

# INDEX

Serial No	Name of the Experiment
1	To realize half/full adder and half/full subtractor using X-OR and Basic gates.
2	(a).Realization of logic expresation using k-map for the expresation $Y=f(x,y,z)=\sum(0,2,4,5,6)$ . (b). Implementation of parity generation checker circuit using Ex-OR and Ex-NOR gates.
3	To design and construct BCD to excess-3-code converter.
4	To design and construct 3-to -8 line Decoder.

# PABNA UNIVERSITY OF SCIENCE AND TECHNOLOGY



DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING

## Practical Lab Report

Course Code: 2103

Course Title: Sessional Based on Digital electronics.

### NAME OF THE EXPERIMENT:

To realize half/full adder and half/full subtractor using X-OR and Basic gates

### Submitted By-

Name: MD:MUNNAF HOSSEN

Roll: 130614

Session: 2012-13

Date of submission: 30.11.14

### Submitted To-

Sohag Sarker

Lecturer,

Department of

Information&Communication

Engineering

Pabna University of Science and

Technology

30.11.14

Sign of Teacher

Name of the Experiment: To realize half/full adder and half/full subtractor using X-OR and basic gates.

Objectives: Construct half/full adder and half/full subtractor using X-OR and basic gates NOT, OR and AND.

Theory:  
Logic gates: A digital circuit which one or more input signal but only one output signal is called a logic gate.

Xe-OR gate: The X-OR gate has a graphical symbol similar to the OR-gate except for the additional curve line on the input side.

primary Logic gate: The logic gate AND, OR, NOT are called the primary or fundamental logic gates or basic gates.

Half Adder: A Half Adder is a combinational circuit that forms the arithmetic sum of two inputs bits. It consists two binary input and two binary output s, for sum and c for carry.

The truth table for half-adder:

x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full adder: A full adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs and two outputs. The outputs are designated by the symbols s for sum and c for carry. The truth table for full adder:

x	y	z	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Half Subtractor: A half-subtractor is a combinational circuit that subtracts two bits and produces their difference. It also has an output to specify if a 1 has been borrowed. Designate the minuend bit by  $x$  and the subtrahend bit by  $y$ . To perform  $x-y$ .

The Boolean function for two outputs of the half subtractors are derived directly from the truth table

$$D = \bar{x}y + xy'$$

$$B = \bar{x}y$$

$x$	$y$	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Full Subtractor: A full-subtractor is a combinational circuit that performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage. This circuit has three input and two outputs. The three inputs are  $x, y, z$ .

