A Comparative Study of 16-bit Adder using Reversible Logic Gates.

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

in

Electronics and Telecommunication

by

Zaid Surve (15DET133)
Faraz Khan (15ET19)
Saddam Ansari (16DET75)
Mohammad Ibrahim Khan (16DET94)

Under the guidance of

Professor Afzal Shaikh

Department of Electronics and Telecommunication Engineering

Anjuman-I-Islam's Kalsekar Technical Campus
Sector 16, New Panvel , Navi Mumbai
University of Mumbai

2019-2020

CERTIFICATE

Department of Electronics and Telecommunication Engineering
Anjuman-I-Islam's Kalsekar Technical Campus
Sector 16, New Panvel , Navi Mumbai
University of Mumbai

This is to certify that the project entitled **A Comparative Study 16-bit Adder using Reversible Logic Gates** is a bonafide work of **Zaid Surve (15DET133), Faraz Khan (15ET19), Saddam Ansari (16DET75), Mohammad Ibrahim Khan (16DET94)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Department of Electronics and Telecommunication Engineering.

1

Supervisor

Examiner

Head of Department

Director

PROJECT REPORT APPROVAL FOR BACHELOR OF ENGINEERING

This project entitled **"A Comparative Study of 16-bit Adder using Reversible Logic Gates"** by of **Zaid Surve (15DET133), Faraz Khan (15ET19), Saddam Ansari (16DET75), Mohammad Ibrahim Khan (16DET94)** is approved for the degree of **Bachelor of Engineering in Electronics & Telecommunication Engineering**.

Examiner

....

Supervisor

....

Date:

Place:

DECLARATION

We declare that this written submission represents my ideas in my own words and where others ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

……………………
Zaid Surve
(15DET133)

……………………
Faraz Khan
(15ET19)

……………………
Saddam Ansari
(16DET75)

……………………
Mohammad Ibrahim Khan
(16DET94)

Date:

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them. We are highly indebted to Professor Afzal Shaikh for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We would like to express my gratitude towards my parents & Staff of Anjuman-I-Islam's Kalsekar Technical Campus for their kind co-operation and encouragement which help me in completion of this project. Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped me out with their abilities.

Zaid Surve (15DET133)
Faraz Khan (15ET19)
Saddam Ansari (16DET75)
Mohammad Ibrahim Khan (16DET94)

ABSTRACT

A reversible logic gate is a gate in which the output is guaranteed, but the input is not confirmed, in other words, reversible functions are those that perform permutations of the set of input vectors. Reversible Gates are circuits in which outputs are equal to the number of inputs, and there is one to one correspondence between the vectors of inputs and outputs. Reversible logic circuits have emerged as a promising technology has deficient power consumption with high speed and less delay, which is in demand with the increasing universal growth. Reversible logic Gates have its application in low power CMOS, Quantum Computing, Nanotechnology, and Optical Communication. As we know, the Reversible logic circuits deal with low power consumption with high speed and less delay we are going to design and analysis of 16-bit Full Adder using Reversible logic gates, and we need to observe the result and compare with conventional logic gates and observe the essential parameters between both designs.

CONTENTS

LIST OF FIGURE

Chapter 1

INTRODUCTION

In the modern VLSI system, power dissipation is very high due to the rapid switching of internal signals. The complexity of VLSI circuits increases with each year due to packing more and more logic elements into smaller volumes. Hence power dissipation has become the central area of concern in VLSI design. Reversible logic has its basics from the thermodynamics of information processing. According to this, traditional irreversible circuits generate heat due to the loss of information during computation. So the conventional circuits are designed using reversible logic to avoid this information loss. Landauer [1961] showed that the circuits designed using irreversible elements dissipate heat due to the loss of information bits. It is proved that the loss of one bit of information results in the dissipation of KT*log2 joules of heat energy where K is the Boltzmann constant and T is the temperature at which the operation is performed. Benett [1973] showed that this heat dissipation due to information loss could be avoided if the circuit is designed using reversible logic gates. Gate is considered to be reversible only if, for every input, there is a unique output assignment. Hence there is a mapping between the input and output vectors. A circuit that is built from reversible gates is known as a reversible logic circuit. Thus inputs to reversible gates can be uniquely determined from its outputs. A reversible logic gate must have the same number of inputs and outputs. In an n-output reversible gate, the output vectors are a permutation of the numbers 0 to 2n-1. A reversible gate is balanced, i.e., the outputs are 1s for exactly half of the inputs. A circuit without constants on its inputs and composed of reversible gates realizes only balanced functions. It can realize unbalanced functions only with garbage outputs. Some of the significant problems with reversible logic synthesis are fan-outs cannot be used, and also feedback from gate outputs to inputs are not permitted. In this paper, the implementation of three types of adder circuits using the reversible gate R is presented.

Energy dissipation is one of the significant issues in present-day technology. R. Landauer demonstrated energy dissipation due to information loss in high technology circuits and systems constructed using irreversible hardware in the year 1960. According to Landauer's principle, the loss of one bit of information lost 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 first combinational logic circuits dissipate heat energy for every bit -of information that is lost during the operation. So according to the second law of thermodynamics, information once lost cannot be recovered by any methods. In 1973, Bennett showed that reversible circuits could avoid the loss of kTln2 joules of energy dissipation in a circuit. According to Moore’s law, the numbers of transistors 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. Energy dissipation and overheating is a serious concern for both manufacturers and consumers in digital circuit design. The problem of energy dissipation is related to the non-ideality of switches, which results in high information losses. Higher levels of integrations and the use of new fabrication processes have been dramatically reduced the energy loss over the last decades.

1.1 FUNDAMENTAL REVERSIBLE LOGIC GATES USED

Reversible computing is a bijective device with an equal number of input and output lines that provide one to one mapping between input and output lines. The purpose of reversible computing is eliminating the energy dissipation that is produced by information destruction. Reversible logic supports the process of running the system both forward and backward. In the synthesis of reversible gate circuits, direct fan-out is not allowed as a one-to-many concept is not reversible. However, fan-out in reversible circuits is achieved using additional gates. A reversible circuit should be designed using a minimum number of reversible logic gates.

- Constant input - Constant input refers to the number of inputs that are to be kept constant at either 0 or 1 so that they synthesize the given logical function.
- Garbage Outputs - Whenever an equal number of “Garbage” is the number of outputs added to make an (n, k) functions reversible.
- Quantum Cost - Quantum cost refers to the cost of the circuit in terms of the cost of a first gate. It is calculated knowing the number of first reversible logic gates (1*1 or 2*2) required to realize the circuit.

There are many numbers of reversible logic gates that exist at present. The quantum cost of each reversible logic gate is a vital optimization parameter. The quantum cost of a 1x1 reversible gate is assumed to be zero, while the quantum cost of a 2x2 reversible logic gate is taken as unity.

Some of the important reversible logic gates are,

- ❖ NOT Gate: The simplest Reversible gate is NOT gate and is a 1*1 gate. The Reversible 1*1 gate is NOT Gate with zero Quantum Cost is as shown in the Figure 1.1.1.

