DIGITAL ELECTRONICS & MICROPROCESSOR LAB

GOVERNMENT POLYTECHNIC, RAGADI, JAJPUR, ODISHA

BY: NIHARIKA SETHY . LECT ETC

EXPT. 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 & NOR gates and verify truth table.
3	Implement half adder and full adder using logic gates.
4	Implement half subtracter and full subtracter using logic gates.
5	Implement a 4 bit Binary to Gray code converter.
6	Implement a single bit digital comparator.
7	Study Multiplexer and demultiplexer.
8	Study of flip-flops. i) S-R flip flop ii) J-K flip-flop iii)D flip-flop iv) T flip-flop
9	To design IC 74193 as a up/down counter
10	To design a decade counter.
11	Study shift registers.
	Write 8085 assembly language program for one's complement of an 8-bit numbers
12	Write 8085 assembly language program for two's complement of an 8-bit numbers
13	.Addition of 8-bit number using 8085 MP kit.
15	Subtraction of 8 bit number resulting 8/16 bit number
16	Program for Decimal Addition of Two 8-Bit Numbers and Sum is 16 Bit
17	Program for Decimal Subtraction of Two 8-Bit Numbers
18	To find the larger among two numbers using 8085 Microprocessor
19	To find the largest element in an array of size 'n' using 8085 Microprocessor
20	Write a program to control the traffic light system using 8085 & 8255 ppi

DO'S and DON'TS in Laboratory

1) Check for appropriate power supply before connecting to the equipment.
2) Decide the appropriate range of the measuring instruments on the basis of quantity to be measured.
3) Make the connections without connecting the leads to the supply.
4) Re-check the connections and show it to the teacher /instructor before switching-on the power supply to the circuit.
5) Energize the circuit only with the permission of the teacher/instructor.
6) After the experiment, disconnect the connections and put back the connecting wires/leads at appropriate place.
7) Return all the apparatus to the lab-staff.
8) In case of shock, switch-off the power supply immediately.
9) Strictly follow the procedure given with the respective experiments.
10) Avoid loose connections.
11) Don't touch the main power supply leads with bare hand and avoid body earth.
12) Don't use the mobile phones during laboratory.

EXPERIMENT NO: 1

AIM: Verify truth tables of AND, OR,NOT, NOR,NAND,XOR, XNOR gates
Using ICs& simplifications of Boolean gates

THEORY:

AND Gate:
The AND operation is defined as the output as (1) one if and only if all the inputs are (1) one.
7408 is the two Inputs AND gate IC.A&B are the Input terminals & Y is the Output terminal.
Y = A.B

OR Gate:
The OR operation is defined as the output as (1) one if one or more than 0
Inputs are (1) one. 7432 is the two Input OR gate IC. A&B are the input terminals & Y is the Output terminal.
Y = A + B

NOT GATE:
The NOT gate is also known as Inverter. It has one input (A) & one
Output (Y). IC No. is 7404. Its logical equation is,
Y = A NOT B,
Y = A'

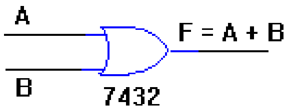
NAND GATE:
The IC no. for NAND gate is 7400. The NOT-AND operation is known
as NAND operation. If all inputs are 1 then output produced is 0. NAND gate is inverted AND gate.
Y = (A. B)'

NOR GATE:
The NOR gate has two or more input signals but only one output signal. IC
7402 is two I/P IC. The NOT- OR operation is known as NOR operation. If all the inputs are 0 then the O/P is 1. NOR gate is inverted OR gate.
Y = (A+B)'

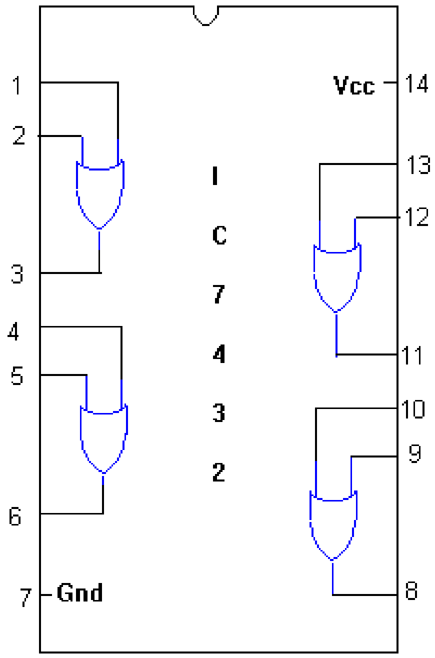
EX-OR GATE:
The EX-OR gate can have two or more inputs but produce one output.
7486 is two inputs IC. EX-OR gate is not a basic operation & can be performed using basic gates.
Y = A E-XOR B

Logic Symbol of Gates

SYMBOL :

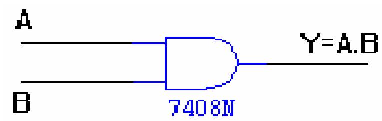


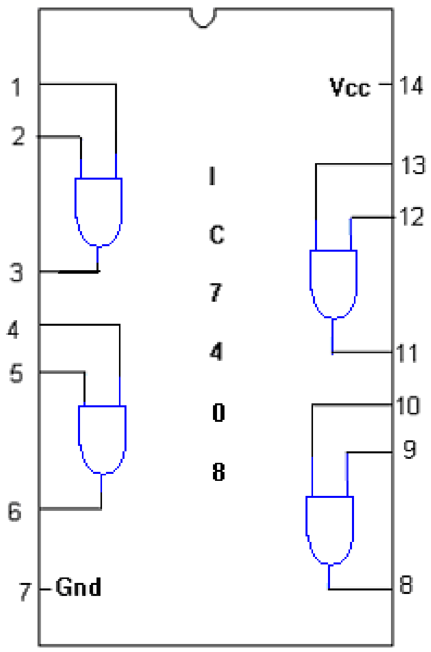
PIN DIAGRAM :



TRUTH TABLE

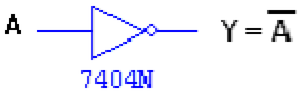
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

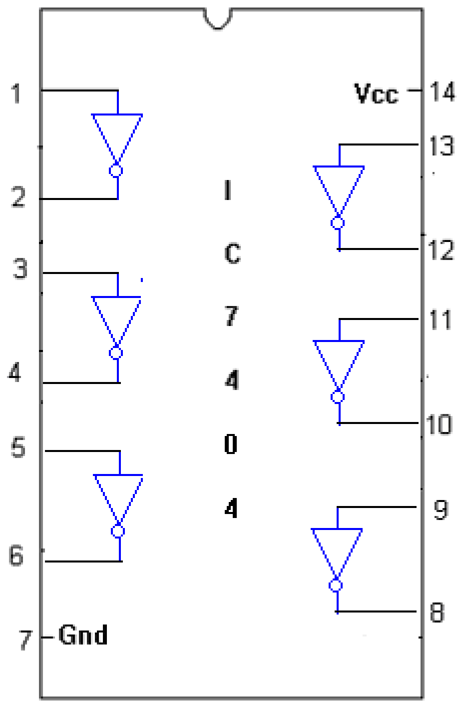




TRUTH TABLE

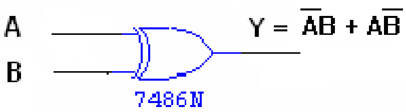
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

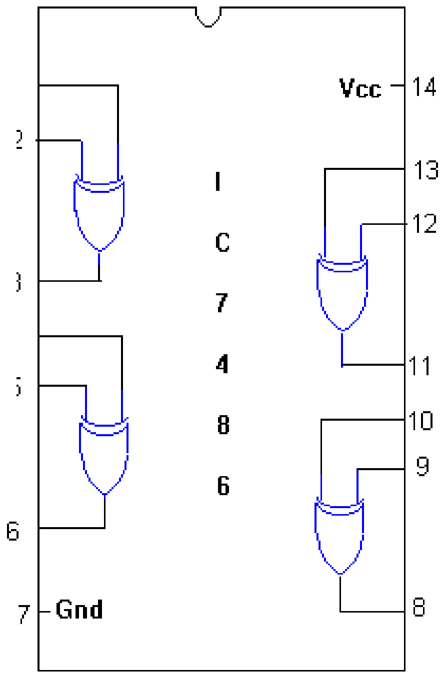




TRUTH TABLE :

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

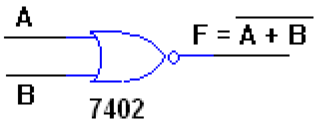




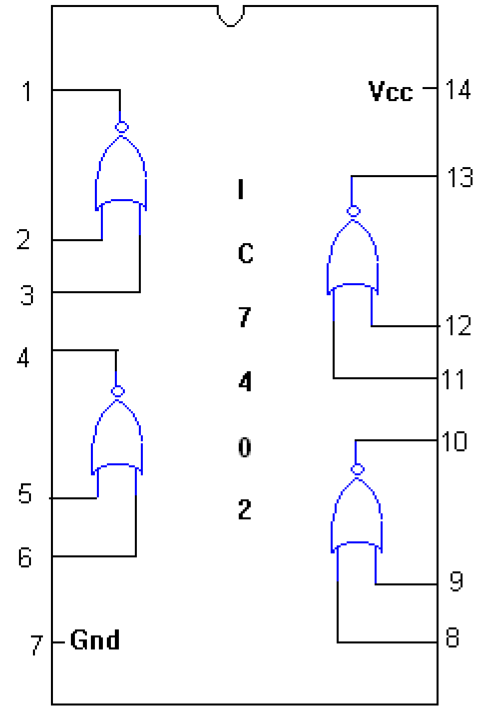
TRUTH TABLE :

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

SYMBOL :



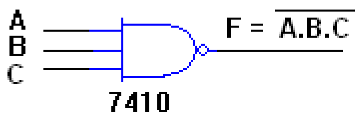
PIN DIAGRAM :



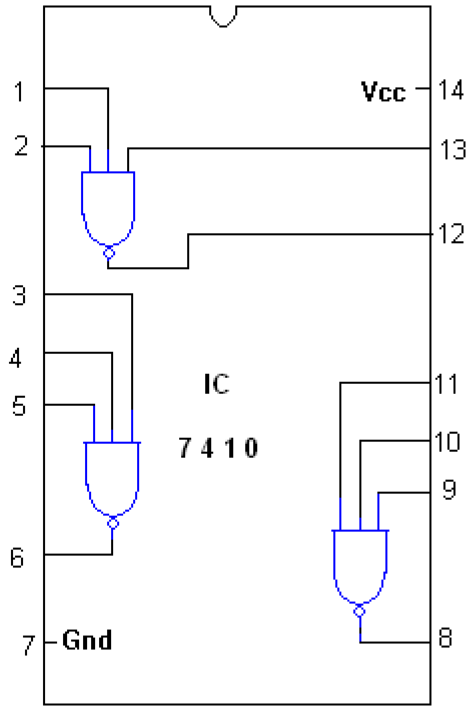
TRUTH TABLE

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

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

A	B	C	$\overline{A.B.C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

PROCEDURE:

(a) Fix the IC’s on breadboard & gives the supply.