$$B = \bar{x}y + \bar{x}z' + yz$$

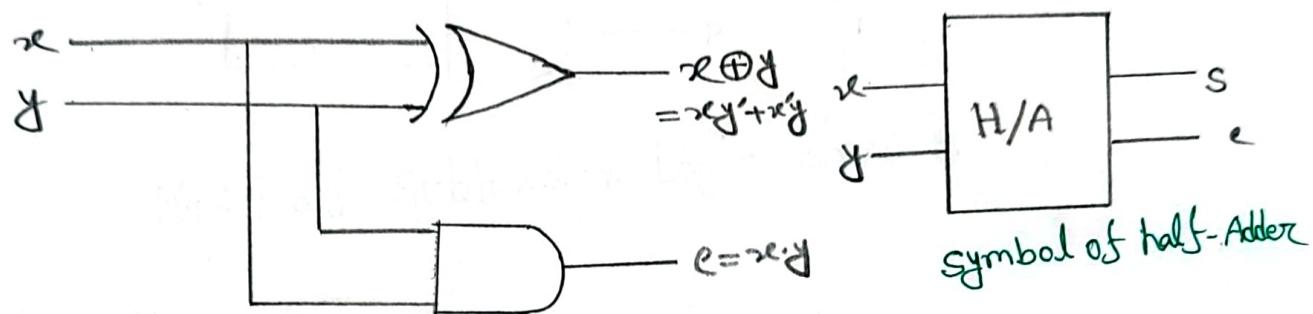
$$D = \bar{x}y'z + xy'z' + xyz$$

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

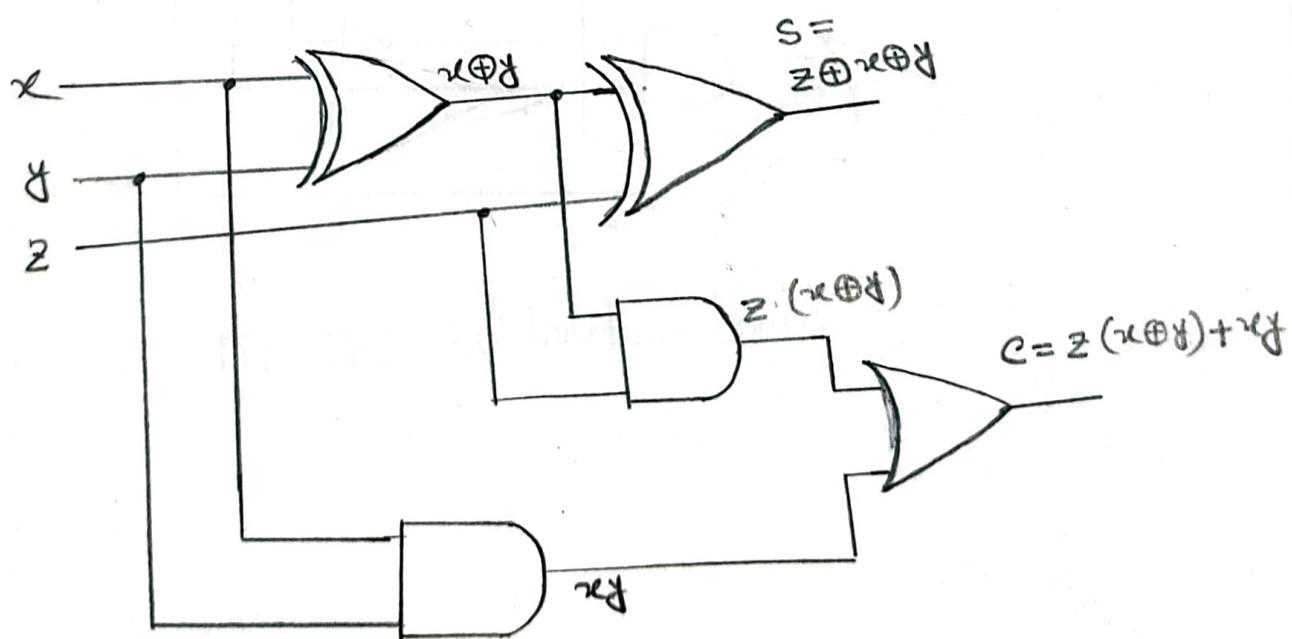
### Apparatus Required:

- i) Bread board.
- ii) AVO meter
- iii) Connecting wires.
- iv) IC-7486, IC-7432, IC-7408, IC-7400
- v) LED
- vi) D.c supply etc.

## Experimental - Set up



Half-Adder Logic Circuit



Full-Adder logic circuit

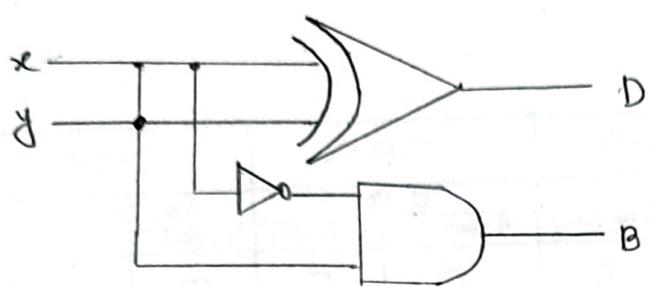


Fig: Half-Subtractor Logic circuit.

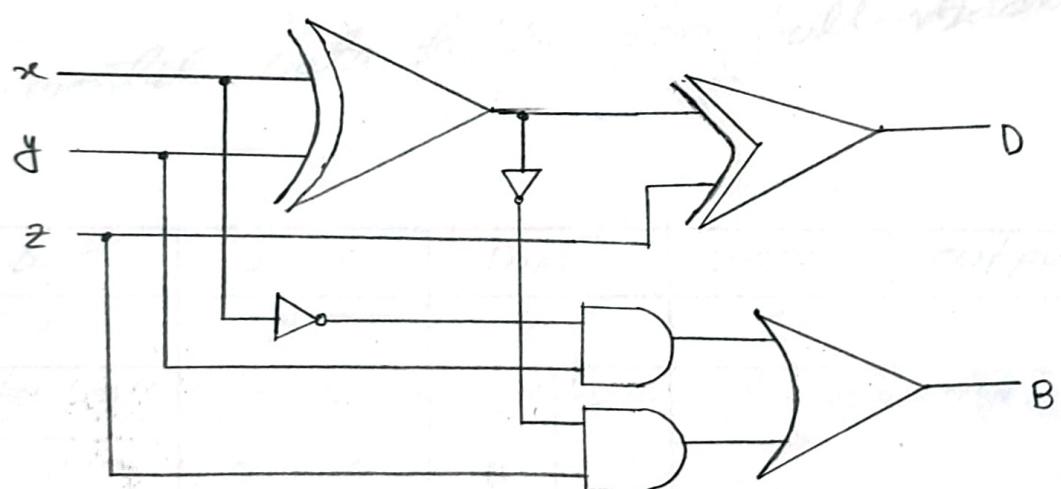


Fig: Full-Subtractor Logic circuit.

