

B.Sc. Engg. Part-2 Odd Semester Final Examination, 2021

## Lab Reports

Course Title: Digital System Design Lab

Course Code: CSE 2112



Submitted by	Submitted to
Name: Sakib Bin Faruque Rusho ID: 2010776109 Session: 2019-20 Department of CSE, University of Rajshahi	Mahboob Qaosar Associate Professor Department of CSE, University of Rajshahi

Department of CSE

University of Rajshahi

Date of Submission : 28<sup>th</sup> November, 2022

Sl.no	Name of the experiment	Page
1	Design of 4-bit adder/subtractor	1-6
2	Design and implementation of 2-bit magnitude comparator	7-16
3	Design and implementation of the BCD to XS-3 code converter	17-27
4	Design and implementation of the Excess-3 to BCD code converter	29-39
5	Design and implementation of the <del>Binary</del> to Gray code converter	41-48
6	Design and implementation of Gray to <del>Binary</del> code converter	49-54
7	Design and verification of 4-bit ripple counter	55-66
8	Design and verification of MOD-10 counter	67-76
9		

Name of the experiments:

Design of 4 bit adder

Subtractor.

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

Apparatus:

- ① IC - 7483
- ② X-OR gate (IC - 7486.)
- ③ Not gate (IC - 7404)
- ④ Breadboard
- ⑤ Trainer board.
- ⑥ Connecting wires.

Theory: 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, when  $M=1$  it becomes subtractor.

## Pin diagram:

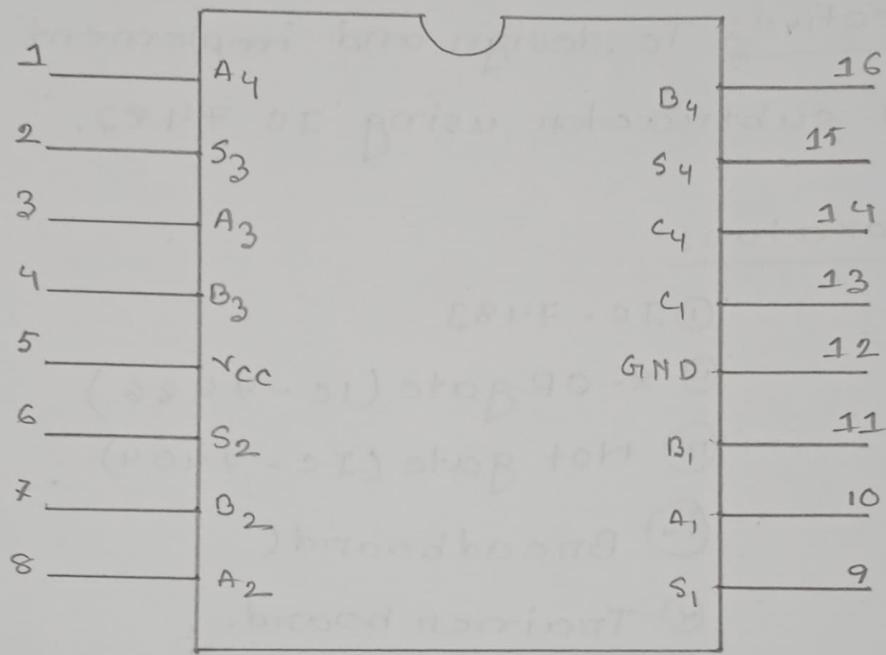
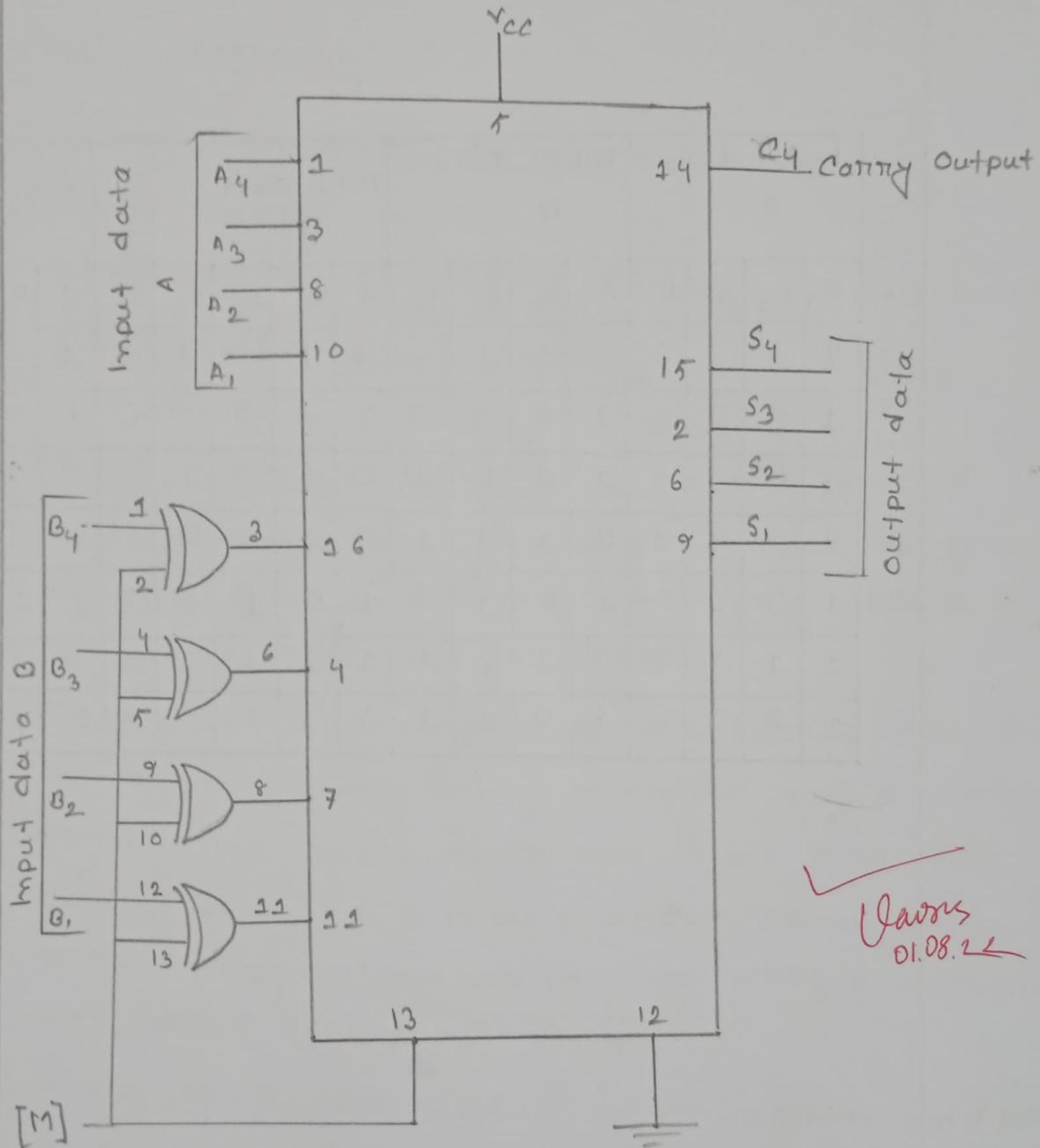


