GOVERNMENT POLYTECHNIC, NAYAGARH

DIGITAL ELECTRONIC & MICROPROCESSOR LABORATORY MANUAL

ER. JADUNATH MURMU
Sr. Lecture in Electronics

ER. KUMAR AVISEK
Lab. Asst. (A.E. & I)

DEPARTMENT OF ELECTRICAL ENGINEERING

LABORATORY MANUAL
Digital Electronics & Microprocessor Lab

(5th Semester)

Govt. Polytechnic, Nayagarh

At - Ghasadeipur, Po - Nandighore
Pin - 752081

Approved by : Jadunath Murmu
Prepared By : Er. Kumar Avisek
Lab. Asst. (A. E. & I.)

GOVT. POLYTECHNIC, NAYAGARH

GOVT. POLYTECHNIC NAYAGARH -752081
Department of Skill Development & Technical Education
Govt. of Odisha ,Bhubaneswar

The Vision of the Institute

To emerge as a leading Technical Institution by imparting knowledge in field of Engineering and Technology to its pass outs to make them Industry employable and self enterprising so as to serve society in order to achieve harmonious relation between Human habitant and Nature in an Eco friendly environment

The Mission of the Institute:

- To deliver knowledge at par with cutting edge technology& promote academic growth
- To facilitate a creative and independent learning environment
- To develop a co-relation between academia, industry and society through various consultancy and testing constructional materials.
- To transform individuals by inculcating values, ethics and leadership qualities
- To establish an atmosphere where management principles and techniques will nurture in fulfilment of institutional aims and objectives

GOVT. POLYTECHNIC NAYAGARH -752081
Department of Skill Development & Technical Education
Govt. of Odisha, Bhubaneswar

PROGRAM EDUCATIONAL OBJECTIVE

PEO 1: Students are able to apply knowledge of basic science, computer fundamentals and basic principles of electrical engineering.

PEO 2: Students are able to enhance their analytical and problem solving ability using theoretical and practical knowledge to solve real time problem in electrical domain.

PEO 3: Students become adaptable to work in multi-disciplinary environment through continuous development of technical skills, professional efficiency.

PROGRAM SPECIFIC OUTCOME

PSO1: Ability to apply domain knowledge in testing and maintenance of electrical machine and equipment

PSO2: Ability to prepare design and estimate of various electrical installations.

PSO3: Contribute for the generation and utilization of green energy to meet the increasing demand of the society

GOVT. POLYTECHNIC NAYAGARH -752081
Department of Skill Development & Technical Education
Govt. of Odisha, Bhubaneswar

THE VISION OF THE DEPARTMENT

Empowering Electrical Professionals with strong ethical values and sound technical knowledge and unique set of skills which will enable to a rousing start to fulfill the future demands and needs of the industry and society to achieve sustainable development.

THE MISSION OF THE DEPARTMENT

- To align the teaching learning process and to provide basic foundation for the students to adapt to the changing industrial needs.
- To provide an atmosphere of Independent learning where they can be nurtured with qualities like leadership, responsibility and optimistic.
- To update the curriculum and learning materials in consultation with industry professionals to make students industry ready.
- To enrich with the latest developments through seminars, guest lectures, workshop and industry visits through interaction with industry professionals.
- To develop professional competency and technical expertise individually and through team effort thereby exhibit leadership in industry.

DIGITAL ELECTRONICS & MICROPROCESSOR LAB

CO1. Familiar with use of Digital ICs.

CO2. Understand the simple Digital circuit

CO3. Understand the counter and register

CO4.write and execute Assembly Language program

CO5. Application of 8085 using interfacing

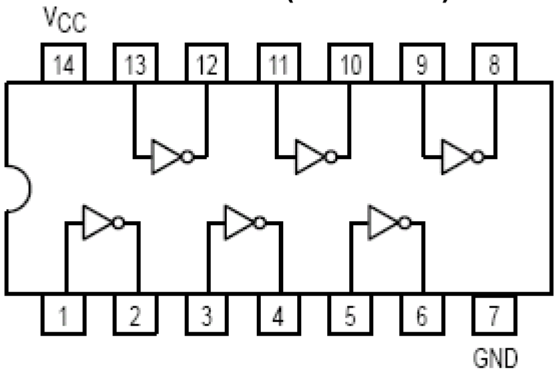
CO and PO Mapping Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
Familiar with use of Digital ICs.	3	1	1	-	2	-	-	2	-	-
Understand the simple Digital circuit	3	1	2	2	1	-	-	-	-	-
Understand the counter and register	3	1	1	2	1	-	-	2	-	-
write and execute Assembly Language program	3	1	2	2	2	-	2	-	-	-
Application of 8085 using interfacing	3	-	2	3	3	-	-	-	-	-
AVERAGE	3	0.8	1.6	1.8	1.8	-	0.4	0.8	-	-

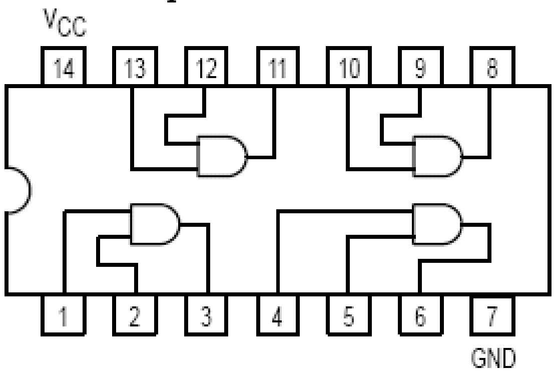
TABLE OF CONTENTS

TABLE OF CONTENTS

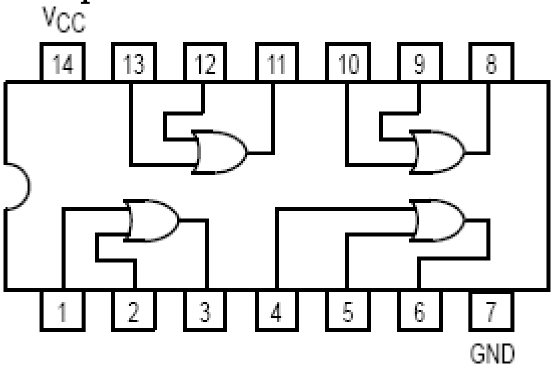
Inverter Gate (NOT Gate) 7404LS



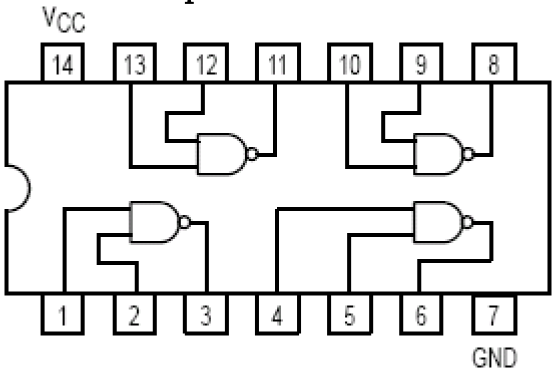
2-Input AND Gate 7408LS



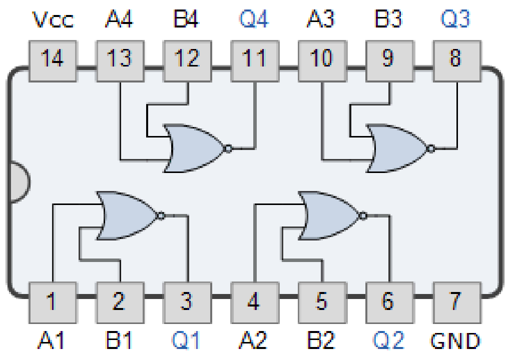
2-Input OR Gate 7432LS



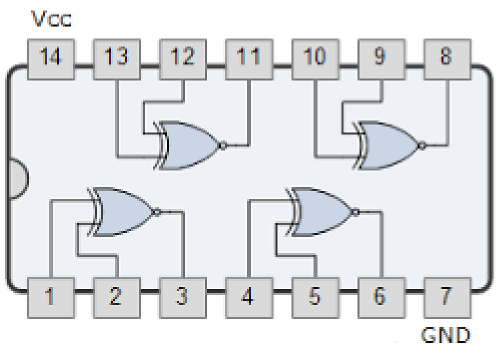
2-Input NAND Gate 7400LS



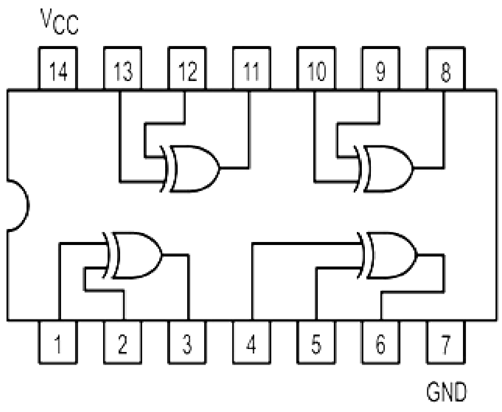
2-Input NOR Gate 7402LS



2-Input XNOR Gate 74266LS



2-Input XOR Gate 7486LS



Experiment No:1 Date: __/__/______

Aim: - To Verify truth tables of AND, OR, NOT, NOR, NAND, XOR, XNOR gates.

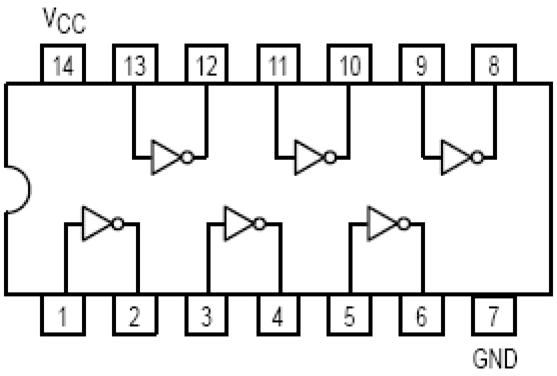
Apparatus Required: -

1.All the basic gates mention in the fig.
2.IC Trainer Kit

Procedure: -

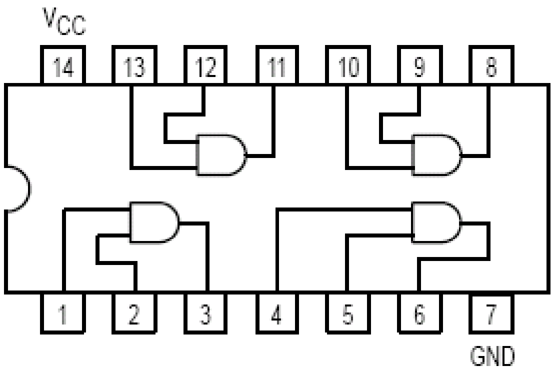
1. Place the IC on IC Trainer Kit.
2. Connect V_{CC} and ground to respective pins of IC Trainer Kit.
3. Connect the inputs to the input switches provided in the IC Trainer Kit.
4. Connect the outputs to the switches of O/P LEDs,
5. Apply various combinations of inputs according to the truth table and observe condition of LEDs.
6. Disconnect output from the LEDs and note down the corresponding multimeter voltage readings for various combinations of inputs.

Inverter Gate (NOT Gate) 7404LS



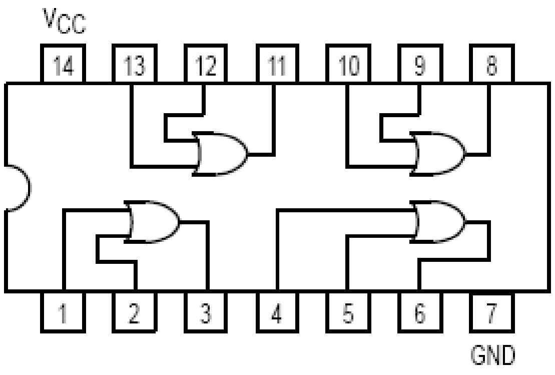
A	O/P
0	1
1	0

2-Input AND Gate 7408LS



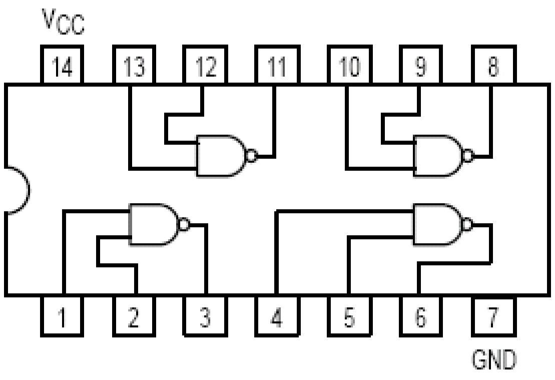
A	B	O/P
0	0	0
0	1	0
1	0	0
1	1	1

2-Input OR Gate 7432LS



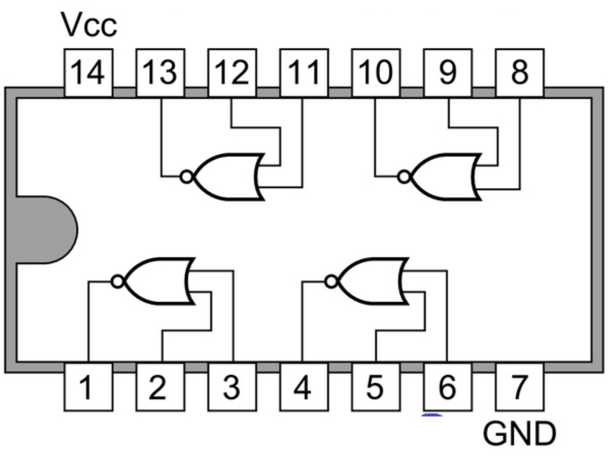
A	B	O/P
0	0	0
0	1	1
1	0	1
1	1	1

2-Input NAND Gate 7400LS



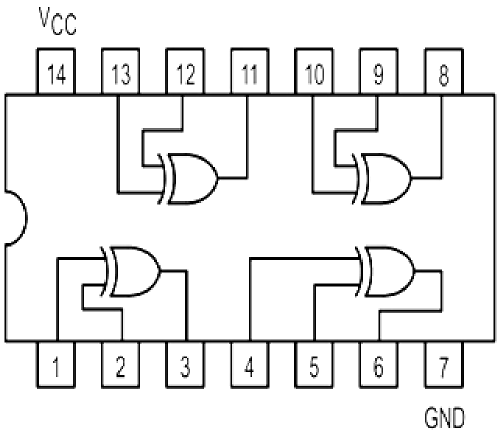
A	B	O/P
0	0	1
0	1	1
1	0	1
1	1	0

2-Input NOR Gate 7402LS



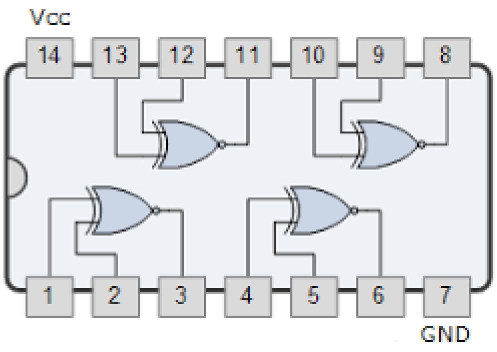
A	B	O/P
0	0	1
0	1	0
1	0	0
1	1	0

2-Input XOR Gate 7486LS



A	B	O/P
0	0	0
0	1	1
1	0	1
1	1	0

2-Input XNOR Gate 74266LS



A	B	O/P
0	0	1
0	1	0
1	0	0
1	1	1

Conclusion:-

Truth table of logic gates are verified.

Experiment No:2 **Date: __/__/_______**

Aim: - Implementation of various gates by using universal properties of NAND & NOR gates and
Verify truth table.

APPARATUS REQUIRED

1. Digital IC trainer kit
2. IC 7400 (NAND gate)
3. IC 7402(NOR gate)

THEORY:

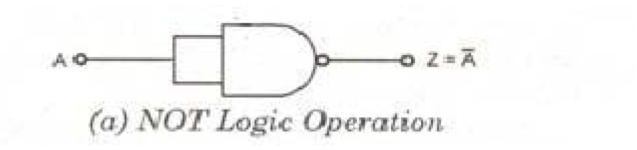
NAND OR NOR gates are sufficient for the realization of any logic expression. because of this reason, NAND and NOR gates are known as UNIVERSAL gates.

1. For NAND gate as universal gate

PROCEDURE:

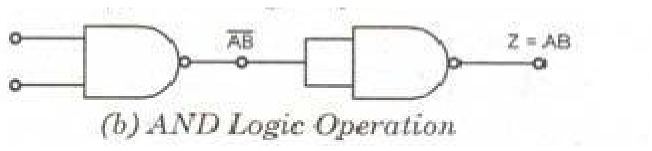
1. Make the connections as per the logic diagram.
2. Connect +5v to pin 14 & ground to pin 7 of IC 7400
3. Apply diff combinations of inputs to the i/p terminals.
4. Note o/p for NAND as universal gate.
5. Verify the truth table.