Roll No: 130614

## Experimental Data Table for Half-Adder

No of operation	Input		Input		output		output	
	x	y	voltage	Logic	voltage	Logic	voltage	Logic
1	0	0	0	0	0	0	0	0
2	0	0	5	1	0	0	5	1
3	5	1	0	0	0	0	5	1
4	5	1	5	1	5	1	5	1

## Experimental Data Table for full-Adder

No of operation	Input			Input			Input		output		
	x	y	z	voltage	Logic	voltage	Logic	voltage	Logic	voltage	Logic
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	5	1	0	0	5	1	1
3	0	0	5	1	0	0	0	0	5	1	1
4	0	0	5	1	5	1	5	1	0	0	0
5	5	1	0	0	0	0	0	0	5	1	1
6	5	1	0	0	5	1	5	1	0	0	0
7	5	1	5	1	0	0	5	1	0	0	0
8	5	1	5	1	5	1	5	1	5	1	1

Roll No 8130614

## Experimental data table for half-subtractor

no of separation	Input	Input	output		output	
	$\oplus x$	$\oplus y$	$\oplus B$	$\oplus D$	Logic	Logic
	voltage	Logic	voltage	Logic	voltage	Logic
1	0	0	0	0	0	0
2	0	0	5	1	5	1
3	5	1	0	0	5	1
4	5	1	5	1	0	0

## Experimental data table for full-subtractor

no of separation	Input	Input	Input	output		output	
	$\oplus x$	$\oplus y$	$\oplus z$	$\oplus B$	$\oplus D$	Logic	Logic
	voltage	Logic	voltage	Logic	voltage	Logic	
1	0	0	0	0	0	0	0
2	0	0	0	0	5	5	1
3	0	0	5	1	0	5	1
4	0	0	5	1	5	1	0
5	5	1	0	0	0	0	5
6	5	1	0	0	5	1	0
7	5	1	5	1	0	0	0
8	5	1	5	1	5	1	5

0	5	1	5	1	2	1	5	1	5	1
---	---	---	---	---	---	---	---	---	---	---

Table 3  
Experimental Data Table for Half Adder

No of the observation	Input (x) voltage (V)	Input (y) voltage (V)	Output (c) voltage (V)	Output (s) voltage (V)
	Logic 0	Logic 1	Logic 0	Logic 1
1	0	0	0	0
2	0	0	5	1
3	5	1	0	0
4	5	1	5	1

Experimental Data Table for full Adder

No of the observation	Input (x) voltage (V)	Input (y) voltage (V)	Input (z) voltage (V)	Output (c) voltage (V)	Output (s) voltage (V)	Output (s) voltage (V)
	Logic 0	Logic 1	Logic 0	Logic 1	Logic 0	Logic 1
1	0	0	0	0	0	0
2	0	0	0	5	1	0
3	0	0	5	1	0	0
4	0	0	5	1	5	1
5	5	1	0	0	0	0
6	5	1	0	5	1	5
7	5	1	5	1	0	0
8	5	1	5	1	5	1

Result and Discussion: From the experimental data table we see that the half-adder when the input variable  $i, o$  and  $o, i$  the sum is 1 and light has on. otherwise 0 and light has off. From the table of a full-adder circuit, we see that the input variable and position  $2, 3, 5, 8$  then sum is 1. so light has on and otherwise off. again position  $4, 6, 7, 9$  output carry is 1. so light has on and - otherwise off.

From the table of half-subtractor when input  $o, i$  and  $i, o$  then subtrac' is 1 and when input  $o, i$  carry is 1. so light is on. otherwise off.

From the table of full-subtractor we see that when input variable position is  $2, 3, 5, 8$  then subtrac' is 1 and when input position variable is  $2, 3, 4, 8$  then carry is 1. so light has on. otherwise light has off.

Experimental data table for Half-subtractor:

No of the observation	Input ( $s_1$ )		Input ( $s_0$ )		Output ( $B$ )		Output ( $D$ )	
	voltage (V)	logic	voltage (V)	logic	voltage (V)	logic	voltage (V)	logic
1	0	0	0	0	0	0	0	0
2	0	0	5	1	5	1	5	1
3	5	1	0	0	0	0	5	1
4	5	1	5	1	0	0	0	0

Experimental data table for full-subtractor:

No of the observation	Input ( $s_1$ )		Input ( $s_0$ )		Input ( $b_1$ )		Output ( $B$ )		Output ( $D$ )	
	voltage (V)	logic	voltage (V)	logic	voltage (V)	logic	voltage (V)	logic	voltage (V)	logic
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	5	1	5	1	5	1
3	0	0	5	1	0	0	5	1	5	1
4	0	0	5	1	5	1	5	1	0	0
5	5	1	0	0	0	0	0	0	5	1
6	5	1	0	0	5	1	0	0	0	0
7	5	1	5	1	0	0	0	0	0	0
8	5	1	5	1	5	1	5	1	5	1

## Precaution:

- ① The connecting wires and all required components were connected on the breadboard very carefully.
- ② Voltage measured very carefully.
- ③ The logic combination always take carefully.

