



Apollo Engineering College

Chennai

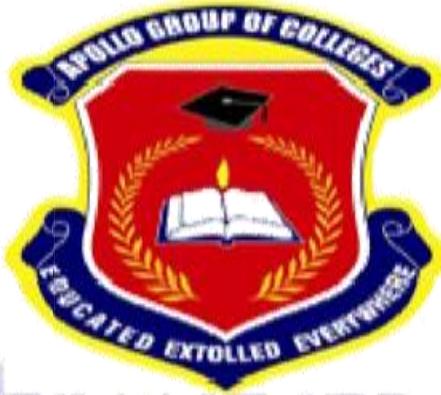
LABORATORY RECORD

Name: _____

Class : _____ Semester: _____

Subject Code & Name: _____

Academic Year: _____



APOLLO ENGINEERING COLLEGE

Chennai

Certificate

Name:

Class:

Reg. No:

Certificate that this is a bonafide record of work done by the above student in
the _____ during the year 202 - 202 .

Staff-In-Charge

Head of Department

Submitted for the practical examination held on

Internal Examiner

External Examiner

CONTENTS

S.NO	DATE	NAME OF THE EXPERIMENTS	MARKS	SIGN
1.		Verification of Boolean theorems using logic gates.		
2.		Design and implementation of combinational circuits using gates for arbitrary functions.		
3.		Implementation of 4-bit binary adder/subtractor circuits		
4.		Implementation of code converters.		
5.		Implementation of BCD adder, encoder and decoder circuits		
6.		Implementation of functions using Multiplexers		
7.		Implementation of the synchronous counters		
8.		Implementation of a Universal Shift register		
9)		Simulator based study of Computer Architecture		

EXPT NO. : 1

STUDY OF LOGIC GATES

DATE :

AIM:

To study about logic gates and verify their truth tables.

APPARATUS REQUIRED:

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

THEORY:

Circuit that takes the logical decision and the process are called logic gates.

Each gate has one or more input and only one output.

OR, AND and NOT are basic gates. NAND, NOR and X-OR are known as universal gates. Basic gates form these gates.

AND GATE:

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

OR GATE:

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

NOT GATE:

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

NAND GATE:

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

NOR GATE:

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

X-OR GATE:

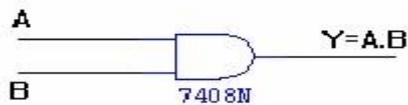
The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high. When number of inputs for an ex-or gate exceeds two the operation can be described as follows: “When the input contains odd number of 1s then the output becomes high, when the number of 1s in the input contains even number of 1s or zero 1s then the output becomes low.

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.

AND GATE:

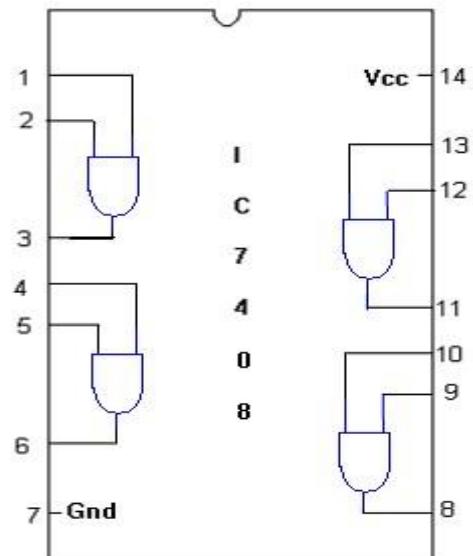
SYMBOL:



TRUTH TABLE

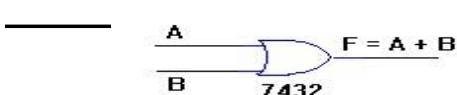
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

PIN DIAGRAM:



OR GATE

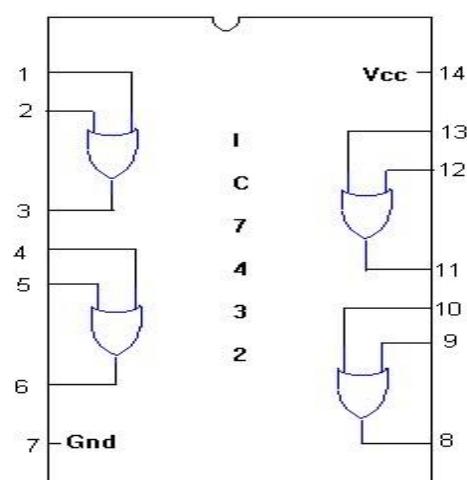
SYMBOL :



TRUTH TABLE

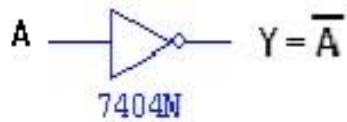
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

PIN DIAGRAM :

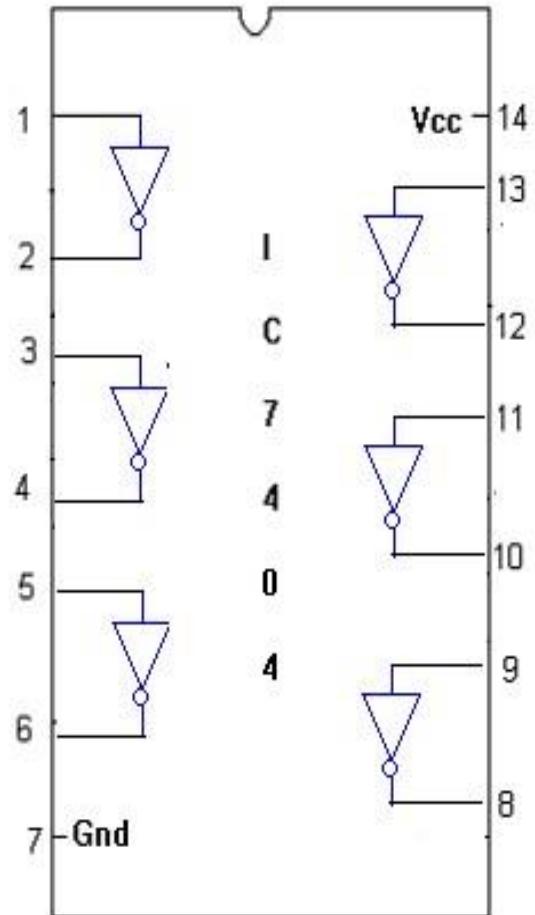


NOT GATE:

SYMBOL:



PIN DIAGRAM

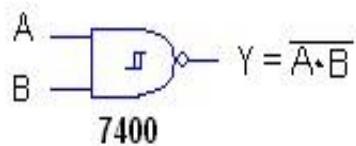


TRUTH TABLE :

A	\bar{A}
0	1
1	0

2-INPUT NAND GATE:

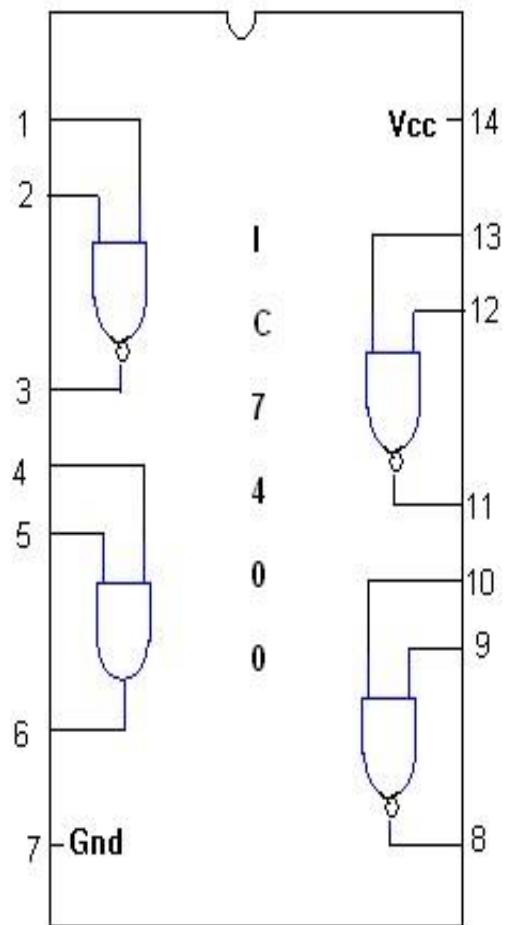
SYMBOL:



TRUTH TABLE

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

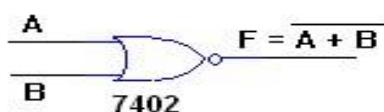
PIN DIAGRAM:



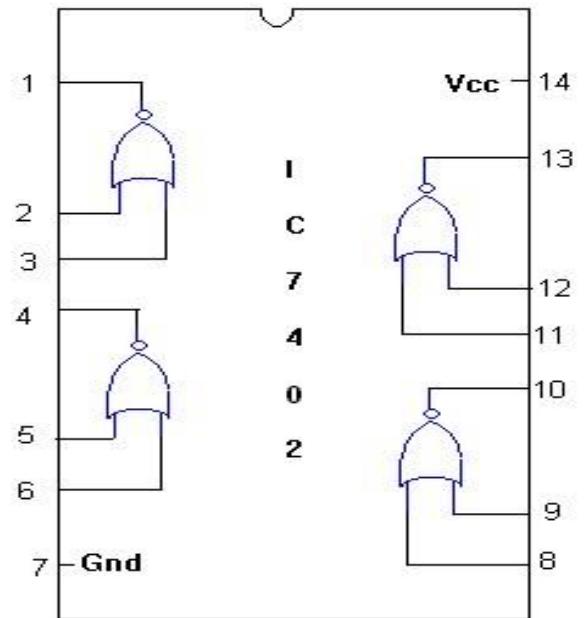
3-INPUT NAND GATE :

NOR GATE:

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

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

RESULT:

**Ex. No: 2 DESIGN AND IMPLEMENTATION OF COMBINATIONAL
Date: CIRCUITS USING BASIC GATES FOR ARBITRARY FUNCTIONS**

OBJECTIVE:

To design and implement the circuit for the following function

1. $F=ABC+C'$
2. $F=A'B'+C$

OUTCOME:

It gives the idea about the construction of combinational circuits for arbitrary functions.

PRE-REQUSITE:

Users must have basic ideas about the gates and their truth table and also the idea about combinational circuits.

INTRODUCTION:

The purpose of this experiment is to construct the combinational circuits for arbitrary functions using the basic gates.

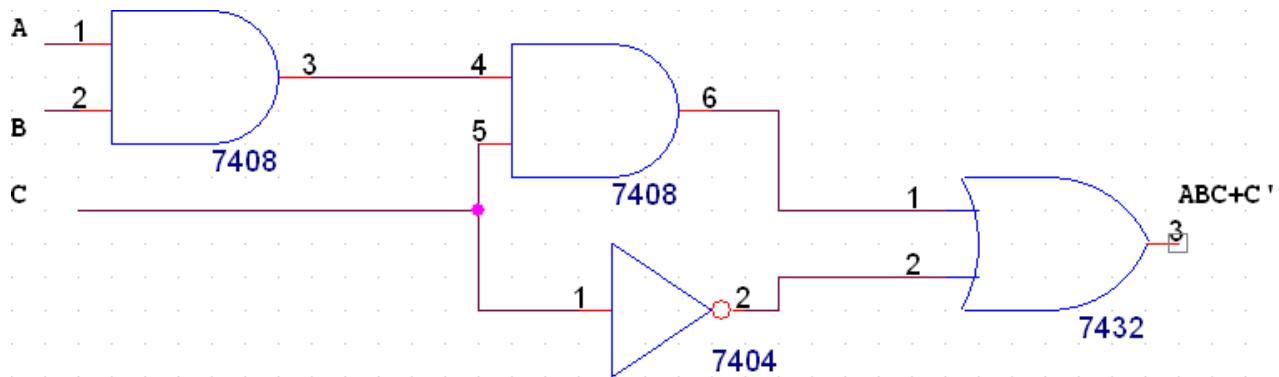
APPARATUS REQUIRED:

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

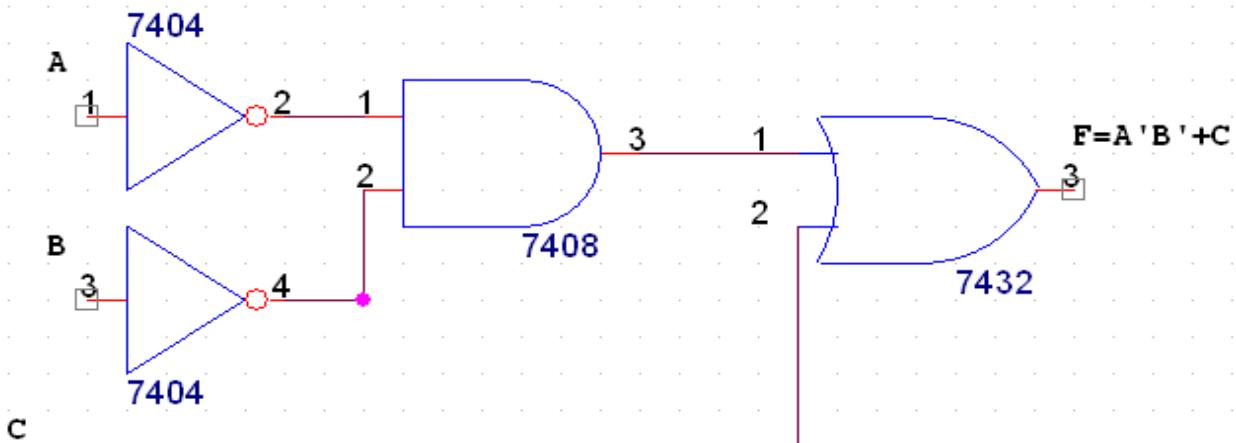
PROCEDURE:

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

1. LOGIC DIAGRAM:



LOGIC DIAGRAM:



TRUTH TABLE:

$$F = ABC + C'$$

			OUTPUT
A	B	C	$F = ABC + C'$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$$F = A'B' + C$$

			INPUT	OUTPUT
A	B	C	$F = A'B' + C$	
0	0	0	1	
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	0	
1	1	1	1	

RESULT:

EXPT NO. : 3

DATE :

IMPLEMENTATION OF 4-BIT ADDER AND SUBTRACTOR

AIM:

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

APPARATUS REQUIRED:

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

THEORY:

4 BIT BINARY ADDER:

A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of next full adder in chain. The augends bits of ‘A’ and the addend bits of ‘B’ are designated by subscript numbers from right to left, with subscript 0 denoting the least significant bits. The carries are connected in chain through the full adder. The input carry to the adder is C_0 and it ripples through the full adder to the output carry C_4 .

