CSE260 Lab Report

Experiment Name: PARITY GENERATOR AND CHECKER

Submitted by

Name: SHAIANE PREMA BAROI

ID: 19241019

Section: 05

Date: 17 / 11 / 2020

1> Name of the Experiment: Parity Generator and Checker

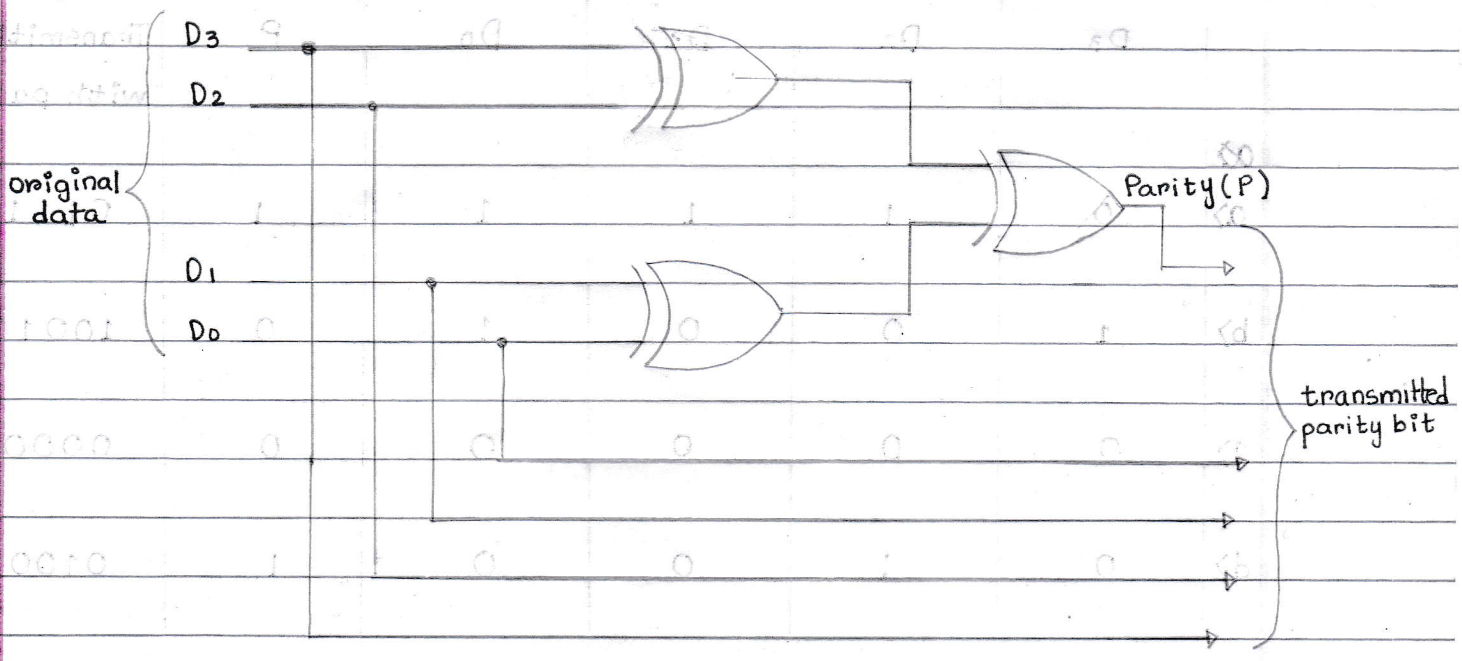
2> Objective:

- to design and implement an Even Parity Generator and Even Parity Checker using XOR gates (IC-7486)

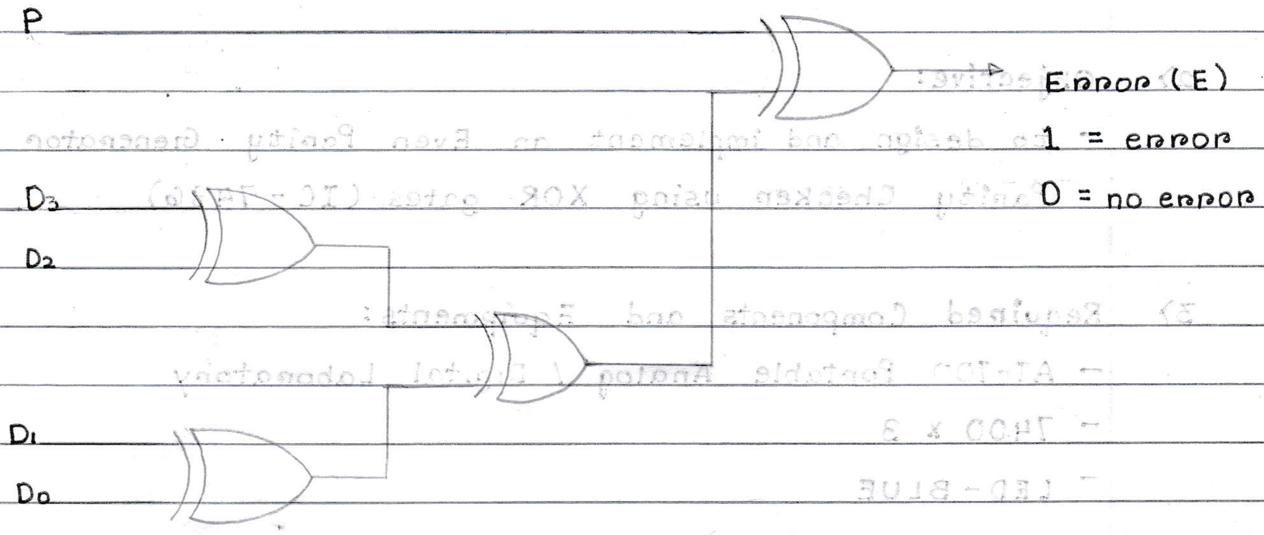
3> Required Components and Equipments:

- AT-700 Portable Analog / Digital Laboratory
- 7400 x 3
- LED-BLUE
- LOGICPROBE (BIG)
- LOGICSTATE
- XOR
- GROUND
- WIRES

4> Experimental Setup



Even Parity Generator



5) Results in Tabulated Form:

Parity Generator's Output

	D3	D2	D1	D0	P	Transmitted data with parity bit
a)	0	1	1	1	1	01111
b)	1	0	0	1	0	10010
c)	0	0	0	0	0	00000
d)	0	1	0	0	1	01001

Parity Checker's Output

P	D_3	D_2	D_1	D_0	E	Status
0	1	0	1	0	0	no error
1	1	1	1	0	0	no error
1	1	1	1	1	1	error
1	0	0	0	0	1	error

6> Discussions

In this experiment, we had learnt about the Parity Generator and Parity Checker. The Parity Checker generates an extra bit according to the data which is then transmitted to the receiver. The data along with the parity bit is then checked by the Parity Checker which validates if the transmission has been done properly or not. In the first part of the experiment, we observed the process which takes place on the sender's end. We constructed an Even Parity using a combination of three XOR gates and used the different values as inputs (according to the question) and obtained the corresponding bits. We noticed how the parity bit changed each time the input was changed in order to ensure that there was an even number of 1's in the entire series of bits. In the second part of the experiment, we saw the process which takes place on the receiver's end. Once a series of bit is received, the Parity Checker checks if the number of 1's is even or odd as per the Parity used. Since, we used Even Parity in our experiment, the Parity Checker checked if the number of 1's received were even or not. If they were not even, an error was generat~~or~~ed and otherwise, no error. At this point, we realised that there were some limitations in the Parity Checker: If a series of data, such that 1001 is being transmitted, the parity bit would be 0. Hence, the transmitted bits would be 01001. Then, if by any chance, the data gets altered to 1001, in which case, the parity bit would still be 0 and the Parity Checker wouldn't produce any error even though the data is corrupt. Hence, the Parity Checker has these limitations.