(b) Connect the +ve terminal of supply to pin14 & -ve to pin7.

(c) Give input at pin1, 2 & take output from pin3.It is same for all except NOT & NOR IC.

(d) For NOR, pin1 is output & pin2&3 are inputs.

(e) For NOT, pin1 is input & pin2 is output.

(f) Note the values of output for different combination of inputs & draw the ckt

PRECAUTIONS:

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The Vcc and ground should be aapplied carefully at the specified pin only.

RESULTS: Hence the truth table of basic logic gates are verified.

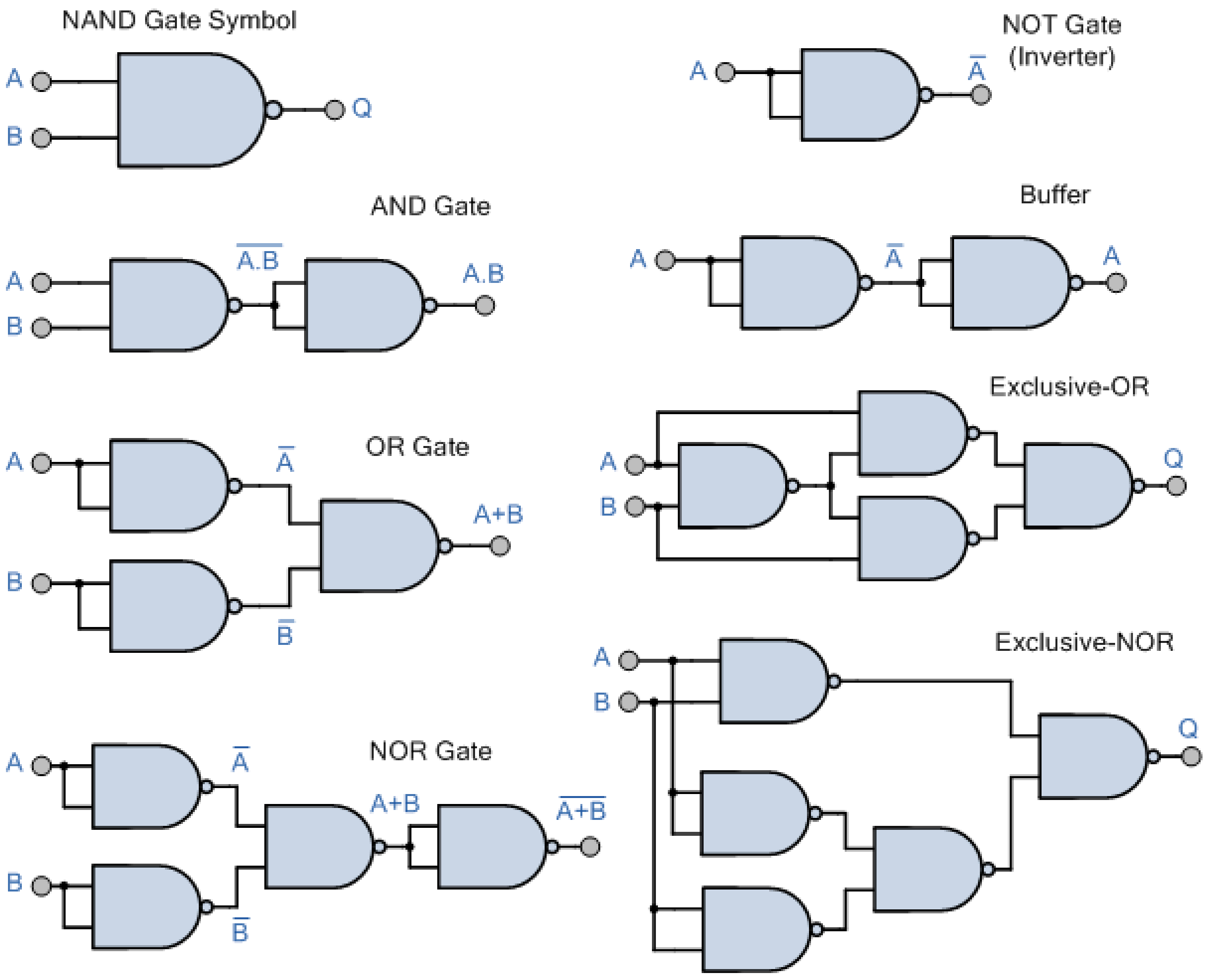
EXPERIMENT NO: 2

AIM: -Realize Basic gates (AND, OR, NOT) From Universal Gates (NAND & NOR).

APPARATUS: -L.E.D., Bread-Board, I.C.‟s, Wires, “5.0” V dc. Supply etc.

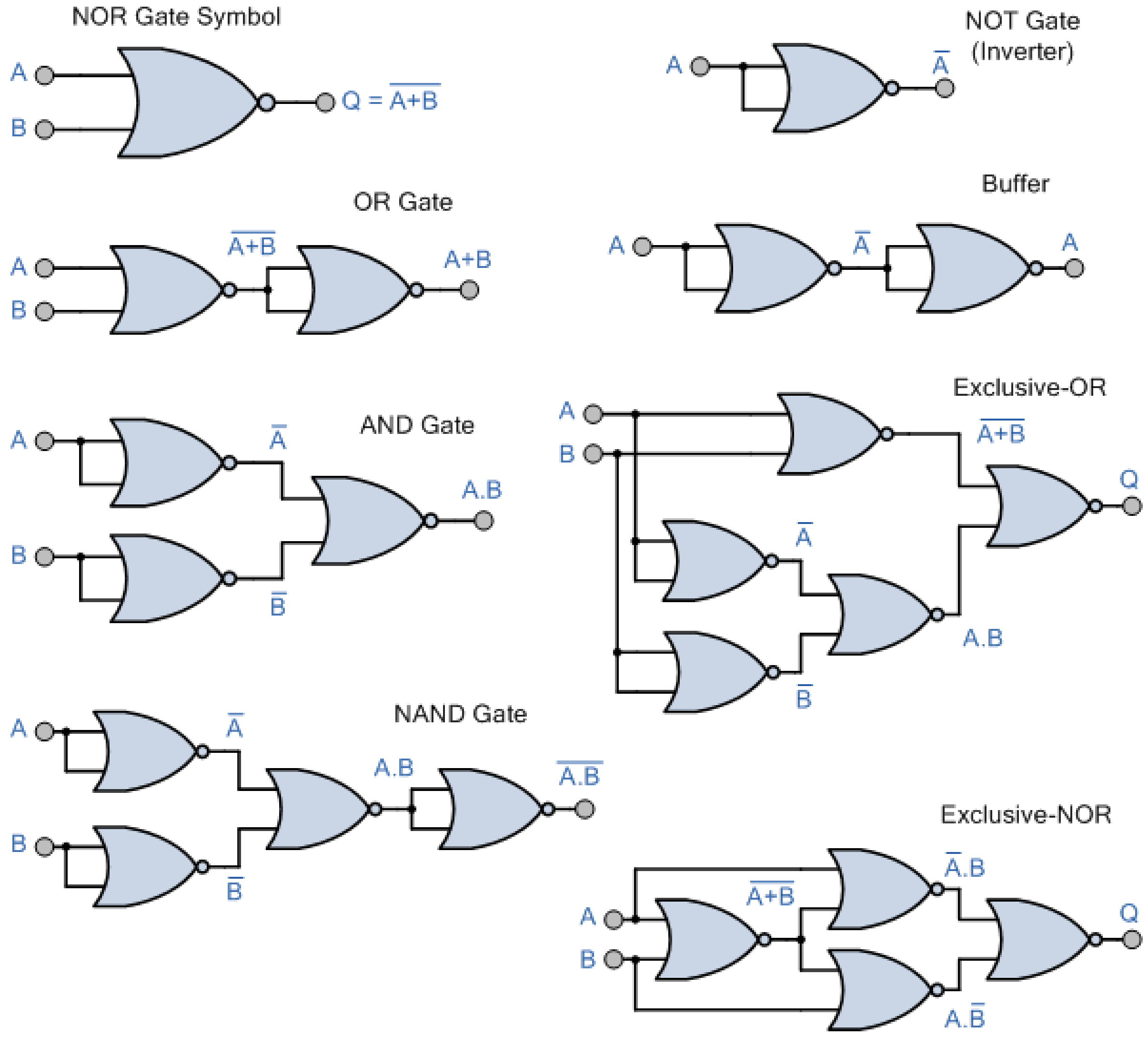
THEORY: - NAND Gates to AND, OR, NOT Gates: -

NAND gates is Universal gate. The Basic gates AND, OR, NOT, EXOR can be realized from it. The Boolean equations and logic diagrams are as follows:



NOR gate is also a Universal gate. The Basic gates AND, OR, NOT can be realized from it.

The Boolean equations and logical diagrams are as follows:



NAND TO AND Gate:

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

NAND to OR Gate

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

NAND to NOT Gate

input	**output**
A	Y
0	1
1	0

NAND to EXOR Gate

Inputs	**Inputs**	**Output**
0	0	0
0	1	1
1	0	1
1	1	0

NOR to AND Gate

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

NOR to OR Gate

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

NOR to NOT Gate

input	**output**
A	Y
0	1
1	0

NOR to EXOR gate

input	**input**	**output**
0	0	0
0	1	1
1	0	1
1	1	0

RESULTS:

The realization of basic gates(AND,OR, NOT) from universal gates (NAND& NOR) is successful. The corresponding truth-tables are also verified.

PRECAUTIONS: -

1) Supply should not exceed 5v.

2) Connections should be tight and easy to inspect.

3) Use L.E.D. with proper sign convention and check it before connecting in circuit.

EXPERIMENT NO: 3

AIM: - Construct and verify operation of Half Adder & Full Adder.

APPARATUSREQUIRED: Power supply, IC's, Digital Trainer, Connecting leads.

THEORY: We are familiar with ALU, which performs all arithmetic and logic operation but ALU doesn't perform/ process decimal numbers. They process binary numbers.

Half Adder: It is a logic circuit that adds two bits. It produces the O/P, sum & carry. The Boolean equation for sum & carry are:

SUM = A + B

CARRY = A. B

Therefore, sum produces 1 when A&B are different and carry is 1when A&B are 1. Application of Half adder is limited.

