

**Course Title :** Digital Logic Design

**Course Code :** CSE345

**Section :** (03)

**Semester :** Summer 2021

**Project Report**

**Project Name: 4-bit Binary to Even Parity Code Converter.**

**Submitted To**

Musharrat Khan

Senior Lecturer

Department of Computer Science and Engineering

East West University

**Submitted By**

Name: Md. Jannatul Haq

ID: 2018-1-60-224

**Submission Date**

12/09/2021

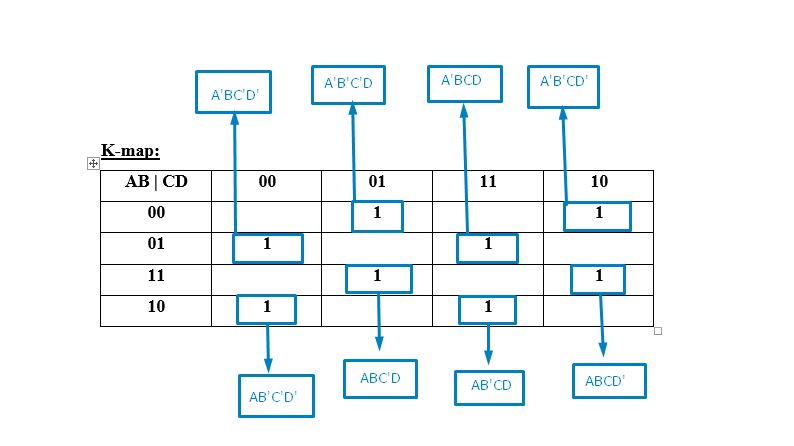
**1. Problem Statement**

A 4-bit code converter for even parity converts the 4-bit input to 5-bit output so that even parity is ensured.

**2. Design Details**

**Truth Table:**

|  |  |
| --- | --- |
| **ABCD** | **F** |
| **0000** | **0** |
| **0001** | **1** |
| **0010** | **1** |
| **0011** | **0** |
| **0100** | **1** |
| **0101** | **0** |
| **0110** | **0** |
| **0111** | **1** |
| **1000** | **1** |
| **1001** | **0** |
| **1010** | **0** |
| **1011** | **1** |
| **1100** | **0** |
| **1101** | **1** |
| **1110** | **1** |
| **1111** | **0** |



