EAST WEST UNIVERSITY

Department of CSE

PROJECT REPORT

Course Code: CSE345 Course Name: Digital Logic Design	
Project name: 4-bit Binary to Even Parity Code Converter	
Summer 2021	
Submitted By	Submitted To
Hasib Ar Rafiul Fahim ID: 2019-1-60-036	Musharrat Khan Senior Lecturer Department of Computer Science and Engineering, East West University, Bangladesh.

1.Problem statement:

This project is about a 4-bit code converter for even parity converts the 4-bit input to 5-bit output so that even parity is ensured. In even parity, parity bit is '0' if there are even number of 1s in inputs and the parity bit is '1' if there are odd number of 1s in inputs. Here 'S' is the parity bit.

2.Design Details:

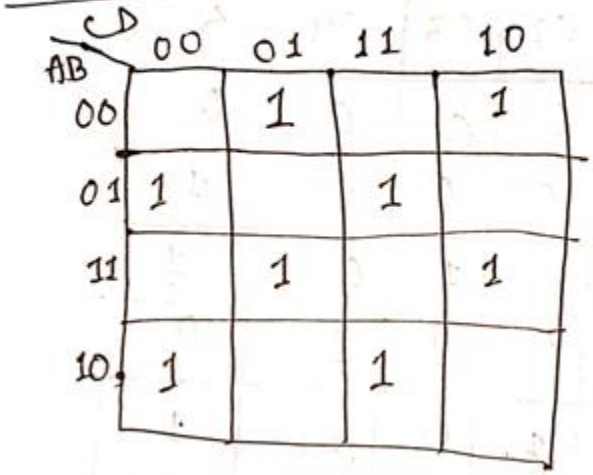
Truth Table:-

Inputs				Outputs				
A	B	C	D	O1	O2	O3	O4	S
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	0	1
0	0	1	1	0	0	1	1	0
0	1	0	0	0	1	0	0	1
0	1	0	1	0	1	0	1	0
0	1	1	0	0	1	1	0	0
0	1	1	1	0	1	1	1	1
1	0	0	0	1	0	0	0	1
1	0	0	1	1	0	0	1	0
1	0	1	0	1	0	1	0	0
1	0	1	1	1	0	1	1	1
1	1	0	0	1	1	0	0	0
1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1	0

From above table:

O1=A, O2=B, O3=C, O4=D.

K-map of S:-



S=A’B’C’D + A’B’C D’ + A’B C’D’ + A’B C D + A B’C’D’ + A B’C D + A B C’D + A B C D’

= A’B’ (C’D + C D’) + A’B (C’D’ + C D) + A B (C’D + C D’) + A B’ (C’D’ + C D)

= A’B’ (C xor D) + A’B (C xnor D) + A B (C xor D) + A B’ (C xnor D)

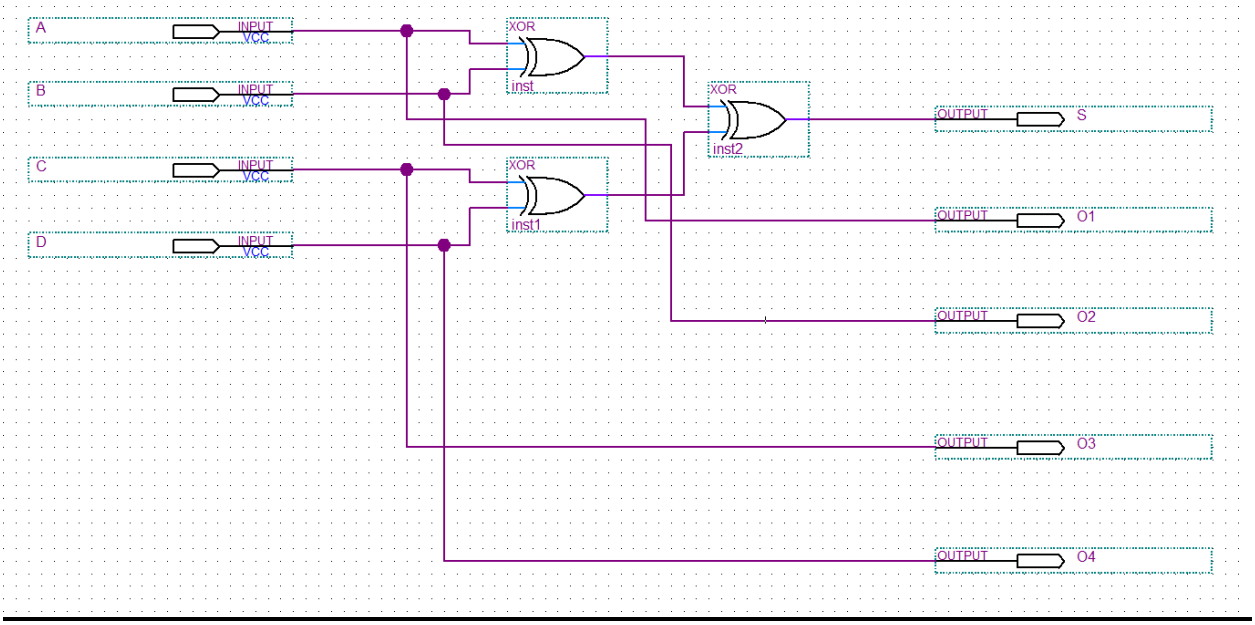
= (A’B’ + A B) (C xor D) + (A’B + A B’) (C xnor D)