4 BIT BINARY SUBTRACTOR:

The circuit for subtracting $A - B$ consists of an adder with inverters, placed between each data input ‘B’ and the corresponding input of full adder. The input carry C_0 must be equal to 1 when performing subtraction

4 BIT BINARY ADDER/SUBTRACTOR:

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

4 BIT BCD ADDER:

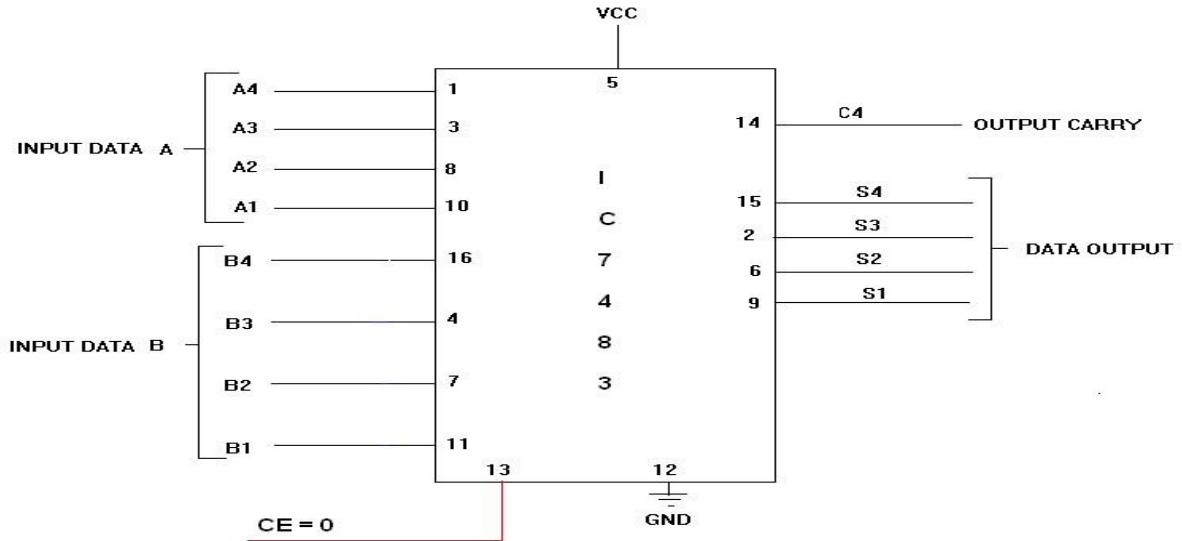
Consider the arithmetic addition of two decimal digits in BCD, together with an input carry from a previous stage. Since each input digit does not exceed 9, the output sum cannot be greater than 19, the 1 in the sum being an input carry. The output of two decimal digits must be represented in BCD and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

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

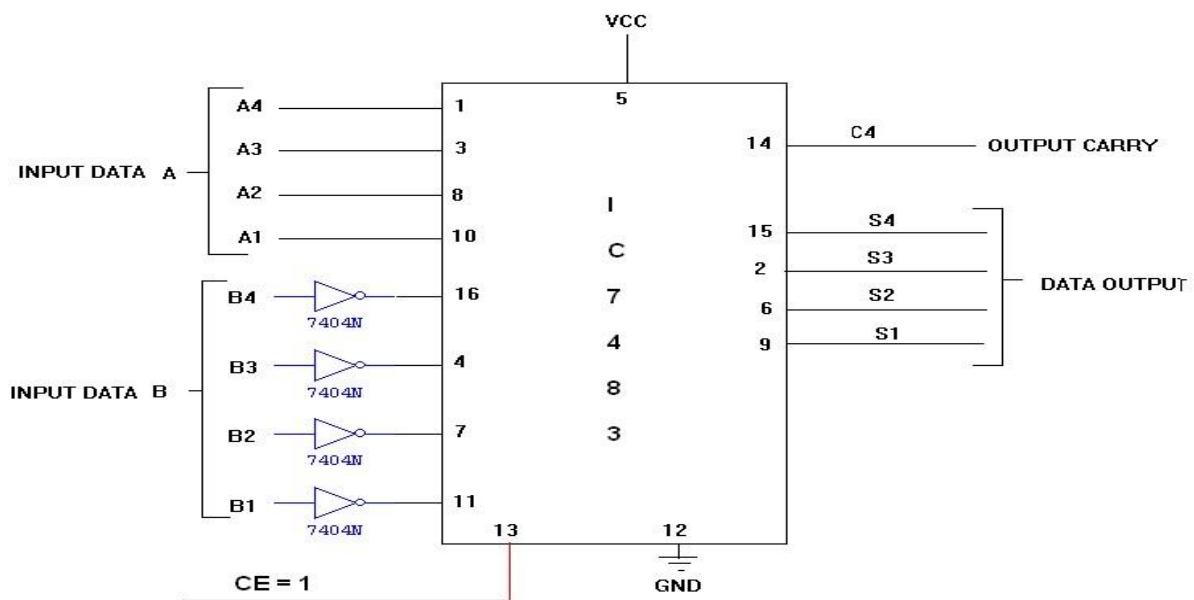
PIN DIAGRAM FOR IC 7483:

1	A4		B4	16
2	S3	I	S4	15
3	A3	C	C4	14
4	B3	7	C1	13
5	VCC	4	GND	12
6	S2	8	B1	11
7	B2	3	A1	10
8	A2		S1	9

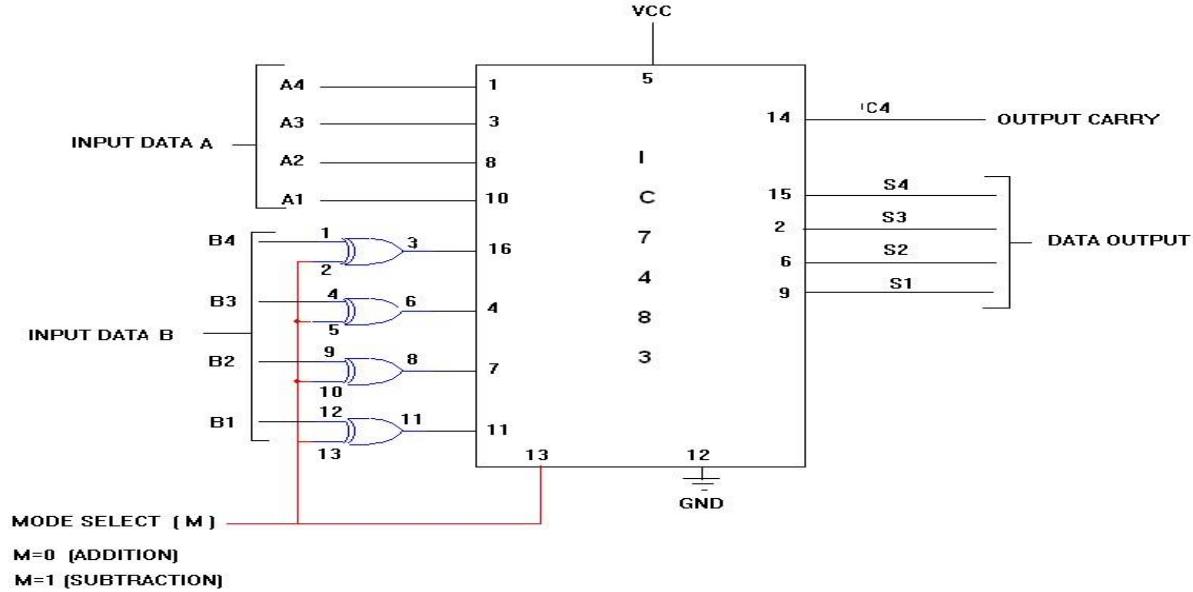
LOGIC DIAGRAM: 4-BIT BINARY ADDER



LOGIC DIAGRAM: 4-BIT BINARY SUBTRACTOR



LOGIC DIAGRAM: 4-BIT BINARY ADDER/SUBTRACTOR

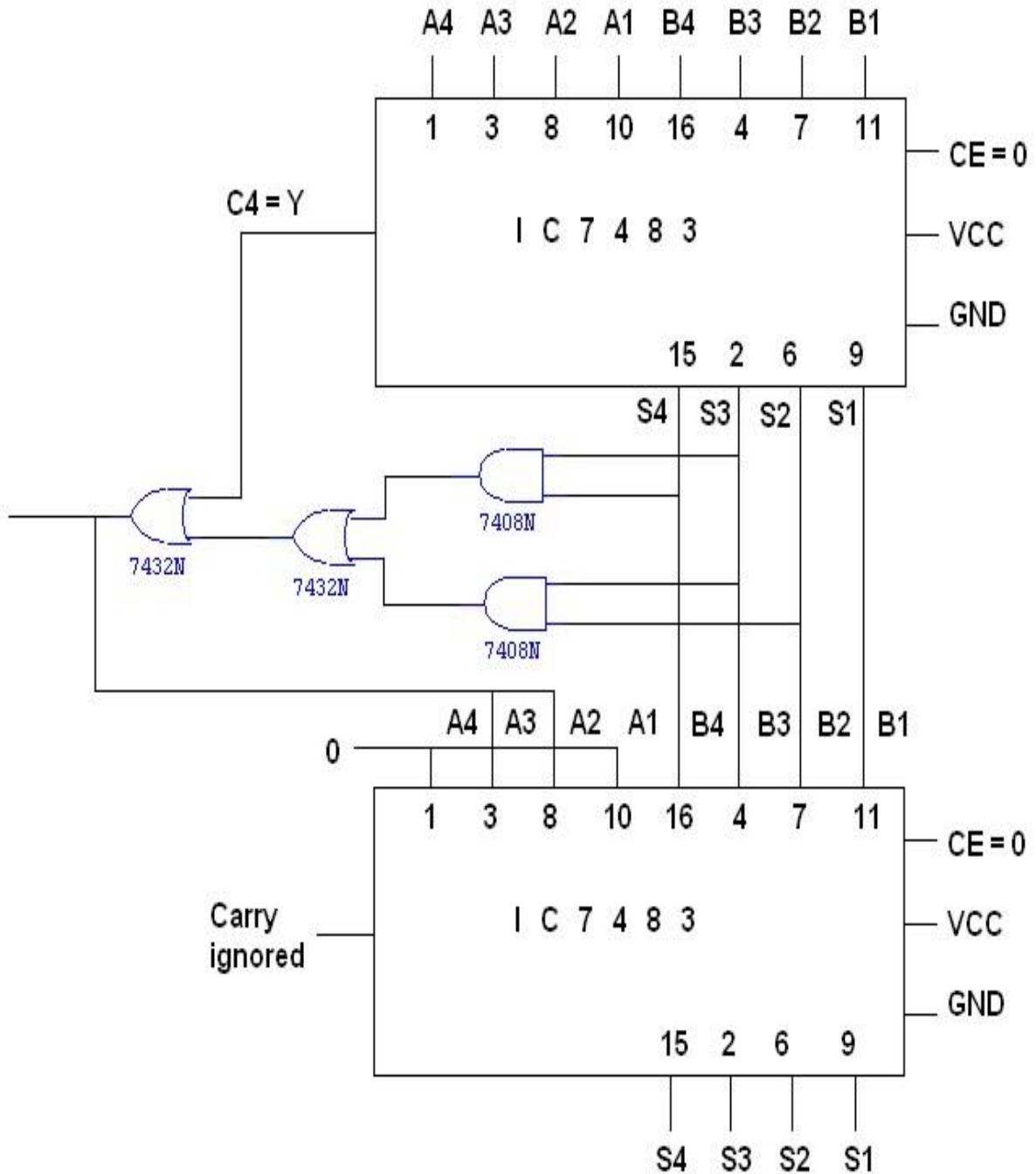


TRUTH TABLE:

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

LOGIC DIAGRAM:

BCD ADDER



K – Map:

		S1 S2	00	01	11	10
		S3 S4	00	0	0	0
			01	0	0	0
			00	0	0	0
			01	0	0	0
			11	1	1	1
			10	0	0	1

$$Y = S4 (S3 + S2)$$

TRUTH TABLE:

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

PROCEDURE:

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

RESULT:

EXPT NO. : 4

DATE :

IMPLEMENTATION OF CODE CONVERTOR

AIM:

To design and implement 4-bit

- (i) Binary to gray code converter
- (ii) Gray to binary code converter
- (iii) BCD to excess-3 code converter
- (iv) Excess-3 to BCD code converter

APPARATUS REQUIRED:

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

THEORY:

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

The bit combination assigned to binary code to gray code. Since each code uses four bits to represent a decimal digit. There are four inputs and four outputs. Gray code is a non-weighted code.

The input variable are designated as B3, B2, B1, B0 and the output variables are designated as C3, C2, C1, Co. from the truth table, combinational circuit is

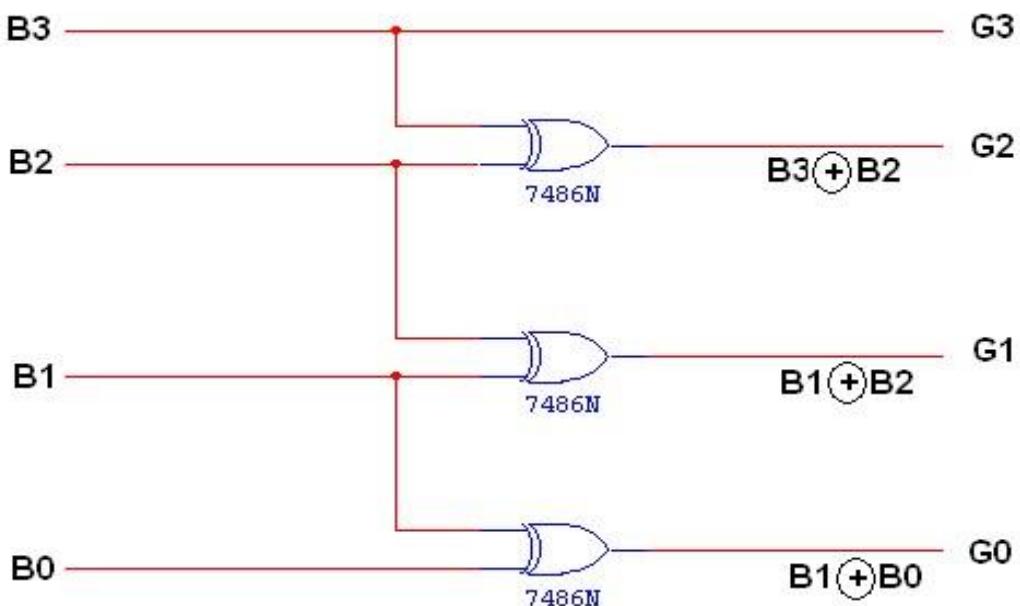
designed. The Boolean functions are obtained from K-Map for each output variable.

A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. To convert from binary code to Excess-3 code, the input lines must supply the bit combination of elements as specified by code and the output lines generate the corresponding bit combination of code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables.

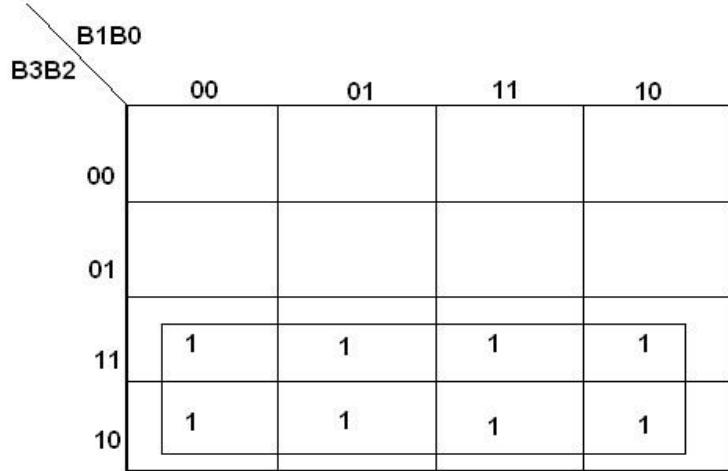
A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps. These are various other possibilities for a logic diagram that implements this circuit. Now the OR gate whose output is $C+D$ has been used to implement partially each of three outputs.