# PABNA UNIVERSITY OF SCIENCE AND TECHNOLOGY



DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING

## Practical Lab Report

Course Code: 2103

Course Title: Sessional Based on Digital electronics.

### NAME OF THE EXPERIMENT:

- ① Realization of logic expression using k-map for the Expression  $f = \sum(0, 2, 4, 5, 6)$
- ② Implementation of parity generation checker circuit using k-OR and Ex-NOR gates.

### Submitted By-

Name: MD:MUNNAF HOSSEN

Roll: 130614

Session: 2012-13

Date of submission:

### Submitted To-

Sohag Sarker

Lecturer,

Department of

Information & Communication

Engineering

Pabna University of Science and

Technology

31.03.15  
Sign of Teacher

Q) Name of the Experiment: Q) Realization of logic expression using K-map for the expression

$$Y = f(x_1, y_2, z) = \Sigma (0, 2, 4, 5, 6)$$

⑥ Implementation of parity generation checker circuits using Ex-OR and Ex-NOR gates.

8 Theory: K-map:

① The map method provides a simple straight forward procedure for minimizing Boolean functions. This method may be regarded either as a pictorial form of a truth table or as an extension of the Venn diagram. The map method first proposed by Veitch and modified by Karnaugh is also known as the 'Veitch diagram' or 'Karnaugh map'.

The map is a diagram made up squares. Each square represents one minterm. The map presents a visual diagram of all possible ways a function may be expressed in a standard form. The simplest algebraic expression is any one in a sum of products or product of sum that has a minimum number of literals.

For example,  $f(x_1, y_2, z) = \Sigma (0, 2, 4, 5, 6)$ . Here we are given the minterms by their decimal numbers. Using K-map the corresponding squares are marked by 1's in down in below.

		Y2	00	01	11	10
		0	1			1
		1	1	1		1

Fig: k-map for  $F(x_2x_1x_0) = \Sigma(0, 2, 4, 5, 6)$   
 From the map we obtain the simplified function:  
 $F = z' + xy'$

### ⑥ Parity bit generator and Parity checker:

A parity bit is a scheme for detecting errors during transmission of binary information. A parity bit is an extra bit included with a binary message to make the number either odd or even.

The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker.

Consider, a three bit message to be transmitted as odd parity bit. The three bit  $x_2, x_1, x_0$  constitute the message and are inputs to the circuit. The parity bit  $p$  is output. For odd parity, the bit  $p$  is generated so as to make the total number of 1's odd. The truth table for the odd parity generator is given below.

Input			Output
Parity bit message			Parity bit generator
x	y	z	p
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

The simplified the sum of products expression are:-

$$\begin{aligned}
 p &= xy'z' + x'y'z + xy'z + xyz' \\
 &= z'(xy + x'y) + z(x'y + xy') \\
 &= z'(x \oplus y) + z(x \oplus y) \\
 &= z'(x \oplus y)' + z(x \oplus y) \\
 &= z \oplus (x \oplus y)
 \end{aligned}$$

The hence bit message and the parity bit are transmitted to their destination, when they are applied to a parity checker circuit.

The output e of the parity checker should be a 1 when an errors i.e when the number of 1's in the four inputs is even. The truth table for parity checker circuit is given below:-

Four bit received				Parity checker
x	y	z	p	c
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

The simplified sum of product expressions are:

$$Q = xy'z'p' + xy'zp + xyz'p + xyzp' + xy'z'p$$

$$+ xyz'p' + xyzp' + xyzp$$

$$= xy(zp + z'p') + x'y'(zp + z'p')$$

$$+ xyz'(zp + z'p) + xyz(zp + z'p)$$

$$= (xy + x'y')(zp + z'p') + (xyz' + xyz)(zp + z'p)$$

$$= (x \oplus y)(zp + z'p) + (x \oplus y)(zp + z'p)$$

$$= (x \oplus y)(zp + z'p) + (x \oplus y)'(zp + z'p)$$

$$= (x \oplus y)Q(zp + z'p)$$

### Q) Apparatus Required:

- i. Bread Board
- ii. AVO meter
- iii. Connecting wires
- iv. IC-7432 for OR gate  
IC-7408 for AND gate  
IC-7404 for NOT gate  
IC-7486 for Ex-OR gate