A	$\bar{A}$
0	1
1	0

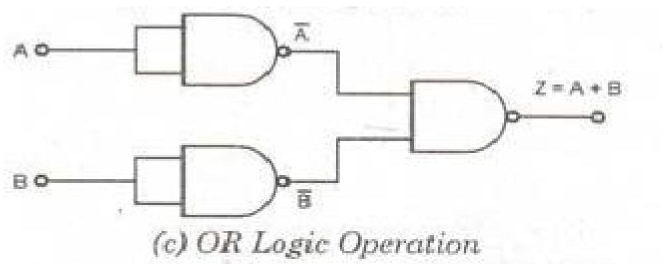


(a) NOT Logic Operation

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

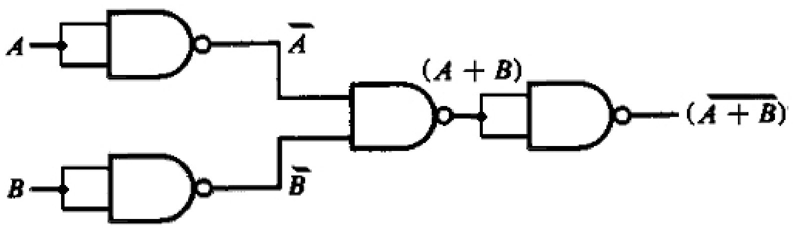


(b) AND Logic Operation



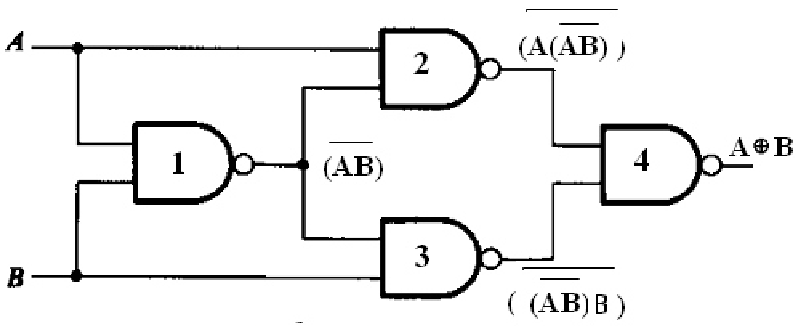
(c) OR Logic Operation

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

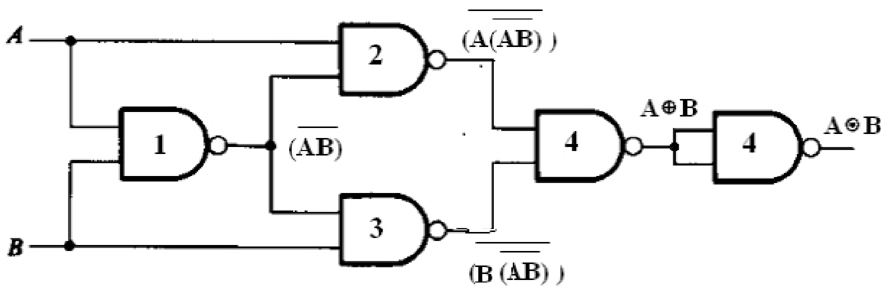


NOR Logic operation

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


A
2
$\overline{(A(\overline{AB}))}$
1
$(\overline{AB})$
4
A⊕B
3
B
$\overline{(\overline{AB})B}$
XOR Logic operation

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

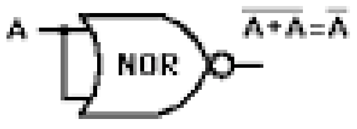

A
2
$\overline{(A(\overline{AB}))}$
1
$(\overline{AB})$
4
A⊕B
4
A⊙B
3
B
$\overline{(B(\overline{AB}))}$
XNOR Logic operation

A	B	AʘB
0	0	1
0	1	0
1	0	0
1	1	1

2.For NOR gate as universal gate

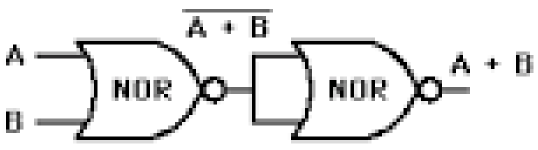
PROCEDURE:

1. Make the connections as per the logic diagram.
2. Connect +5v to pin 14 & ground to pin 7 of IC 7402
3. Apply diff combinations of inputs to the i/p terminals.
4. Note o/p for NAND as universal gate.
5. Verify the truth table



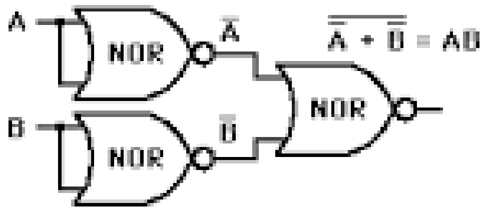
NOT Logic operation

A	$\bar{A}$
0	1
1	0



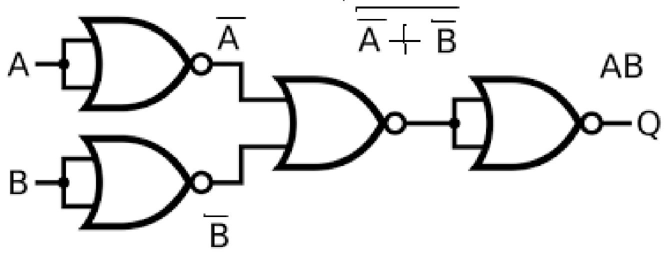
OR Logic operation

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



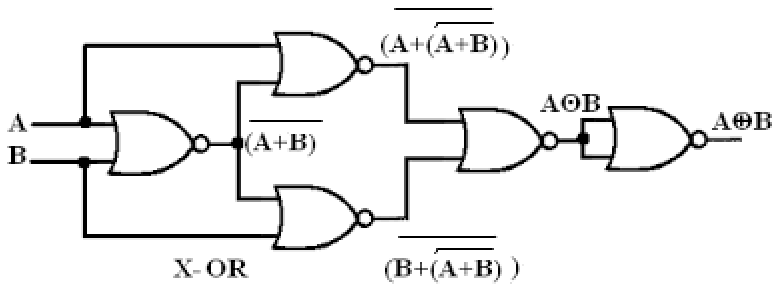
AND Logic operation

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1



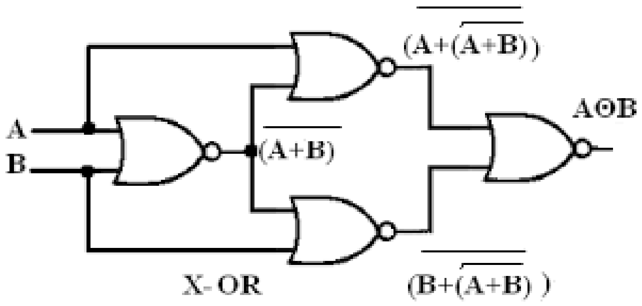
NAND Logic operation

A	B	AB
0	0	1
0	1	1
1	0	1
1	1	0


A
B
$\overline{(A+\overline{(A+B)})}$
$\overline{(A+B)}$
AΘB
A⊕B
X-OR
$\overline{(B+\overline{(A+B)})}$

XOR Logic operation

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0


A
B
$\overline{(A+\overline{(A+B)})}$
$\overline{(A+B)}$
AΘB
X-OR
$\overline{(B+\overline{(A+B)})}$

A	B	AΘB
0	0	1
0	1	0
1	0	0
1	1	1

Conclusion:-
We have constructed and verified truth table of all gates using universal gates NAND and NOR gate.

Experiment No:3 **Date: __/__/______**

Aim: - Implementation of half adder and Full adder using logic gates.

APPARATUS REQUIRED

1.IC 7486, IC 7432, IC 7408, IC 7400.
2.Digital trainer kit.

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

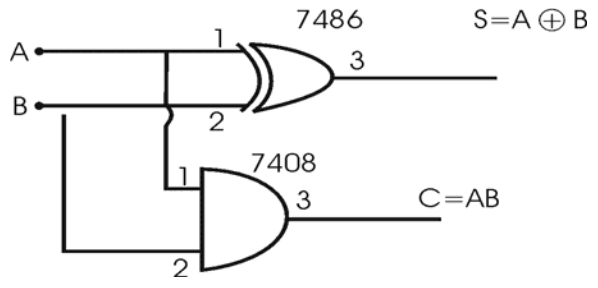
$$S = A \oplus B \qquad C = A\,B$$

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

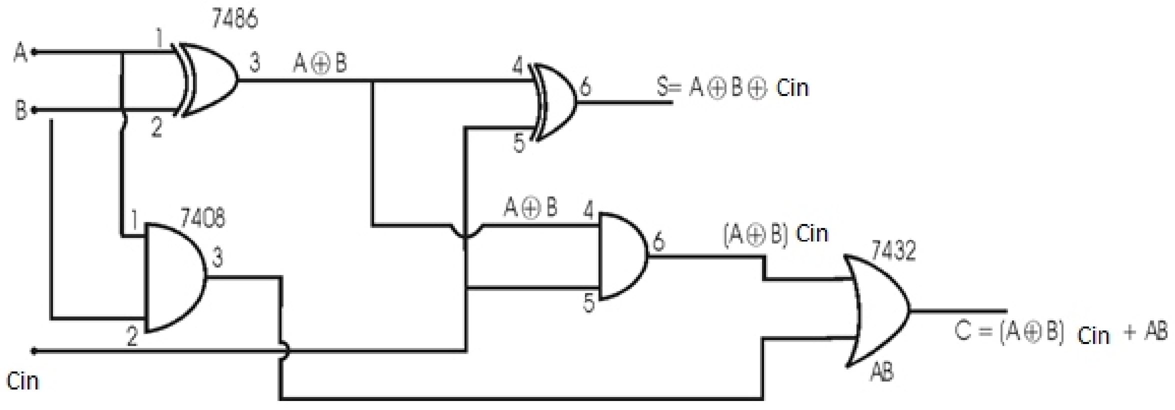
$$S = (x \oplus y) \oplus C_{in} \qquad C = xy + C_{in}\,(x \oplus y)$$

Procedure: -
1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on V_{CC} and apply various combinations of input according to the truth table.
4. Note down the output readings for half and full adder sum and the carry bit for different combinations of inputs.

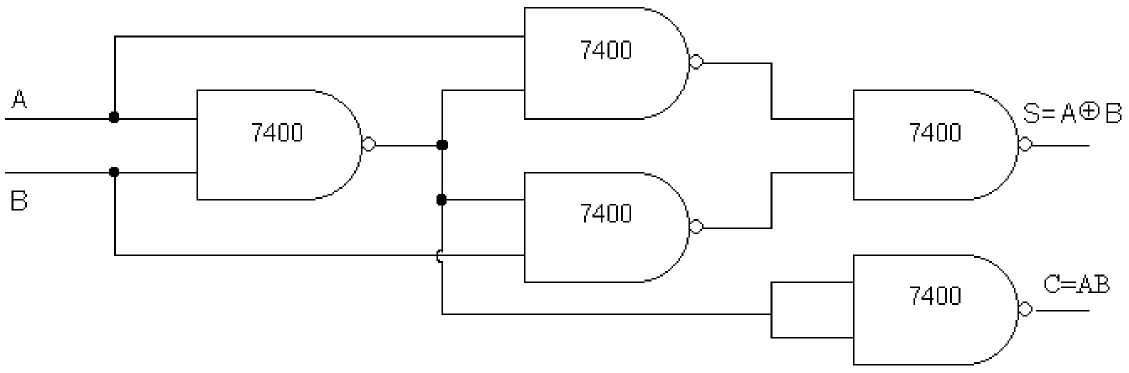
Half Adder using basic gates:-



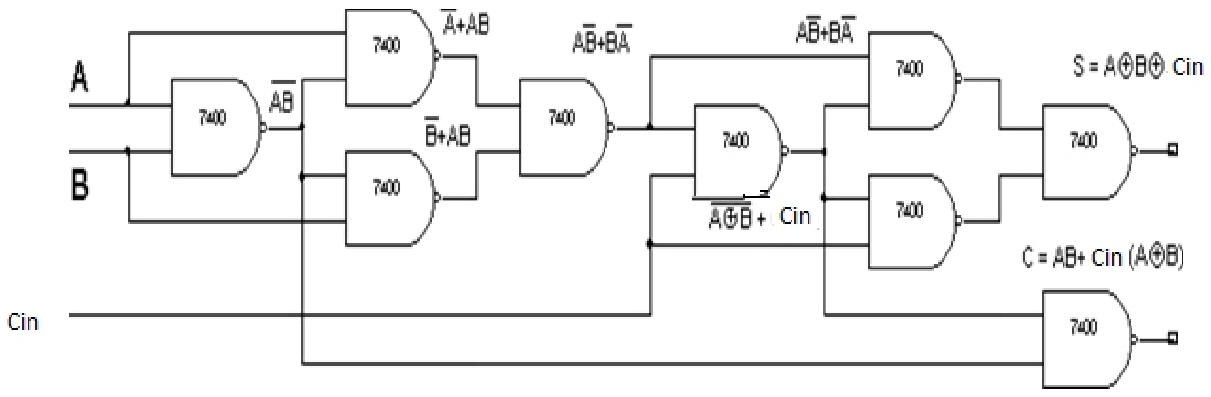
Full Adder using basic gates:-



Half Adder using NAND gates only:-

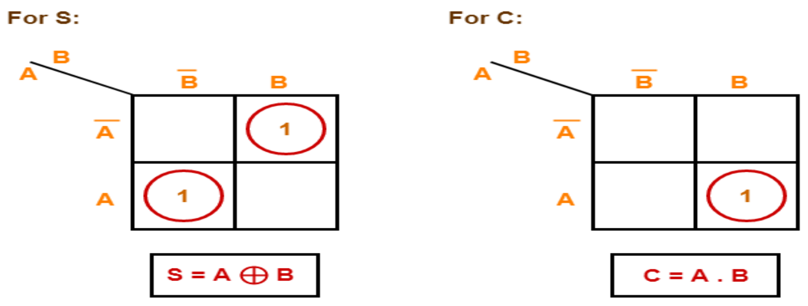


Full Adder using NAND gates only:-



K-map for half adder

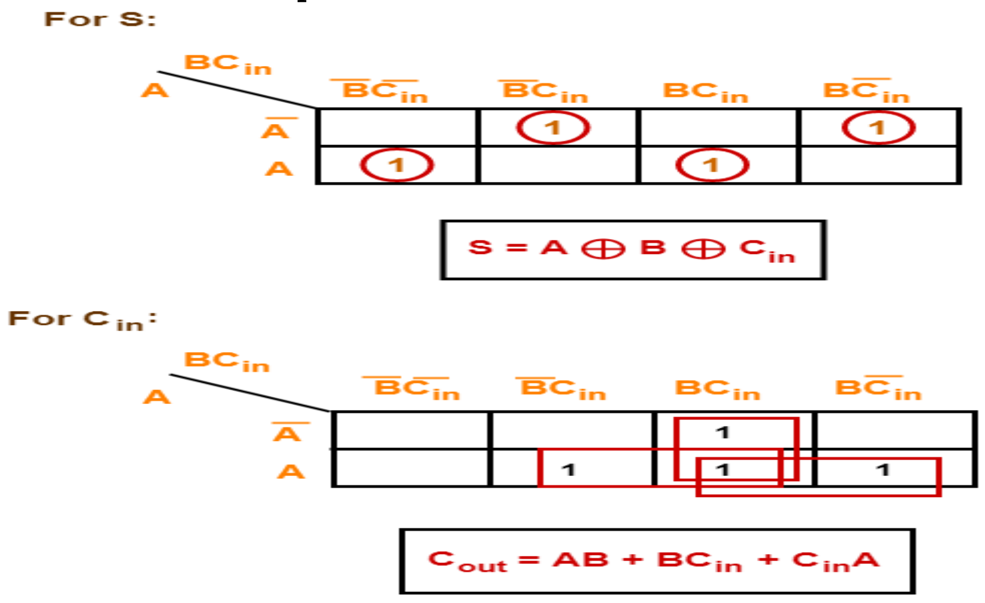
Half adder			
A	B	S	C
0	0		
0	1		
1	0		
1	1		



K Maps

K-map for full adder

Full Adder				
A	B	Cin	S	C
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



<u>Conclusion: -</u>

Half adder and full adder are constructed and their truth tables are verified.

Experiment No:4 **Date: __/__/_**

Aim: - Implementation of half subtractor and Full subtractor using logic gates.

APPARATUS REQUIRED

1.IC 7486, IC 7432, IC 7408,IC7404, IC7400.
2.Digital trainer kit.

THEORY:

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

$$D = A \oplus B \qquad B_r = \bar{A}\,B$$

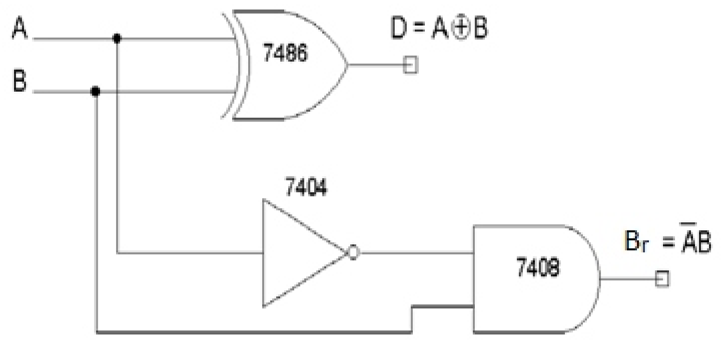
Full Subtractor: Subtracting two single-bit binary values, B, Cin from a single-bit value A produces a difference bit D and a borrow out Br bit. This is called full subtraction. The Boolean functions describing the full-subtracter are:

$$D = (x \oplus y) \oplus B_{in} \qquad B_r = \bar{A}B + \bar{A}\,(B_{in}) + B\,(B_{in})$$

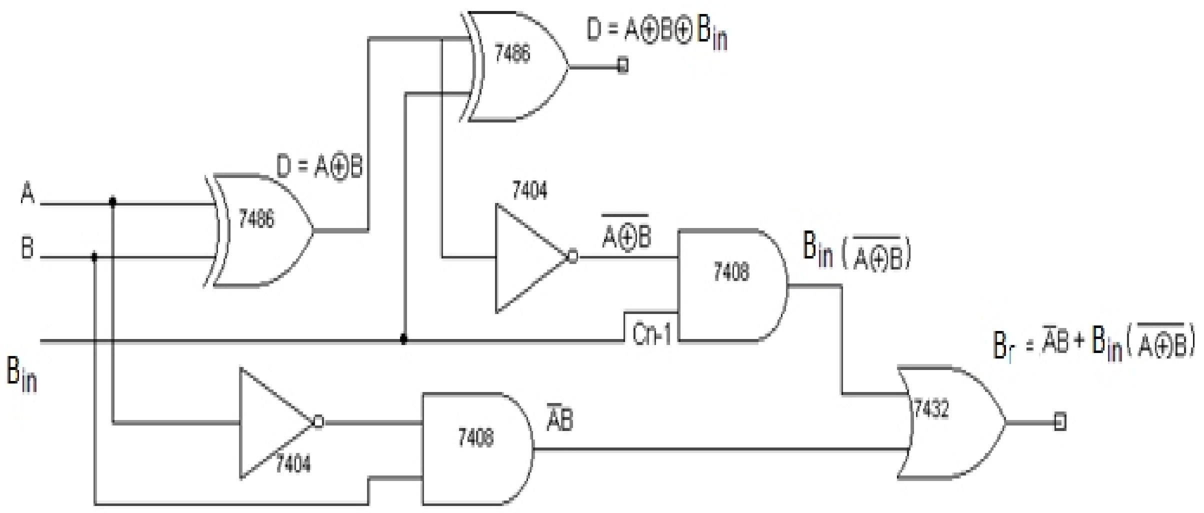
Procedure: -

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on V_{CC} and apply various combinations of input according to the truth table.
4. Note down the output readings for half and full subtractor difference and borrow bit for different combinations of inputs.

