Lab Report - 05

Course No: 206

Course Title: Digital Logic Design

Submitted To:

Iffat Tamanna

Dept: CSE

Submitted By:

Md: Zobayer Hasan Nayem

Id: 19202103274

Section: 07

Dept: CSE

Lab Report → 05

ID: 19202103274

Name of Experiment: Construct and test combination Logic circuit parity Generator.

Equipment:

1. 8 input OR gate
2. 4 input AND gate
3. Not gate
4. Logic probe
5. Logic State

Description: A parity Bit is a Bit added to a string of Binary code. parity bit are a simple form of error detecting code. parity Bit are generally applied to the smallest units of a communication protocol, typically 8-bit octets (bytes). although they can also be applied separately to an entire massage string of bits. The parity bit ensure that

the total number of 1-bits in the string is even or odd. There are two various of parity bit: even parity bit and ODD parity bit.

Even parity: parity bit are added to transmitted massage to ensure that the number of bit with a value of one in aset of bits add up to even or odd numbers. Even and odd parities are the two variants of parity checking modes. even parity can be more clearly explained by means of an example Consider the transmitted massage 1010001, which has three ones in it. This turned into even parity by adding a one, making the sequence 1 1010001, so that there are four ones (an even number). If the transmitted massage has the form 1101001, which is already an even number, a zero is added to sustain the even parity.

Even_parity

This is the truth table of Even_parity.

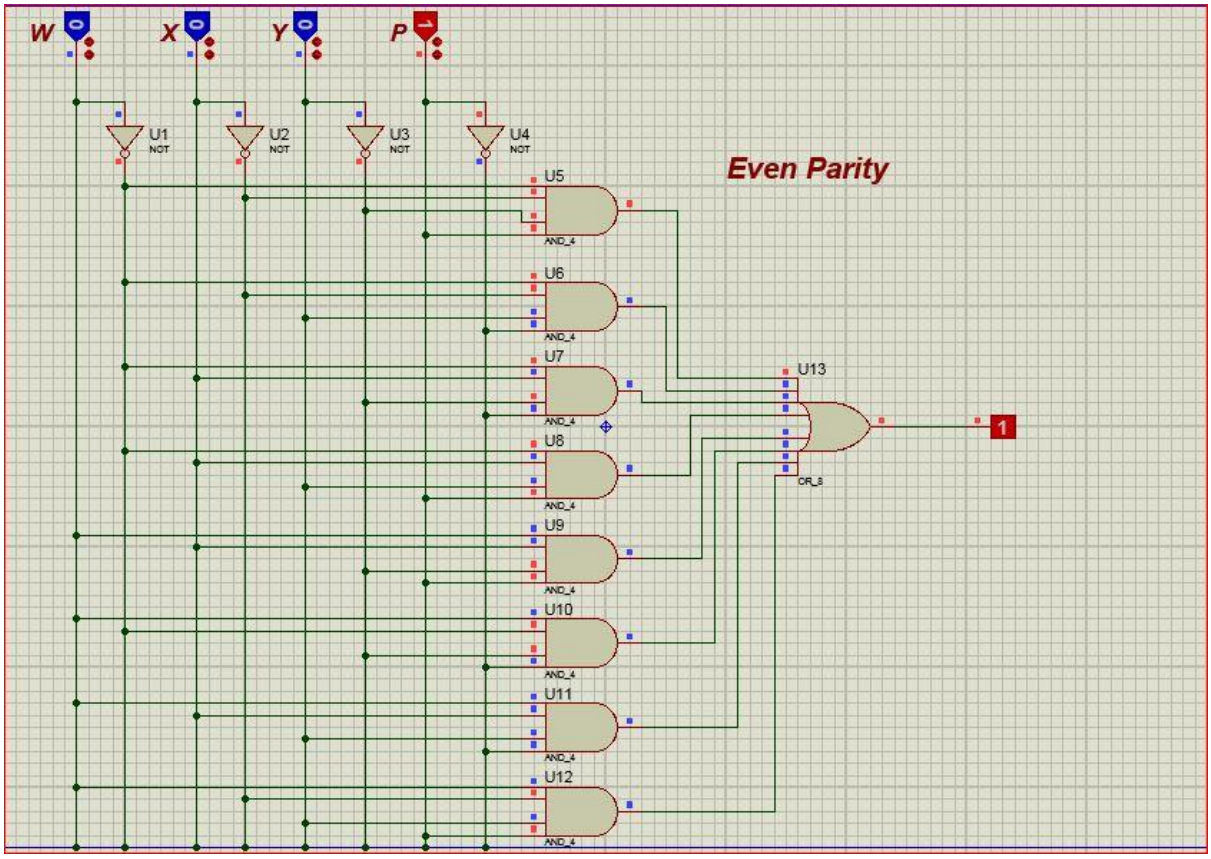
w	x	y	P	Even parity
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Ke-map

wx \ yp	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$f = w'x'y'p + w'x'yp' + w'xy'p' + w'xyp + wxy'p + wxy'p' + wxyp' + wx'yp$$

will Design the circuit in proteus Software.