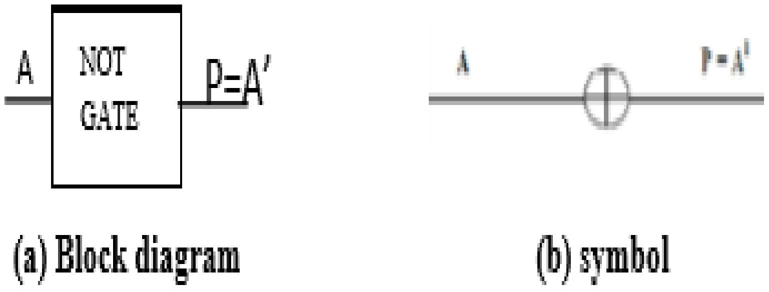


Fig. 1.1.1 NOT Gate

- ❖ CNOT GATE: CNOT gate is also known as controlled-not gate. It is a 2*2 reversible gate. The CNOT gate can be described as:

I_v = (A, B) ; O_v = (P= A, Q= A B) I_v and O_v are input and output vectors respectively.

Quantum cost of CNOT gate is 1. Fig. 1.1.2 shows a 2*2 CNOT gate and its symbol.

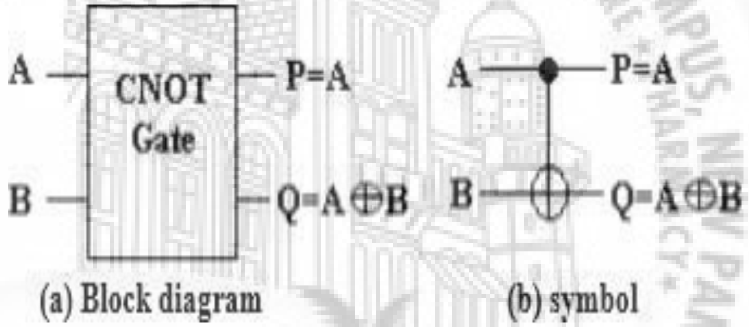


Fig. 1.1.2 CNOT Gate

- ❖ FEYNMAN Gate: The Feynman gate which is a 2*2 gate and is also called as Controlled NOT and it is widelyused for fan-out purposes. The inputs (A, B) and outputs P=A, Q= A XOR B. It has Quantum cost one.

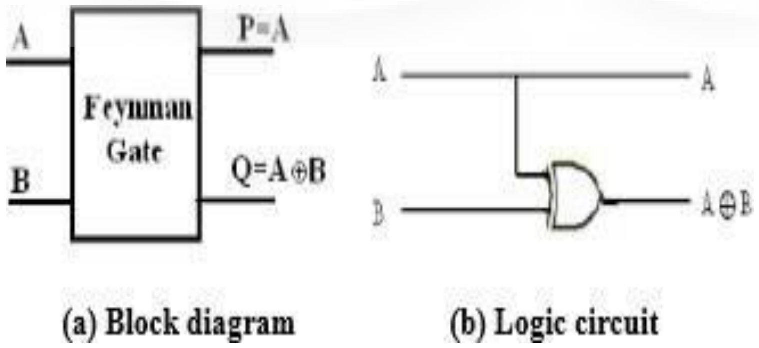


Fig. 1.1.3 Feynman Gate

- Double Feynman Gate (F2G): It is a 3*3 Double Feynman gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The outputs are defined by P = A, Q=AÅB, R=AÅC. Quantum cost of Double Feynman gate is 2.

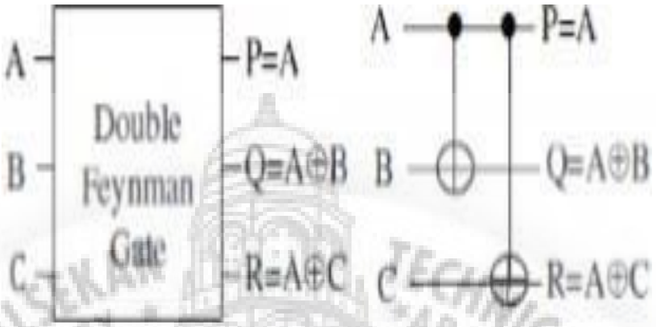


Fig. 1.1.4 Double Feynman Gate

- TOFFOLI Gate: TOFFOLI gate which is a 3*3 gate with inputs (A, B, C) and outputs P=A,Q=B, R=AB XOR C. It has Quantum cost five.

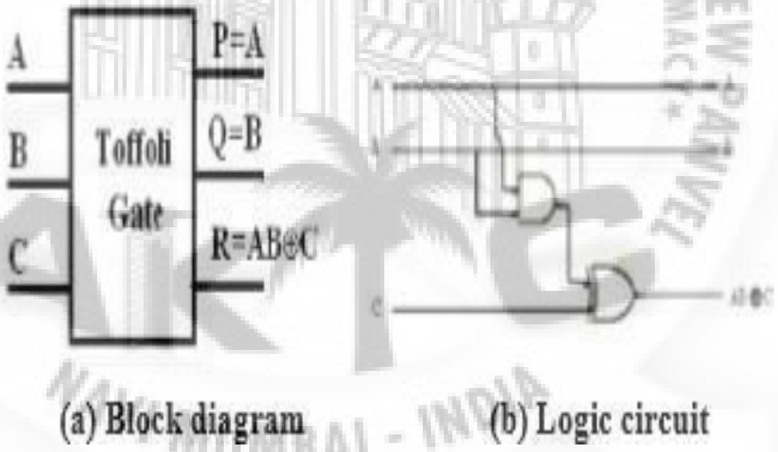
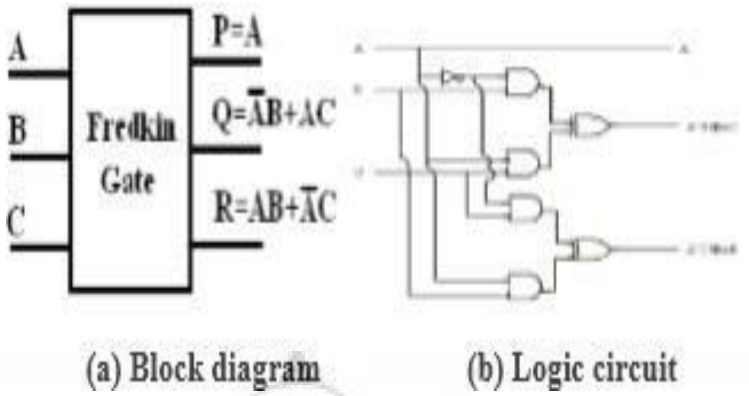


Fig. 1.1.5 Toffoli gate

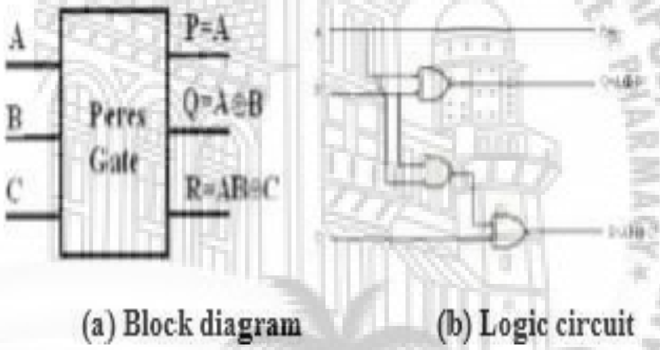
- FREDKIN Gate: Fredkin gate which is a 3*3 gate with inputs (A, B, C) and outputs P=A, Q=A'B+AC, R=AB+A'C. It has Quantum cost five.