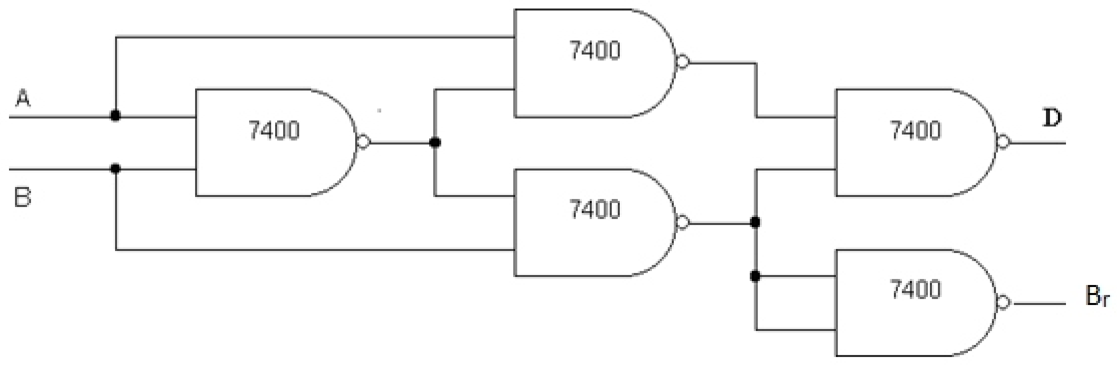
Using X – OR and Basic Gates (a)Half Subtractor



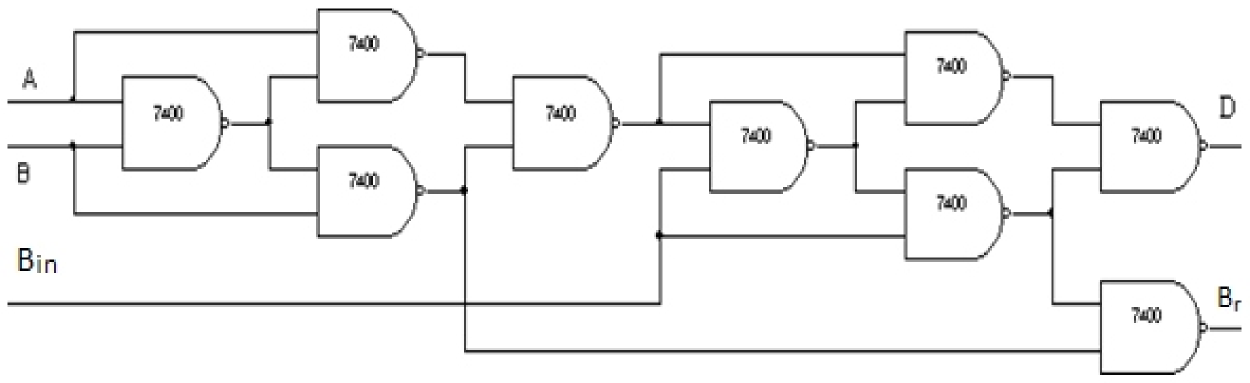
Full Subtractor



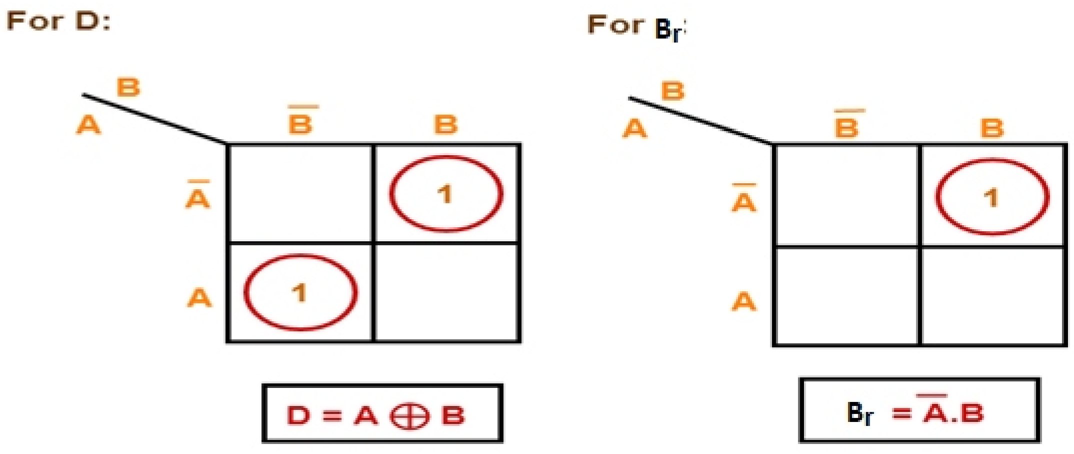
Using only NAND gate (a) Half subtractor



(b) Full Subtractor



Half Subtractor			
A	B	D	B_r
0	0		
0	1		
1	0		
1	1		



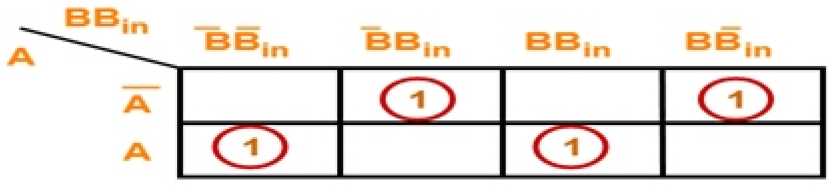
$\boxed{D = A \oplus B}$

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

K Maps

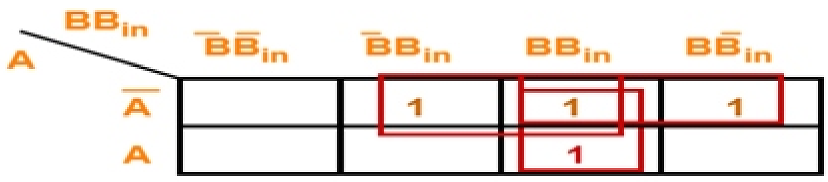
Full Subtractor				
A	B	B_{in}	D	B_r
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

For D:



$$\boxed{D = A \oplus B \oplus B_{in}}$$

For B_r



$$\boxed{B_r = \bar{A}B + (\bar{A} + B)B_{in}}$$

Conclusion: -

Half subtractor and full subtractor are constructed and their truth tables are verified.

Experiment No:5 Date: __/__/_

Aim: - Implementation of a 4-bit Binary to Gray code converter.

APPARATUS REQUIRED

1. IC 7486
2. Digital trainer kit

THEORY:

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.
The input variable are designated as B3, B2, B1, B0 and the output variables are designated as G3, G2, G1, G0.

Procedure: -

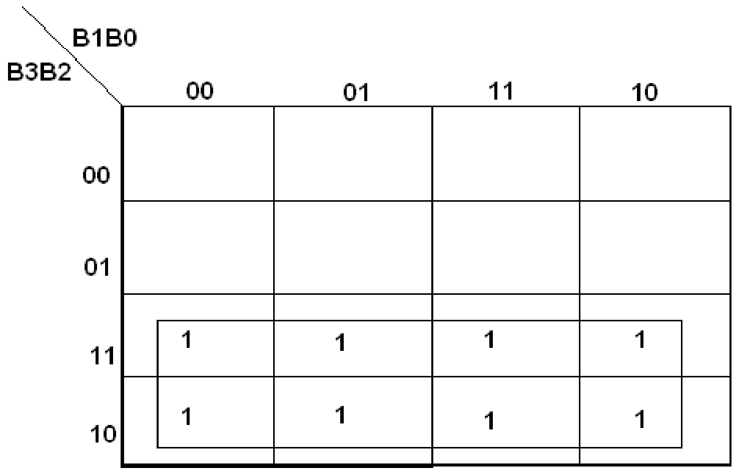
1. The circuit connections are made as shown in fig.
2. Pin (14) is connected to +Vcc and Pin (7) to ground.
3. In the case of binary to gray conversion, the inputs B0, B1, B2 and B3 are given at respective pins and outputs G0, G1, G2, G3 are taken for all the 16 combinations of the input.
4. The values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

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

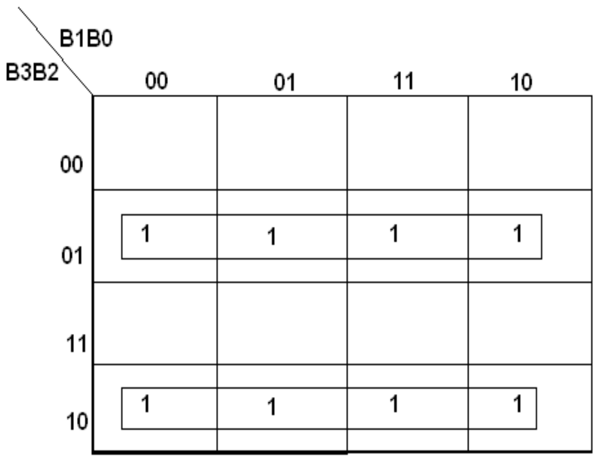
0	0	1	0		0	0	1	1
0	0	1	1		0	0	1	0
0	1	0	0		0	1	1	0
0	1	0	1		0	1	1	1
0	1	1	0		0	1	0	1
0	1	1	1		0	1	0	0
1	0	0	0		1	1	0	0
1	0	0	1		1	1	0	1
1	0	1	0		1	1	1	1
1	0	1	1		1	1	1	0
1	1	0	0		1	0	1	0
1	1	0	1		1	0	1	1
1	1	1	0		1	0	0	1
1	1	1	1		1	0	0	0

K-Map for G_3:



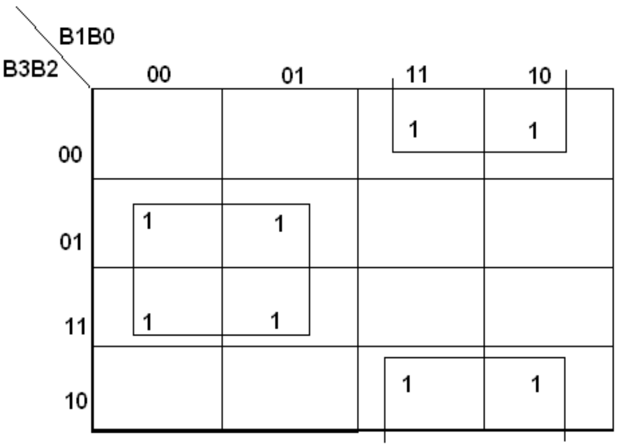
$G_3 = B_3$

K-Map for G_2:



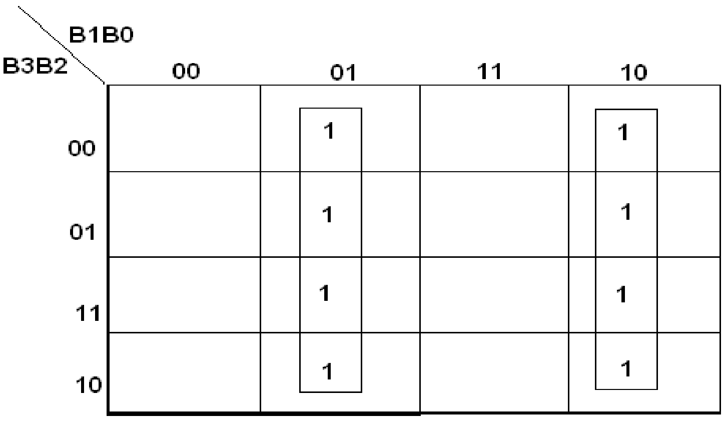
$G2 = B3 \oplus B2$

K-Map for G_1:



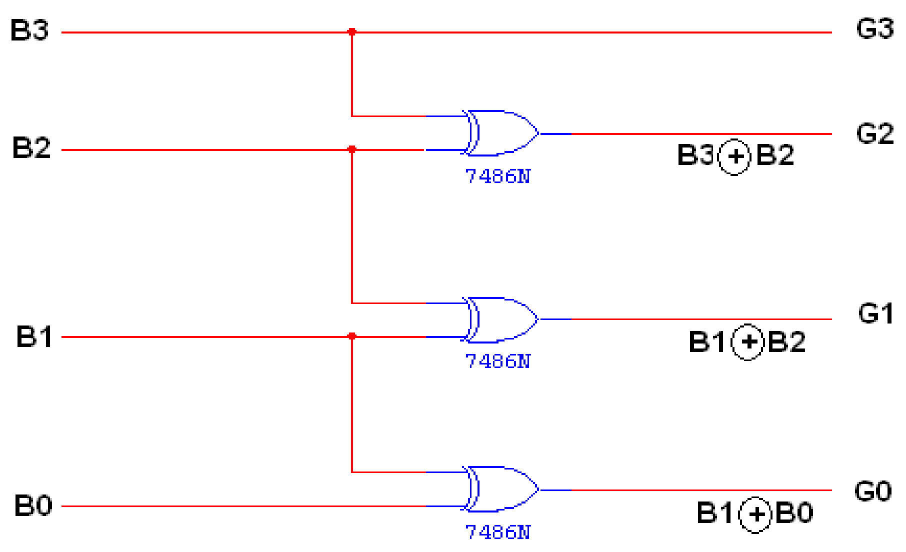
$G1 = B1 \oplus B2$

K-Map for G_0:



$G0 = B1 \oplus B0$

LOGIC DIAGRAM



Conclusion: -

4-bit Binary to Gray code converter is constructed and their truth tables are verified.

Experiment No:6 **Date: __/__/_**

Aim: - Implementation of a Single bit digital comparator.

APPARATUS REQUIRED

1. IC 7404,IC 7408,IC 74266
2. Digital trainer kit

THEORY:

Magnitude Comparator is a logical circuit, which compares two signals A and B and generates three logical outputs, whether A > B, A = B, or A < B.

Procedure: -

1. The circuit connections are made as shown in fig.
2. Pin (14) is connected to +Vcc and Pin (7) to ground.
3. The inputs A,B are given at respective pins and outputs A > B, A = B, or A < B are connected to the output LED.
4. The values of the outputs are tabulated.

TRUTH TABLE

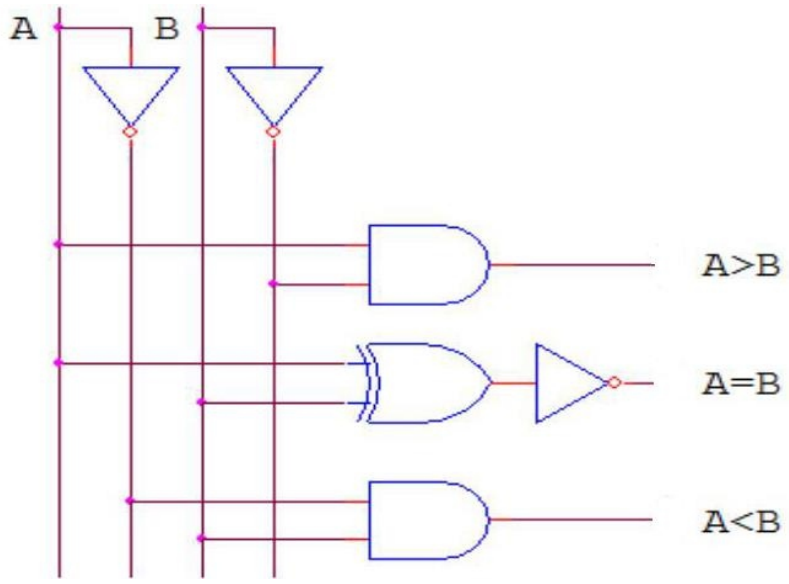
$$A>B = A\,\overline{B}$$

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

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

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

LOGIC DIAGRAM



Conclusion: -

A Single bit digital comparator is constructed and it’s truth tables are verified.

EXPERIMENT NO: -7

AIM: -

To study Multiplexer and Demultiplexer.

APPARATUS REQUIRED: -

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	4x1 MULTIPLEXER	IC 74153	1
2	1x4 DEMULTIPLEXER	IC 74156	1
3	BREAD BOARD	-	1
4	RESISTOR	220Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY : -

MULTIPLEXER :

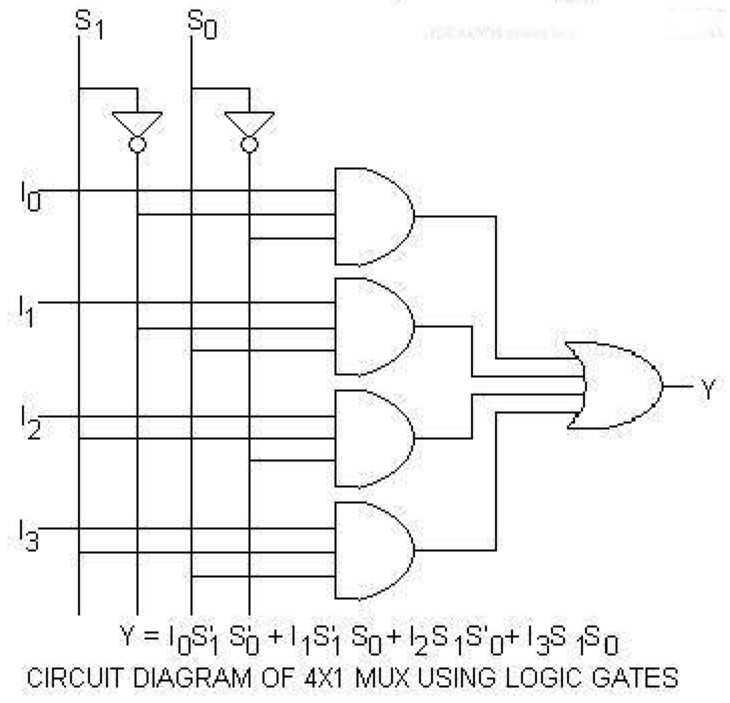
Multiplexer generally means many into one. A multiplexer is a circuit with many inputs but only one output. By applying control signals we can steer any input to the output. The circuit has n input signal, control signal (m) & one output signal, where $2^{m} = n$. One of the popular multiplexers is the 16 to 1 multiplexer, which has 16 input bits, 4 control bits & 1 output bit.

4x1 Multiplexer

The 4x1 Multiplexer has four input lines I_0, I_1, I_2 and I_3 and one output line Y. The selection of a particular input is controlled by set of selection lines S_1 and S_0.

TRUTH TABLE

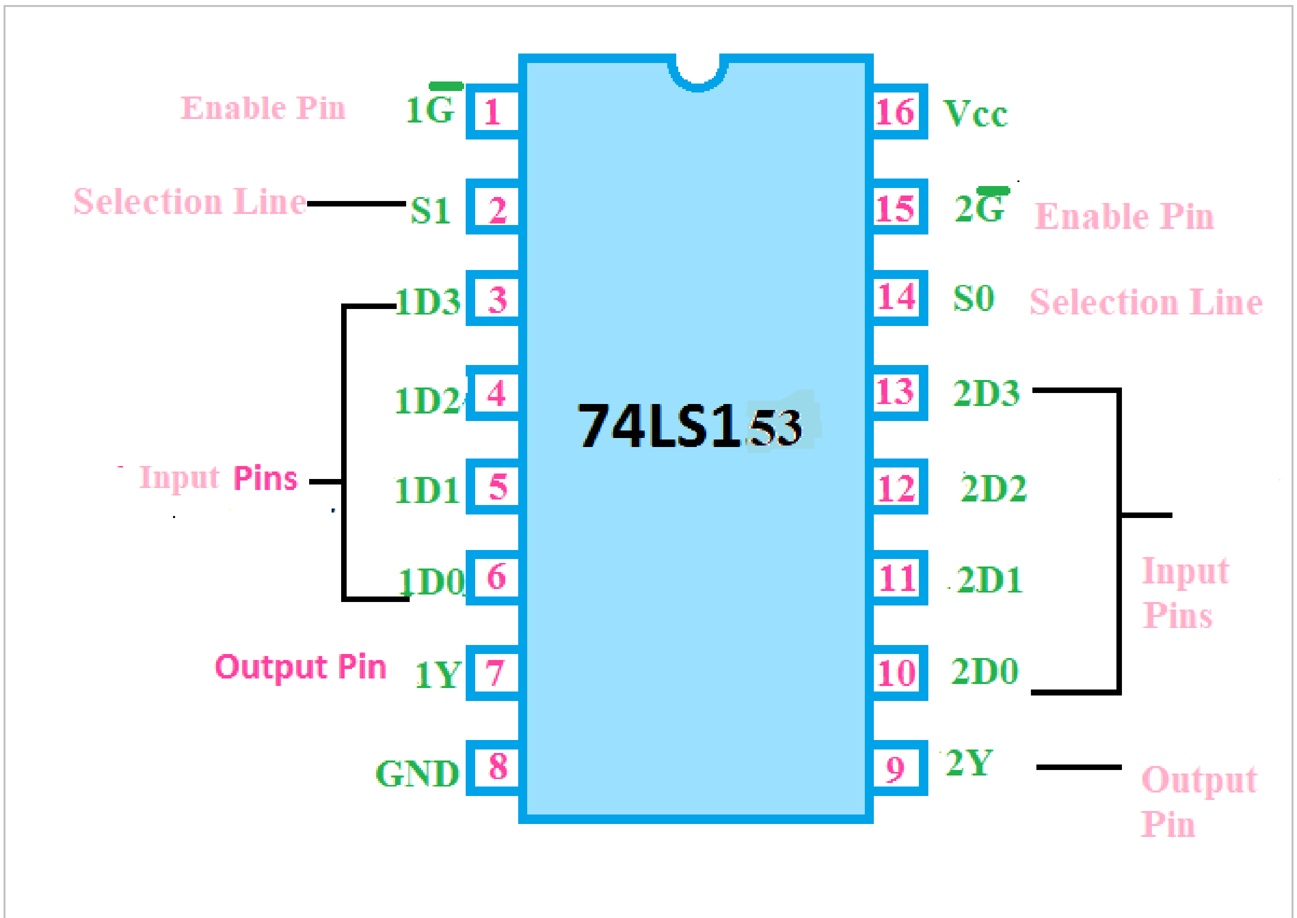
Selection line		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



$$Y = I_0S'_1\,S'_0 + I_1S'_1\,S_0 + I_2S_1S'_0 + I_3S_1S_0$$

CIRCUIT DIAGRAM OF 4X1 MUX USING LOGIC GATES

PIN DIAGRAM OF IC 74153



IC 74LS153 Pin Diagram

IC 74153Truth Table

Thetruth tableof IC 74153 isgivenbelow

Selectinputs		Inputs(a orb)					Output
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Z
X	X	H	X	X	X	X	L
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

H = High voltage
L = Low voltage
X = Don't care

DEMULTIPLEXER:

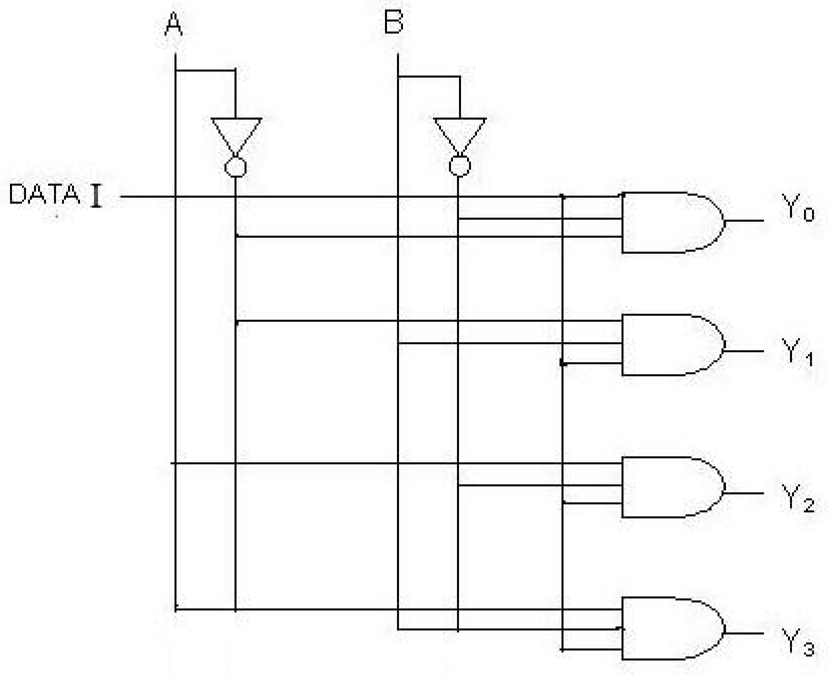
Demultiplexer means generally one into many. A demultiplexer is a logic circuit with one input and many outputs. By applying control signals, we can steer the input signal to one of the output lines. The circuit has one input signal, m control signal and n output signals, where 2[illegible] = [illegible] functions as an electronic switch to route an incoming data signal to one of several outputs.

1x4 Demultiplexer

The 1x4 Demultiplexer has one input I and four outputs Y_0, Y_1, Y_2 and Y_3.

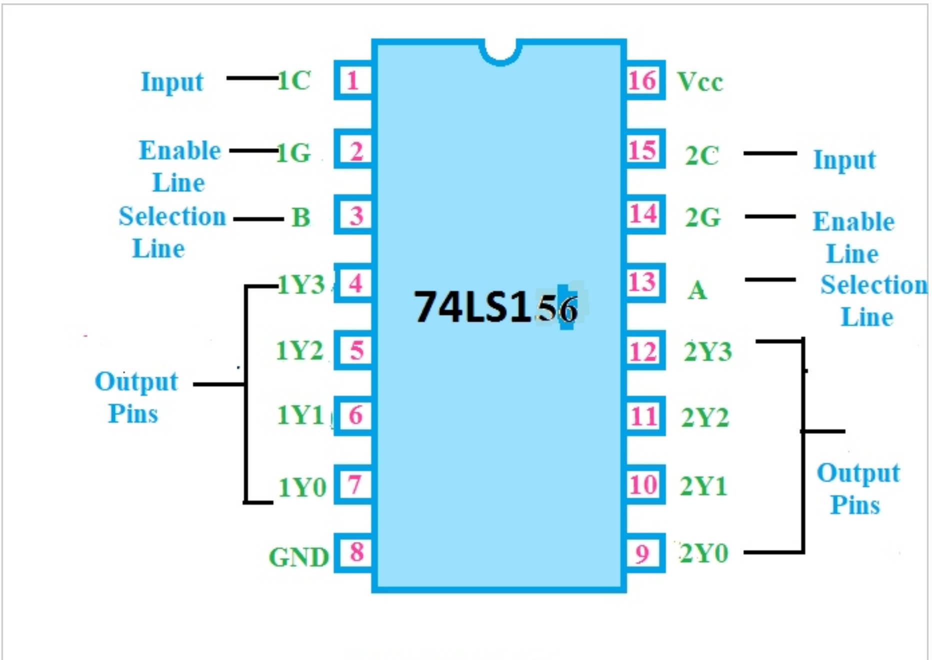
TRUTH TABLE

Input	Selection line		Output			
I	S_1	S_0	Y_3	Y_2	Y_1	Y_0
I	0	0	0	0	0	I
I	0	1	0	0	I	0
I	1	0	0	I	0	0
I	1	1	I	0	0	0



CIRCUIT DIAGRAM OF 1 x 4 DEMUX

PIN DIAGRAM OF IC 74156



IC 74LS156 Pin Diagram

PROCEDURE: -

1. Connect the circuit as shown in figure.
2. Apply V_{CC} & ground signal to the IC.
3. Observe the input & output according to the truth table.

PRECAUTIONS: -

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.

OBSERVATION: - [L=logic0, H=logic1]

4x1 Multiplexer

Input line				**Selection line**		**Output**
I_0	I_1	I_2	I_3	S_1	S_0	Y
H	L	L	H	L	L	
L	L	H	H	L	H	
H	H	L	L	H	L	
L	H	L	H	H	H	

1x4 Demultiplexer

Input	**Selection line**		**Output**			
I	S_1	S_0	Y_3	Y_2	Y_1	Y_0
H	L	L				
H	L	H				
H	H	L				
H	H	H				

RESULT: - Hence, the truth table of 4x1 Multiplexer and 1x4 Demultiplexer are verified and found ok.

Experiment No : 8

AIM:

To verify the characteristic table of RS, D, JK, and T Flip flops .

APPARATUS REQUIRED:

S.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	NOR gate	IC 7402	
3.	NOT gate	IC 7404	
4.	AND gate (three input)	IC 7411	
5.	NAND gate	IC 7400	
6.	Connecting wires		As required

THEORY:

A Flip Flop is a sequential device that samples its input signals and changes its output states only at times determined by clocking signal. Flip Flops may vary in the number of inputs they possess and the manner in which the inputs affect the binary states.

RS FLIP FLOP:

The clocked RS flip flop consists of NAND gates and the output changes its state with respect to the input on application of clock pulse. When the clock pulse is high the S and R inputs reach the second level NAND gates in their complementary form. The Flip Flop is reset when the R input high and S input is low. The Flip Flop is set when the S input is high and R input is low. When both the inputs are high the output is in an indeterminate state.

D FLIP FLOP:

To eliminate the undesirable condition of indeterminate state in the SR Flip Flop when both inputs are high at the same time, in the D Flip Flop the inputs are never made equal at the same time. This is obtained by making the two inputs complement of each other.

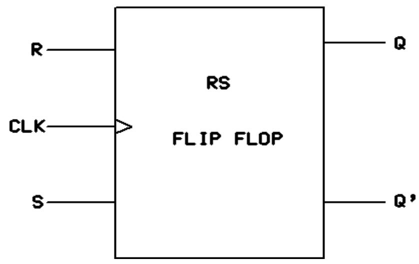
JK FLIP FLOP:

The indeterminate state in the SR Flip-Flop is defined in the JK Flip Flop. JK inputs behave like S and R inputs to set and reset the Flip Flop. The output Q is ANDed with K input and the clock pulse, similarly the output Q' is ANDed with J input and the Clock pulse. When the clock pulse is zero both the AND gates are disabled and the Q and Q' output retain their previous values. When the clock pulse is high, the J and K inputs reach the NOR gates. When both the inputs are high the output toggles continuously. This is called Race around condition and this must be avoided.

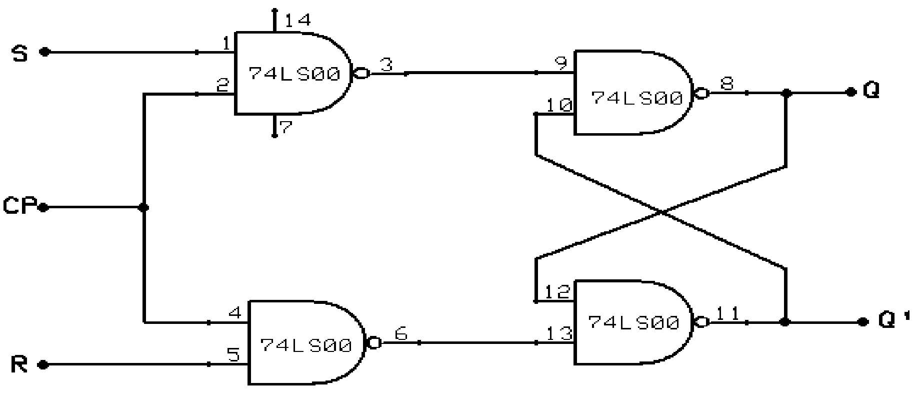
T FLIP FLOP:

This is a modification of JK Flip Flop, obtained by connecting both inputs J and K inputs together. T Flip Flop is also called Toggle Flip Flop.

RS FLIP FLOP
LOGIC SYMBOL :



CIRCUIT DIAGRAM :

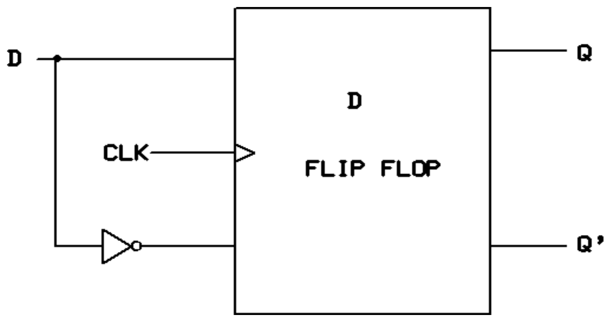


CHARACTERISTIC TABLE:

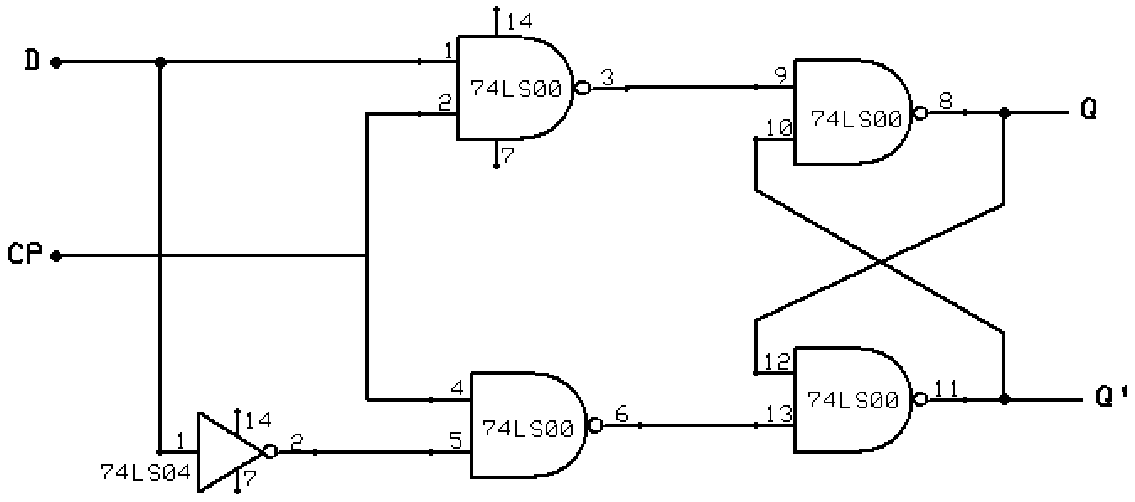
CLOCK PULSE	INPUT		PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
	S	R			
1	0	0	0	0	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	0	
5	1	0	0	1	
6	1	0	1	1	
7	1	1	0	X	
8	1	1	1	X	