W
X
Y
P
U1
NOT
U2
NOT
U3
NOT
U4
NOT
Even Parity
U5
AND_4
U6
AND_4
U7
AND_4
U8
AND_4
U9
AND_4
U10
AND_4
U11
AND_4
U12
AND_4
U13
OR_8
1

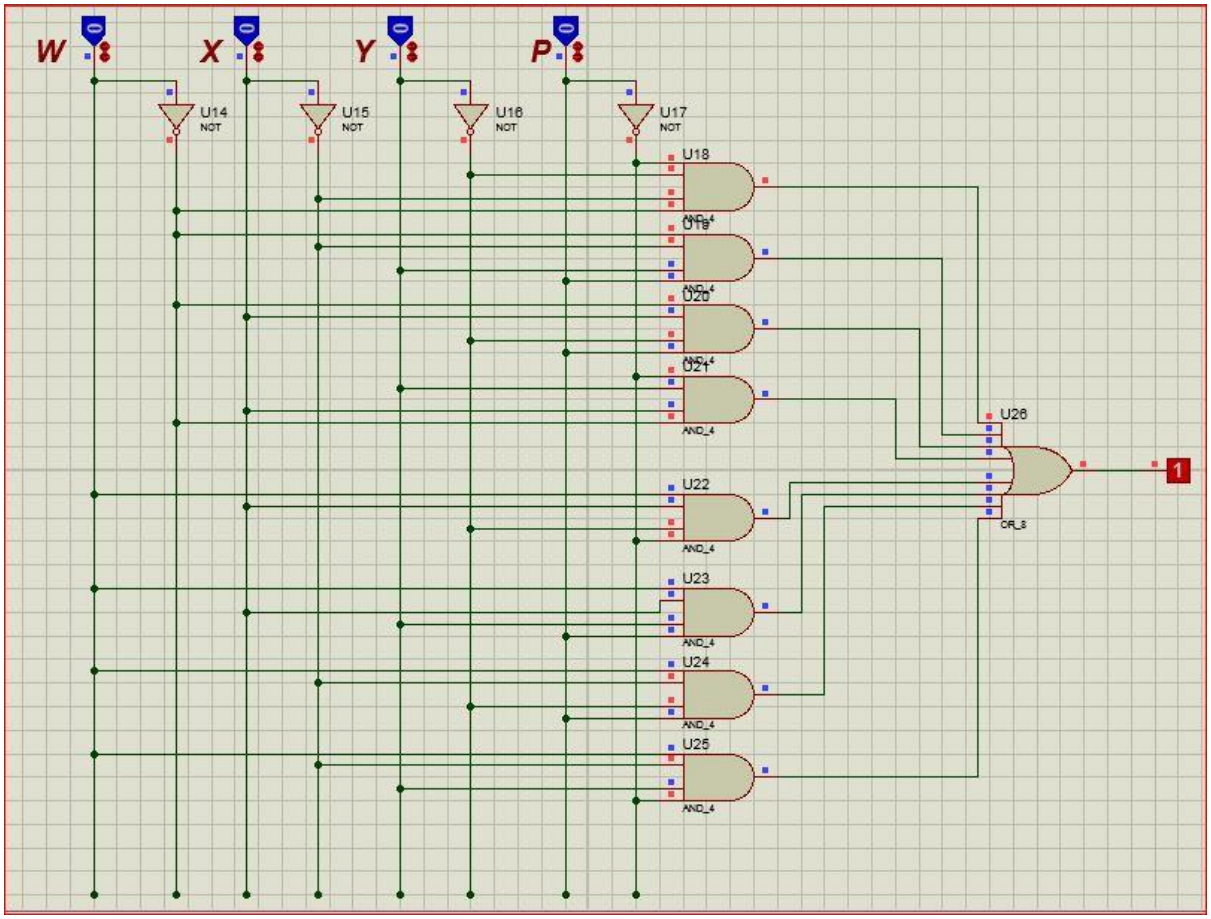
Odd parity: odd parity can be more clearly explained through an example. Consider the transmitted message 1010001, which has three ones in it. This is turned into odd parity by adding a zero, making the sequence 0 1010001. Thus, the total number of ones remain at three, an odd number.

if the transmitted massage has the form 1101001, which has four ones in it, this can be turned into odd parity by adding a one, making the sequence 1 1101001.

K-map for odd parity

wx \ yp	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

$$F = w'x'y'p' + w'x'yp + w'xy'p + w'xyp' + wxy'p' + wxyp + wx'y'p + wx'yp'$$


W
X
Y
P
U14
NOT
U15
NOT
U16
NOT
U17
NOT
U18
U19
U20
U21
AND_4
U22
U23
U24
U25
U26
OR_4
1

Conclusion:

① We have learnt how to implement Even_parity & odd parity.

② We also know how to work even & odd parity.

③ We also know odd parity wants to transmit 1001 and add parity bit and sends 10011.

④ We also know even_parity wants to transmit 1001 and add parity bit and sends 10010

⑤ We also know how to create function of parity Bit.

