**VERILOG IMPLEMENTATION OF REVERSIBLE LOGIC GATE**

**A PROJECT REPORT**

***Submitted by***

**SANTHIYA P 710021106317**

**ABINAYA SARASWATHI S 710021106316**

**PRITHIKA K 710021106701**

***In partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**EC3552 VLSI AND CHIP DESIGN**



**ANNA UNIVERSITY REGIONAL CAMPUS, COIMBATORE**

**ANNA UNIVERSITY : CHENNAI**

**NOVEMBER 2023**

**BONAFIDE CERTIFICATE**

Project report titled **“VERILOG IMPLEMENTATION OF REVERSIBLE LOGIC GATES ”** is the bonafide work of **“SANTHIYA P (710021106317), ABINAYASARASWATHI S (710021106316) , PRITHIKA K (710021106701)”**who carried out the project work under my supervision.

**Dr. V.R.Vijaykumar M.E, Ph.D., Ms. S.Manju M.E.,**

Head of the Department, Faculty of the Department & Mentor,

Associate Professor,

Department of ECE, Department of ECE,

Anna University Regional Campus, Anna University Regional Coimbatore – 641 046.

**ACKNOWLEDGEMENT**

First and foremost, we place this project work on the feet of **GOD ALMIGHTY** who is the power of strength in each step of progress towards the successful completion of the project.

We owe wholehearted sense of reverence and gratitude to our Guide

**Ms. S. MANJU M.E.,** and Associate Professor and Head of the Department of ECE **Dr. V.R. VIJAYKUMAR M.E, Ph.D.,** for their efficient and excellent guidance, inspiring discussions insightful and timely encouragements for the successful completion of the project.

I thank **Dr. M. SARAVANAKUMAR MBA., Ph.D.,** Dean Regional Campus, Anna University Regional Campus, Coimbatore for his great support with blessings. I also extend my heartfelt thanks to all Faculties and staff of ECE department who have rendered their valuable help in making this project successful.

**SANTHIYA P** - **710021106317**

**ABINAYA SARASWATHI S** - **710021106316**

**PRITHIKA K**  - **710021106701**

**ABSTRACT**

Reversible logic is one of the most important issues at the moment and has a wide range of applications, such as low power CMOS, quantum computing, nanotechnology, cryptography, optical computing, DNA computing, and -digital signal processing (DSP), quantum Dot automata for mobile, communication, computer graphics. It is not possible to detect quantum computing without the implementation of a postponed brain operation. The main objectives of design are logical thinking to reduce quantum costs, circuit depths and the amount of waste disposal. This paper provides basic logical retrospective gates, which in the construction of a highly sophisticated system with retractable circuits as part of the old and unable to perform the most complex operations using quantum computers. Reversible circuits form the basic building block of quantum computers as all quantum functions are reversed. This paper presents data related to older retractable gates found in books and assists research in the design of complex computer circuits using retractable gates.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ACKNOWLEDGEMENT** | 3 |
|  | **ABSTRACT** | 4 |
|  | **TABLE OF CONTENTS** | 4 |
|  | **LIST OF FIGURES** | 7 |
| **CHAPTER 1** | **INTRODUCTION** | 8 |
|  | 1.1 INTRODUCTION | 9 |
|  | 1.2 PROBLEM STATEMENT | 10 |
|  | 1.3 OBJECTIVE | 10 |
| **CHAPTER 2** | **LITERATURE SURVEY** |  |
|  | 2.1 INTRODUCTION | 11 |
|  | 2.2THEORITICAL OF REVERSIBLE COMPUTING | 11 |
|  | 2.3 DESIGN AND OPTIMIZATION OF REVERSIBLE LOGIC GATE | 11 |
|  | 2.4 CIRCUIT IMPLEMENTATION | 12 |
|  | 2.5 APPLICATION IN QUANTUM COMPUTING | 12 |
|  | 2.6 SUMMARY | 12 |
| **CHAPTER 3** | **EXISTING SYSTEM** |  |
|  | 3.1 INTRODUCTION | 13 |
|  | 3.2 EXISTING SYSTEM | 13 |
|  | 3.3 BLOCK DIAGRAM | 14 |
|  | 3.4 SUMMARY | 14 |
| **CHAPTER 4** | **PROPOSED SYSTEM** |  |
|  | 4.1 BASIC DEFINITION OF REVERSIBLE LOGIC GATE | 15 |
|  | 4.2 REVERSIBLE FUNCTION | 15 |
|  | 4.3 REVERSIBLE LOGIC GATE | 16 |
|  | 4.4 GARBAGE OUTPUT | 16 |
|  | 4.5 QUANTUM COST | 16 |
|  | 4.6 GATE LEVEL | 16 |
|  | 4.7 DESIGN CONSTRAINTS OF REVERSIBLE LOGIC GATE | 17 |
| **CHAPTER 5** | **SOFTWARE DESCRIPTION** |  |
|  | 5.1 REVERSIBLE LOGIC GATE | 17 |
|  | 5.2 METHODS 0F REVERSIBLE LOGIC GATE | 17 |
|  | 5.2.1 NOT GATE | 18 |
|  | 5.2.2 FEYNMAN GATE | 19 |
|  | 5.2.3 TOFFOLI GATE | 21 |
|  | 5.2.4 FREDKIN GATE | 22 |
|  | 5.2.5 PERES GATE | 24 |
| **HAPTER 6** | **HARDWARE DESCRIPTION** |  |
|  | 6.1 FPGA IMPLEMENTATION | 27 |
|  | CONCLUSION | 32 |
|  | REFERANCES | 33 |

**LIST OF FIGURES**

**FIGURE TITLE PAGE**

**NO NO**

1 INTRODUCTION 8

2 DESIGN AND OPTIMIZATION 11

OF REVERSIBLE LOGIC GATES

3 CIRCUIT IMPLEMENTATION 12

4 BLOCK DIAGRAM 14

5 METHODS OF REVERSIBLE 17

REVERSIBLE LOGIC GATES

6 SOFTWARE DESCRIPTION 18

7 HARDWARE DESCRIPTION 27

**CHAPTER 1**

**INTRODUCTION**

# INTRODUCTION

Energy dissipation is one of the major issues in present day technology. Energy dissipation due to information loss in high technology circuits and systems constructed using irreversible hardware was demonstrated by R. Landauer[1] in the year 1960. According to Landauer’s principle, the loss of one bit of information lost, will dissipate kT\*ln(2) joules of energy where, k is the Boltzmann’s constant and k=1.38x10-23 J/K, T is the absolute temperature in Kelvin. The primitive combinational logic circuits dissipate heat energy for every bit of information that is lost during the operation. This is because according to second law of thermodynamics, information once lost cannot be recovered by any methods. Bennett showed that if a computation is carried out in Reversible logic zero energy dissipation is possible, as the amount of energy dissipated in a system is directly related to the number of bits erased during computation. The design that does not result in information loss is irreversible. A set of reversible gates are needed to design reversible circuit. Several such gates are proposed over the past decades. According to Moore’s law the numbers of transistors will double every 18 months. Thus energy conservative devices are the need of the day. The amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation. Reversible circuits are those circuits that do not lose information. The most prominent application of reversible logic lies in quantum computers. Quantum networks composed of quantum logic gates; each gate performing an elementary unitary operation on one, two or more two–state quantum systems called qubits. Each qubit represents an elementary unit of information; corresponding to the classical bit values 0 and Quantum arithmetic must be built from reversible logical components . Reversible computation in a system can be performed only when the system comprises of reversible gates. A circuit/gate is said to be reversible if the input vector can be uniquely recovered from the output vector and there is a one-to-one correspondence between its input and output.

# PROBLEM STATEMENT

Reversible logic gates form the basis of computational circuits where information can be encoded and processed in a manner that allows the reconstruction of input values from output values. Unlike traditional logic gates (which are irreversible), reversible gates ensure that no information is lost during computation, enabling computations to be theoretically performed without dissipating heat. The problem statement in reversible logic gates involves designing circuits composed of reversible gates to perform specific computational tasks while satisfying constraints such as minimizing the number of gates used, adhering to energy conservation principles, and ensuring reversibility. This field is crucial for quantum computing, low-power computing, and certain specialized applications where energy efficiency and information preservation are paramount.

# OBJECTIVE

In Verilog, the objective of implementing reversible logic gates aligns with the broader objective of designing circuits that are reversible and perform computations without losing information. Verilog, a hardware description language, enables the creation of digital circuits, including reversible logic gates.

**CHAPTER 2**

**LITERATURE SURVEY**

# 2.1 INTRODUCTION

In the pursuit of advancing computational architectures, reversible logic gates have emerged as a pivotal domain with profound implications for diverse fields ranging from classical computing to quantum information processing. The fundamental distinction of reversible logic gates lies in their ability to execute computations without losing information, presenting a promising avenue towards more energy-efficient and information-conserving computing paradigms.

**2.2 Theoretical Underpinnings of Reversible Computing**

Reversible logic gates operate on principles distinct from their irreversible counterparts. This section delves into the theoretical foundations underlying reversible computing, exploring concepts of information conservation, the role of unitarity, and the mathematical formulations governing reversible gate operations.

**2.3 Design and Optimization of Reversible Logic Gates**

A critical aspect involves the design and optimization of reversible logic gates. This segment surveys the diverse array of reversible gate constructions, such as Toffoli gates, Fredkin gates, and their variants. It explores methodologies for gate optimization, gate count minimization, and the trade-offs between gate complexity and functionality.

**2.4 Circuit Implementations**

Beyond theoretical constructs, this section delves into practical implementations of reversible logic gates. It explores Verilog and other hardware description languages for modeling, simulating, and synthesizing reversible circuits. Additionally, it covers synthesis techniques optimizing circuit layouts for hardware realization.

**2.5 Applications in Quantum Computing**

Reversible logic gates hold promise in quantum computing due to their inherent reversibility. This part investigates their role in quantum circuitry, quantum algorithms, and their potential contributions to quantum information processing. Furthermore, it explores applications in classical computing paradigms, emphasizing low-power computing and specialized computational tasks.

**2.6 Challenges and Future Directions**

Highlighting the current challenges within the realm of reversible logic gates, this section discusses issues such as scalability, fault tolerance, and integration with existing computing architectures. Moreover, it speculates on future directions, including interdisciplinary collaborations and potential breakthroughs.

# 2.7 SUMMARY

The literature survey on reversible logic gates navigates through the multifaceted landscape of computing architectures, focusing on the significance and applications of reversible computing. It aims to explore theoretical foundations, gate design and optimization, practical implementations, and diverse applications in both quantum and classical computing.

**CHAPTER 3**

**EXISTING SYSTEM**

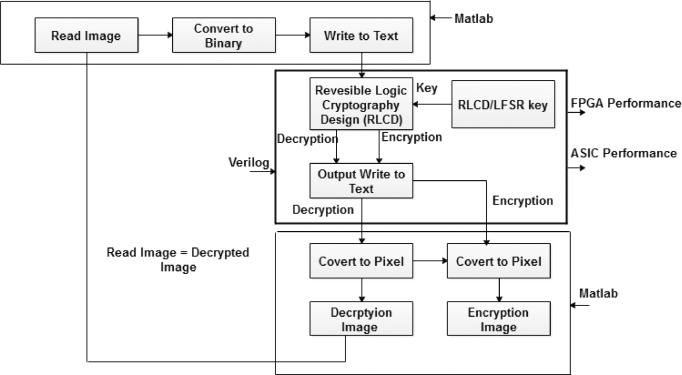
# 3.1 INTRODUCTION

The literature survey on reversible logic gates navigates through the multifaceted landscape of computing architectures, focusing on the significance and applications of reversible computing. It aims to explore theoretical foundations, gate design and optimization, practical implementations, and diverse applications in both quantum and classical computing.