Full Adder: It is a logic circuit that can add three bits. It produces two O/P sum & carry. The Boolean Equation for sum & carry are:

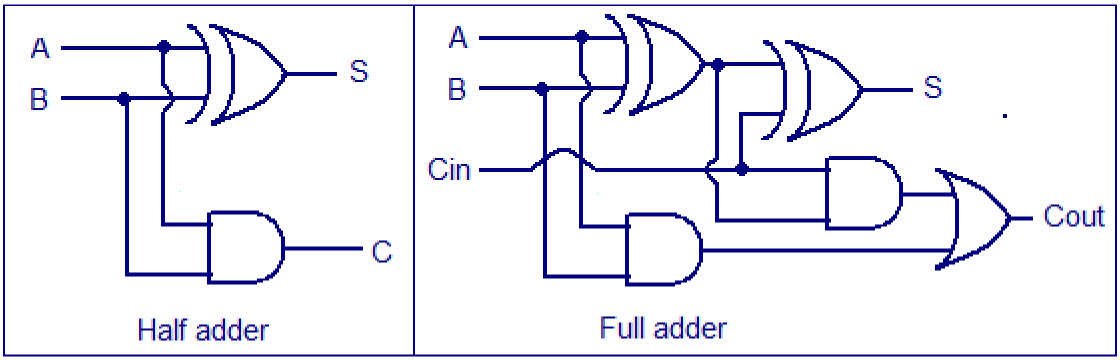
SUM = A + B + C

CARRY = A.B + (A+B) C

Therefore, sum produces one when I/P is containing odd no's of one & carry is one when there are two or more one in I/P.

LOGIC DAIGRAM:

Half Adder Full Adder



PROCEDURE:

(a) Connect the ckt. as shown in fig. For half adder.
(b) Apply diff. Combination of inputs to the I/P terminal.
(c) Note O/P for Half adder.
(d) Repeat procedure for Full wave.
(e) The result should be in accordance with truth table.

OBSERVATIONTABLE:
HALF ADDER:

INPUTS		OUTPUT	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

FULL ADDER:

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

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The Vcc and ground should be applied carefully at the specified pin only.

RESULTS: The Half Adder & Full Adder circuits are verified.

EXPERIMENT NO: 4

AIM: To construct &verify operation of half subtractor and full subtractor using logic gates

APPARATUS REQUIRED:IC 7408,IC 7486,probes,

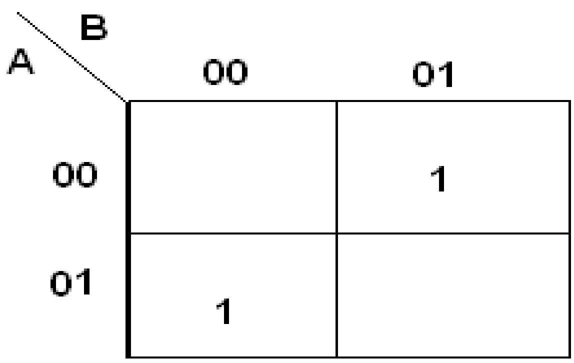
THEORY:

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

TRUTH TABLE

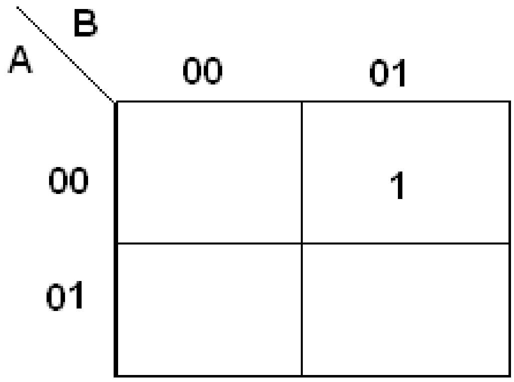
A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

K-Map for DIFFERENCE:



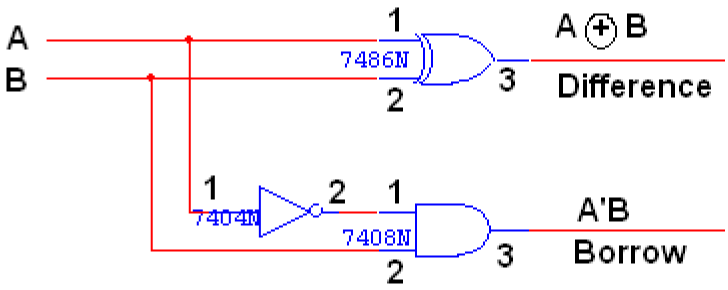
DIFFERENCE = A'B + AB'

K-Map for BORROW:



BORROW = A'B

LOGIC DIAGRAM:



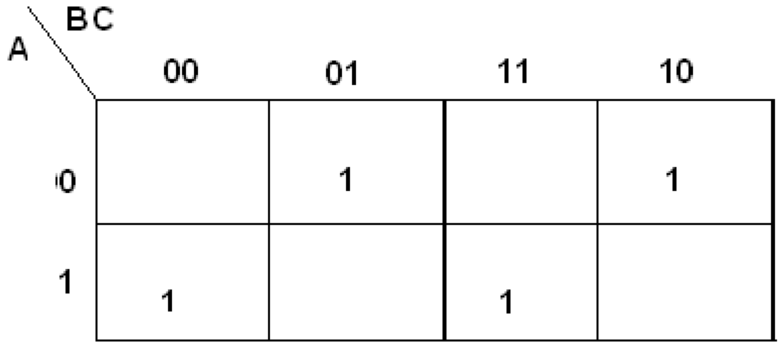
FULL SUBTRACTOR

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

TRUTH TABLE:

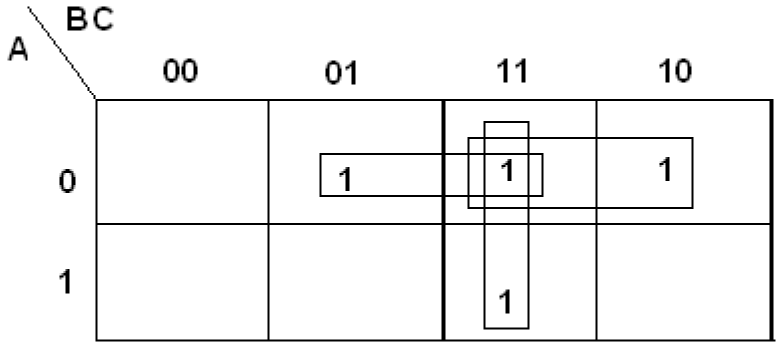
A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-Map for Difference



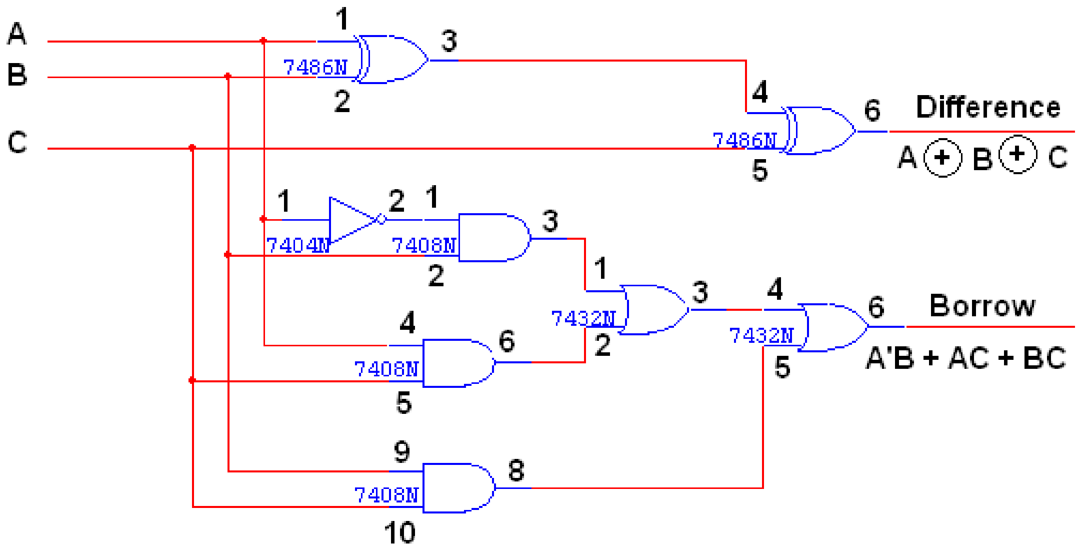
Difference = A'B'C + A'BC' + AB'C' + ABC

K-Map for Borrow:

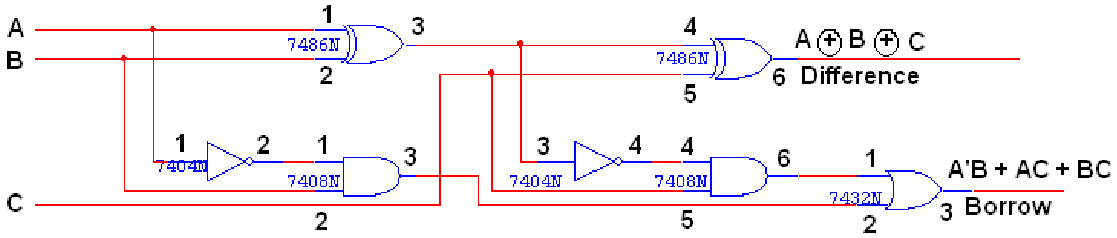


Borrow = A’B + BC + A’C

LOGIC DIAGRAM:



FULL SUBTRACTOR USING TWO HALF SUBTRACTOR



PROCEEDURE: i) Connections are given as per circuit diagram.

ii) Logical inputs are given as per circuit diagram.

RESULTS- Observed the output and verified the truth table.

EXPERIMENT NO:5

AIM: To design and implement a 4-bit Binary to gray code converter.

APPARATUS REQUIRED:

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

THEORY:

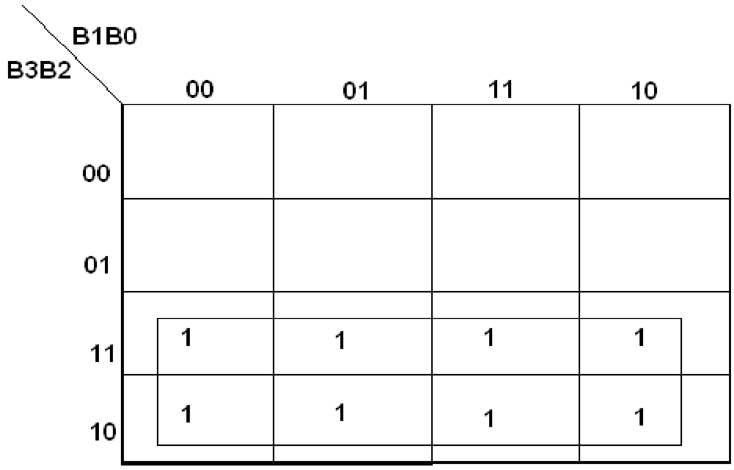
The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code.

The bit combination assigned to binary code to gray code. Since each code uses four bits to represent a decimal digit. There are four inputs and four outputs. Gray code is a non-weighted code. The input variable are designated as B3, B2, B1, B0 and the output variables are designated as G3, G2, G1, Go. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable. A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps. These are various other possibilities for a logic diagram that implements this circuit

TRUTH TABLE:

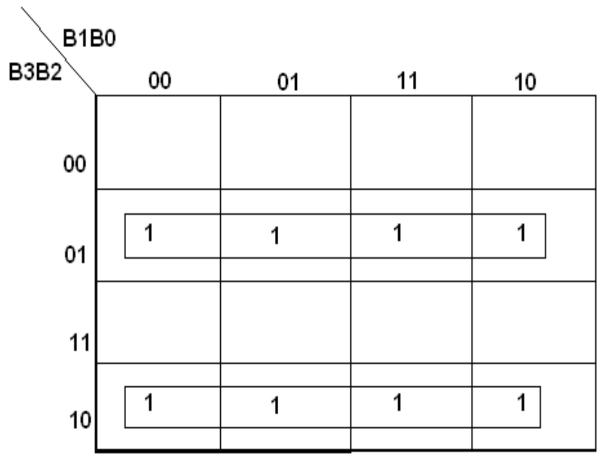
Binary input				Gray code output			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	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:



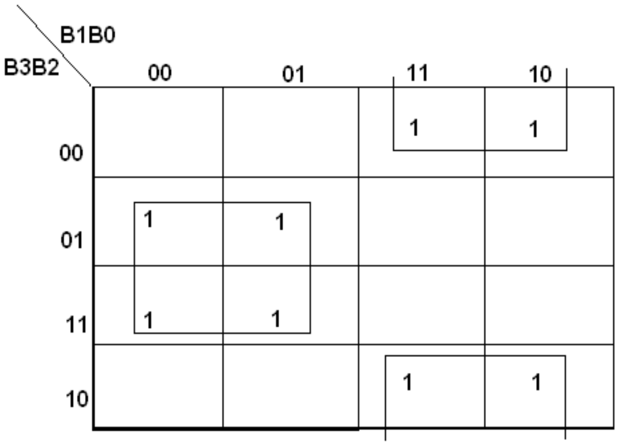
$G_3 = B_3$

K-Map for G_2:



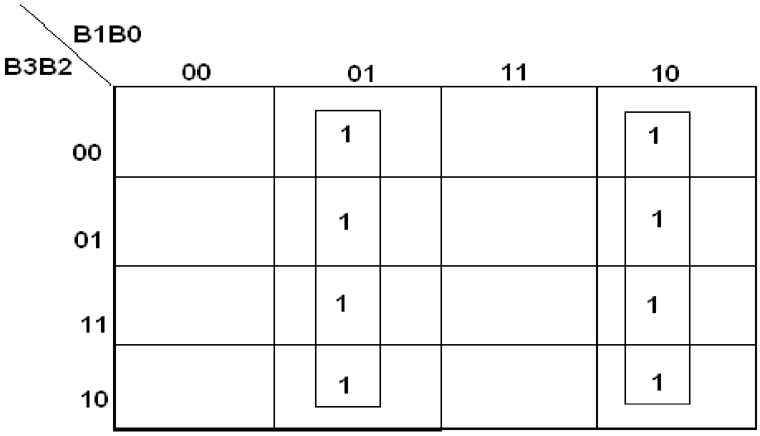
$G2 = B3 \oplus B2$

K-Map for G_1:



$G1 = B1 \oplus B2$

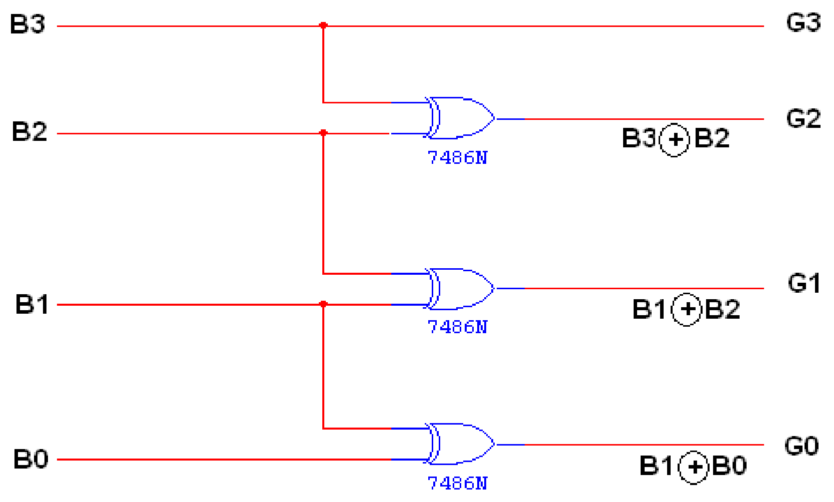
K-Map for G_0:



$G0 = B1 \oplus B0$

LOGIC DIAGRAM:

BINARY TO GRAY CODE CONVERTOR



PROCEDURE: i) Connections were given as per circuit diagram.

ii) Logical inputs were given as per truth table.

RESULTS: Observed the logical output and verified with the truth tables.

EXPERIMENT NO:6

AIM: To Design &Implement Single bit/ two-bit digital comparator circuit

APPARATUS REQUIRED:

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

THEORY:

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

$A = A_3\ \ A_2\ \ A_1\ \ A_0$

$B = B_3\ \ B_2\ \ B_1\ \ B_0$

comparator is specified by three binary variables that indicate whether A>B, A=B (or) A<B.

$A = A_3\ \ A_2\ \ A_1\ \ A_0$

$B = B_3\ \ B_2\ \ B_1\ \ B_0$

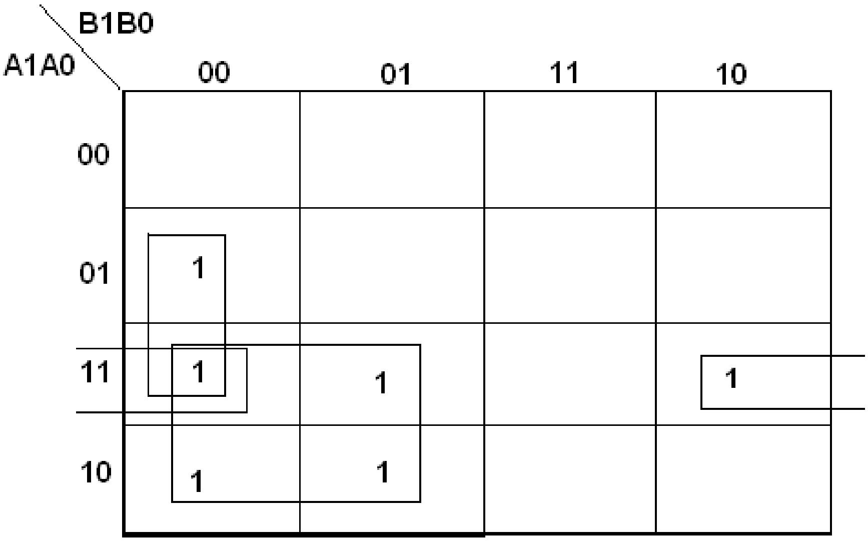
The equality of the two numbers and B is displayed in a combinational circuit designated by the symbol (A=B). This indicates A greater than B, then inspect the relative magnitude of pairs of significant digits starting from most significant position. A is 0 and that of B is 0. We have A<B, the sequential comparison can be expanded as

$A>B = A3B_3^1 + X_3A_2B_2^1 + X_3X_2A_1B_1^1 + X_3X_2X_1A_0B_0^1$

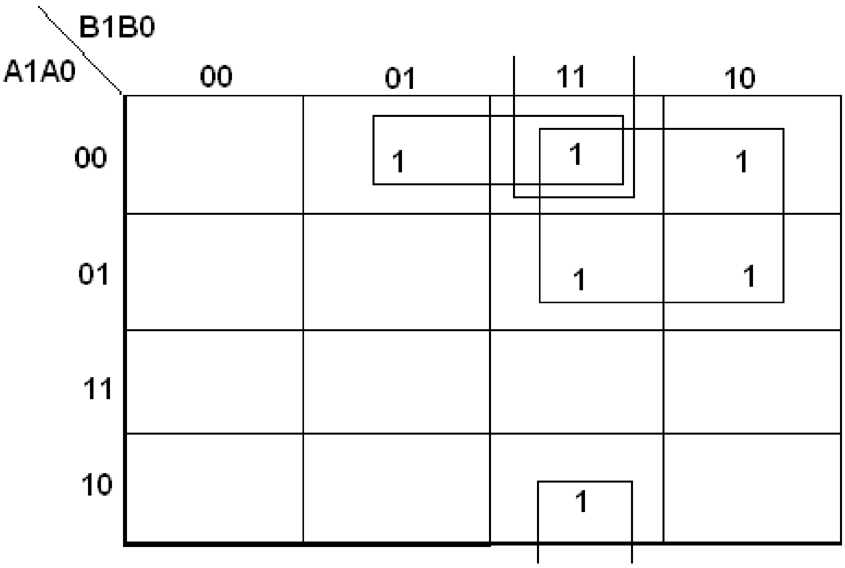
$A<B = A_3^1B_3 + X_3A_2^1B_2 + X_3X2A_1^1B_1 + X_3X_2X_1A_0^1B_0$

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

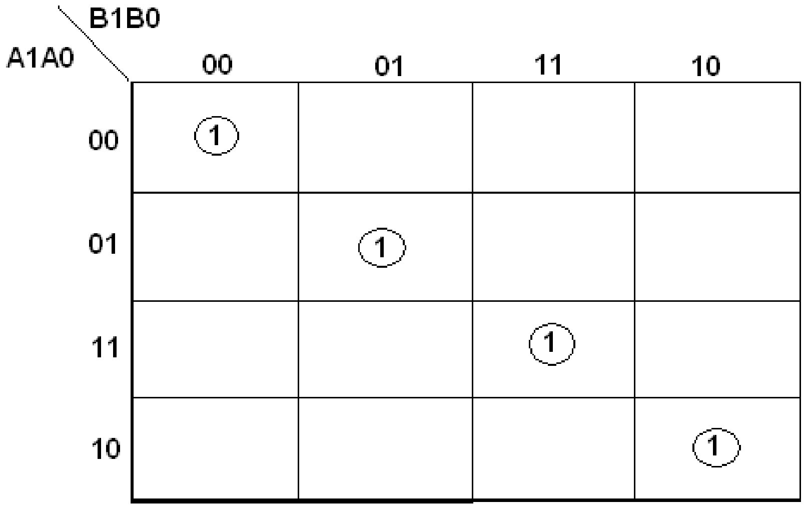
K MAP



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



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

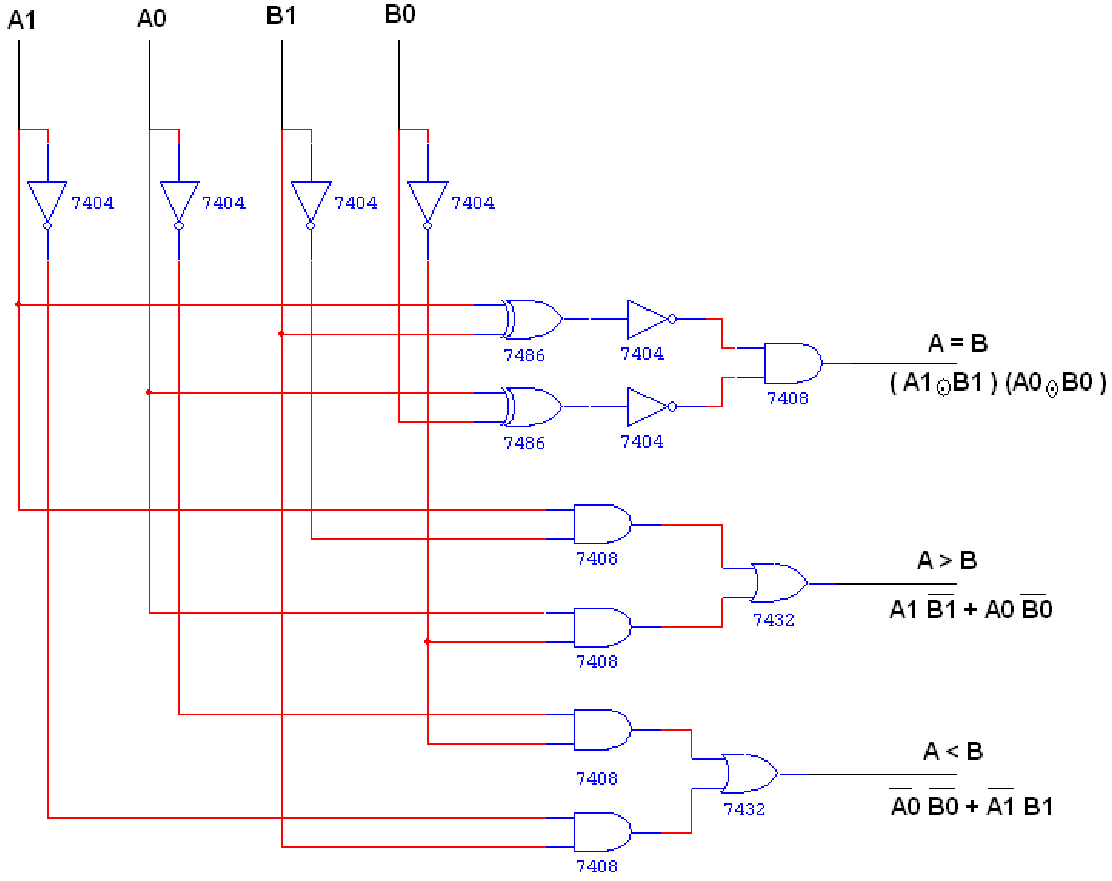


A = B =(A0⊙B0)(A1⊙B1)

TRUTH TABLE

A1	A0	B1	B0	A > B	A = B	A < B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

LOGIC DIAGRAM: 2 BIT MAGNITUDE COMPARATOR



PROCEDURE: i)Connections are given as per circuit diagram.

ii)Logical inputs are given as per circuit diagram.

RESULTS: Observe the output and verify the truth table.

EXPERIMENT NO.7

AIM: Design Multiplexer (4:1) and De-multiplexer (1:4).

APPARATUS REQUIRED:

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

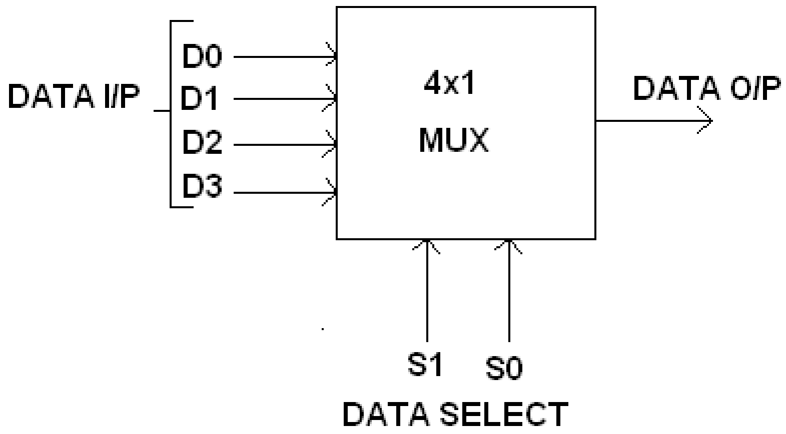
THEORY:MULTIPLEXER:

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

DEMULTIPLEXER: The function of Demultiplexer is in contrast to multiplexer function. It takes information from one line and distributes it to a given number of output lines. For this reason, the demultiplexer is also known as a data distributor. Decoder can also be used as demultiplexer.

In the 1: 4 demultiplexer circuit, the data input line goes to all of the AND gates. The data select lines enable only one gate at a time and the data on the data input line will pass through the selected gate to the associated data output line.

BLOCK DIAGRAM FOR 4:1 MULTIPLEXER:

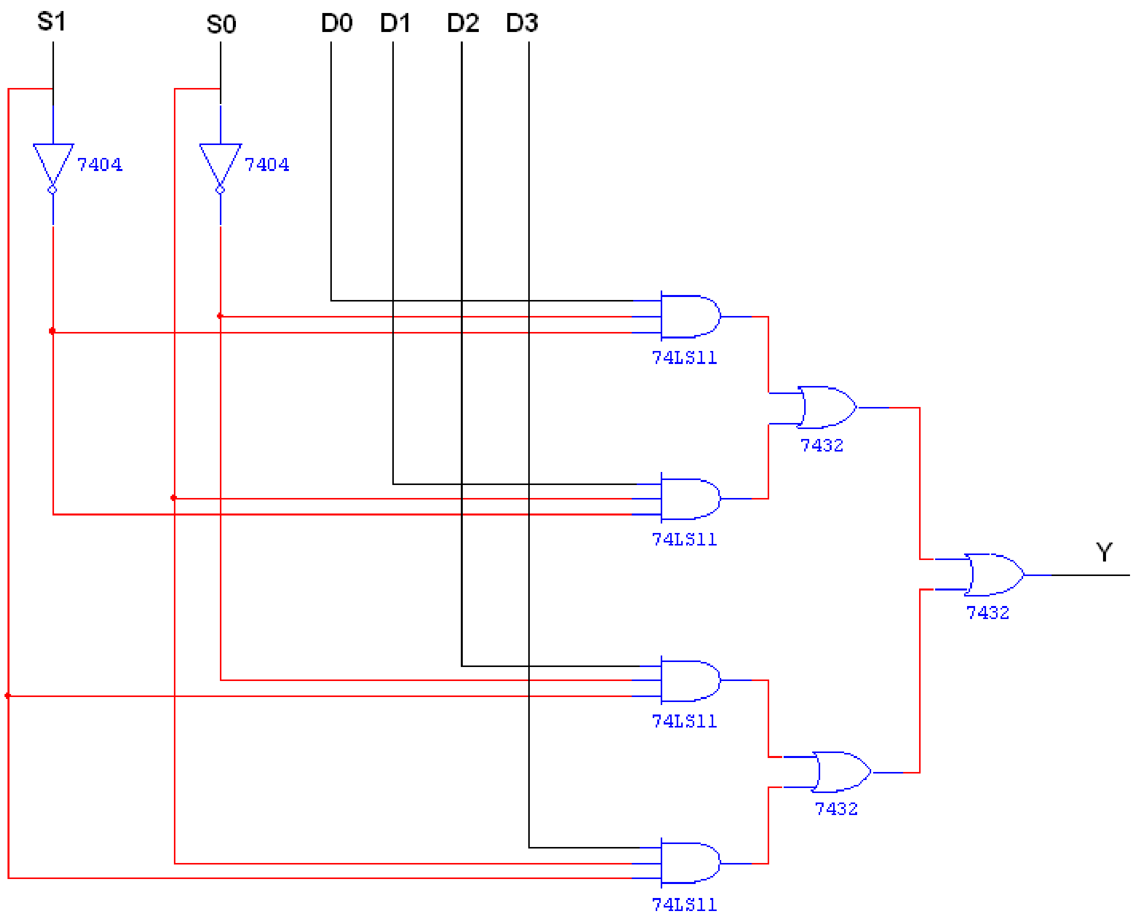


FUNCTION TABLE:

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

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

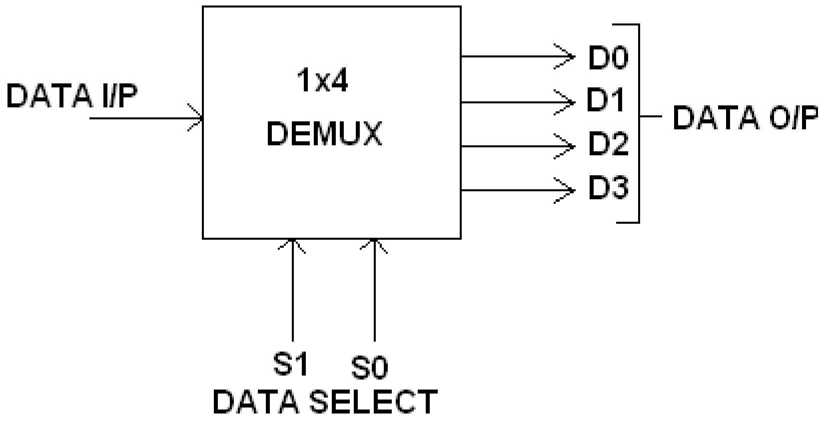
CIRCUIT DIAGRAM FOR MULTIPLEXER:



TRUTH TABLE:

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

BLOCK DIAGRAM FOR 1:4 DEMULTIPLEXER:



FUNCTION TABLE:

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

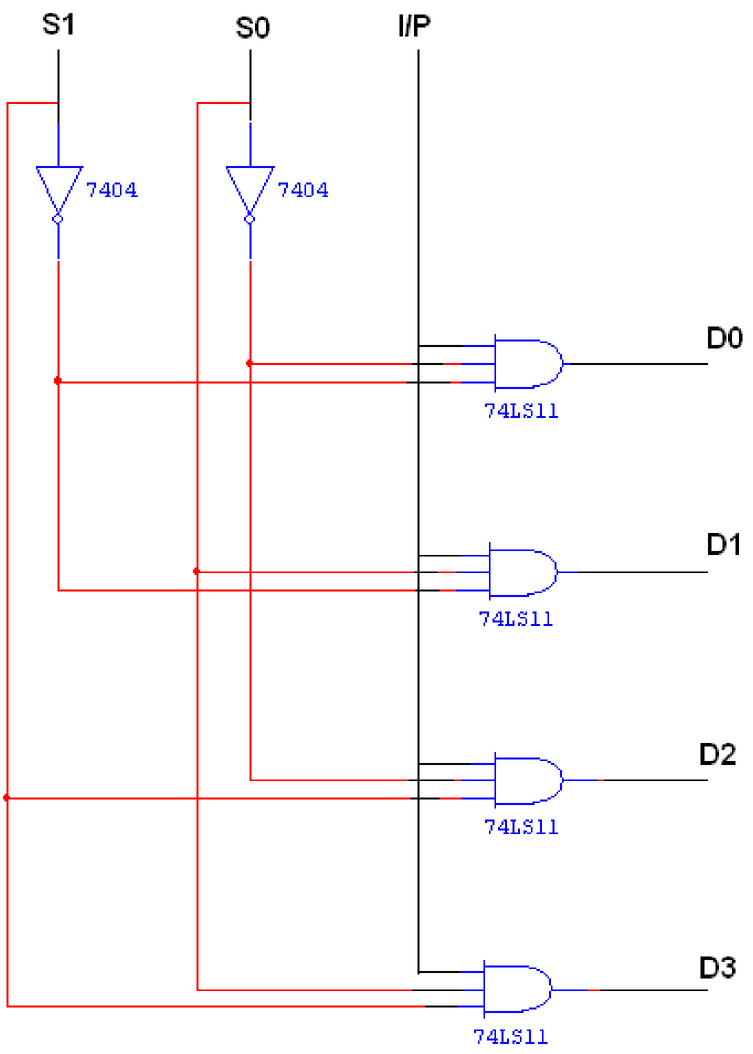
Y = X S1’ S0’ + X S1’ S0 + X S1 S0’ + X S1 S0

TRUTH TABLE:

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

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

LOGIC DIAGRAM FOR DEMULTIPLEXER:



PROCEDURE: i)Connections are given as per circuit diagram.

ii)Logical inputs are given as per circuit diagram.

RESULT: Observe the output and verify the truth table.

EXPERIMENT NO: 8

AIM: Study the operation of flip flops.

I. S-R Flip-Flops
II. J-K Flip-Flops
III. T Flip-Flops
IV. D Flip-Flops

THEORY: -

•RS FLIP-FLOP:

There are two inputs to the flip-flop defined as R and S . When I/Ps R=0 and S=0 then O/P remains unchanged. When I/Ps R=0andS=1 the flip-flop is switches to the stable state where O/P is 1i.e. SET. The I/P condition is R=1 and S=0 the flip-flop is switched to the stable state where O/P is 0i.e. RESET. The I/P condition is R=1 and S=1 the flip-flop is switched to the stable state where O/P is forbidden.

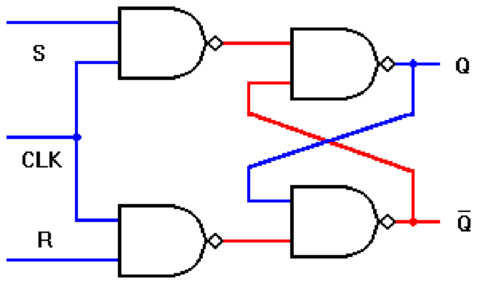
•JK FLIP-FLOP:

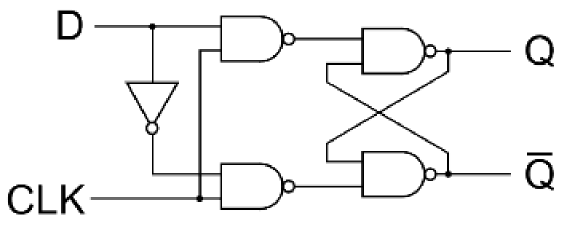
For purpose of counting, the JK flip-flop is the ideal element to use. The variable J and K are called control I/Ps because they determine what the flip-flop does when a positive edge arrives. When J and K are both0s, both AND gates are disabled and Q retain sits last value.

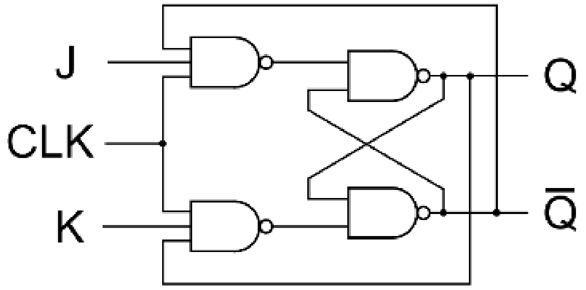
•D FLIP–FLOP:

This kind of flip flop prevents the value of D from reaching the Q output until clock pulses occur. When the clock is low, both AND gates are disabled D can change value without affecting thevalueofQ.Ontheother hand, whentheclockishigh,bothANDgatesareenabled.Inthiscase, QisforcedtoequalthevalueofD. Whentheclockagaingoeslow, QretainsorstoresthelastvalueofD. A Dflip-flopisabi-stablecircuitwhoseDinputistransferredtotheoutputafteraclockpulseisreceived.

•T FLIP-FLOP: TheTor"toggle"flip-flopchangesitsoutputoneaclockedge,givinganoutputwhichishalfthefrequencyofthesignaltotheT input.It is usefulforconstructingbinarycounters,frequencydividers,andgeneralbinaryadditiondevices.Itcanbemadefromaj-Kflip-flopbytyingbothofitsinputshigh.







APPARATUS USED:-

IC' S 7400, 7402 Digital Trainer & Connecting leads.

PROCEDURE:

1. Connect the circuit as shown in figure.

2. Apply Vcc& ground signal to every 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 Vcc and ground should be applied carefully at the specified pin only

OBSERVATION DATA: -

TRUTH TABLE: SRFLIP FLOP:

CLOCK	**S**	**R**	**Q**
1	**0**	**0**	**NOCHANGE**
1	**0**	**1**	**0**
1	**1**	**0**	**1**
1	**1**	**1**	**?**

D FLIP FLOP: -

CLOCK	D	Q
1	0	0
1	1	1

JK FLIP FLOP: -

CLOCK	J	K	Q
1	0	0	NOCHANGE
1	0	1	0
1	1	0	1
1	1	1	Q’

T FLIP FLOP: -

CLOCK	T	Q
1	0	NOCHANGE
1	1	Q’

RESULTS: -Truth table is verified on digital trainer

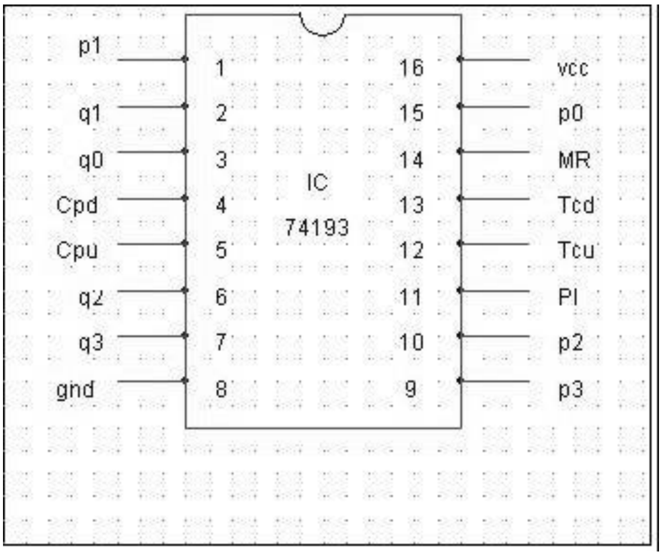
EXPERIMENT NO:9

AIM: To design IC 74193 as a up/down counter

COMPONENTS REQUIRED:

IC 74193, Patch Cords & IC Trainer Kit

PIN DETAILS OF IC 74193



1. P1,P2,P3 and P0 are parallel datainputs
2. Q0,Q1,Q2 and Q3 are flip-flop outputs
3. MR: Asynchronous masterreset
4. PL: Asynchronous parallel load(active low)input
5. TCd : Terminal count-downoutput
6. TCu : Terminal count-upoutput

Upcounter

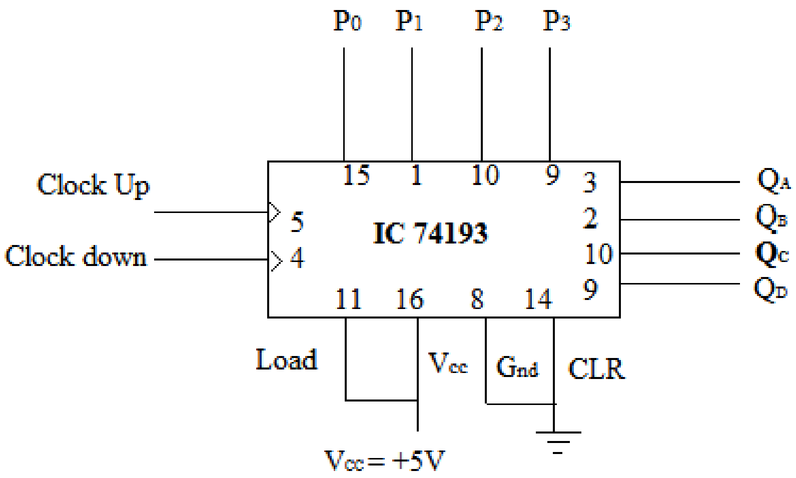
CLK	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Down counter

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

Design up/ down counter using

CIRCUIT DIAGRAM



PROCEDURE:

1. Check the components for theirworking.
2. Insert the appropriate IC into the ICbase.
3. Rig up the circuit as shown in the logic circuitdiagram.
4. Apply the parallel input to p_0 to p_3 and Give the Load pin to logicLOW.
5. To start counting connect Load input to logicHIGH.
6. Apply the clock pulse to observe the output.(For up-counter make clock down to be at logic 1 and give the clock input to Clock up input, for down counter make clock up to be at logic 1 and give the clock input to Clock downinput)

RESULTS:

1. For up-counter make clock down to be at logic 1 and give the clock input to Clock up input.
2. For down counter make clock up to be at logic 1 and give the clock input to Clock downinput
3. Write the pin numbers of each gate and also write the intermediateexpressions.

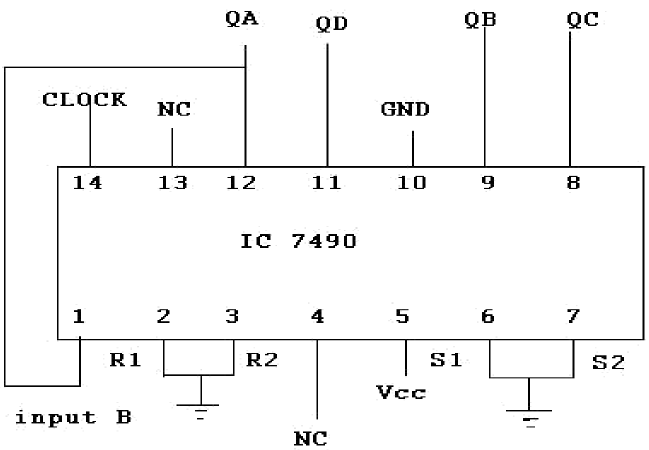
EXPETRIMENT NO: 10

AIM: To design a decade counter.