Fig-1: pin diagram IC-7483

## Logic Diagram:



$M = 0$  [ADDING]

$M = 1$  [SUBTRACTING] Fig 2: 4-bit binary adder/subtractor

✓ Davis  
01.08.12

## Truth tables:

Input data A				Input data B				Addition					Subtraction					
A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	C	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	C	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	
1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0
1	0	0	0	1	0	0	0	1	0	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	1
0	0	0	1	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	0	1	0	1	0	1	1	1	1	0	1	0	1	0	1	1	1	0

### Working Procedure:

- ① implemented the circuit according to the logic diagram.
- ② logic inputs were given according to truth table
- ③ observed the output of logic circuit and verify with truth table

Result and Discussion: The addition and subtraction operations can be done into one circuit with one common binary adder by including X-OR gate in input.

The 4-bit adder takes A and B two inputs data or 4-bit binary values or input and a mode controller and outputs are 4 bit sum and carry out. The B inputs are X-OR with the mode controller while the mode controller was input to the 13<sup>th</sup> pin.

When  $M=0$ , the circuit is an adder and when  $M=1$  circuit is a subtractor. Each OR gate receives input M and one of the inputs of B. When  $M=0$ , we have  $B \oplus 0 = B$ . The IC-7483 receives the value of B, the input carry is

When  $M=2$ , we have  $B \oplus 1 = B'$  and  $c_{in} = 1$  the input B, all one complemented and 1 added through the input carry. The circuit performs  $A + B$ 's complement of B.

### Precaution:

- ① IC's were checked before implementing the circuit.
- ② All connections were solid
- ③ Before changing anything in the circuit the power was turned off

Name of the experiment:

Design and implementation of 2 bit magnitude comparators using basic gates.

Objective: To design and implement 2 bit magnitude comparators using basic gates.

Apparatus:

- ① AND gate (IC-7408)
- ② X-OR gate (IC-7486)
- ③ OR-gate (IC-7432)
- ④ NOT-gate (IC-7404)
- ⑤ Trainer board.
- ⑥ Connecting wires.

Theory: The comparison of two numbers in an operators that determine one number is greater than, less than or equal to other numbers. 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 indicates whether  $A > B$ ,  $A = B$  or  $A < B$ .

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

Truth table:

A <sub>1</sub>	A <sub>0</sub>	B <sub>1</sub>	B <sub>0</sub>	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	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

K-map for  $A > B$ :

		B <sub>1</sub> B <sub>0</sub>	A <sub>1</sub> A <sub>0</sub>			
		00	01	11	10	00
		00	0	0	0	0
		01	1	0	0	0
		11	1	1	0	1
		10	1	1	0	0

$$\begin{aligned}
 A > B &= A_1 \bar{B}_1 + A_0 \bar{B}_1 \bar{B}_0 + A_1 A_0 \bar{B}_0 \\
 &= A_1 \bar{B}_1 + A_0 \bar{B}_0 (A_1 + \bar{B}_1)
 \end{aligned}$$

K-map for  $A = B$ :

		B <sub>1</sub> B <sub>0</sub>	A <sub>1</sub> A <sub>0</sub>			
		00	01	11	10	00
		00	1	0	0	0
		01	0	1	0	0
		11	0	0	1	0
		10	0	0	0	1

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

K-map for  $A < B$ :

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

		B <sub>1</sub> B <sub>0</sub>	A <sub>1</sub> A <sub>0</sub>			
		00	01	11	10	00
		00	0	1	1	1
		01	0	0	1	1
		11	0	0	0	0
		10	0	0	1	0

Circuit diagram:

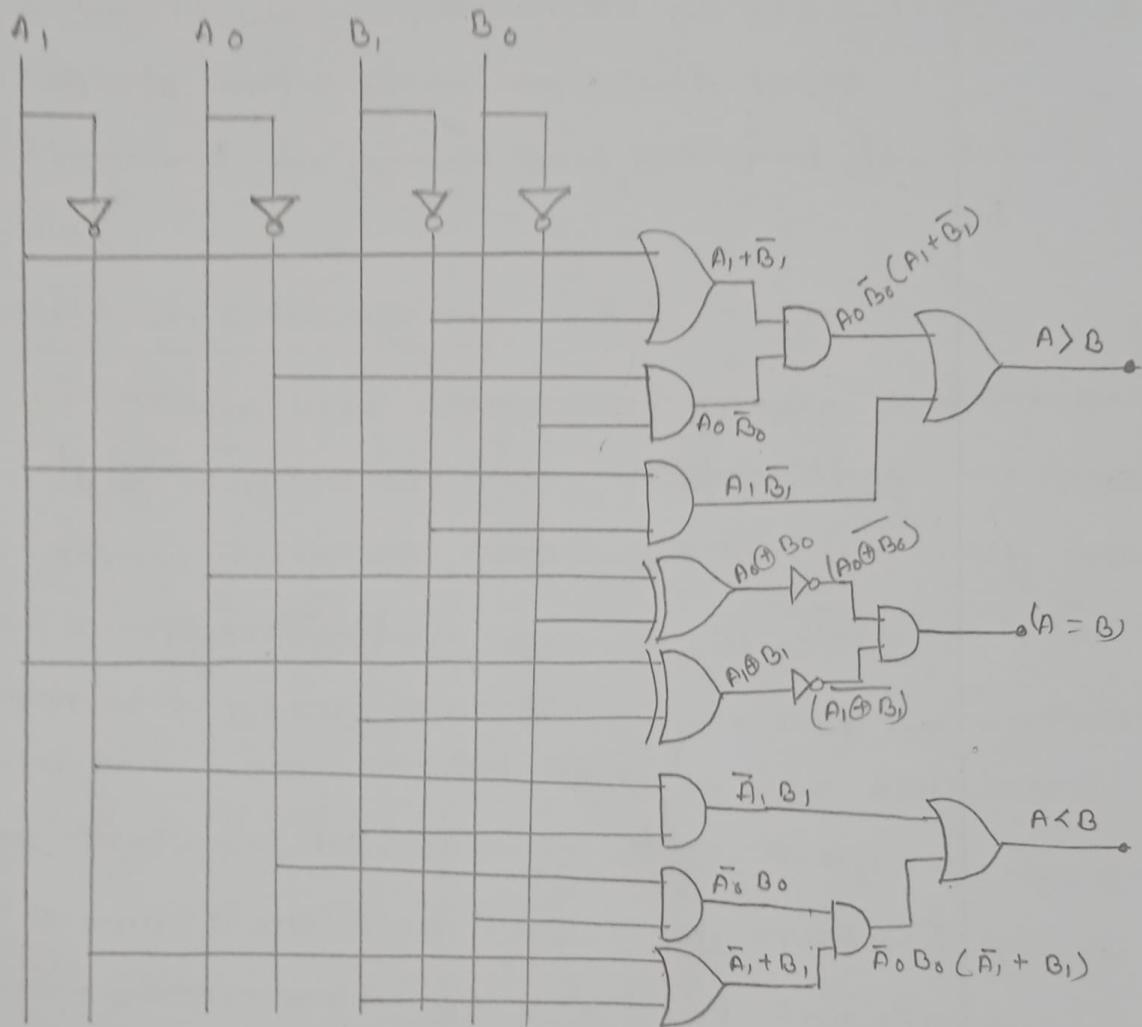


Fig : 2 bit magnitude comparators.