**3.2 EXISTING SYSTEM**

Verilog is a hardware description language commonly used for the design and verification of digital circuits. Reversible logic gates are unique in that they have an equal number of inputs and outputs, and information can theoretically be reversed without loss. In existing system Programmable Array Logic (PAL), Programmable Array Logic (PAL) & Gate Array Logic (GAL) are designed using the reversible logic. The programmable AND gate plane & OR gate plane are designed using reversible fuse while fixed connections by using the CNOT gate . The time delays for reversible PAL, PLA & GAL are 5.847nsec, 5.847nsec & 5.847nsec. Time delay increases if the quantum cost is increased. The quantum cost increases with increase in length of Boolean equation. The Circuit has been designed and simulated using Xilinx 14.7 software and implemented on FPGA.

# 3.3 BLOCK DIAGRAM



**SUMMARY**

The existing system details the use of reversible logic in crafting Programmable Array Logic (PAL), Programmable Logic Array (PLA), and Gate Array Logic (GAL). Within this system, the programmable AND gate plane and OR gate plane are constructed using reversible fuse technology, while fixed connections are established using the CNOT gate. Recorded time delays for reversible PAL, PLA, and GAL stand at 5.847 nanoseconds each, indicating an increase in delay correlated with higher quantum cost. The quantum cost itself escalates with the elongation of Boolean equations.

**CHAPTER 4**

**PROPOSED SYSTEM**

**4.1 BASIC DEFINITIONS OF REVERSIBLE LOGIC GATES**

In this section some important factors in reversible logic are explained. The main object in reversible logic theory is the reversible function, which is defined as follows :

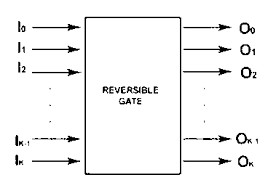
**4.2 Reversible Function:**

The Boolean function f(x1, x2 … xn) of n Boolean variables is called reversible if:

1. The number of outputs is equal to the number of inputs.

2. Any output pattern maps to a unique input pattern.

In other words, reversible functions are those that perform permutations of the set of input vectors .

For an (n, k) function, that is ,function with n-input koutput, it is necessary to add inputs and/or outputs to make it reversible. 

**4.3 Reversible logic gate:**

Reversible Gates are circuits in which number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs . It not only helps us to determine the outputs from the inputs but also helps us to uniquely recover the inputs from the outputs.

**4.4 Garbage outputs:**

Garbage is the number of outputs added to make an n-input k-output function ((n; k) function) reversible. The relation between garbage outputs and constant inputs is [7] Input + constant input = output + garbage. As with reversible gates, a reversible circuit has the same number of input and output wires; the reversible circuit with n inputs is called an n X n circuit or a circuit on n wires.

**4.5 Quantum cost:**

Quantum cost refers to the cost of the circuit in terms of the cost of a primitive gate. It is calculated knowing the number of primitive reversible logic gates (1\*1 or 2\*2) required to realize the circuit. F. Flexibility: Flexibility refers to the universality of a reversible logic gate in realizing more functions .

**4.6 Gate Level:**

This refers to the number of levels in the circuit which are required to realize the given logic functions. H. Hardware Complexity: This refers to the total number of logic operation in a circuit. Means the total number of AND, OR and EXOR operation in a circuit.

**4.7 Design Constraints for Reversible Logic Circuits:**

Reversible logic imposes many design constraints that need to be either ensured or optimized for implementing any particular Boolean functions.

1) In reversible logic circuit the number of inputs must be equal to the number of outputs.

2) For each input pattern there must be a unique output pattern.

3) Each output will be used only once, that is, no fan out is allowed.

4) The resulting circuit must be acyclic.

**CHAPTER 5**

**SOFTWARE DESCRIPTION**

# 5.1 REVERSIBLE LOGIC GATES

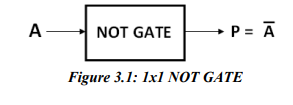
# In this section, we describe all about reversible logic and reversible logic gates. Though it is already briefly described about garbage outputs, in this section we will define these with more appropriate Reversible logic gates.

**5.2 METHODS OF REVERSIBLE LOGIC GATE**

* NOT GATE
* FEYNMAN GATE
* TOFFOLI GATE
* FREDKIN GATE
* PERES GATE

**5.2.1 NOT Gate:**

# 1\*1 NOT gate is the simplest among all the reversible gates where the gate has only one input (A) and one output (B) such that B = A’. The block diagram for 1\*1 NOT gate (Quantum Cost