COMPONENTS REQUIRED:

IC 7490, Patch Cords & IC Trainer Kit

DECADE COUNTER:



TRUTH TABLE:

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

PROCEDURE:

7. Check the components for theirworking.
8. Insert the appropriate IC into the ICbase.
9. Rig up the circuit as shown in the logic circuitdiagram.
10. Apply various input data to the logic circuit via the input logicswitches.
11. Note down the corresponding output and verify the truthtable.

Experiment No: 11

AIM: Study of 4-bit shift registers.

(i) Serial in serial out
(ii) Serial in parallel out
(iii) Parallel in serial out
(iv) Parallel in parallel out

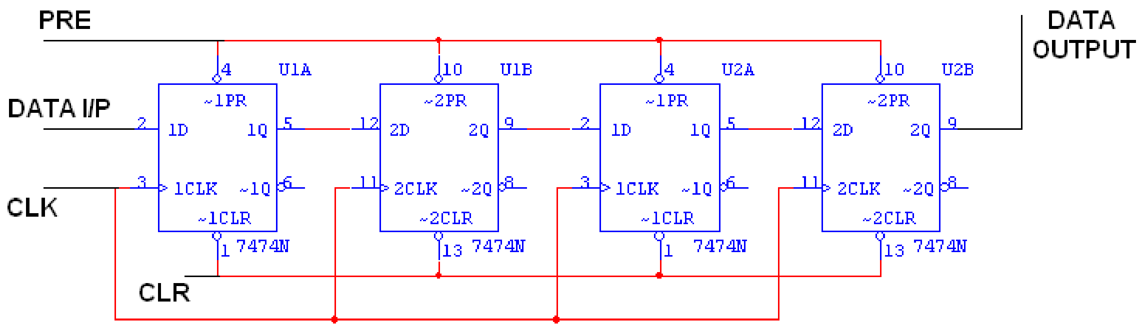
APPARATUS REQUIRED:

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

THEORY:

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

LOGIC DIAGRAM: SERIAL IN SERIAL OUT:

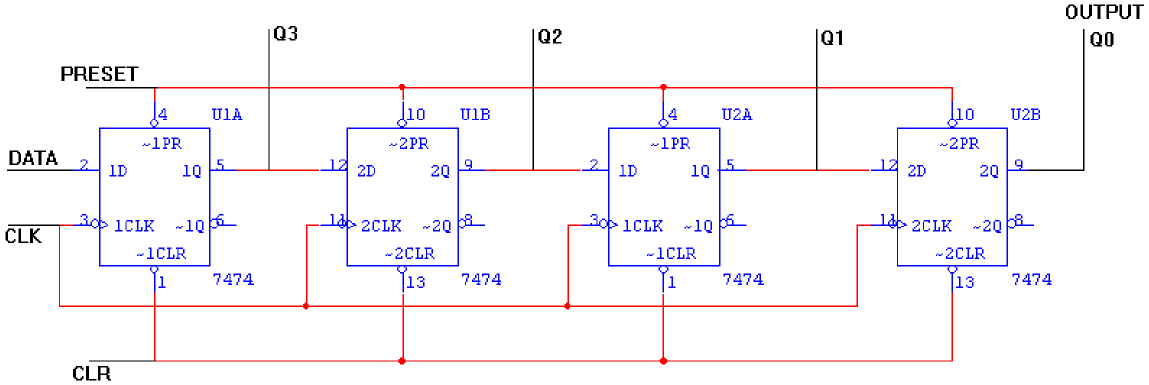


TRUTH TABLE:

CLK	Serial in	Serial out
1	1	0
2	0	0
3	0	0
4	1	1
5	X	0
6	X	0
7	X	1

LOGIC DIAGRAM:

SERIAL IN PARALLEL OUT:

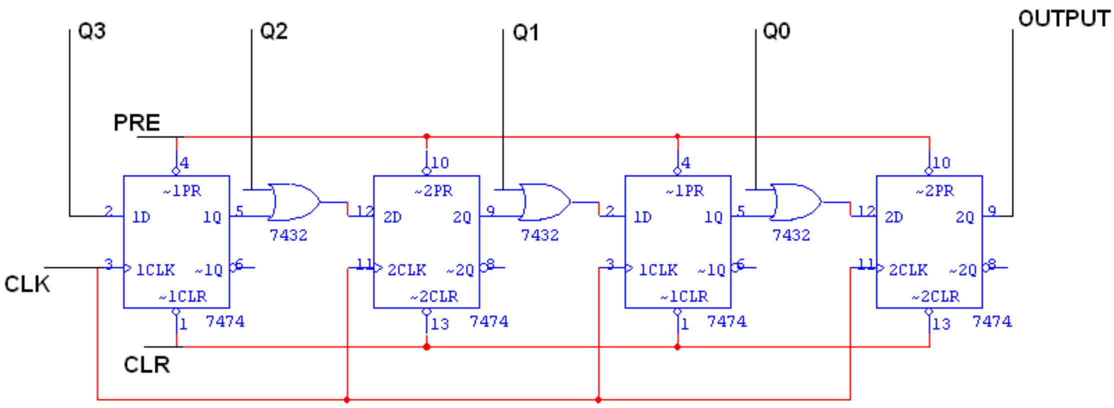


TRUTH TABLE:

CLK	DATA	OUTPUT			
		Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

LOGIC DIAGRAM:

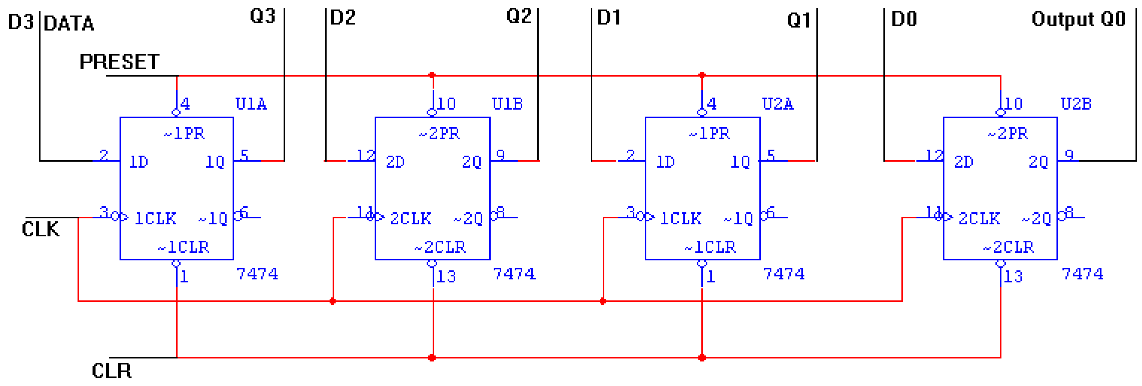
PARALLEL IN SERIAL OUT:



TRUTH TABLE:

CLK	Q3	Q2	Q1	Q0	O/P
0	1	0	0	1	1
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1

LOGIC DIAGRAM:PARALLEL IN PARALLEL OUT:



TRUTH TABLE:

CLK	DATA INPUT				OUTPUT			
	D_A	D_B	D_C	D_D	Q_A	Q_B	Q_C	Q_D
1	1	0	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0

PROCEDURE: i)Connections are given as per circuit diagram.

ii)Logical inputs are given as per circuit diagram.

RESULTS: Observed the output and verified the truth table.

EXPERIMENT NO-12

Aim: Write 8085 assembly language program for one's complement of an 8-bit numbers

Instruments Required: 1. 8085 Microprocessor Kit

2. +5V Power supply

Theory: The number is stored in memory location 8050H and one's complement of number will be stored in location 8051H. Assume the program memory starts from 8000H.

Algorithm

1. Load memory location of data 8000H in H-L registers pair.
2. Move data into accumulator
3. Complement accumulator
4. Store the result in memory location8050H

Program

Memory Address	Hex Code	Mnemonics	Comments
8500	21	LXIH,8000H	Load address of number in H-L register pair
8501	00		
8502	80		
8503	7E	MOVA,M	Move number into accumulator
8504	3F	CMA	Complement accumulator
8505	32	STA8050H	Store the result in 8050H
8506	50		
8507	80		
8508	76	HLT	Stop Execution

Experimental Results

Input Data		Result	
Input Address	**Value**	**Output Address**	**Value**
8000	F0H	8050	0FH

Conclusion

The one’s complement of an 8-bit numbers is performed using 8085microprocessor**. Precautions**

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

EXPERIMENT NO: 13

Aim: Write 8085 assembly language program for two’s complement of an 8-bit numbers

Instruments Required: 1. 8085 Microprocessor Kit

2. +5V Power supply

Theory: The number is stored in memory location 8000H. The two’s complement will be stored in 8050H. The program is written from memory location 8500H.

Algorithm

1. Transfer the content of memory location 8500H to accumulator.
2. Complement the content of accumulator
3. Add 01H with accumulator to get two’s complement of number
4. Store the result in memory location8501H

Program

Memory Address	Hex Code	Mnemonics	Comments
8500	21	LXIH,8000H	Load address of number in H-L register pair
8501	00		
8502	80		
8503	7E	MOVA,M	Move number into accumulator
8504	3F	CMA	Complement accumulator
8505	C6	ADI 01	Add 01H with accumulator to

8506	01	01	find two's complement of number
8507	32	STA8050H	Store the result in 8050H
8508	50		
8509	80		
850A	76	HLT	Stop Execution

Experimental Results

Input Data		Result	
Input Address	**Value**	**Output Address**	**Value**
8000	F0H	8050	10H

Conclusion:

The two's complement of an 8-bit numbers is performed using 8085 microprocessor.

EXPERIMENT NO-14

Aim: Write 8085 assembly language program for addition of two 8-bit numbers.

Instruments Required: 1. 8085 Microprocessor Kit

2. +5V Power supply

Theory : Consider the first number 42H is stored in memory location 8000H and the second number 35H is stored in memory location 8001H. The result after addition of two numbers is to be stored in the memory location 8002 H. Assume program starts from memory location 8500H.

Algorithm

1. Initialize the memory location of first number in HL register pair
2. Move first number/data into accumulator
3. Increment the content of HL register pair to initialize the memory location of second data
4. Add the second data with accumulator
5. Store the result in memory location8003H

Program

Memory address	Machine Codes	Mnemonics	Comments
8500	21	LXI H, 8000 H	Address of first number in H-L registerpair.
8501	00		
8502	80		
8503	7E	MOVA,M	Transfer first number in accumulator.
8504	23	INXH	Increment content of H-L register pair
8505	66	ADDM	Add first number and second number
8506	32	STA8003H	Store sum in 8003 H
8507	03		
8508	80		
8509	76	HLT	Halt

Experimental Results

Input Data		Result	
Memory location	**Data**	**Memory location**	**Data**
8000	42H	8003	77H
8001	35H		

Calculation

Data 1: 42 - 01000010

Data 2: 35 - 00110101

Sum: 77 – 01110111

Carry:00

Conclusion

The addition of two 8-bit numbers is performed using 8085 microprocessor where sum is 8-bit.

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

EXPERIMENT NO-15

AIM-Write 8085 an assembly language program for subtraction of two 8-bit numbers.

Instruments Required: 1. 8085 Microprocessor Kit

2. +5V Power supply

Theory : Consider the first number 55H is stored in memory location 8000H and the second number 32H is stored in memory location 8001H. The result after subtraction of two numbers is to be stored in the memory location 8002H. Assume program starts from memory location 8500H.

Algorithm

1. Initialize the memory location of first number in HL register pair
2. Move first number/data into accumulator
3. Increment the content of HL register pair to initialize the memory location of second data
4. Subtract the second data with accumulator
5. Store the result in memory location8003H

Program

Memory address	Machine Codes	Mnemonics	Comments
8500	21	LXI H, 8000 H	Address of first number in H-L register pair.
8501	00		
8502	80		
8503	7E	MOVA,M	Transfer first number in accumulator.
8504	23	INXH	Increment content of H-L register pair
8505	66	SUBM	Subtract first number and second number
8506	32	STA8003H	Store sum in 8003 H
8507	03		
8508	80		
8509	76	HLT	Halt

Experimental Results

Input Data		Result	
Memory location	**Data**	**Memory location**	**Data**
8000	55H	80 03	23H
8001	32H		

Calculation

Data 1: 55 -01010101

Data 2: 32 -00110010

Difference: 23 -0010 0011

Borrow: 00

Conclusion

Subtraction of two 8-bit numbers is performed using 8085 microprocessor where sum is 8-bit.

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

EXPERIMENT NO: 16

Aim: Program for Decimal Addition of Two 8-Bit Numbers and Sum is 16 Bit

Apparatus required: 8085 Microprocessor Kit, +5V Power supply, keyboard

Theory: Two decimal numbers are stored in 8501H and 8502H. The result is to be stored in 8503H location. Consider program starts from memory location 8000H

Memory Address	Machine Codes (Data)	Labels	Mnemonics	Operands	Comments
8000	21		LXI	H, 8501 H	Address of first number in H-L register pair.
8001	01				Lower byte data isstored in memory
8002	85				Higher byte data is stored in memory

8003	0E		MVI	C,00H	Sum of msb's& register value in 00h
8004	00				
8005	7E		MOV	A,M	Transfer first number in accumulator.
8006	23		INX	H	Increment content of H-L register pair
8007	86		ADD	M	Add first number and second number
8008	27		DAA		
8009	D2		JNC	800CH	Jump if no carry to 800Ch location
800A	0C				Lower byte data isstored in memory
800B	80				Higher byte data is stored in memory
800C	0C		INR	C	Increment register C
800D	32	AHEAD	STA	8503H	Data of accumulatoris stored into 8503haddress
800E	03				Lower byte data isstored in memory
800F	85				Higher byte data is stored in memory
8010	79		MOV	A,C	MSB'S of sum in A
8011	32		STA	8504H	MSB'S of sum in A is transferred to 8504h location
8012	04				Lower byte data isstored in memory
8013	85				Higher byte data is stored in memory
	EF		RST.5		Terminate program

RESULTS:

INPUT DATA	RESUL T

Memory location	Data	Memory location	Data
8501	34H	8503	5AH
8502	26H	8504	00H

EXPERIMENT NO: 17

Aim: Program for Decimal Subtraction of Two 8-Bit Numbers

Apparatus required: 8085 Microprocessor Kit, +5V Power supply, keyboard

Theory: Two decimal numbers are stored in 8501H and 8502H. The result is to be stored in 8503H location. Consider program starts from memory location 8000H

Program:

Memory Address	Machine Codes (Data)	Labels	Mnemonics	Operands	Comments
8000	21		LXI	H, 8501 H	Address of first number in H-L register pair.
8001	01				Lower byte data isstored in memory
8002	85				Higher byte data is stored in memory
8003	3E		MVI	A,99H	Copy immediate data 99 in A
8004	99				
8005	96		SUB	M	9’s complement
8006	3C		INR	A	Increment content of A register
8007	2B		DCX	H	Decrement content of H-L register pair

8008	86		ADD	M	Addition of complemented data and the second number
8009	27		DAA		Decimal Accumulator Adjust
800A	32		STA	8503H	Data of accumulatoris stored into 8503haddress
800B	03				
800C	85				
800D	EF		RST.5		Terminate program

RESULTS-

INPUT DATA		RESULT	
Memory location	**Data**	**Memory location**	**Data**
8501	34H	8503	5AH
8502	26H	8504	00H

EXPERIMENT NO: 18

Aim: Program to find largest between two numbers.

Apparatus required: 8085 Microprocessor Kit, +5V Power supply, keyboard

Theory: *The first number is stored in 8200h and the HL reg.pair is incremented and the second number is compared with the first. Depending on the sign flag (if positive) first number is stored in memory*

Program:

Memory Address	Machine Codes (Data)	Labels	Mnemonics	Operands	Comments

8000	21		LXI	H, 8200 H	Load data into HL register pair
8001	00				
8002	82				
8003	7E		MOV	A, M	Copy data into A
8004	D3		INX	H	Increment HL register pair
8005	BE		CMP	M	Compare
8006	F2		JP	800AH	Jump on Plus
8007	0A				
8008	80				
8009	7E		MOV	A, M	Copy data into A
800A	32		STA	8400H	Store The Result In Memory
800B	00				
800C	84				
800D	76		HLT		Halt

Result:

INPUT DATA		RESULT	
Memory location	**Data**	**Memory location**	**Data**
8200	03	8400	03
8201	01		

EXPERIMENT NO-19

Aim: To find the largest element in an array of size 'n' using 8085 Microprocessor.

Instruments required: 1. 8085 Microprocessor Kit

2. +5V Power supply

Theory: Find the largest number in a block of data. The length of the block is in memory location 8000H and the block itself starts from memory location 8001H. Store the maximum number in memory location 8050H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.

Algorithm

1. Initialize counter
2. Maximum = Minimum possible value =0
3. Initialize pointer
4. Is number> maximum
5. Yes, replace maximum
6. Decrement counter by one
7. Go to step 4 until counter =0
8. Store maximum number
9. Terminate program execution

Program

Memory address	Label	Mnemonics	Hex Code	Comments
8500		LDA 8000	3A	Load the number of values
8501			00	
8502			80	
8503		MOV C,A	79	Initialize counter
8504		XRA A	AF	Clear Accumulator
8505		LXI H, 8001	21	Set the pointer for array
8506			01	
8507			80	
8508	BACK	CMP M	BD	Is number> maximum
8509		JNC SKIP	D2	No, jump to SKIP
850A			0D	
850B			85	
850C		MOV A,M	7E	replace maximum
850D	SKIP	INX H	23	Increment pointer
850E		DCR C	0D	Decrement counter by one
850F		JNZ BACK	C2	

8510			08	Go to next iteration
8511			85	

Memory address	Label	Mnemonics		Hex Code	Comments
850D		DCR	C	0D	Decrement counter
850E		JNZ	BACK	C2	If count 0 repeat
850F				0C	
8510				85	
8511		SHLD	8001	22	Store result
8512				01	
8513				80	
8514		HLT		76	Stop execution

Experimental Results

Input Data		Result	
Memory location	**Value**	**Memory location**	**Value**
8000	05	8001	0F
Accumulator	03	8002	00

Calculation

05 - 0000 0101

+ 05 -0000 0101

----- 0A - 0000 1010

+ 05 -0000 0101

= 0F - 0000 1111

Conclusion

The multiplication of two 8-bit numbers is performed using 8085 microprocessor where result is 16-bit.

Precautions

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

EXPERIMENT NO: 20

AIM-Write a program to control the traffic light system using 8085 & 8255 PPI.

APPARATUS REQUIRED- 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY- This Program controls light of one square. By changing the delay between two signals one can change the speed of traffic. 8255 Port Address**.**

Port A- 00H

Port B -01H

Port C- 02H

Control Word 03H

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
2000		3E 80	MVI	A,80H	Init PA &PB as output
2002		D3 03	OUT	03H	
2004		3E 11	MVI	A,11H	Stop all four ends
2006		D3 00	OUT	00H	
2008		D3 02	OUT	02H	
200A		CD 50 20	CALL	DELAY1	
200D	LOOP	3E 44	MVI	A,44H	GO STR signal of North & South, STOP signal of East &West
200F			OUT	00H	
2011			CALL	DELAY1	
2014			MVI	A,22H	Alert signal for traffic
2016			OUT	00H	
2018			CALL	DELAY2	
201B			MVI	A,99H	GO LEFT signal of North & South
201D			OUT	00H	
201F			CALL	DELAY1	
2022			MVI	A,22H	Alert signal for traffic
2024			OUT	00H	
2026			CALL	DELAY2	
2029			MVI	A,11H	STOP signal of North & South
202B			OUT	00H	
202D			MVI	A,44H	GO STR signal of East & West
202F			OUT	02H	

2031			CALL	DELAY1	
2034			MVI	A,22H	Alert signal for traffic
2036			OUT	02H	
2038			CALL	DELAY2	

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
203B			MVI	A,99H	GO Left signal of East & West
203D			OUT	02H	
203F			CALL	DELAY1	
2042			MVI	A,22H	Alert signal for traffic
2044			OUT	02H	
2046			CALL	DELAY2	
2049			MVI	A,11H	STOP signal of East &West
204B			OUT	02H	
204D			JMP	LOOP	Jump to loop
2050		DELAY1:	MVI	B,25H	Delay of 10 sec.
2052		LP3:	MVI	C,0FFH	
2054		LP2:	MVI	D, 0FFH	
2056		LP1:	DCR	D	
2057			JNZ	LP1	
205A			DCR	C	
205B			JNZ	LP2	
205E			DCR	B	
205F			JNZ	LP3	
2062			RET		
2063		DELAY2:	MVI	B,05H	Delay of 2 sec
2065		LP6:	MVI	C,0FFH	
2067		LP5:	MVI	D,0FFH	
2069		LP4:	DCR	D	
206A			JNZ	LP4	
206D			DCR	C	
206E			JNZ	LP5	
2071			DCR	B	
2072			JNZ	LP6	
2075			RET		

RESULTS- Traffic Signal Timing observed for four lane .