### Q) Circuit Diagram:

(a)

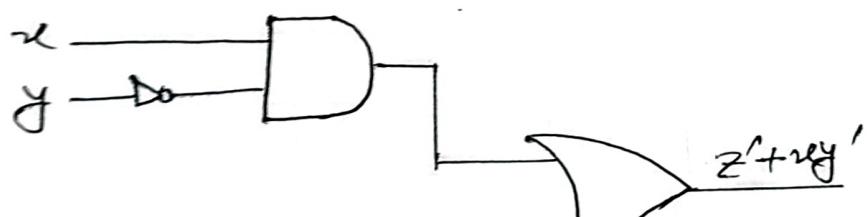


Fig 1: K-maping circuit for function  $f(x,y,z) = \Sigma(0,2,4,5,6)$ .

(b)

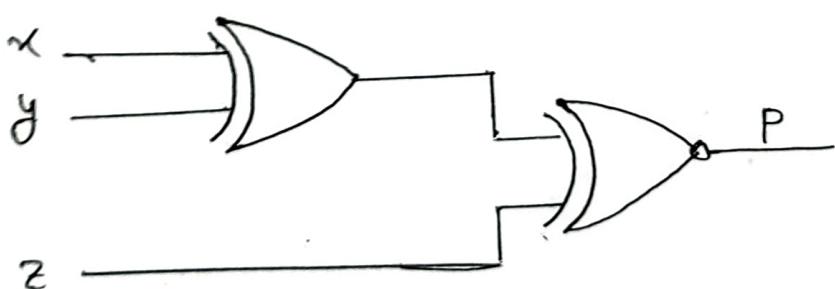


Fig: Odd parity generator circuit by using Ex-OR and Ex-NOR gate

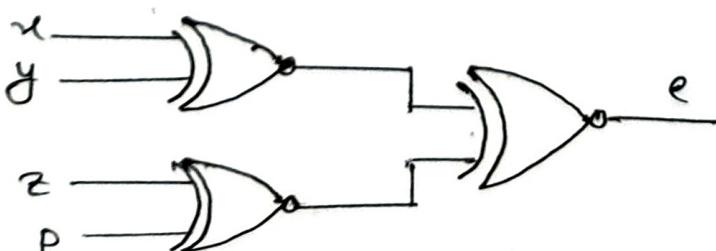


Fig: Parity checker by using Ex-OR and Ex-NOR gate

Table - 1 : Truth table for  $y = f(x_1, x_2, x_3) = \sum(0, 2, 4, 5, 6)$   
 using K-mapping.

No of decimation abstraction	Input ( $x_1$ )		Input ( $x_2$ )		Input ( $x_3$ )		Output, $= 2^3 + xy$	
	voltage	Logic	voltage	Logic	voltage	Logic	voltage	Logic
1	0	0	0	0	0	0	5	1
2	0	0	0	0	5	1	0	0
3	0	0	5	1	0	0	5	1
4	0	0	5	1	5	1	0	0
5	5	1	0	0	0	0	5	1
6	5	1	0	0	5	1	6	1
7	5	1	5	1	0	0	5	1
8	5	1	5	1	5	1	0	0

Table-3: Truth table for parity check.

Observation number	Input (x)		Input (y)		Input (z)		Input (p)		Output (parity check)	
	voltage (V)	logic (V)	voltage (V)	logic (V)	voltage (V)	logic (V)	voltage (V)	logic (V)	voltage (V)	logic (V)
1	0	0	0	0	0	0	0	0	5	1
2	0	0	0	0	0	0	5	1	0	0
3	0	0	0	0	5	1	0	0	0	0
4	0	0	0	0	5	1	5	1	5	1
5	0	0	5	1	0	0	0	0	0	0
6	0	0	5	1	0	0	5	1	5	1
7	0	0	5	1	5	1	0	0	0	0
8	0	0	5	1	5	1	5	1	0	0
9	5	1	0	0	0	0	5	1	5	1
10	5	1	0	0	0	0	0	0	5	1
11	5	1	0	0	5	1	5	1	0	0
12	5	1	0	0	5	1	5	1	5	1
13	5	1	5	1	0	0	5	1	0	0
14	5	1	5	1	0	0	5	1	0	0
15	5	1	5	1	5	1	0	1	5	1
16	5	1	5	1	5	1	5	1	5	1

Q) Experimental Data: Table for realization of expression using k-map for the expression and Implementation of parity generator check Ex-OR and Ex-NOR gates.

Table-1: Truth table for  $x = F(x,y,z) = \Sigma(0,3,5,6)$  using k-map.

Input			Output		
$x$	$y$	$z$	$z'$	$xy$	$z+xy$
0	0	0	1	0	1
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	0	0	0

Table-2: Truth Table for odd parity

$x$	$y$	$z$	$P$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Table-2: Truth table for odd parity

no of observation	Input ( $x_0$ )		Input ( $x_1$ )		Input ( $x_2$ )		output ( $P$ ) (odd parity)	
	voltage (V)	Logic	voltage (V)	Logic	voltage (V)	Logic	voltage (V)	Logic
1	0	0	0	0	0	0	5	1
2	0	0	0	0	5	1	0	0
3	0	0	5	1	0	0	0	0
4	0	0	5	1	5	1	5	1
5	5	1	0	0	0	0	0	0
6	5	1	0	0	5	1	5	1
7	5	1	5	1	0	0	5	1
8	5	1	5	1	5	1	0	0

~~25.0.3~~  
25.0.3

Table-2: Truth table for odd parity

Table-3: Truth table for odd parity check.

x	y	z	p	c
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Result And Discussion: (a) For k-mapping, we can use AND OR and NOT gate. Then we are constructed the circuit and then the truth table are verified.

(b) It is obvious from that parity generation and checking circuits have an output function that includes half of the minterms whose numerical values have either an even or odd number of 1's. For parity bit generation and checker we can used EX-OR and NOT gate. Then the circuit are constructed and truth table are verified.

Precaution:

1. All the connections should be made properly.
2. IC should not be reversed.

# PABNA UNIVERSITY OF SCIENCE AND TECHNOLOGY



DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING

## Practical Lab Report

Course Code: 2103

Course Title: Sessional Based on Digital electronics.

### NAME OF THE EXPERIMENT:

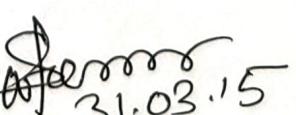
To Design and Construct BCD Code to  
excess-3-code converter.

### Submitted By-

Name: MD:MUNNAF HOSSEN  
Roll: 130614  
Session: 2012-13  
Date of submission:

### Submitted To-

Sohag Sarker  
Lecturer,  
Department of  
Information&Communication  
Engineering  
Pabna University of Science and  
Technology

  
31.03.15  
Sign of Teacher

Q) Name of the Experiment: To design and construct BCD code to excess-3-code.

Q) Theory:

BCD codes: The term BCD refers to representing the ten decimal digits in binary forms, which simply means to count in binary. Numeric codes represent numeric information i.e. only numbers as a series of 0's and 1's. Numerical codes used to represent decimal digits are called Binary coded decimal (BCD) codes. A BCD code is one, in which the digits of a decimal number are encoded one at a time into group of four digits. There are a large number of BCD codes in order to represent decimal digits 0, 1, 2, ..., 9. It is necessary to use a sequence of at least four binary digits. Such a sequence of binary digits which represents a decimal digit is called code word.

Excess-3-Code: It is non-weighted code. It is also a self-complementing BCD code used in decimal arithmetic units. The excess-3-code for the decimal number is performed in the same manner as BCD except that decimal number 3 is added to each decimal unit before encoding it to binary.

Code converters: The availability of a large variety of codes for the same discrete elements of information results in the use of different codes by different digital systems. It is sometimes necessary to use the output of one system as the input to the other. The conversion circuit must be inserted between the two systems if each uses different codes for the same information. Thus a code converter is a circuit that makes the two systems compatible even though each uses the different code. The bit combinations for BCD and excess-3 codes are listed below:

Input BCD				Output excess-3 code				
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	

To simplifying the function in sum of products is listed under the map of each variable.

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

$$Z = D'$$

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

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

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

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

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

$$w = A + BC + BD$$

A two level logic diagram may be obtain directly from the map.

Apparatus Required:

- (i) Bread board
- (ii) Connecting wires
- (iii) Power supply
- (iv) AVO meter
- (v) IC-7408 for AND gate
- (vi) IC-7432 for OR gate
- (vii) IC-7404 for NOT gate

Circuit diagram:

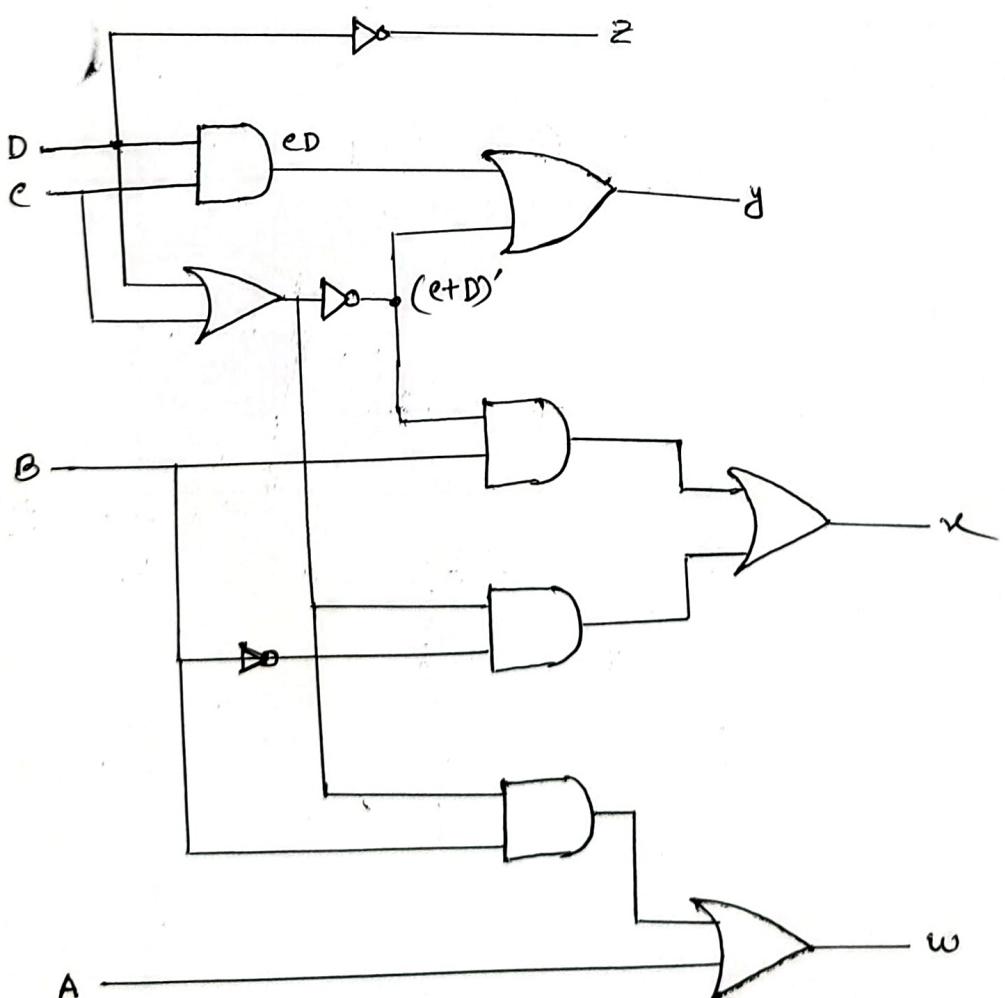


Fig: Logic diagram for BCD-to-excess-3 code converter.

Truth Table for Code Conversion from  
BCD code to excess-3 code.

Input BCD				Output Excess-3 code			
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	2	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

if  
at  
of

- ① Preparing  
table
- ② All the components of the given problem
- ③ Great a part
- ④ To highlight out the required

④ Experimental data: Truth Table for code conversion from BCD code to excess-3 code.

Table :

Input BCD				Output Excess-3 code			
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

⑤ Result and discussion: By using four AND gates, four OR gates and one inverter gate the circuit was constructed and verified the truth table of BCD to excess-3 code.

⑥ Precaution:

- ① All the component set up properly on the Bread board.
- ② IC should not be reversed.

# PABNA UNIVERSITY OF SCIENCE AND TECHNOLOGY



DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING

## Practical Lab Report

Course Code: 2103

Course Title: Sessional Based on Digital electronics.

### NAME OF THE EXPERIMENT:

To Design AND Construct 3-to-8 line Decoder.

### Submitted By-

Name: MD:MUNNAF HOSSEN

Roll: 130614

Session: 2012-13

Date of submission:

### Submitted To-

Sohag Sarker

Lecturer,

Department of

Information & Communication

Engineering

Pabna University of Science and

Technology

31.03.15  
Sign of Teacher

Q) Name of the Experiment: To design and construct 3-to-8 line decoder.

Q) Theory:

Decoder: A decoder is a combinational circuit that converts binary information from  $n$  input lines to a maximum of  $2^n$  unique output lines. If the  $n$ -bit decoded information has unused or don't combinations, the decoder output will have less than  $2^n$  outputs.

The decoder presented here are called  $n$ -to- $m$  line decoders where  $m \leq 2^n$ . Their purpose is to generate the  $2^n$  minterms of  $n$  input variables, In 3-to-8 Line Decoder. The three inputs are decoded into eight outputs, each output representing one of the minterms of the 3-input variables. Three inverters provide the complement of the input and each of the eight AND gates generates one of the minterms. The input variable may represent a binary number and the output will represent the eight digit in the octal number system. However, a 3-to-8 line decoder can be used for decoding any 3-bit code to provide eight outputs, one for each element of the code.