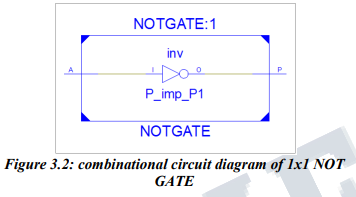


**Code :**

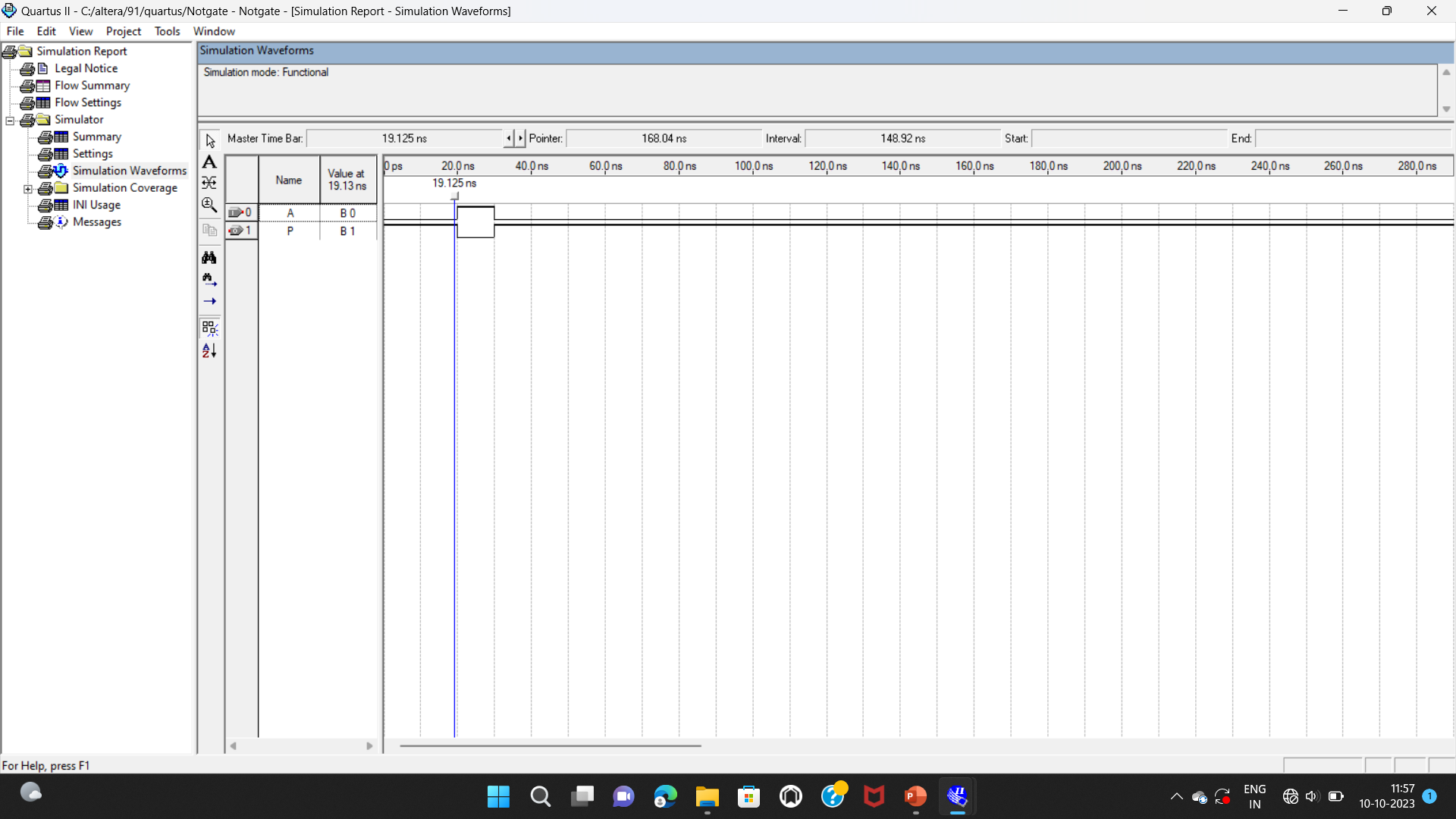
module NOTGATE( input A, output P );

assign P = ~A;

endmodule

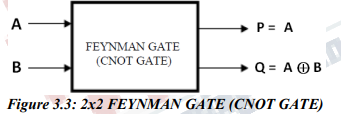


**OUTPUT**



**5.2.2 Feynman Gate:**

Let Iv and Ov be the input and output vector of a 2\*2 Feynman gate (FG) [16,17] respectively, where Iv= (A,B) and Ov = (P=A, Q=A⊕B). The block diagram for 2\*2 Feynman gate is shown in Fig.3.3. (Quantum Cost = 1)



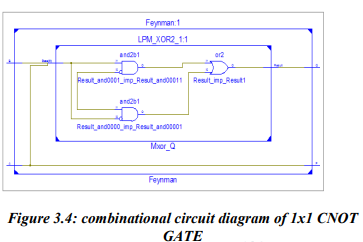
**Code:**

module Feynman( input A, B, output P, Q );

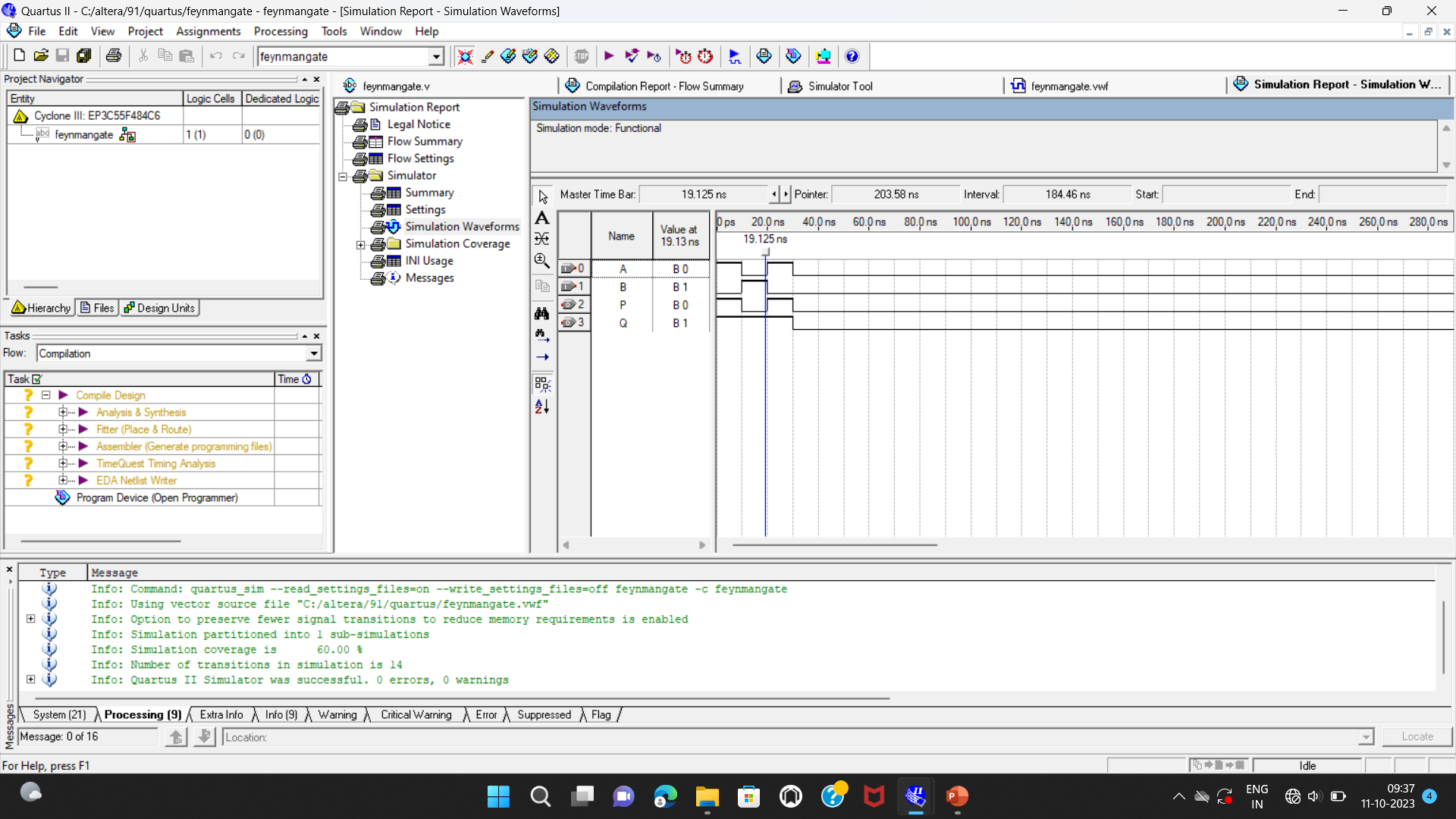
assign P = A;