Name of Experiment : Design of code converters like BCD to Excess-3.

Describe : To understand the process of converting BCD to excess-3, it is requried to have knowledge of numbers system and numbers base conversion. The excess-3 binary code is an example of a self-complementary BCD code. A self-complementary binary code is a code which is always complimented in itself. By replacing the bit 1 to 0 and 0 to 1 of a numbers. The sum of all the 1's complement and the binary numbers of a decimal is equal to the binary numbers of decimal 9.

The process of BCD to excess-3. The Excess-3 code can be calculating added 3 (0011) to each BCD code.

Truth table

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	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

we will use the K-map method to design and make function for the convertion of BCD to Excess-3.

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

$$W = A + BD + BC$$

AB \ CD	00	01	11	10
00		1	1	1
01	1	0	0	0
11	x	x	x	x
10	0	1	x	x

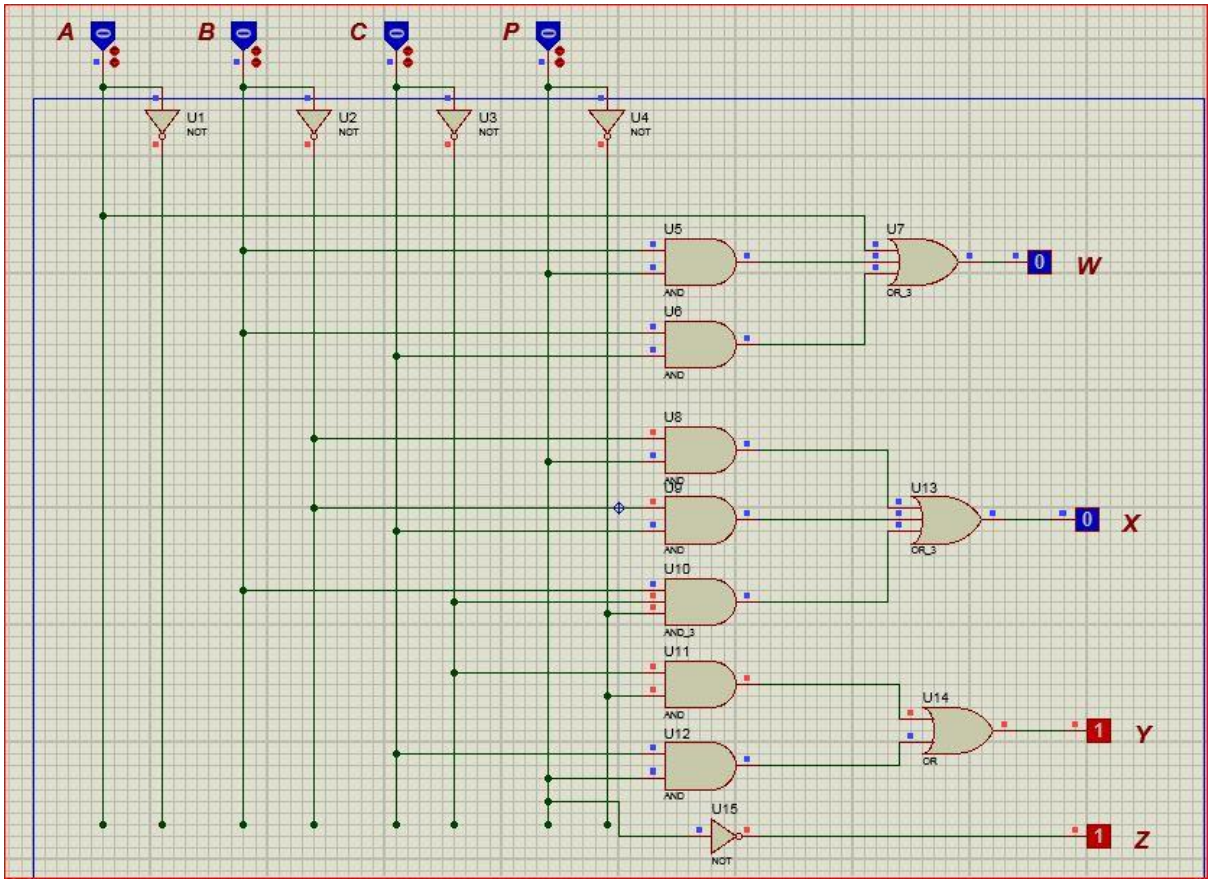
$$X = B'D + B'C + BC'D'$$

AB \ CD	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	x	x	x	x
10	1	0	x	x

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

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	x	x	x	x
10	1	0	x	x

$Z = D'$


A
B
C
P
U1
NOT
U2
NOT
U3
NOT
U4
NOT
U5
AND
U6
AND
U7
OR_3
0
W
U8
AND
U9
AND
U13
OR_3
0
X
U10
AND_3
U11
AND
U12
AND
U14
OR
1
Y
U15
NOT
1
Z

Conclusion:

(i) we have learn't how to convert BCD to Excess-3 code.

(ii) we have learn't that how to implement code.

(iii) we also learn't that how to use k-map using to the BCD to Excess-03 code.