(a) Block diagram (b) Logic circuit

Fig. 1.1.6 Fredkin Gate

- ❖ PERES Gate: Peres gate which is a 3*3 gate having inputs (A, B, C) and outputs P = A; Q=AXOR B; R = AB XOR C. It has Quantum cost four.



(a) Block diagram (b) Logic circuit

Fig. 1.1.7 Peres Gate

- ❖ HNG Gate: The reversible HNG gate can work singly as a reversible full adder. If the input vector I_v = (A, B, Cin, 0), then the output vector becomes O_v = (P=A, Q=C_{in}, R=Sum, S=Count).

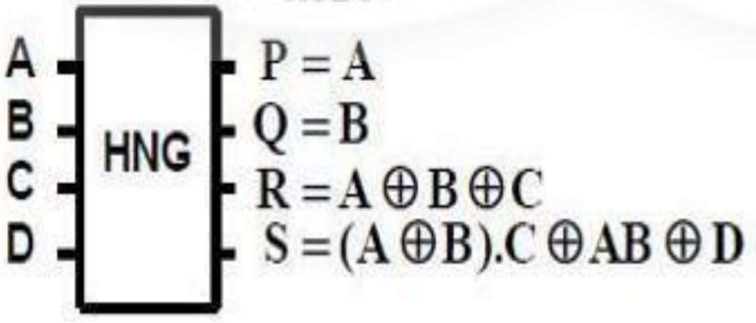


Fig. 1.1.8 HNG Gate

1.2 Statement of Project

The project will deal with the design and analysis of a 16-bit Full adder using Reversible Logic and comparison with the design using conventional logic gates. The parameters such as Power dissipation, Delay, Area, LUT's, etc. will be compared after the implementation is done using VHDL programming.

1.3 Motivation

In the area of digital design, an aspect of sustainability is the power dissipation of integrated circuits. **Reversible logic circuits** have emerged as a promising technology that has deficient **power consumption** with **high speed** and **less delay**, which is in demand with the increasing universal growth. **The complexity of VLSI circuits increases with each year due to packing more and more logic elements into smaller volumes. Hence power dissipation has become the central area of concern in VLSI design.** Due to power dissipation delay take place, which results in lagging of circuit and more power consumption, to overcome power dissipation, we are going to use and implement reversible logic gates.

1.4 Objective

The primary purposes of designing the circuit using Reversible logic gates are to decrease quantum cost, depth of the circuits, and the number of garbage outputs. The purpose of this project is to give a frame of reference, understanding, and overview of reversible gates. In this project, various reversible logic gates and their applicability to logic design have been discussed. Also, a brief framework of comparisons between various reversible circuits with conventional logic gates presented based on various parameters. As we know, the Reversible logic circuits deal with low power consumption with high speed and less delay we are going to design and analysis of **16-bit Full Adder** using **Reversible logic gates**.

Chapter 2

Literature Review

Energy dissemination because of data misfortunes in high innovation framework developed utilizing irreversible equipment was appeared by R. Landauer in 1960 portrayed that consistent reversibility is related with physical reversibility and requires a negligible heat generation for every machine cycle which expresses that rationale calculation that are not reversible fundamentally create kT∗ln2 joules of heat energy for all of data that is lost, where k is the Boltzmann's steady and T is the outright temperature at which calculation is performed. For room temperature, this measure of dispersing heat can be little, however, not negligible. This sum may not appear to be critical, yet it ends up significant later on. In 1965, Gordon. E. Moore forecasts that several transistors on a chip would double every two years and so. His work is known as Moore's law. Because of the impact of Moore's law, the number of segments on-chip prompt increments in power dissemination. The measure of intensity scattered is equivalent to the heat disseminated in the chip. Along these lines, control minimization has turned into a significant factor. Further, in 1973 C.H. Bennett, which expresses that if a calculation is completed in reversible logic, zero energy scattering is conceivable as the measure of energy dispersed in the framework is legitimately relative to the number of bits deleted during the calculation. The structure that not brings about data misfortune is irreversible. A lot of reversible gates are expected to structure reversible circuits. Although he uncovered that the calculation, which was performed on irreversible gates, can also be performed with the same effect on the reversible gates. Edward Fredkin and Tommaso Toffoli in the 1980s had presented with new reversible gates referred to by their name as Fredkin & Toffoli gates, which work on reversibility. These gates are general gates with 3 data sources and 3 output terminals, consequently otherwise called 3*3 reversible doors. These reversible gates have very nearly zero power dissemination. It is broadly utilized among every one of the gates contributions to its less quantum cost (equivalents to 4) as for all-inclusive gates. It is widely used among all the gates dues to its less quantum cost (equals to 4) with respect to universal gates.shor in 1994 generated an algorithm, this algorithm uses the concept of reversibility .It runs on quantum computer in polynomial time for integer factorization. M.L. Chaung and C.Y. Wang offered that with minimal number of gate and garbage output while implementing the latches . Using combinations of 4*4 reversible gates many sequential circuits are designed by 2008 by S. ChiwandePrashant R. Yelker. TR gate was invented in 2009 by N Ranganathan and H Thalpiyal .This TR gate is used to reduce the garbage output of the reversible gate circuits. In 2010, implementation of sequential circuits (D latch, T latch, J K Latch etc.) using Feynman gate and fredkin gate has been carried out. Although lots of work had been done on reversible circuit but still the work on counters are still not performed. In 2011, V. Ranganthan and V. Raj Mohan , synchronous and asynchronous both type of counters were designed using reversible gates.Its applications in building reversible ALU, reversible processor etc. Basically this work leads to design and implementation of large and complex sequential circuits for quantum computers.

Chapter 3

Methodology

Full adder:

A full adder is a digital circuit that performs addition. Full adder is implemented with logic gate in hardware. An adder adds three one-bit binary number, two operands and a carry bit. The adder outputs two numbers, a sum and a carry bit. The term is contrasted with a half adder, which adds two binary digits.

Equation of full adder:

$= A \oplus B \oplus C_i$

$C_{i+1} = (A \oplus B) C_i \oplus AB$ (i = 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

Full adder diagram:

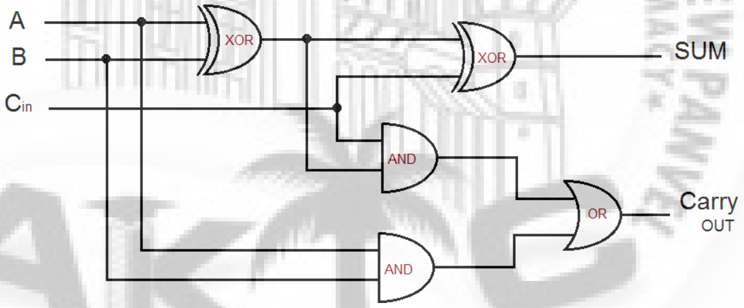


Fig. 3.1: Adder Using Conventional Gate

Full adder using reversible gate:

Peres gate:

Here 3*3 Peres Gate Work as Half Adder circuit And Third is Set as Zero Third input Act as Constant input for implement Reversible full adder circuit using Peres Gate and it require which should be in arranged as follows show in figure.

Now circuit is denoted as name Peres full Adder Gate (PFAG) and Peres full adder gate generate two Garbage output such as G1and G2 it requires one constant input and we set zero input to get the Desire output

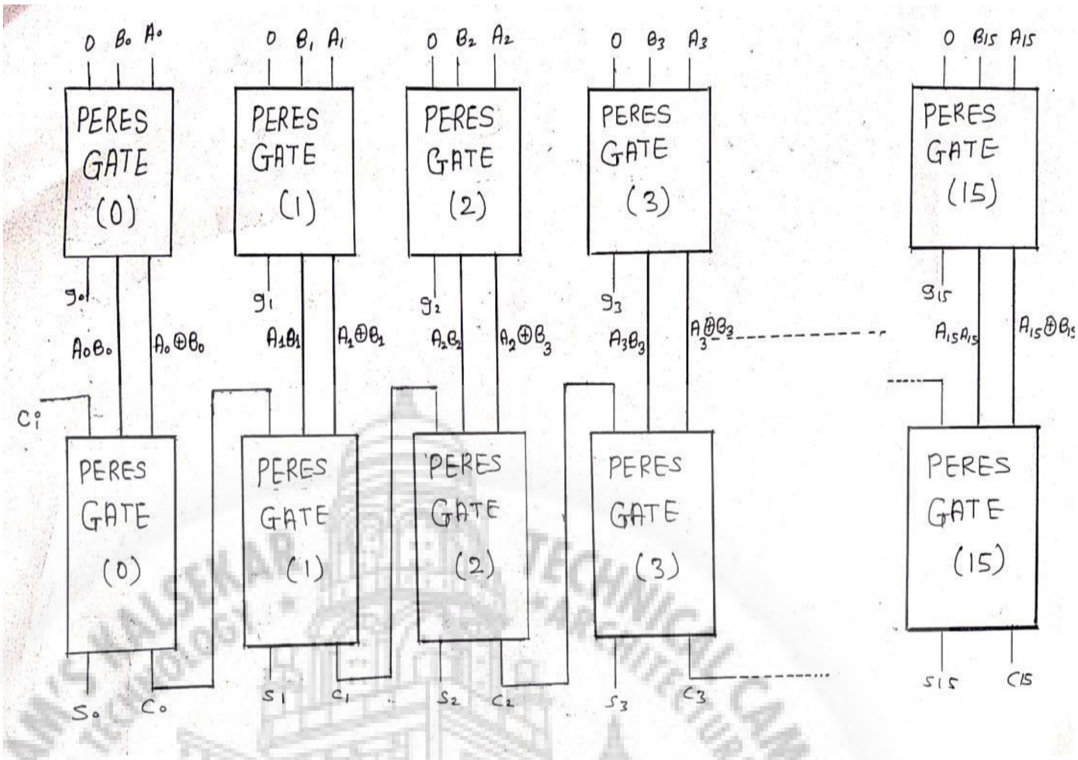


Fig. 3.2: 16-bit adder using Peres gate

HNG gate:

HNG Gate is nothing but a Full Adder used for designing Ripple Carry Adders. It can produce both the sum and carry in single Gate thus minimizing the Garbage and Gate Count.

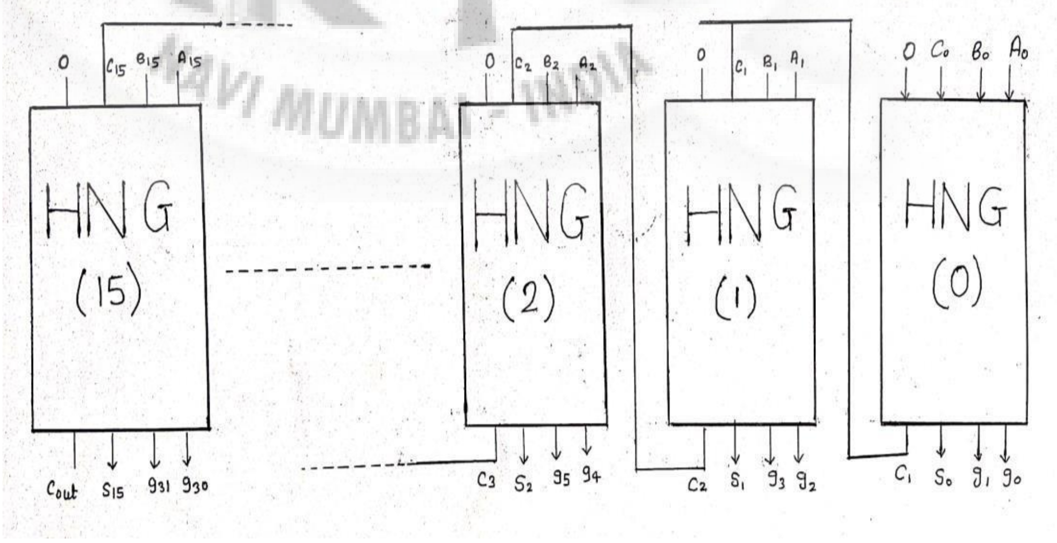


Fig. 3.3: 16-bit adder using HNG gate

16-bit full adder using above Peres and HNG gate to compare power analysis and delay with conventional gate.

3.2 Project Requirements

3.2.1 Software Requirements

In our project we are using Xilinx software VHDL programming on windows operating system this are the software requirements which help to run our project.

3.2.2 Hardware Requirements

In our project we are using FPGA board, FPGA is basically stand for Field Programmable Gate Array which is semiconductor device Based on matrix of configurable logic block and the FPGA board can be reprogrammed as per our requirements

CHAPTER 4

RESULT

The output is of 16-bit adder using conventional logic gate with the help VHDL programming and we observed the important parameter like time delay, I/O BUF, number of LUT used, memory usage, average fan-out & number of occupied slices.

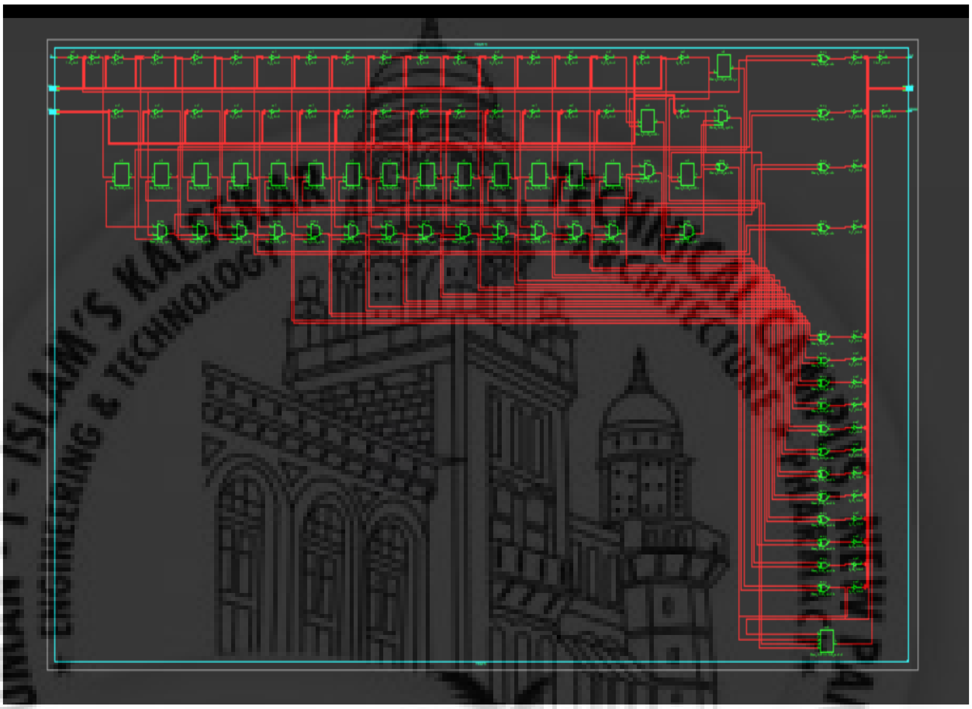
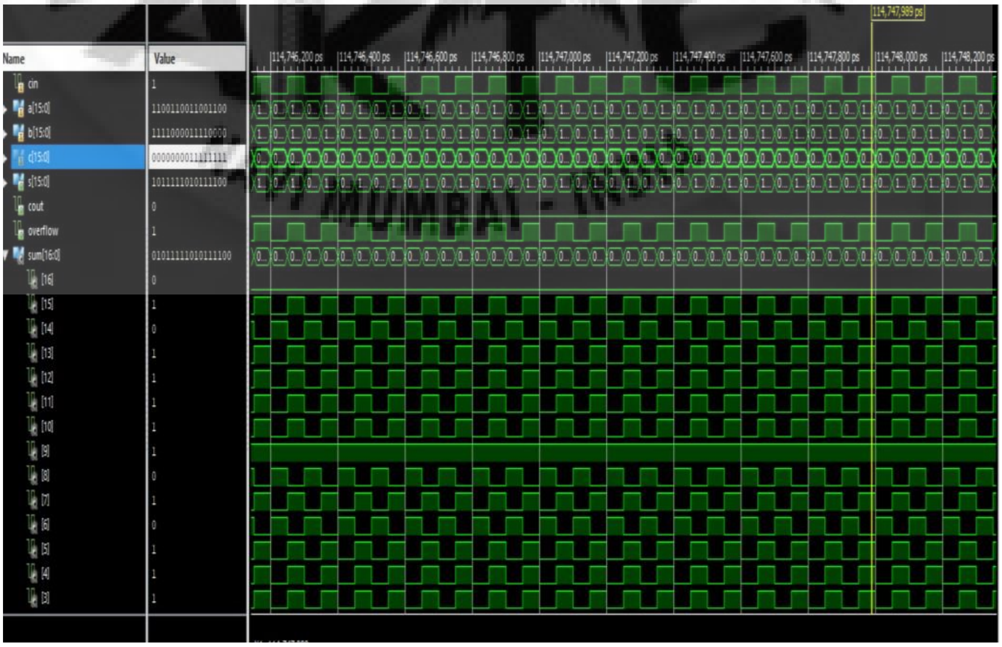


Fig. 4.1.1: 16-bit conventional adder layout



Name	Value
cin	1
a[15:0]	1100110011001100
b[15:0]	1111000011110000
c[15:0]	0000000011111111
s[15:0]	1011111010111100
cout	0
overflow	1
sum[16:0]	01011111010111100
[16]	0
[15]	1
[14]	0
[13]	1
[12]	1
[11]	1
[10]	1
[9]	1
[8]	0
[7]	1
[6]	0
[5]	1
[4]	1
[3]	1

Fig. 4.1.2: 16-bit conventional adder output waveform

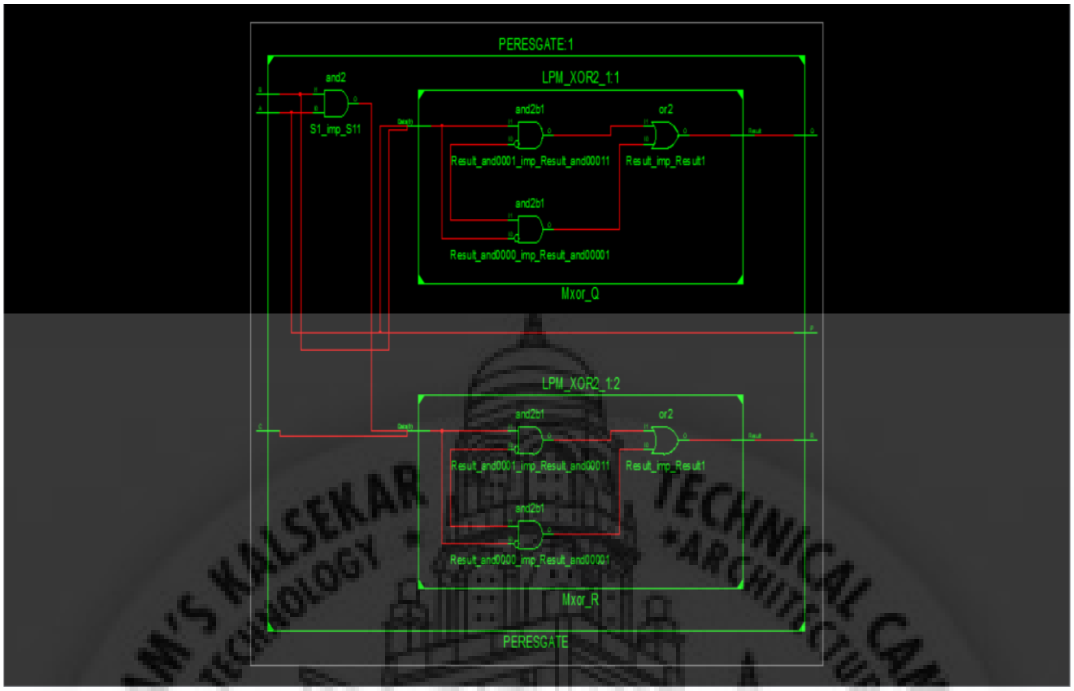
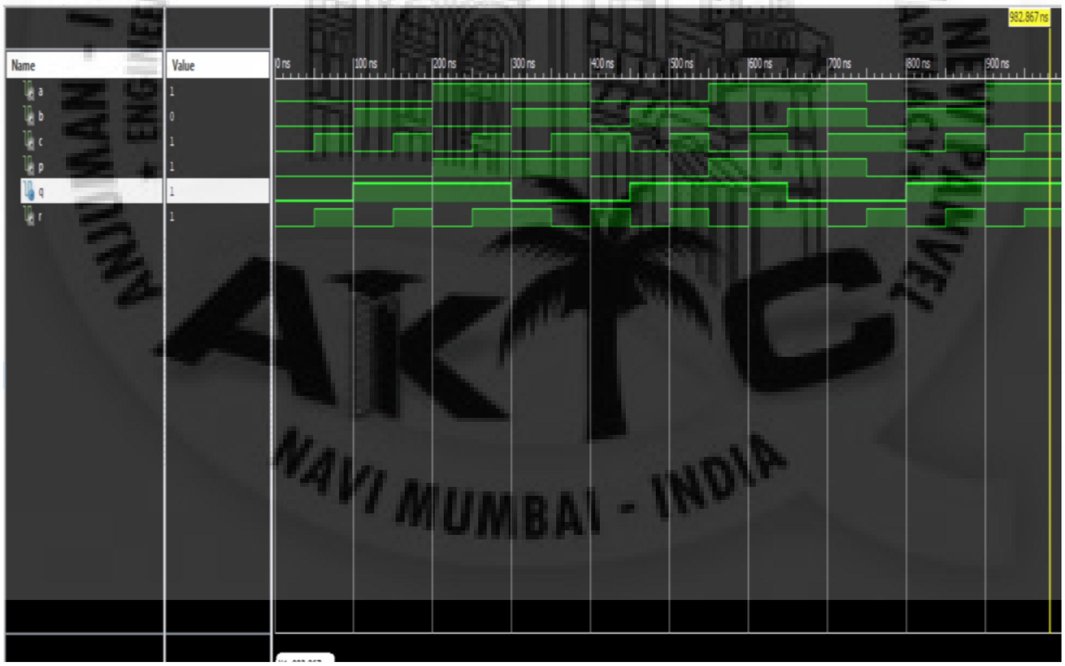


Fig. 4.2.1: 4-bit adder layout using peres gate.



Name	Value
a	1
b	0
c	1
p	1
q	1
r	1

Fig. 4.2.2: 4-bit adder output waveform using peres gate

The output is of 16-bit adder using HNG logic gate with the help VHDL programming and we observed the important parameter like time delay, I/O BUF, number of LUT used, memory usage, average fan-out & number of occupied slices. We compare the different parameter of conventional logic gate output and reversible logic gate using HNG like garbage output, quantum cost, no of input gates, and constant inputs.

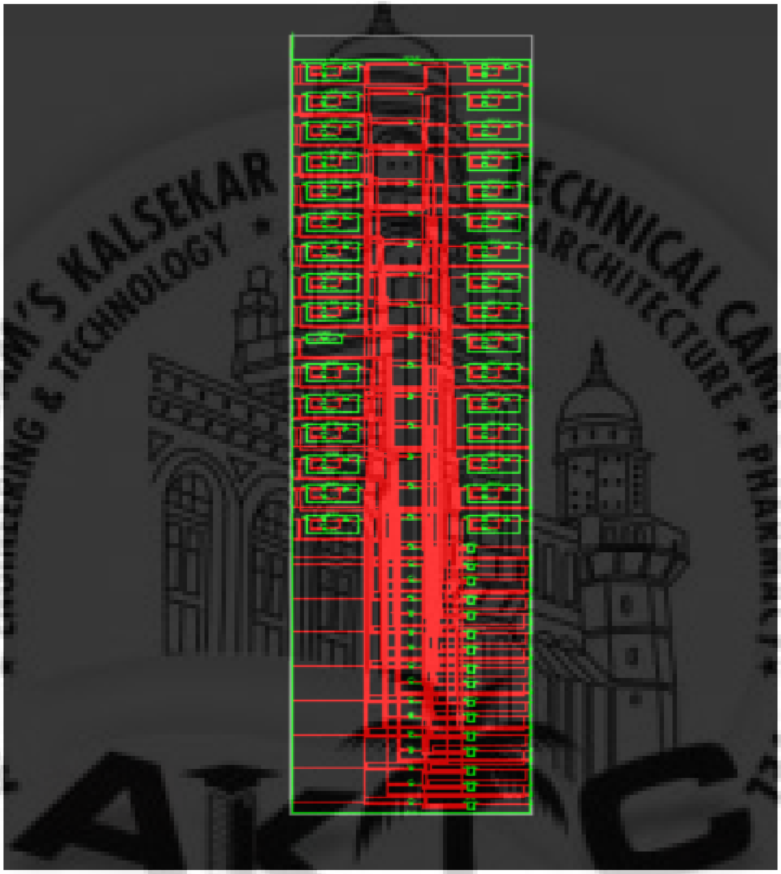
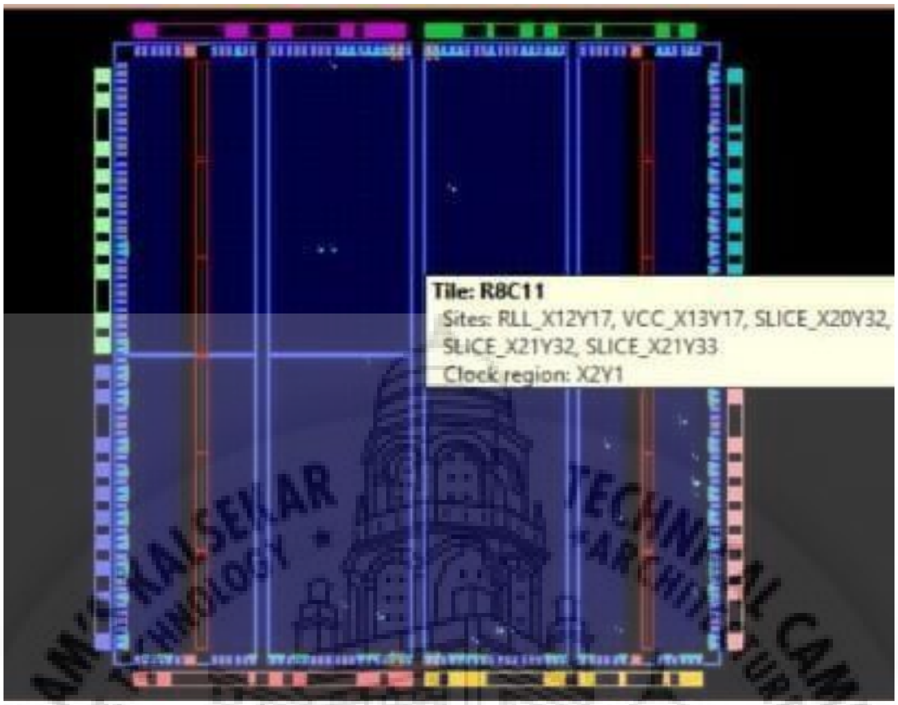
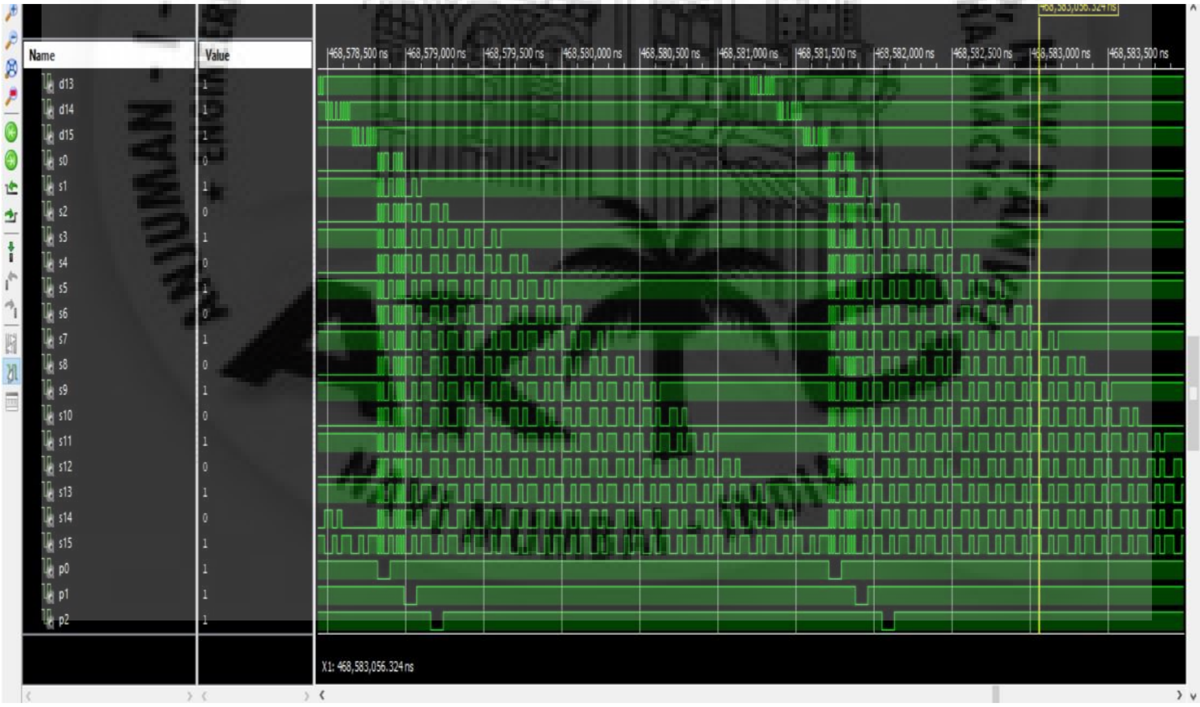