assign Q = A^B;

endmodule

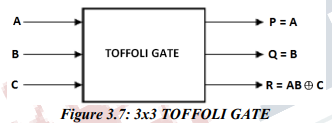


**OUTPUT**



**5.2.3 Toffoli Gate:**

Let Iv and Ov be the input and output vector of a 3\*3 Toffoli Gate (TG) [18,19] respectively, where Iv =(A, B, C) and Ov=(P=A, Q=B, R=AB⊕ C). Fig.3.7 shows the 3\* 3 Toffoli gate. (Quantum Cost = 5)



**Code:**

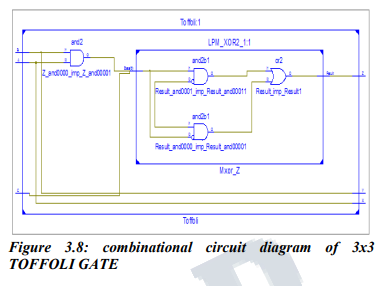
module Toffoli( input A,B,C, output P,Q,R );

assign P = A;

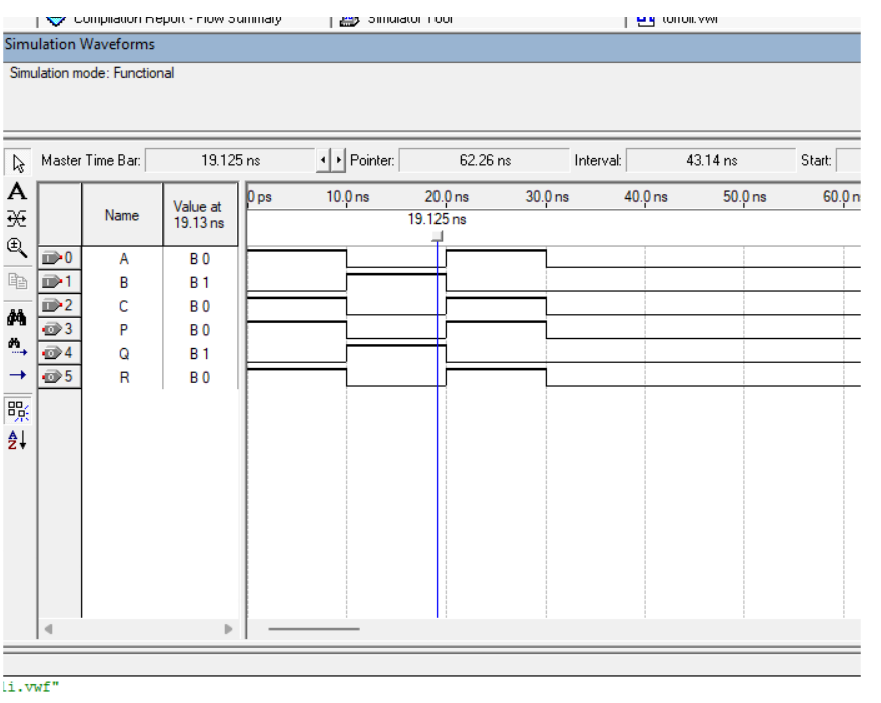
assign Q = B;

assign R = (A&B)^C;

endmodule

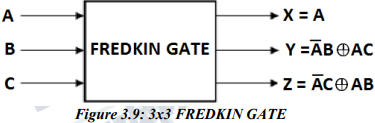


OUTPUT



**5.2.4 Fredkin Gate:**

Let Iv and Ov be the input and output vector of a 3\*3 Fredkin Gate [18,20] respectively, where Iv=(A,B,C) and Ov=(X=A,Y=A’B⊕AC , Z=A’C⊕AB). Fig. 3.9 shows the block diagram of 3\*3 Fredkin gate. (Quantum Cost = 5)



**Code:**

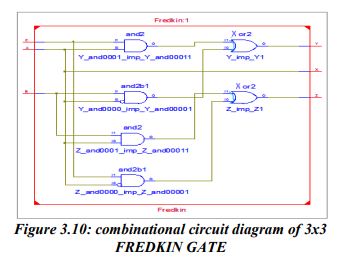
module Fredkin( input A, B, C, output X, Y, Z );

