* Digital Electronics and Logic Design (DELD) - Practical Number - 7

Name:- Kaustubh Shrikant Kabra
Class:- Second Year Engineering
Div:- A Roll Number:-
Batch:-
Department:- Computer Department
College:- AISSMS's IOIT

Title:-
Parity Generator and Checker.

Aim:-
Design and Implement parity generator and checker using EX-OR gate.

Objective:-
Learn even/odd parity generator/checker using EX-OR gate.

Theory:-
Parity - A term used to specify the member of one's in a digital word as odd/even.

Even Parity Generator -
Will produce a logic 1 at its output if the data word contain an odd number of ones. If the data contain an even number of ones then the output of the parity generator will be low.

Parity Checker -
At the reciving end a logic circuit is used to check the parity

of receiving information, and determines whether error is included in the message or not.

Even bit Parity code-
Total number of ones in parity code is even.

Odd bit Parity code-
Total number of one's in parity code word is odd.

Truth Table :-

Ⓐ Parity Generator :-

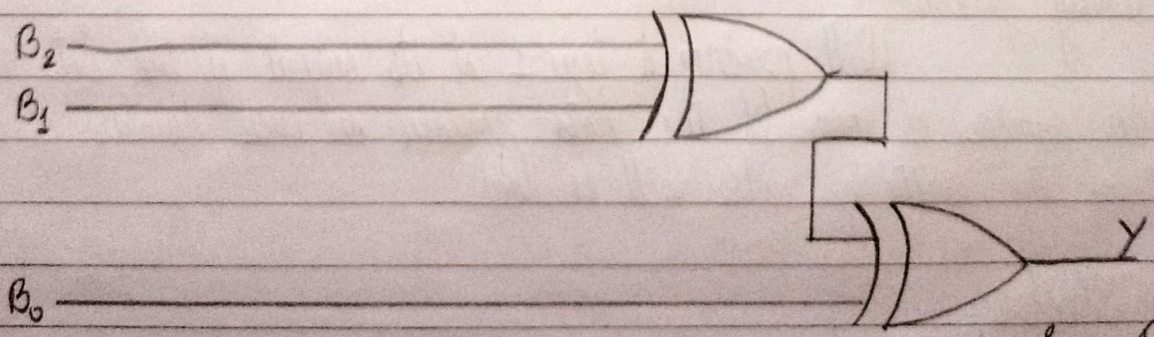
① Even Parity Generator -

Input			Output
B_2	B_1	B_0	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Expression for output using k-map

B_2 \ B_1B_0	00	01	11	10
0	0	[1]	0	[1]
1	[1]	0	[1]	0

$f = B_2 \text{ (EX-OR) } B_1 \text{ (EX-OR) } B_0$



- Even Parity Generator

OUTPUT SECTION
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
VCC
Even Parity Generator
Power
74LS86
IC BASE 2
IC BASE 3
IC BASE 4
IC BASE 5
This simulator is protected by ©DeldSim Community
GND
HIGH
LOW
GENERATE PULSE
10 5 1 0.5
INPUT SECTION

② Odd Parity Generator :-

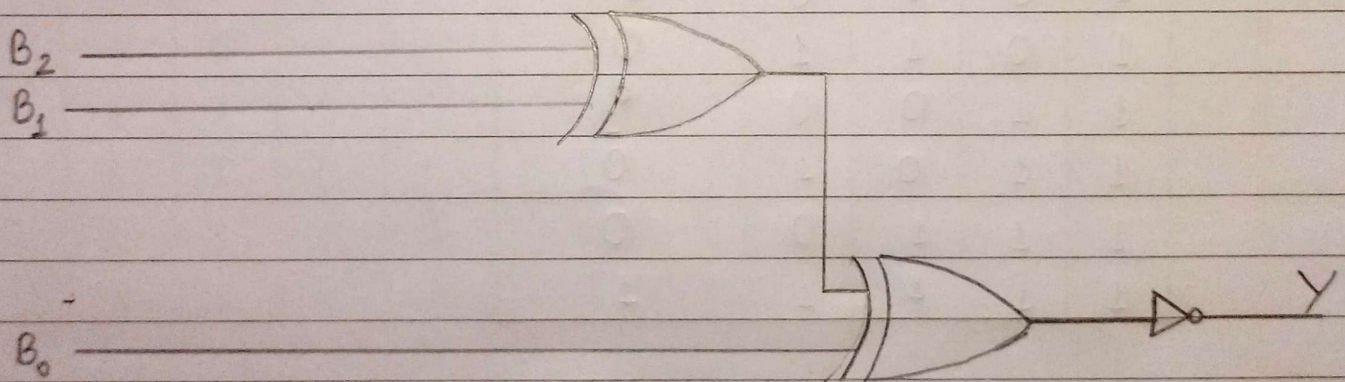
Input			Output
B_2	B_1	B_0	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Expression using k-map.