✓  
Dawn  
01.08.22

### Working procedure:

- ① circuit was implemented as circuit diagram.
- ② Inputs were given as truth table
- ③ Observed the output and verified the truth table.

### Result and discussion:

2 bit magnitude comparators can determine a value is greater than or less than or equal to other numbers. Here we can use only two bit numbers, they are [0, -3]. Here A is a number represented by  $A_1A_0$  and B another number represented by  $B_1B_0$ . We developed the boolean expression for  $A > B$ ,  $A = B$  and  $A < B$  and simplified by k-map method. From the experiment we see that the circuit matches the truth table. The 2 bit magnitude comparator has been verified.

### Precaution:

- ① All IC's were checked before implementing the circuit.
- ② All connections were solid and tight.
- ③ The power was turned off before changing anything in the circuit.

Name of the experiment:

Design and implementation of the code converter circuit using Digital IC trainer,

BCD to Excess-3 conversion

Objectives:

- ① To convert BCD(842) to Excess-3 code
- ② To develop the code converter using basic gates.

Theory:

Electronic digital systems use signals that have two distinct values and circuit elements have two stable states. There is a direct analogy among binary signals, binary circuit elements, and binary digits. A binary number of  $n$  digits, for example, may be represented by  $n$  circuit elements, each having an output signal equivalent to 0 or 1.

Binary codes for decimal digits require a minimum of four bits. Numerous different codes can be obtained by arranging four or more bits in ten distinct possible combinations. The BCD is a straight assignment of the binary equivalent. It is possible to assign weights to binary bits according to their positions. The weights in the BCD code are 8, 4, 2, 1. The bit assignment 0110, for example can be represented in decimal 6.

Truth table:

BCD				Excess-3			
A	B	C	D	W	X	Y	Z
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

To find the corresponding digital circuit, we used the K-Map technique for each of the Excess-3 code:

		CD	A'B		
		00	01	11	10
AB	00	1			1
	01	1			1
AB	11	x	x	x	x
	10	1		x	x

$Z = \bar{D}$

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

$Y = CD + \bar{C}\bar{D} = C \oplus D$

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

$$X = BC + \bar{B}D + B\bar{C}D'$$

Minimized expressions are,

$$W = A + BC + BD$$

$$X = BC + \bar{B}D + B\bar{C}D'$$

$$Y = CD + \bar{C}\bar{D} = C \oplus D$$

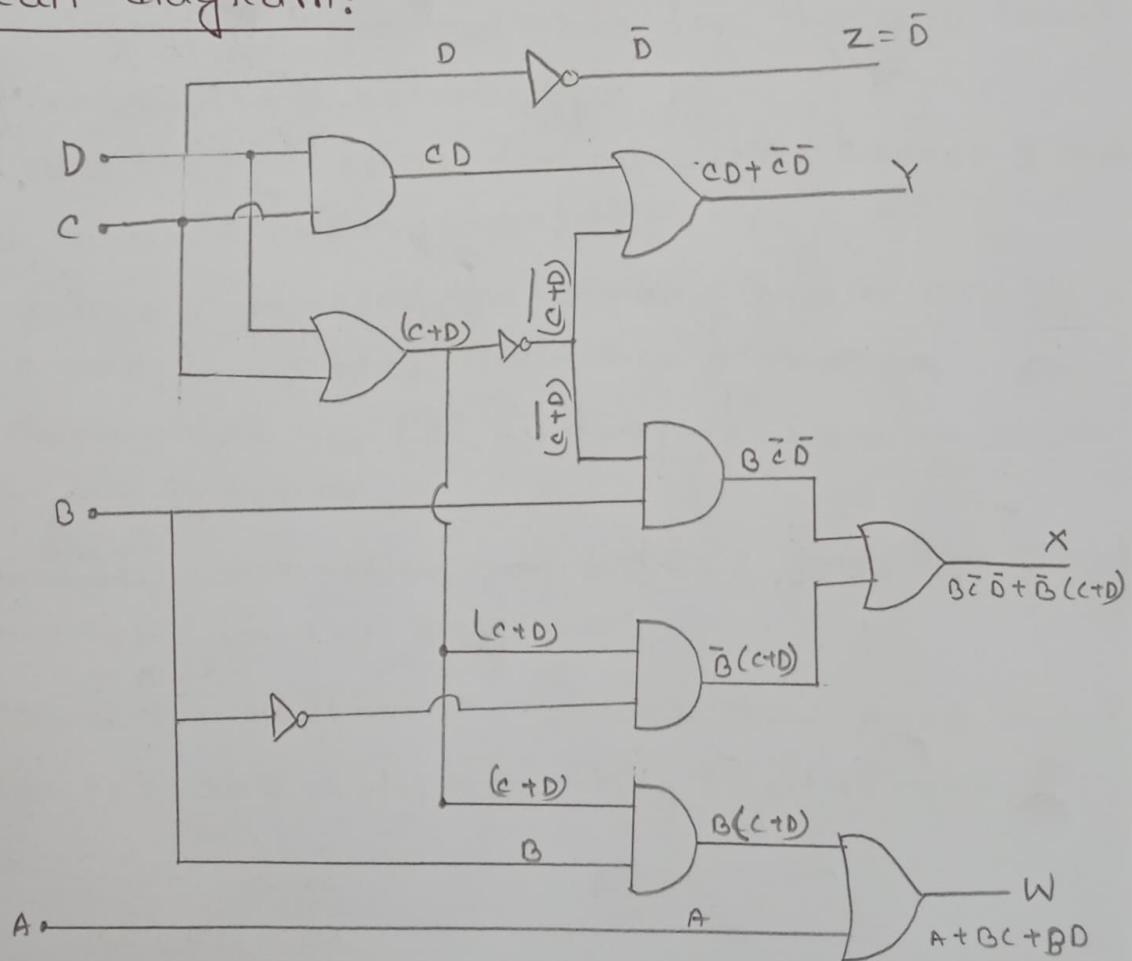
$$Z = \bar{D}$$

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

$$W = A + BC + BD$$

Apparatus:

- ① 2-input XOR gate (IC 7486)
- ② 2-input OR gate (IC 7432)
- ③ 2-input AND gate (IC 7408)
- ④ ~~2-input~~ NOT gate (IC 7404)
- ⑤ Power supply.
- ⑥ Bread Board
- ⑦ connecting wires.

Circuit diagram:

✓  
Date  
30.05.22

Working procedure:

1. Used the pre-lab for Boolean expression, and make the connections.
2. Connected the Digital Logic trainer to 220V AC power supply.
3. Turned on the
3. Insert the required IC's on the Trainer's breadboard.
4. Selected the logic switches for the BCD input and marked them carefully.
5. Selected the LED's for output as Excess-3 code and marked them carefully
6. Then selected the binary inputs one by one and completed the circuit slowly.
7. Connected V<sub>cc</sub> (of all IC's) to +5V available in the trainer.
8. Gave connection of all IC's GND to ground available in the trainer.
9. Then turned on the power and gave input (0-15) and verified the truth table.

### Result and discussion:

The process of conversion BCD to excess-3 is required to have knowledge of number system. The Excess-3 binary code is an example of a self-complementary BCD code. The sum of the 1st complement and binary numbers of a decimal is equal to the binary number of decimal 9.

In the above table, we take 4 variables  $A_1, B, C_1, D$  represent the bits of the binary numbers.

The variable  $D$  is LSB and  $A$  is MSB. In the XS-3 conversion  $w$  is MSB,  $z$  is LSB.

To convert BCD to XS-3, we added 3 i.e. 0011 to each of the BCD code.

From the experiment, we see that the circuit matches the truth table. Thus the operation of the BCD to XS-3 has been verified.

### Precautions:

- ① The circuit was implemented carefully.
- ② Wires were connected carefully.
- ③ The circuit was not powered on before the circuit computed.

Name of the experiment:

Design and implementation of the code converter Excess-3 to BCD circuit using Digital IC trainer.

Objectives:

- ① To convert Excess-3 to BCD code.
- ② To develop the code converter using basic gates.

Theory:

Electronic digital systems use digital signals that have two distinct values and circuit elements have two stable states. There is a direct analogy among binary signals, binary circuit elements, and binary digits. A binary number of  $n$  digits may be represented by  $n$  circuit elements each having an output signal equivalent to 0 or 1.

Truth table:

Excess-3				BCD			
W	X	Y	Z	A	B	C	D
0	0	0	0	x	x	x	x
0	0	0	1	x	x	x	x
0	0	1	0	x	x	x	x
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
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

To find corresponding digital circuit, we used K-map technique.

	Wx	YZ	00	01	11	10
00	x	x	0	x		
01	1	0	0	1		
11	1	x	x	x		
10	1	0	0	1		

$$D = \bar{z}$$

	Wx	YZ	00	01	11	10
00	x	x	0	x		
01	0	1	0	1		
11	0	x	x	x		
10	0	1	0	1		

$$C = Y'Z + YZ' = Y \oplus Z$$

	Wx	YZ	00	01	11	10
00	x	x	0	x		
01	0	0	1	0		
11	0	x	x	x		
10	1	1	0	1		

$$B = \bar{x}\bar{y} + \bar{x}\bar{z} + xy\bar{z}$$

	Wx	YZ	00	01	11	10
00	x	x	0	x		
01	0	0	0	0		
11	1	x	x	x		
10	0	0	1	0		

$$A = w\bar{x} + w\bar{y}z$$

The minimized expression for each output obtained from K-map are given below,

$$A = w\bar{x} + w\bar{y}z$$

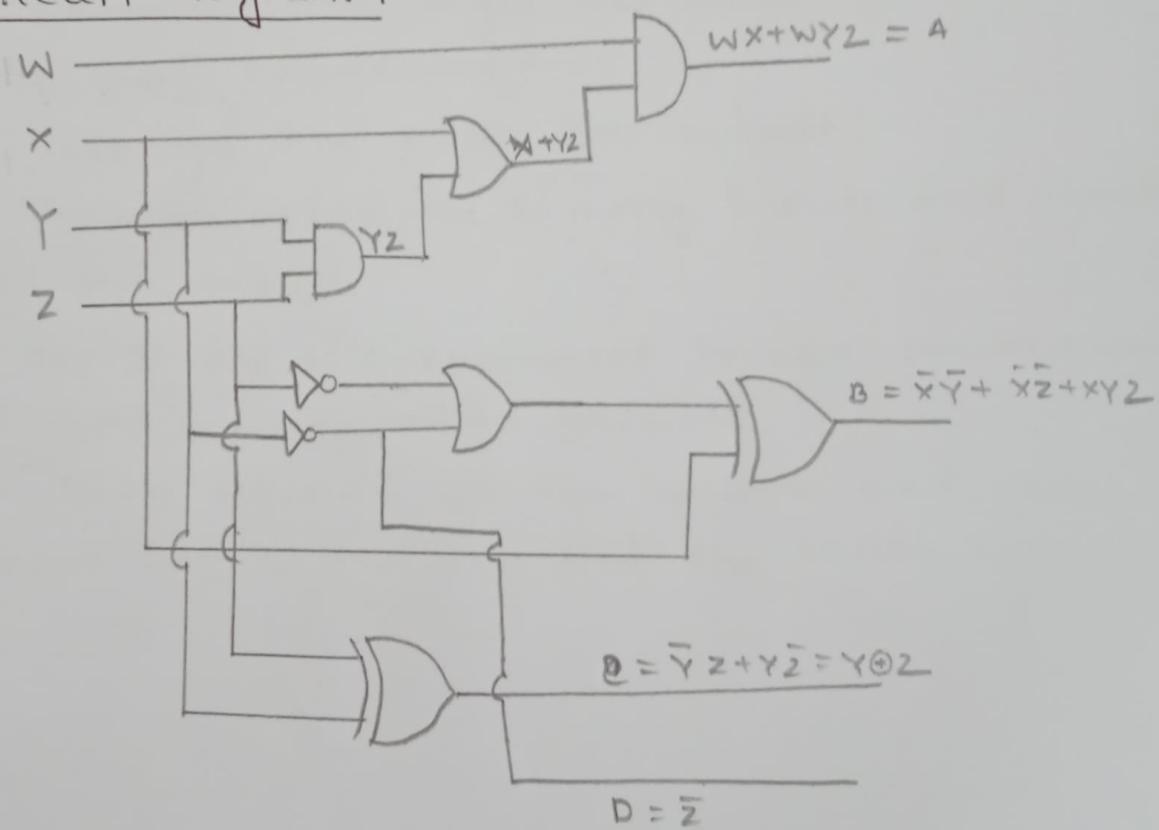
$$B = \bar{x}\bar{y} + \bar{x}\bar{z} + xy\bar{z}$$

$$C = \bar{y}z + y\bar{z} = Y \oplus Z$$

$$D = \bar{z}$$

Apparatus:

1. 2-input XOR gate (IC 7486)
2. 2-input OR gate (IC 7432)
3. 2-input AND gate (IC 7408)
4. 2 input Not gate (IC 7404)
5. Power Supply.
6. Bread Board.
7. Wires.

Circuit diagram:

✓ *Darry  
30.05.22*

### Working Procedure:

1. Used the pre-lab for Boolean expression.
2. Connected the Digital Logic Trainer to 220V AC power supply.
3. Inserted the required IC's on the trainer breadboard.
4. Selected the logic switches for the ~~LED~~ <sup>RS-3</sup> trainers breadboard.
5. Selected the LED's for output.
6. Then selected the binary inputs and completed the circuit.
7. Vcc of all IC's connected to +5V power supply
8. GND connected to ground
9. Then turned on the power and gave input (0-15) and verified the truth table

### Result and discussion:

From the truth table, we took variables A, B, C, D. W, X, Y, Z represent the XS-3 code. Here W is MSB and Z is LSB. Again we took variables A, B, C, D represent the BCD. A is MSB, D is LSB.

XS-3 code begins with 0011 and ends at 1100. Since we don't use 0, 1, 2, 13, 14 and 15 as inputs. we put don't care term :

From the experiment, we see that the circuit matches the truth table. Thus the operation of XS-3 to BCD conversion has been verified.

### Precaution:

1. The circuit was implemented carefully.
2. Wires were connected carefully.

Name of the experiment :

Design and implementation of the code conversion circuit using digital IC 4011 <sup>Binary</sup> to gray code.

Apparatus:

1. 2-input X-OR gate.
2. 2 - input OR gate.
3. 2 - input AND gate.
4. NOT gate
5. power supply.
6. Bread board
7. wire.

Truth table:

Binary				Gray			
$B_3$	$B_2$	$B_1$	$B_0$	$G_{r_3}$	$G_{r_2}$	$G_{r_1}$	$G_{r_0}$
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	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

To find corresponding digital circuit, we used the k-map technique.

$B_3 B_2$	$B_3 B_2$	00	01	11	10
$B_3 B_2$	$B_3 B_2$	00	01	11	10
00	00	0	0	1	1
01	01	0	0	1	1
11	11	0	0	1	1
10	10	0	0	1	1

$$G_{T_3} = B_3$$

$B_3 B_2$	$B_3 B_2$	00	01	11	10
$B_3 B_2$	$B_3 B_2$	00	01	11	10
00	00	0	1	0	1
01	01	0	1	0	1
11	11	0	1	0	1
10	10	0	1	0	1

$$\begin{aligned} G_{T_2} &= \bar{B}_2 B_3 + \bar{B}_3 B_2 \\ &= B_2 \oplus B_3 \end{aligned}$$

$B_3 B_2$	$B_3 B_2$	00	01	11	10
$B_3 B_2$	$B_3 B_2$	00	01	11	10
00	00	0	1	1	0
01	01	0	1	1	0
11	11	1	0	0	1
10	10	1	0	0	1

$$G_{T_1} = \bar{B}_2 \oplus B_1$$

$B_3 B_2$	$B_3 B_2$	00	01	11	10
$B_3 B_2$	$B_3 B_2$	00	01	11	10
00	00	0	0	0	0
01	01	1	1	1	1
11	11	0	0	0	0
10	10	1	1	1	1

$$G_T = \cancel{B_0} \cancel{B_1} B_2 \oplus B_0$$

Mimimized statements,

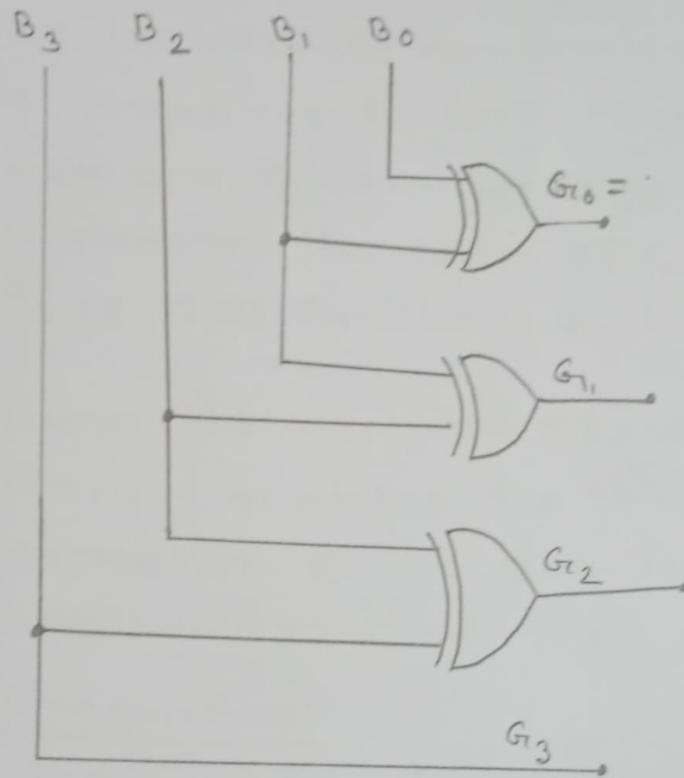
$$G_{T_0} = B_0 \oplus B_1$$

$$G_{T_1} = B_1 \oplus B_2$$

$$G_{T_2} = B_2 \oplus B_3$$

$$G_{T_3} = B_3$$

### Circuit diagram:



✓ Jaiswal  
30.05.12

Fig: 8-bit binary to Gray converter circuit.

### Working procedure:

1. Inserted the IC's in the breadboard.
2. Connected all the wires as shown in the figure.
3. Check the output in the LED.
4. V<sub>cc</sub> connected to +5V AC power supply
5. GND connected to ground in the trainer.

LHM

### Result and discussion:

The process of conversion BCD to Gray code is required to have to have knowledge of number system. We took variables  $B_0 B_1 B_2 B_3$  represent the codes of BCD,  $B_0$  is ~~MSB~~<sup>LSB</sup> and  $B_3$  is MSB. Similarly  $G_3$  is MSB and  $G_0$  is LSB.

From the experiment, we can see that the circuit matches the truth table. Thus the operation of the BCD to gray code conversion has been verified.

### Precaution:

1. Make the connection according to the IC pin diagram.
2. The connections should be tight on breadboard kit.
3. The Vcc ground should be applied carefully of the specified pin only

Name of the experiment:

Design and implementation of the code convert circuit using digital IC trainee array code to BCD

Apparatus:

1. 2-input X-OR gate
2. 2-input OR gate
3. Power supply
4. Breadboard.
5. Wires.

Truth table: Gray | Binary

Gray	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
0000	0	0	0	0	0	0	0	0
0001	0	0	0	1	0	0	0	1
0011	0	0	1	1	0	0	1	0
0010	0	0	1	0	0	0	1	1
0110	0	1	1	0	0	1	0	0
0111	0	1	1	1	0	1	0	1
0101	0	1	0	1	0	1	1	0
0100	0	1	0	0	0	1	1	1
1100	1	1	0	0	1	0	0	0
1101	1	1	0	1	1	0	0	1
1111	1	1	1	1	1	0	1	0
1110	1	1	1	0	1	0	1	1
1010	1	0	1	0	1	0	0	0
1011	1	0	1	1	1	1	0	1
1001	1	0	0	1	1	1	1	0
1000	1	0	0	0	1	1	1	1

		G <sub>1</sub> , G <sub>0</sub>				
		G <sub>1</sub> ⊕ G <sub>2</sub>	00	01	11	10
G <sub>1</sub> , G <sub>0</sub>		00	0	1	0	1
G <sub>1</sub> , G <sub>0</sub>		01	1	0	1	0
G <sub>1</sub> , G <sub>0</sub>		11	0	1	0	1
G <sub>1</sub> , G <sub>0</sub>		10	1	0	1	0

$$B_0 = G_{T_3} \oplus G_{T_2} \oplus G_{T_1} \oplus G_T$$

		G <sub>1</sub> , G <sub>0</sub>				
		G <sub>1</sub> ⊕ G <sub>2</sub>	00	01	11	10
G <sub>1</sub> , G <sub>0</sub>		00	0	0	1	1
G <sub>1</sub> , G <sub>0</sub>		01	1	1	0	0
G <sub>1</sub> , G <sub>0</sub>		11	0	0	1	1
G <sub>1</sub> , G <sub>0</sub>		10	1	1	0	0

$$B_1 = G_{T_3} \oplus G_{T_2} \oplus G_{T_1}$$

		G <sub>1</sub> , G <sub>0</sub>				
		G <sub>1</sub> ⊕ G <sub>2</sub>	00	01	11	10
G <sub>1</sub> , G <sub>0</sub>		00	0	0	0	0
G <sub>1</sub> , G <sub>0</sub>		01	1	1	1	1
G <sub>1</sub> , G <sub>0</sub>		11	0	0	0	0
G <sub>1</sub> , G <sub>0</sub>		10	1	1	1	1

$$B_2 = G_{T_2} \oplus G_{T_1}$$

		G <sub>1</sub> , G <sub>0</sub>				
		G <sub>1</sub> ⊕ G <sub>2</sub>	00	01	11	10
G <sub>1</sub> , G <sub>0</sub>		00	0	0	0	0
G <sub>1</sub> , G <sub>0</sub>		01	0	0	0	0
G <sub>1</sub> , G <sub>0</sub>		11	1	1	1	1
G <sub>1</sub> , G <sub>0</sub>		10	1	1	1	1

$$B_3 = G_{T_3}$$

Mimized expression:

$$B_0 = G_{T_3} \oplus G_{T_2} \oplus G_{T_1} \oplus G_T$$

$$B_1 = G_{T_3} \oplus G_{T_2} \oplus G_{T_1}$$

$$B_2 = G_{T_2} \oplus G_{T_1}$$

$$B_3 = G_{T_3}$$

## Circuit diagram:

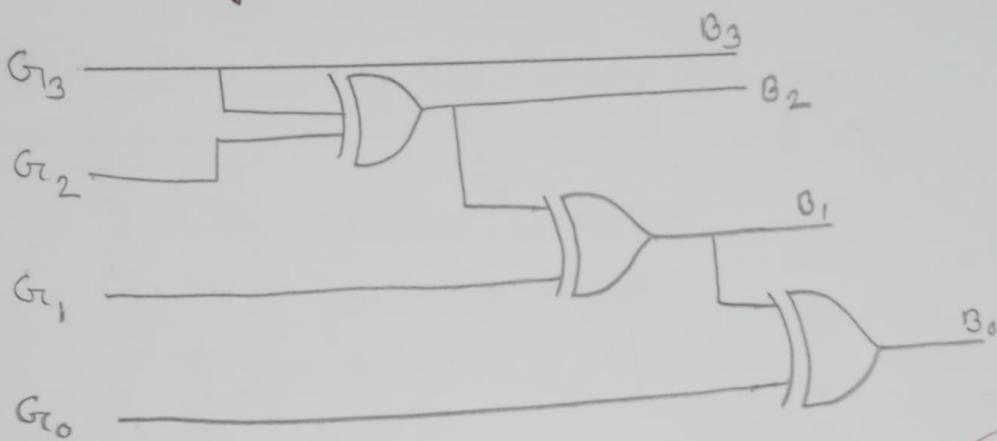


Fig : Gray to Binary converter.

✓ Date  
30/05/22

## Result:

From the experiment, we can see that the circuit matches the truth table. Thus operation of the Gray to BCD conversion has been verified.

## Precaution:

- ① Make the connection as shown in the figure.

Name of the experiment:

Design and verify 4 bit ripple counter

Apparatus:

1. 2 JK Flip-flop (IC 7476)
2. Nand Gate (IC 7400)
3. IC Trainer kit
4. Patch cards

Theory:

A computer is a circuit consisting of a number of flipflop and gates working together to count the number of a clock pulses applied to its input. Such counters are used in digital clocks, frequency counters, digital computers and numerous other applications. There are types of counters and we can't look at them in this experiment. The basic binary counter is probably the simplest to construct and form the basis for more advanced types of counters. In this experiment, we look at some the counter circuits found most often and give you an opportunity to conceal and observe them.

The ripple counter is a serial counter. The clock input is applied to only the first of the series of the flipflop. Gate clock pulse for the other flipflop come from preceding flip flop. Thus the clock pulse "ripple" through

the circuit in a series fashion. Such circuit is also called asynchronous since the only pulse required for the operation is the clock-pulse. Soon the clock of second stage is triggered by output of first stage. Because of inherent propagation delay time all flip-flops are not activated at same time which results in asynchronous operation.

Pin diagram for IC 7476:

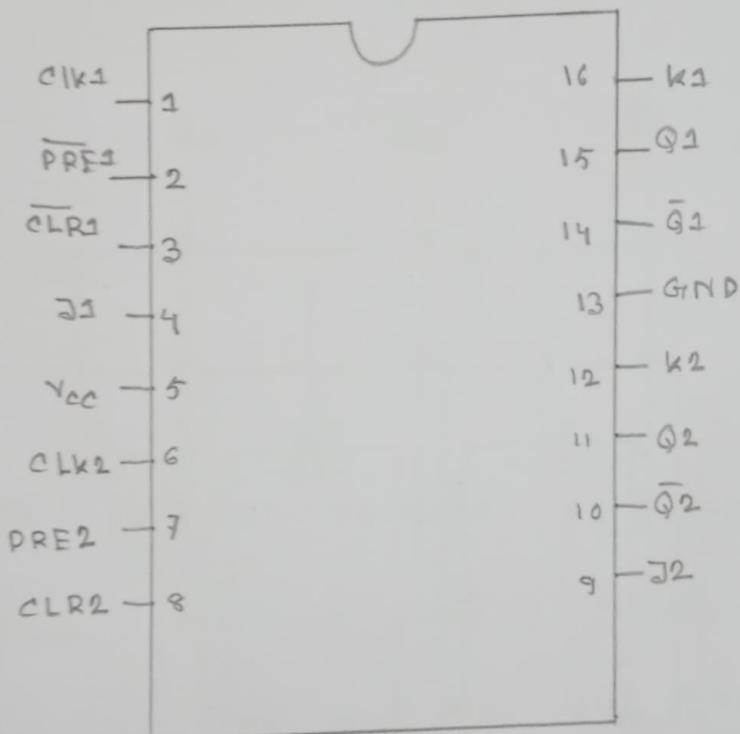


fig: pin diagram for IC 7476

Logic diagram for 4 bit ripple counters:

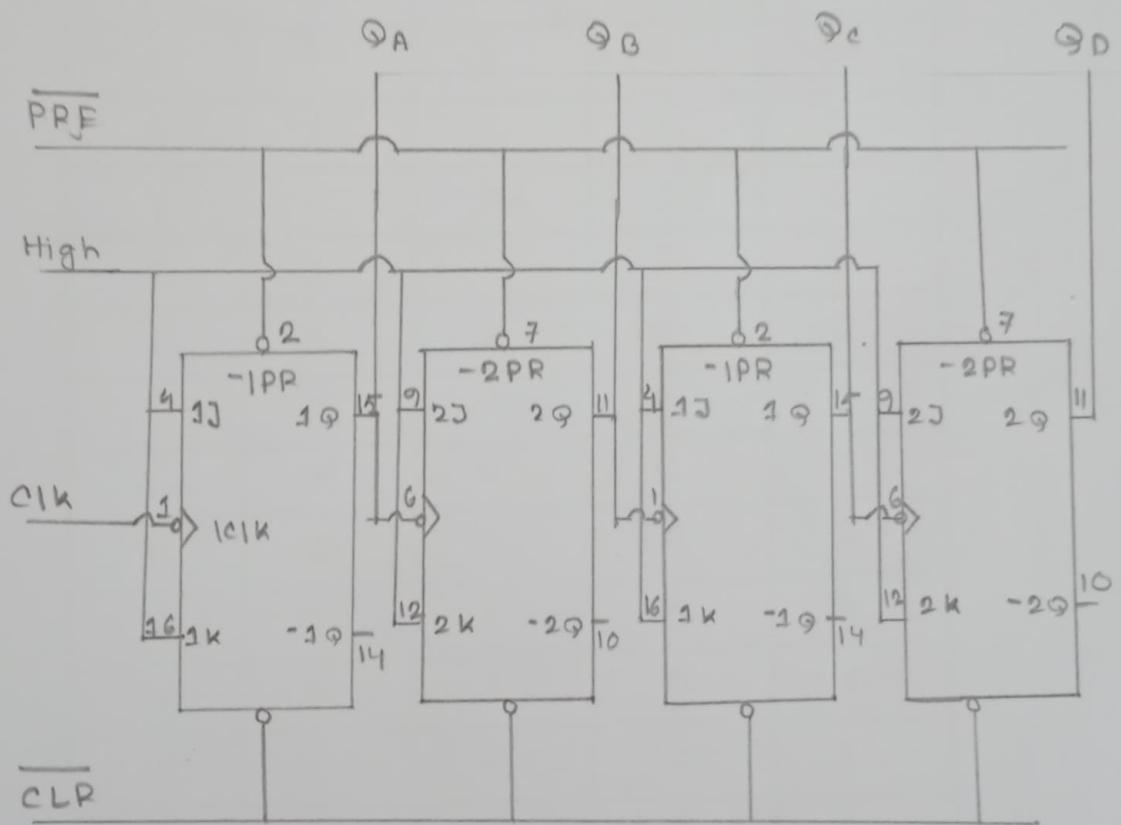
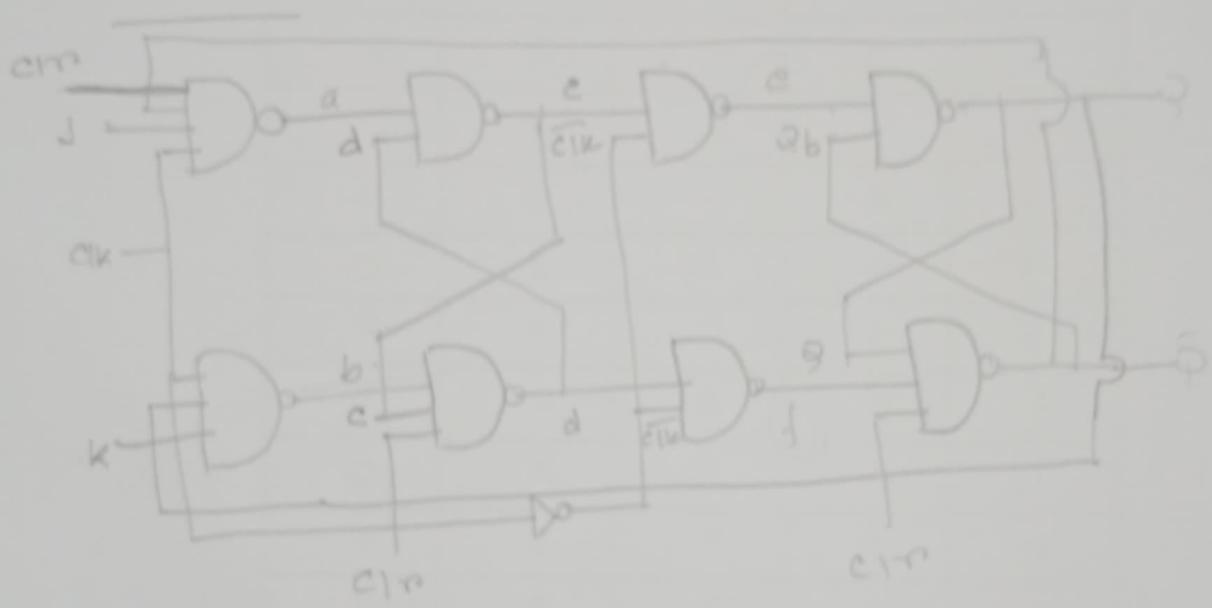


fig: 4 bit ripple counter diagram

✓ *Jerry  
01.08.21*

Truth table:

C1K	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>c</sub>	Q <sub>D</sub>
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1



Procedures:

- ① Connections are given as per circuit diagram.
- ② logical inputs are given as per circuit diagram
- ③ Observed the output and verified the truth table

Name of the experiments:

Design and verify mod 10 counter

Apparatus:

- ① 2JK flip flops (1C7476)
- ② NAND gate (1C7400)
- ③ IC Trainer kit.
- ④ Patch cords

Theory:

A counter is a register capable of counting numbers of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appear as counter output. This is the main difference between a register and a counter. There are two types of counter synchronous and asynchronous. In synchronous, in synchronous clock is given to all flip flop and in asynchronous first flip flop is clocked by external pulse and then each successive flip flop is clocked by the output of previous stage. An asynchronous counter can have  $2^n$  possible counting states e.g. MOD 16 for a 4-bit counter (0-15) making it ideal for use in frequency division application. But it is also possible to use the basic asynchronous counter configuration.