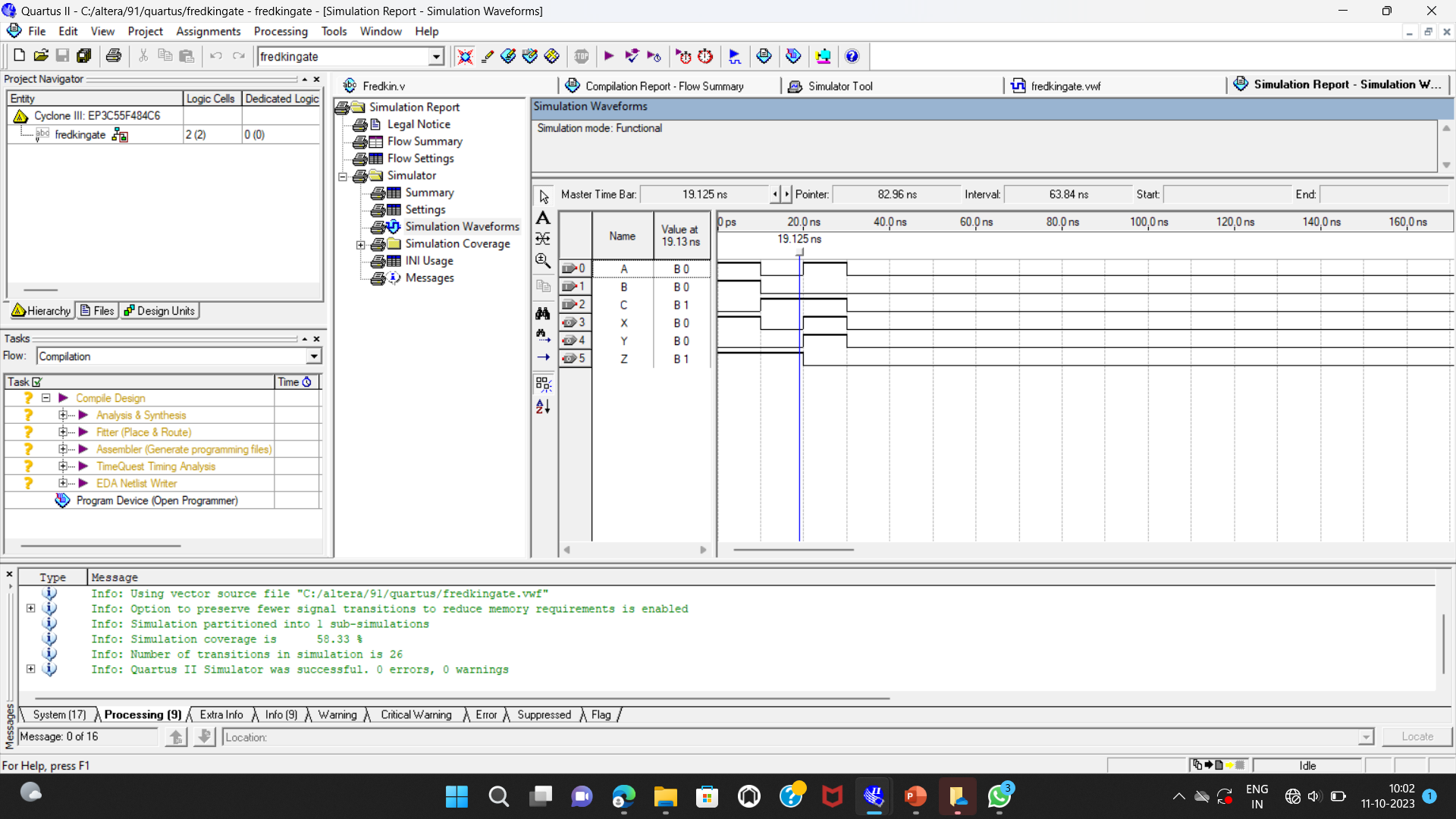
assign X = A;

assign Y =((~A)&B) ^ (A&C);

assign Z =((~A)&C) ^( A&B);

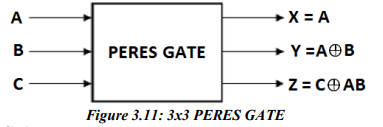
endmodule



**OUTPUT**

**5.2.5 Peres Gate:**

Let Iv and Ov be the input and output vector of a 3\*3 Peres Gate [18,20,21] respectively, where Iv=(A,B,C) and Ov=(X=A,Y=A⊕B , Z=AB⊕C). Fig. 3.11 shows the block diagram of 3\*3 Peres gate. (Quantum Cost = 4)



**Code:**

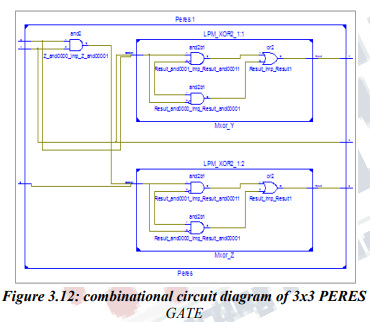
module Peres(input A,B,C, output X,Y,Z );

assign X = A;

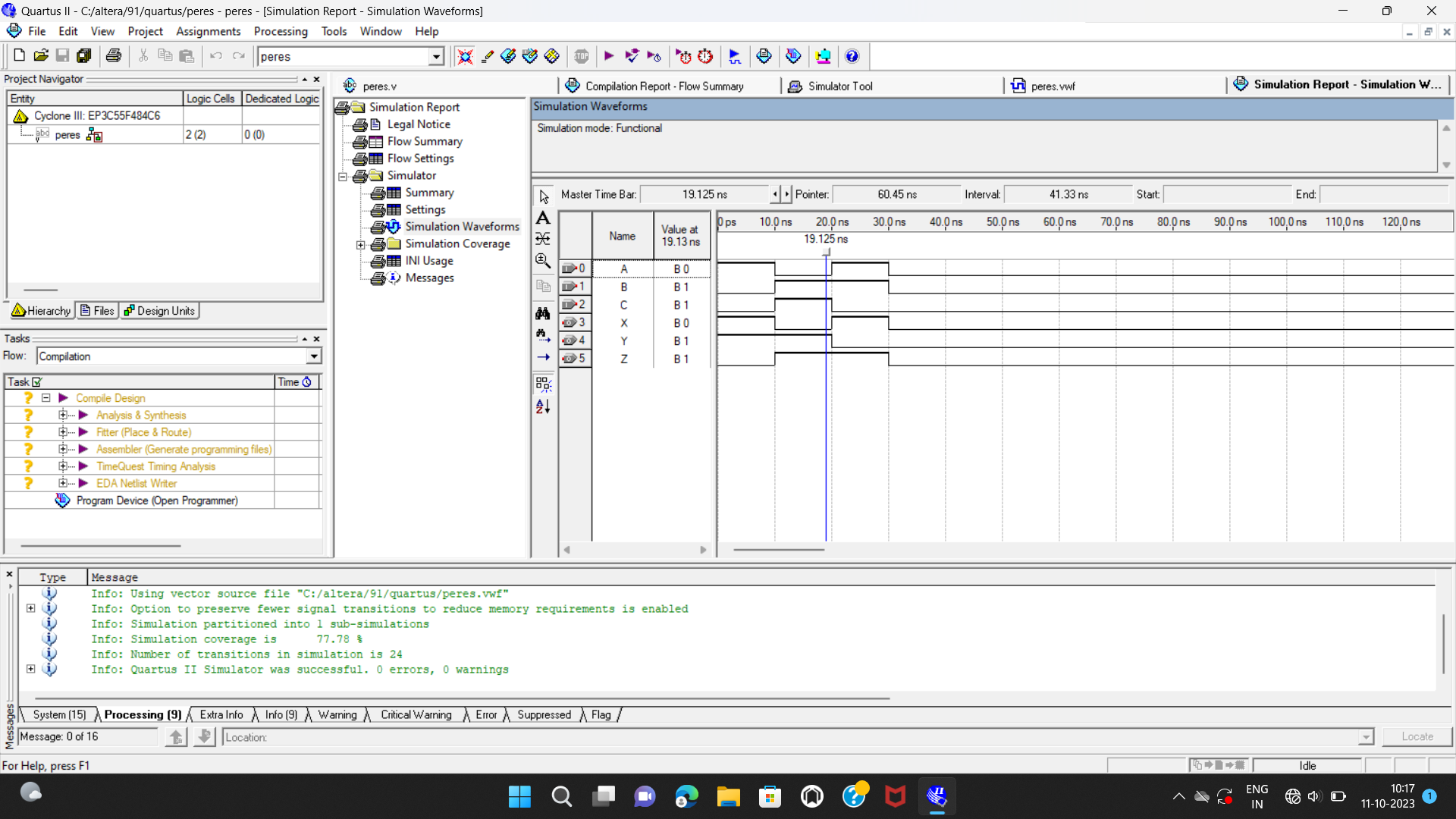
assign Y =A^B;

assign Z =(A&B)^C;