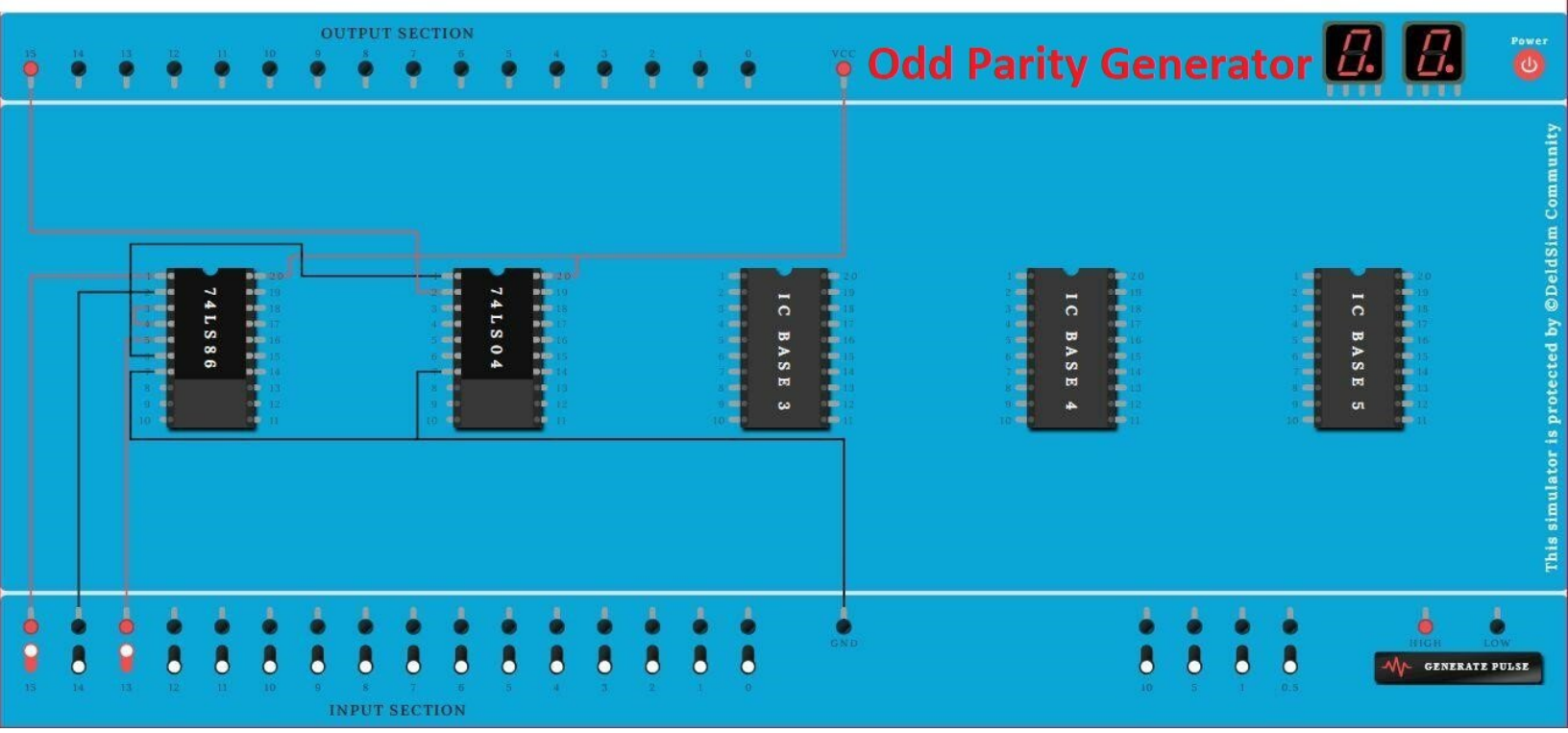
B_2 \ B_1B_0	00	01	11	10
0	1	0	1	0
1	0	1	0	1

$$Y = B_2 (EX\text{-}NOR) B_1 (EX\text{-}NOR) B_0$$

$$= B_2 (EX\text{-}OR) B_1 (EX\text{-}OR) B_0$$



- Odd Parity Generator


OUTPUT SECTION
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
VCC
Odd Parity Generator
Power
74LS86
74LS04
IC BASE 3
IC BASE 4
IC BASE 5
This simulator is protected by ©DeldSim Community
GND
HIGH
LOW
GENERATE PULSE
10 5 1 0.5
INPUT SECTION