= (A xnor B) (C xor D) + (A xor B) (C xnor D)

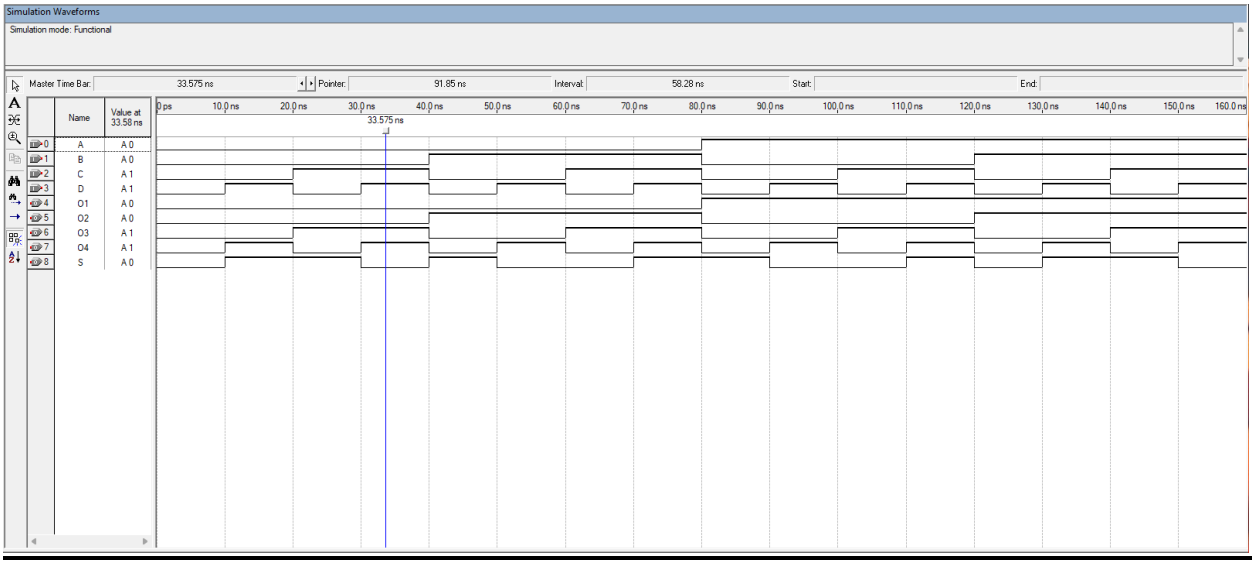
= (A xor B)’ (C xor D) + (A xor B) (C xor D)’

= A xor B xor C xor D.

3. Circuit Diagram:



Simulation result:

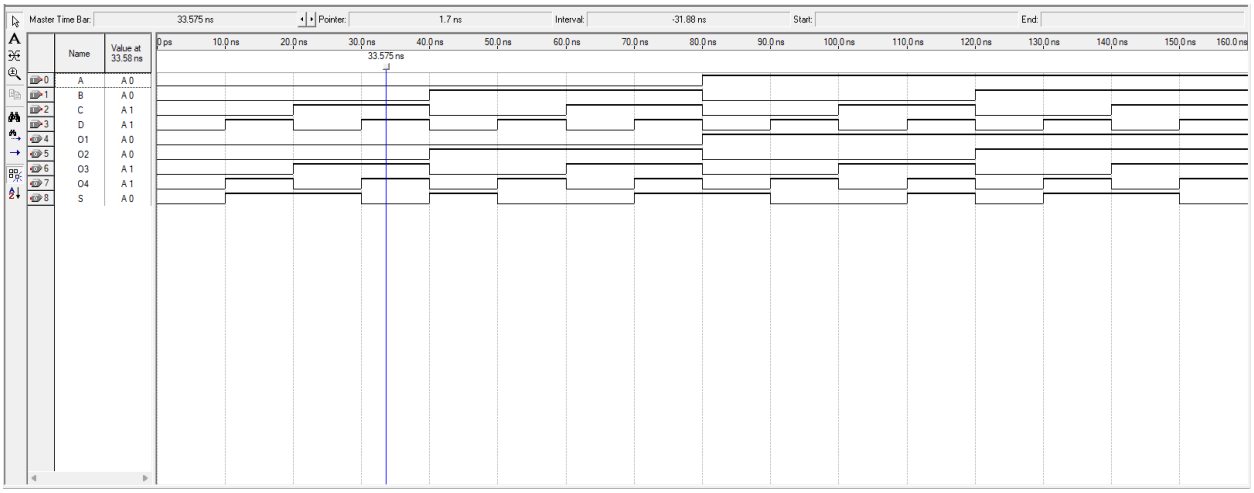


4. Structural and Behavioral Verilog Code and Simulation Results:

Structural verilog code:

```
module even_parity_struct(input A,B,C,D,
                          output O1,O2,O3,O4,S);
            wire w1,w2;
            assign  O1=A,
                    O2=B,
                    O3=C,
                    O4=D;
            xor g1(w1,A,B),
                g2(w2,C,D),
                g3(S,w1,w2);
endmodule
```

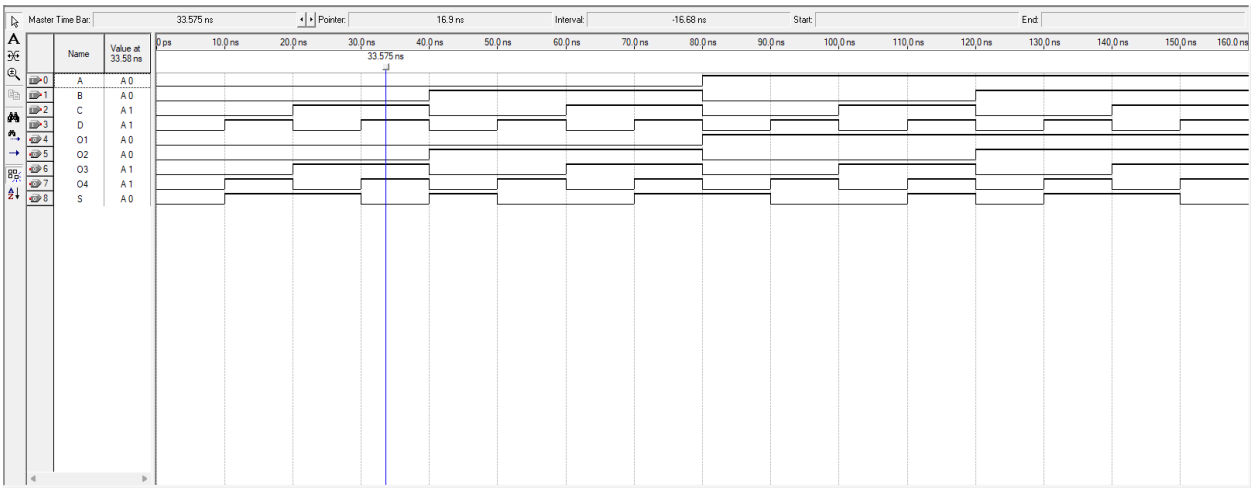
Simulation result:



Procedural model:

```
module even_parity_procedural(input A,B,C,D, output reg O1,O2,O3,O4,S);
        always@(A,B,C,D)begin
                O1=A;
                O2=B;
                O3=C;
                O4=D;
                S=0;
        if(~A&~B&~C&D)S=1;
        if(~A&~B&C&~D)S=1;
        if(~A&B&~C&~D)S=1;
        if(~A&B&C&D)S=1;
        if(A&B&~C&D)S=1;
        if(A&B&C&~D)S=1;
        if(A&~B&~C&~D)S=1;
        if(A&~B&C&D)S=1;
        end
endmodule
```

Simulation result:



Continuous assign statement:

```
module even_parity_continuosassign(input A,B,C,D,
                    output O1,O2,O3,O4,S);
                assign O1=A, O2=B, O3=C, O4=D,
                S=(~A&~B&~C&D)|(~A&~B&C&~D)|(~A&B&~C&~D)|(~A&B&C&D)|
                    (A&B&~C&D)|(A&B&C&~D)|(A&~B&~C&~D)|(A&~B&C&D);
endmodule
```

Simulation result: