

1/2

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 17-10-24 Name Abidya Bhandari

Dept./Lab ADLD Lab Class CD Expt./No. 1

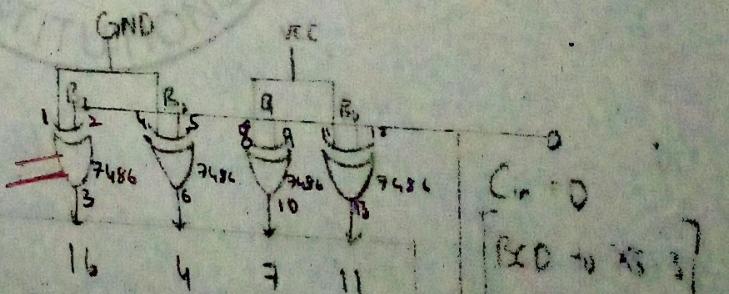
Title Realization of XS-3 code converter with parallel adder and subtractor using IC - 74283

Aim: To design XS-3 code converter using IC-74283 parallel adder and verify truth table.

Components:

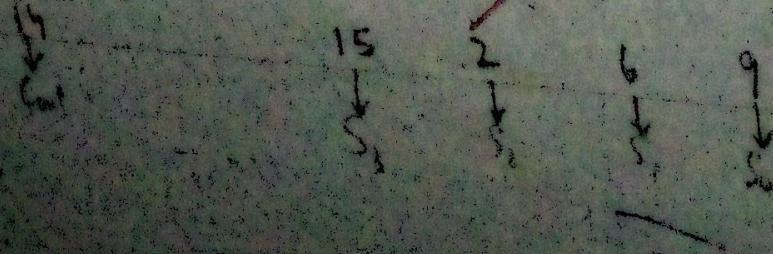
- Parallel adder - IC-74283 (+1)
- 2 input XOR - IC-7486 (+1)
- Digital trainer kit and patch cables.

Circuit diagram:



74283

13 ← [XS-3 → BCD]



Signature of
Teacher Incharge

Experiment 01

	BCD				XS-3			
	0	0	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1	1
1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	1	1	1

	XS-3				BCD			
	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	0	1	0	1	0	0	0	1
0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	1	1	0
0	1	1	0	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	0	1	1	0	0	1

Applications:

- Used in ALU
- Used to convert BCD to XS-3

R.V. COLLEGE OF ENGINEERING

3/2®

OBSERVATION / DATA SHEET

Date 8-11-24 Name Aditya Bhandari
 Dept./Lab _____ Class CD Expt./No. 1
 Title _____

K-map (BCD - Excess 3)

	CD	00	01	11	10
AB	00	1			1
	01	1		1	
	11	x	x	x	x
	10	1	x	x	

$$Z = D'$$

	CD	00	01	11	10
AB	00	1	1		
	01	1	1		
	11	x	x	x	x
	10	1		x	x

$$Y = CD + C'D'$$

	CD	00	01	11	10
AB	00	1	1	1	1
	01				
	11	x	x	x	x
	10	1	x	x	x

$$X = B'C + B'D + BC'D'$$

	CD	00	01	11	10
AB	00				
	01				
	11	x	x	x	x
	10	1	1	x	x

$$W = A + BC + BD$$

$$W = A + BC + ABCD$$

$$X = B'C + B'D + BC'D'$$

$$Y = CD + C'D'$$

$$Z = D'$$

Signature of
Teacher Incharge

Experiment 01
K-map (Express -3 - B(D))

W\X	Y\Z	00	01	11	10
00	x	x		x	
01	1			1	
11	1	x	x	x	
10	1			1	

$$D = Z'$$

W\X	Y\Z	00	01	11	10
00	x	x		x	
01		1		1	
11		x	x	x	
10		1		1	

$$C = y'z + yz'$$

W\X	Y\Z	00	01	11	10
00	x	x		x	
01		1			
11		x	x	x	
10	1	1		1	

$$B = x'y' + x'z' + xyz$$

$$A = wx + wxyz$$

$$\Rightarrow A = wx + yzw$$

$$B = x'y + x'z + xyz$$

$$C = y'z + yz'$$

$$D = z'$$

Applications:

- Used in ALU.

Truth Table:**D BCD - EXCESS-3 CODE**

BCD				EXCESS-3			
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	0	0	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

(ii) Excess-3 - BCD code

EXCESS-3				BCD			
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	1	0	1
0	1	1	1	0	1	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	1	0	0	1	0	0	1

Inference:

Realization of excess-3 to BCD using 74283 involves using its parallel adder and subtractor functionality by feeding one's complement and utilizing appropriate logic gates. The converter can accurately convert Excess-3 encoded into performing addition and subtraction operations as needed.

$$\left(\begin{matrix} 6 \\ 6 \end{matrix} \right) + \left(\begin{matrix} 3 \\ 4 \end{matrix} \right) \Rightarrow \left(\begin{matrix} a \\ 10 \end{matrix} \right)$$

Date
6/1/2023

For Sum:

C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	1	1	0	Required Output
1	1	0	0	1	
	I0a=C	I1a=C'	I2a=C'	I3a=C	Input to selected lines

For Carry:

C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	0	0	1	Required Output
1	0	1	1	1	
	I0b=0	I1b=C	I2b=C	I3b=1	Input to selected lines

Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connects VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

The full adder demonstration using multiplexer demonstrate an efficient and compact digital circuit.

Here we use 2 4:1 multiplexers and their design achieves a reduced component count and faster operation.

B. Aim: To Realize a Full Subtractor using IC74153
Components Required:

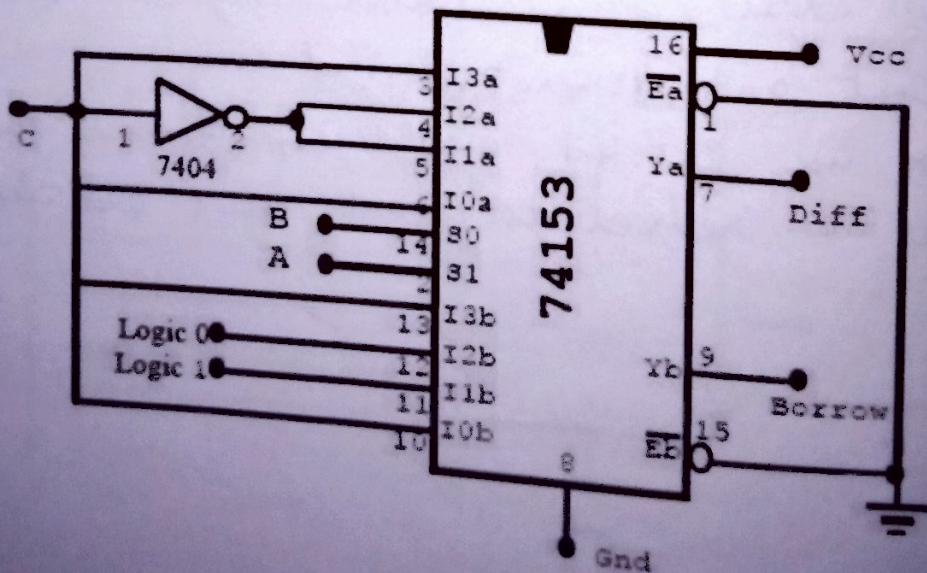
Sl. No.	Components	Specification	Quantity
1	Multiplexer	IC -74153	1 No
2	NOT GATE	IC -7404	1 No.
3	Digital trainer kit & patch cables	-	1 No, 20 patch cables

Full Subtractor using 74153

Truth table of Full subtractor:

Inputs			Outputs	
A	B	C	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Logic Diagram:



R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

1/2

Date 25-10-2024 Name Aditya Bhandari

Dept./Lab ADLD Lab Class CD Expt./No. 2

Title Realization of full adder and subtractor using IC-74153

Aim: To realize & combinational circuit full adder using IC-74153.

Components: IC - 74153 (multiplexer)
IC - 7404 (NOT gate)

Truth table (addition)

<u>A</u>	<u>B</u>	<u>C</u>	<u>Sum</u>	<u>Carry</u>
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

10/10/2024
Signature of
Teacher Incharge

Sum expression:

C
0
1

$$A=B=0$$

0
1

$$A=0, B=1$$

1
0

$$A=1, B=0$$

1
0

$$I_{0a} = C$$

$$I_{1a} = C'$$

$$I_{2a} = C'$$

Carry expression:

C
0
1

$$A=B=0$$

0
0

$$A=0, B=1$$

0
1

$$A=1, B=0$$

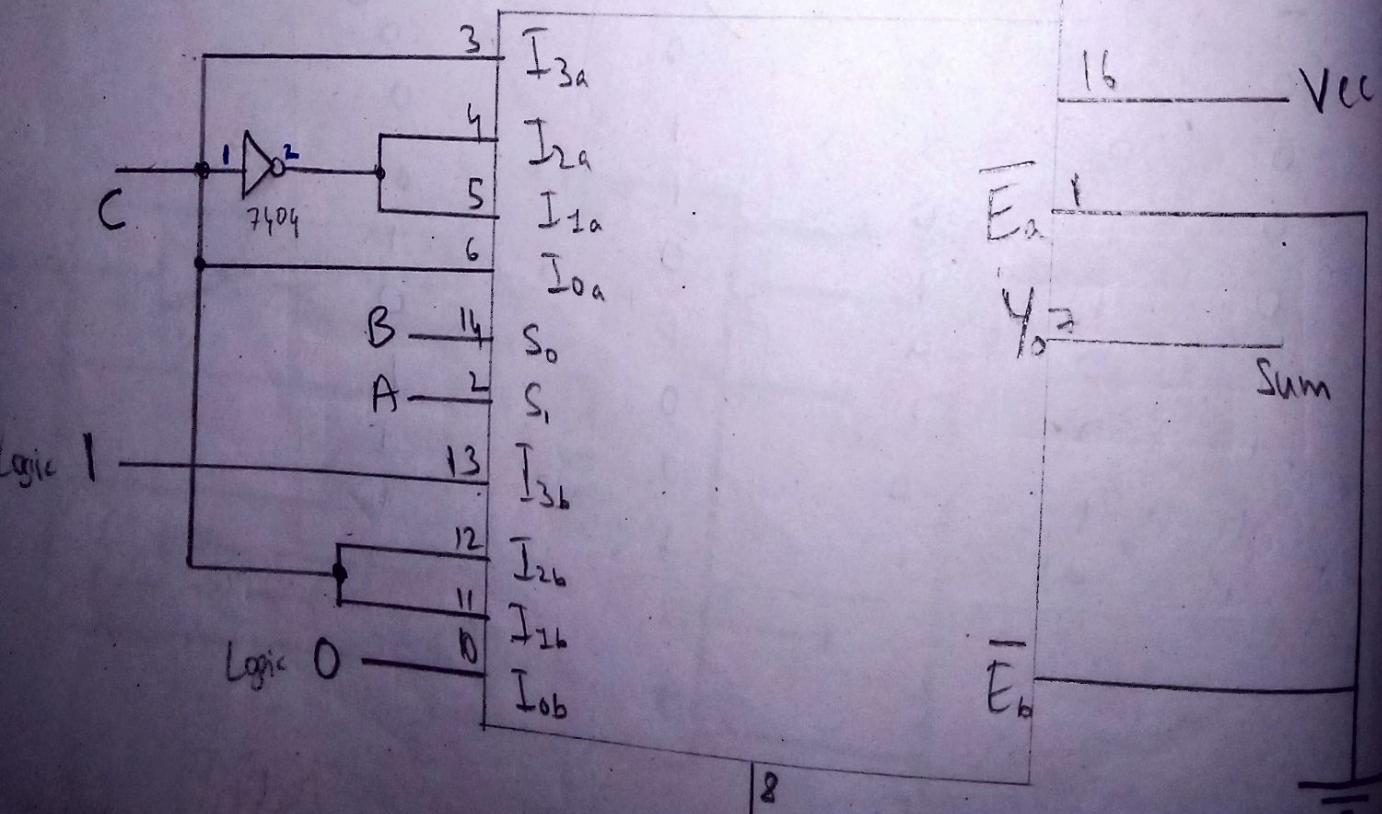
0
1

$$I_{0b} = 0$$

$$I_{1b} = C$$

$$I_{2b} = C$$

Circuit Diagram (adder)

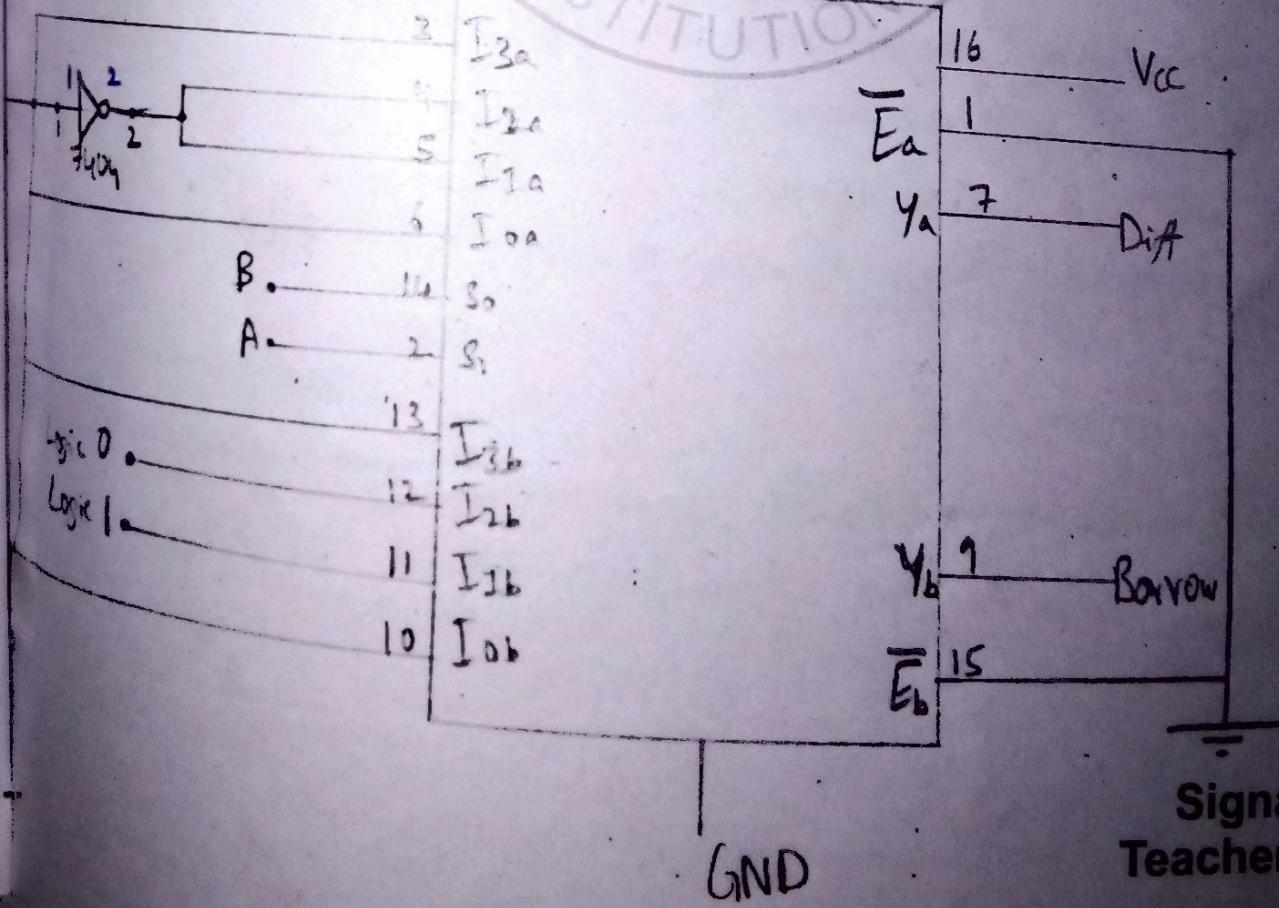


Date 15-11-24 Name Aditya Bhandari
 Dept./Lab ADLO Lab Class 10 Expt./No. 29

Title
 Truth table (subtraction) :

A	B	C	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Wheat diagram (Subtractor) :



Signature of
 Teacher Incharge

				Computer Organization Lab	
				For Difference	
C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	1	1	0	Required Output
1	1	1	0	1	
	I0b=C	I1a=C'	I2a=C'	I3a=C	Input to selected lines

For Borrow:

C	A=0, B=0	A=0, B=1	A=1, B=0	A=1, B=1	Remark
0	0	1	0	0	Required Output
1	1	1	0	1	
	I0b=C	I1b=1	I2b=0	I3b=C	Input to selected lines

Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connects VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

Assignment 3: Realize 8:1 Mux using 2:1 & 4:1 Mux, 16:1 using 8:1 & 4:1

The full adder implementation using multiplexers demonstrates an efficient & compact digital circuit. ~~design~~
Here we use 2 4:1 multiplexers & this design achieves a reduced component count and faster operation.

Applications:

- Digital signal processing, microprocessors and embedded systems with arithmetic operations.

$$\left(\frac{6}{6} \right) + \left(\frac{3}{4} \right) = \left(\frac{9}{10} \right)$$

Date: 6/1/25

Experiment 03

Design and Realization of One and Two Bit Magnitude Comparator using Basic Gates

A. Aim: To design one bit comparator, two bit comparator using logic gates

Components Required:

Sl. No.	Components	Specification	Quantity
1	AND GATE	IC -7408	1 No
2	NAND GATE	IC-7400	1 No.
3	3 Input NAND GATE	IC-7410	1 No.
4	NOT GATE	IC -7404	1 No.
5	XOR GATE	IC-7486	1 No.
6	Digital trainer kit & patch cables	-	1 No, 20 patch cables

Theory

One Bit Comparator

A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitude. The outcome of the comparison is specified by 3 binary variables that indicate whether $A = B$ or $A < B$ or $A > B$.

Truth table for one bit Comparator

Inputs		Outputs		
A	B	$A < B$	$A = B$	$A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Boolean Expression
 $(A < B) =$

Boolean Expression
 $(A = B) =$

Boolean Expression $(A > B) =$

K Map ($A < B$)

		B	
		0	1
A	0	0	1
	1	0	0

K Map ($A = B$)

		B	
		0	1
A	0	1	0
	1	0	1

K Map ($A > B$)

		B	
		0	1
A	0	0	0
	1	1	0

R.V. COLLEGE OF ENGINEERING

1/2®

OBSERVATION / DATA SHEET

Date 26-10-24Name Aditya BhandariDept./Lab ADLD Lab Class CD Expt./No. 3Title 1 & 2 bit comparator

Aim: To design 1-bit and 2-bit comparators using logic gates.

Components: AND gate - IC-7408

NAND gate - IC-7400

3-input NAND - IC-7410

NOT gate - IC-7404

XOR gate - IC-7486

Truth table :

A	B	<u>(A < B)</u>	<u>(A = B)</u>	<u>(A > B)</u>
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

K-map:

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

$A < B$

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

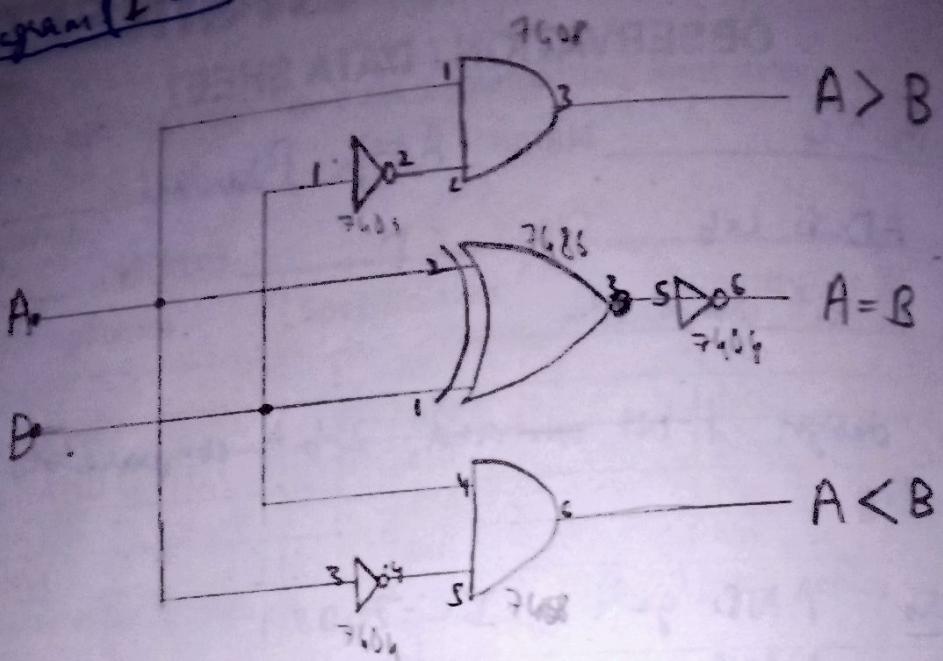
$A = B$

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

$A > B$

Signature of
Teacher Incharge

Circuit Diagram (2-bit):



2-bit Comparison:

Truth table:

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

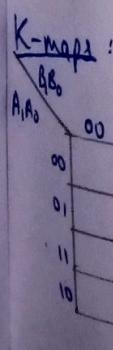
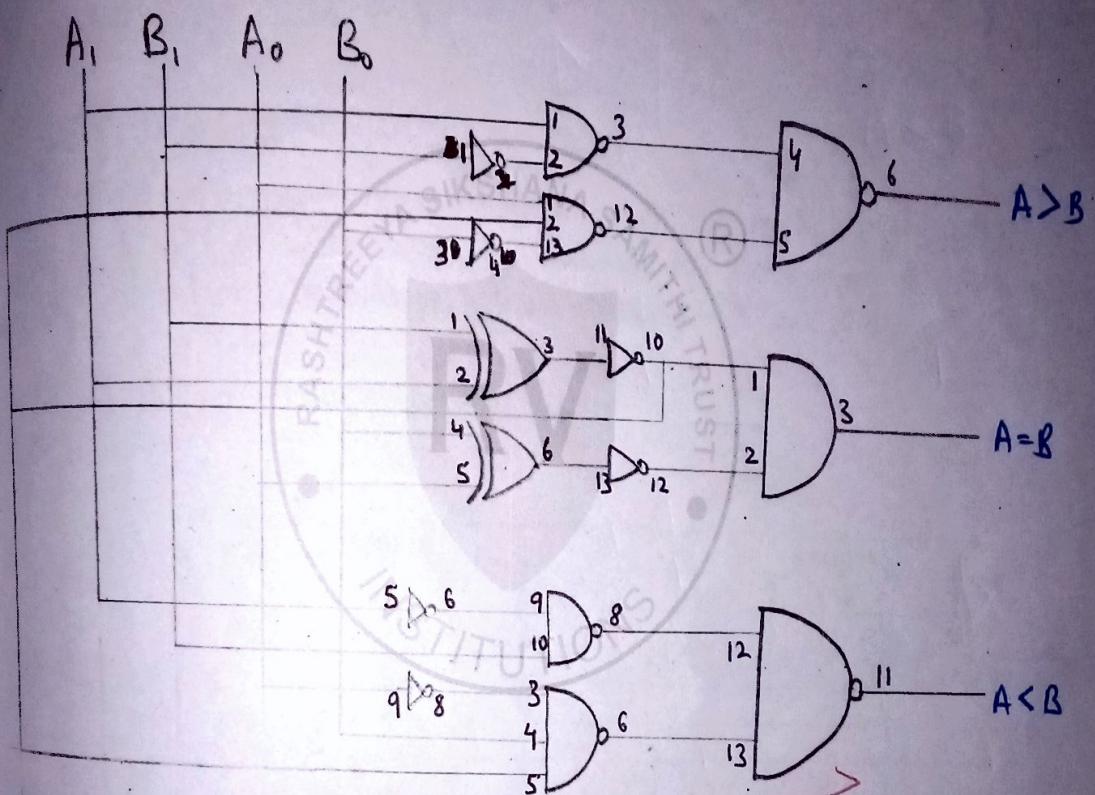
R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 8-11-24
 Dept./Lab _____
 Title _____

Name Aditya Bhandari
 Class CD Expt./No. 3

Circuit Diagram (2-bit)



	$\bar{B}_1 \bar{B}_0$	$\bar{B}_1 B_0$	$B_1 \bar{B}_0$	$B_1 B_0$
$\bar{A}_1 \bar{A}_0$	00	01	11	10
$\bar{A}_1 A_0$	00	01	11	10
$A_1 \bar{A}_0$	00	01	11	10
$A_1 A_0$	00	01	11	10

	$\bar{B}_1 \bar{B}_0$	$\bar{B}_1 B_0$	$B_1 \bar{B}_0$	$B_1 B_0$
$\bar{A}_1 \bar{A}_0$	00	01	11	10
$\bar{A}_1 A_0$	00	01	11	10
$A_1 \bar{A}_0$	00	01	11	10
$A_1 A_0$	00	01	11	10

$$\begin{aligned} \underline{A < B} \\ \bar{A}_1 B_1 + \bar{A}_0 B_1 B_0 + \bar{A}_1 \bar{A}_0 B_0 \end{aligned}$$

$$\begin{aligned} \underline{A > B} \\ A_1 \bar{B}_1 + A_0 \bar{B}_1 \bar{B}_0 + A_1 A_0 \bar{B}_0 \end{aligned}$$

$A = B$
 Signature of
 Teacher Incharge

Two Bit Comparator

Magnitude comparator is a combinational circuit that compares four digital Bits A1, A0 B1, B0 and determines their relative magnitude. The outcome of the comparison is classified by 3 binary variables that indicate whether $A = B$ or $A < B$ or $A > B$.

Truth table for Two bit Comparator

Observations:

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

Boolean

K-Map for ($A < B$)

		Expression ($A < B$)			
		00	01	11	10
		B ₁ B ₀	A ₁ A ₀		
00			1	1	1
01				1	1
11					
10				1	

$$\bar{A}_1 \bar{A}_0 B_0 + \bar{A}_0 B_1 B_0 + \bar{A}_1 B_1$$

Boolean Expression ($A = B$)K-Map for ($A = B$)

		00	01	11	10
		B ₁ B ₀	A ₁ A ₀		
00					
01					
11		1	1		1
10		1	1		1

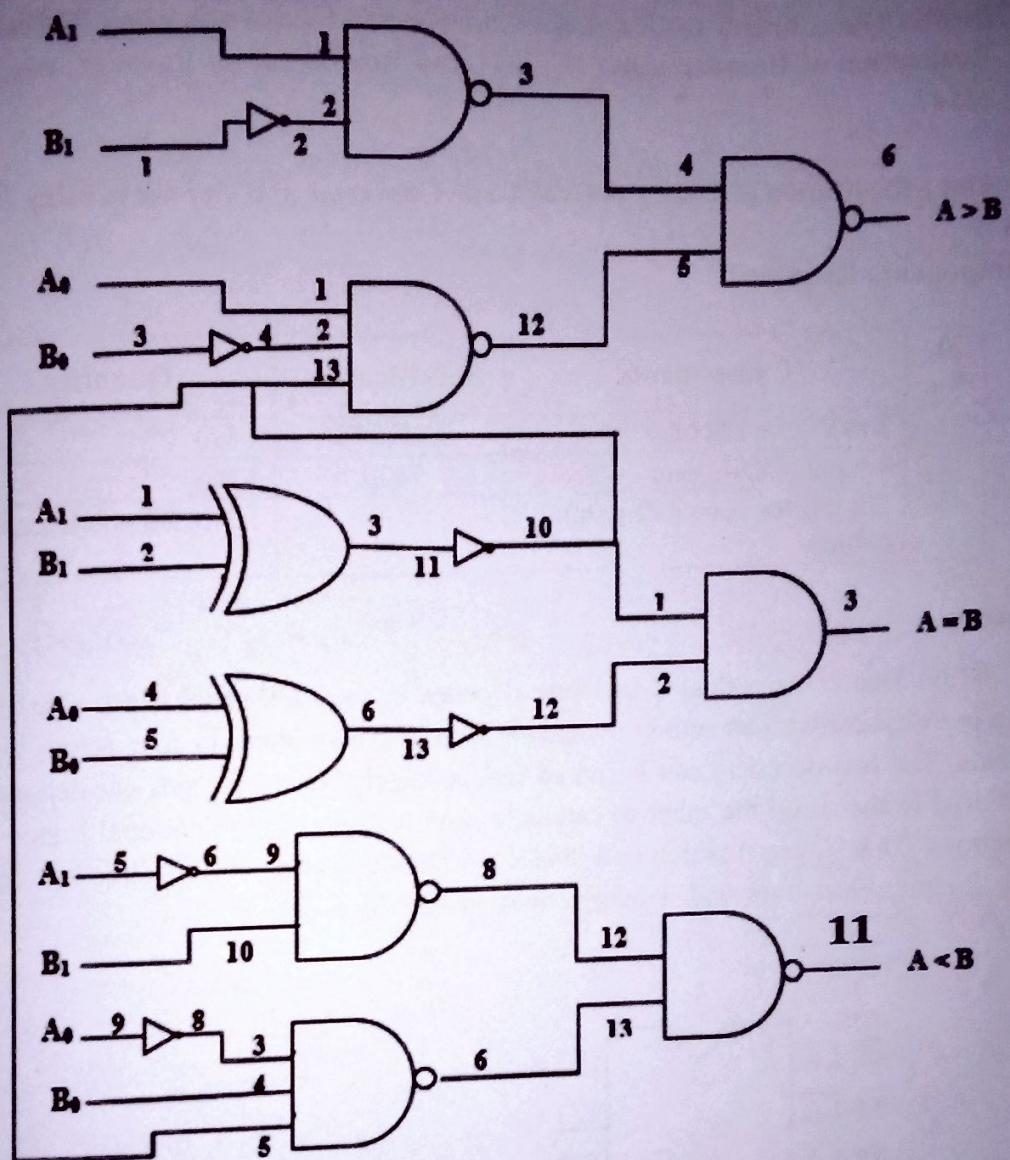
$$A_0 \bar{B}_1 \bar{B}_0 + A_1 A_0 \bar{B}_0 + A_1 \bar{B}_1$$

Boolean Expression ($A > B$)K-Map for ($A > B$)

		00	01	11	10
		B ₁ B ₀	A ₁ A ₀		
00		1			
01			1		
11				1	
10					1

$$\begin{aligned} & \bar{A}_1 \bar{A}_0 \bar{B}_1 \bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 + A_1 A_0 B_1 B_0 + A_1 \bar{A}_0 B_1 \\ &= (A_1 \odot B_1) \cdot (A_0 \odot B_0) \end{aligned}$$

Logic diagram of two-bit comparator:



Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connect VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

Comparator realization is a fundamental building block of arithmetic logical units and microprocessors

$$(G \oplus G) + (3 \oplus 10) \Rightarrow (9 \oplus 10)$$

R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

Date 27-10-24 Name Aditya Bhandari
 Dept./Lab ADLO Lab Class CD Expt./No. 4
 Title _____

Aim (A): Realization of Binary to Grey code converter using IC - 74139

Components:

- 2-4 Decoder [IC-74139]: *1
- 4 input NAND [IC-7420]: *1

Truth table:

B2	B1	B0	G2	G1	G0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

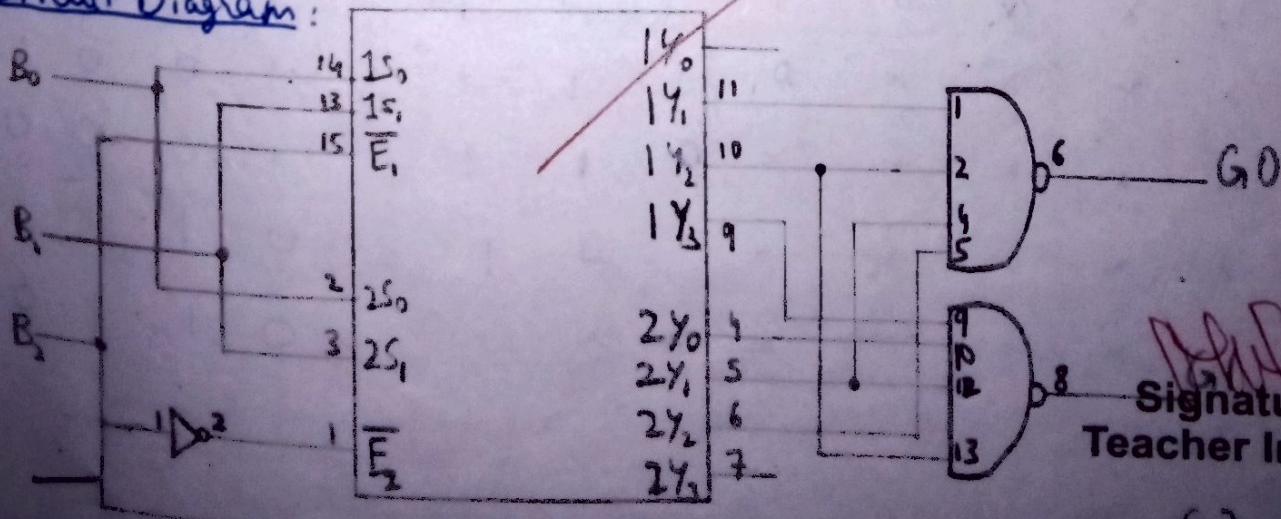
$$\Rightarrow G_0 = B_1 \oplus B_2 \\ G_1 = B_1 \oplus B_2 \\ G_2 = B_2$$

$$G_0 = \sum m(1, 2, 5, 6)$$

$$G_1 = \sum m(2, 3, 4, 5)$$

$$G_2 = \sum m(4, 5, 6, 7)$$

Circuit Diagram:



W.D. 27/11/2024
Signature of Teacher Incharge

G2

IC-7447 & Realization of

Ex

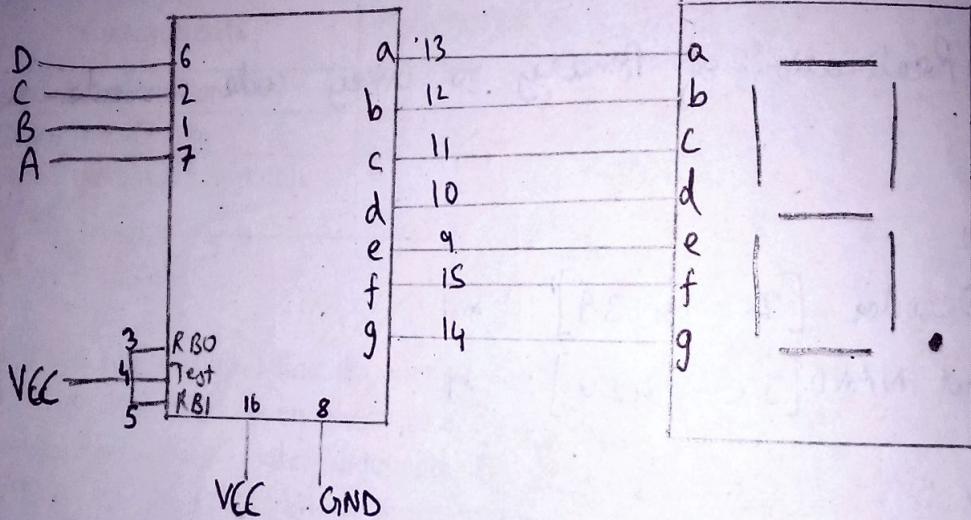
Aim (B): Realization of decoder using
using IC-7447

A.

B.

Components:

- Decoder [IC-7447] : x 1
- 7-Segment LED : x 1

A. A
7413
ComCircuit Diagram:Theor
C 74
chip is
signals
wir
actic
, 2Realization of encoder:

	1	2	3	4	5	6	7	8	9	D	C	B
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1
X	0	1	1	1	1	1	1	1	1	1	1	1
X	X	0	1	1	1	1	1	1	1	1	1	1
X	X	X	0	1	1	1	1	1	1	1	1	1
X	X	X	X	0	1	1	1	1	1	1	1	0
X	X	X	X	X	0	1	1	1	1	1	0	1
X	X	X	X	X	X	0	1	1	1	1	0	0
X	X	X	X	X	X	X	Q	1	1	1	0	0
X	X	X	X	X	X	X	X	0	1	1	0	1
X	X	X	X	X	X	X	X	X	0	1	0	1
X	X	X	X	X	X	X	X	X	0	0	1	1

Applied Digital Logic Design & Computer Organization

R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

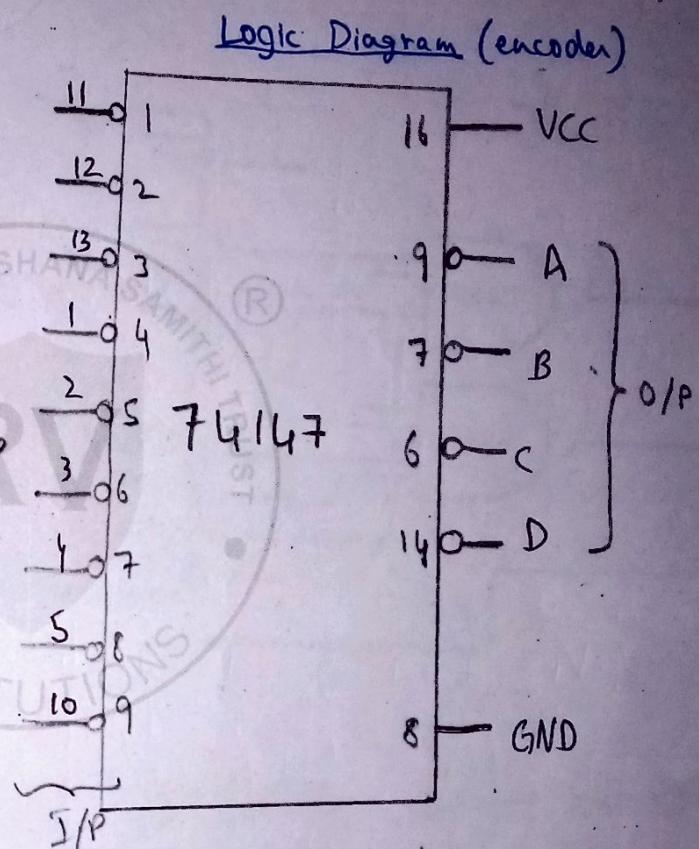
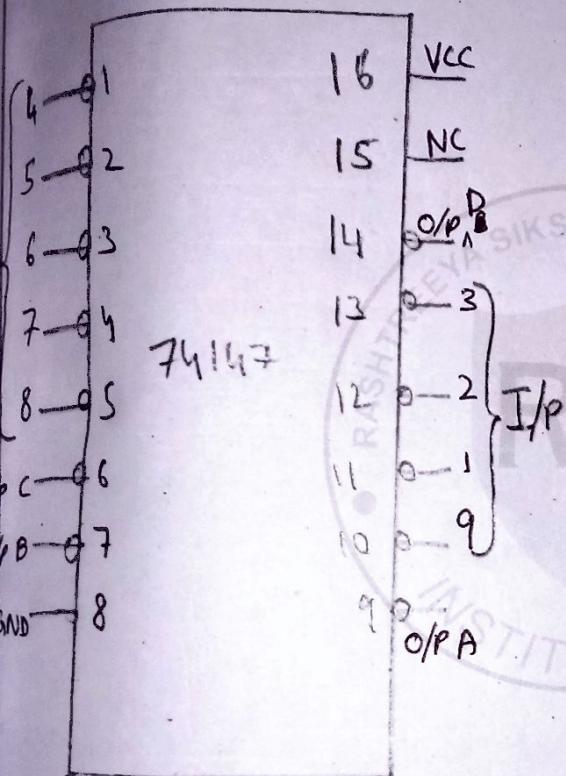
Date 22-11-29

Dept./Lab

Name Aditya Bhandari

Class CD Expt./No.

Title Pin Diagram (encoder)

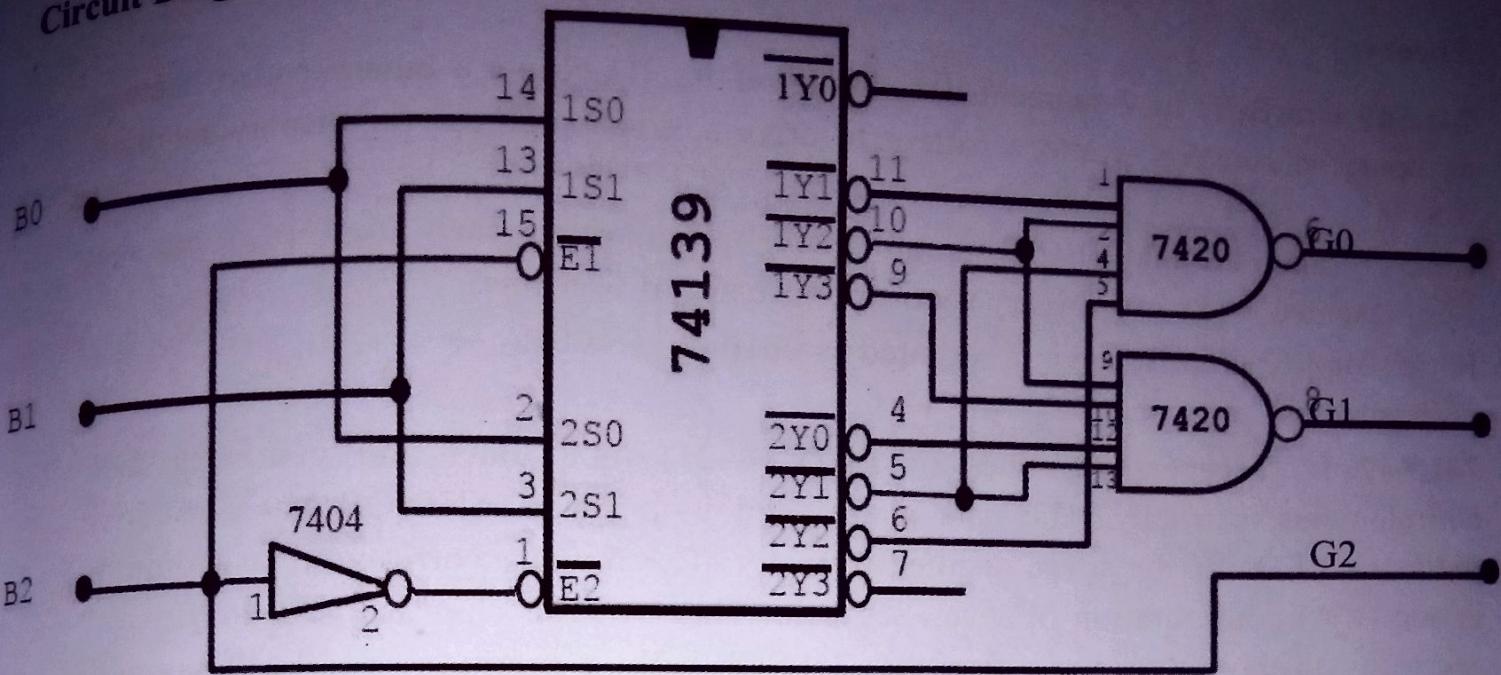


Operation of decoder:

Decoder Inputs			Decoder Outputs							Display Output	
B	A		a	b	c	d	e	f	g		
0	0		1	1	1	1	1	1	0	0	0
0	0	1	0	1	1	0	0	0	0	1	1
0	1	0	1	1	0	1	1	0	1	2	2
0	1	1	1	1	1	1	0	0	1	3	3
1	0	0	0	1	1	0	0	1	1	4	4
1	0	1	1	0	1	0	0	1	1	5	5
1	1	0	1	0	1	1	0	1	1	6	6
1	1	1	1	0	1	1	1	0	0	7	7
0	0	0	1	1	1	1	1	1	1	8	8
0	0	1	1	1	1	1	0	1	1	9	9

Signature of
Teacher Incharge

Circuit Diagram:



Truth Table: 3 Bit binary code conversion using IC74139

Inputs			Outputs		
B2	B1	B0	G2	G1	G0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

$$G_0 = B_1 \oplus B_0$$

$$G_1 = B_2 \oplus B_1$$

$$G_2 = B_2$$

$$G_0 = \sum m(1, 2, 5, 6)$$

$$G_1 = \sum m(2, 3, 4, 5)$$

$$G_2 = \sum m(4, 5, 6, 7)$$

Inference:

Procedure:

1. Verified IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connects VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

BCD to gray code is straight-forward and can be implemented using basic logic gates.

Decoder using 8-segment display provides a simpler and efficient way to display decimal numbers. Decoder takes a 4-bit binary input and converts it to decimal code that drives an 8-segment ~~display~~ display.

Truth Table:

Active - Low Decimal Inputs										Active Low BCD Output			
1	2	3	4	5	6	7	8	9		D	C	B	A
1	1	1	1	1	1	1	1	1		1	1	1	0
0	1	1	1	1	1	1	1	1		1	1	0	1
X	0	1	1	1	1	1	1	1		1	1	0	0
X	X	0	1	1	1	1	1	1		1	0	1	1
X	X	X	0	1	1	1	1	1		1	0	1	0
X	X	X	X	0	1	1	1	1		1	0	0	1
X	X	X	X	X	0	1	1	1		1	0	0	0
X	X	X	X	X	X	0	1	1		0	1	1	1
X	X	X	X	X	X	X	0	1		0	1	1	0
X	X	X	X	X	X	X	X	0		0	1	0	1

Procedure:

1. Verify IC Number before conducting experiment
2. The IC is fixed into IC socket of the trainer kit and connect VCC and GND.
3. Connections are made as shown in logic diagram.
4. Connect all the inputs to switches and out to the LED's
5. The inputs are applied using the toggle switches from the trainer kit according to the truth table and the corresponding outputs are verified on the LEDs.
6. Truth table is verified for different combination of input.

Inference:

IC 74147 has a specified design & provides a prebuilt priority encoder and is designed for high-speed operation, making it suitable for applications that require fast encoding.

Applications:

- Encoders are used in digital systems like computers and calculate encoded data.

$$\begin{array}{c}
 \textcircled{6} \times \textcircled{3} \Rightarrow \textcircled{10} \\
 \text{Pulse}
 \end{array}$$

R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

Date 28-10-24 Name Aditya Bhandari
 Dept./Lab AOLD Lab Class CD Expt./No. 5

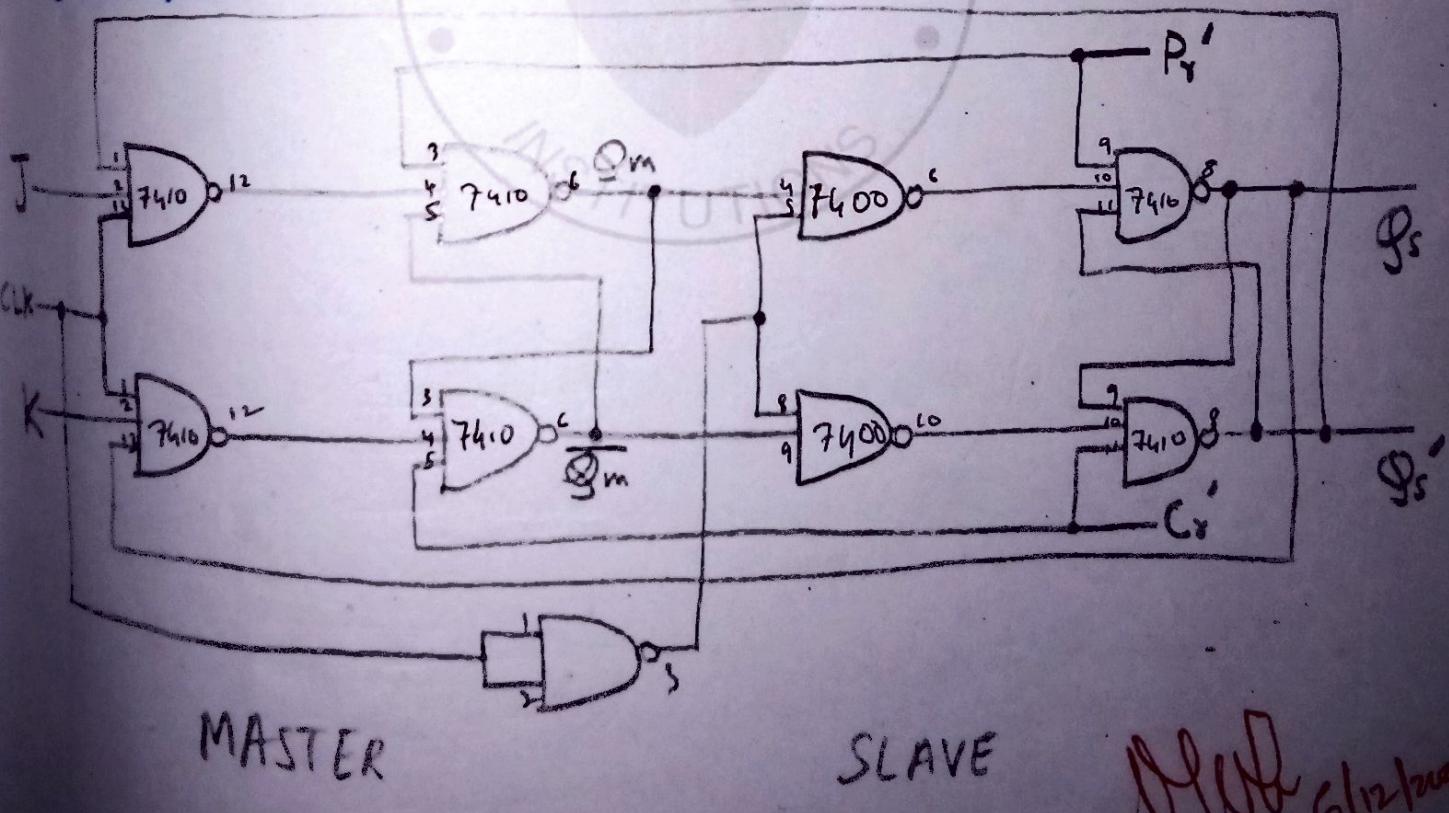
Title Realization of Master-Slave JK flip flop

Aim: To realize Master-Slave JK flipflops using NAND gates and IC-7476

Components:

- Flipflop [IC-7476]
- NAND Gate [IC-7400]
- 3-Input NAND [IC-7410]

Logic Diagram:



MUD 6/12/2024
 Signature of
 Teacher Incharge

is 0, the output of the inverter is 1. Since the clock is enabled and because $CP = 1$, S inputs is 1 as long as the returns to 0, affecting it. Then

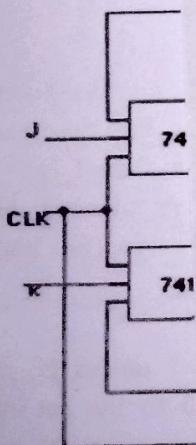
Truth Table:

<u>CLK</u>	I	K	<u>Q_{n+1}</u> (next)
H	0	0	Q
H	0	1	0
H	1	0	1
H	1	1	\bar{Q}

Procedure:

1. Rig up the circuit
2. Keep both power supplies connected
3. Verify the truth table

Logic Diagram



Applications:

- facilitates synchronized sequential operations
- Used for pulse counting and frequency division
- Precise frequency outputs digitally
- Enables serial data storage and manipulation

Truth Table:

CLK	J	K	Q_{n+1} (next state)
H	0	0	?
H	0	1	?
H	1	0	?
H	1	1	?

Inference:

Assignment: Realize D and T flip-flops using SR/JK flip-flops.

M.S JK flip flop is designed using NAND gates and is a sequential circuit where 2 stages ~~are~~ are connected to synchronize state transitions. The M is enabled during one clock cycle, while S operates during the opposite phase. This prevents race condition, ensuring reliable operation.

$$\begin{array}{r} 6/6 \\ \times 3/1 \\ \hline 10/a \end{array}$$

MD

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 28-10-24 Name Aditya Bhendari
Dept/Lab ADLD Lab Class CD Expt./No. 6.A.

Title Realization of Up & Down programmable counter.

Aim: To realize the counters using IC-74192 & IC-74193 :

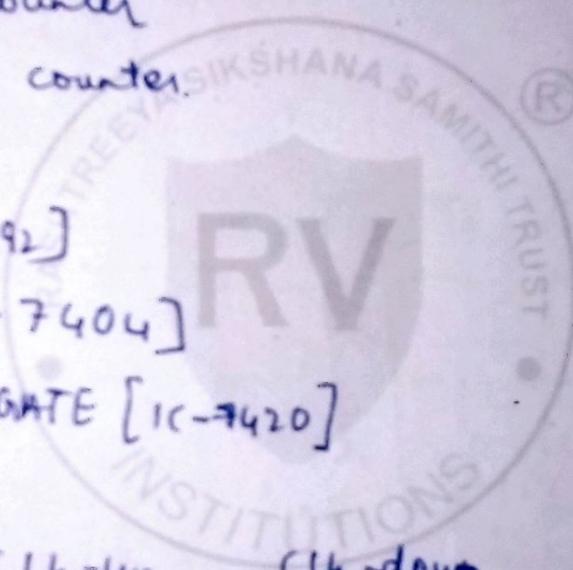
- Up/Down counter
- Presettable up counter
- Presettable down counter

Components:

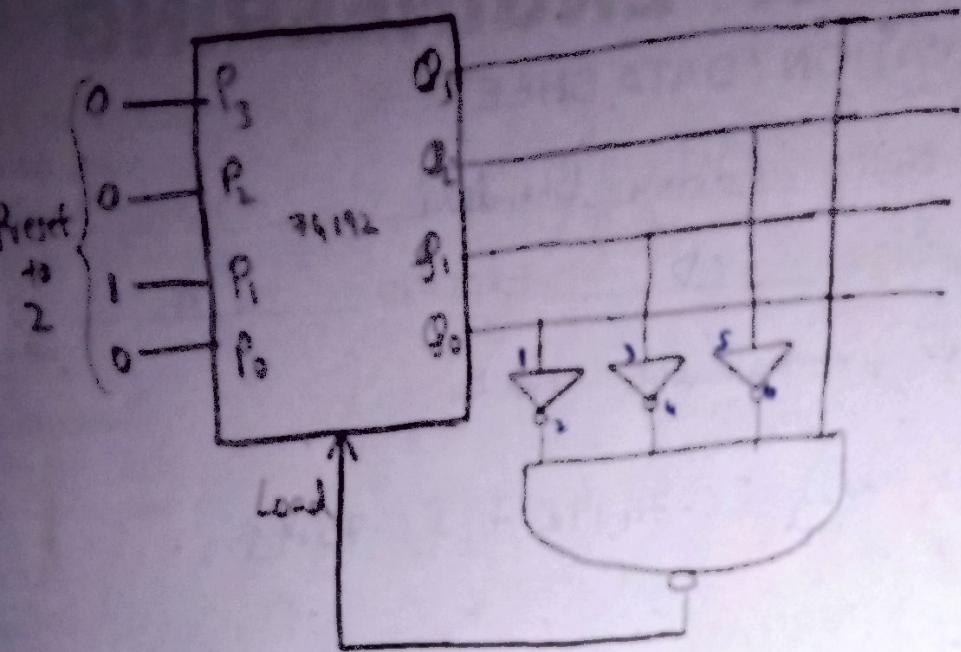
- Counter [IC-74192]
- NOT gate [IC-7404]
- 4-Input NAND GATE [IC-7420]

Truth Table:

<u>Load</u>	<u>Clear</u>	<u>Clk-up</u>	<u>Clk-down</u>	<u>Mode</u>
X	1	X	X	Preset to zero
1	0	↑	1	Up-count
1	0	1	↑	Down-count
0	0	X	X	Preset
1	0	1	1	Stop count


Mulu
Signature of
Teacher Incharge

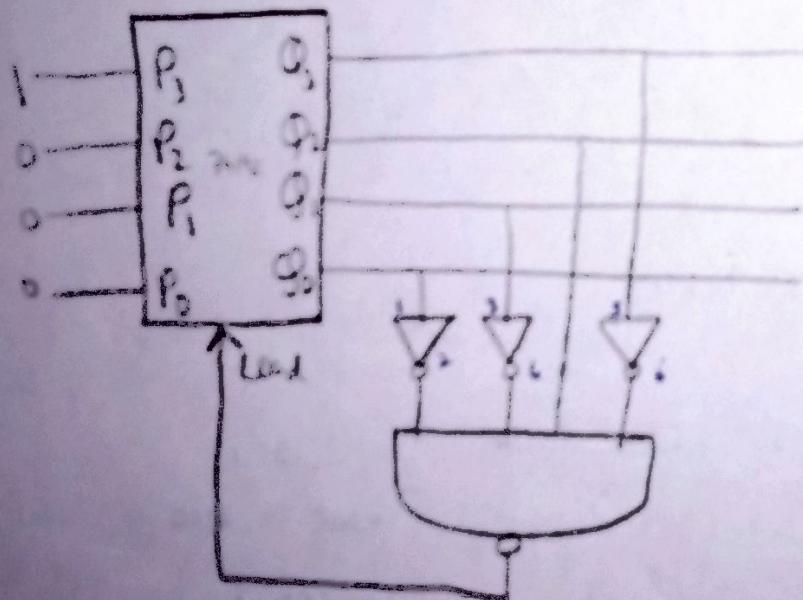
Mod 6 up counter:



Q_3	Q_2	Q_1
0	0	0
0	0	1
0	1	0
0	1	1
0	1	0
0	1	1

Mod 4 down counter:

Q_3	Q_2	Q_1
1	0	0
0	1	1
0	1	0
0	1	1



Applications:

- Divides clock frequencies for timing circuits
- Tracks occurrences in industrial processes.
- Used for signal processing tasks.
- Enables memory addresses in processors.

2/2

V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

Date 29-10-24 Name Aditya Bhanderi

Dept./Lab ADLD lab Class CD Expt./No. 6.B.

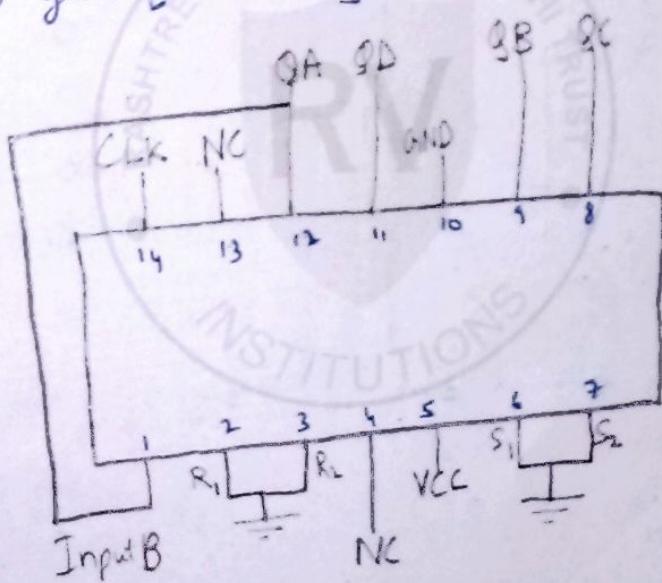
Title Realizing decade counter using IC - 7490

Ques: To design and realize a decade counter using IC-7490

Components:

- BCD counter [IC-7490]
- 3 input NAND gate [IC-7410]

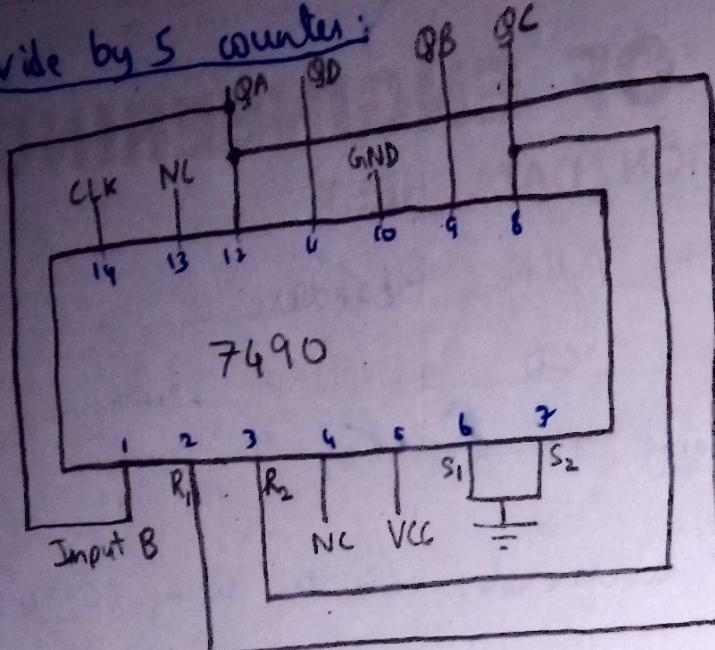
Circuit Diagram:



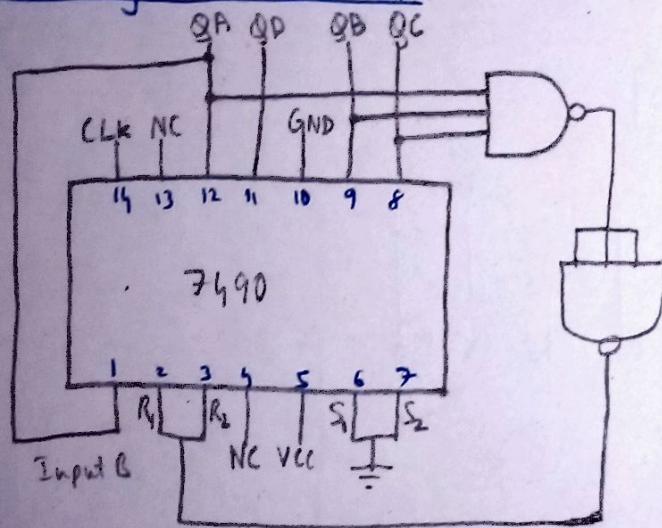
Truth table:

$\frac{Q_D}{0}$	$\frac{Q_C}{0}$	$\frac{Q_B}{0}$	$\frac{Q_A}{0}$
-----------------	-----------------	-----------------	-----------------

Divide by 5 counter:



Divide by 7 counter:



Applications:

- Frequency division
- Digital control systems
- Security systems
- Testing & measuring equipments.

- mode
- 3) In up - counting mode carry goes low when $Q_3 Q_2 Q_1 Q_0$ changes from 1111 to 0000
- 4) In down - counting mode, borrow goes low when $Q_3 Q_2 Q_1 Q_0$ changes from 0000 to 1111

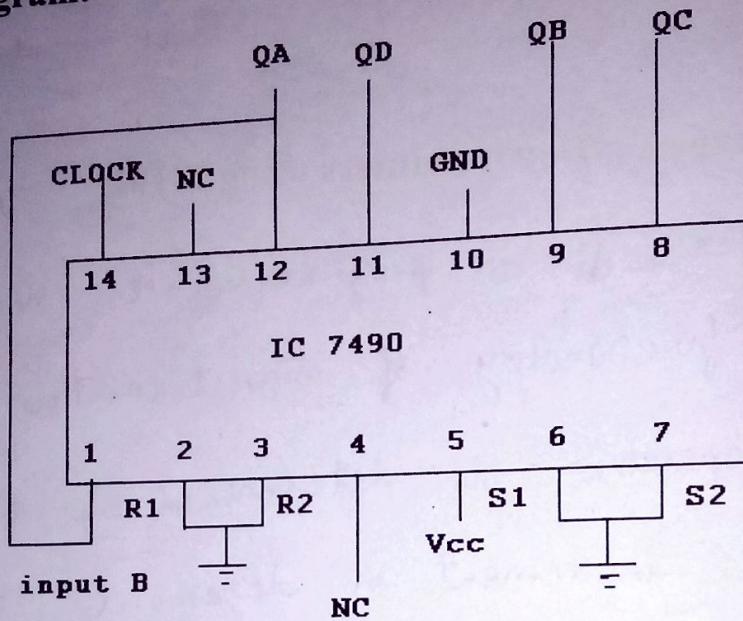
Inference:

Assignment: Design of Multi-stage Up/Down counters using 74192

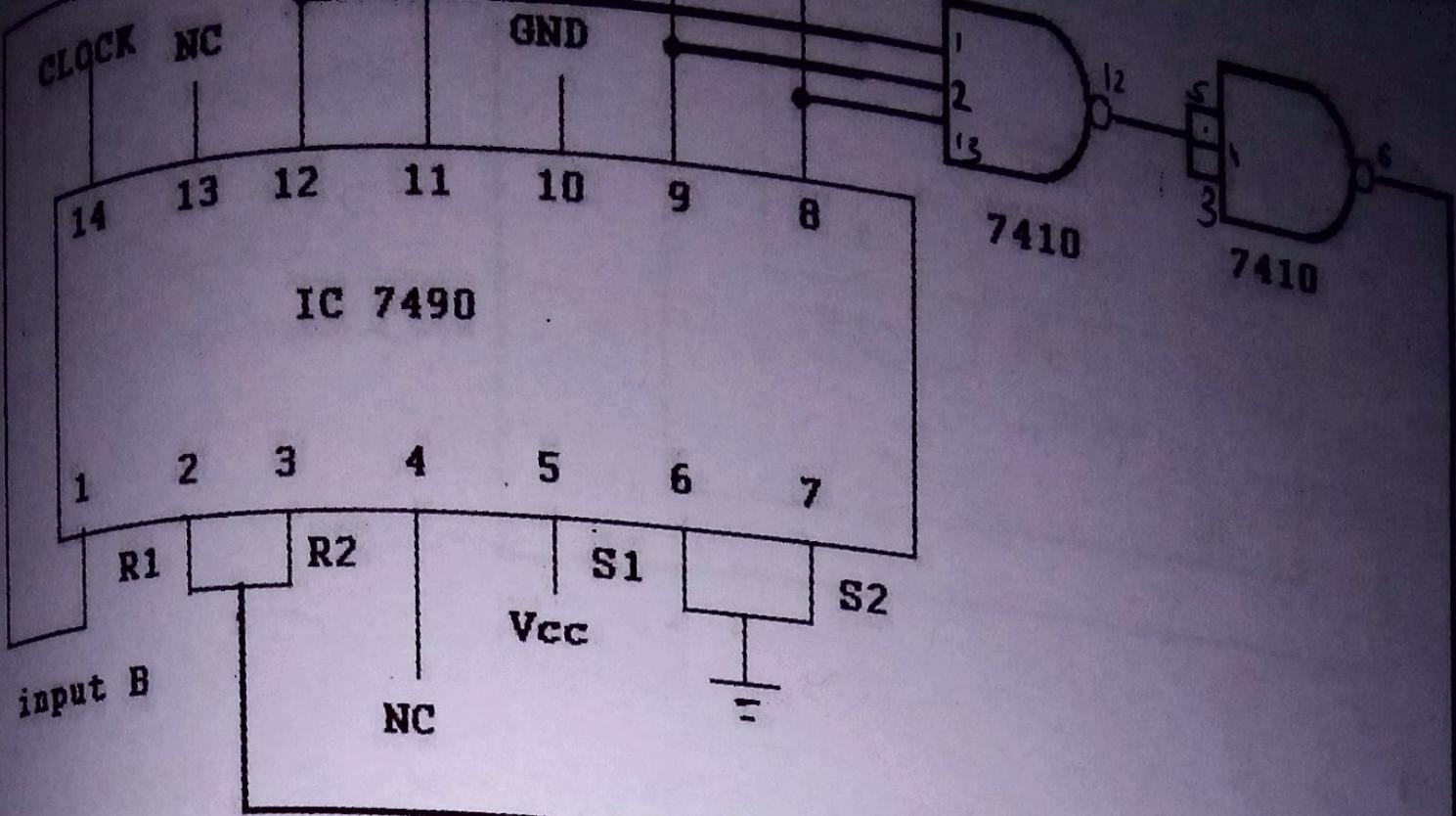
We designed and implemented a programmable up/down counter, to demonstrate the functionality of digital counters in sequential circuits. The counter's operation is determined by a control input, allowing it to either increment or decrement its value based on the input signal.

Procedure:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

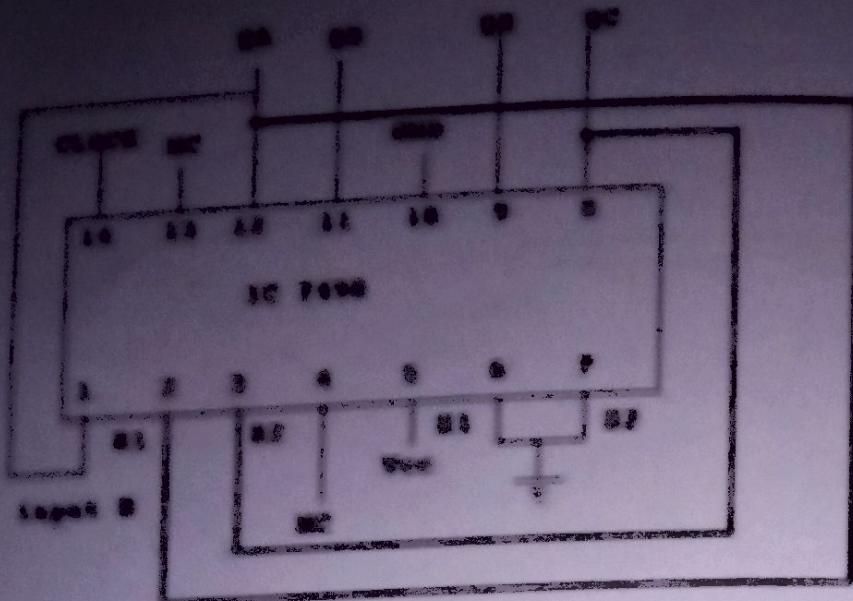
Circuit Diagram:**Truth Table:**

Q_D	Q_C	Q_B	Q_A
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



Truth Table:

Q_D	Q_C	Q_B	Q_A
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	0	0	0



French Teacher

Q_2	Q_1	Q_3	Q_4
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	0	0	0

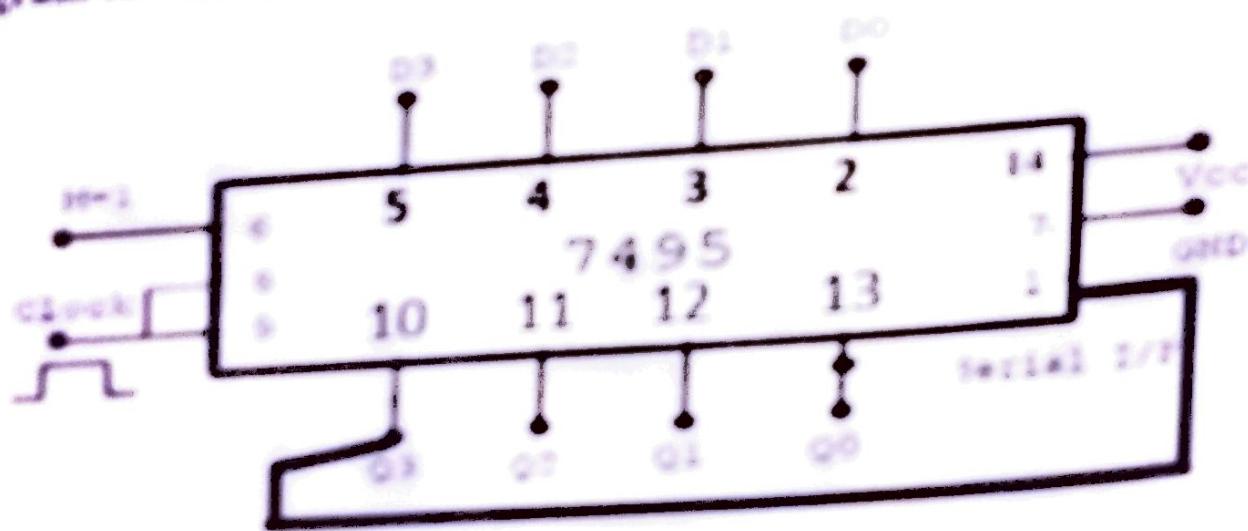
LaTeX with Beamer

We designed & implemented mod-10, mod-7, and mod-5 counters to demonstrate the operation of counters with different modulus values. Each counter was configured to reset after reaching its specified modulus, ensuring all counts accurately.

Truth Table:

Mode M	Clock	Inputs				Output			
		D3	D2	D1	D0	Q3	Q2	Q1	Q0
1	0	1	0	0	0	0	0	0	0
0	1					1	0	0	0
0	2					0	1	0	0
0	3					0	0	1	0
0	4					0	0	0	1
0	5					1	0	0	0

Logic Diagram for Ring Counter:



Procedures for Ring counter:

1. Connection is made as shown in the logic diagram.
2. 4 bit Data is loaded into shift registers **D3, D2, D1 & D0**.
3. Keeping Mode control M=1, clock pulse are applied, the data present in **D3, D2, D1 & D0** will appear at **Q3, Q2, Q1 & Q0 respectively**. Set 1000 at the outputs.
4. Connect the Q3 to Serial i/p, (Remove if any connection to Serial I/P for giving serial data). Make M =0, and keep applying clock pulses to realize ring counter

R.V. COLLEGE OF ENGINEERING®

1/2

OBSERVATION / DATA SHEET

Date 29-10-24 Name Aditya Bhandari
Dept./Lab EAOLD Let Class CD Expt./No. 7A.

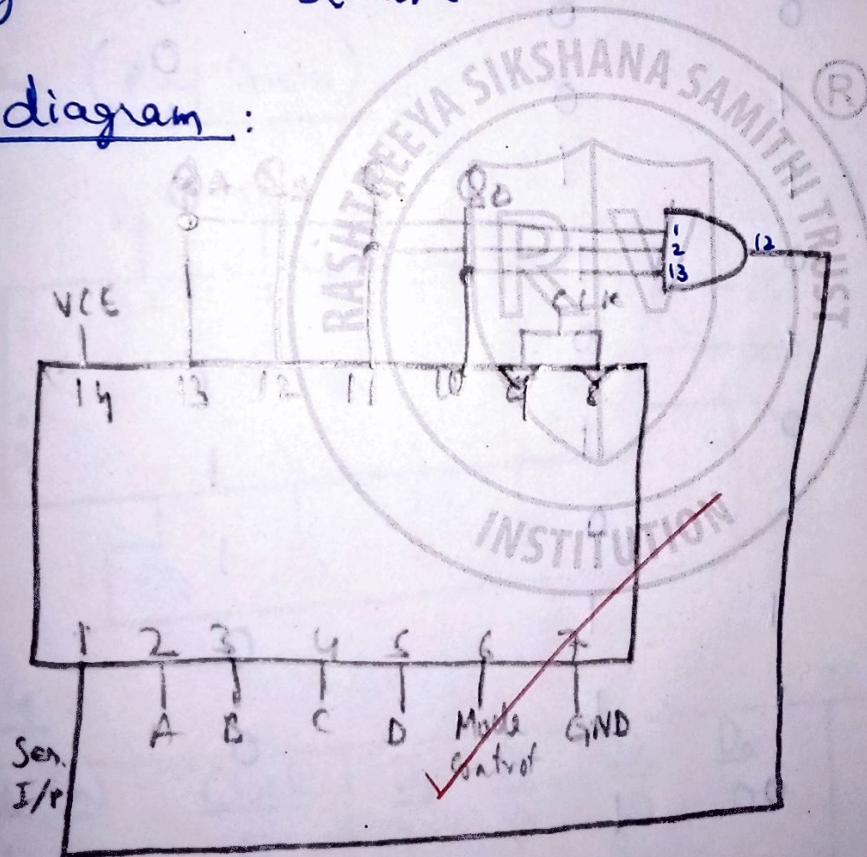
Title Sequence generator using IC - 7495

Components :

Shift register: IC-7495

XOR gate: IC7486

Logic of diagram :



Signature of
Teacher Incharge

Truth table

Φ_A	Φ_B	Φ_C	Φ_D	Φ_E	Φ_F	$y = \Phi_C \oplus \Phi_F$
1	1	1	1	1	0	0
0	1	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	1	1	1
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	1	1
0	0	0	1	1	1	1
1	0	0	1	1	1	1
1	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	1	0	1	0	0
0	1	0	1	1	1	1
1	0	1	0	0	1	1
1	1	0	0	1	1	1
1	1	1	0	0	0	1
0	1	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	1	1	1

V.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 29-10-24 Name Aditya Bhandari
 pt./Lab ADLD Lab Class CD Expt./No. 7.B.
 e sequence Ring and Johnson counters.

4 bit shift register [IC - 7495]

NOT GATE [IC - 7404]

diagram (Ring Counter)

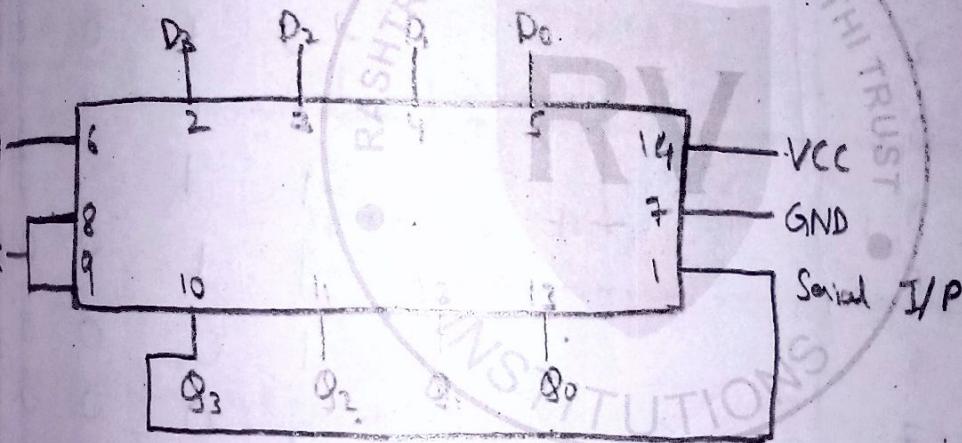
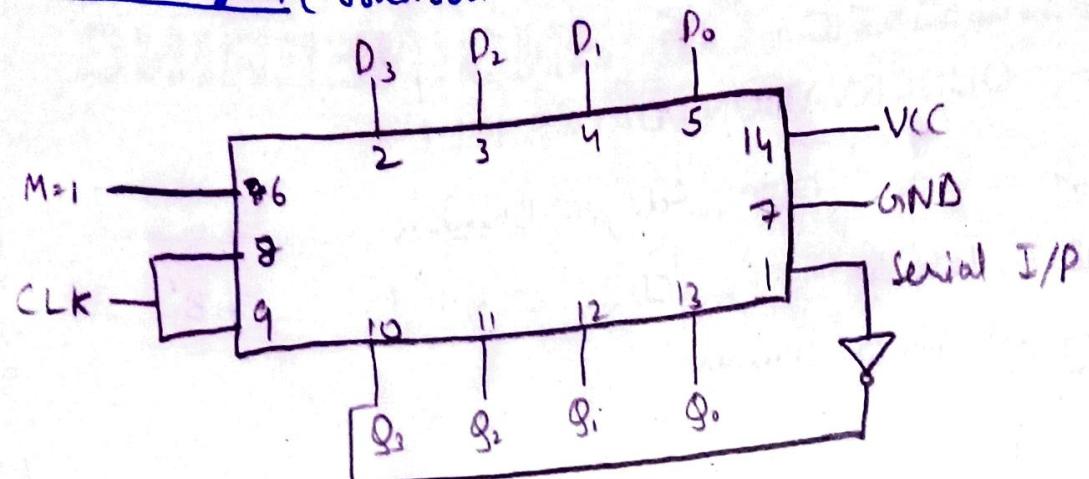


table :

Mode (m)	Clock	D ₃	D ₂	D ₁	D ₀	Q ₃	Q ₂	Q ₁	Q ₀
1	0	1	0	0	0	0	0	0	0
0	1					1	0	0	0
0	2					0	1	0	0
0	3					0	0	1	0
0	4					0	0	0	1
0	5					1			

Signature of
Teacher Incharge

Logic diagram (Johnson counter):



Truth table:

Mode (m)	Clock	D ₃	D ₂	D ₁	D ₀	Q ₃	Q ₂	Q ₁	Q ₀
1	0	0	0	0	0	0	0	0	0
0	1					0	0	1	
0	2					0	1	0	
0	3					1	1	1	
0	4					1	1	1	0
0	5					1	1	0	0
0	6					1	0	0	0
0	7					0	0	0	0
0	8					0	0	0	1

Applications:

Ring:

- Used as shift registers for serial data
- Used in synchronous communication protocols
- Microcontrollers & processors

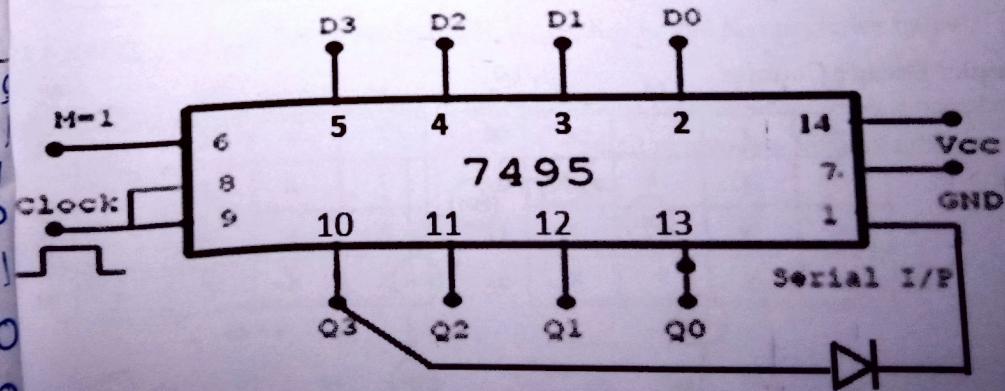
Johnson:

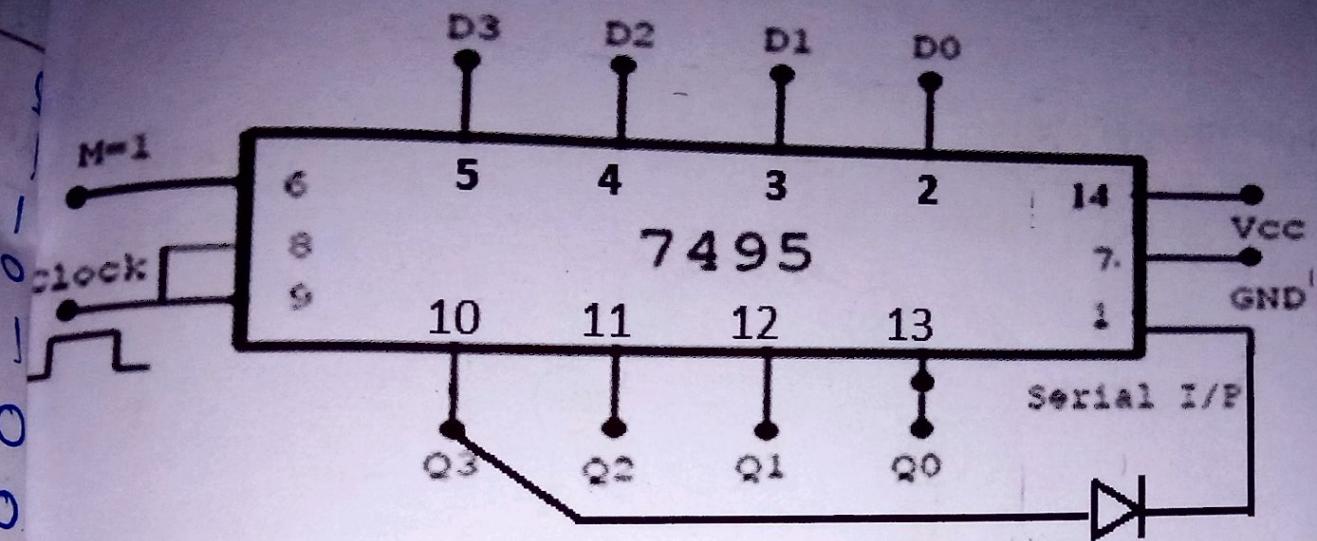
- Divides the clock frequencies
- Addressing memory locations
- Sensor data acquisition

Truth Table:

Mode M	Clock	Inputs				Output			
		D3	D2	D1	D0	Q3	Q2	Q1	Q0
0		0	0	0	0	0	0	0	1
1						0	0	1	1
2						0	1	1	1
3						1	1	1	1
4						1	1	1	0
5						1	0	0	0
6						0	0	0	0
7						0	0	0	0
8						0	0	0	1

Logic Diagram for Johnson Counter





Procedure for Johnson Counter:

Connection is made as shown in the logic diagram.

4 bit Data is loaded into shift registers **D3, D2, D1 & D0**.

Keeping Mode control **M=1**, clock pulse are applied, the data present in **D3, D2, D1 & D0** will appear at **Q3, Q2, Q1 & Q0** respectively. Set **0000** as output.

Connect the output **Q3** to Serial input through a NOT GATE. Then make **M=0**, and keep applying clock pulses to realize Johnson counter.

We successfully realized a sequence generator using
IC- 7495 and a Johnson counter using the IC-
7495. Realization of ring counter has also been
achieved through this.

$$\begin{array}{c} \text{Diagram showing two circles with numbers 6 and 3, followed by a plus sign and a circle with 10.} \\ 6 + 3 \Rightarrow 10 \\ \text{Below the diagram, handwritten text reads:} \\ \text{Date: 6/11/2015} \end{array}$$

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 29-10-2024 Name Aditya Bhandari

Dept./Lab ADLD lab Class CD Expt./No. 8

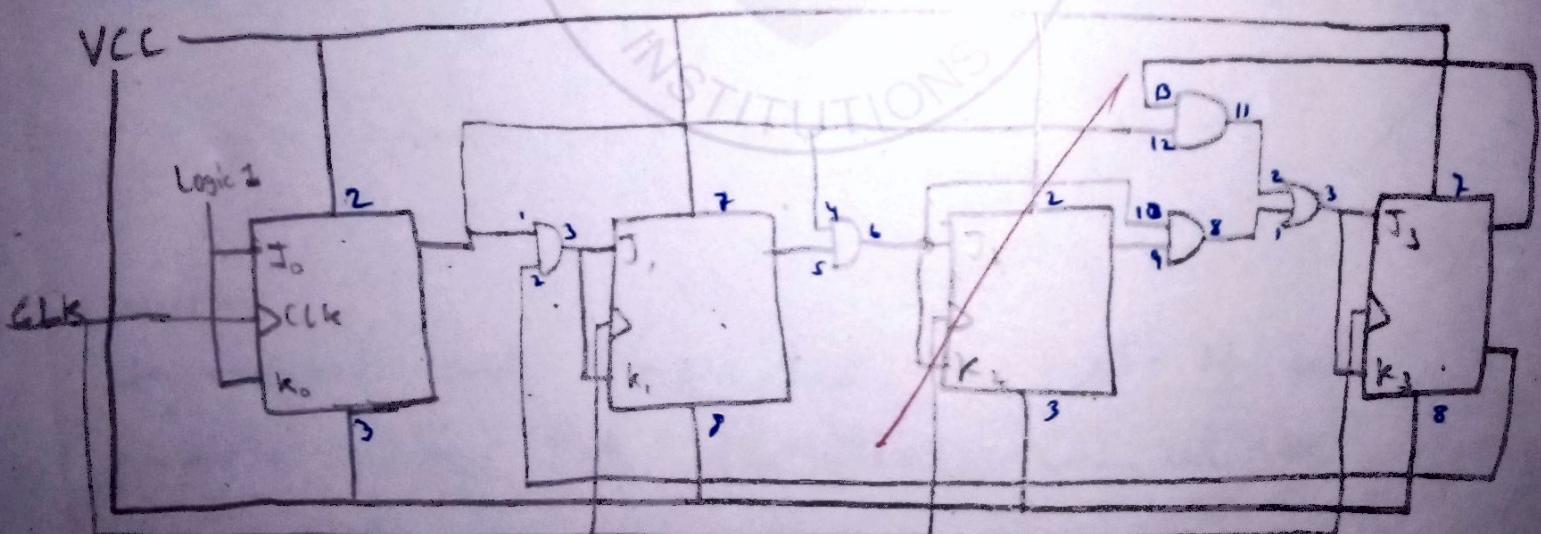
Title Mod-N synchronous up-counter.

Aim: To design and test Mod-N(5, 6, 9, 10) synchronous up-counter using J-K flip-flop.

Components:

- AND gate (IC-7408)
- OR gate (IC-7432)
- JK flipflop (IC- 7476)

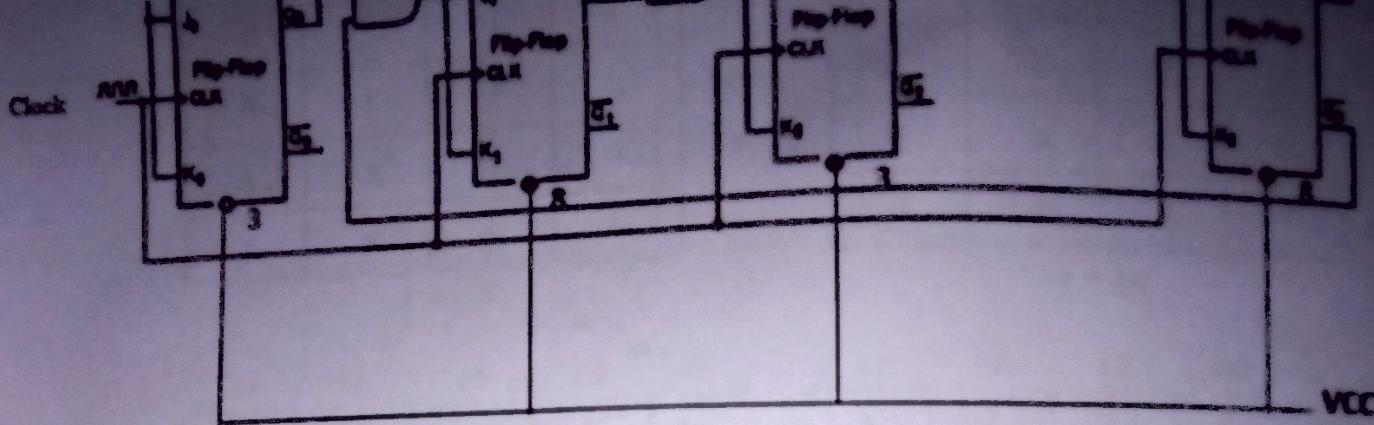
Logic diagram:



Signature of
Teacher Incharge

Truth table :

<u>Present state</u>				<u>Next state</u>					<u>Output</u>				
<u>Q_3</u>	<u>Q_2</u>	<u>Q_1</u>	<u>Q_0</u>	<u>Q_3</u>	<u>Q_2</u>	<u>Q_1</u>	<u>Q_0</u>	<u>J_3</u>	<u>K_3</u>	<u>J_2</u>	<u>K_2</u>	<u>J_1</u>	<u>K_1</u>
0	0	0	0	0	0	0	1	0	x	0	x	0	x
0	0	0	1	0	0	1	0	0	x	0	x	1	x
0	0	1	0	0	0	1	1	0	x	0	x	x	0
0	0	1	1	0	1	0	0	0	x	1	x	x	1
0	1	0	0	0	1	0	1	0	x	x	0	0x	
0	1	0	1	0	1	1	0	0	x	x	0	1x	
0	1	1	0	0	1	1	1	0	x	x	0	x	0
0	1	1	1	0	0	0	0	1	x	x	1	x	1
1	0	0	0	1	0	0	1	x	0	0x	0	x	
1	0	0	1	0	0	0	0	x	1	0x	0	x	



Inference:

We designed and implemented a mod- N synchronous counter using J-K flip-flops and utilized AND and OR gates for this. The counter counts from 0 to $N-1$ and then back to 0. The clock signals all flip-flops act in unison.

X X X
6/6 3/4 10/10
JWB 6/1/2025

PART B - Innovative Experiments (IE) / Open Ended Experiments

Design a 4/8-bit CPU using the LOGISIM simulator, for the following specifications.

- 1) Program Counter (Assume 256 locations of program/code memory).
- 2) Instruction Register (Assume instruction size as 16 bit)
- 3) General Purpose Registers (RISC type-R0-R7): Use Harvard & Multiple Bus
Architecture
- 4) ALU (to support 4-bit integer arithmetic operations & 4-bit logical operations)
- 5) Memory – 1024 locations of ROM (to store instructions of size 16 bit) and 256 RAM (to store 4-bit data)
- 6) Implement the following instructions namely: MOV, ADD, SUB, LOAD, STORE, AND, XOR, NOT, BRANCH, BRANCH ON CONDITION.
- 7) Result to be displayed on 7-segment displays / reg tab of LOGISIM

1. Title:

3-bus CPU architecture (4-bit)

2. Aim:

To simulate a 4-bit CPU architecture on Logisim

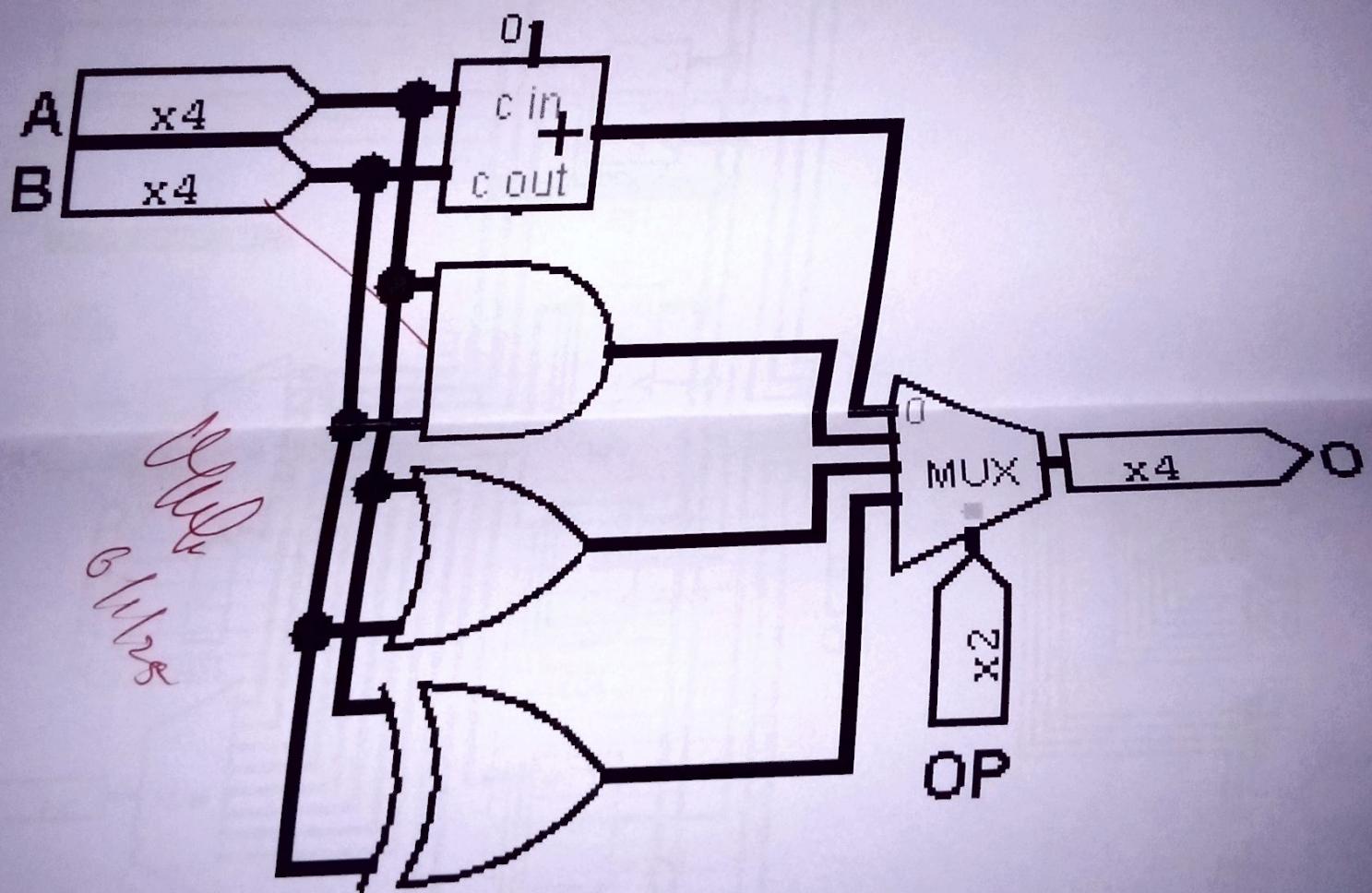
3. Team Members with USN:

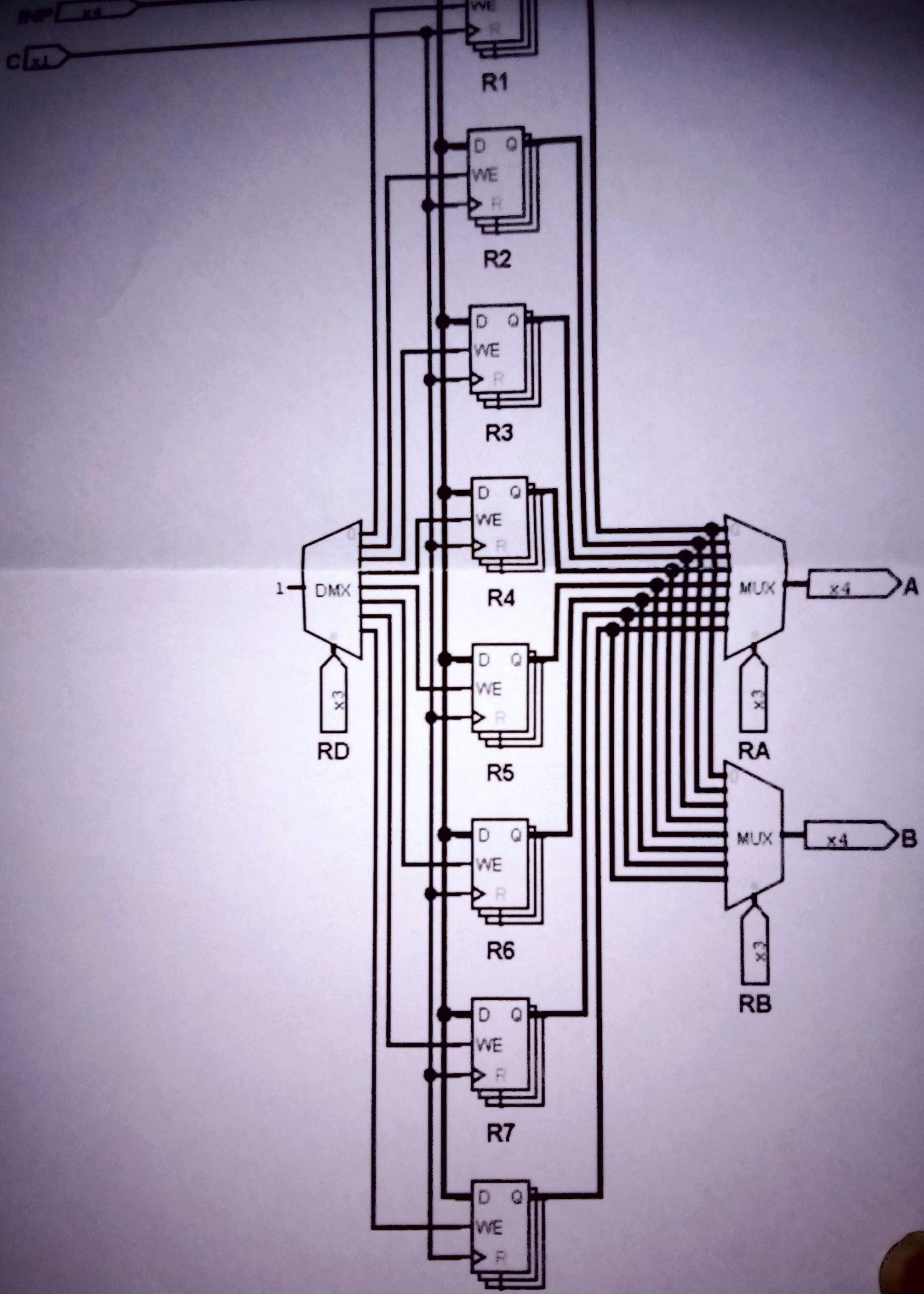
Aditya Bhandari (IRV23CD003)

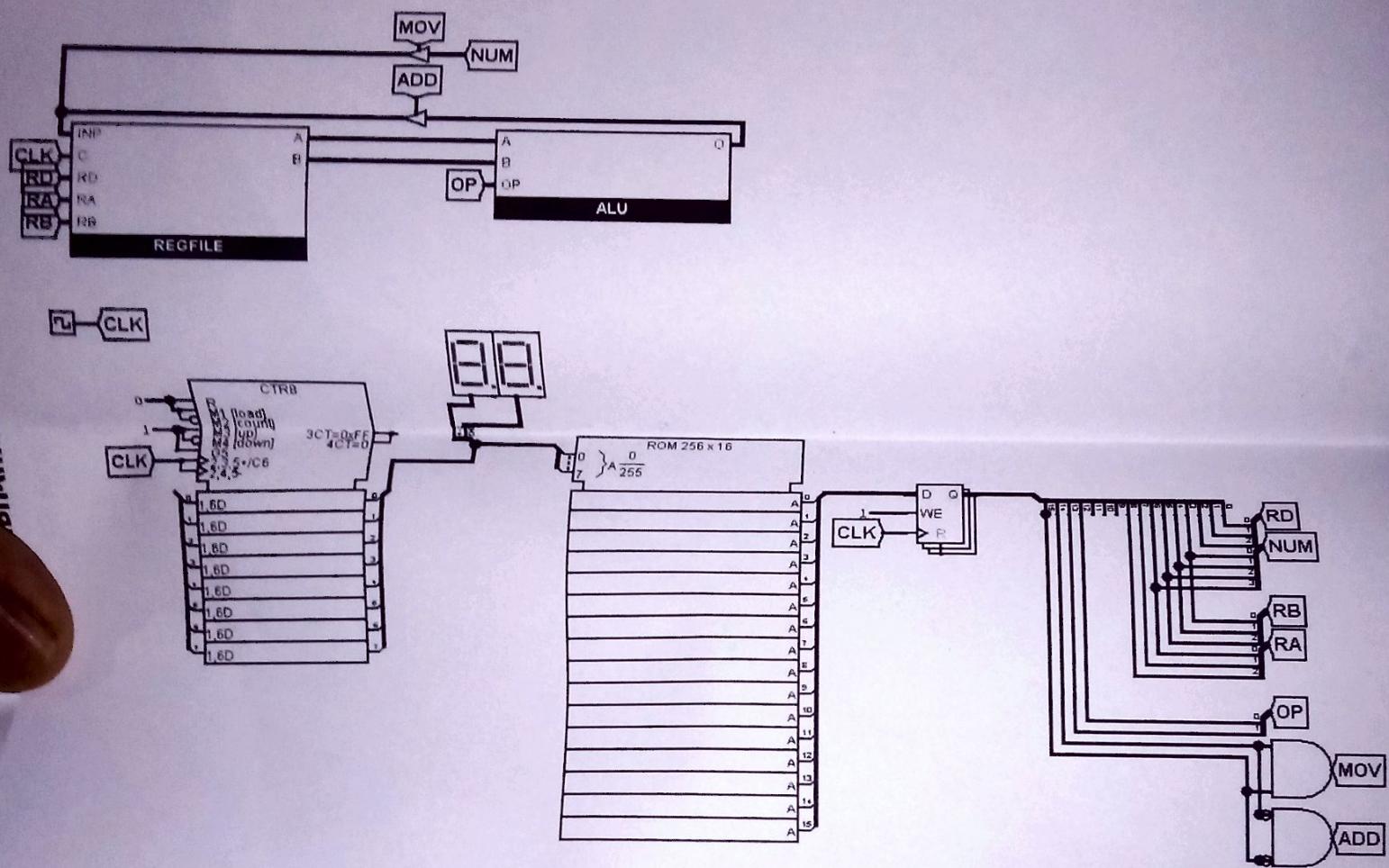
4. Components Used in Design and their specifications:

- Register file has 8 registers : R₀ - R₇
- RISC microcontroller
- 3-bus architecture (A, B, C)
- Memory is word addressable & program memory is 256x16.
- PC keeps track of clock cycles,

5. Block Diagram and Description (attach the print out)







R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

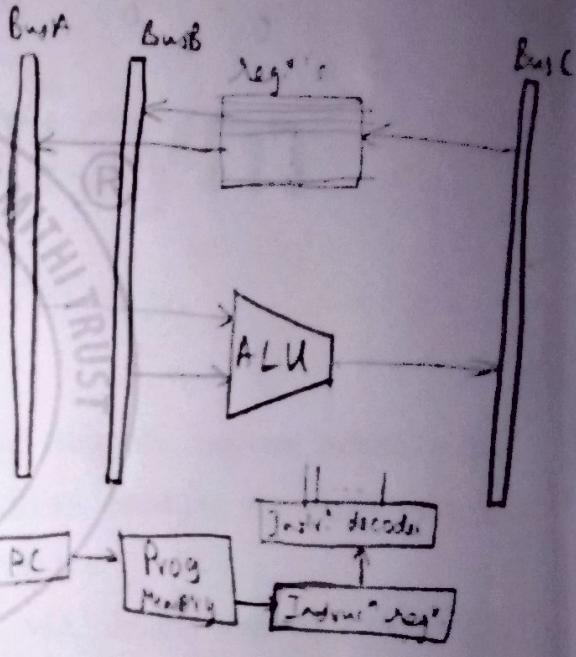
Date 30-12-24 Name Aditya Bhandari
 Dept./Lab _____ Class CD Expt./No. _____
 Title ADLD workshop

3-bit CPU architecture :

. ALU - 4-bit (ADD, AND, OR, XOR, ...)

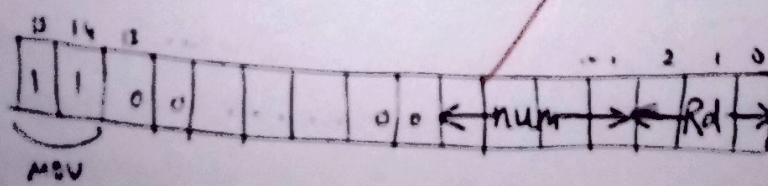
- . Register
- . Prog. counter: 8-bits
- . Prog. memory-
 - Instruction size: 16 bits.
 - 256 Instructions.
- . Instruction reg. - 16 bits.

- . . . decoder & control.
- . 3 buses - A, B, C.



Instruction template :

~~MOV~~ ~~#num~~ ~~→ Rd~~
 num
 operand



[Rd: 3 possible]
 0 1 2

R0: 000
 R1: 001
 R2: 010
 :
 R3: 111

e.g. ~~MOV~~ ~~→ S~~ ~~R1~~:

1100000000101001

C 0 2 9

~~30/12/24~~
 Signature of
 Teacher Incharge