Fig. 4.3.1: 16-bit HNG Gate layout



Tile: R8C11
Sites: RLL_X12Y17, VCC_X13Y17, SLICE_X20Y32, SLICE_X21Y32, SLICE_X21Y33
Clock region: X2Y1

Fig. 4.3.2: 16-bit HNG Gate FPGA module



Name	Value
d13	1
d14	1
d15	1
s0	0
s1	1
s2	0
s3	1
s4	0
s5	1
s6	0
s7	1
s8	0
s9	1
s10	0
s11	1
s12	0
s13	1
s14	0
s15	1
p0	1
p1	1
p2	1

Fig. 4.3.3: 16-bit HNG Gate waveform

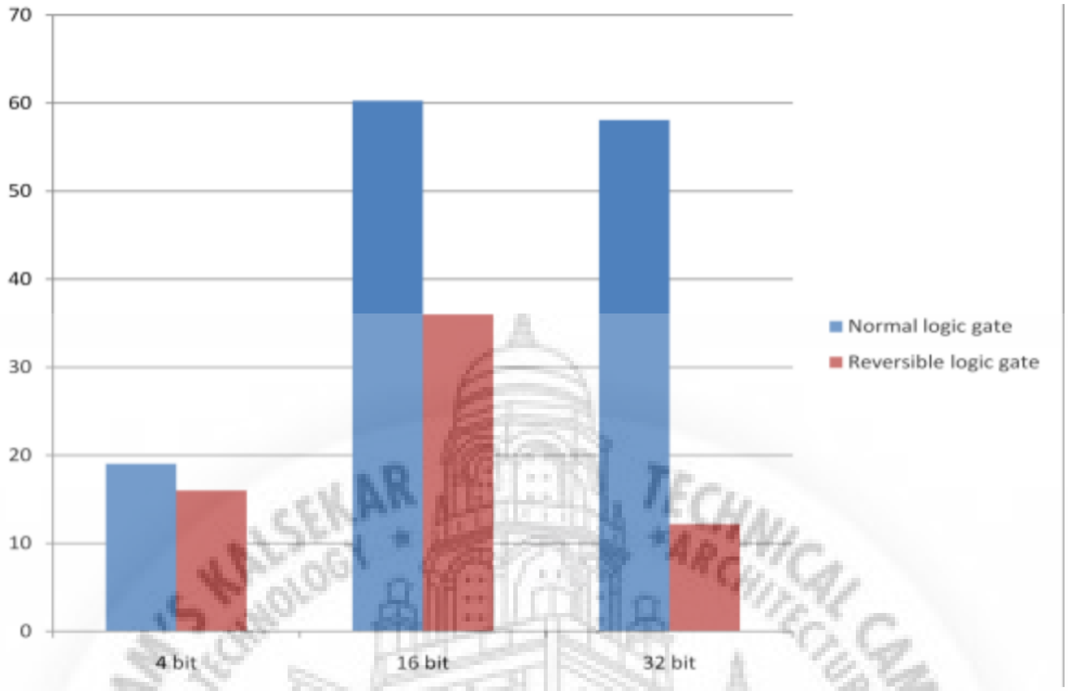


Fig. 4.3.4: Number of LUT used

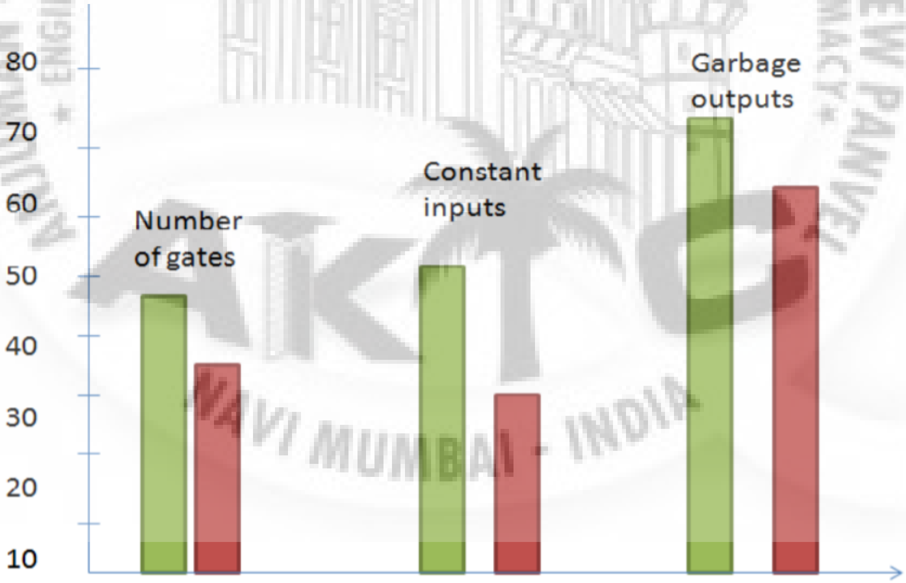


Fig. 4.3.5: Difference between conventional logic gate output and reversible logic gate

Program code

Conventional logic gate:

```
Entity adder 16 ia
Port ( Cin : in std_logic;
        A: in std_logic_vector(15 DOWN 0);
        B : in std_logic_vector(15 DOWNTO 0);
        S : OUT std_logic_vector(15 DOWNTO 0);
        Cout : out std_logic;
        Overflow: out std_logic);
        END adder 16;
Architecture Behaviour of adder 16 is
        Signal sum : std_logic_vector(16 downto 0);
        BEGIN
        sum <= ('0' & A) + B Cin;
        S <= sum (16 downto 0);
        Cout <= sum (16);
        Overflow <= sum (16) XOR A (15) XOR B (15) XOR sum (15);
End behaviour
```