D FLIP FLOP

LOGIC SYMBOL:



CIRCUIT DIAGRAM:

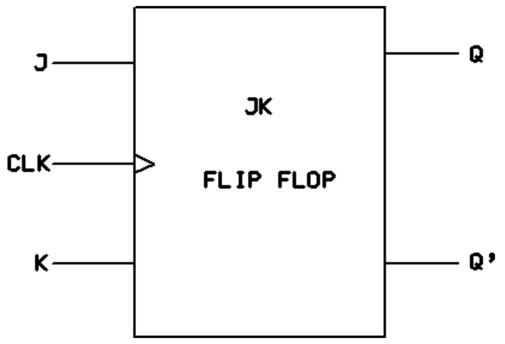


CHARACTERISTIC TABLE:

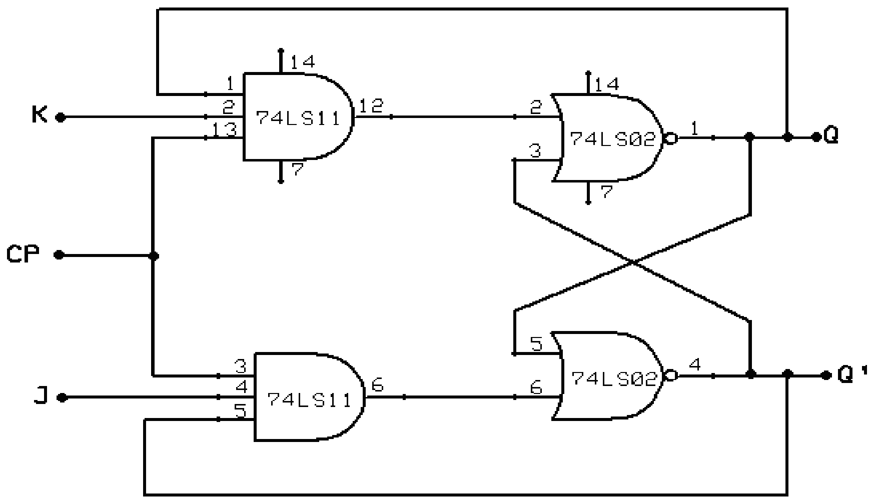
CLOCK PULSE	INPUT D	PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
1	0	0	0	
2	0	1	0	
3	1	0	1	
4	1	1	1	

JK FLIP FLOP

LOGIC SYMBOL:



CIRCUIT DIAGRAM :

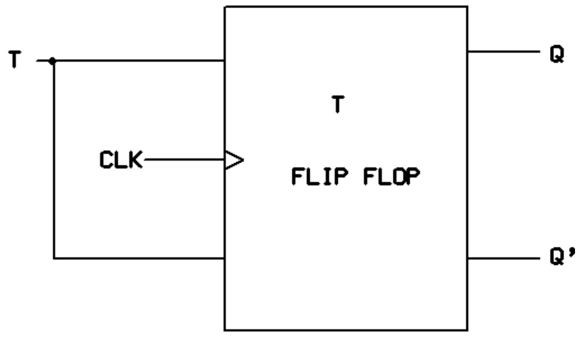


CHARACTERISTIC TABLE:

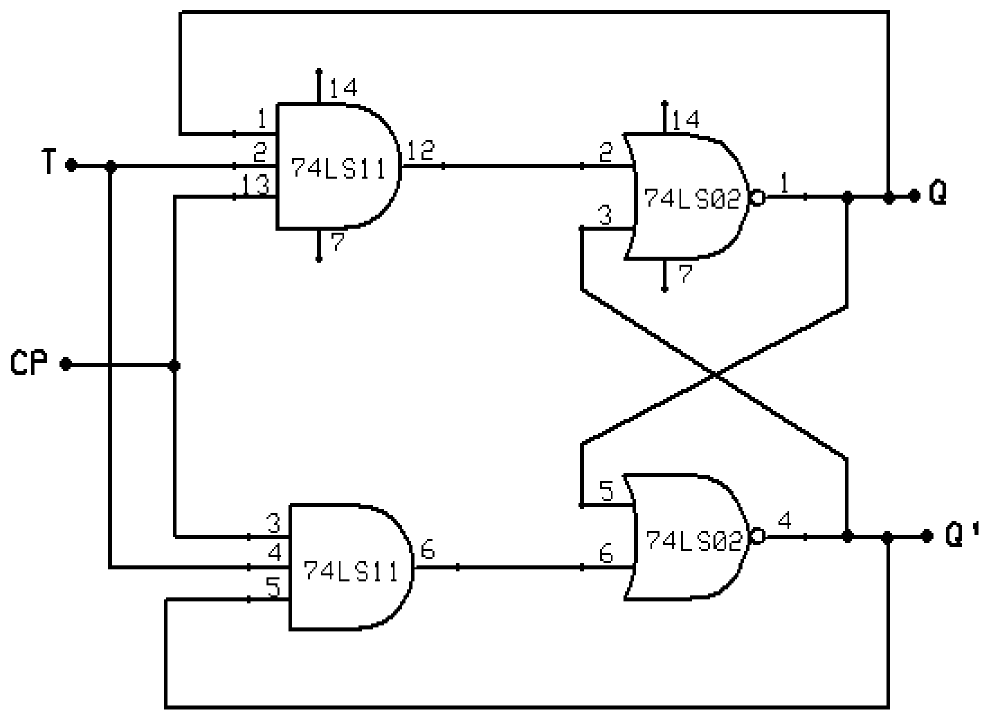
CLOCK PULSE	INPUT		PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
	J	K			
1	0	0	0	0	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	0	
5	1	0	0	1	
6	1	0	1	1	
7	1	1	0	1	
8	1	1	1	0	

T FLIPFLOP

LOGIC SYMBOL:



CIRCUIT DIAGRAM:



CHARACTERISTIC TABLE:

CLOCK PULSE	INPUT T	PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
1	0	0	0	
2	0	1	0	
3	1	0	1	
4	1	1	0	

PROCEDURE:

1. Connections are given as per the circuit diagrams.
2. For all the ICs 7th pin is grounded and 14th pin is given +5 V supply.
3. Apply the inputs and observe the status of all the flip flops.

RESULT : The Characteristic tables of RS, D, JK, T flip flops were verified.

Experiement No : 9

AIM :

Realize a 4-bit asynchronous UP/Down counter with a control for up/down counting .

APPARATUS REQUIRED :

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	JK FLIP FLOP	IC 7473	2
2.	IC TRAINER KIT	-	-
3.	PATCH CORDS	-	30

THEORY:

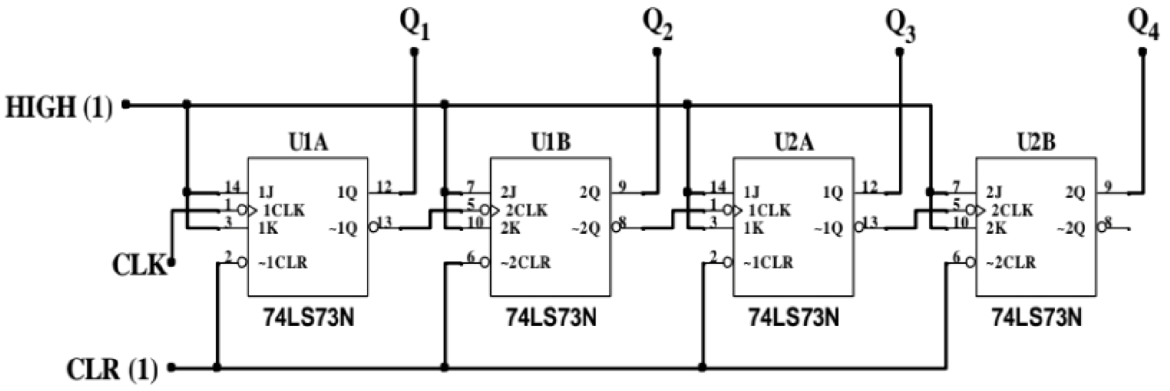
UP-COUNTER :

Asynchronous counters are the circuit that used to count the binary numbers in prescribed sequence. In asynchronous counter the flip-flops are not triggered with common clock pulse. Except the first (least significant) flip-flop others are triggered by the output of previous flip-flop while the first one is triggered by the clock pulse. Hence asynchronous counter is also called as ripple counter. When inputs set into logic high the JK flip-flops are continuously present in the toggle condition which complements the output continuously. This cause to prevent the circuit from triggering of two adjacent flip-flops simultaneously.

DOWN-COUNTER :

Asynchronous down counter performs the reverse operation of up-counter which counts the binary number by decreasing one when the flip-flops are activated by the clock pulse. First flip-flop triggered by clock pulse the remaining flip-flops are triggered by the inverted output of previous flip-flop. It is an only difference from up-counter.

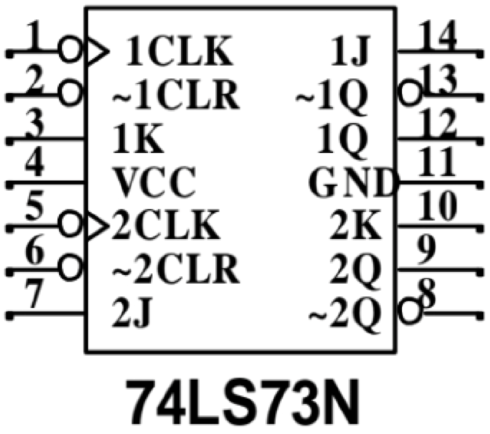
LOGIC DIAGRAM OF 4-BIT ASCHNCRONOUS (RIPPLE) DOWN-COUNTER:



TRUTH TABLE:

INPUT	OUTPUTS			
CLK	Q_4	Q_3	Q_2	Q_1
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	1	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	0	1	1	1
9	0	1	1	0
10	0	1	0	1
11	0	1	0	0
12	0	0	1	1
13	0	0	1	0
14	0	0	0	1
15	0	0	0	0

PIN DIAGRAM OF IC 7473



PROCEDURE :

1. Connections are given as per the logic diagram.
2. Apply the clock pulse and verify the truth table.

RESULT :

Experiement No : 10

AIM :

Realize a 4-bit synchronous UP/Down counter with a control for up/down counting .

APPARATUS REQUIRED :

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	JK FLIP FLOP	IC 7473	2
2.	IC TRAINER KIT	-	-
3.	PATCH CORDS	-	30

THEORY:

SYNCHRONOUS UP/DOWN COUNTER :

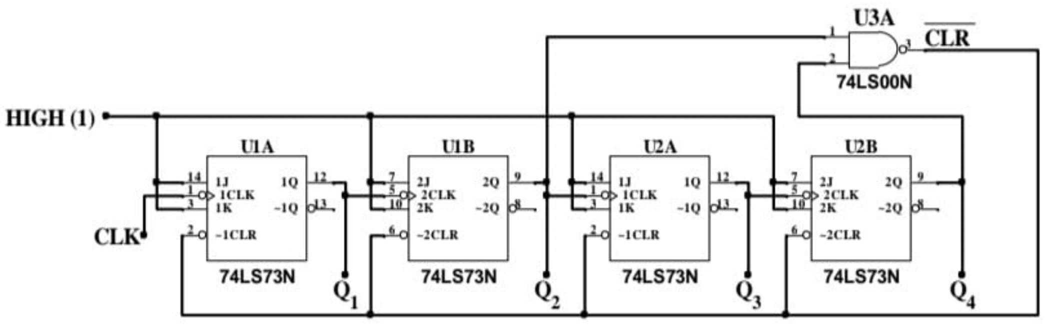
Up/Down counter is a circuit which perform the logic count either up or down by increasing or decreasing a number by 1. Synchronous Up/Down counter is a circuit which executes the counting operation either up or down with a commonly clocked flip-flops. In a counter the progress of Up and Down counting operations are selected by the control signal. After selecting the counting operation by enforcing the clock pulses to the flip-flops desired counting operation is executed. If the control signal is logic low (0) then the counter counts in the decreasing order that is down counter. When the control signal is logic high (1) then the counter counts in the increasing order that is up counter.

PROCEDURE :

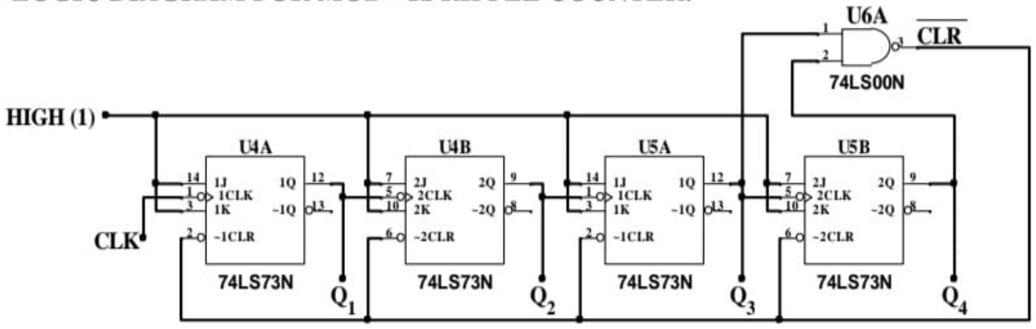
1. Connections are given as per the logic diagram.
2. Apply the clock pulse and verify the truth table.

RESULT :

LOGIC DIAGRAM FOR MOD - 10 RIPPLE COUNTER:



LOGIC DIAGRAM FOR MOD - 12 RIPPLE COUNTER:



TRUTH TABLE of MOD-10:

Input	OUTPUTS			
CLK	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

TRUTH TABLE of MOD-12:

Input	OUTPUTS			
CLK	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	0	0	0	0

Aim(s) / Objective(s) / Purpose

Answer in a sentence or two, "what is the purpose of this experiment?"

1. Design a 4-bit ripple/asynchronous counter and a mod-10 ripple counter

Introduction / Background

Introduction provides explanation of the topic of experiment. This section should explain the theoretical principles. This section should include block diagrams / logic diagrams where needed. Label the main components, input and output variables. Write equations using an equation writer in a word processor. Draw the truth table / function table / state table where needed.

Tips: Any information copied directly or verbatim from Lab manuals or other references should be stated within quotes and referred, otherwise, it is considered plagiarism

Counter:

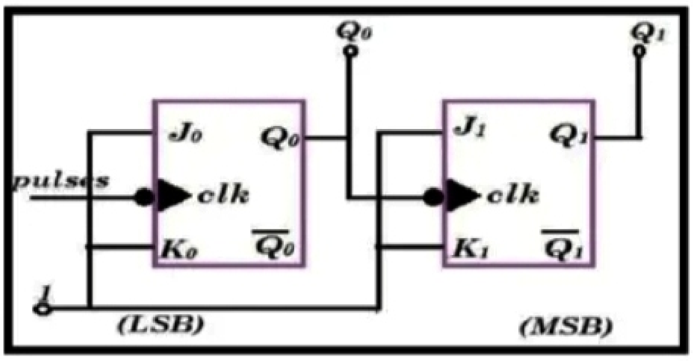
A counter is a special type of register which counts a desired sequence of numbers. The main memory element used in implementing counter is FLIP-FLOP. Counters are broadly classified into two types:

- Asynchronous/Ripple counters
- Synchronous counters

In this lab we'll look into Asynchronous/ripple counters.

Asynchronous/Ripple counters:

Asynchronous counter is one of the most common counters used to count numbers. In asynchronous counters, clock pulse is connected to one of the flip-flops and the output of that flip-flop is connected to the clock of the second flip-flop. Basically, output of one flip-flop drives the other flip-flop. Asynchronous counters are easy to design for counting numbers of higher order but it has one major drawback: it is slow. Flip-Flop takes time to generate the result. The output of that flip-flop is connected to the clock input of the next flip-flop. As a result, there will be larger delay to generate the output.



BIT SYNCHRONOUS UP COUNTER

Materials / Equipment

List any and all materials / expendables / equipment / parts you will need for the experiment. Use the proper name and include the number if you need multiples

- J-K Flip-Flops (×4)
- AND Gate (IC 7408)
- Input pins
- LEDs
- Clock pulse

Procedure

Write out step by step instructions on how to perform the lab. Include exact measurements if needed. Remember, your goal here is to make this exact experiment reproducible. Be specific. Your procedure should be written so than anyone else could repeat the experiment.