(B) Parity Checker :-

[0 → Error
1 → Not Error]

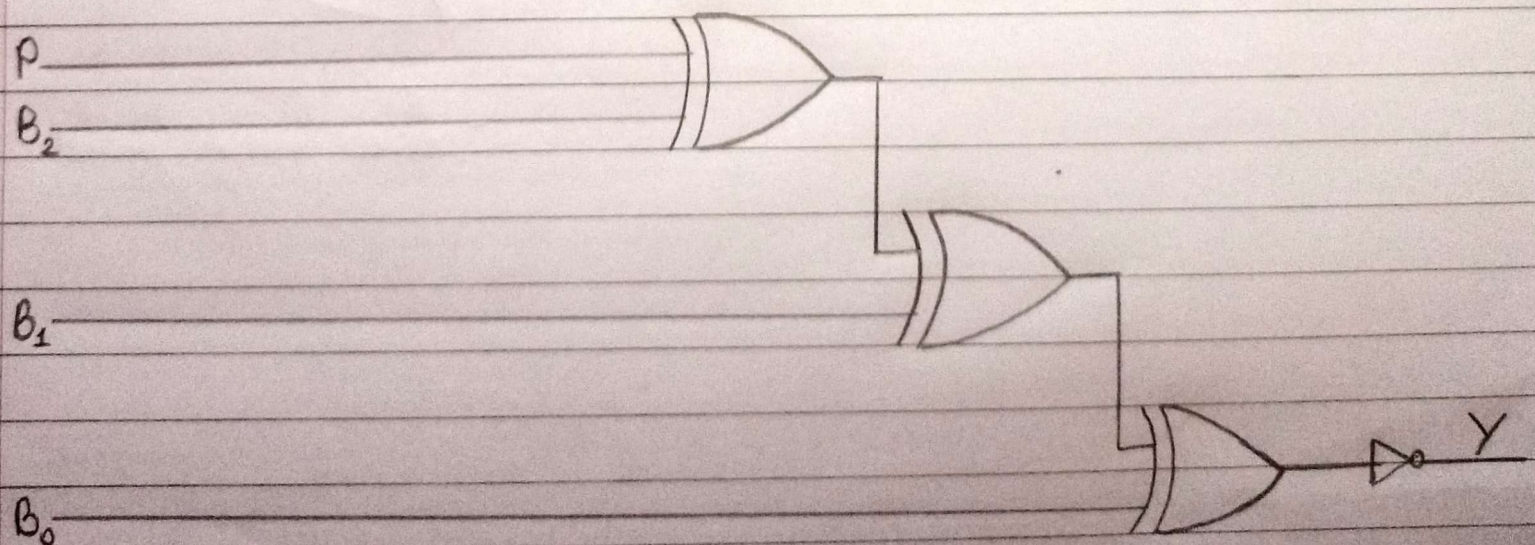
(1) Odd Parity Checker :-

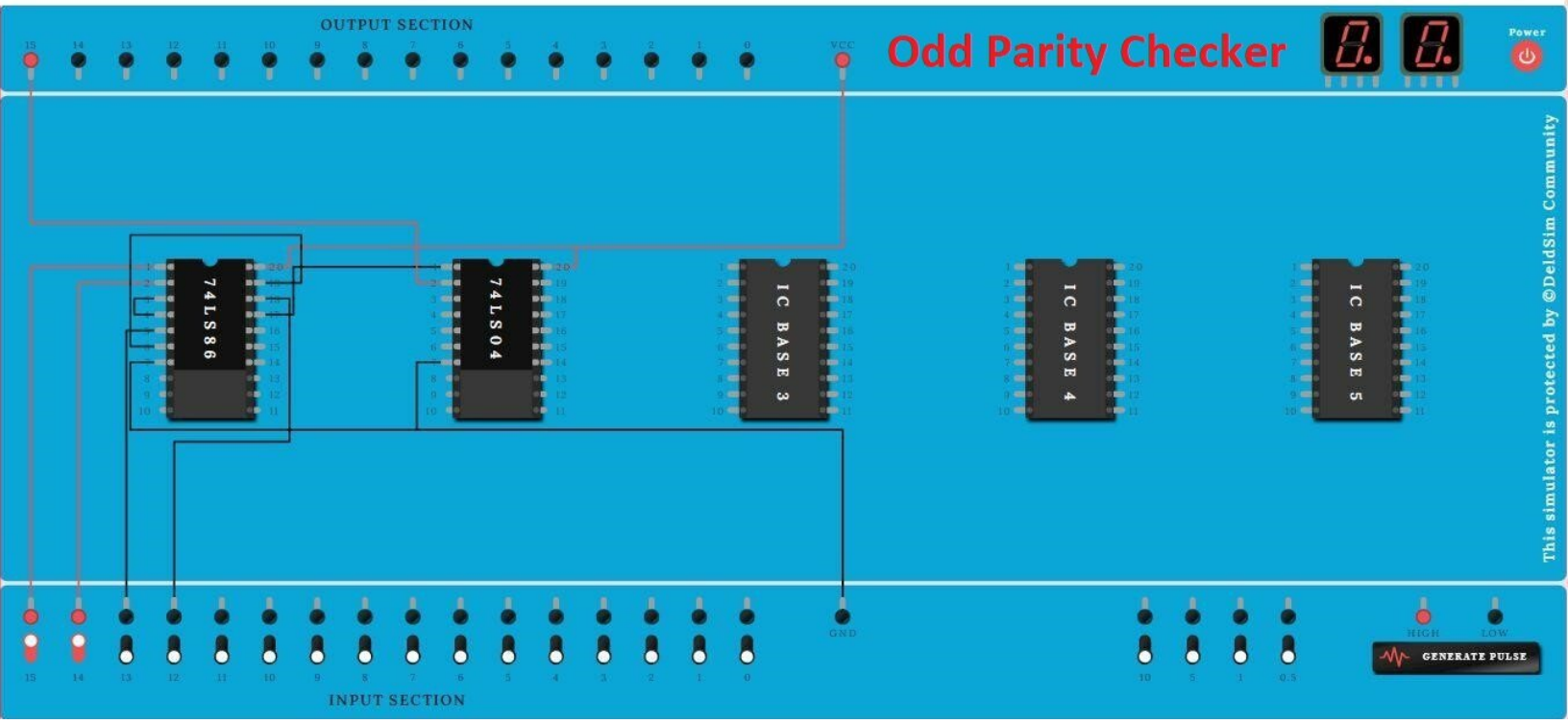
Input				Output
Parity(P)	B_2	B_1	B_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Expression using k-map

PB_2 \ B_1B_0	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

$Y = P\,(EX\text{-}OR)\,B_2\,(EX\text{-}OR)\,B_1\,(EX\text{-}OR)\,B_0$




OUTPUT SECTION
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
VCC
Odd Parity Checker
Power
74LS86
74LS04
IC BASE 3
IC BASE 4
IC BASE 5
This simulator is protected by ©DeldSim Community
GND
HIGH
LOW
GENERATE PULSE
10 5 1 0.5
INPUT SECTION

② Even Parity Checker -

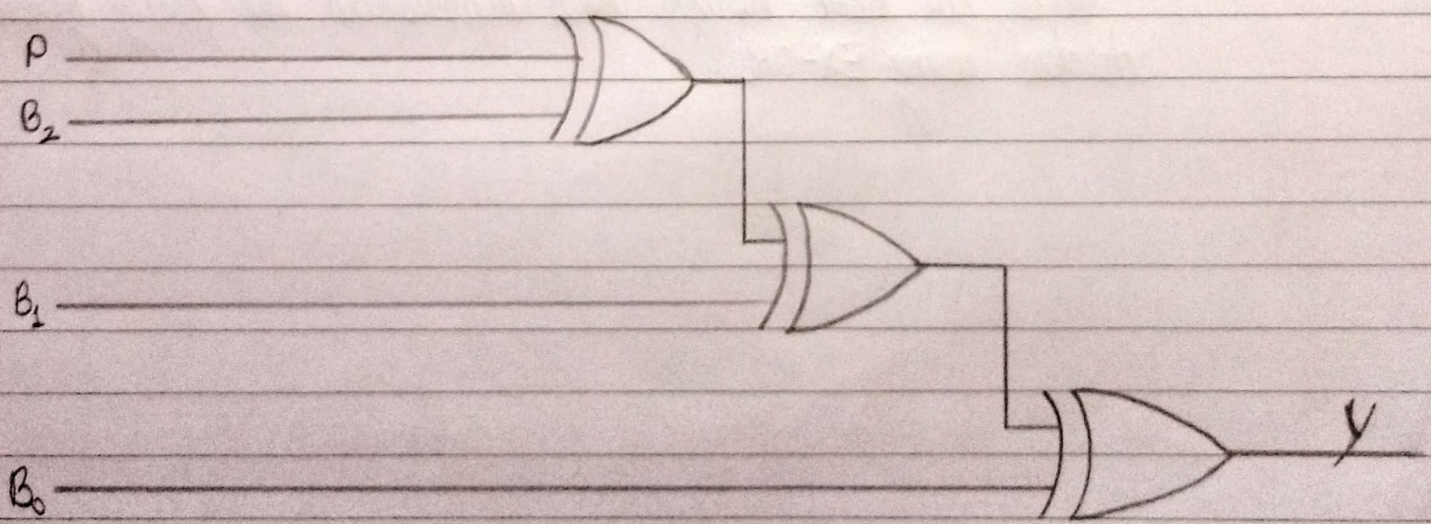
[0 → Error
1 → Not Error]

Input				Output
Parity (P)	B_2	B_1	B_0	Y
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

Expression using k-map.

PB_2 \ B_1B_0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

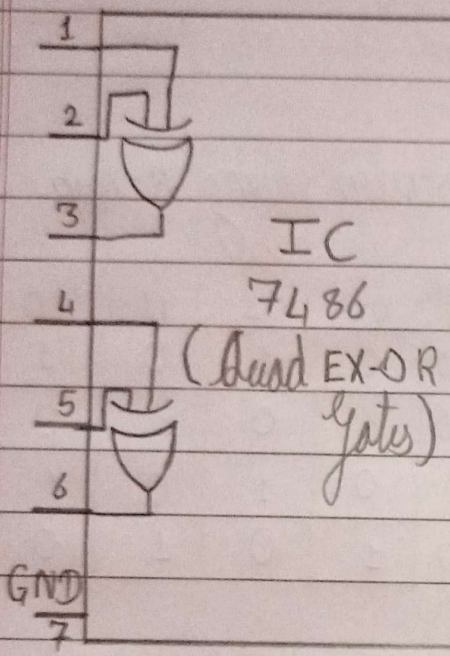
$Y = P \text{ (EX-OR) } B_2 \text{ (EX-OR) } B_1 \text{ (EX-OR) } B_0$



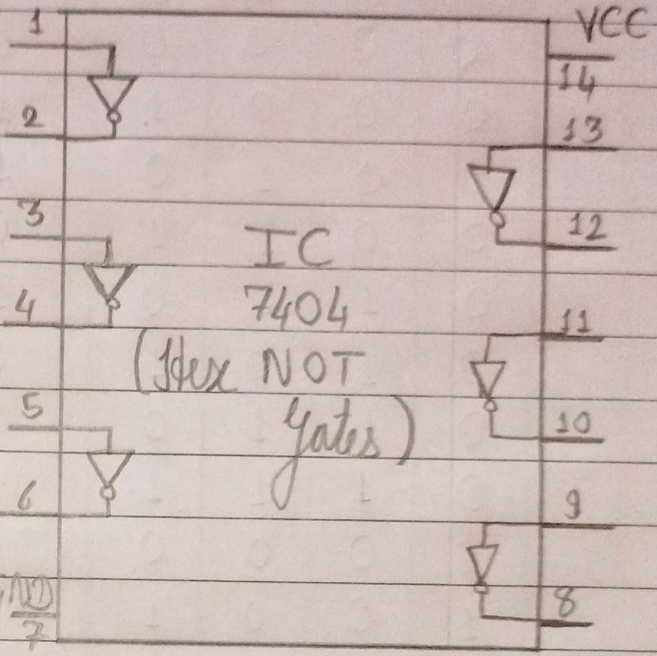
OUTPUT SECTION
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
VCC
Even Parity Checker
Power
74LS86
IC BASE 2
IC BASE 3
IC BASE 4
IC BASE 5
This simulator is protected by ©DeldSim Community
GND
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
INPUT SECTION
10 5 1 0.5
HIGH
LOW
GENERATE PULSE

* IC's Used:-

1) EX-OR Gate - (IC 7486)



2) NOT Gate - (IC 7404)



Outcomes:-

Thus we study parity generator and checker and their truth tables.

Conclusion:-

Hence, we have design and implemented the parity generator and checker using EX-OR.