**So,**

**F = AꞌBꞌCꞌD + AꞌBꞌCDꞌ + AꞌBCꞌDꞌ + AꞌBCD + ABCꞌD + ABCDꞌ + ABꞌCꞌDꞌ + ABꞌCD**

**= AꞌBꞌ (CꞌD + CDꞌ) + AꞌB (CꞌDꞌ + CD) + AB (CꞌD + CDꞌ) + ABꞌ (CꞌDꞌ + CD)**

**= AꞌBꞌ (C XOR D) + AꞌB (C XNOR D) + AB (C XOR D) + ABꞌ (C XNOR D)**

**= (C XOR D) (AꞌBꞌ + AB) + (C XNOR D) (AꞌB + ABꞌ)**

**= (C XOR D) (A XNOR B) + (C XNOR D) (A XOR B)**

**= (A XNOR B) (C XOR D) + (A XOR B) (C XNOR D)**

**= (A XOR B)ꞌ (C XOR D) + (A XOR B) (C XOR D)ꞌ**

**Assuming,**

**x = (A XOR B),**

**y = (C XOR D)**

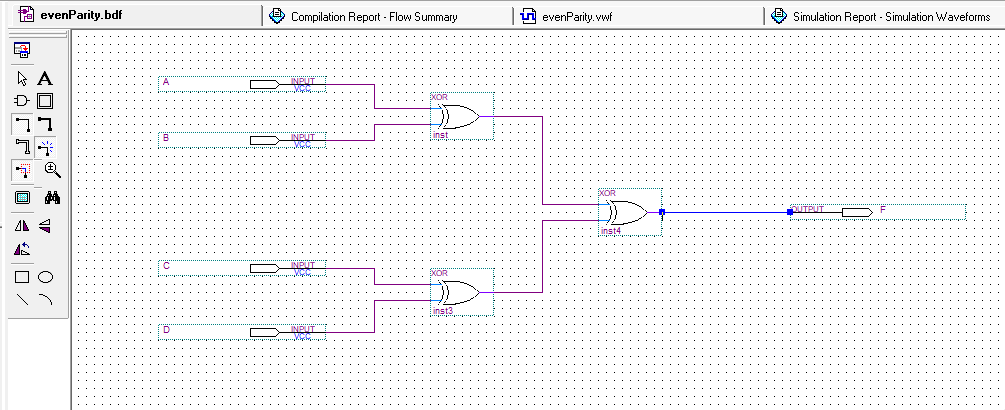
**So,**

**F = xꞌy + xyꞌ**

**= x XOR y**

**= (A XOR B) XOR (C XOR D)**

**3. Circuit Diagram:**



**4. Structural and Behavioral Verilog Code and Simulation Results:**

**Structural Verilog Code:**

**module evenParity(input A,B,C,D, output F);**

**wire w1,w2;**

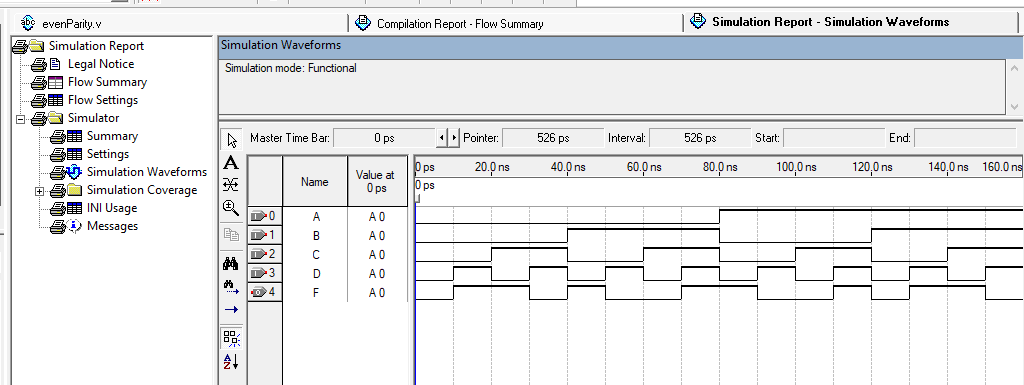
**xor g1(w1,A,B),**

**g2(w2,C,D);**

**xor g3(F,w1,w2);**

**endmodule**

**Simulation Result:**



**Procedural Model:**

**module evenParity(input A, B, C, D, output reg F);**

**always @(A, B, C, D)begin**

**F=0;**

**if(~A&~B&~C&D) F=1;**

**if(~A&~B&C&~D) F=1;**

**if(~A&B&~C&~D) F=1;**

**if(~A&B&C&D) F=1;**

**if(A&B&~C&D) F=1;**

**if(A&B&C&~D) F=1;**

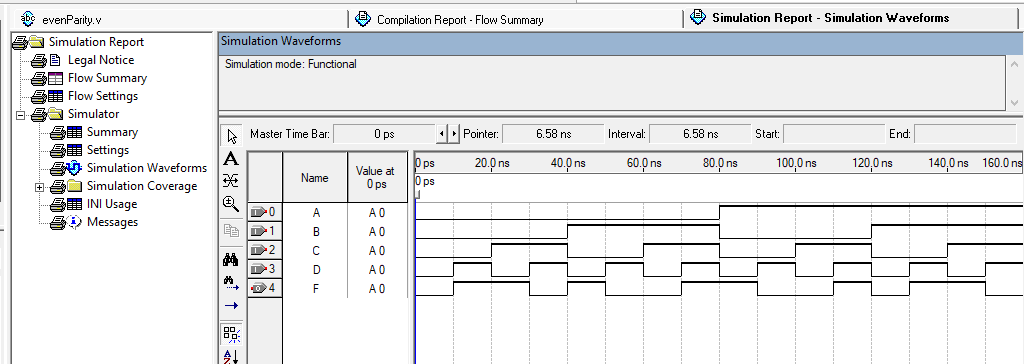
**if(A&~B&~C&~D) F=1;**

**if(A&~B&C&D) F=1;**

**end**

**endmodule**

**Simulation Result:**



**Continuous Assign Statement:**

**module evenParity(input A,B,C,D, output F);**

**assign**

**F=(~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:**