Pin diagram for 7476:

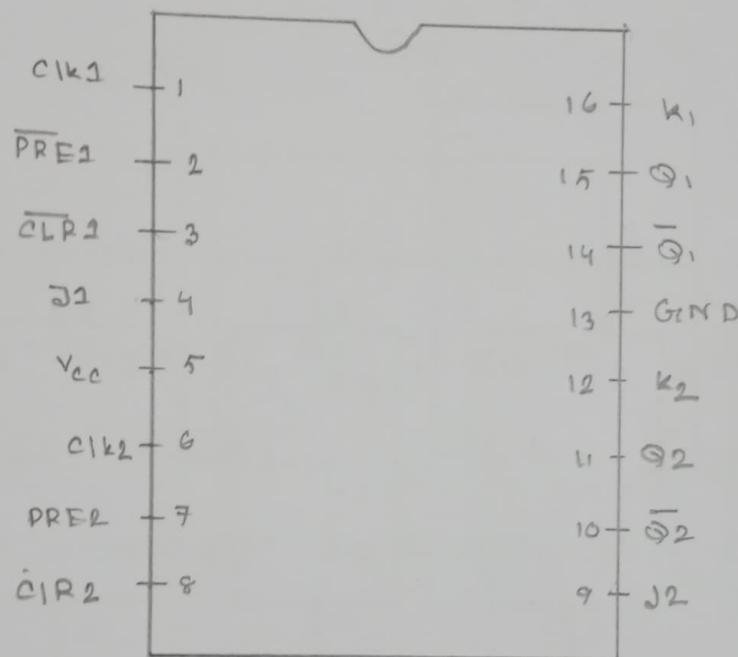


fig: Pin diagram for IC 7476

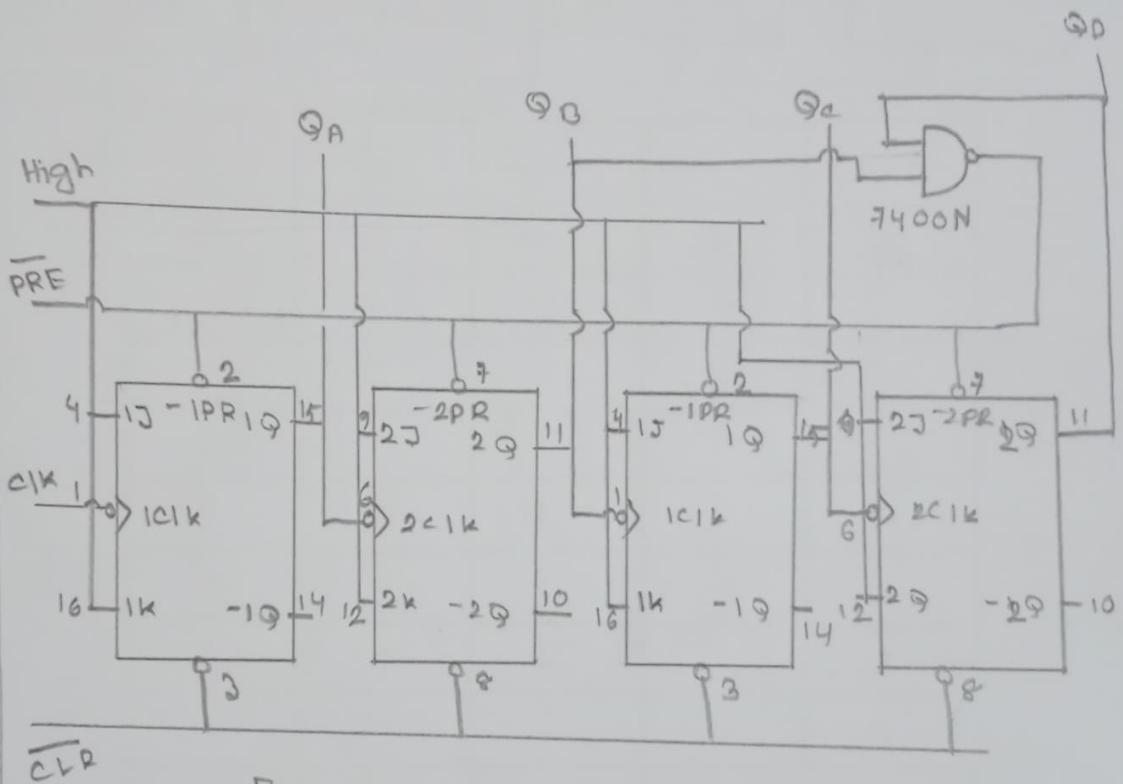
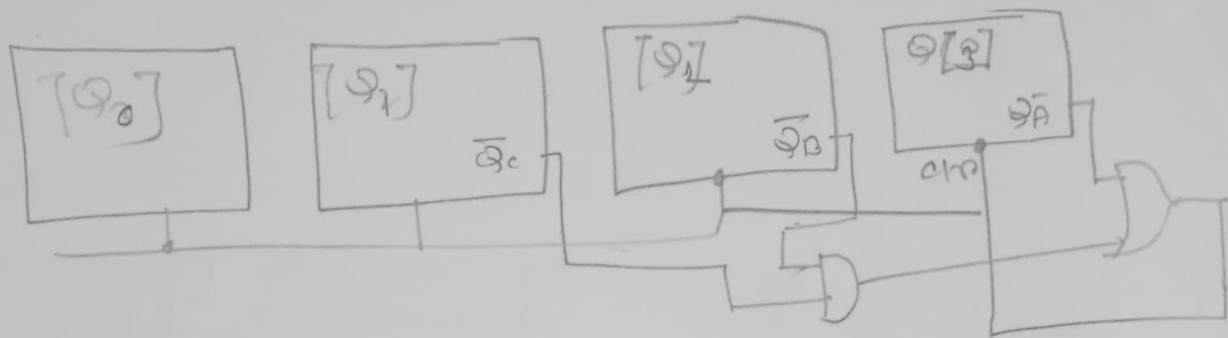


Fig: Mod 10 counter

✓ Deonus  
01.08.22



Truth table:

C1K	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	0	0	0
11	1	1	1	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

Procedure:

- ① Connections are given as per circuit diagram
- ② Logical inputs are given as per circuit diagram
- ③ Observe the output and verified the truth table