	32, 04, 25	STA	2504	Store 8 MSBs of the result
200E	76	HLT		Halt

Data -
2501 – 85 H, LSBs of the number.

2502 – 54 H, MSBs of the number.

Result -
2503 – 7 A H, LSBs of the result.

2504 – AB H, MSBs of the result.

PROGRAM (2's Complement) -

Find 2's complement of 5B8C

5B8C = 0101 1011 1000 1100

1's complement = 1010 0100 0111 0011 = A473

2's complement = 1010 0100 0111 0100 = A474

2's complement of a number is obtained by adding 1 to the 1's complement of the number.

The number is stored in the memory location, 2501 and 2502H.

The result is to be stored in the memory location 2503 and 2504H.

Memory Address	**Machine Codes**	**Mnemonics**	**Operand**	**Comments**
2000	21, 01, 25	LXI	H, 2501 H	Address of 8 LSBs of the number
2003	06, 00	MVI	B, 00	Use register B to store carry
2005	7E	MOV	A, M	8 LSBs in accumulator
2006	2F	CMA		1's complement of 8 LSBs of the number.
2007	C6, 01	ADI	01	2's complement of 8 LSBs of the number
2009	32, 03, 25	STA	2503 H	Store 8 LSBs of the result
200C	D2, 10, 20	JNC	GO	
200F	04	INR	B	Store carry
2010	23	INX	H	Address of 8 MSBs of the number

2011	7E	MOV	A, M	8 MSBs in accumulator
2012	2F	CMA		1’s complement of 8 MSBs of the number
2013	80	ADD	B	Add carry
2014	32, 04, 25	STA	2504 H	Store 8 MSBs of the result
2017	76	HLT		Stop

Data -
2501 – 8C, LSBs of the number.

2502 – 5B, MSBs of the number.

Result -
2503 – 74, LSBs of the result.

2504 – A4, MSBs of the result.

2’s complement of the number is A474.

Conclusion :- From the above experiment, we have found the 1’s and 2’s complement of an 16-bit number by using 8085 microprocessor kit.

Experiment 12

Aim of the experiment :- To write program for finding binary addition and decimal addition of two 8-bi numbers; sum: 16- bits.

Apparatus required :-

8085 microprocessor kit.

Theory :-

PROGRAM (Binary addition)-

Add 98 H and 9A H.

SUM = 01, 32 H.

The 1st Number 90 8H is in the memory location 2501H.

The 2nd number 9A H is in the memory location 2502H.

The results are to be stored in 2503 and 2504H.

Numbers are represented in hexadecimal.

Memory address	Machine codes	Mnemonics	Operands	Comments
2000	21,01,25	LXI	H,2501 H	Address of 1st number in H-L pair
2003	0E,00	MVI	C,00	MSBs of sum in register C. initial value=00.
2005	7E	MOV	A, M	1st number in accumulator
2006	23	INX	H	Address of 2nd number 2502 in H-L pair.
2007	86	ADD	M	1st number +2nd number.
2008	D2,0C,20	JNC	AHEAD	Is carry? No, go to the label AHEAD.
200B	0C	INR	C	Yes, increment C.
200C	32,03,25	STA	2503 H	LSBs of sum in 2503 H
200F	79	MOV	A, C	MSBs of sum in accumulator.
2010	32,04,25	STA	2504H	MSBs of sum in 2504 H.
2013	76	HLT		Halt

Data -
2501- 98 H
2502- 9 A H

Result -
2503- 32 H, LSBs of sum.
2504- 01 H, LSBs of sum.

PROGRAM (Decimal addition)-

Add 84 D and 75 D, D stands for decimal number.

Sum = 159 = 01, 59 D

59 is the LSDs of the sum.

01 is the MSDs of the sum.

The first number 84 D is in the memory location 2501 H.

The second number 75 D is in the memory location 2502H.

The sum is to be stored in 2503 and 2504 H.