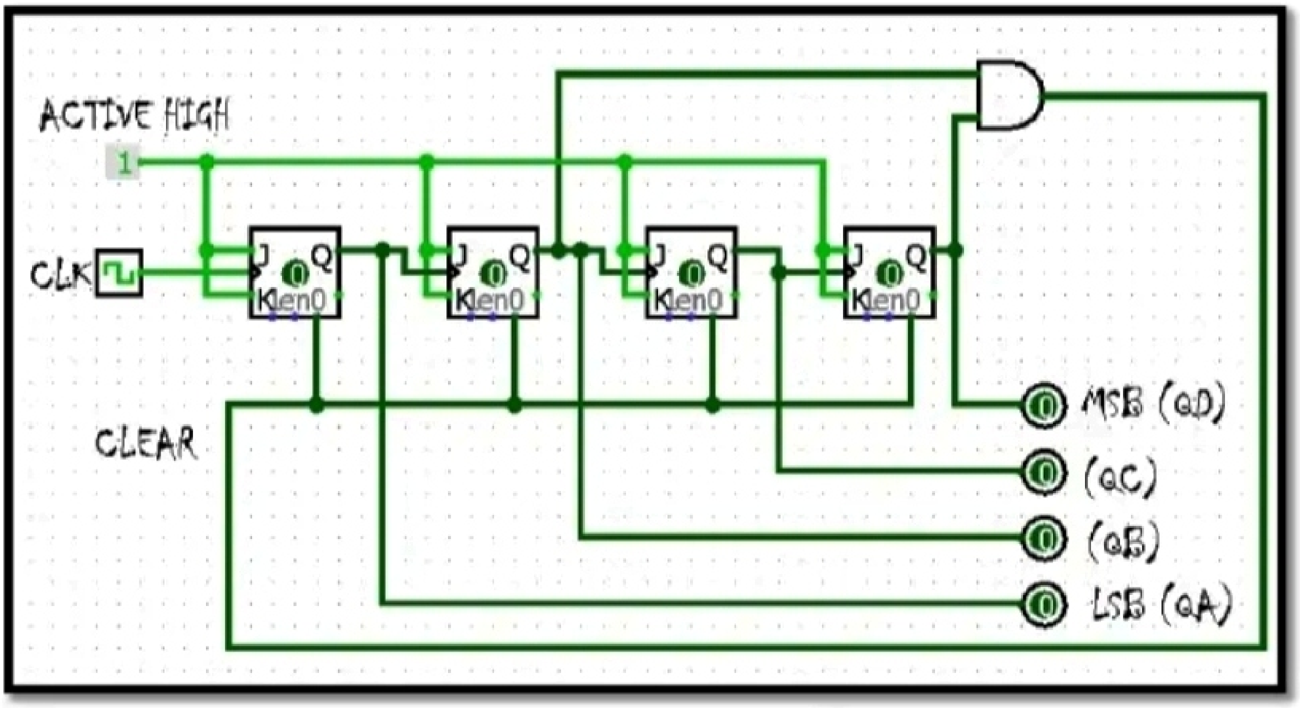
- Gather all the J-K flip-flops. Connect the first J-K flip-flop to clock pulse. Connect the main output of the first flip flop to the clock input of the second flip-flop. Repeat the same for the rest of the two flip-flops.
- Connect the inputs of all the J-K flip-flops to active high input because asynchronous counter yields the desired sequence of outputs in TOGGLE state. Connect every output of the J-K Flip-flop to (main output
  Q) to an LED. The LED connected to the main output of the first J-K Flip-flop (the one connected to the clock pulse) represents the LSB of the output and the LED connected to the last J-K Flip-flop represents the MSB of the output.
- The counter counts 16 states: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

  This is mod-16 ripple counter.
- For Mod-10 ripple counter, an additional truth table is required to implement the desired output. 4 columns specifying main output of every J-K flip-flop and one column specifying at which state the counter has to reset to the first state (0000 in this case). After the truth table is formed, CLR expression is obtained as a function of output of all the J-K Flip-Flops using K-map. The Boolean expression obtained is implemented with the help of logic gates and is connected to the clear pin of every flip-flop. Hence, the counter starts counting from 0000 and resets after 1001 state. This is MOD-10 ripple counter.

Data

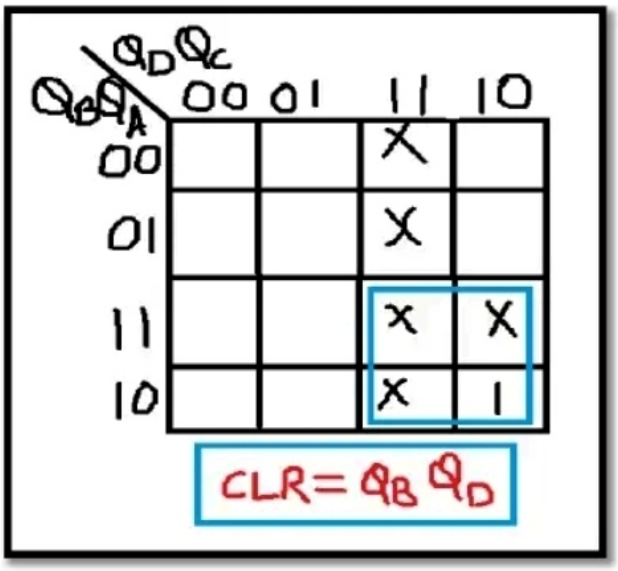
Label clearly what was measured or observed throughout the lab. Include all data tables and/or observation.

MOD-10 ASYNCHRONOUS COUNTER



TRUTH TABLE FOR MOD-10 ASYNCHRONOUS COUNTER

CLOCK	??	??	??	??	CLEAR
0 (INITIAL)	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	X
↓	1	1	0	0	X
↓	1	1	0	1	X
↓	1	1	1	0	X
↓	1	1	1	1	X



AP FOR CLEAR EXPRESSION

Discussion

If your teacher asks you to answer any discussion (post-lab) questions, this is the place. Write out the questions then answer it below. Be very thorough, detailed and precise. Answer the questions with complete thoughts. Assume the reader does not know anything about this topic.

Results and Conclusion

Summarize your results in the introductory sentence. Relate your results to your objective. Present the results in the easiest way for your reader to understand: graphs, tables, figures, observations, etc., which you make during the lab. All tables and figures should be accompanied by comments or discussions in the text of report; use a numbering system for identification of each one. All figures and tables must have numbers and captions. While the table captions should be placed over the table, figure captions should be placed below the figure.

4-bit asynchronous counter and mod-10 asynchronous counter was implemented successfully.

Experiement No : 11

AIM :

Study shift registers .

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 :

SHIFT REGISTER :

Shift register is a group of flip- flops that has the capability of storing and shifting the binary information. In digital system the binary datum are re quired to shift in the register from one position to next position. Shift register performs the logic operation shifting of binary data from one flip - flop to next flip- flop. In the shift register, the shifting operation is controlled by common clock pulse. All the flip- flops employed with shift register receives the clock pulse that helps to shift the data from one position to next.

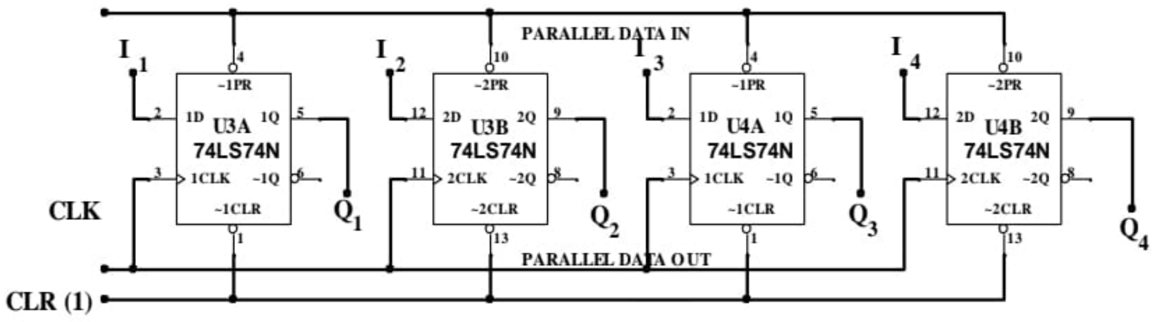
SERIAL-IN-SERIAL-OUT SHIFT REGISTER :

This shift register is constructed in the way of connecting the output of one flip- flop to the input of next flip- flop. All the flip- flops are connected with a common clock pulse. The binary inputs are applied in the input terminal of first flip- flop in series and the outputs are obtained from the output terminal of last flip- flop in series.

TRUTH TABLE OF SERIAL-IN-PARALLEL-OUT SHIFT REGISTER

Clock Pulse	Serial Input	Parallel Output			
	Data In	Q_1	Q_2	Q_3	Q_4
Initial Vaue	1011	0	0	0	0
After 1st Clk	1 1 1	1	0	0	0
After 2nd Clk	11	1	1	0	0
After 3rd Clk	1	0	1	1	0
After 4th Clk	-	1	0	1	1

LOGIC DIAGRAM: PARALLEL-IN-PARALLEL-OUT SHIFT REGISTER



TRUTH TABLE OF PARALLEL-IN-PARALLEL-OUT SHIFT REGISTER

Clock Pulse	Parallel Input				Parallel Output			
	I_1	I_2	I_3	I_4	Q_1	Q_2	Q_3	Q_4
Initial Value	0	0	0	0	0	0	0	0
After 1st Clk	1	0	0	0	1	0	0	0
After 2nd Clk	1	1	0	0	1	1	0	0
After 3rd Clk	0	1	1	0	0	1	1	0
After 4th Clk	1	0	1	1	1	0	1	1

SERIAL-IN-SERIAL-OUT SHIFT REGISTER :

This register accepts the binary data in series to provide the output in parallel. In logic diagram the flip- flops are connected similar to SISO shift register that logic diagram is constructed in the way of connecting the output of one flip- flop to the input of next flip- flop. The binary inputs are applied in the input terminal of first flip- flop in series and the outputs are obtained in parallel from the output terminal of each flip- flop employed with register. Hence this configuration is called serial -in-serial-out shift register.

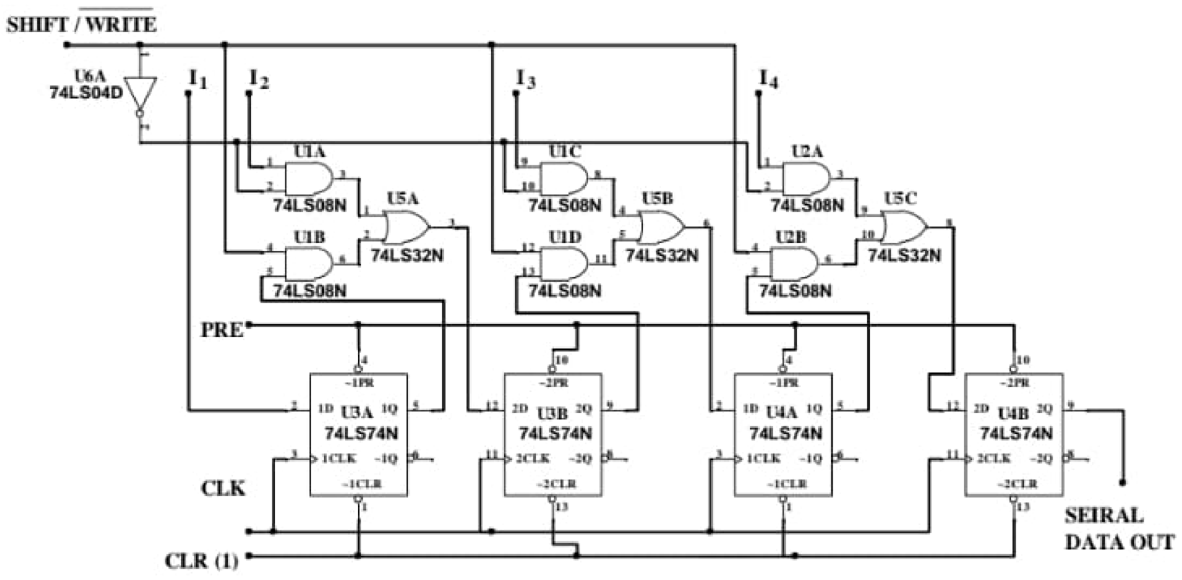
PARALLEL-IN-PARALLEL-OUT SHIFT REGISTER :

This register accepts the binary input in parallel to provide the output in parallel. All the flip- flops present in the register are triggered with common clock pulse. The binary inputs are applied in parallel to all the flip- flops and the outputs are obtained in parallel from the output terminal of each flip- flop employed with register. Hence this configuration is called parallel -in-parallel-out shift register.

PARALLEL-IN-SERIAL-OUT SHIFT REGISTER :

This register accepts the binary input in parallel to provide the output in series from right most flip-flop. The logic diagram is constructed by flip- flop and combination of logic gates. The combinational logic gates are functioning as a control circuit to load the input of shift the stored binary information. A control signal ‘shift’ is an active high signal i.e., when the control signal line is activated with logic high input a shift register performs the shifting operation. A control signal “Write” is an active low signal i.e., when the control signal line is activated with logic low input a shift register loads the newly appeared input signals into the flip- flops parallel. These logic operations are controlled by the combinational circuits.

LOGIC DIAGRAM: PARALLEL-IN-SERIAL-OUT SHIFT REGISTER



TRUTH TABLE OF PARALLEL-IN-PARALLEL-OUT SHIFT REGISTER

Control signals		Parallel Input				Serial Output
Clock Pulse	**SHIFT / $\overline{\text{WRITE}}$**	**I_1**	**I_2**	**I_3**	**I_4**	**Q_4**
0	X	0	0	0	0	No Change (Initial state)
1	0	0	0	0	1	1
1	0	1	1	0	0	0
1	1	1	1	0		0
1	1	1	1			0
1	1	1				1
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:

Experiement No : 12

AIM :

Objective: Find 1's & 2's complement of a 8 bit number

Apparatus required: 8085 Simulator, PC.

Program:

1's complement of 8 bit number

```
LDA 2200H          Get the number
CMA                Complement number
STA 2300H           Store the result
HLT                Terminate program execution
```

2's complement of a no

```
LDA 2200 H         Get the number
CMA                Complement the number
ADI 01H           ; Add one in the number
STA 2300H          Store the result
HLT                Terminate program execution
```

Result: 1's & 2's complement of a 8 bit number is executed.

Experiement No : 13(A)

Objective : Addition of two 8-bit numbers.

Apparatus Required: 8085 Simulator, PC.

Program:

```
        MVI C 00             Initialize C register to 00
        LDA  4150             Load the value to Accumulator.
        MOV  B A              Move the content of Accumulator to B register.
        LDA 4151              Load the value to Accumulator.
        ADD  B               Add the value of register B to Accumulator.
        JNC   LOOP           Jump on no carry.
        INR  C                Increment value of register C.
LOOP:   STA  4152             Store the value of Accumulator.
        MOV A C              Move content of Register C to  Accumulator
        STA 4153              Store the value of Accumulator
        HLT
```

Obsevation:

```
      Input:        80 (4150)
                    80 (4251)
      Output:       00 (4252)
                    01 (4253)
```

Result: Thus the program to add two 8 bit numbers was executed.

Experiement No : 13(B)

Objective: Subtraction of two 8-bit numbers.

Apparatus required: 8085 Simulator, PC.

.

Program:

```
      MVI C 00          Initialize C register to 00
      LDA  4150         Load the value to Accumulator.
      MOV  B A          Move the content of Accumulator to B register.
      LDA 4151          Load the value to Accumulator.
      SUB  B            Add the value of register B to Accumulator.
      JNC   LOOP        Jump on no carry.
      CMA               Complement Accumulator Content
      INR A             Increment value of register C.
      INR  C            Increment value of register C.
LOOP: STA  4152         Store the value of Accumulator.
      MOV A C           Move content of Register C to  Accumulator
      STA 4153          Store the value of Accumulator
      HLT
```

Observation:

Input:	06 (4150)
	02 (4251)
Output:	04 (4252)
	01 (4253)

Result: Thus the program to subtract two 8 bit numbers was executed.

Experiement No : 14

Objective: Find largest Number From an array.

Apparatus required: 8085 Simulator, PC.

Program:

```
          LXI    H, 4200   Set pointer for array
          MOV    B, M      Load the count
          INX    H
          MOV    A, M      Set  1st element as largest data
          DCR    B         Decrement the count
   LOOP:  INX    H
          CMP    M         If A reg.>M go to AHEAD
          JNC    AHEAD
          MOV    A, M      Set the new value as largest
  AHEAD:  DCR    B
          JNC    LOOP      Repeat comparisons till count=0
          STA    4300      Store the largest value at 4300
          HLT
```

Observation:

```
        Input    05 (4200)……Array size
                 0A (4201)
                 F1 (4202)
                 1F (4203)
                 26 (4204)
                 FE (4205)

        Output   FE (4300)
```

Result: Thus the program to find largest number in an array was executed.

Experiement No : 15

Objective: Transfer Block of data bytes from one memory location to another .

Apparatus required: 8085 Simulator, PC.

Program:

```
        MVI C, 0AH          Initialize counter
        LXI H, 2200H        Initialize source memory pointer
        LXI D, 2300H        Initialize destination memory pointer
BACK    MOV A, M            Get byte from source memory block
        STAX D              Store byte in the destination memory block
        INX H               Increment source memory pointer
        INX D               Increment destination memory pointer
        DCR C               Decrement counter
       JNZ   BACK           If counter  0 repeat
       HLT                  Terminate program execution
```

Result: Transfer Block of data bytes from one memory location to another is executed.

Experiment No :- 16 B)1

AIM:

Traffic Light Controller using 8255

INTRODUCTION

In this modern life, the number of vehicles increase more day by day. The increase of vehicle may cause accidents and other problems on the road. Controlling traffic at regular intervals of time with accuracy and uniformity has become a necessity to avoid accidents, discomfort of drivers. The microprocessor controls the traffic signals very effectively and with accurate timings.

CIRCUIT DESCRIPTION

In this Traffic Light Module one square is shown which has four ends called East, W est, North & South. The Traffic Light Module is interfaced with Port A and Port C of 8255. Port A is connected with North and South LEDs. Port C is connected with East and West LEDs. Each end has four LEDs called STOP, START, GO STRAIGHT & GO LEFT.

PORT-A

NORTH				SOUTH			
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
GO-LEFT	GO-STR	START	STOP	GO-LEFT	GO-STR	START	STOP

PORT-B

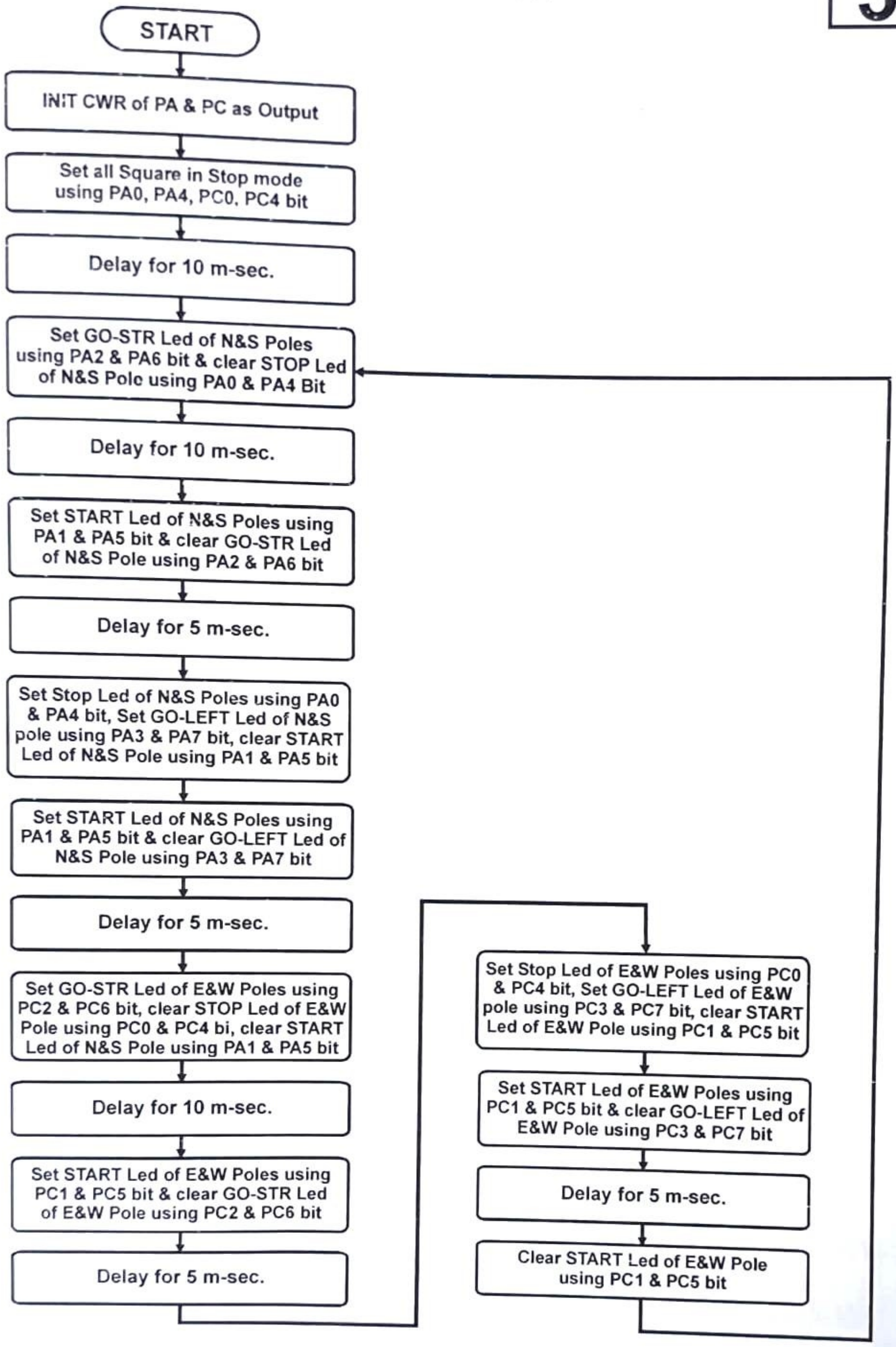
EAST				WEST			
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
GO-LEFT	GO-STR	START	STOP	GO-LEFT	GO-STR	START	STOP

HARDWARE INSTALLATION

Connect Traffic Light Controller Module to 8255-l of 8051/8085/8086 Trai ner kit through 26 pin FRC Cable.
Be sure about the direction of the cable i.e. Pin No. 1 of Module should be connected to Pin No. 1 of 8255 connector.
Connect +5V, GND from the Trainer kit (+5V & GND signals are available in the 25 & 26 pin of FRC 8255-l Connector)

FLOWCHART

3


START
INIT CWR of PA & PC as Output
Set all Square in Stop mode using PA0, PA4, PC0, PC4 bit
Delay for 10 m-sec.
Set GO-STR Led of N&S Poles using PA2 & PA6 bit & clear STOP Led of N&S Pole using PA0 & PA4 Bit
Delay for 10 m-sec.
Set START Led of N&S Poles using PA1 & PA5 bit & clear GO-STR Led of N&S Pole using PA2 & PA6 bit
Delay for 5 m-sec.
Set Stop Led of N&S Poles using PA0 & PA4 bit, Set GO-LEFT Led of N&S pole using PA3 & PA7 bit, clear START Led of N&S Pole using PA1 & PA5 bit
Set START Led of N&S Poles using PA1 & PA5 bit & clear GO-LEFT Led of N&S Pole using PA3 & PA7 bit
Delay for 5 m-sec.
Set GO-STR Led of E&W Poles using PC2 & PC6 bit, clear STOP Led of E&W Pole using PC0 & PC4 bi, clear START Led of N&S Pole using PA1 & PA5 bit
Delay for 10 m-sec.
Set START Led of E&W Poles using PC1 & PC5 bit & clear GO-STR Led of E&W Pole using PC2 & PC6 bit
Delay for 5 m-sec.
Set Stop Led of E&W Poles using PC0 & PC4 bit, Set GO-LEFT Led of E&W pole using PC3 & PC7 bit, clear START Led of E&W Pole using PC1 & PC5 bit
Set START Led of E&W Poles using PC1 & PC5 bit & clear GO-LEFT Led of E&W Pole using PC3 & PC7 bit
Delay for 5 m-sec.
Clear START Led of E&W Pole using PC1 & PC5 bit

PROGRAMS

4

PROGRAM FOR MICRO-85 LCD TRAINER KIT

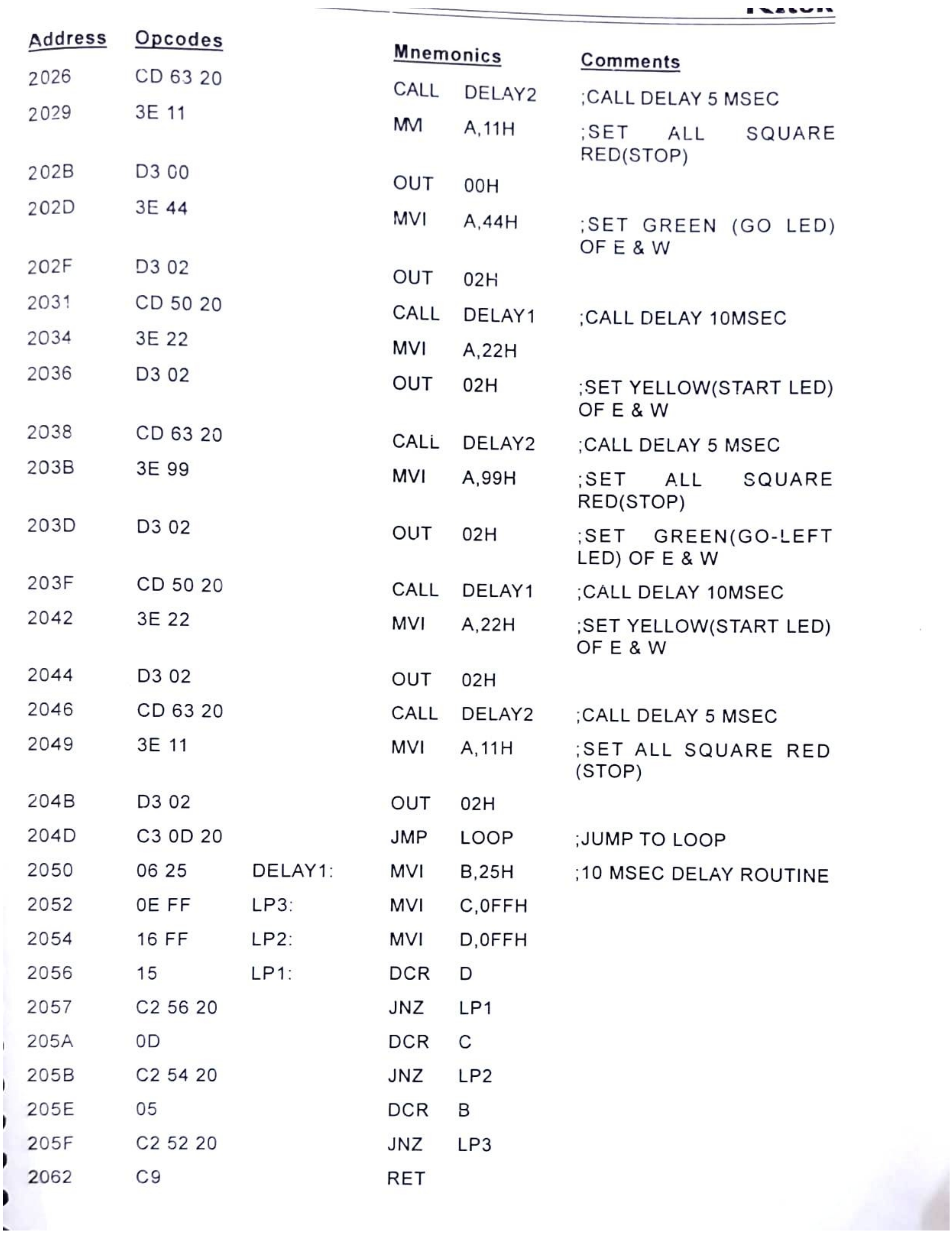
8255 Port Address

Port A	-	00H
Port B	-	01H
Port C	-	02H
Control Word	-	03H

Program

This program controls Traffic Light of One Square. By changing the delay between Two Signals one can change the Speed of Traffic.

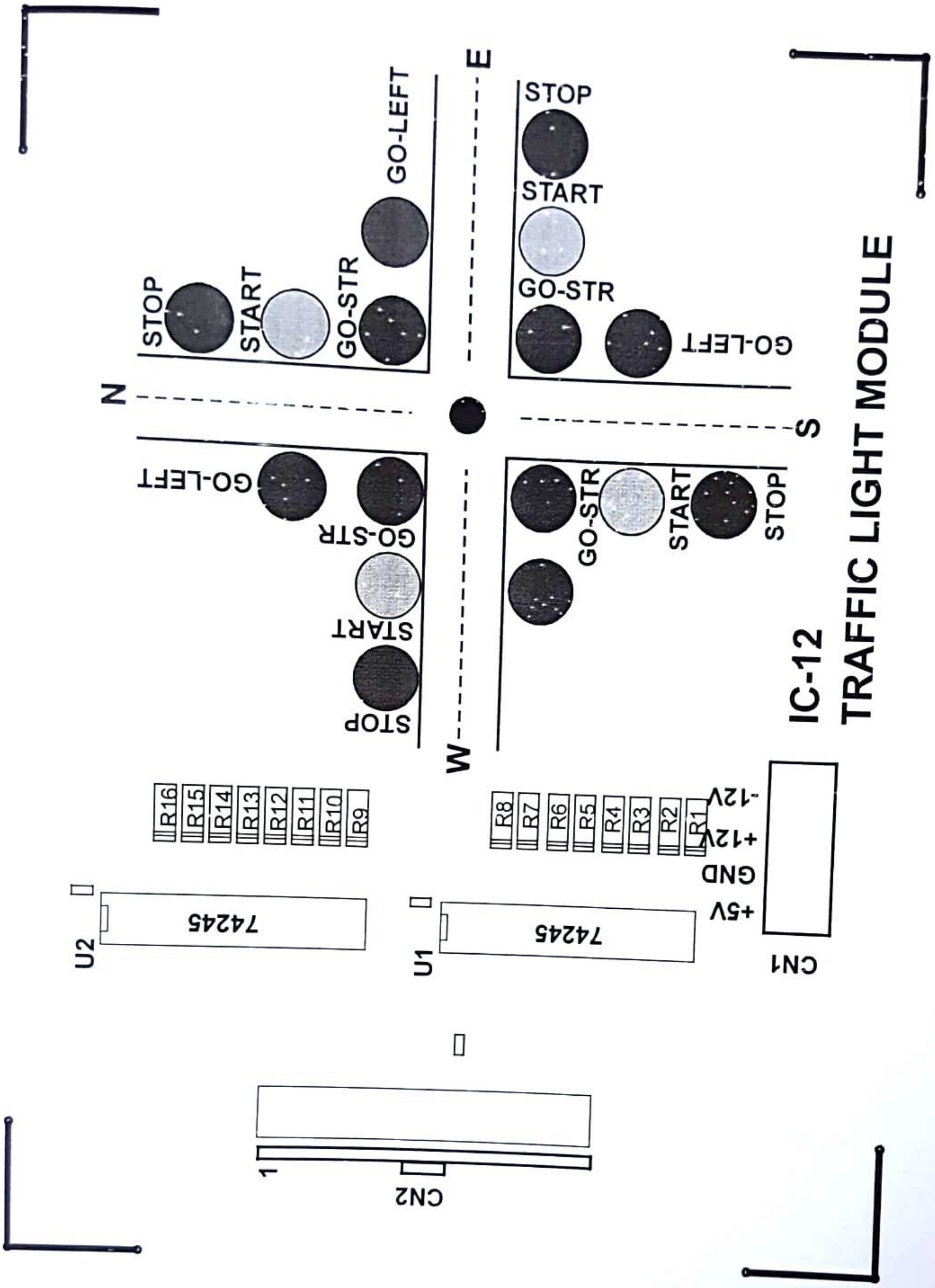
Address	Opcodes		Mnemonics		Comments
2000	3E 80		MVI	A,80H	;SET ALL PORTS AS OUTPUT
2002	D3 03		OUT	03H	
2004	3E 11		MVI	A,11H	;SET ALL SQUARE RED(STOP)
2006	D3 00		OUT	00H	
2008	D3 02		OUT	02H	
200A	CD 50 20		CALL	DELAY1	;CALL DELAY 10MSEC
200D	3E 44	LOOP:	MVI	A,44H	;SET GREEN (GO LED) OF N & S
200F	D3 00		OUT	00H	;SET RED (STOP LED) OF E & W
2011	CD 50 20		CALL	DELAY1	;CALL DELAY 10MSEC
2014	3E 22		MVI	A,22H	
2016	D3 00		OUT	00H	;SET YELLOW(START LED) OF N & S
2018	CD 63 20		CALL	DELAY2	;CALL DELAY 5 MSEC
201B	3E 99		MVI	A,99H	;SET ALL SQUARE RED(STOP)
201D	D3 00		OUT	00H	;SET GREEN(GO-LEFT LED) OF N & S
201F	CD 50 20		CALL	DELAY1	;CALL DELAY 10MSEC
2022	3E 22		MVI	A,22H	
2024	D3 00		OUT	00H	;SET YELLOW(START LED) OF N & S



Address	Opcodes		Mnemonics		Comments
2026	CD 63 20		CALL	DELAY2	;CALL DELAY 5 MSEC
2029	3E 11		MVI	A,11H	;SET ALL SQUARE RED(STOP)
202B	D3 00		OUT	00H	
202D	3E 44		MVI	A,44H	;SET GREEN (GO LED) OF E & W
202F	D3 02		OUT	02H	
2031	CD 50 20		CALL	DELAY1	;CALL DELAY 10MSEC
2034	3E 22		MVI	A,22H	
2036	D3 02		OUT	02H	;SET YELLOW(START LED) OF E & W
2038	CD 63 20		CALL	DELAY2	;CALL DELAY 5 MSEC
203B	3E 99		MVI	A,99H	;SET ALL SQUARE RED(STOP)
203D	D3 02		OUT	02H	;SET GREEN(GO-LEFT LED) OF E & W
203F	CD 50 20		CALL	DELAY1	;CALL DELAY 10MSEC
2042	3E 22		MVI	A,22H	;SET YELLOW(START LED) OF E & W
2044	D3 02		OUT	02H	
2046	CD 63 20		CALL	DELAY2	;CALL DELAY 5 MSEC
2049	3E 11		MVI	A,11H	;SET ALL SQUARE RED (STOP)
204B	D3 02		OUT	02H	
204D	C3 0D 20		JMP	LOOP	;JUMP TO LOOP
2050	06 25	DELAY1:	MVI	B,25H	;10 MSEC DELAY ROUTINE
2052	0E FF	LP3:	MVI	C,0FFH	
2054	16 FF	LP2:	MVI	D,0FFH	
2056	15	LP1:	DCR	D	
2057	C2 56 20		JNZ	LP1	
205A	0D		DCR	C	
205B	C2 54 20		JNZ	LP2	
205E	05		DCR	B	
205F	C2 52 20		JNZ	LP3	
2062	C9		RET		

Address	**Opcodes**		**Mnemonics**		**Comments**
2063	06 05	DELAY2:	MVI	B,05H	;5 MSEC DELAY ROUTINE
2065	0E FF	LP6:	MVI	C,0FFH	
2067	16 FF	LP5:	MVI	D,0FFH	
2069	15	LP4:	DCR	D	
206A	C2 69 20		JNZ	LP4	
206D	0D		DCR	C	
206E	C2 67 20		JNZ	LP5	
2071	05		DCR	B	
2072	C2 65 20		JNZ	LP6	
2075	C9		RET		

BLOCK DIAGRAM



Experiement No : 16 B 2

AIM

To write a program to generate square wave and triangular wave using microprocessor 8085 by interfacing with a programmable peripheral interface 8255.

THEORY

The Intel 8255 Programmable Peripheral Interface (PPI) chip is a peripheral chip made in DIP 40 and PLCC 44 pins encapsulated versions. The 8255 has 24 input/output pins in all. These are divided into three 8-bit ports. Port A and port B can be used as 8-bit input/output ports. Port C can be used as an 8-bit input/output port or as two 4-bit input/output ports or to produce handshake signals for ports A and B.