Reversible logic gate:

```
Library ieee;
USE  ieee  std_logic_1164  all;
USE   ieee   numeric_std.   all;
Entity sixteen-bit-full-adder is
Port(    Cin: in std_logic;
         g0  :  in  std_logic;
         g1  :  in  std_logic;
         g2  :  in  std_logic;
         g3  :  in  std_logic;
         g5  :  in  std_logic;
         g6  :  in  std_logic;
```

```
g7  :  in  std_logic;
g8  :  in  std_logic;
g9  :  in  std_logic;
g10:  in  std_logic;
g11:  in  std_logic;
g12;  in  std_logic;
g13:  in  std_logic;
g15:  in  std_logic;
p0  :  in  std_logic;
p1  :  in  std_logic;
p2  :  in  std_logic;
p3  :  in  std_logic;
p5  :  in  std_logic;
p6  :  in  std_logic;
p7  :  in  std_logic;
p8  :  in  std_logic;
p9  :  in  std_logic;
p10:  in  std_logic;
p11:  in  std_logic;
p12;  in  std_logic;
p13:  in  std_logic;
p15:  in  std_logic;
C0  : out  std_logic;
C1  : out  std_logic;
C2  : out  std_logic;
C3  : out  std_logic;
C4  : out  std_logic;
C5  : out  std_logic;
C6  : out  std_logic;
C7  : out  std_logic;
C8  : out  std_logic;
C9  : out  std_logic;
```

```
          C10 : out std_logic;
          C11 : out std_logic;
          C12 : out std_logic;
          C13:  out  std_logic;
          C14:  out  std_logic;
          C15: out std_logic);

End sixteen-bit-full-adder:

Architecture behavioral of sixteen-bit-full-adder is
Begin
C0<=((g0 or p0) and cin);
C1<=((g1 or p1) and (g0 or p1) and p0 and cin);
C2<=((g2 or p2) and (g1 or p2) and p1 and (g0 or p2) and p1 and p0 and cin);
C3<=((g3 or p3) and (g2 or p3) and p2 and (g1 or p3) and p2 and p1 and (g0 or p3) and p2 and p1
and p0 and cin);
C4<=((g4 or p4) and (g3 or p4) and p3 and (g2 or p4) and p3 and p2 and (g1 or p4) and p3 and p2
and p1 and cin);
C5<=((g5 or p5) and (g4 or p5) and p4 and (g3 or p5) and p4 and p3 and (g2 or p5) and p4 and p3
and p2 and cin);
C6<=((g6 or p6) and (g5 or p6) and p5 and (g4 or p6) and p5 and p4 and (g3 or p6) and p5 and p4
and p3 and cin);
C7<=((g7 or p7) and (g6 or p7) and p6 and (g5 or p7) and p6 and p5 and (g4 or p7) and p6 and p5
and p4 and cin);
C8<=((g8 or p8) and (g7 or p8) and p7 and (g6 or p8) and p7 and p6 and (g5 or p8) and p7 and p6
and p5 and cin);
C9<=((g9 or p9) and (g8 or p9) and p8 and (g7 or p9) and p8 and p7 and (g6 or p9) and p8 and p7
and p6 and cin);
C10<= ((g10 or p10) and (g9 or p10) and p9 and (g8 or p10) and p9 and p8 and (g7 or p10) and p9
and p8 and p7 and cin);
C11<= ((g11 or p11) and (g10 or p11) and p10 and (g9 or p11) and p10 and p9 and (g8 or p11) and
p10 and p9 and p8 and cin);
```

```
C12<= ((g12 or p12) and (g11 or p12) and p11 and (g10 or p12) and p11 and p10 and (g9 or p12)
and p11 and p10 and p9 and cin);
C13<= ((g13 or p13) and (g12 or p13) and p12 and (g11 or p13) and p12 and p11 and (g10 or p13)
and p12 and p11 and p10 and cin);
C14<= ((g14 or p14) and (g13 or p14) and p13 and (g12 or p14) and p13 and p12 and (g11 or p14)
and p13 and p12 and p11 and cin);
C15<= ((g15 or p15) and (g14 or p15) and p14 and (g13 or p15) and p14 and p13 and (g12 or p15)
and p14 and p13 and p12 and cin);

End sixteen-bit-full-adder
```

Chapter 5

Conclusion

We have studied about conventional logic gates and different reversible logic gates and we implemented those gates on Xilinx software 14.7 using VHDL programming. We also implemented the 16 bit adder using conventional logic gates and observed the important parameters and we will implement the same using Peres reversible logic gates and compare the important parameters between them and we will burn the program on the FPGA board and observe the required outputs.

- **COMPARATIVE STUDY OF CONVENTIONAL LOGIC GATE OUTPUT VS REVERSIBLE GATE OUTPUT**

Parameters	**Conventional Gate**	**Reversible Gate**
No. of LUT's used	49	32
No. of occupied slices	25	24
Delay	42.138ns	27.102ns
Memory used	6372mb	4486mb
No. of I/O buffers	49	118
Average fanout	2.29	2.17

References

[1] R. Landauer, " Irreversibility and Heat Generation in the Computational Process", IBM Journal of Research and Development, vol.5,no.3, pp. 183-191, 1961.

[2] Gordon. E. Moore, "Cramming more components onto integrated circuits Electronics", Electronics magazine, vol 38, no. 8, pp.114, April 19, 1965.

[3] Bennett C.H., "Logical reversibility of Computation", IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532, 1973

[4] E. Fredkin, & T Toffoli, "Conservative Logic", International Journal of Theoretical Physics, vol.21, no.3, pp. 219-253, April 1982. [5] T. Toffoli, "Reversible Computing", Tech memo MIT/LCS/TM-151, MIT Lab for Computer Science, pp. 632-644,1980.

[6] A. Peres, "Reversible Logic and Quantum Computers", Physical Review A, vol. 32, no. 6, pp. 3266-3276, 1985.

[7] P. Shor, "Algorithms for quantum computation- discrete log and factoring", SFCS'94 Proceedings of 35th Annual Symposium On Foundations of Computer Science , IEEE Computer Society, Los Alamitos, pp. 124-134, 1994.

[8] H Thapliyal ,N Ranganathan, "Design of Efficient Reversible Binary Sub Tractors Based on a New Reversible Gate", IEEE Proceedings of the Computer Society Annual Symposium on VLSI, pp. 229-234,2009.

[9] H Thapliyal and N Ranganathan, "Design of Reversible Latches Optimized for Quantum Cost, Delay and Garbage Outputs", Proceedings of 17th IEEE International Conference on VLSI Design, pp. 235240,2010.

[10] Sujata S. Chiwande , Prashant . Yelekar, "Design of sequential circuit using reversible logic", IEEE International Conference Advances in Engineering, Science and Management, March30, 31, 2012.

[11] M. L. Chuang and C.Y. Wang, "Synthesis of reversible sequential elements", ACM journal of Engineering Technologies in Computing Systems (JETC), Article 19, pp. 1-19, vol. 3, no. 4, 2008.

[12] P. Picton, "Multi-valued sequential logic design using Fredkin gates," Multiple-Valued

Logic Journal, vol. 1, pp. 241-251, 1996.

[13] Abu Sadat Md. Sayem and Masashi Ueda, “Optimization of Reversible Sequential Circuits,” Journal of Computing, vol.2, no.6, pp. 208-214, 2010.

[14] SKS Hari, S Shroff, Sk Noor Mahammad and V. Kamakoti, “Efficient Building Blocks for Reversible Sequential Circuit design,” Proceedings of the International Midwest Symposium on Circuits and Systems, pp. 437-441,6-9 August, 2006.

[15] H M H Babu, M R Islam , A R Chowdhury, S M A Chowdhury, “ Synthesis of full-adder circuit using reversible logic”, 17th International Conference on VLSI Design, pp. 757-760, 2004.

[16] Ashis Kumer Biswas, Md. Mahmudul Hasan, Moshaddek Hasan, Ahsan Raja Chowdhury and Hafiz Md. Hasan Babu. “A Novel Approach to Design BCD Adder and Carry Skip BCD Adder”, 21st IEEE International Conference on VLSI Design, pp.566-571, 4-8 January, 2008.