An Efficient Synthesis of Reversible Fault Tolerant Circuit for Sorting Algorithms Optimization

Submitted by:

Moumita Saha
ID: 2012-2-60-005

Sarah Ahmed
ID: 2012-2-60-016

Christopher D' Costa
ID: 2012-2-60-019

Supervised by:

Md. Shamsujjoha
Senior Lecturer
Department of Computer Science and Engineering
East West University

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

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

April 2017

Abstract

Conventional circuits dissipates power by losing bits of information whereas reversible circuit recovers bit loss through unique input-output mapping for all possible combination and logically dissipate zero heat. In addition, fault tolerant reversible circuit consists of only reversible parity preserving gates can detect faulty signal from its primary outputs. Moreover, Quantum computers, more specifically quantum circuits requires their computational process maintain the reversibility during entire computation. On the other hand, use of sorting algorithms to manage large volume of data is increasing. In these consequences, this research studies the optimization of Selection sort, Bubble Sort, Merge Sort and Insertion Sort in reversible fault tolerant circuits composed of only reversible fault tolerant Fredkin gates. The proposed algorithm performs sorting among optimal implementations of designed essential functions to implement the reversible fault tolerant boolean function. The proposed optimization algorithm also considers both gate and quantum level circuit costs. First, reversible cost is reduced by considering adjacent gate pairs. Then, inner quantum structures of the gates are minimized. The proposed scheme is evaluated with respect to existing approaches which showed that the proposed method performs much and are much more scalable.

Declaration

We hereby declare that, this project was done under CSE497 and has not been submitted elsewhere for requirement of any degree or diploma or for any purpose except for publication.

_____

Moumita Saha
ID: 2012-2-60-005
Department of Computer Science and Engineering
East West University

_____

Sarah Ahmed

ID: 2012-2-60-016
Department of Computer Science and Engineering
East West University

_____

Christopher D' Costa
ID: 2012-2-60-019
Department of Computer Science and Engineering
East West University

Letter for Acceptance

We hereby declare that thesis is from the student's own work and best effort of ourselves and all other sources of information used have been acknowledged. This thesis has been submitted with our approval.

Supervisor

_____

Md. Shamsujjoha

Senior Lecturer

Department of Computer Science and Engineering,

East West University, Dhaka, Bangladesh

Chairperson (acting)

_____

Dr. Ahmed Wasif Reza

Associate Professor,

Department of Computer Science and Engineering,

East West University, Dhaka, Bangladesh

Acknowledgements

We are thankful to Almighty who offers us divine blessings, patience, mental and psychical strength and we are able to complete this project only because of His blessings. 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. We are deeply indebted to our project supervisor Mr Md. Shamsujjoha for his scholarly guidance, especially for his tolerance with our persistent bothers and unfailing support. He gave us the freedom to pursue aspects of reversible fault tolerant computing, which we found interesting and compelling. This helped our project to achieve its desired goals.

We 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, we would like to thank our family members and friends. Their continued tolerance with our growing demands and tendency to disappear for weeks at a time gave us a much-needed break from the world computing.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Moore's Law had kept it's validity for decades where it has been stated that "the number of transistors on a chip doubled in a regular basis of every 18 to 24 months and along with that the chip's power dissipation increases simultaneously". This gradual increase of power dissipation has now reached to a crucial stage that it has created a deem necessity of low-power designs with new technologies. Reversible circuit could be the solution for this crisis. Reversible logic plays an extensively important role in low power computing[16] as it recovers from bit loss through unique mapping between input and output vectors[9].

However this can only solve bit loss problem not bit error. Hence along with reversible logic, fault tolerant logic has been used as it can check parity of essential function and can solve the bit error problem.

Parity checking is a popular mechanism for detecting faulty signal. Reversible fault tolerant circuit based on reversible fault tolerant gates allows to detect faulty signal in the primary outputs of the circuit through parity checking [14].

Sorting algorithms is needed to manage large volume of data which is increasing rapidly. Optimization of sorting cost is much important for this reason. This research studies the optimization of Selection sort, Bubble Sort, Merge Sort and Insertion Sort in reversible fault tolerant circuits composed of only reversible fault tolerant Fredkin gates. The proposed algorithm performs sorting among optimal implementations of designed essential functions to implement the reversible fault tolerant boolean function.

1.1 Motivation

In logic computation, every bit of information loss generates $kTln2$ joules of heat, where k is the Boltzmann constant and T is the absolute temperature of the environment [1]. Bennett showed that zero energy dissipation is possible if the circuit consists of only reversible gates [2]. Reversible circuit consists of only reversible gates recovers from bit loss through unique mapping between inputs and outputs. No bit loss property results less power dissipation [3]. Moreover, it is viewed as a special case of quantum circuit as quantum evolution must be reversible [4], [22]. Over the last twenty years, reversible computation gained remarkable interests in the development of highly efficient algorithms [3], [5], optimal architecture [6],

[5], simulation and testing [7], [8], [21] DNA [23] and nano-computing [9], [10], [24] quantum dot cellular automata [11], [12], [17], discrete event simulation [25] and in the development of highly efficient algorithms [26] etc. In addition, fault tolerant circuit based on reversible fault tolerant gates detect faulty signal in low level through parity checking [3], [6], [7], [13]. Parity checking is probably the most easiest mechanisms for detecting single level fault. An entire circuit can preserve parity if its individual gate is parity preserving. Generally it is used to detect errors in the storage or transmission of information. If the parity of the input data is maintained throughout the computation, then the intermediate checking would not be required and the entire circuit can preserve parity if its individual gate is parity preserving [27].

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 recover bit loss or power through unique mapping.
- Detect faulty signal through parity preserving gates.
- Optimize the sorting cost.

1.3 Overview

This document presents the implementation of different types of sorting algorithms including merge sort, selection sort, bubble sort, insertion sort with respect to reversible logic. We have proposed a synthesis process of reversible and fault tolerant circuit with improvement in terms of cost comparing with the existing designs. The different components are reversible fault tolerant computing, sorting, optimization, synthesis.

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 equivalent realizations etc.
- We took a decision whether to use garbage outputs or not for this synthesis algorithm. Here, instead of garbage outputs minimum number of bit lines has been used.

- Designing various reversible and fault tolerant combinational circuits for existing and proposed essential function in order to calculate gate cost.

- Designing 4 sorting algorithm to sort existing and proposed essential function and thus we get sorting cost for both methods.

- Multiplying gate cost and sorting cost, finally we get the circuit cost to prove the efficiency of our proposed method.

1.5 Outline

The next chapter (Chap.2) discusses the Background study i.e basic definition and literature overview relating to reversible and fault tolerant computing. The study includes understanding of the reversible and fault tolerant logic gates along with their quantum equivalent realizations and applications. It also includes definition of used sorting algorithms which are selection sort, bubble sort, merge sort, insertion sort.

Chap.3 briefly discusses about the existing synthesis process with their performance and evaluation. Chap. 4 introduces several components of proposed sorting algorithms and also lock diagrams of fredkin and toffoli gate applying proposed essential function. Chap.5 illustrates the performance evaluation of the proposed method. Chap. 6 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 and fault tolerant logic and different sorting methods. In this chapter, reversibility is apprised and how reversibility solve the loss of bit pattern and bit error problem 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.

2.1Why Reversible:

Losing information in a circuit causes losing power. Information lost when the input vector cannot be uniquely recovered from the output vector of a combinational circuit. Interest in reversible computation arises from the desire to reduce this dissipation. Reversible computing is the path to future computing technologies, which all happen to use reversible logic. In addition, reversible computing will become mandatory because of the necessity to decrease power consumption.

2.2 Reversible Gate

An $n \times n$ *reversible gate* is a data stripe block that uniquely maps between input vector $Iv = (I_0, I_1, .., .In-1)$ and output vector $Ov = (O_0, O_1, \ldots, On-1)$ denoted as $Iv \leftrightarrow Ov$. Two prime requirements for the reversible logic circuit are as follows [14]:

- 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.

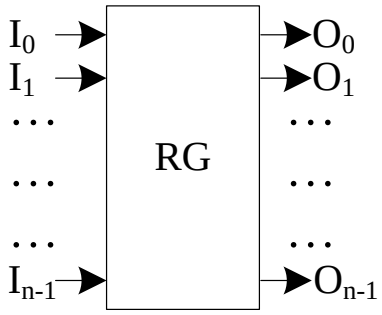


Fig. 2.1: A n × n Reversible Gate

2.3 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.

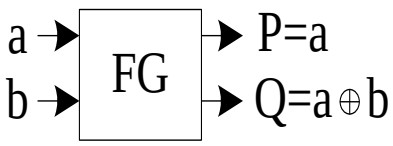


Fig. 2.2: A Reversible Gate with One Garbage Output

2.4 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 [15]. 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 Feynman gate in Fig. 2 is 1 and the quantum cost of Feynman Double gate in Fig. 3 is 2.

2.5 Fault Tolerant Gate

A Fault tolerant gate is a reversible gate that constantly preserves same parity between input and output vectors. Such parity-preserving reversible gates, when used with an arbitrary synthesis strategy for reversible logic circuits, allow any fault that affects no more than a single logic signal to be detectable at the circuit's primary outputs. More specifically, an n × n fault tolerant gate clarify the following
Property between the input and output vectors [12]:

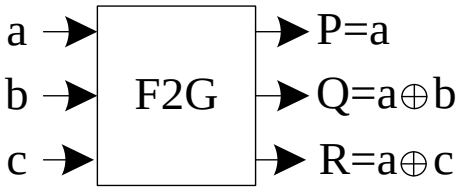
$I_0 \oplus I_1 \oplus I_2 \oplus \ldots. I_{n-1} = O_0 \oplus O_1 \oplus O_2 \ldots.. O_{n-1}$ (2.1)

where $I_0, I_1, .., I_{n-1}$ are input vectors and $O_0, O_1, \ldots, O_{n-1}$ are output vectors.
Parity preserving property of Eq.1 allows to detect a faulty signal from the circuit's primary output. Researchers [11],[12], [15], [28] have showed that the circuit consist of only reversible fault tolerant gates preserves parity and thus able to detect the faulty signal at its primary output.

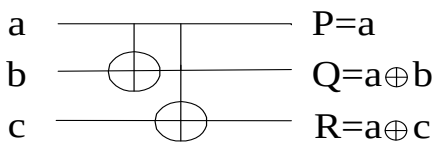
2.6 Popular Reversible Fault Tolerant gates used for proposed method

2.6.1 Feynman Double Gate

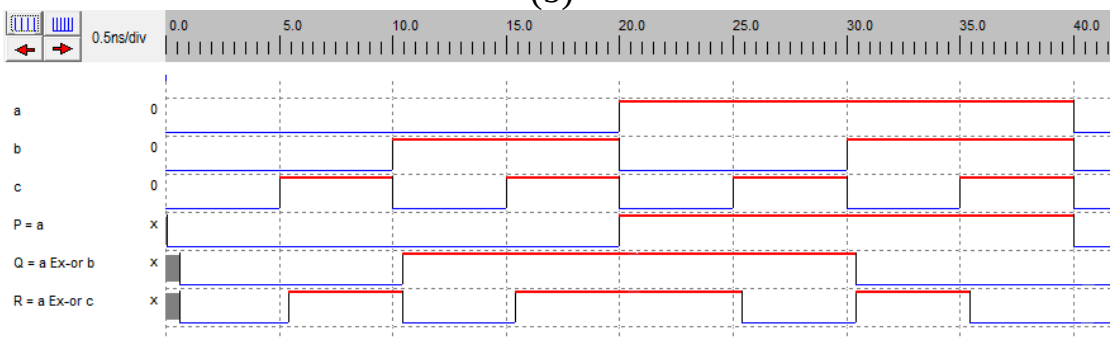
Input vector (*Iv*) and output vector (*Ov*) for 3 × 3 reversible Feynman double gate (*F*2*G*) is defined as follows [16]: $Iv = (a, b, c)$ and $Ov = (a, a \oplus b, a \oplus c)$. Block diagram of *F*2*G* is shown in Fig. 3(a). Fig. 3(b) represent the quantum equivalent realization of *F*2*G*. From 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* [29].



(a)



(b)



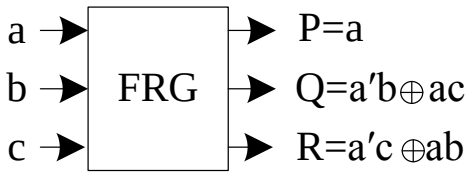
(c)