endmodule



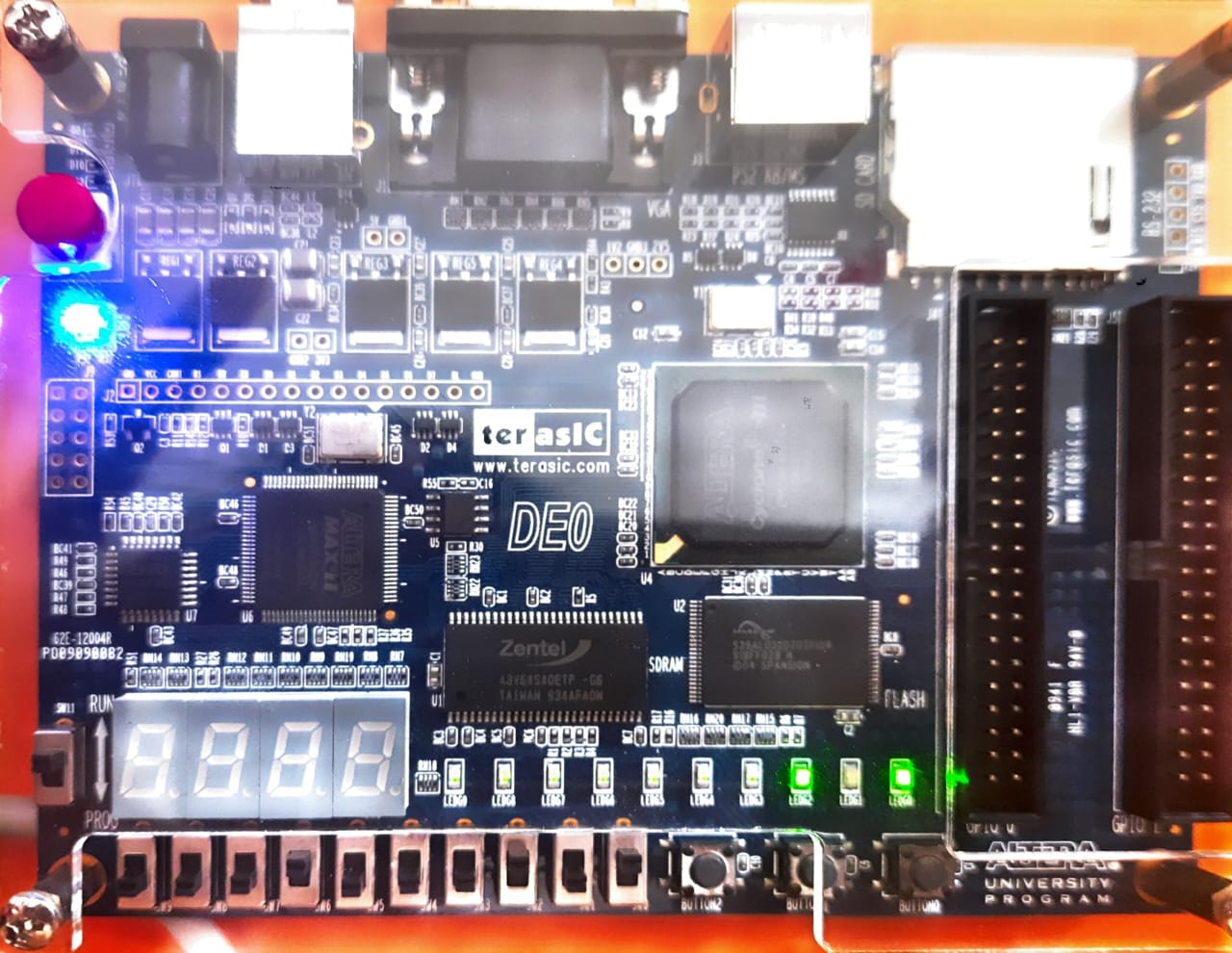
**OUTPUT**

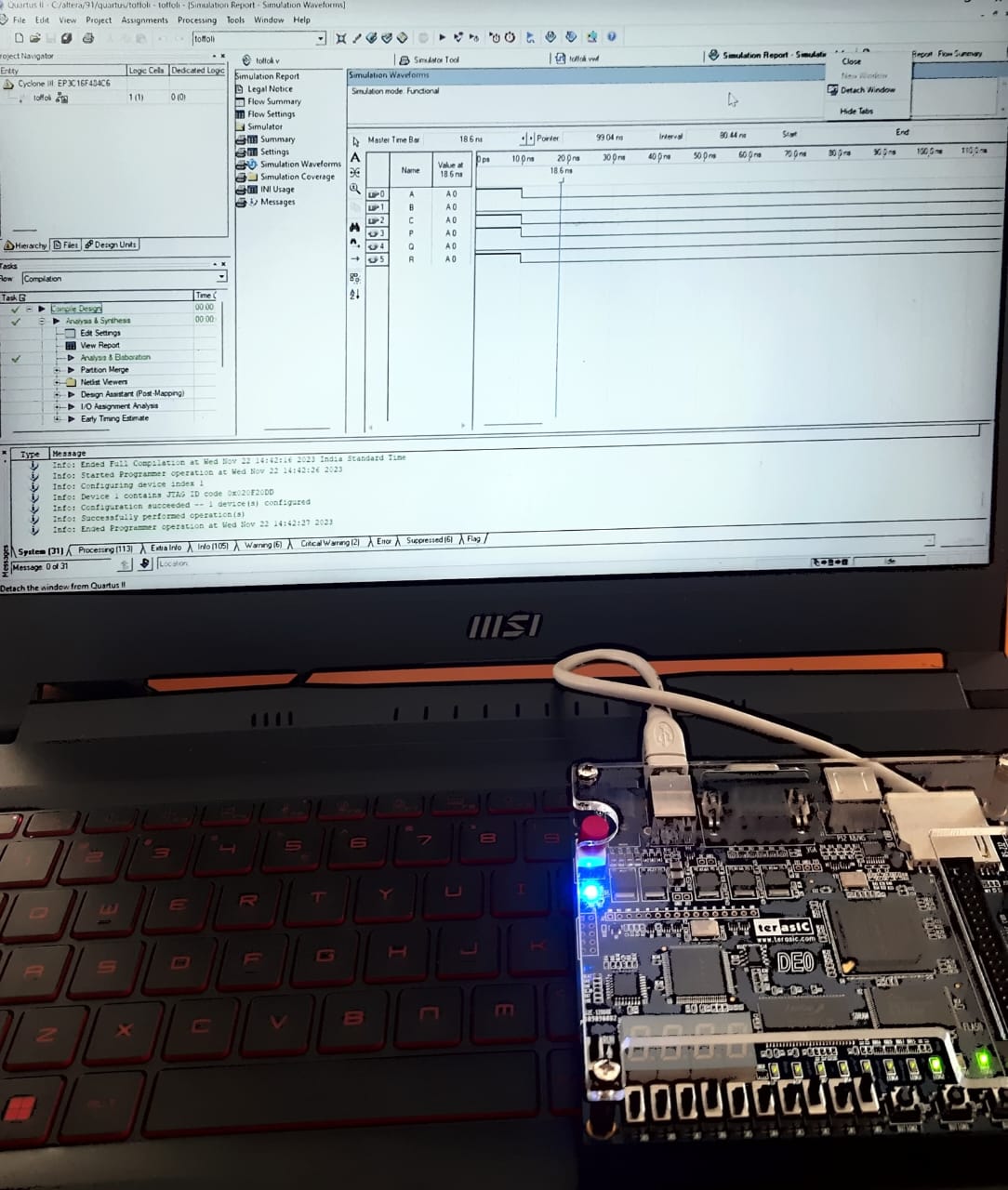
****

**CHAPTER 6**

**HARDWARE DESCRIPTION**

**6.1 FPGA IMPLEMENTATION**





**CHAPTER 7**

**CONCLUSION**

**CONCLUSION**

This paper presents Verilog CODE for all Reversible Logic Gates, which provide us to design Verilog CODE of any complex combinational circuit. Here we have tried to make the Verilog code as much as possible. We can simulate and synthesis it using Xilinx 15.1 software and verified using Z series board and also calculate the power consumption and compare it with the irreversible Combinational Circuits.

**REFERENCES**

1. R.Landauer, “Irreversibility and Heat Generation in the Computational Process”, IBM Journal of Research and Development, 5,pp. 183-191,2022.

2. C.H.Bennett, “Logical Reversibility of Computation”, IBM J.Research and Development, pp.525-532, November2022.

3. Vlatko Vedral, Adriano Bareno and Artur Ekert, “QUANTUM Networks for Elementary Arithmetic Operations”, arXiv:quantph/ 9511018 v1, nov 2021.

4. Perkowski, M., A. Al-Rabadi, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, M. Azad Khan, A. Coppola, S.Yanushkevich, V. Shmerko and L. Jozwiak, ―”A general decomposition for reversible logic”, Proc. RM’2021,

5. Perkowski, M. and P. Kerntopf, ―”Reversible Logic. Invited tutorial”, Proc. EURO-MICRO, Sept 2020, Warsaw, Poland.

6. Hafiz Md. Hasan and A.R. Chowdhury, ―”Design of Reversible Binary Coded decimal Adder by using Reversible 4 – bit Parallel Adder”, VLSI Design 2020, pp. 255 – 260, Kolkata, India, Jan 2020.

7. V. V. Shende, A. K. Prasad, I. L. Markov, and J. P .Hayes,“ Synthesis of Reversible Logic Circuits”, IEEE Transaction on computer-aided design of integrated circuits and systems, vol. 22, No. 6, June 2020.

8. Mounika.N and V.Vijay Bhaskar, “Realization of Programmable Logic Array Using Compact Reversible Logic Gates”, International journal & magazine of engineering, technology, management and research July 2020.