LOGIC DIAGRAM:

BINARY TO GRAY CODE CONVERTOR

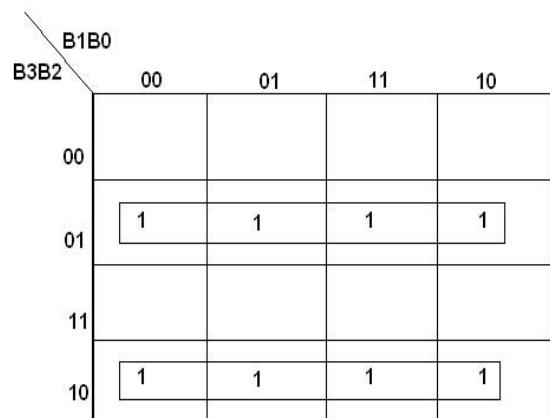


K-Map for G_3 :



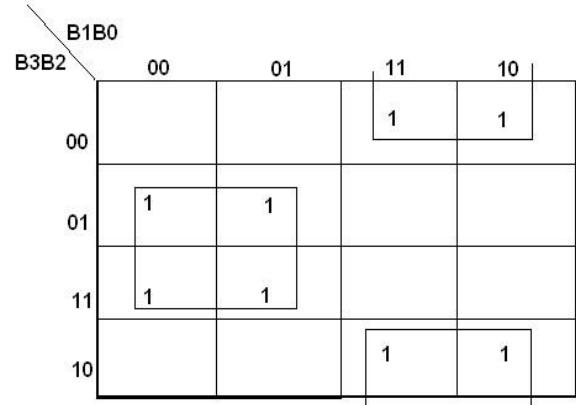
$$G_3 = B_3$$

K-Map for G_2 :



$$G_2 = B_3 \oplus B_2$$

K-Map for G_1 :



$$G1 = B1 \oplus B2$$

K-Map for G_0 :

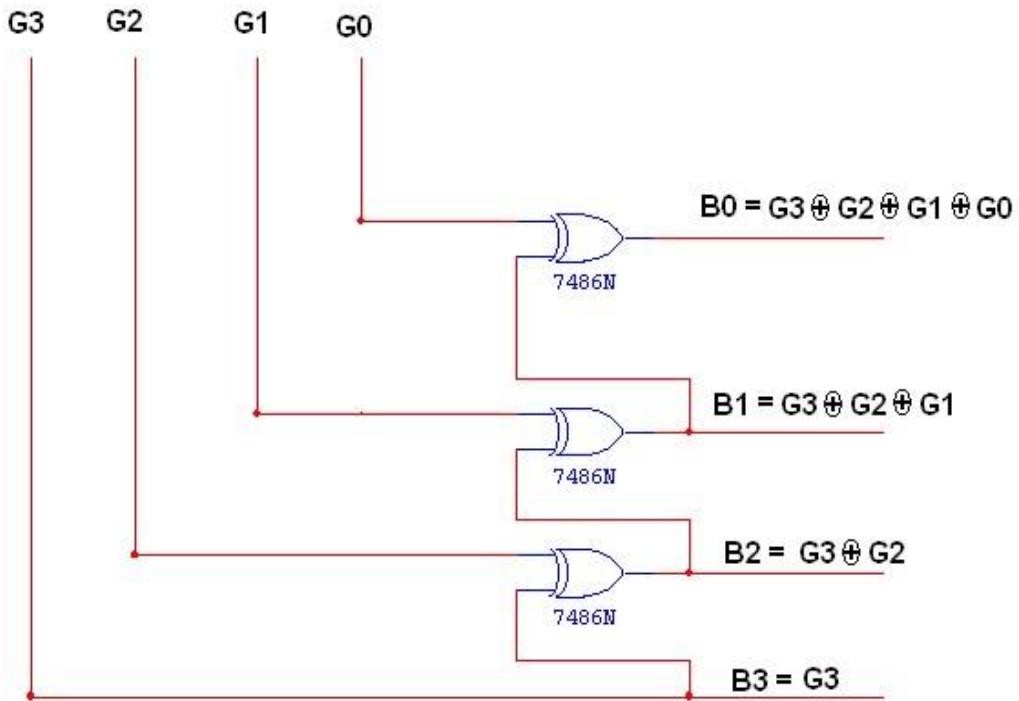
		B1B0	00	01	11	10
		B3B2	00	1		1
			01	1		1
			11	1		1
			10	1		1

$$G0 = B1 \oplus B0$$

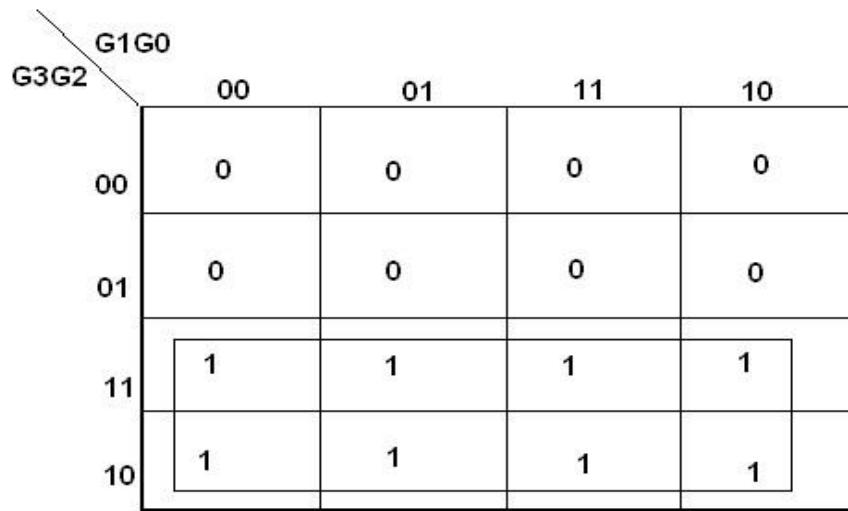
TRUTH TABLE:

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

LOGIC DIAGRAM:
GRAY CODE TO BINARY CONVERTOR



K-Map for B_3 :



$$B_3 = G_3$$

K-Map for B_2 :

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

$$B_2 = G_3 \oplus G_2$$

K-Map for B_1 :

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

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

K-Map for B_0 :

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

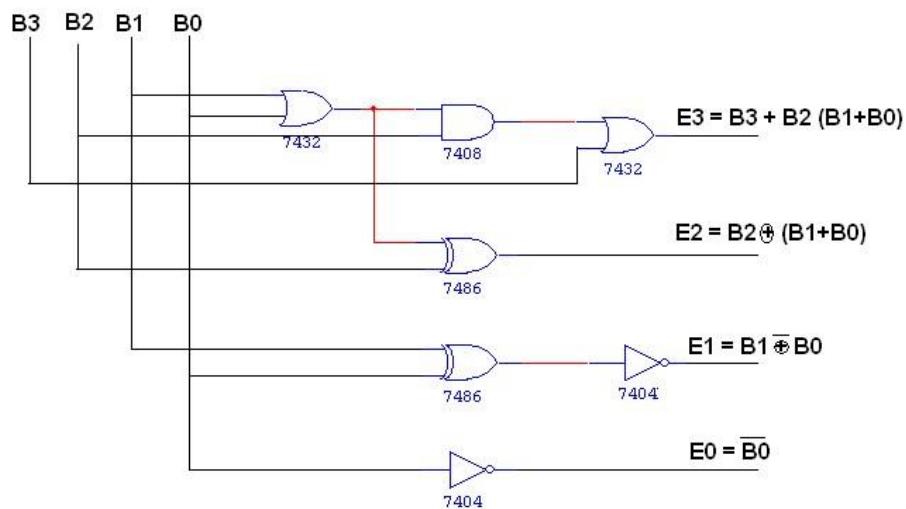
$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

TRUTH TABLE:

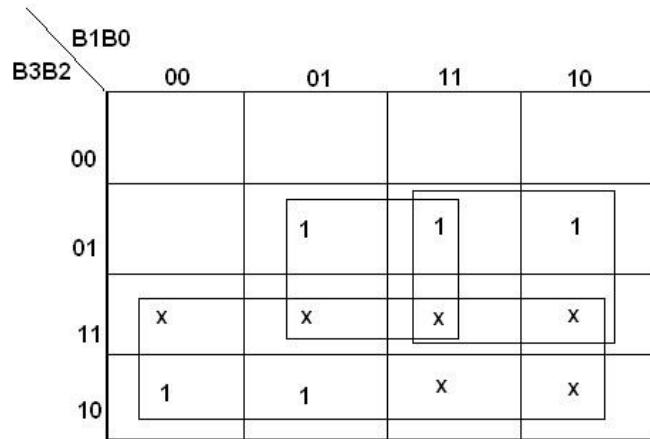
Gray Code				Binary Code			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	1	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

LOGIC DIAGRAM:

BCD TO EXCESS-3 CONVERTOR

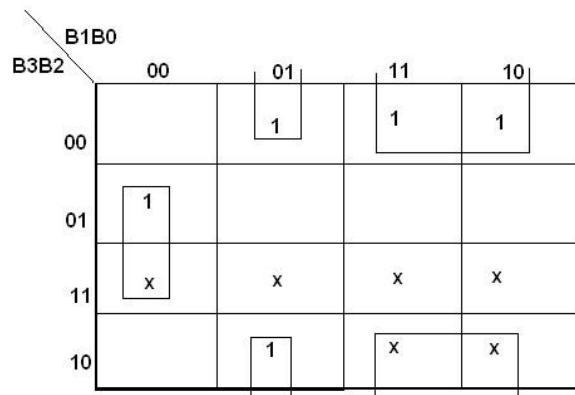


K-Map for E₃:



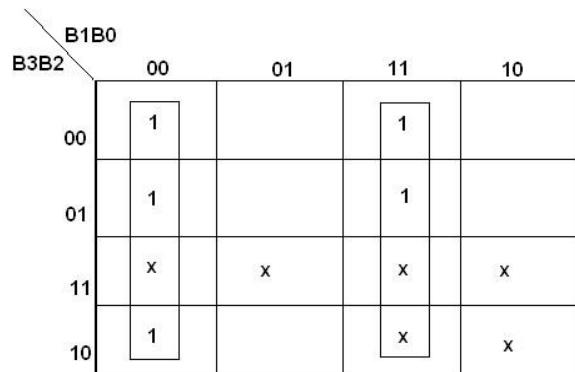
$$E_3 = B_3 + B_2 (B_0 + B_1)$$

K-Map for E₂:



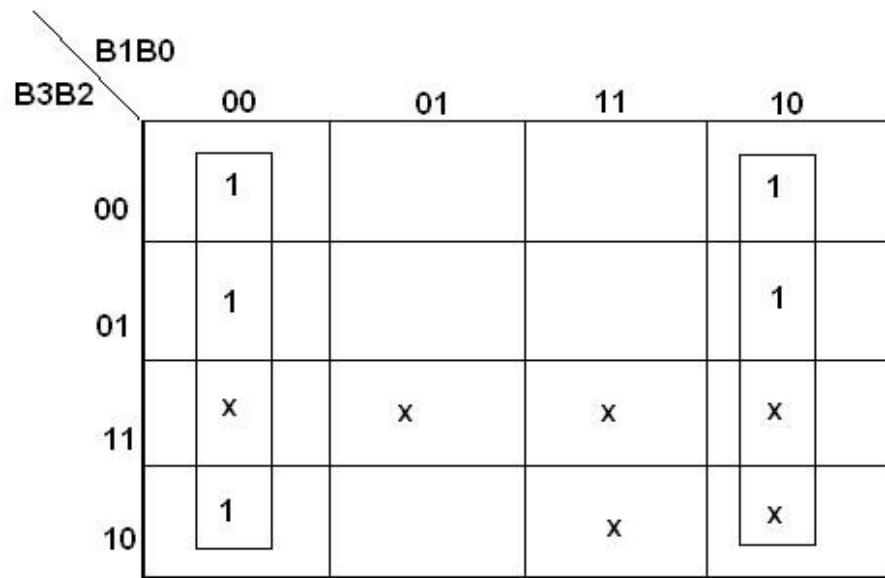
$$E_2 = B_2 \oplus (B_1 + B_0)$$

K-Map for E₁:



$$E_1 = B_1 \bar{+} B_0$$

K-Map for E₀:



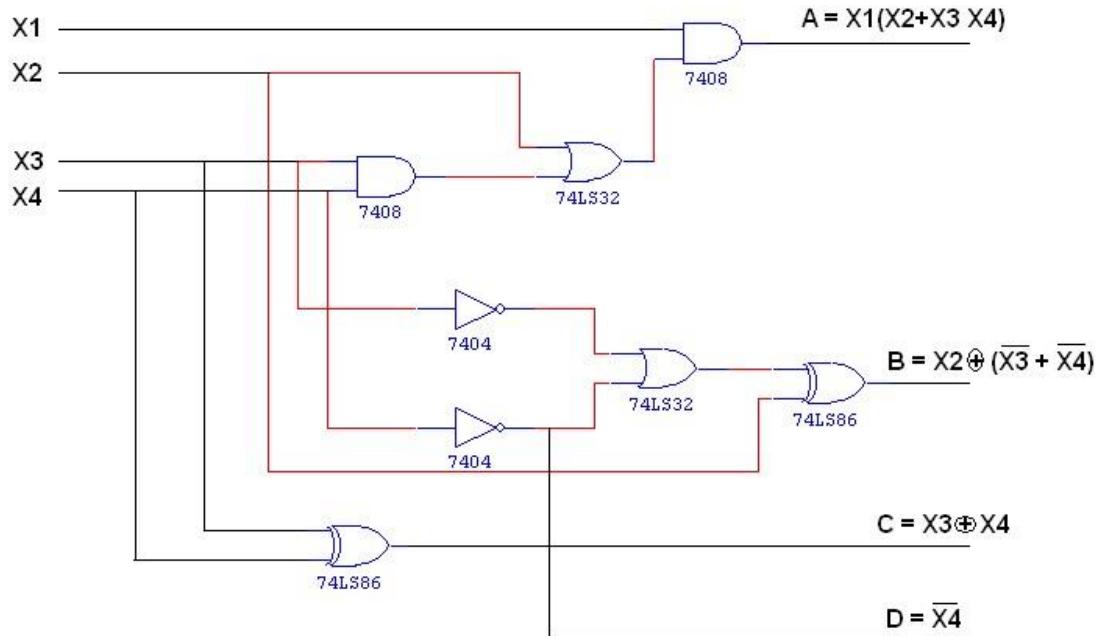
$$E_0 = \overline{B_0}$$

TRUTH TABLE:

BCD input				Excess – 3 output				
B3	B2	B1	B0	G3	G2	G1	G0	
0	0	0	0	0	0	1	1	
0	0	0	1	0	1	0	0	
0	0	1	0	0	1	0	1	
0	0	1	1	0	1	1	0	
0	1	0	0	0	1	1	1	
0	1	0	1	1	0	0	0	
0	1	1	0	1	0	0	1	
0	1	1	1	1	0	1	0	
1	0	0	0	1	0	1	1	
1	0	0	1	1	1	0	0	
1	0	1	0	x	x	x	x	
1	0	1	1	x	x	x	x	
1	1	0	0	x	x	x	x	
1	1	0	1	x	x	x	x	
1	1	1	0	x	x	x	x	
1	1	1	1	x	x	x	x	

LOGIC DIAGRAM:

EXCESS-3 TO BCD CONVERTOR

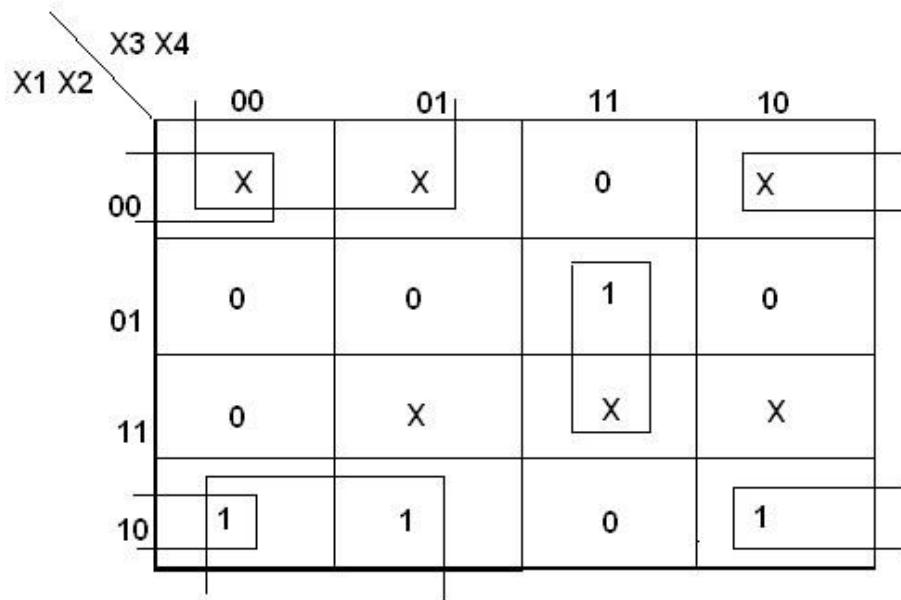


K-Map for A:

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

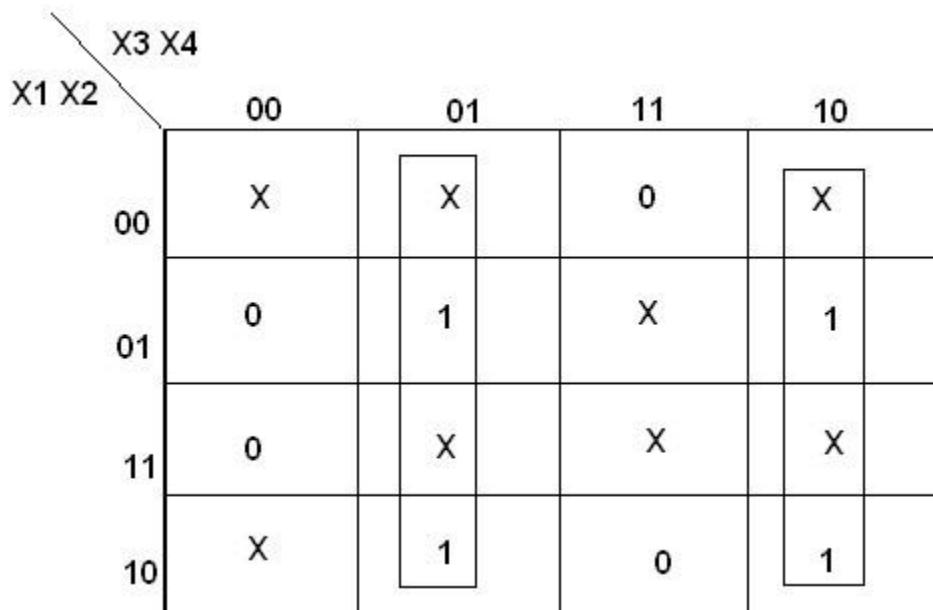
$$A = X_1 X_2 + X_3 X_4 X_1$$

K-Map for B:



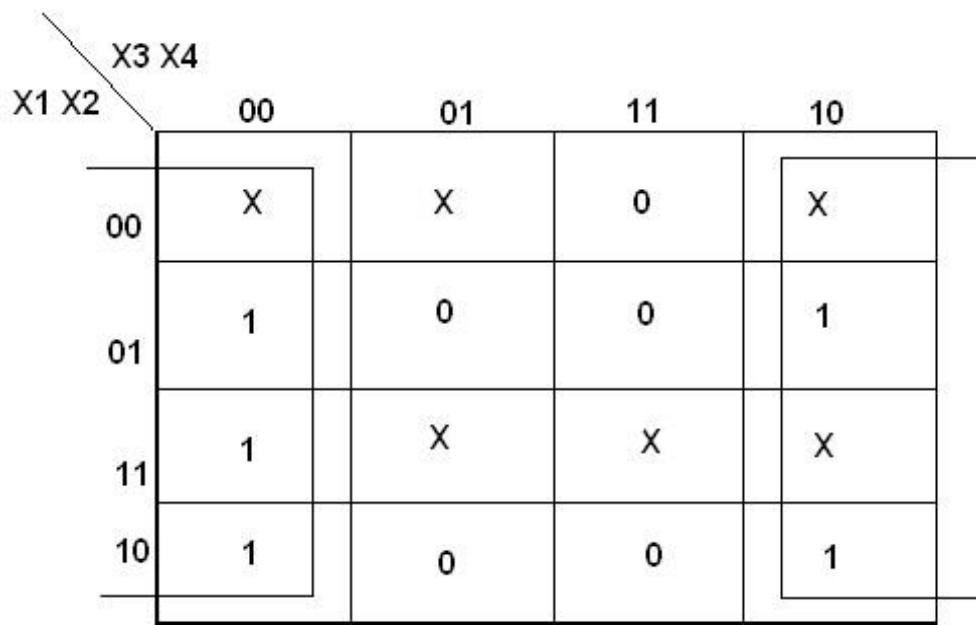
$$B = X_2 \oplus (\overline{X}_3 + \overline{X}_4)$$

K-Map for C:



$$C = X_3 \oplus X_4$$

K-Map for D:



$$D = \overline{X_4}$$

TRUTH TABLE:

Excess – 3 Input				BCD Output			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

PROCEDURE:

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

RESULT:

EXPT NO. : 5

DATE :

IMPLEMENTATION OF BCD ADDER ENCODER AND DECODER

AIM:

To Implement Using BCD adder encoder and decoder using logic gates of IC 7445 and IC 74147.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	3 I/P NAND GATE	IC 7410	2
2.	OR GATE	IC 7432	3
3.	NOT GATE	IC 7404	1
2.	IC TRAINER KIT	-	1
3.	PATCH CORDS	-	27

THEORY:

ENCODER:

An encoder is a digital circuit that performs inverse operation of a decoder. An encoder has 2^n input lines and n output lines. In an encoder the output lines generate the binary code corresponding to the input value. In octal to binary encoder it has eight inputs, one for each octal digit and three output that generate the

ambiguity that when all inputs are zero the outputs are zero. The zero outputs can also be generated when $D_0 = 1$.

DECODER:

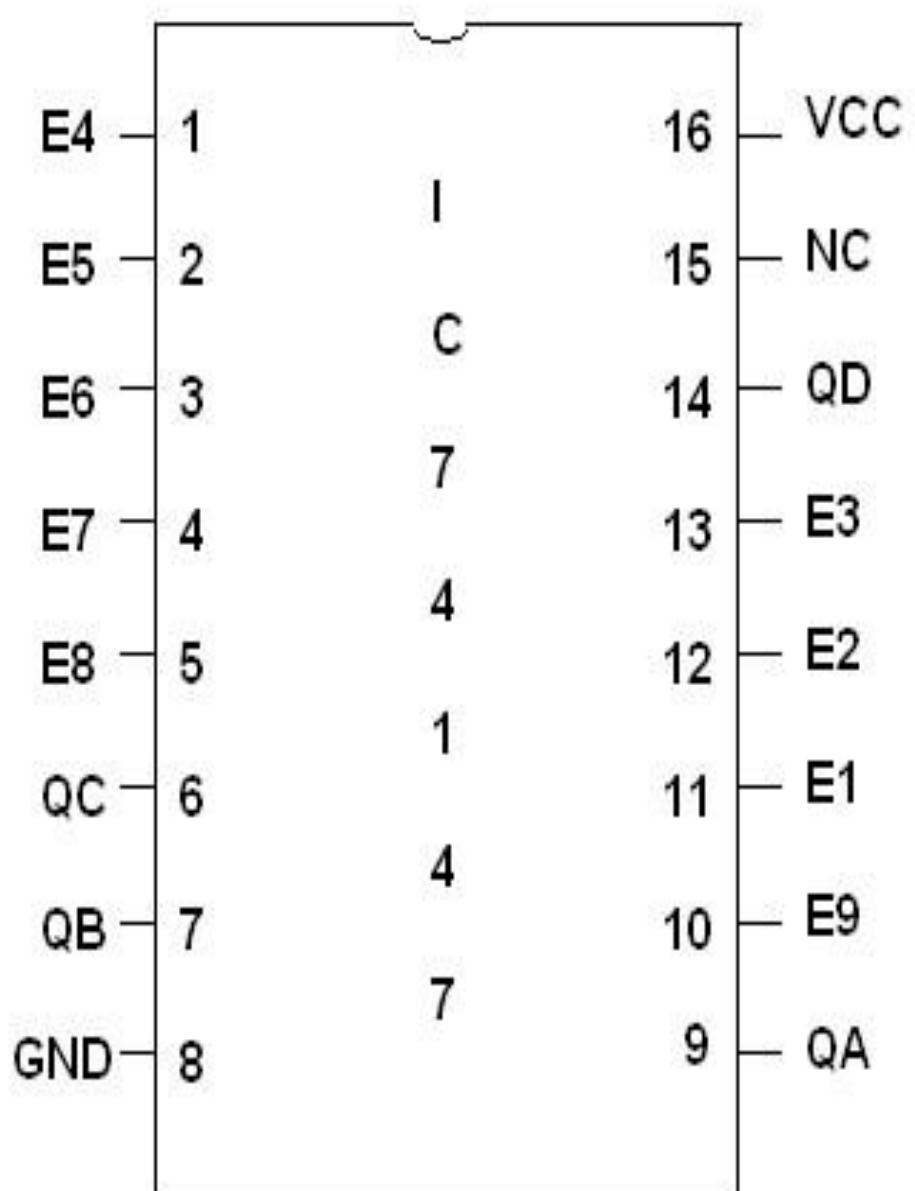
A decoder is a multiple input multiple output logic circuit which converts coded input into coded output where input and output codes are different. The input code generally has fewer bits than the output code. Each input code word produces a different output code word i.e there is one to one mapping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

PIN DIAGRAM FOR IC 7445:

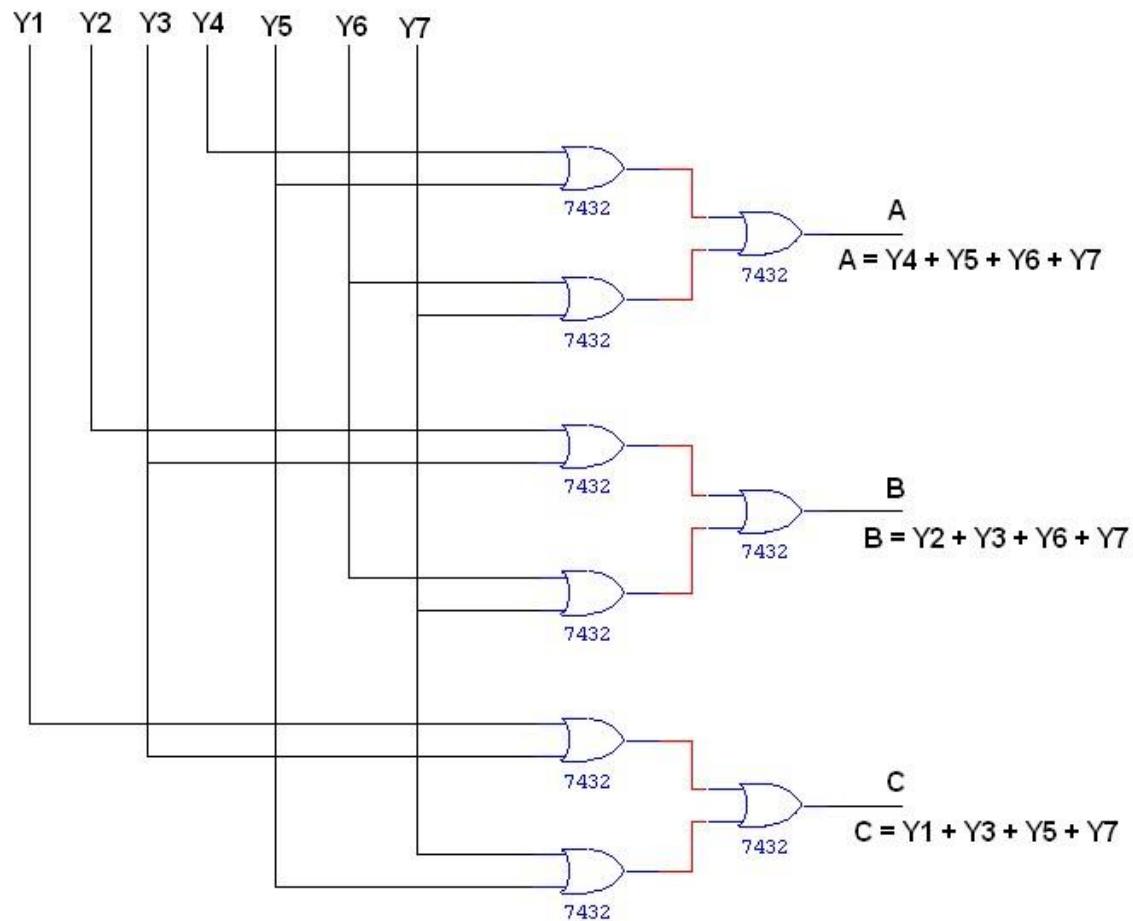
BCD TO DECIMAL DECODER:

1	O/P	VCC	16
2	O/P	I	I/P 15
3	O/P	C	I/P 14
4	O/P	7	I/P 13
5	O/P	4	I/P 12
6	O/P	4	O/P 11
7	O/P	5	O/P 10
8	GND		O/P 9

PIN DIAGRAM FOR IC 74147:



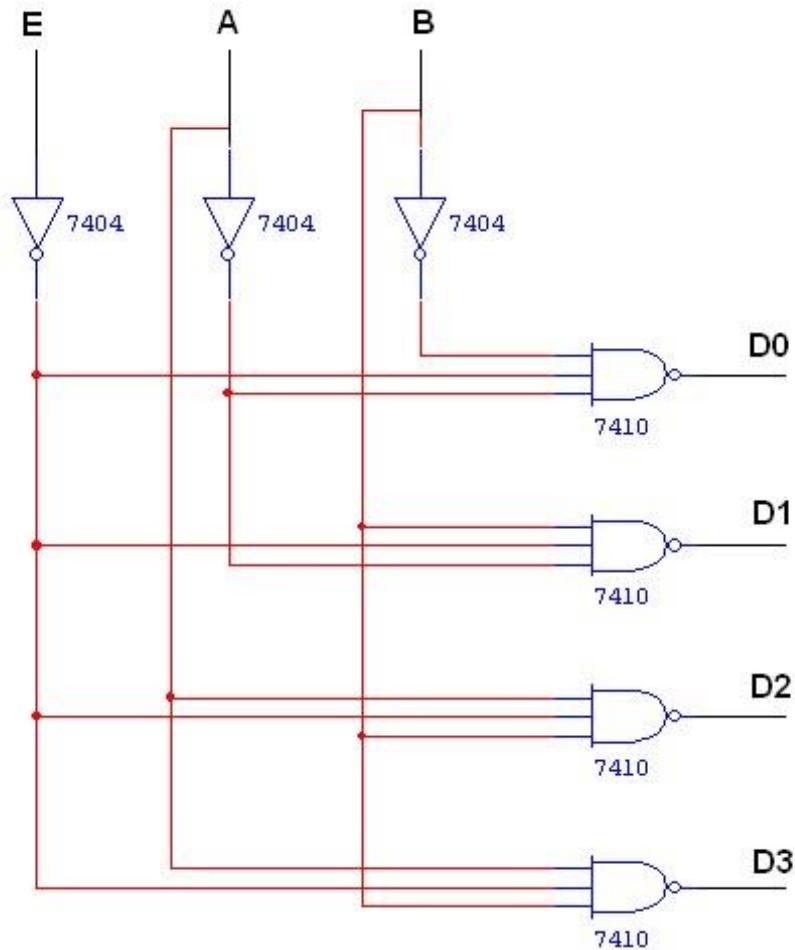
LOGIC DIAGRAM FOR ENCODER:



TRUTH TABLE:

INPUT							OUTPUT		
Y1	Y2	Y3	Y4	Y5	Y6	Y7	A	B	C
1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1
0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	1	1	1	1

LOGIC DIAGRAM FOR DECODER:



TRUTH TABLE:

INPUT			OUTPUT			
E	A	B	D0	D1	D2	D3
1	0	0	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

PROCEDURE:

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

RESULT:

EXPT NO. : 6

DATE :

DESIGN AND IMPLEMENTATION OF MULTIPLEXER

AIM:

To design and implement multiplexer and demultiplexer using logic gate and study of IC 74150 and IC 74154.