Q) Apparatus Required:

- (i) Bread Board
- (ii) Connecting wire
- (iii) AVO meter
- (iv) Power Supply (d.c)
- (v) IC - 7408 for AND gate
- (vi) IC - 7404 for NOT gate.

Q) Circuit Diagram:

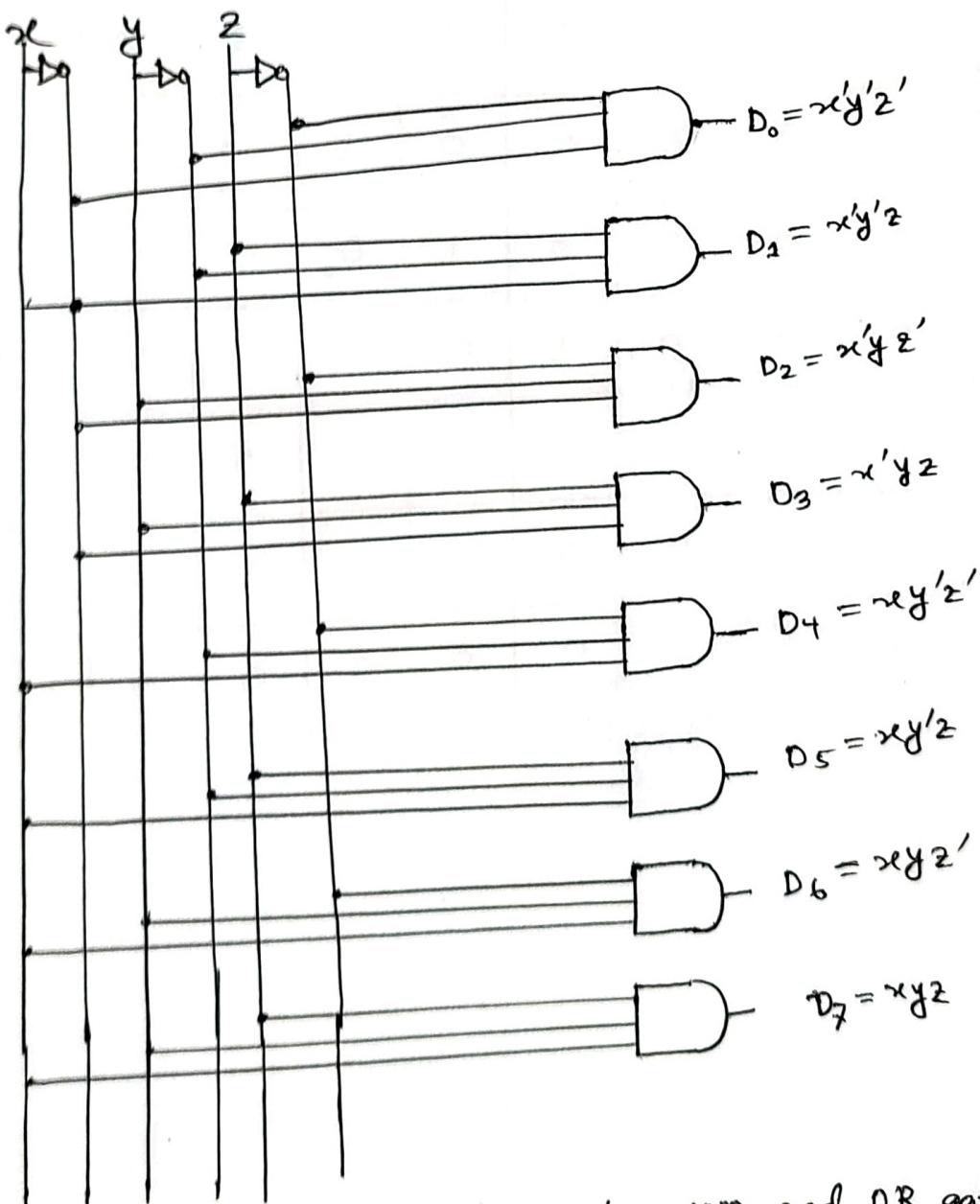


Fig: 3-to-8 line decoder using AND and OR gate.

Line Decoder

## Truth Table for Decoder

Input		Output							
$y$	$x$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
0	0	1	0	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	0	0	0	0	0	0	1

encoder,  
truth  
table  
equal to  
value  
invalent  
in the

perly.

(ii) Experimental Data: Truth table for 3-to-8 line decoder

Table:

Input			Output							
x	y	z	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

(iii) Result and Discussion: For 3-to-8 line decoder, we observe that, the output variable are mutually exclusive because only one output can be equal to 1 at any one time. The output line whose value is equal to 1 represents the minterms equivalent to the binary number presently available in the input lines.

(iv) Precaution:

- (i) All the connection should be made properly.
- (ii) Ie should not be reversed.