The three ports are further grouped as follows:

1. Group A consisting of port A and upper part of port C.
2. Group B consisting of port B and lower part of port C.

Eight data lines (D0 - D7) are available (with an 8-bit data buffer) to read/write data into the ports or control register under the status of the RD (pin 5) and WR (pin 36), which are active low signals for read and write operations respectively. The address lines A_1 and A_0 allow to successively access any one of the ports or the control register as listed below:

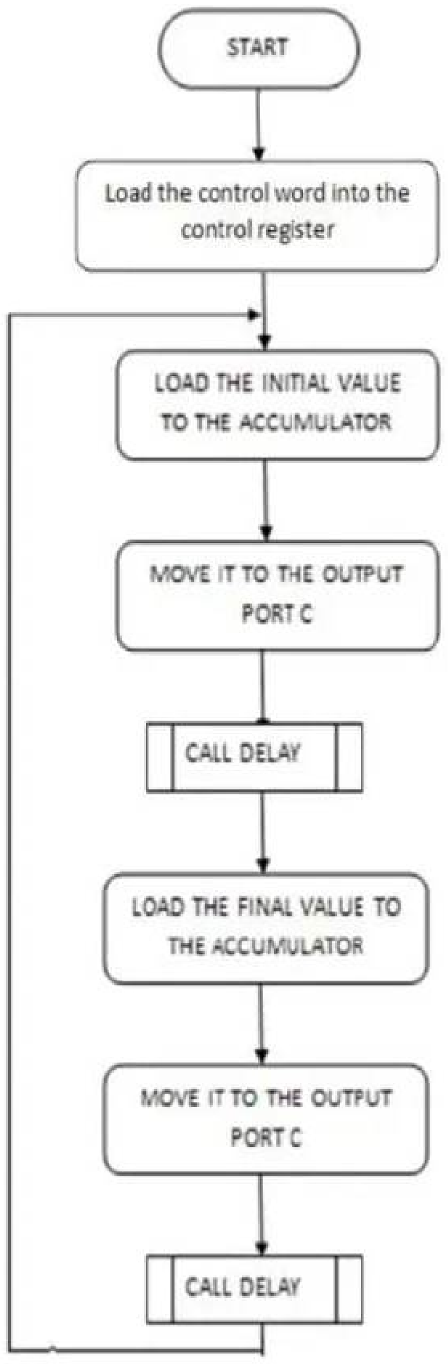
The control signal CS (pin 6) is used to enable the 8255 chip. It is an active low signal, i.e., when CS = '0', the 8255 is enabled. The RESET input (pin 35) is connected to the RESET line of system like 8085, 8086, etc., so that when the system is reset, all the ports are initialized as input lines. This is done to prevent 8255 and/or any peripheral connected to it, from being destroyed due to mismatch of ports. As an example, consider an input device connected to 8255 at port A. If from the previous operation, port A is initialized as an output port and if 8255 is not reset before using the current configuration, then there is a possibility of damage of either the input device connected or 8255 or both since both 8255 and the device connected will be sending out data.

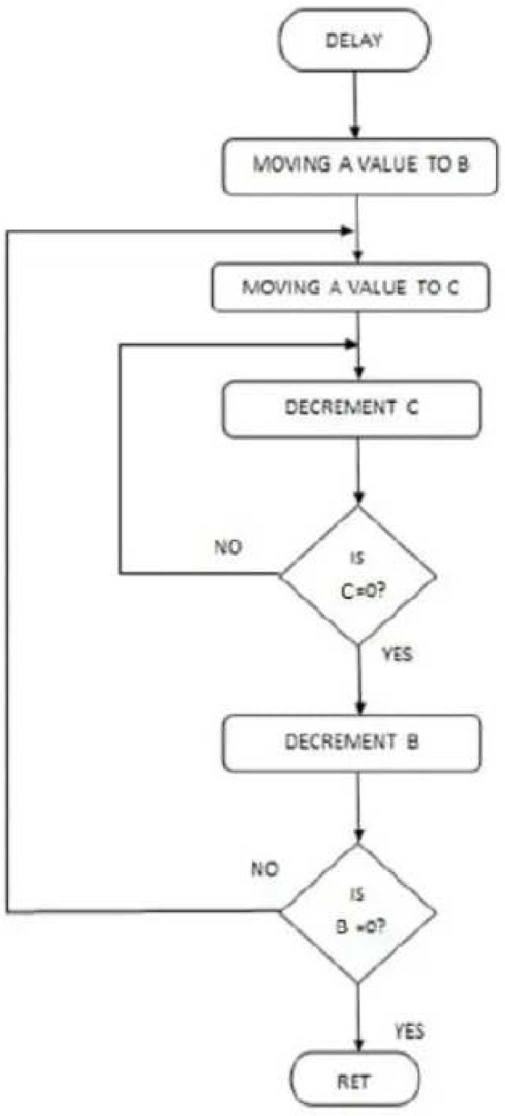
The control register or the control logic or the command word register is an 8-bit register used to select the modes of operation and input/output designation of the ports.

ALGORITHM

1. Load the control word into the control register
2. Load the initial value to accumulator and move it to output port.
3. Call the delay program.
4. Load the final value to accumulator and move it to output port.
5. Call the delay program.
6. Repeat steps 2 to 5

FLOW CHART

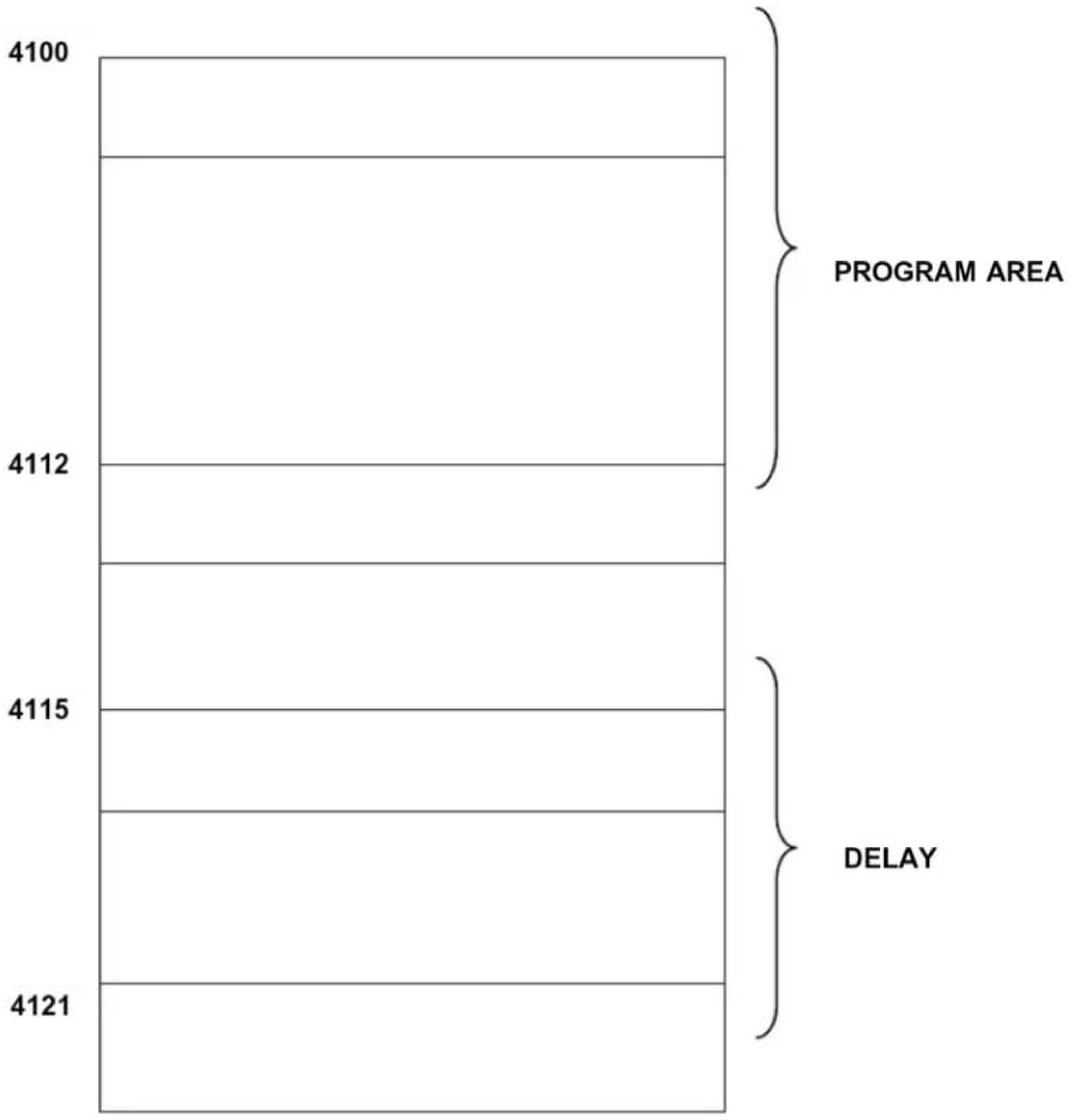

START
Load the control word into the control register
LOAD THE INITIAL VALUE TO THE ACCUMULATOR
MOVE IT TO THE OUTPUT PORT C
CALL DELAY
LOAD THE FINAL VALUE TO THE ACCUMULATOR
MOVE IT TO THE OUTPUT PORT C
CALL DELAY


DELAY
MOVING A VALUE TO B
MOVING A VALUE TO C
DECREMENT C
NO
IS C=0?
YES
DECREMENT B
NO
IS B =0?
YES
RET

CODING FOR SQUARE WAVE GENERATION

Address	Op Code	Label	Instructions	COMMENTS
4100	3E, 80		MVI A,80	;Control word is initialized
4102	D3, 0F		OUT 0F	;Control register
4104	3E, FF	L	MVI A, 00	; load the initial value to the accumulator
4106	D3,0C		OUT 0C	;output port C
4108	CD,		CALL DELAY	;Call the delay program
410B	3E, 00		MVI A, FF	; load the final value to the accumulator
410D	D3,0C		OUT 0C	; output port C
410F	CD,		CALL DELAY	; Call the delay program
4112	C3		JMP L	;Jump
4115	06	DELAY	MVI B, 05	;move a value to B
4117	0E	L1	MVI C, FF	;move a value to C
4119	0D	L2	DCR C	;decrement the value in C register
411A	C2		JNZ L2	;jump if no zero to loop 2
411D	05		DCR B	;decrement the value in B register
411E	C2		JNZ L1	;jump if no zero to loop 1
4121	C9		RET	;return

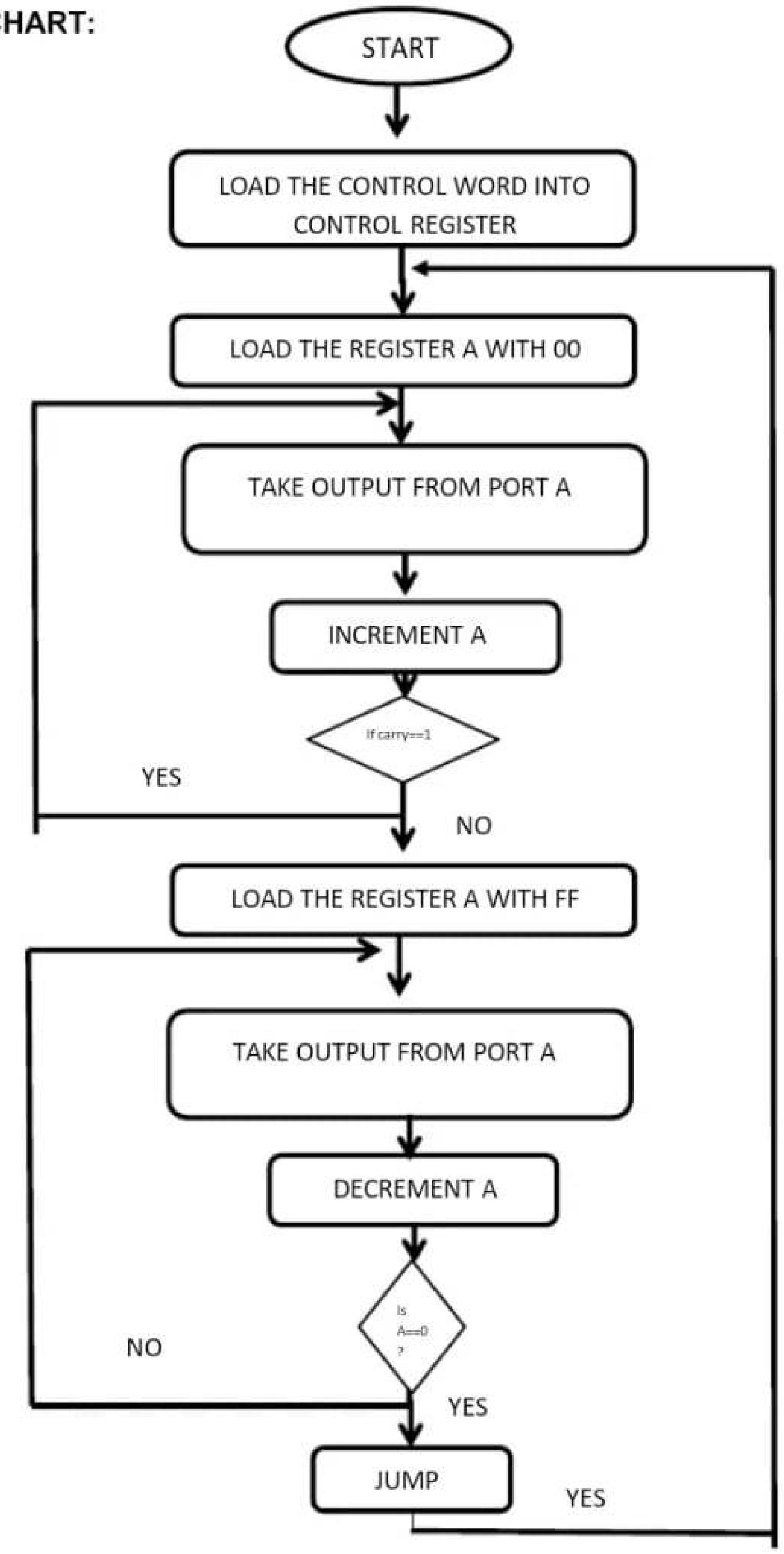
MEMORY MAP FOR SQUARE WAVE GENERATION



ALGORITHM

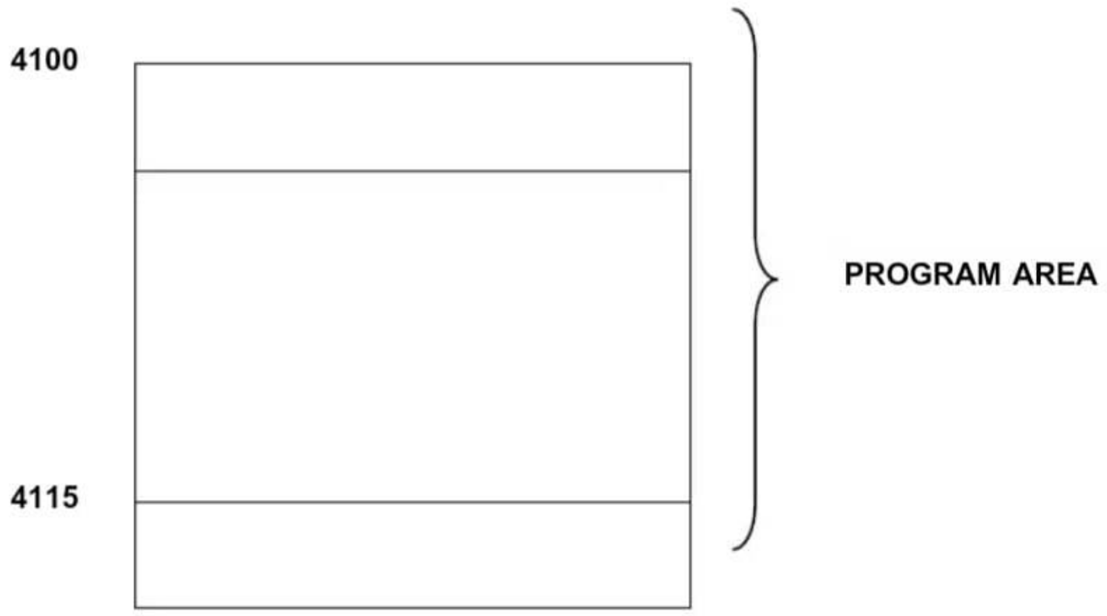
1) Load the control word into the control register
2) Move 00 into A register
3) Move the value of accumulator to port A
4) Increment the value in A register by 1.
5) If there is no carry repeat from step 3 or else continue with the next step
6) Move FF into A register.
7) Move the value of accumulator to port A
8) Decrement the value at A register by 1.
9) If the value in the A register is not zero then repeat step 8 to step 9 or else continue with the next step.
10) Go to step 2

FLOW CHART:


START
LOAD THE CONTROL WORD INTO CONTROL REGISTER
LOAD THE REGISTER A WITH 00
TAKE OUTPUT FROM PORT A
INCREMENT A
If carry==1
YES
NO
LOAD THE REGISTER A WITH FF
TAKE OUTPUT FROM PORT A
DECREMENT A
Is A==0 ?
NO
YES
JUMP
YES

CODING FOR TRIANGULAR WAVE GENERATION

ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENT
4100	3E,80		MVI A,80	;the value 80 is loaded to accumulator
4103	D3,0F		OUT 0F	;The value in A is loaded into 0F
4105	3E,00	L	MVI A,00	; the value 0 is moved into A register.
4107	D3,0C	L1	OUT A	;the output is taken from port A.
4109	3C		INR A	;the register value A is incremented by 1.
410A	C2,05,41		JC L1	;the zero flag is checked.
410D	3E,FF		MVI A,FF	; the value FF is moved into A register.
410F	D3,0C	L2	OUT A	;the output is taken from port A.
4111	3D		DCR A	;the register value A is decremented by1.
4112	C2,0D,41		JNZ L2	;the zero flag is checked.
4115	C3,00,41		JMP L	;jump to L



OUTPUT

When the above code is executed a triangular wave is generated from the output ports of 8255.

RESULT

Thus square wave form and triangular waveform were generated using peripheral programmable device 8255