APPARATUS REQUIRED:

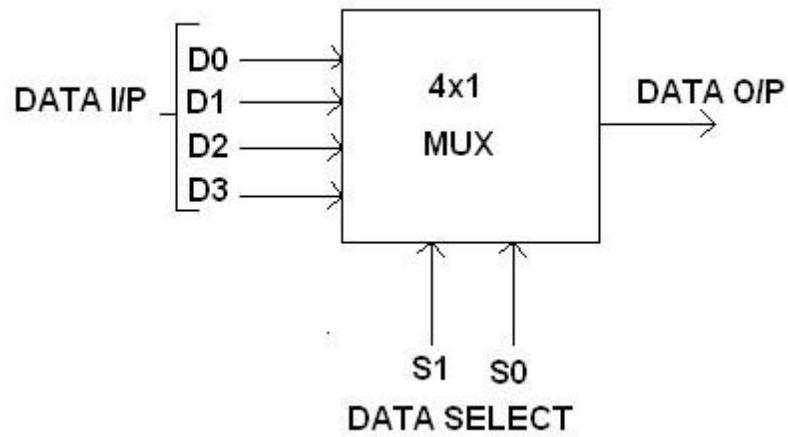
S. No.	COMPONENT	SPECIFICATION	QTY.
1.	IC	IC 7411	2
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
2.	IC TRAINER KIT	-	1
3.	PATCH CORDS	-	32

THEORY:

MULTIPLEXER:

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

BLOCK DIAGRAM FOR 4:1 MULTIPLEXER:

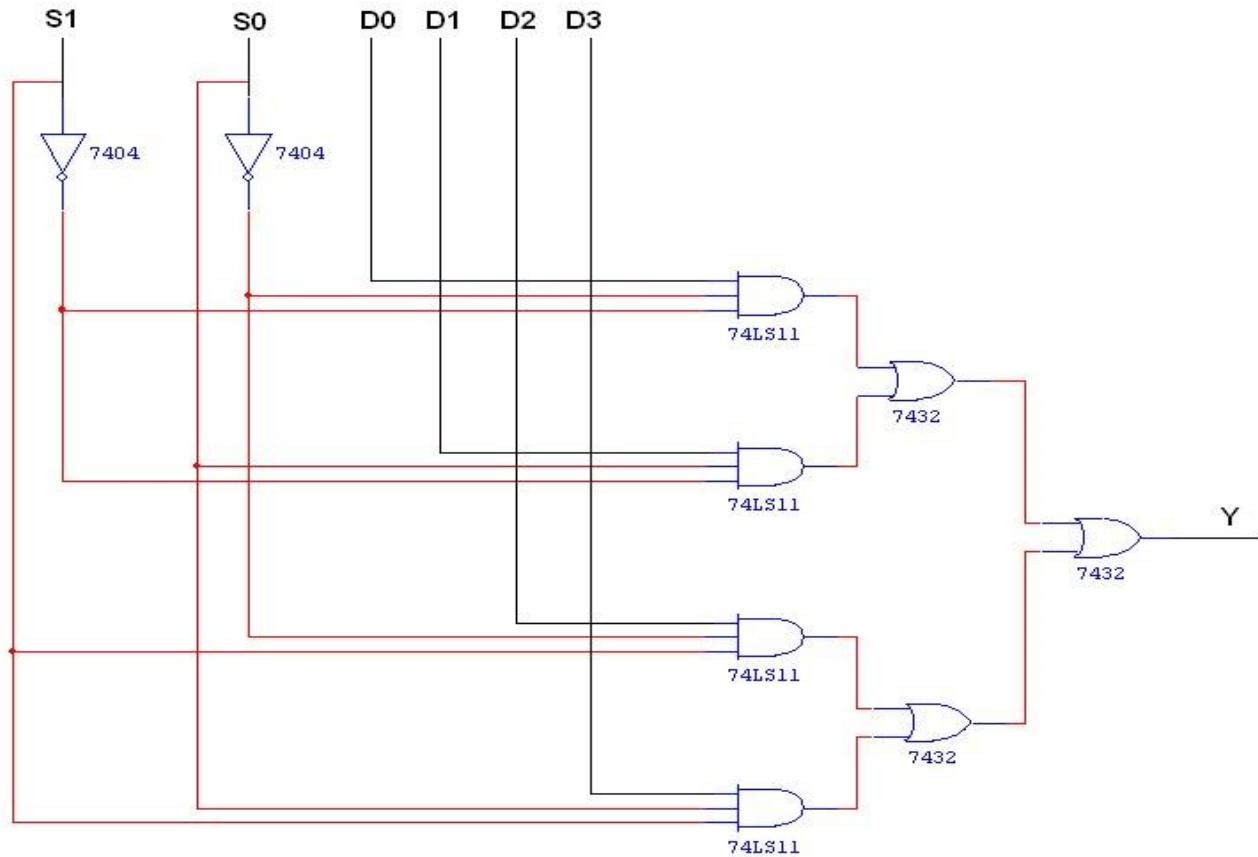


FUNCTION TABLE:

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

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

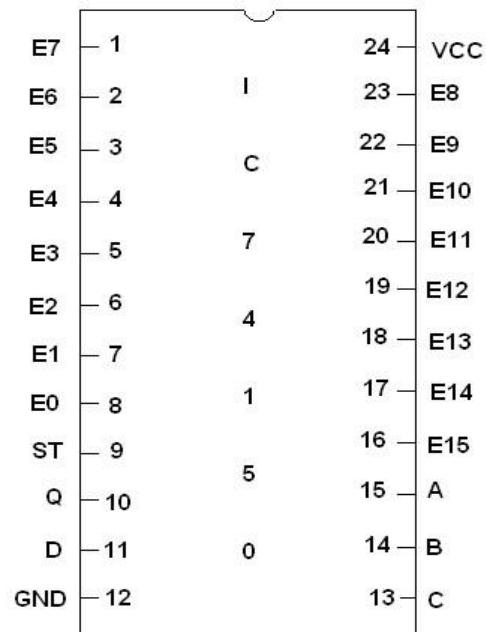
CIRCUIT DIAGRAM FOR MULTIPLEXER:



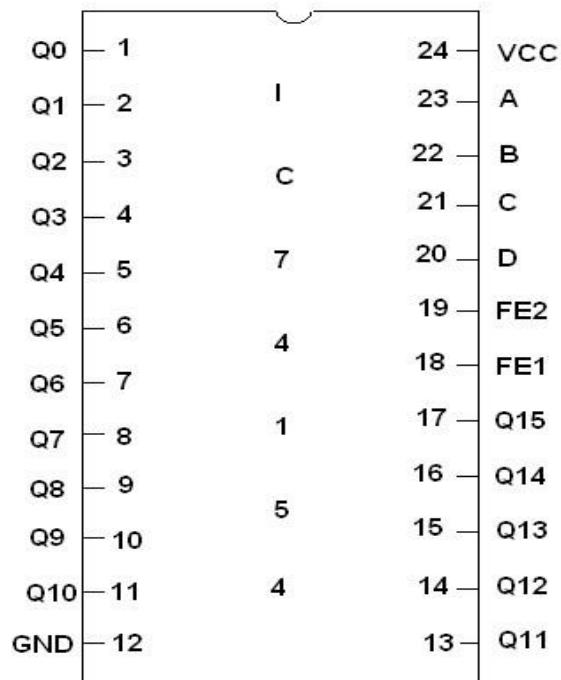
TRUTH TABLE:

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

PIN DIAGRAM FOR IC 74150:



PIN DIAGRAM FOR IC 74154:



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:

EXPT NO. : 7

DATE :

IMPLEMENTATION OF SYNCHRONOUS COUNTER

AIM:

To Implement 3 bit synchronous up/down counter.

APPARATUS REQUIRED:

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

THEORY:

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. An up/down counter is one that is capable of progressing in increasing order or decreasing order through a certain sequence. An up/down counter is also called bidirectional counter. Usually up/down operation of the counter is controlled by up/down signal. When this signal is high counter goes through up sequence and when up/down signal is low counter follows reverse sequence.

K - MAP

QB QC	UD QA	1	0	0	0
UD QA	QB QC	X	X	X	X
QB QC	UD QA	1	0	0	0
UD QA	QB QC	X	X	X	X
QB QC	UD QA	0	0	1	0

$$JA = \overline{UD} \overline{QB} \overline{QC} + UD QB QC$$

QB QC	UD QA	X	X	X	X
UD QA	QB QC	1	0	0	0
QB QC	UD QA	0	0	1	0
UD QA	QB QC	X	X	X	X
QB QC	UD QA	X	X	X	X

$$KA = \overline{UD} \overline{QB} \overline{QC} + UD QB QC$$

QB QC	UD QA	1	X	X	1
UD QA	QB QC	1	X	X	1
QB QC	UD QA	1	X	X	1
UD QA	QB QC	1	X	X	1
QB QC	UD QA	1	X	X	1

$$JC = 1$$

QB QC	UD QA	1	0	X	\overline{X}
UD QA	QB QC	1	0	X	\overline{X}
QB QC	UD QA	1	0	X	\overline{X}
UD QA	QB QC	0	1	X	\overline{X}
QB QC	UD QA	0	1	X	\overline{X}

$$JB = \overline{UD} \oplus QC$$

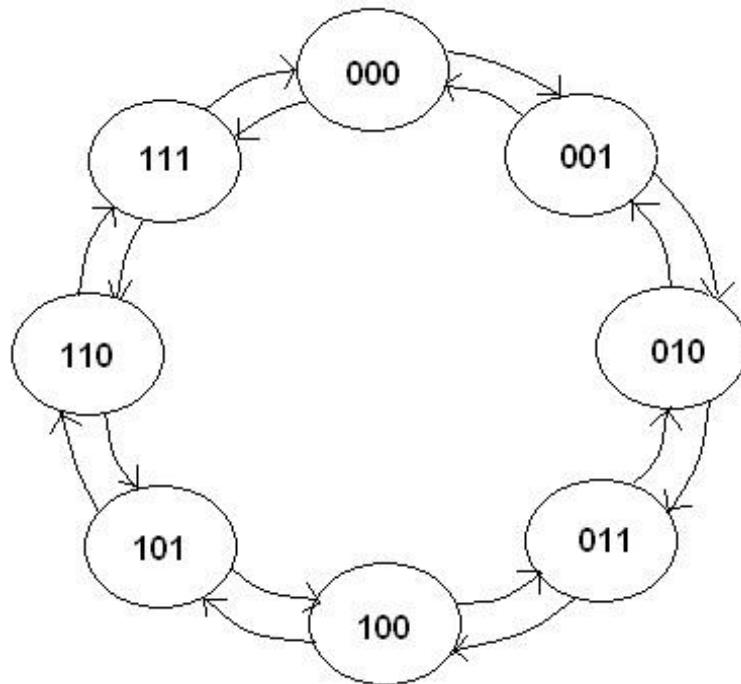
QB QC	UD QA	X	X	0	1
UD QA	QB QC	X	X	0	1
QB QC	UD QA	X	X	1	0
UD QA	QB QC	X	X	1	0
QB QC	UD QA	X	X	1	0

$$KB = \overline{(UD \oplus QC)}$$

QB QC	UD QA	X	1	1	X
UD QA	QB QC	X	1	1	X
QB QC	UD QA	X	1	1	X
UD QA	QB QC	X	1	1	X
QB QC	UD QA	X	1	1	X

$$KC = 1$$

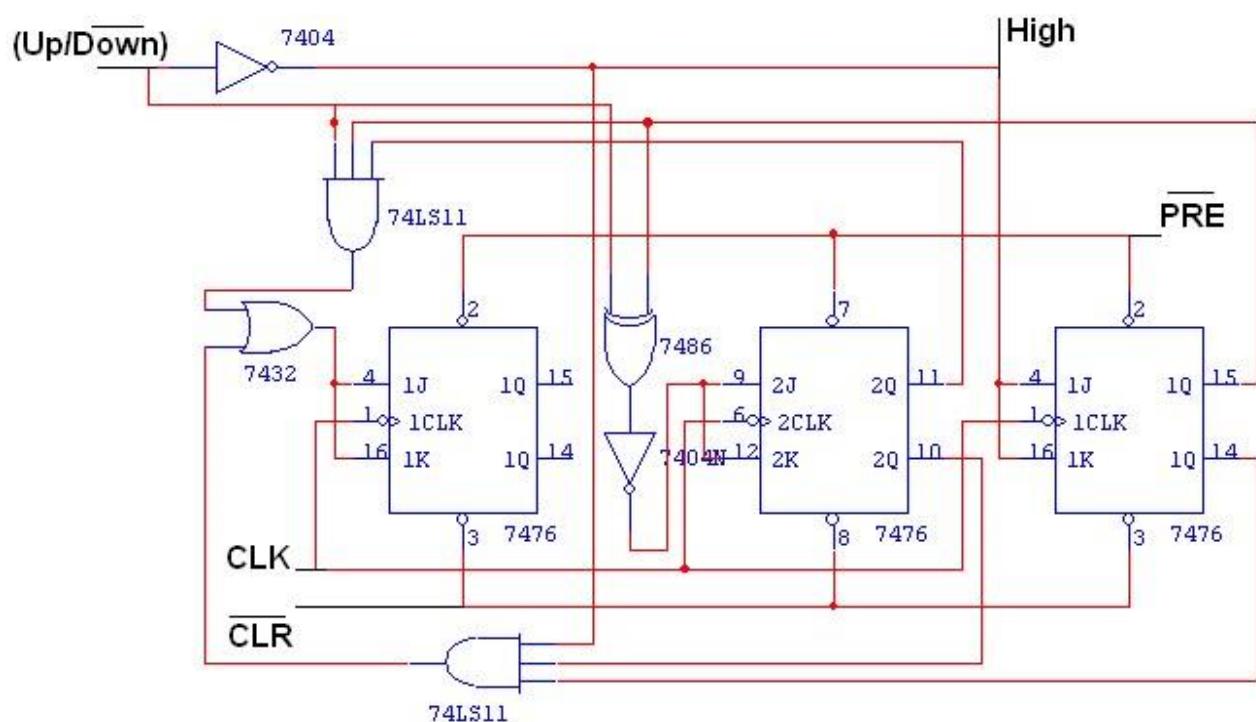
STATE DIAGRAM:



CHARACTERISTICS TABLE:

Q	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

LOGIC DIAGRAM:



TRUTH TABLE:

Input Up/Down	Present State			Next State			A		B		C	
	Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	1	1	1	1	X	1	X	1	X
0	1	1	1	1	1	0	X	0	X	0	X	1
0	1	1	0	1	0	1	X	0	X	1	1	X
0	1	0	1	1	0	0	X	0	0	X	X	1
0	1	0	0	0	1	1	X	1	1	X	1	X
0	0	1	1	0	1	0	0	X	X	0	X	1
0	0	1	0	0	0	1	0	X	X	1	1	X
0	0	0	1	0	0	0	0	X	0	X	X	1
1	0	0	0	0	0	1	0	X	0	X	1	X
1	0	0	1	0	1	0	0	X	1	X	X	1
1	0	1	0	0	1	1	0	X	X	0	1	X
1	0	1	1	1	0	0	1	X	X	1	X	1
1	1	0	0	1	0	1	X	0	0	X	1	X
1	1	0	1	1	1	0	X	0	1	X	X	1
1	1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	1	0	0	0	X	1	X	1	X	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:

EXP NO :08

DATE :

IMPLEMENTATION OF A UNIVERSAL SHIFT REGISTER

AIM:

To implement the universal shift register using lm7S45LM125

APPARTUS REQUIRED

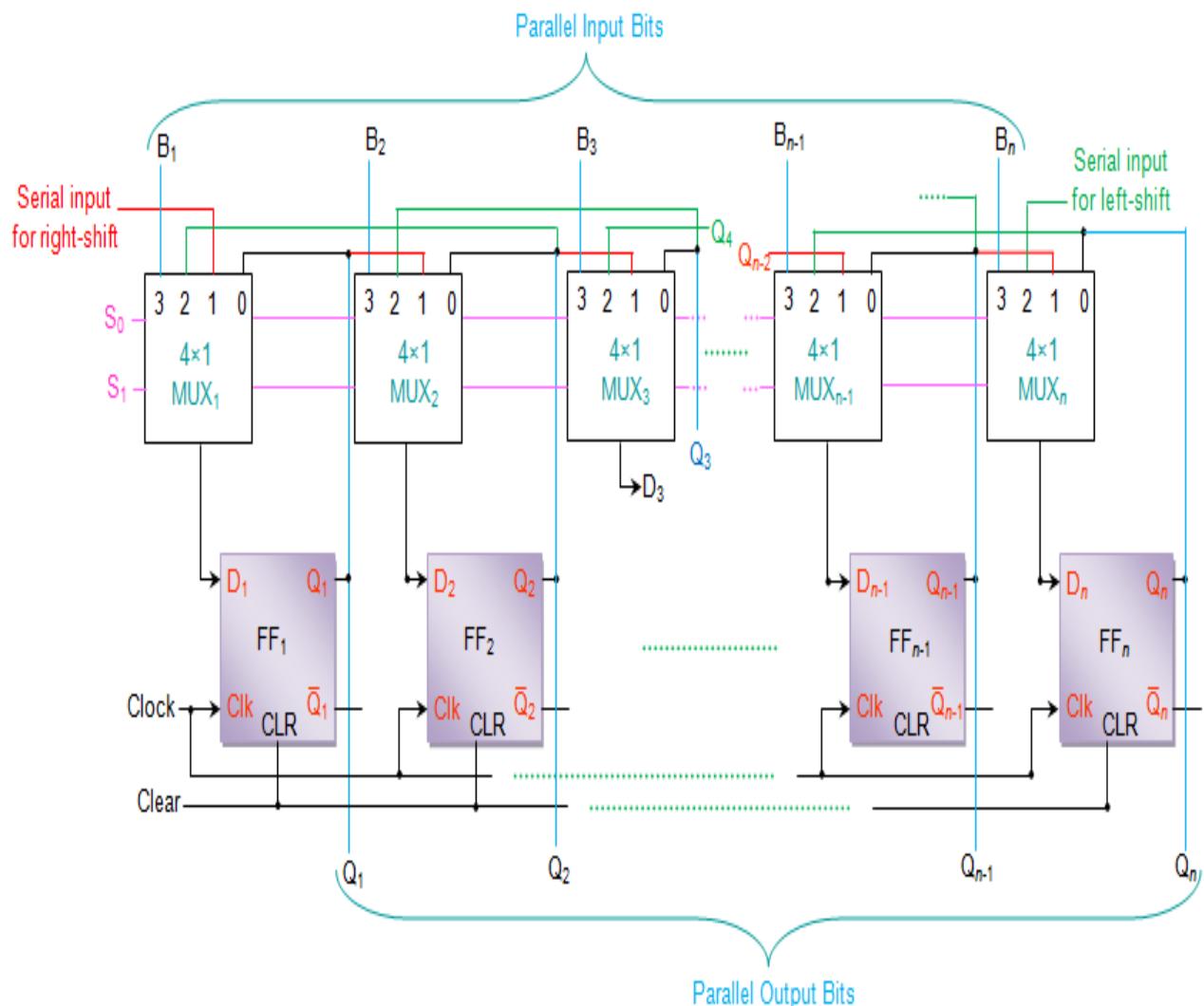


Figure 1 n -bit Universal Shift Register

THEORY :

niversal Shift Register is a register which can be configured to load and/or retrieve the data in any mode (either serial or parallel) by shifting it either towards right or towards left. In other words, a combined design of unidirectional (either right- or left-shift of data bits as in case of **SISO**, **SIPO**, **PISO**, **PIPO**) and **bidirectional shift register** along with parallel load provision is referred to as **universal shift register**. Such a **shift register** capable of storing n input bits $\times 1$ multiplexers to drive the input pins of n flip-flops in the register which are also connected to clock and clear inputs. All of the **multiplexers** in the circuit share the same select lines, S_1 and S_0 (pink lines in the figure), in order to select the mode in which the shift registers operates. It is also seen that the MUX driving a particular **flip-flop** has its

First input (Pin Number 0) connected to the output pin of the same flip-flop i.e. zeroth pin of MUX_1 is connected to Q_1 , zeroth pin of MUX_2 is connected to Q_2 , ... zeroth pin of MUX_n is connected to Q_n .

Second input (Pin Number 1) connected to the output of the very-previous flip-flop (except the first flip-flop FF_1 where it acts like an serial-input to the input data bits which are to be shifted towards right) i.e. first pin of MUX_2 is connected to Q_1 , first pin of MUX_3 is connected to Q_2 , ... first pin of MUX_n is connected to Q_{n-1} .

Third input (Pin Number 2) connected to the output of the very-next flip-flop (except the first flip-flop FF_n where it acts like an serial-input to the input data bits which are to be shifted towards left) i.e. second pin of MUX_1 72

is connected to Q_2 , second pin of MUX_2 is connected to Q_3, \dots second pin of MUX_{n-1} is connected

TRUTH TABLE

Table II Truth Table for n -bit Universal Shift Register

Serial Input for Left Shift				$L_1 L_2 \dots L_n$			
Serial Input for Right Shift				$R_1 R_2 \dots R_n$			
Parallel Input				$B_1 B_2 \dots B_n$			
Clk	CLR	Mux Output	Outputs				
			Q_1	Q_2	---	Q_{n-1}	Q_n
1	1	X	0	0	---	0	0
2	0	1	R_1	0	---	0	0
3	0	1	R_2	R_1	---	0	0
.	---	.	.
.	---	.	.
$n+1$	0	1	R_n	R_{n-1}	---	R_2	R_1
$n+2$	1	X	0	0	---	0	0
$n+3$	0	2	0	0	---	0	L_1
$n+4$	0	2	0	0	---	L_1	L_2
.	---	.	.
.	---	.	.
$2n+2$	0	2	L_1	L_2	---	L_{n-1}	L_n
$2n+3$	0	0	L_1	L_2	---	L_{n-1}	L_n
$2n+4$	0	0	L_1	L_2	---	L_{n-1}	L_n
$2n+5$	0	3	B_1	B_2	---	B_{n-1}	B_n
.	---	.	.
.	---	.	.

Register is Cleared

Right-shift of Data Bits

Register is Cleared

Left-shift of Data Bits

No Change

Parallel Data Loading

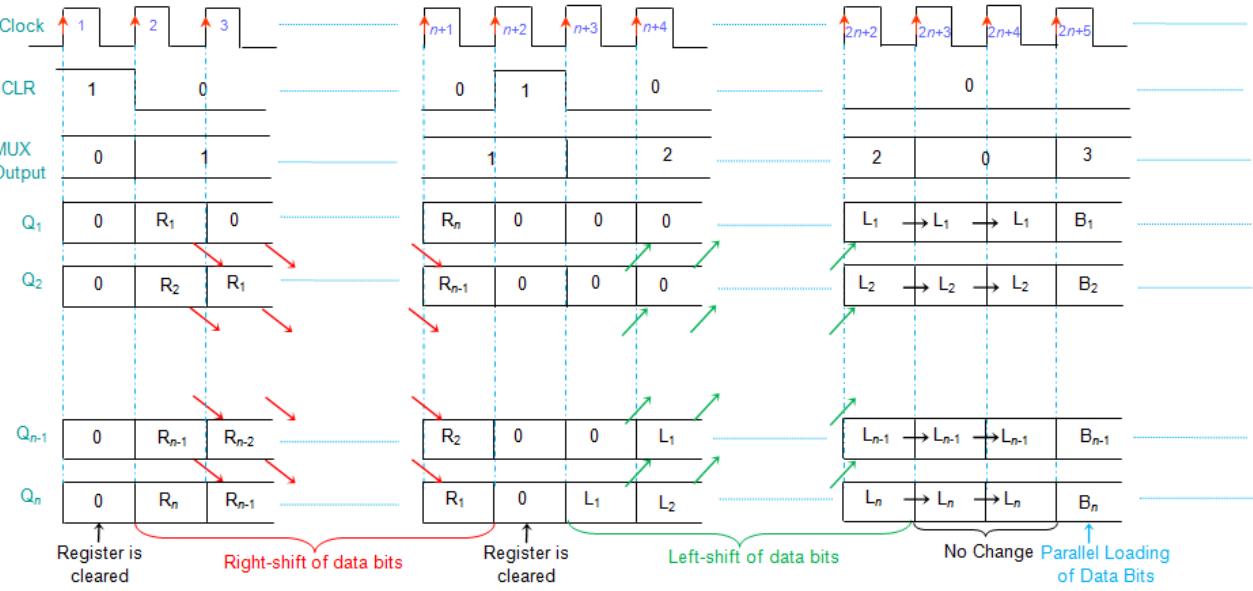


Figure 2 Output Waveform of Universal Shift Register

PROCEDURE:

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

RESULT :

EXP NO :9

DATE :

SIMULATOR BASED STUDY OF COMPUTER ARCHITECTURE

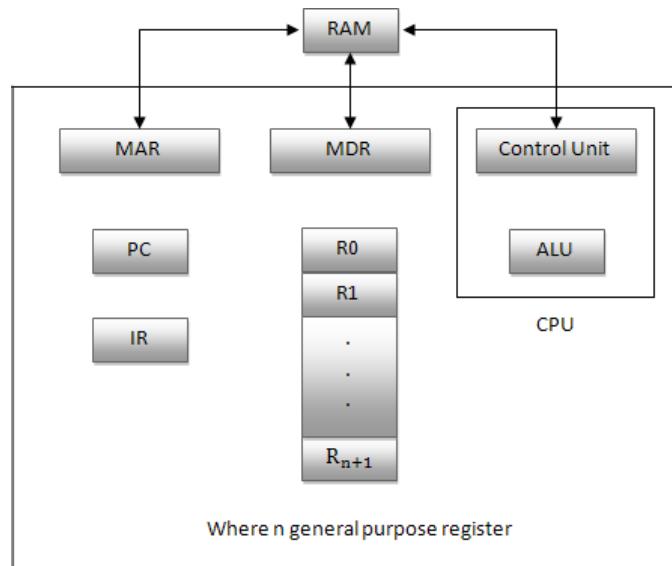


Fig: Internal Connection between Processor & Memory.

1. MAR:

- MAR means Memory Address Register.
- It holds the address of the main memory from where data is to be transferred.

2. PC:

- PC means Program Counter.
- It holds the address of the next instructions.

3. IR:

- IR means Instruction Register.
- It stores the instruction that is currently being executed.
- It gives the operation to the CU to generate the timing signal that controls the execution.

4. MDR:

- MDR means Memory Data Register / Buffer Register.
- It holds the data which write into / Read out of address location.

5. Register:

- n General purpose register R₀ through R_{n+1} is used for storing data.

6. CU:

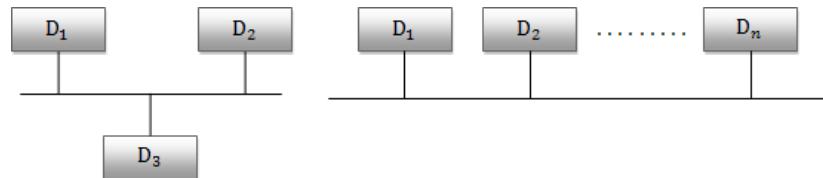
- CU means Control Unit.
- The memory, arithmetic & logic, input & output units store & process information & perform input / output operation. These operation units

7. ALU:

- ALU means Arithmetic Logic Unit.
- It performs the arithmetic operation like addition, subtraction etc.

8. Bus:

- It is a common path way connecting two or more devices.
- It is a shared transmission medium.



- At a time only one device should be transmitted signal.

9. Width of Bus:

Width of Bus means at a time how many amount of data (bit) can be transferred.

10. Single Bus Structure:

- The system structure where all units are connected to a bus.
- The bus can be used for only one transfer at a time so that only two units can actively use the bus at any given time.

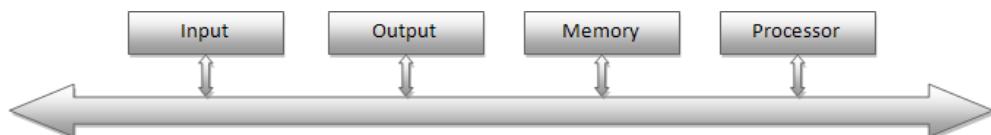


Fig: Single Bus Structure

Advantages:

- ✓ Low cost
- ✓ Flexibility
- ✓ Easy to design

Disadvantages:

- ✓ Only one data transfer at a time.
- ✓ Time consuming to any program.

Multiple Bus Structure:

- The system structure where multiple buses are used.
- Multiple buses active more concurrently in operation by allowing two or more transfers.

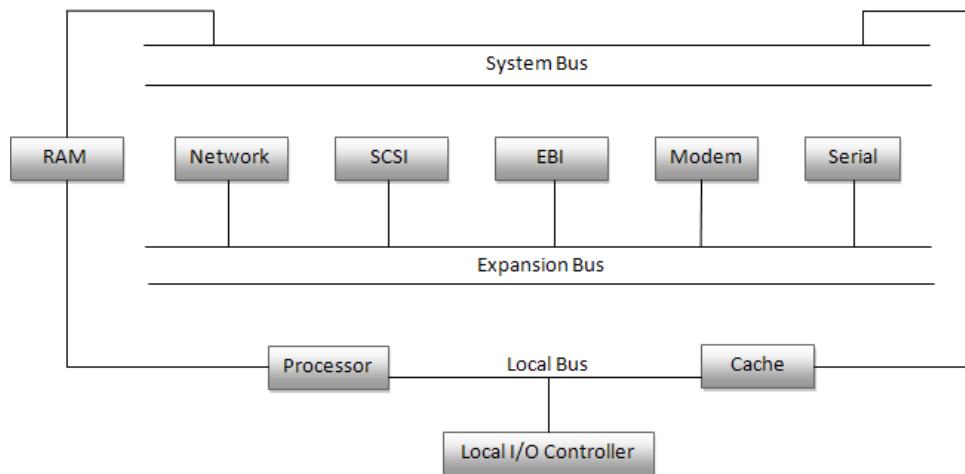


Fig: Multiple Bus structure

Advantages:

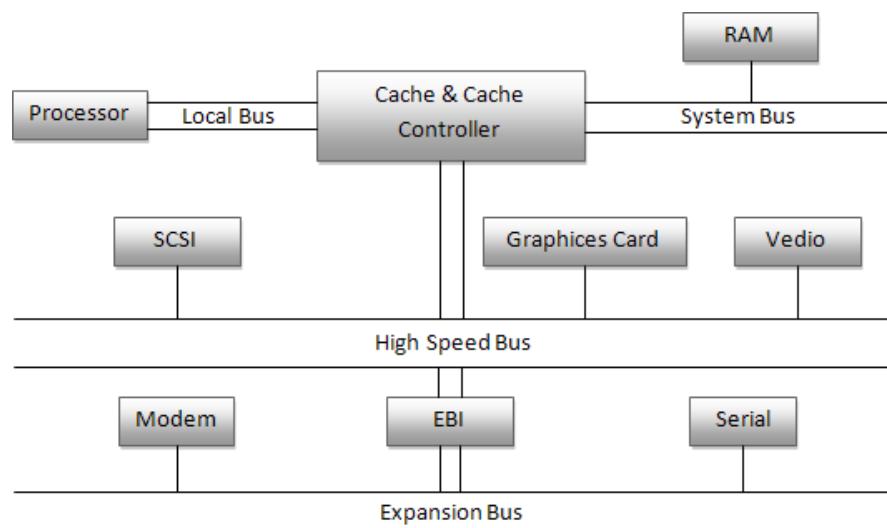
- ✓ Allows the system to support a wider variety of devices.
- ✓ More speed than single bus structure.

Disadvantage:

- ✓ Low performance due to the more devices greater the bus length propagation delays.
- ✓ The bus become bottleneck as the aggregate data transfer rate increase towards the capacity of the bus.

High Speed Architecture:

- High speed bus brings high devices closer to the Processor.
- High speed buses are used.
- They are independent to the processor.



77

Fig: High Speed Architecture

Advantage:

- ✓ High speed

Disadvantage:

- ✓ Costly
- ✓ Implementation is very difficult

System Bus:

- A bus that connect major components (Processor, memory, I/O).
- It consists of 50 to 100 separate lines; each line is used for a particular purpose.
- 50 to 100 lines can classify into functional groups.
 - i. Data lines.
 - ii. Address lines.
 - iii. Control lines.

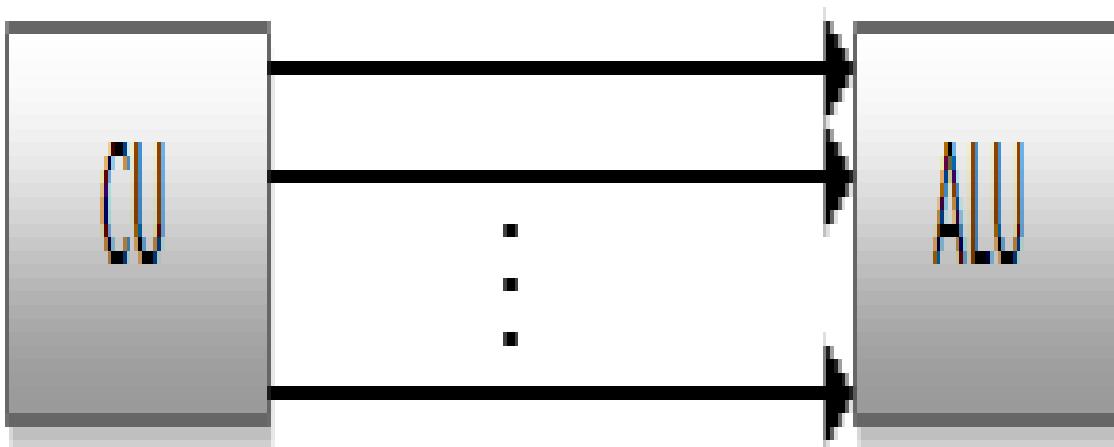


Fig: System Bus

i. Address bus:

- ✓ Address lines are collectively called address bus.
- ✓ 4-bit address bus means $2^4 = 16$ B. So, if address width is 20 bit then $2^{20} = 1$ MB.

ii. Control bus:

- ✓ Control lines are collectively called control bus.
- ✓ Used to control signals.

Bus Arbitration:

- The process of selecting bus master. That means select the device which can transmit data on the bus right now.
- Only one bus master at a time.

Daisy Chaining Approach:

- Every device must be set a priority.
- If device1 priority is greater than other device then device1 is selected as a bus master.

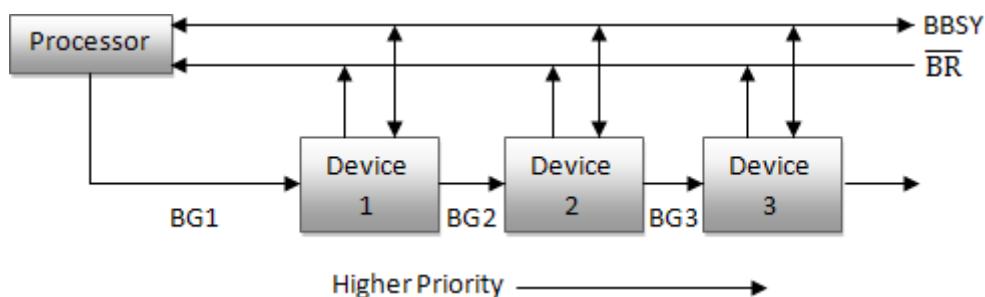


Fig: Daisy Chaining Approach

8. Advantage:

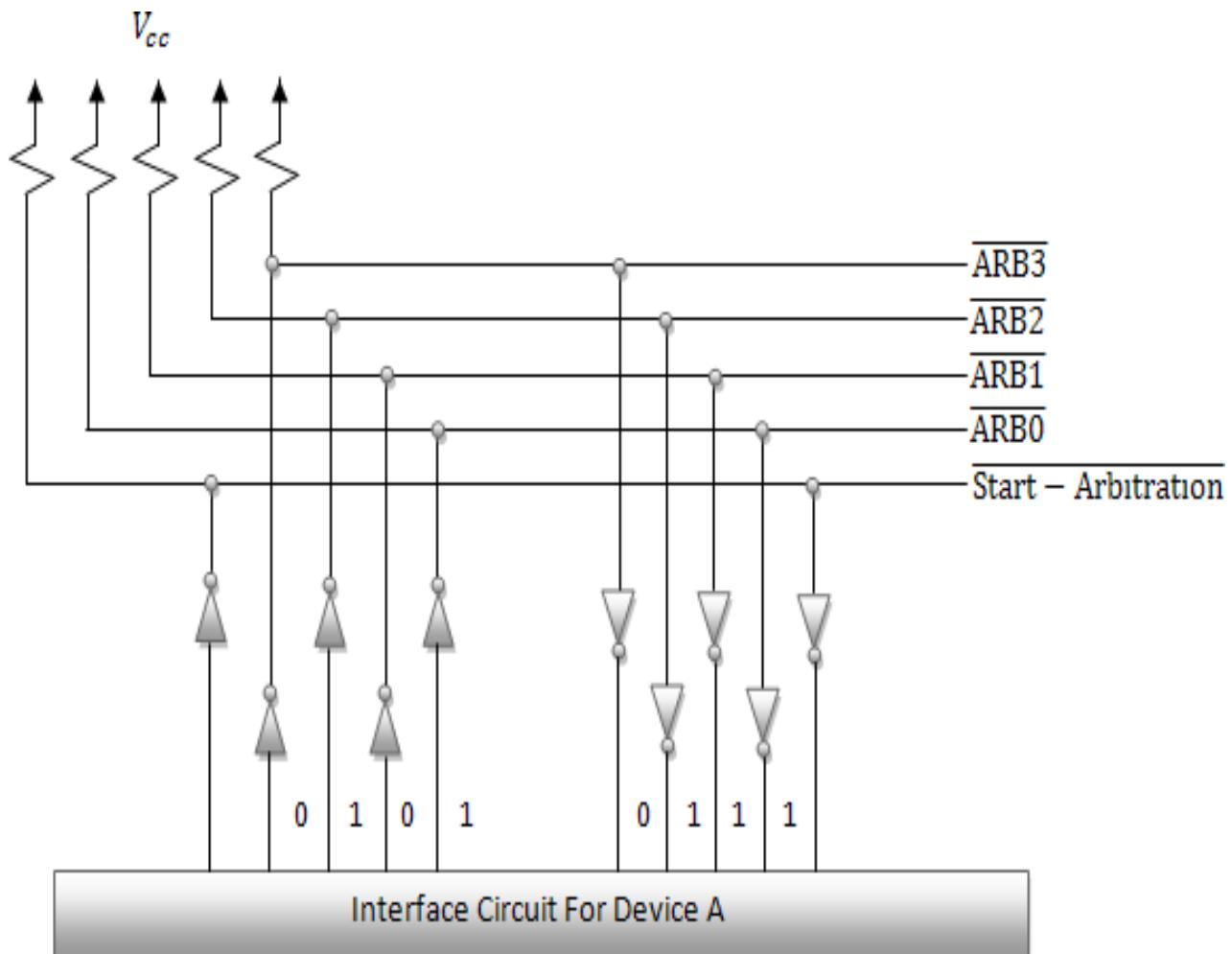
- ✓ Very simple concept
- ✓ Centralized, very easy to select bus master.
- ✓ Synchronizing process.

9. Disadvantage:

- ✓ Starvation problem (bored for waiting)
- ✓ Propagation delay
- ✓ The entire system fails if the higher priority device fails.

Distributed Arbitration:

- Distributed Arbitration means that all devices waiting to use the bus that have equal responsibility in carrying out the arbitration process, without using a central arbiter.
- Each device on the bus is assigned a 4-bit identification number.
- When one or more devices request the bus, they assert the ~~Start signal~~ and place their 4-bit ID numbers on their open collector lines, through ~~R~~

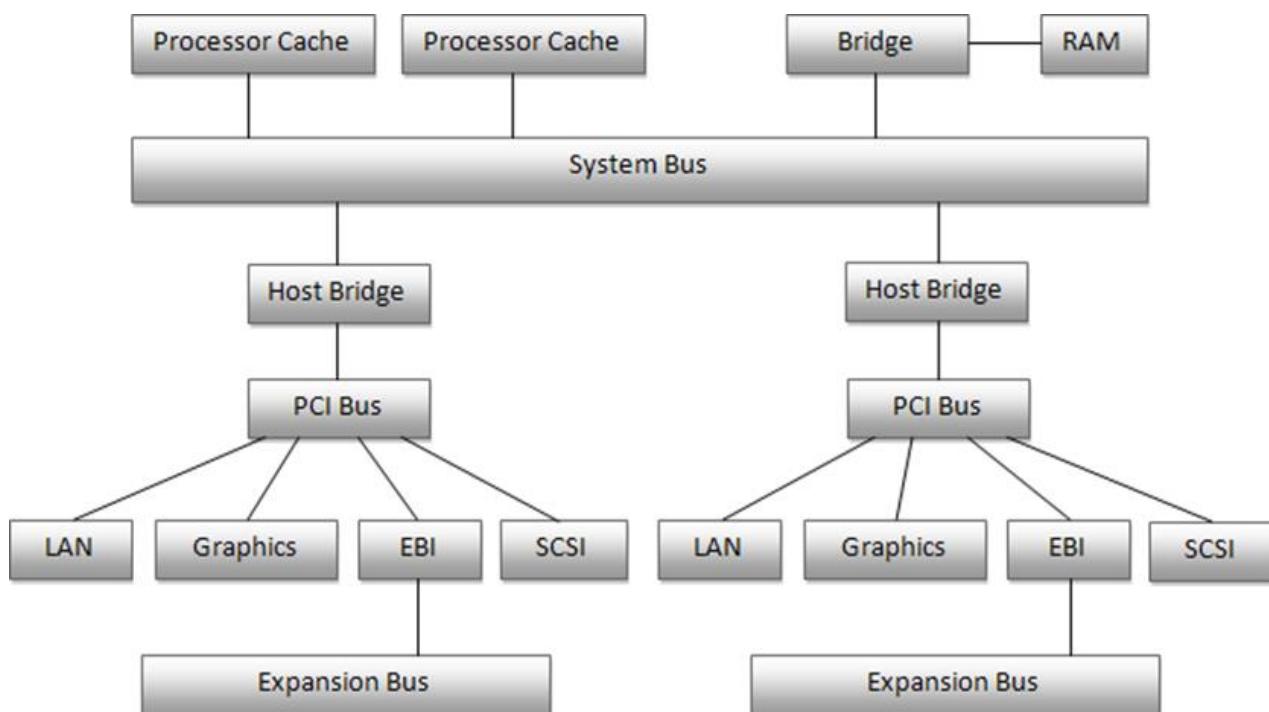


PCI Bus:

- ✓ PCI means Peripheral Component Interaction.
- ✓ It has high bandwidth & it is very popular.
- ✓ It is an independent bus that can function as a peripheral bus.
- ✓ It requires very few chips to implement & supports other busses connected to it.
- ✓ It used centralized arbitration scheme.
- ✓ It can be used in both signal Processor (desktop system) & multiprocessor (server system) system.

PnP:

- PnP means Plug and Play.
- The device which we insert into the computer & it automatically recognized & configured in computer is called PnP.
- Intel created the PnP standard and incorporated it into the design for PCI



- ✓ It is a process in which several storages of the CPU are used to execute more than one instruction concurrently.
- ✓ It is an effective way of organizing concurrently activity in any system.
- ✓ A pipelined processor may process each instruction in four steps:

- Fetch F : Read the instruction from the memory.
- Decode D : Decode the instruction & fetch the source operand.
- Execute E : Perform the operation specified by instruction.
- Write W : Store the result in the destination location.

Instructions	Clock Cycle						
	1	2	3	4	5	6	7
I ₁	F ₁	D ₁	E ₁	W ₁			
I ₂		F ₂	D ₂	E ₂	W ₂		
I ₃			F ₃	D ₃	E ₃	W ₃	
I ₄				F ₄	D ₄	E ₄	W ₄

Fig: Pipelining

Normally for I_1 to I_4 instruction needs $4 \times 4 = 16$ clock cycle. But with the use of pipeline it is done with 7 clock cycle.

Super Pipelining:

- ✓ It's an alternative approach to achieve better performance.
- ✓ Many pipeline stages perform task that requires less than half of a

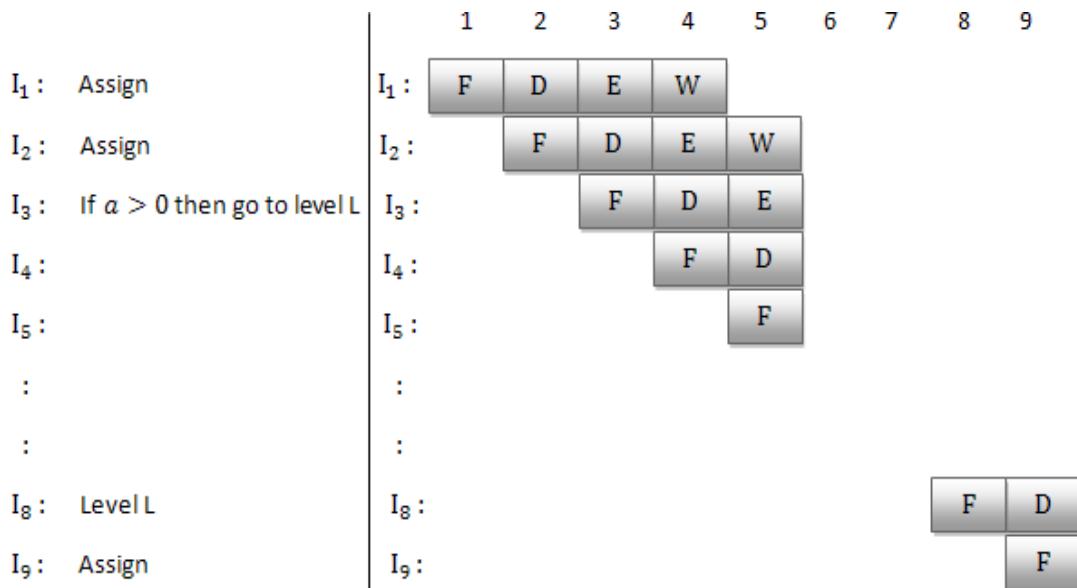
clock cycle, so a double interval clock speed allow the performance of two tasks in one clock cycle.

Instruction Level Parallelism:

The degree to which the instruction of a program can be executed parallel is called instruction level parallelism.

I ₁ :	F ₁	D ₁	E ₁	W ₁
I ₂ :	F ₂	D ₂	E ₂	W ₂
I ₃ :	F ₃	D ₃	E ₃	W ₃
I ₄ :	F ₄	D ₄	E ₄	W ₄

Effect of branch on pipelining:



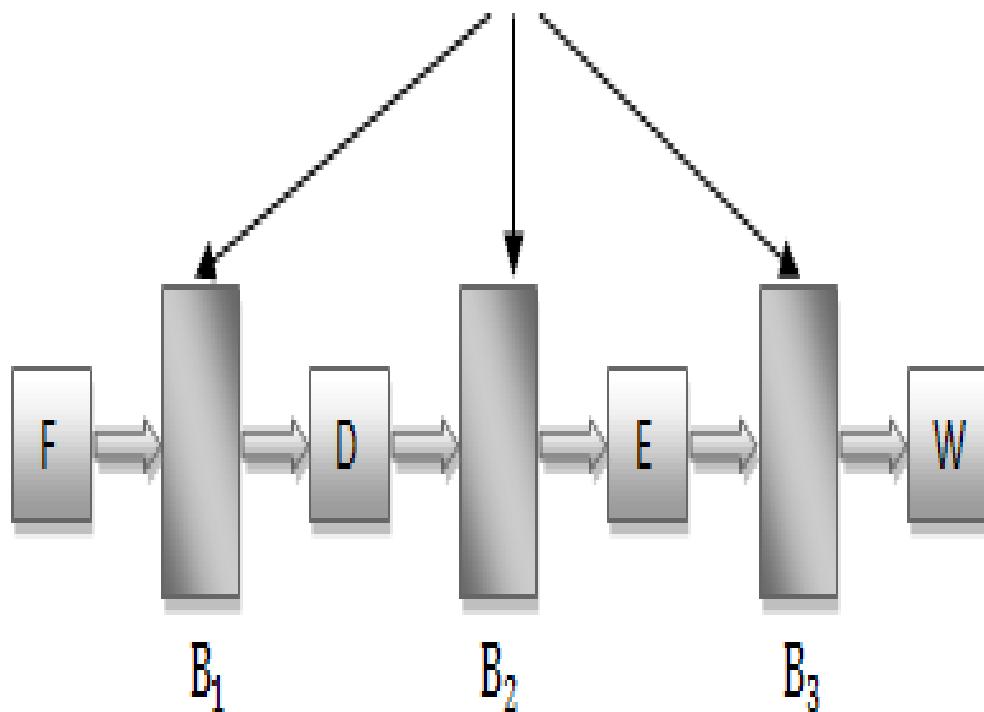
Branch Condition:

A conditional branch instruction introduces the added caused by the dependency of the branch condition on the result of a preceding instruction.

Branch Penalty:

F & D steps block the buffer until solve of the branch condition, in this situation we need some time to free this buffer. This time is called branch penalty.

Interstage Buffers



Logic to deal with branch:

Fig: Branch Penalty.

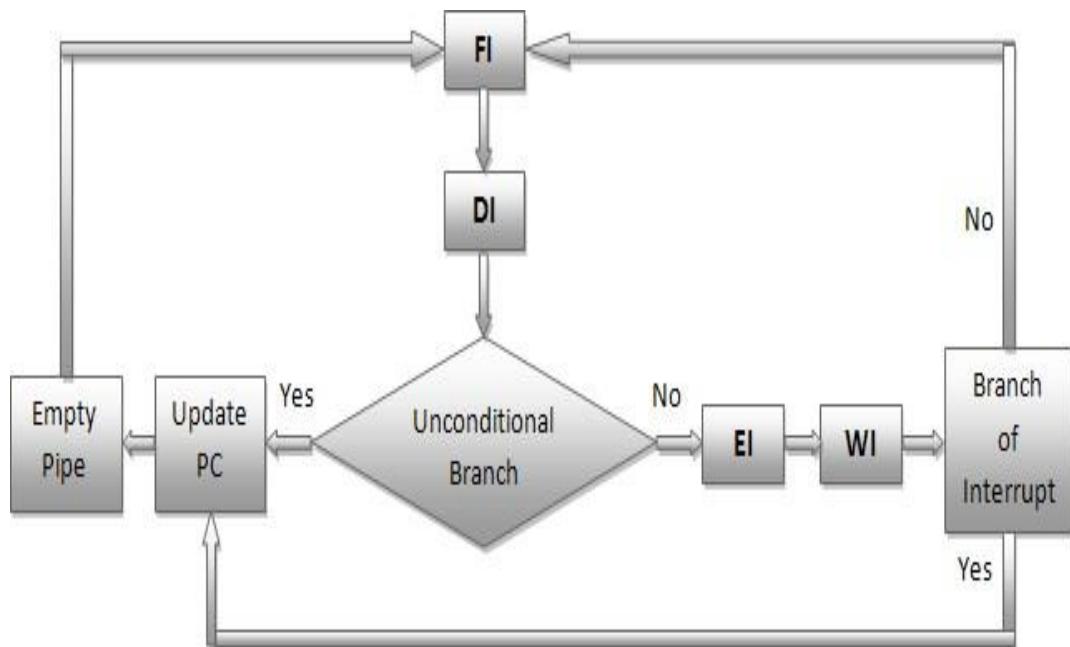


Fig: Logic to deal with branch.

Different ways of implementing a multiprocessor: Multiprocessor System:

A machine that includes an efficient high bandwidth medium for communication among the multiple processors, memory devices of I/O devices.

There are three ways to implement a multiprocessor system:

- (i) Uniform Memory Access (UMA).
- (ii) Non-uniform memory Access (NUMA).
- (iii) Distributed Memory Access (DMA).

Uniform Memory Access (UMA):

- ✓ An interconnection network permits n processors to access k memories so that any of the processors can access any of the memories.
- ✓ The interconnection network introduces considerable delay for each memory access.
- ✓ If this delay is same for all memory access then it is an UMA system.

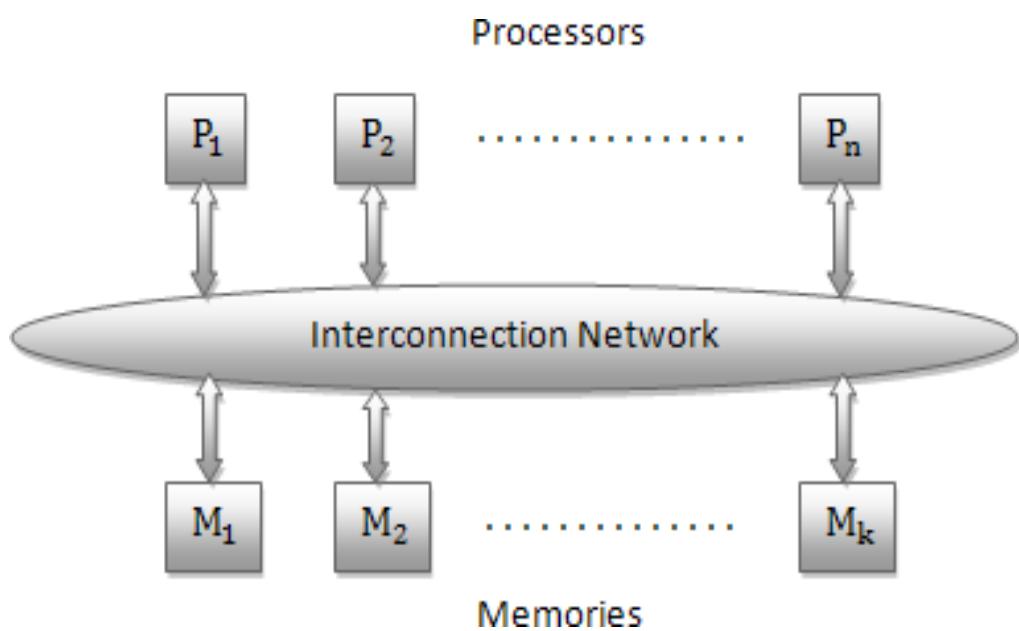


Fig: Uniform Memory Access

Non-uniform memory Access (NUMA):

- ✓ Memory modules are attached directly to the processor.
- ✓ Each processor can access its own memories (local) as well as the other processors memories via network.
- ✓ If local memory access time is T_1 and other memory access time is T_2 then $T_2 > T_1$. In this case, it is called Non-uniform memory access.

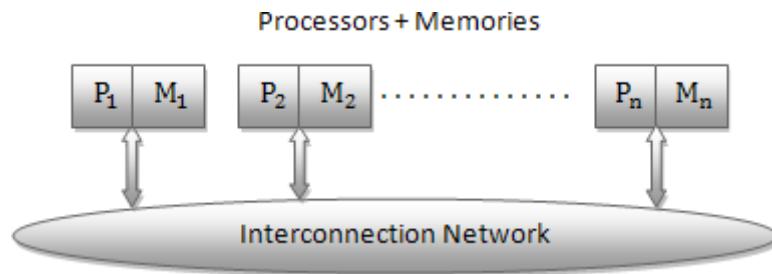


Fig: Non-uniform memory Access

Distributed Memory Access (DMA):

- ✓ All memory modules are private to their corresponding processor.
- ✓ A processor can not access a remote memory without the cooperation of the remote processor. Ex. Processor P_1 can't access the memory of P_2 until P_2 permit.
- ✓ Permission procedure is implemented by message exchanged via the network.

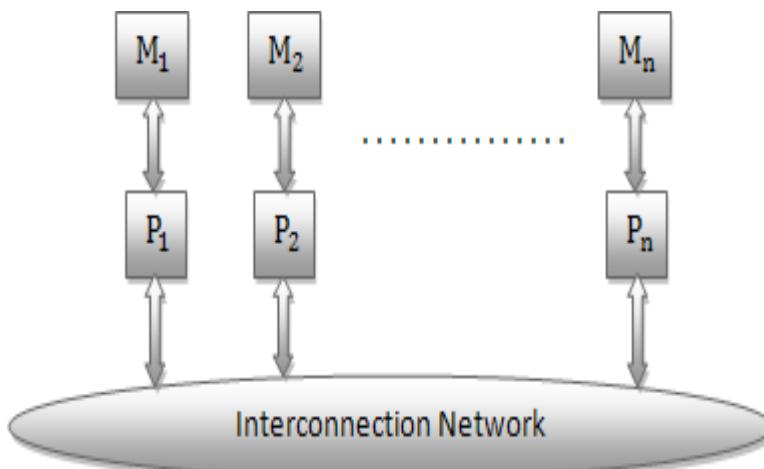


Fig: Distributed Memory

For pipelining it has fast execution rate.

- ✓ Uses VLSI technology.

Disadvantages:

- ✓ Needed more instruction than CISC to perform the same task. So, it is less effective than CISC.

CISC:

- ✓ CISC means “Complex Instruction Set Computer”.
- ✓ Small number of general purpose registers (8).
- ✓ Complex and huge number of instruction set (215).

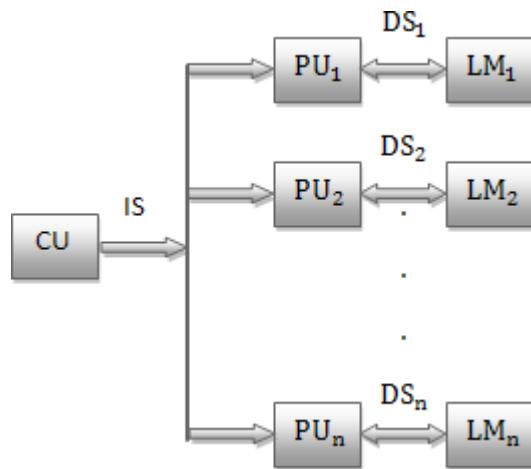
MIMD:

- ✓ MIMD means “Multiple Instruction Multiple Data”.
- ✓ A set of processors simultaneously execute different instruction stream by using different data sets.
- ✓ There are two types, such as Shared memory system & Distributed memory system.

- ✓ Program size or length smaller.
- ✓ Consume less memory for storing a program.
- ✓ Low page fault due to smaller program.

Disadvantage:

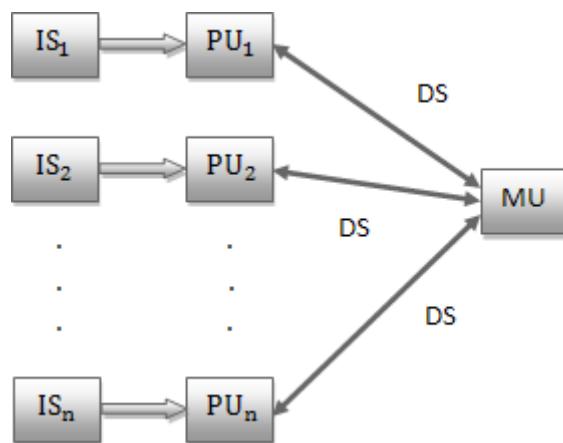
- ✓ Instructions are so complex, so, can't optimize easily.
- ✓ Cost is higher than RISC.



Example: Vector Processor, Array Processor.

MISD:

- ✓ MISD means “Multiple Instruction Single Data”.
- ✓ A sequence of instructions stream are executed by a number of processors by using data from a single memory.



Example: Systolic Array.

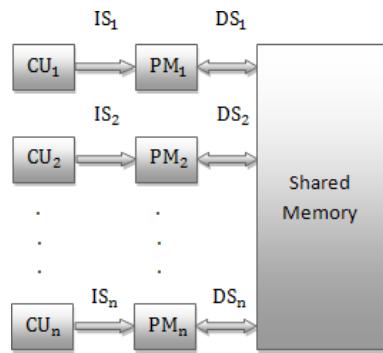


Fig: Shared Memory Schema.

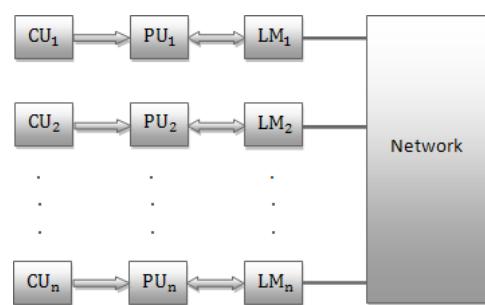
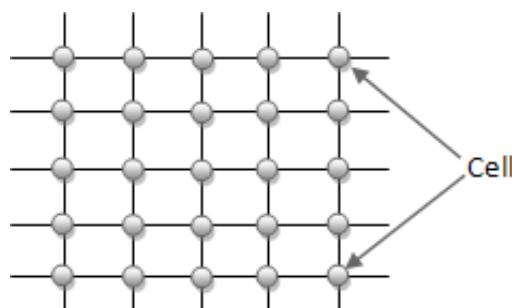


Fig: Distributed Memory System.

ROM:

- ✓ ROM means “Read Only Memory”.
- ✓ It is an IC programmed with data when manufactured.
- ✓ It is used not only computer but also in electrical devices like toys.
- ✓ It contains grid of columns and rows where the columns & rows intersect.
- ✓ A ROM chip needs programming of perfect and complicated data during manufactured.
- ✓ There is a cell. If we send a current with high voltage above then forward break over, the appropriate column with the selective row will selected.
- ✓ If any error is found then the chip thrown away & again we need manufactured.



Advantages:

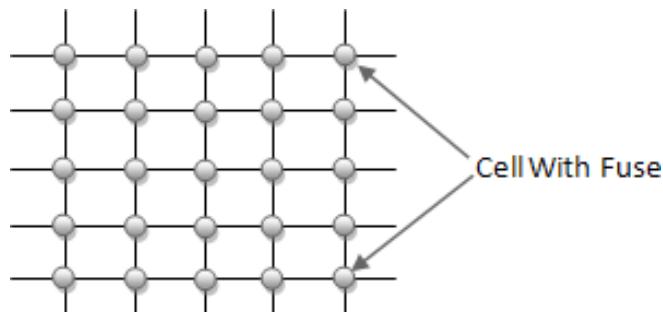
- ✓ Low power consumption.
- ✓ Cost effective.
- ✓ Extremely reliable.

Disadvantages:

- ✓ It is one time device.
- ✓ If any error then it is not usable.

PROM:

- ✓ PROM means “Programmable Read Only Memory”.
- ✓ Creating ROM chip is time consuming & expensive. So, we need to think another device like PROM.
- ✓ PROM inexpensive & can be programmed with a tool called programmer.
- ✓ Like ROM it has grids of columns & rows but the difference is in every intersection between rows and columns there is a fuse.
- ✓ Fuse is connected to logical 1 [+5V to +10V]. So, at the initial state, all cells contain logical 1. To change the value of a cell 1 to 0 the programmer is used to set a specific amount current to the cell. The higher voltage breaks the connection between rows and columns by burning out the fuse that means burning out the PROM.
- ✓ If we want bring logical 1 to 0 then apply fixed amount of current then it goes 1 to 0 and fuse is burn.



Advantage:

- ✓ After manufacture can be programmed but only once.

Disadvantage:

- ✓ PROM is expensive than other device.
- ✓ It can't program more than one.

EPROM:

- ✓ EPROM means “Erasable Programmable Read Only Memory”.
- ✓ It can be overwritten many times.
- ✓ It is a special type memory device that retains its content until the ultraviolet (UV) light is exposed, the UV light clears its contents making it possible to reprogram the memory.
- ✓ To write and erase an EPROM you need a tool called EPROM programmer.
- ✓ Each cell contains two transistors separated by an oxide layer. One of them is known as floating point gate & another is known as control gate.
- ✓ The floating point gate is connected to the row via the control gate. There is a link between them. As long as the link is in place the cell contains 1.
- ✓ To change 1 to 0 requires a process called Fowler-Nordheim Tunneling. It changes the position of electrons in the floating gate.
- ✓ An electron charge (10V to 30V) is applied to the floating gate. This change causes the floating point gate to act as an electron gun. It's the barrier between floating point gate & control gate.

Advantages:

- ✓ Easy to program.
- ✓ Rewritten many times.

Disadvantages:

- ✓ To erase data from EPROM needs UV light.

EEPROM:

- ✓ EEPROM means “Electrically Erasable Programmable Read Only Memory”.
- ✓ The entire chip does not have to completely erase to change a specific portion.

- ✓ Changing the content does not need additional requirement instead of UV light, the electron are return to the normal position by applying electric field to each cell. This erases the target cell of the EEPROM.
- ✓ EEPROM sort by hybrid between a static RAM and EPROM So that data can be changed easily.

Advantages:

- ✓ A single portion can be updated, need not change whole portion.

Disadvantage:

- ✓ At a time more than one byte can't be changed.
- ✓ It's much slower.
- ✓ After 10000 to 40000 writes the chip will be completely deadFlash Memory:
- ✓ A flash cell is based on a single transistor controlled by trapped charge, just like an EEPROM cell.
- ✓ There are some differences between flash and EEPROM. In EEPROM it is possible to read and write the contents of a single cell. In a flash device it is possible to read the contents of a single cell, but it is only possible to write an entire block of cells. Prior to writing, the previous contents of the block are erased.

Advantages:

- ✓ Flash devices have greater density, which leads to higher capacity and a lower cost per bit.
- ✓ They require a single power supply voltage and consume less power in their operation