An Efficient Fault Tolerant Design of Reversible Sequential Circuits

Submitted By
H.M.Tanzil
2010-3-60-00

Maksudul Hasan
2011-1-60-022

Imtiaz Jahan
2011-1-60-023

Supervised By
Md. Shamsujjoha
Lecturer
Dept. of CSE, EWU

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
EAST WEST UNIVERSITY

September 2015

Abstract

In this paper we proposed reversible synthesis for the several popular sequential circuits such as shift register, binary adder and Multipliers. All the proposed circuits are constructed with the parity preserving reversible gates. Thus, the proposed circuits inherently become fault tolerant. In addition, several lower bounds on the number of garbage outputs and constant inputs of the reversible fault tolerant sequential circuits have been proposed. It has also been shown that the proposed circuits are constructed with these optimal parameters. Moreover, the generalized algorithms for the fault tolerant sequential circuits have been presented. The performance study shows that the proposed fault tolerant circuits are much faster than the existing reversible non-fault tolerant counterparts.

Letter for Acceptance

This Project entitled "**An Efficient Fault Tolerant Design of Reversible Sequential Circuits** " submitted by H.M.Tanzil (ID:2010-3-60-004), Maksudul Hasan(ID: 2011-1-60-022), Imtiaz Jahan(ID: 2011-1-60-023) to the Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted by the department in partial fulfillment of requirements for the Award of the Degree of Bachelor of Science in Computer Science and Engineering on September, 2015.

Supervisor

____

Md. Shamsujjoha

Lecturer
Department of Computer Science and Engineering,
East West University, Dhaka, Bangladesh

Chairperson

____

Dr. Shamim Hasnat Ripon

Associate Professor and Chairperson,
Department of Computer Science and Engineering,
East West University, Dhaka, Bangladesh

Declaration by Candidates

We hereby declare that, the work presented in this project is to the best of our knowledge and belief, original, except as acknowledge in the text and that the material has not been submitted, either in whole or in part, for a degree at this or any other university.

H.M. Tanzil
(2010-3-60-004)

Maksudul Hasan
(2011-1-60-022)

Imtiaz Jahan
(2011-1-60-023)

Acknowledgements

First, I am thankful and expressing my gratefulness to Almighty who offers me divine blessings, patience, mental and psychical strength to complete this project. The progression of this thesis could not possibly be carried out without the help of several people who, directly or indirectly, are responsible for the completion of this work. I deeply indebted to my project supervisor Mr. Md. Shamsujjoha. His scholarly guidance, especially for his tolerance with my persistent bothers and unfailing support. He gives me the freedom to pursue aspects of reversible fault tolerant computing which I found interesting and compelling. This helped my project to achieve its desired goals.

I wish to thank the great people of Department CSE at East West University. A special thank goes to all faculties for their well-disposed instructions and Encouragements.

Finally, I would like to thank my friends and family. Their continued tolerance with my moods and tendency to disappear for weeks at a time gave me a much needed break from the world computing.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Reversible logic has captured significant attention in recent time as reducing power consumption is one of the main concern of digital logic. It consumes less power by recovering bit loss from its unique input-output mapping. In this paper, we propose a reversible fault tolerant shift register, binary adder and multiplier which is first ever proposed in literature. Those register is one of the most important parts of any central processing Unit. Several Researchers have been conducted to make the reversible fault tolerant gates, effectively. Reversible computation preserves the one- to-one mapping between input and output vector by adding garbage outputs. The research goals for any reversible computation are to perform computation with significant reduction of reversible gates and cost. Less number of constant input and garbage output. Fault tolerant reversible computation is relatively new for shift register, binary adder and multiplier.

1.1 Motivation

The number of outputs in a reversible gate or circuit must have equal number of input and of inputs must be equal [1]. Reversible computation can be performed reversibly both logically and thermodynamically which conduct to dissipating little energy. Reversible computing was founded when on the basis of thermodynamic of information processing it was shown that conventional irreversible circuits unavoidably generate heat because of losses of information during the computation. Various Scientists have undergone research on reversible logic like Landeuer’s and Benett. Landauer has undergone research on irreversible logic where each bit of information generates KTln2 Joules of heat energy[2]. Fault-tolerant describes a computer system or component designed so that, in the event that a component fails, a backup component or procedure can immediately take its place with no loss of service. A fault-tolerant system may be able to tolerate one or more fault-types including -- i) transient, intermittent or permanent hardware faults, ii) software and hardware design errors, iii) operator errors, or iv) externally induced upsets or physical damage. Fault-tolerant designs have been recovered from random faults occurring in hardware components. Two general approaches to hardware fault recovery have been used: 1) fault masking, and 2) dynamic recovery. Software Fault-Tolerance - Efforts to attain software that can tolerate software design faults (programming errors) have made use of static and dynamic redundancy approaches similar to those used for hardware faults. One such approach, N-version programming, uses static redundancy in the form of independently written programs (versions) that perform the same functions, and their outputs are voted at special checkpoints. An approach called design diversity combines hardware and software fault-tolerance by implementing a fault-tolerant computer system using different hardware and software in redundant channels. Each channel is designed to provide the same function, and a method is provided to identify if one channel deviates unacceptably from the others. At present Reversible logic is one of the most vital issue and it has different areas for its application like low power CMOS, quantum computing, nanotechnology, cryptography, optical computing, DNA computing, digital signal processing (DSP), quantum dot cellular automata, communication, computer graphics[3]. It is not possible to realize quantum computing without implementation of reversible logic. The main purposes of designing reversible logic are to decrease quantum cost, depth of the circuits and the number of garbage outputs [4]. Information is lost when the input vector cannot be uniquely recovered from its output vectors [5]. Reversible logic circuits naturally take care of heating since in a reversible logic every input vector can be uniquely recovered from its output vectors and therefore no information is lost. According to [5] zero energy dissipation would be possible only if the network consists of reversible gates. Thus reversibility will become an essential property in future circuit design. Reversible circuits are also interesting because the loss of bits of information implies energy loss [6].

1.2 Aims and Objectives

The objectives of this study are summarized below:

- To realize the reversible logic and reversible realization of any circuit in detail by examples and theories.
- To design a reversible fault tolerant sequential shift register which inherits properties of reversible logic, performs faster with less power consumption and it work in both falling and rising edge of the clock.
- To design a reversible fault tolerant binary adder which is used to design microprocessor to utilizing full adder and it dependent on both its previous state and clock.
- To design reversible fault tolerant serial multiplier.

1.3 Overview

This document presents the implementation of the internal architecture of reversible fault tolerant sequential circuits. We have proposed different components of fault tolerant reversible circuits with improvement in terms of cost comparing with the existing designs. The different components are reversible fault tolerant shift register ,and multiplier of fault tolerant reversible sequential circuits. The reversible fault tolerant binary adder are also proposed for designing utilizing full adder.

1.4 Methodologies of Research

While working on this research, the following important steps are followed:

- First, understanding of reversibility, its importance in low power circuitry, the basics of fault tolerance, its synthesis, the basics of quantum computation, its synthesis, various existing reversible and fault tolerant logic gates along with the quantum and transistor equivalent realizations etc.

- Designing various reversible and fault tolerant combinational circuits reversible sequential circuits, studying existing reversible sequential circuits design approaches, then analyzing the designs, working procedures, advantages and shortcomings.

- Inventing and contriving the ideas for the fault tolerant reversible logic and basic I/O of reversible sequential circuits under a general and scalable structure. Establishing the novelty of the proposed methods through theoretical explanations. Finally, performing a comparative study among the proposed and the existing works through simulations.

1.5 Outline

Chap.2 briefly discusses about the Existing reversible gate with their performance and evaluation.

Chap.3 discusses the Background study i.e we discuss about the related work of the topics and also briefly discuss about D-Flip-Flop.

Chap. 4 introduces several components of proposed fault tolerant shift register, proposed fault tolerant binary adder and proposed fault tolerant multiplier.

Chap. 5 finally discussed about Conclusions and Future work.

1.6 Summary

This chapter demonstrates motivations and objective of this thesis. Then the methodologies of the research that is being followed are discussed here. A brief elementary instructional text of remaining chapters of this thesis have also been described.

Chapter 2

Background Study

This chapter introduces the basic definition and properties which are related with reversible logic and reversible sequential circuits. In this chapter, reversibility is apprised and how reversibility prevents the loss of bit pattern is also shown as well as popular of reversible logic gates. Besides calculation of quantum cost of different reversible logic gate is also presented here.

2.1 Reversible Gate

An *n×n reversible gate* is a data stripe block that uniquely maps between input vector $Iv = (I_0,I_1, ..,.In-1)$ and output vector $Ov = (O_0 ,O_1, . . . ,On-1)$ denoted as $Iv \leftrightarrow Ov$. Two prime requirements for the reversible logic circuit are
as follows :

- There should be equal number of inputs and outputs.
- There should be one-to-one correspondence between inputs and outputs for all possible input-output sequences.

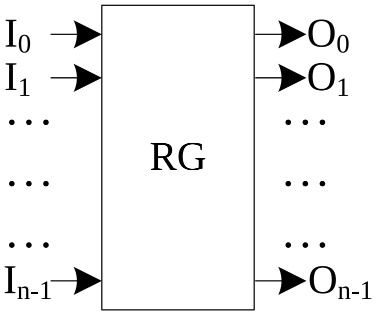


Fig. 2.1: A n × n Reversible Gate

2.2 Garbage Output

Every gate output that is not used as input to other gates or as a primary output is known as garbage. Unused output of a reversible gate (or circuit) is known as *garbage output*, *i.e.*, the output which are needed only to maintain the reversibility are the garbage output.

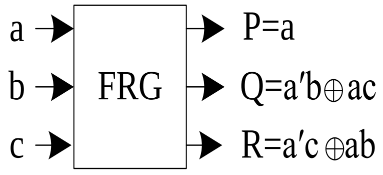


Fig. 2.2: A Reversible Gate with One Garbage Output

2.3 Quantum Cost

Every quantum circuit is built from 1 × 1 and 2 × 2 quantum primitives and its cost is calculated as a total sum of 2 × 2 gates used since 1 × 1 gate costs nothing i.e. zero. Basically the quantum primitives are matrix operation which is applied on qubits state. All the gates of the form 2 × 2 has equal quantum cost and the cost is unity i.e. 1 [7]. Since every reversible gate is a combination of 1 × 1 or 2 × 2 quantum gate, therefore the quantum cost of a reversible circuit calculates the total number of 2 × 2 gates used. The quantum cost of Fredkin Gate in Fig. 2 is 5 and the quantum cost of Feynman Double gate in Fig. 3 is 2.

2.4 Hardware Complexity

As mentioned earlier, determining the critical path from the large circuit is not easy. So, researchers determined the hardware complexity of the circuit and is considered as a seemingly equivalent performance evaluation criteria. The number of basic operations (Ex-OR, AND, NOT etc.) needed to realize the circuit is referred to as the hardware complexity of the circuit. Actually, a constant complexity is assumed for each basic operation of the circuit, such as, α for Ex-OR, β for AND, γ for NOT etc.

Then, total hardware complexity is calculated with respect to this assumed complexities, *i.e.*, in terms of α, β, and γ. For example, the hardware complexity of the circuit shown in Fig.2.is α, since it can be realized with a single Ex-OR only. Therefore, if a circuit only consists of x number of reversible Feynman gate, then the hardware complexity of that circuit is $x\alpha$.

2.5 Delay

The delay of a logic circuit is the maximum number of gates in a path from any input line to any output line. This definition is based on the following two assumptions [7]: Each gate performs the computation in one unit time. This means that every gate in the given circuit will take same amount of time for internal logic operations. All inputs to the circuit are known before the computation begins. Which means that the internal structure and each operation of the gate is known before the calculation. From the above definition, the delay of the logic circuit of Fig. 2.2 which consists of only one gate is obviously 1 as this is the only gate from its input to output line.

2.6 Fault Tolerant Gate

A fault tolerant Gate is a reversible gate which constantly preserves same parity between input and output, more specifically an $\boldsymbol{n} \times \boldsymbol{n}$ fault tolerant gate holds the following property:

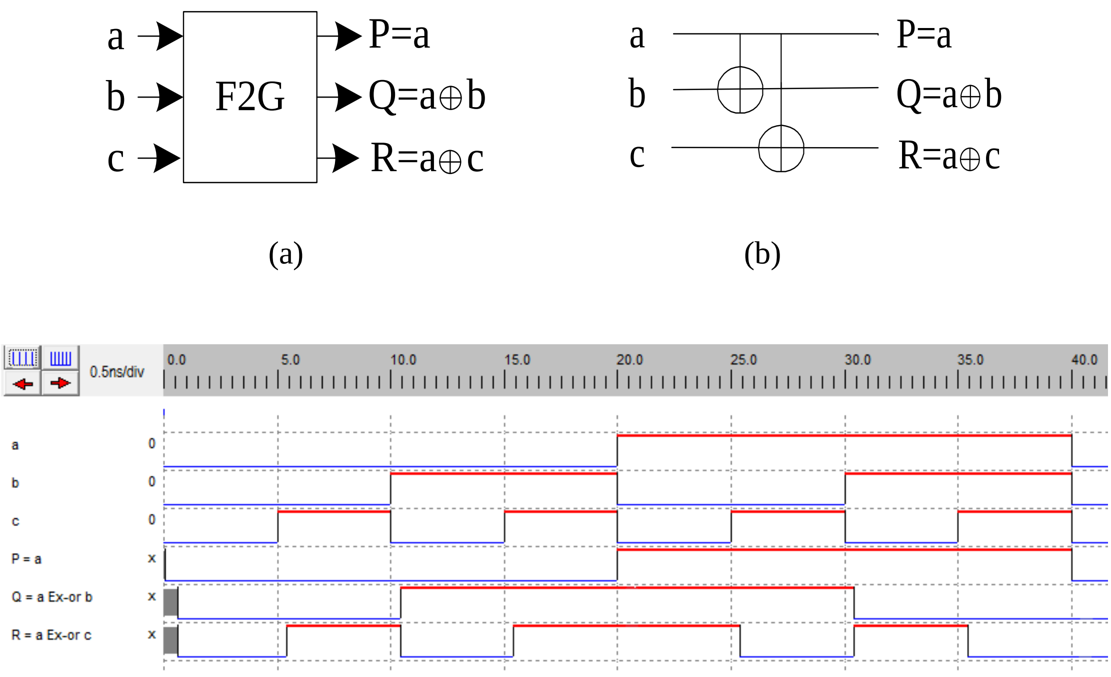
$$I_0 \oplus I_1 \oplus I_2 \oplus \ldots . I_{n-1} = O_0 \oplus O_1 \oplus O_2 \ldots .. O_{n-1} \qquad (2.1)$$

where $I_0, I_1, .., I_{n-1}$ are input vectors and $O_0, O_1, . . . , O_{n-1}$ are output vectors.
The circuit which consists of all fault tolerant gates must preserve the parity pattern in its input and output vectors.

2.7 Popular Reversible Fault Tolerant gate used for fault tolerant reversible sequential circuits

2.7.1 Feynman Double Gate

Input vector (*Iv*) and output vector (*Ov*) for 3 × 3 reversible Feynman double gate (*F2G*) is defined as follows [8]: $Iv = (a, b, c)$ and $Ov = (a, a\oplus b, a\oplus c)$. Block diagram of *F*2*G* is shown in Fig. 3(a). Fig. 3(b) represent the quantum equivalent realization of *F*2*G*. From Fig. 3(b) we find that it is realized with two 2×2 Ex-OR gate, thus its quantum cost is two. Fig. 3(c) represents the corresponding timing diagram of *F*2*G*.



(a)

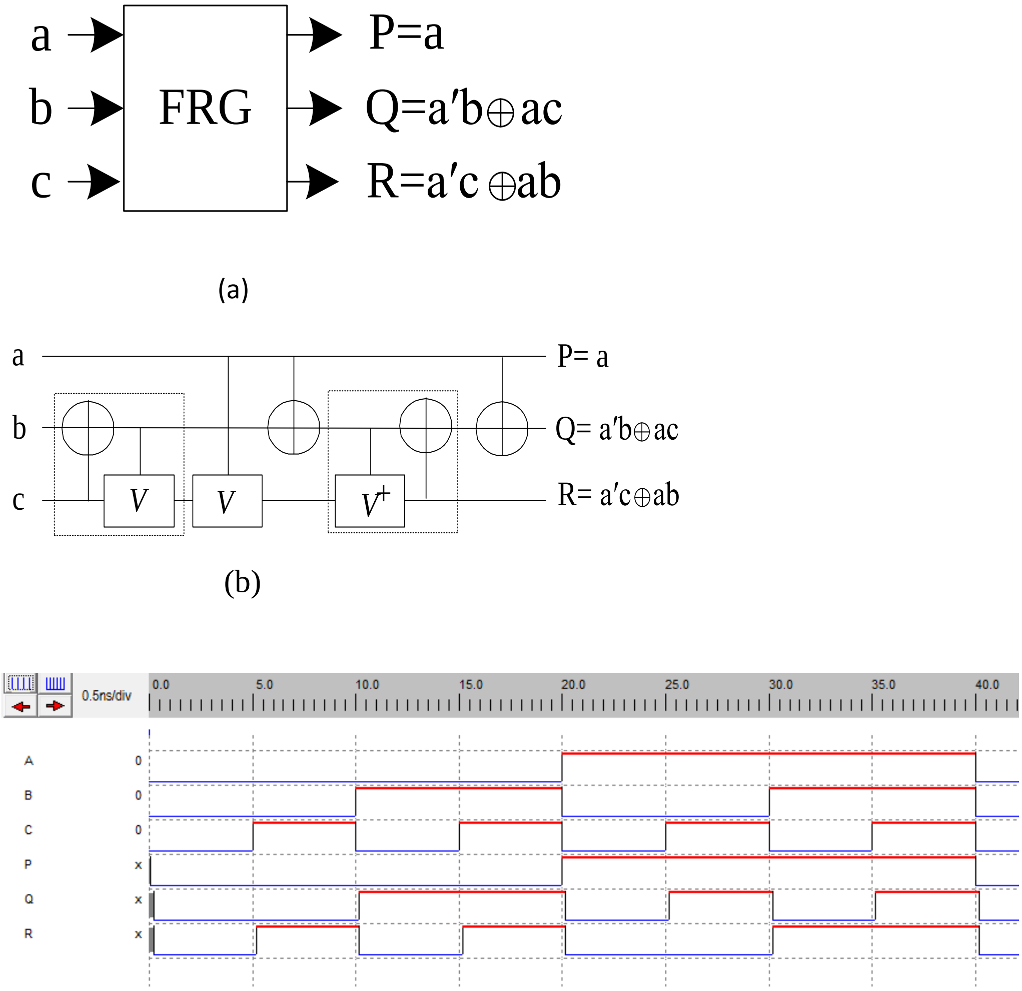
(b)

(c)

Fig. 2.3: Reversible 3x3 Feynman double gate (a) Block diagram (b) Quantum equivalent realization (c) Timing Diagram

2.7.2 Fredkin Gate

The input and output vectors for 3 × 3 Fredkin gate (*FRG*) are defined as follows [9]: $Iv = (a, b, c)$ and $Ov = (a, a'b \oplus ac, a'c \oplus ab)$. Block diagram of *FRG* is shown in Fig. 4(a). Fig. 4(b) represents the quantum realization of *FRG*. In Fig. 4(b), each rectangle is equivalent to a 2 × 2 quantum primitives, therefore its quantum cost is considered as one [10]. Thus total quantum cost of *FRG* is five. To realize the *FRG*, four transistors are needed as shown in Fig. 2(c) and its corresponding timing diagram is shown in Fig. 4(c).



(a)

(b)

(c)

Fig. 2.4: Reversible 3×3 Fredkin gate (a) Block diagram (b) Quantum equivalent realization (c) Timing Diagram

Reversible Fredkin and Feynman double gate obey the rule of Eq.2.1. The fault tolerant (parity preserving) property of Fredkin and Feynman double is shown in Table. 2.1

TABLE 2.1: Truth table for *F*2*G* and *FRG*

Input				Output of F2G				Output of FRG				Parity
A	B	C		P	Q	R		P	Q	R		
0	0	0		0	0	0		0	0	0		Even
0	0	1		0	0	1		0	0	1		Odd
0	1	0		0	1	0		0	1	0		Odd
0	1	1		0	1	1		0	1	1		Even
1	0	0		1	1	1		1	0	0		Odd
1	0	1		1	1	0		1	1	0		Even
1	1	0		1	0	1		1	0	1		Even
1	1	1		1	0	0		1	1	1		Odd

Among the reversible gates discussed above, Fredkin and Feynman double gate comply with the rule of Eq.2.1. Therefore, according to our previous discussion in Sec.2.1, if a circuit is designed using only Fredkin and Feynman double gates the circuit will inherently become fault tolerant. The parity preserving (fault tolerant) property of Fredkin and Feynman double is shown in Table 2.1

2.8 Summary

A brief literature overview and the related terminologies regarding reversible and fault tolerant logic synthesis are presented in this chapter. Definitions of two most popular reversible and fault tolerant logic gates devoted here as well. Equivalent quantum and transistor representations of those reversible and fault tolerant logic gates have also been depicted in this chapter.

Chapter 3 Related Work

3.1 Related Works

In recent years, many reversible gates and circuits have been proposed as an alternative to conventional irreversible circuits [11, 12, 13, 14]. Examples of these circuits are adders, sub tractors and multipliers. Different designs of reversible full adders and adder circuits have been proposed in [11,15,16,17]. Other arithmetic operations such as sub tractors, multipliers, dividers as well as sequential arithmetic units which are the essential blocks of the computer systems have been proposed in recent years [18, 19, 20]. As sequential elements are the basic blocks of todays computer, a veriety of sequential reversible elements have been introduced to date. Fredkin and Toffoli[21] are the first who suggested the reversible design of sequential elements however they didn't discuss the construction of basic elements in reversible logic. They also compared the conventional and reversible sequential networks and claimed that reversible logic could possibly preserve the computing capabilities of ordinary digital logic while satisfying the physical constraints of reversible and conservation. They showed a conservative logic realization of J-K flip flop and a serial adder and introduced a different meaning of delay concept in these circuits. As the first attempts at the realization and designing of sequential reversible circuits were made by Picton [22],constructing a reversible circuit for SR latch. Thapliyal,et.al[23]suggested other designs of reversible sequential elements with reversible logic synthesis of flip flops. The Flip Flops that are synthesized are RS,J-K,D,T and Master Slave Flip Flops. Rice[24] also discusses the construction of reversible basic memory elements including latches and Flip Flops. It uses Toffoli gate as the basic building block of the Flip Flops. Considering sequential reversible circuits, more recent work on designing reversible serial adder/ subtractor has been published by Krishnaveni et al.[25]. They designed D Flip Flop and shift register by the help of a reversible gate called SRK gate which is used in the reversible serial adder/sub tractor.

We consider some criteria in order to compare the results of our proposed approach with the existing literatures. The cost of a reversible circuit is determined by five parameters in the literature, number of garbage output, number of constant inputs, quantum cost, number of gates and delay. Number of garbage output refers to the number of outputs that are neither used as primary outputs nor for further computations, but they are added to make the circuits reversible[18,20,26]. Number of constant inputs refers to the number of inputs which their values are not to be changed in a given circuit and have to be either 0 or 1 in order for the circuit to work. They are also added to make the circuit reversible[20,26,27]. Number of gates refers to the total number of reversible gate in a given circuit. The quantum cost is defined as the number of 1×1 or 2×2 reversible quantum or logic gates that are required to realize the circuit[25]. Table 1 shows the quantum cost of some of the different reversible gates used in the literature. Delay is another parameter recently considered in many reversible circuits papers. Some articles have realized the delay as number of reversible or quantum gates in the critical path of the circuit[23,28] which is an optimistic assumption. To be more precise, we should consider the different delays in each quantum gate separately. According to [29] ,the delay of basic quantum gates, i.e. 1×1 and 2×2 reversible gate is considered and the delay of other quantum circuits is equal to the number of delay levels of the gate which is level can be realized using basic quantum gate.

3.2 Sequential D Flip Flop

The D flip-flop tracks the input, making transitions with match those of the input D. The D stands for "data"; this flip-flop stores the value that is on the data line. It can be thought of as a basic memory cell. A D flip-flop can be made from a set/reset flip flop by tying the set to the reset through an inverter. The result may be clocked. Our proposed Flip Flop is more efficient one then general D Flip Flop.

Fig. 1 represents the architecture of proposed reversible fault tolerant D flip flop. We used a Fredkin and a Feynman double gate in this circuit. The Fredkin gate is needed to produce Q, whereas Feynman double gate produces Q^+ and Q. The equations for the proposed fault tolerant flip flop are Q_n = D.Clk + Clk'.Q_{n-1} and Q^+ = Q' (n is time varying), which are used to capture the logic level when the data is present and the clock is set.

Corresponding timing diagram of the proposed flip flop is shown in fig. (2). The circuit is simulated with the average current of 0.001 mA, average power of 0.001mW and average delay of 0.235 ns. From fig (2), we find that when clock is low the value of D has no effect on Q(0-2.5 ns). However, when clock is high Q follows D, i.e., are in transparent mode (2.5-7.5 ns). If the clock is low again, then Q retains the previous value of D, i.e., the value of D when the clock went from high to low (1→0) last time (7.5 to 10 ns, 15 to 17.5 and so on).

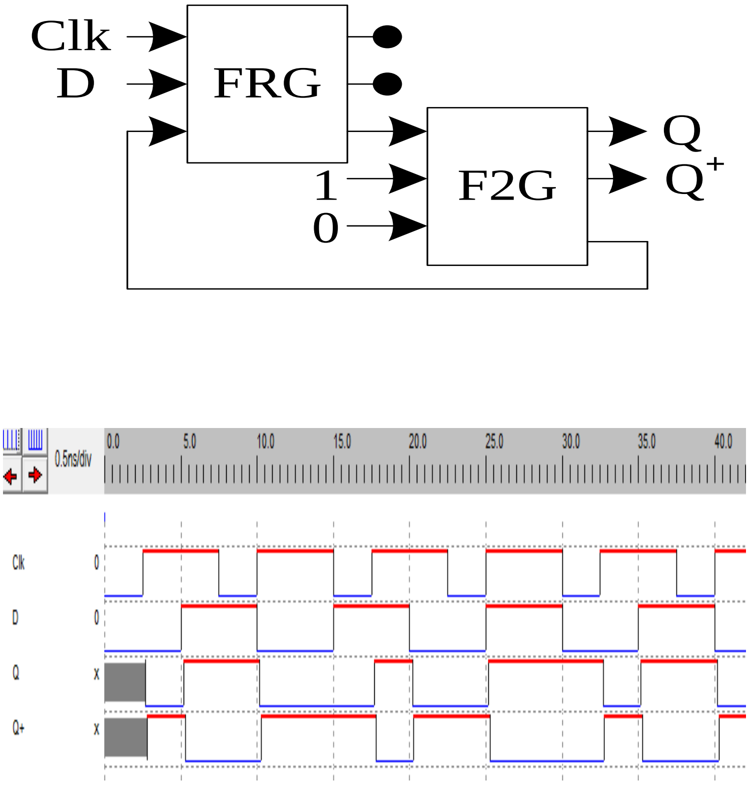


Fig.3.1 (a) Block diagram of proposed reversible fault tolerant D flip flop.3 (b) Timing diagram of proposed reversible fault tolerant D flip flop.

3.3 Summary

This chapters we discuss about the related works of the reversible fault tolerant sequential circuits. And we also used proposed reversible fault tolerant D flip flop. This D-Flip Flop used for our reversible fault tolerant sequential circuits.

Chapter 4

Proposed Reversible Fault Tolerant Circuits

4.1 Proposed Fault Tolerant Sequential Shift Register

Many sequential circuits use shift register to read the inputs and store the intermediate and final results such as serial adders and multipliers. In this section a reversible version of shift register is introduced and is use to construct our purposed reversible serial adder and multiplier. First we use the purposed D flip flop with the capability to sample the input in falling edge of the clock, which is two time faster than the other reversible shift register including. Secondly, by just eliminating the ”shift left” capability which is not actually required, the proposed reversible shift register has considerably reduced the cost of the circuit including quantum cost and number of gates as seen as figure 4 of the proposed reversible shift register. In this circuit in order to have a 4 bit shift register It requires seven 2:1 multiplexers for D flip flop and 4 Feynman gates. Except for the first bit which requires just one multiplexer, the other bits come with two multiplexers. Table 4 shows the operating modes of the reversible shift register.

Table 4.1: Controlling signals of the reversible fault tolerant shift register

S_1	S_0	Operation
0	0	Load
0	1	Previous state
1	0	Shift
1	1	Previous state

For the first bit there Is no different between load and shift. If the S0 is zero it loads with the input data from left. So it works as load, independent of the value of S1. If the S1 is 1, the input data is not really important and it just loads the first bit with a new value. It turns out that the next clock signal is available in the output of each flip flop received the correct output from the previous stage. This is the point which helps in reducing the required element. To show this we assume that in each flip flop the delay until the last stage Fredkin gate is T1 where the delay of the flip flop is T. it is clear that T1<T and each flip flop receives the clock signal from the previous stage at multiples of T1; n×T1 where n is the flip flop number in the range from one to four. But the shifted has

Table 4.2: Comparative results of reversible fault tolerant shift registers without considering the cost of D flip flops

Reversible shift register	Quantum cost	Constant inputs	Garbage outputs	No. of Gates
Proposed design	43	8	12	11
Existing Design [29]	70	10	14	22

becomes available in each flip flop after the last Fredkin gate at (n-1)*T1+T where n is the flip flop number in the range from one to four. Therefore at each flip flop the clock signal becomes active T –T1 time units earlier before the altered data from the previous stage arrived the output. This means that each flip flop receives the correct data from the previous stage to shift it to the next stage.

Table 5 shows the comparative results of our reversible shift register and the one in [29]. Since our proposed designed use a different D flip flop with additional capabilities, it cannot be compared with other existing literature. For example the specified delay is the delay of one bit shift of the register and our proposed shift register can do twice the shifts in every clock's duration. Also the delay of the multiplexers is not taken into account and realistically we consider the same flip flop in two designs and eliminate it in our evaluation and comparison.

One of the most remarkable characteristics is that our proposed shift register can work in edge of the clock and this property has not led a large quantum cost and garbage outputs and has also not led to significant inputs and number of gates compared to the existing literature.

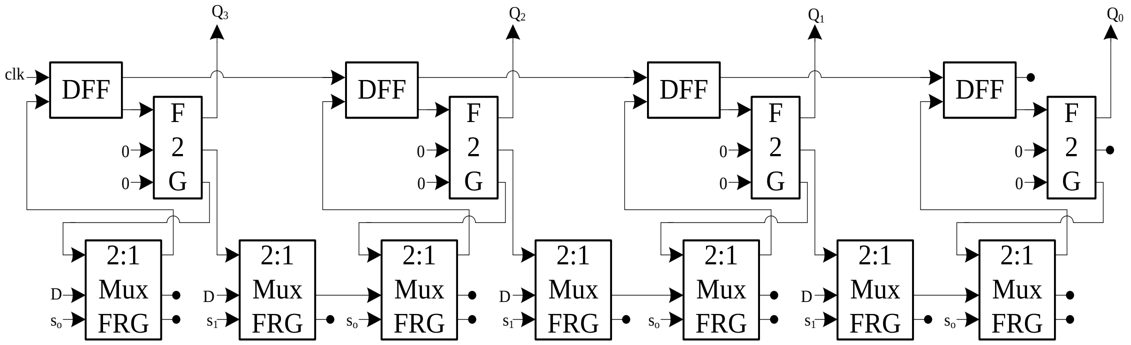


Fig4.1: Proposed reversible fault tolerant shift register

Lemma1: An *n* reversible fault tolerant shift register can be realized with (5*n*-1) gates and (19*n*-5) quantum cost.

Proof: According to our design procedure, an *n*-bitreversible fault tolerant binary adder requires *n* numbers of reversible fault tolerant D flip flop. Every D flip flop requires one Fredkin gate and one Feynman double gates, *i.e.*, two gates in total and every fault tolerant reversible shift register requires one D flip flop, one Feynman Double gate and one 2:1 Fredkin Mux, moreover to add every single bit shift register need one extra 2:1 Fredkin Mux. Thus, the total number of gates required for the reversible fault tolerant sequence counter is

(2+1+1)*n*+(*n*-1) = 5*n*-1.

As shown earlier quantum cost of a reversible fault tolerant D flip flop is 7, Feynman Double gate is 2 and Fredkin gate is 5. So, total quantum cost for n reversible fault tolerant shift register is (7+2+5)*n*-(5*n*-5)=19*n*-5.

Algorithm 1: Algorithm for proposed reversible fault tolerant sequential n-bit shift refister.

Input: Data input set s(s_0,s_1) Feynman double gate**(F2G)** and Fredkin gate**(FRG)**

Output: n-bit reversible fault tolerant sequential circuit

```
1   begin
2          I = input
3         0 = output
4          For x→0 to n-1 do
5               Clk → first.i.DFF
6          For j ← 0 to n-1 do
7               first.i.F2G ← second.o.DFF
8               0 → second.i.F2G
9               0 → third.i.F2G
10             If s=0 then
11                first.i.FRG ← third.o.F2G
12                second.i.DFF ← first.o.FRG
13           end if
14              else
15                first.i.FRG ← second.o.F2G
16           end if
17                  D → second.i.FRG
18           end for
19      end for
20      return  F2G.o.1 → desired outputs
21                remaining FRG.o → garbage outputs
22  End
```

Algorithm 1 presents the design procedure of the proposed n-bit reversible fault tolerant shift register. Line 5 of the algorithm assigns the input to the DFF for the control bit clk. Whereas line 7-9 assigns first, second and third output of F2G for all the remaining control bits. Line 11 assign the first input of FRG and line 12 assign the second input of DFF when s=0. Line 15 and 17 of the algorithm manually assign the first input of FRG and second input of FRG. End of the algorithm line 20-21 returned the desired outputs and garbage. The complexity of the algorithm in zero. To the best of our knowledge, it is one of the simplest algorithm of n-bit reversible fault tolerant sequential decoder.

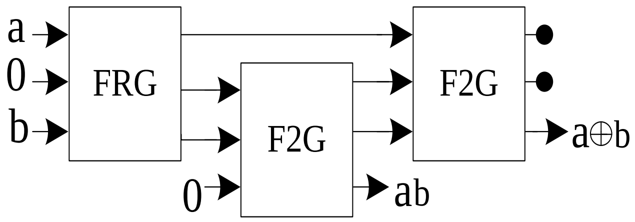
Another remarkable point to note is that we used 2×2 reversible multiplexers instead of 4×4 reversible multiplexer. So looking from the perspective of extra elements required for shift register, our proposed shift register uses seven Feynman gates, seven Fredkin gates and D flip flops, which is a reduction compared to the existing literatures. Considering the fan-out problem, S1 and S0 signals can be passed from the previous multiplexer to the next one.

4.2 Proposed Reversible Fault Tolerant Binary Adder

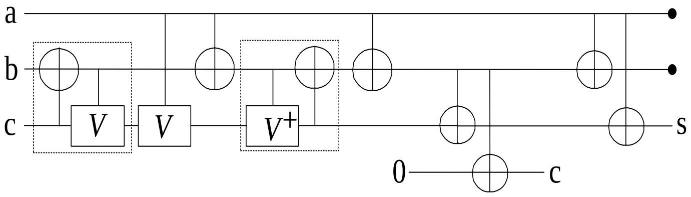
This section illustrates the design layouts, working procedures, transistor and quantum equivalent realization of the proposed reversible fault tolerant binary adder. The proposed adder for a reversible fault tolerant microprocessor is designed utilizing full adder. The working procedure of the proposed full adder is dependent both on its previous state and clock. There are several designs of reversible binary adder in the literature among which the design is considered to be the most compact and efficient. Here, initially we propose a reversible fault tolerant half adder. Then the proposed adder schema is used to design the fault tolerant adder.

4.2.1 Proposed Reversible Fault Tolerant Half Adder

Fig. 4.2(a) represents the architectural block diagram of the proposed reversible fault tolerant half adder. The corresponding quantum equivalent realization is shown in Fig. 4.2(b) From the quantum presentation we find that the proposed half adder is constructed with total of nine 2×2 quantum equivalent gates. Thus, its total quantum cost should be nine. But, in this quantum realization there are two consecutive Ex-OR gates, one from *FRG* and the other one from *F*2*G* and its I/O are from identical levels. There are two consecutive Ex-OR gates with identical I/O level then it only represent a quantum wire and without any quantum cost. Thus, the total quantum cost of the proposed fault tolerant half adder is 9 because of the reduction of two consecutive two Ex-OR operations.



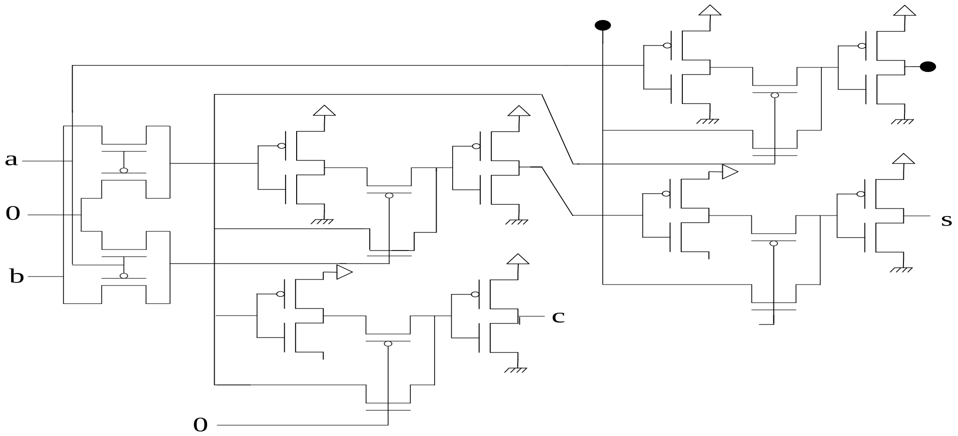
(a)



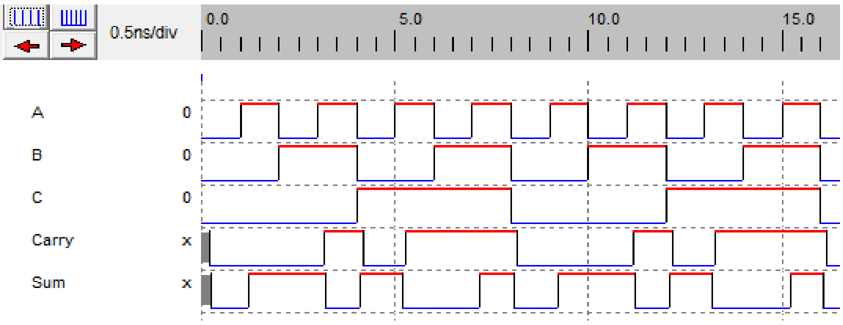
(b)

Fig 4.2*: Proposed reversible fault tolerant half adder (a) Block Diagram (b) Quantum realization*

Throughout the paper we consider the signal less than 0.01ns stability are glitches and thus omitted from the simulation results. Moreover, considering these gates representation as schema all the proposed circuit of Sec. 3 is designed.



(a)



(b)

Fig 4.3: Proposed reversible fault tolerant half adder (a) Transistor Realization (b) Timing Diagram

Fig. 4.3(a) represents transistor realization of the proposed reversible fault tolerant half adder. The corresponding simulation result is shown in Fig. 4.3(b) which proves the functional correctness of the proposed circuit. From Fig. 4.3(b), we find that the output carry *and* sum depends on both the value A and B.. Since, the reversible gates that has been used to design the proposed flip flop preserves are fault tolerant. Hence, the proposed half adder inherently becomes fault tolerant. To the best of our knowledge, fault tolerant design of reversible half adder is a novel approaches in the literature.

***Theorem 1*:** A reversible fault tolerant half adder can be realized with at least two garbage outputs and two constant inputs.

Proof: Fig. 3 is the proof for the existence of a fault tolerant Half Adder with two garbage outputs and two constant inputs. Next, we want to prove that it is not possible to realize the reversible fault tolerant half adder with fewer than two garbage outputs and two constant inputs. Truth table of Half Adder is shown in Table 3. In this table, we find that different input combinations map to identical outputs. For example, both input combinations (0,1) and (1,0) have the output (0,1). This is straight violation of one-to-one mapping property of reversible logic synthesis. So, at least one constant input is needed to realize the reversible Half Adder, which produces one garbage output. The value of this extra constant input can be either 0 or 1. Table 4 and Table 5 show the corresponding two trut tables with the value of this constant input and two garbage outputs.

Table 4.3: Truth table of a half adder

Input		Output	
A	*b*	*Carry*	*Sum*
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

From Tables 4.4 and 4.5 we find that, it maintains the one-to-one property of reversibility3. However, if we consider the left-portioned value of the garbage output as its corresponding value, then the shaded rows of Table 4.4 (rows 1,2) and shaded rows of Table 4.5 (rows 3,4) violate parity preserving rule of Eq.(1)(discussed earlier in Sec. 2.1). On the other hand, if we consider the right portioned value of the garbage output as its corresponding value, then the un-shaded rows of Table 4.4 (rows 3,4) and un-shaded rows of Table 4.5 (rows 1,2) don't maintain the parity preserving property of fault tolerant logic synthesis. Therefore, at least one more constant input is needed to preserve the parity *i.e.*, two constant inputs in total. This extra constant input will produces one more garbage output i.e., three garbage output in total. It justifies our claims of Theorem1.

In these tables, E=Even, O=Odd, *Ip*=Input parity, *Op*=Output parity, C=Constant input and GO=Additional Garbage Output

Table 4.4: Truth table for reversible fault tolerant half adder with one constant input which is set to lowand two garbage outputs

Input			Output			Parity
C	*a*	*B*	*Carry*	*Sum*	*GO*	
0	0	0	0	0	0/1	I_p = *E* O_p=*E/O*
0	0	1	0	1	1/0	I_p = *O* O_p=E/O
0	1	0	0	1	0/1	I_p = *O* O_p=*O/E*
0	1	1	1	0	1/0	I_p = *E* O_p=*E/O*

Table 4.5: Truth table for reversible fault tolerant half adder with one constant input which is set to high and two garbage outputs

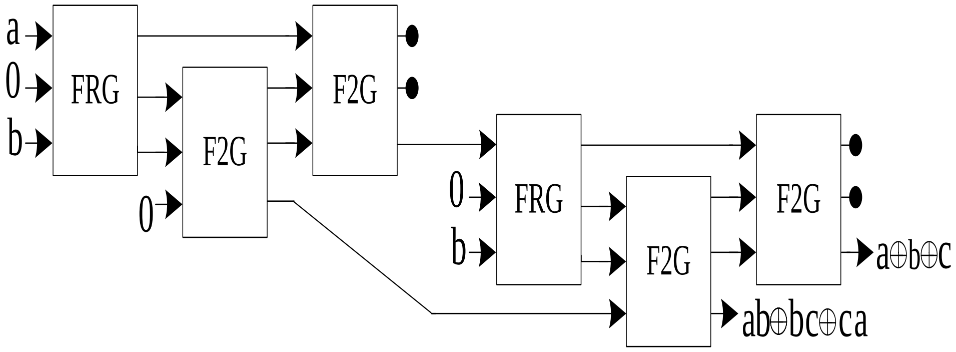
Input			Output			Parity
C	*a*	*b*	*Carry*	*Sum*	*GO*	
1	0	0	0	0	0/1	I_p=*O* O_p=*E/O*
1	0	1	0	1	1/0	I_p=*E* O_p=*E/O*
1	1	0	0	1	0/1	I_p=*E* O_p=*O/E*
1	1	1	1	0	1/0	I_p=*O* O_p=*E/O*

4.2.2 Proposed Reversible Fault Tolerant Full Adder

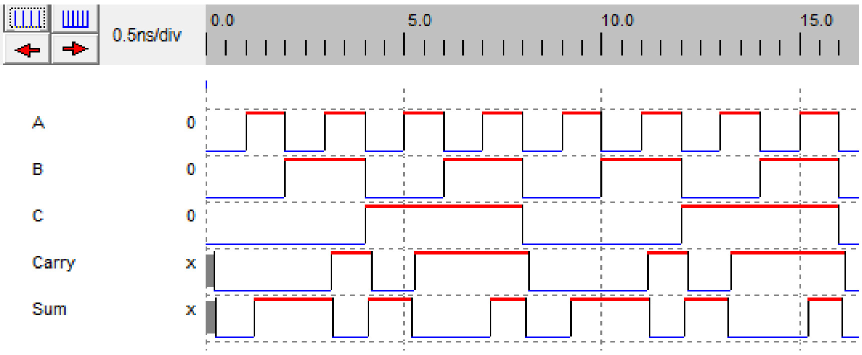
For fault tolerant binary adder we need fault tolerant binary full adder. In this paper we proposed an optimized fault tolerant binary adder.

Corollary: A reversible fault tolerant Full Adder can be realized with at least 3 constant input and 4 garbage output.

Proof: We know that a full adder can be realized with two Half Adder and some other circuits. In theorem 1 we proved that a fault tolerant half adder could be realized with 2 constant input and 2 garbage outputs. In fig 5 we showed that one output of first half adder is used as an input of second half adder, so we have 1 less constant input in second half adder. Figs. 4.4(a) and 4.4(b) represent the architecture of fault tolerant full adder and its corresponding simulation result, respectively.Table. 5shows the comparison of proposed half adder with the existing non-fault tolerant reversible half adder. Generally, fault tolerant design cost more than the non-fault tolerant design. But from Table. 5, we find that the proposed design performs much better than the existing non fault tolerant designs and are much scalable^{2}.



(a)



(b)

Fig 4.4: *Proposed reversible fault tolerant full adder (a) Block Diagram (b) Timing Diagram*

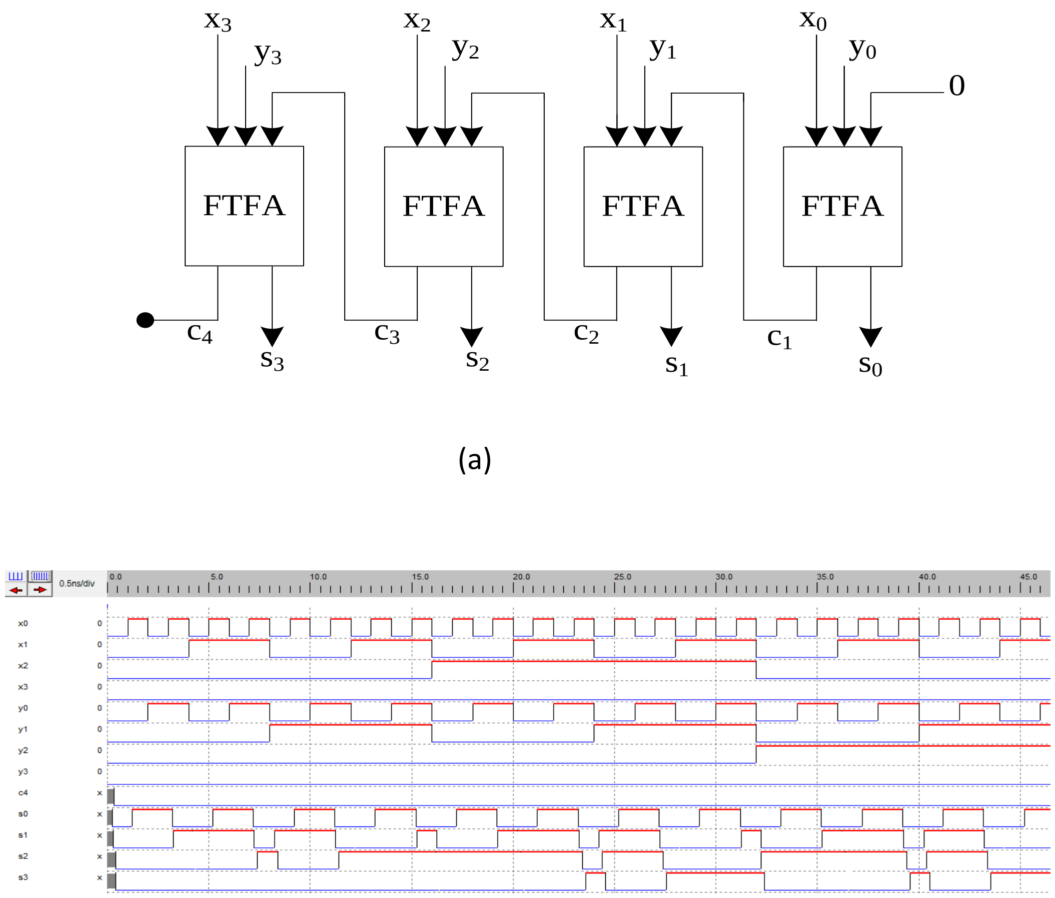
Table 4.6: Comparison of reversible fault tolerant full adder

Full Adder	QC	HC	CI	GO
Existing Circuit [30]	30	$18\alpha + 16\beta + 10\gamma$	9	11
Existing Circuit [31]	18	$12\alpha + 8\beta + 2\delta$	5	6
Proposed Circuit	18	$12\alpha + 8\beta + 2\delta$	3	4

4.2.3 Proposed Reversible Fault Tolerant Binary Adder

In this section we described proposed fault tolerant binary adder. The combination of four reversible fault tolerant full adder schemas can work as a 4-bit Reversible Fault tolerant binary adder. In previous section we proposed an optimized fault tolerant full adder.

QC=Quantum cost, HC=Hardware Complexity, CI=No of Constant Input and GO=No of Garbage Output. We also consider that α, β *and* γ are the hardware complexities of two-input Ex-OR, AND and NOT calculations, respectively.



(a)

(b)

Fig 4.5*: Proposed reversible fault tolerant binary adder (a) Block Diagram (b) Timing Diagram*

Figs. 4.5(a) and (b) represent the architecture of proposed fault tolerant binary adder and its corresponding simulation result, respectively. Carry of every full adder should be used as an input of its next full adder. As there is no carry for first full adder we have to use constant input 0 as an input and the last carry should be counted as a garbage output. Every sum should be counted as an output. *S0, S1, S2, S3* should be the final result of 4-bit binary addition. From the simulation results we find that the initial clock transition occurs at 1*ns* and the output becomes available in 1.637*ns*. Thus, the maximum possible delay of the proposed circuits is less than0.65*ns*. The average power consumption is less than0.0015_W.

Algorithm 1 represents the design procedure of the proposed n-to-n reversible fault tolerant binary adder. Line 5 of the algorithm assigns the second input and line 6 of the algorithm assigns the third input of the FTFA gates. Line 8 and 11 assign the first input of the FTFA gates, when k=0 or else. End of the algorithm line 14 and 15 returned the desired outputs and garbage. To the best of our knowledge, it is one of the simplest algorithms of n-to-n reversible fault tolerant binary adder.

Lemma 2**:** An *n-bit* reversible fault tolerant binary adder can be realized with *6n* gates, *18n* quantum cost and (*12n_+8n_+4n*) hardware complexity, where *nϵN;n* _ 2 and _, _, are the hardware complexity of two input AND, OR, NOT calculation, respectively.

Proof: According to our design procedure, an *n*-bit reversible fault tolerant binary adder requires *n* numbers of reversible fault tolerant full adder. As shown in earlier, a reversible fault tolerant full adder requires two half adder. Every half adder requires one Fredkin gate and two Feynman double gates, *i.e.*, three gates in total. Thus, the total number of gates required for the reversible fault tolerant sequence counter is (1+2) *n+*(1+2)*n* = 6*n.*

As shown earlier quantum cost of a reversible fault tolerant half adder is 9. In Sec. 2 we show that the *F*2*G* can be realized with the 2 quantum cost. So, total quantum cost for the *n*-bit reversible fault tolerant binary adder is, (*n*_9+*n*_9) = 18*n*. Moreover, according to our previous discussions in Sec. 2.2 and Sec.3.1 the hardware complexity of the reversible fault tolerant half adder is 6_ + 4_ + 2. Therefore, the total hardware complexity of the *n*- to-2*n* RFSC is *n*_(6_+4_+2)+ *n*_(6_+4_+2)= (*12n_+8n_+4n*).

***Algorithm 2*:** Algorithm for proposed reversible fault tolerant sequential n-to-n binary adder.

***Input**:* Data input set $X(x_0,x_1....x_n)$, $Y(y_0,y_1....y_n)$ fault tolerant full adder (FTFA)

***Output**:* Reversible fault tolerant sequential n-to-n binary adder.

```
1    begin
2        i = input
3        o = output
4        for k→0 to n-1 do
5            $y_k$ → second.i.$FTFA_k$
6            $x_k$ → third.i.$FTFA_k$
7          if k=0 then
8              0 → first.i.$FTFA_k$
9          end if
10         else
11             second.o.$FTFA_k$ → first.i.$FTFA_k$
12         end if
13       end for
14       return  $FTFA_k$.o.1 → desired outputs
15               remaining outputs are garbage outputs
16   end
```

Algorithm 2 represents the design procedure of the proposed n-to-n reversible fault tolerant binary adder. Line 5 of the algorithm assigns the second input and line 6 of the algorithm assigns the third input of the FTFA gates. Line 8 and 11 assign the first input of the FTFA gates, when k=0 or else. End of the algorithm line 14 and 15 returned the desired outputs and garbage. To the best of our knowledge, it is one of the simplest algorithm of n-to-n reversible fault tolerant binary adder.

4.3 proposed Reversible Fault Tolerant Serial Multiplier Design

Sequential multiplication involves looking at the bits of the multiplier one by one and then adding partial products in an accumulative manner. The addition occurs in consecutive clock cycles, with a common register used for the accumulation of partial products which leads to the final results. The shift register and adder are as introduced in the previous sections.

In this design, we have three register and 7-bit adder. The first two registers are 7-bit width registers and include the multiplier and multiplicand numbers respectively. The third register is 8-bit width and its value includes the step by step summation of partial products and at the end of the calculation it consists of the final product result.

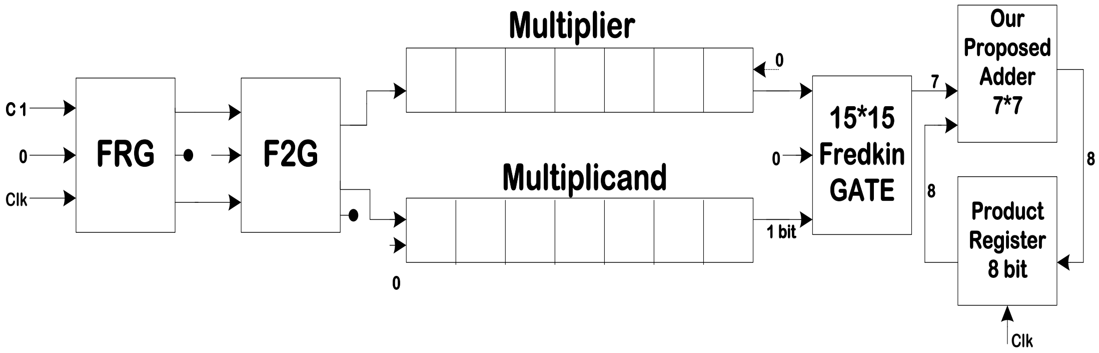


Fig 4.6: Proposed reversible fault tolerant multiplier.

At the beginning of the operation, the first two registers are filled with the numbers which are to be multiplied and the product register is filled with zero. Controlling the operation of the clock on the register is achieved by a one-bit control signal which is first zero and becomes one

after loading it at the pre-operation work of register and remains one

until the operation is completed. There is a 15 bit Fredkin gate (as Fig 1(b)), which with one control and 14 data line and with the same

Implementation and structure as a 3×3 Fredkin gate can be implemented by seven 3×3 Fredkin gates and its quantum cost is 35. If the control signal is one, multiplier is directed to the output; otherwise all-zero 7-bit number is directed to the output. The control signal is the current right-hand side of the multiplicand number. Each clock cycle, shifts the multiplier one bit to the left and enters zero in the right hand side bit. In the same clock, multiplicand is shifted one bit to the right and bit zero enters the left-hand side of the register. Each clock cycle also loads the product register with the currently calculated sum of the partial products. Therefore after three clock cycles the final result is prepared in the product register.

Algorithm 3: Algorithm for proposed reversible fault tolerant n-to-n multiplier.
Input: Data input set C(C$_0$,C$_{1.....}$ C$_n$), Feynman double gate**(F2G)** and register.
Output: Reversible n-to-n fault tolerant multiplier.

```
begin
       i = input
      o = output
       for j→0 to n-1 do
              c1 → first.i.FRG_j , 0→ second.i.FRG_j
              clk→ third.i.FRG_j
              first.oFRG_j → first.i.F2G_j , third.o.FRG_j → third.i.F2G_j
       for k ← 0 to n-1 do
              first.o.F2G_j →  first.i.multiplier
             second.o.F2G_j → first.i.multiplicand
             0 → second.i.multiplicand
             first.o.multiplier → first.i.FRG
            first.o.multiplicand → third.i.FRG
           0 →second.i.FRG
           fnd for
                 first.o.FRG → first.i.PA, first.o.PA→second.i.PR
                first.o.PA → first.i.PR, clk → second.i.PR
          end for
     return  PA.o.1 → desired outputs
               remaining outputs garbage outputs
end
```

Algorithm 3 represents the design procedure of the proposed n-to-n reversible fault tolerant multiplier. Line 5-6 of the algorithm assigns sequentially the first,second and third input of FRG.line 7 assign the first and third input of F2G.line 9 show the first input of multiplier and line 10-11 shows the first and second input of multiplicand.line 12-14 assign the input of FRG.line 16-17 assign the input and out put of PA and PR. End of the algorithm line 19-20 returned the desired outputs and sgarbage. To the best of our knowledge, it is one of the simplest algorithm of n-to-n reversible fault tolerant multiplier.

4.4 Summary

This chapter we proposed our reversible fault tolerant shift register. Then we also proposed a reversible fault tolerant binary adder and proposed reversible fault tolerant multiplier.

Chapter 5

Conclusions

5.1 Conclusion

In this paper reversible fault tolerant sequential circuit has been proposed which is much faster than any other existing reversible non-fault tolerant synthesis. The reversible design of this components can used in low number of outputs and constant inputs of the reversible fault tolerant sequential circuits. We also generalized algorithms for reversible consistency. In this paper we proposed reversible fault tolerant shift register, reversible fault binary adder and also proposed reversible fault tolerant multiplier.

5.2 Future work

Reversible engineering has been one of the thrust areas ensuring that continual process of the innovation trends that explore and sustain the resources of the nature. This reversible engineering is used in many fields like quantum computing, low power CMOS design, nanotechnology, optical information processing. This elements can be used for constructing reversible computers containing sequential parts as well as combinational parts.

References

[1]P. Kerntopf.:' A Comparison of logical efficiency Of reversible and conventional gates' ,Warsaw University of Technology, POLAND.

[2] Anuradha. B, Sivakumar.S.:' *A Fault Analysis in Reversible Sequential Circuits*' ,International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 2, February 2014

[3] Rawat.N, Bhardwaj.T.:' *Reversible Logic Gates and Their Applications* ' *,*International Journal of Innovative Research in Technology,volume 4,issue 6, ISSN : 2349-6002

[4] Ali.MB, Hossain.MM, Ullah.ME.:' *Design of Reversible Sequential Circuit Using Reversible Logic Synthesis*' *International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.4, December 2011.*

[5] I. Jamal, M. Shamsujjoha, and H. M. Hasan Babu.:"*Design of optimal reversible carry look-ahead adder with optimal garbage and quantam cost*," *International Journal of Engineering and Technology, vol.2, pp. 44-50, 2012.*

[6] Bennett. CH.:*'Logical reversibility of computation', IBM J. Research & Development, vol. 17, no.6, pp. 525–532, Nov. 1973.*

[7] N. Huda, S. Anwar, L. Jamal, H.M.H. Babu, Design of a reversible random accessmemory, Dhaka University Journal of Applied Science and Engineering 2 (1)(2011 July) 31–38.

[8] *R. Feynman, "Quantum mechanical computers,"* Foundations of Physics*, vol. 16, pp. 507–531, 1986, 10.1007/BF01886518. [Online]. Available: http://dx.doi.org/10.1007/BF01886518*

[9] M.M. Mano, Computer System Architechture, 3rd ed., Pearson Education, 2013.

[10] *F. Sharmin, M. M. A. Polash, M. Shamsujjoha, L. Jamal, and H. M. Hasan Babu, "Design of a compact reversible random access memory," in 4th IEEE International Conference on Computer Science and Information Technology, vol. 10, june 2011, pp. 103–107.*

[11] Christina.XS, Justine.MS, Rekha.K, Subha.U et al(2010).:

'*Realization of BCD adder using Reversible Logic*', *International Journal of Computer Theory and Engineering.2,3,1793-8201,333-337.*

[12] Lala.PK,Parkerson.JP AND Chakraborty.P(2010).:'*Adder Design using Reversible Logic Gates*', WSEAS Transactions on circuits and systems.9, Issue 6, 369-378.

[13] Sngupta.D,Sultan.M and Chaudhuri.A (2011).:'*Realization of a Novel Reversible SCG Gate and its Application for Designing parallel Adder/Subtractor and Match Logic*', *International journal of Computer Application(0975-8887).31,9,300-35.*

[14]Yang.G, Hung.WNN, Song.X and Perkowski.M (2005).:'*Majority-Based Reversible Logic Gates*', Theoretical Computer Science.334,Issues 1-3,259-274.

[15]Rentergem.YV and De Vos.A (2005).:'*Optimal Design of a Reversible Full Adder*', *International Journal of Unconventional Computing.1,339-355*

.

[16] Islam.S and Islam.R (2005),'*Minimization of Reversible Adder Circuits*', *Asian Journal of Information Technology.4,12,1146-1151*

.

[17]Shams.M, Haghparast.M and Navi.K(2008).:'*Novel Reversible Multiplier Circuit in Nanotechnology'*, World Appl.Sci.J.. 3(5),806-810.

[18]Banerjee.A and Pathak.A (2009),'*An Analysis of Reversible Multiplier circuits*',arXiv:0907.3357.1-10.

[19]Islam.MS et al.(2009),'*Low Cost Quantum Realization of Reversible Multiplier Circuit* ', *Information Technology Journal. 8,208*

.

[20] Haghparast.M, Mohammadi.M, Navi.K and Eshghi.M (2009),'

Optimized Reversible Multiplier Circuit', *Journal of Circuits, Systems and computers.18,2, 311-323.*

[21] Fredkin.E, Toffoli.T(1982).:*'Conservative Logic'* , International Journal of Theoretical physics.219-253.

[22]Picton.P (2006),'*Multi-Valued Sequential Logic Design Using Fredkin Gates*', *MVL Journal.1, 241.*

[23] Thapliyal.H, Srinivas.MB and Zwolinski.M(2005). '*A Beginning in the Reversible Logic Synthesis of Sequential Circuits*'. Proc. Military and Aerospace Programmable Logic Deices(MAPLD) Int. Conf. Washington, DC, September 7-9, Submission no. 1012. NASA office of Logic Design, Washington DC.

[24] Rice.JE (2006),'*A New Look at Reversible Memory Elements*', *Proc. Int. Symp.Circuits and Systems (ISC AS). Kos, Greece, May 21-24, IEEE, Piscataway, NJ, 1243-1256.*

[25] Krishnaveni.D and GeethaPriya.M (2011),'*A Novel Design of Reversible Serial and Parallel Adder/Subtractor*', *International Journal of Engineering Science and Technology (IJEST).3, 3, 2280-2288.*

[26]Maslov.D (2003),'*Reversible Logic Synthesis*', Doctoral dissertation, The University of New Brunswick.

[27]Maslov.D and Dueck.GW(2004).:'*Reversible Cascades with Minimal garbage*',IEEE Transactions on CAD of Integrated Circuits and Systems.

[28] Rice.JE(2006).:’*An Analysis of several proposals for reversible Latches*’, In the Proceedings of the Second International Joint Conferences on computer, Information and Systems Science and Engineering(CISSE), December, CDROM paper no.548(e-conference),published by Springer.

[29] Thapliyal.H and Ranganathan.N (2010).’ *Design of Reversible Sequential Circuits Optimizing Quantum Cost, Delay and Garbage output*’ ACM Journal on Emerging Technologies in Computer System.6,4,Article 14.

[30] Parminder Kaur & BalwindersinghDhaliwal “Design of Fault Tolerant Full Adder/Subtractor Using Reversible Gates” 2012 International Conference on Computer Communication and Informatics (ICCCI -2012), Jan. 10 – 12, 2012, Coimbatore, INDIA 978-1-4577-1583-9/ 12/ $26.00 © 2012 IEEE.

[31] Haghparast, M. and K. Navi (2008). A novel reversible bcd adder for nanotechnology based systems. American Journal of Applied Sciences 5 (3), 282–288.