Memory address	**Machine codes**	**Mnemonics**	**Operands**	**Comments**
2000	21,01,25	LXI	H,2501 H	Address of 1st number in H-L pair
2003	0E,00	MVI	C,00	MSDs of sum in register C. initial value=00.
2005	7E	MOV	A, M	1st number in accumulator
2006	23	INX	H	Address of 2nd number 2502 in H-L pair.
2007	86	ADD	M	1st number +2nd number
2008	27	DAA		Decimal adjust.
2009	D2,0D,20	JNC	AHEAD	Is carry? No, go to the label AHEAD.
200C	0C	INC	C	Yes, increment C.
200D	32,03,25	STA	2503 H	LSDs of sum in 2503 H.
2010	79	MOV	A, C	MSDs of sum in accumulator.
2011	32,04,25	STAHLT	2504H	MSDs of sum in 2504 H.
2014	76	HLT		

Data-
2501 – 84 D

2502 – 75 D

Result-
2503 – 59 D, LSDs of the sum.

2504 – 01 D, MSDs of the sum.

Conclusion :- From the above experiment, we have found the binary addition and decimal addition of two 8-bit numbers; sum: 16- bits by using 8085 microprocessor kit.

Experiment-13

Aim of the experiment :- To write program for finding larger of two numbers and largest number in a data array.

Apparatus required :-

8085 microprocessor kit.

Theory :-

PROGRAM (larger of two numbers)-

Find the larger of 98 H and 87 H.

The first number 90 8H is placed in the memory location 2501H.

The second number 80 7H is placed in the memory location 2502H.

The result is stored in memory location 2503H.

Memory address	**Machine codes**	**Mnemonics**	**Operands**	**Comments**
2000	21,01,25	LXI	H,2501 H	Address of 1st number in H-L pair
2003	7E	MOV	A, M	1st number in accumulator
2004	23	INX	H	Address of 2nd number in H-L pair.
2005	BE	CMP	M	Compare 2nd number with 1st number. IS the 2nd number > 1st ?
2006	D2, 0A, 20	JNC	AHEAD	No, larger number is in accumulator. Go to AHEAD
2009	7E	MOV	A, M	Yes, get 2nd number in accumulator
200A	32, 03, 25	STA	2503 H	Store larger number in 2503 H.
200D	76	HLT		Stop

Data -
2501 – 98 H

2502 – 87 H

Result is 98 H and it is stored in memory location 2503 H.

2503 – 98 H

PROGRAM(largest number in a data array)-

The number in a series are; 98, 75, and 99.

As there are three numbers in the series, the count = 03. The count is placed in the memory location 2500H. The numbers are placed in memory location 2501 to 2503 H. The result is to be stored in memory location 2450 H.

Memory address	**Machine codes**	**Mnemonics**	**Operands**	**Comments**
2000	21,00,25	LXI	H,2500 H	Address of 1st number in H-L pair
2003	4E	MOV	C, M	Count in register C
2004	23	INX	H	Address of 1st number in H-L pair
2005	7E	MOV	A, M	1st number in accumulator
2006	0D	DCR	C	Decrement count
2007	23	INX	H	Address of next number
2008	BE	CMP	M	Compare next number with previous maximum. Is next number > previous maximum ?
2009	D2, 0D, 20	JNC	AHEAD	No, larger number is in accumulator. Go to the label AHEAD
200C	7E	MOV	A, M	Yes, get larger number in accumulator
200D	0D	DCR	C	Decrement count
200E	C2, 07, 20	JNZ	LOOP	
2011	32, 50, 24	STA	2450 H	Store result in 2450 H
2014	76	HLT		Stop

Data -
2500 – 03
2501 - 98
2502 – 75
2503 – 99

Result –
2450 – 99

Conclusion :- From the above experiment, we have found the larger of two numbers and largest number in a data array by using 8085 microprocessor kit.