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

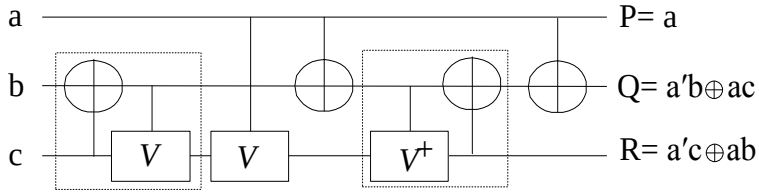
2.6.2 Fredkin Gate

The basic Fredkin gate [1] is a controlled swap gate that maps three inputs (C, I1, I2) onto three outputs (C, O1, O2). The C input is mapped directly to the C output. If C = 0, no swap is performed; I1 maps to O1, and I2 maps to O2. Otherwise, the two outputs are swapped so that I1 maps to O2, and I2 maps to O1. It is easy to see that this circuit is reversible, i.e., "undoes" itself when run backwards. A generalized n×n Fredkin gate passes its first n-2 inputs unchanged to the corresponding outputs, and swaps its last two outputs if and only if the first n-2 inputs are all 1. The input and output vectors for 3 × 3 Fredkin gate (F RG) are defined as follows [20]:

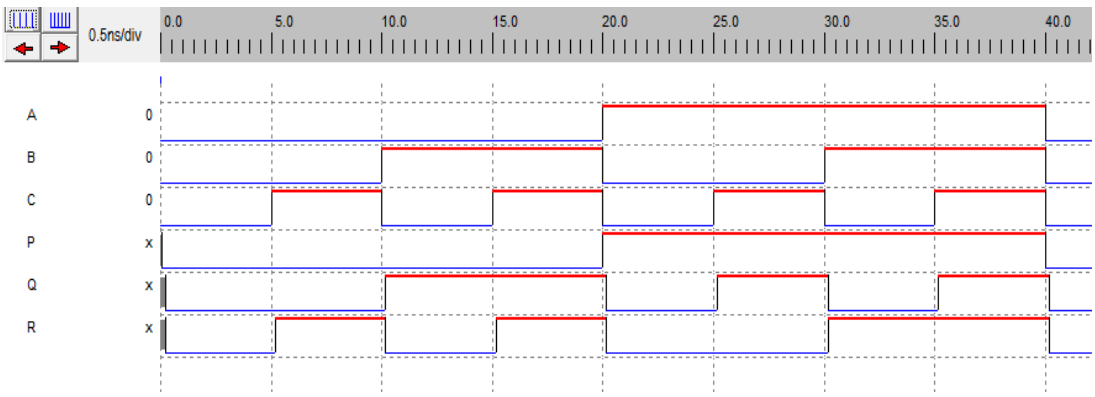
Iv = (a, b, c) and *Ov = (a, ab ⊕ ac, ac ⊕ ab).*Block diagram of *FRG* is shown in Fig. 4(a). Fig. 4(b) represents the quantum realization of *FRG*. In Fig. 4(b), each rectangle is equivalent to a 2 × 2 quantum primitives, therefore its quantum cost is considered as one [17]. Thus total quantum cost 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 *F2G* and *FRG*

Input			Output of F2G			Output of FRG			
A	B	C	P	Q	R	P	Q	R	Parity
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

2.7 Circuit cost:

In this study, circuit costs have been determined in two steps. At first reversible cost has been realized then by multiplying this with sorting cost the total circuit cost has been calculated.

2.8 Reversible cost:

Reversible cost is the total number of reversible gates used in a given circuit considering each gate as one.

2.9 Sorting Cost:

Sorting cost can be simply determined by counting the total move done for reordering the given function. Four different sorting algorithms have been used to show the efficiency of proposed algorithm.

2.10 Selection sort:

A sorting technique that is typically used for sequencing small lists. It starts by comparing the entire list for the lowest item and moves it to the #1 position. It then compares the rest of the list for the next-lowest item and places it in the #2 position and so on until all items are in the required order.

2.11 Insertion sort:

A simple sorting technique that scans the sorted list, starting at the beginning, for the correct insertion point for each of the items from the unsorted list. Insertion sort keeps making the left side of the array sorted until the whole array is sorted.

2.12 Merge sort:

A sorting technique that sequences data by continuously merging items in the list. Divide and Conquer is a method of algorithm design that has created such efficient algorithms as Merge Sort. It has three steps:

=If the input size is too large to deal with in a straightforward manner, divide the data into two or more disjoint subsets.

=Use divide and conquer to solve the sub problems associated with the data subsets.

=Take the solutions to the sub problems and “merge” these solutions into a solution for the original problem.

2.13 Bubble sort:

A sorting technique that is typically used for sequencing small lists. It starts by comparing the first item to the second, the second to the third and so on until it finds one item out of order. It then swaps the two items and starts over. The sort may alternate from the top of the list to the bottom and then from the bottom to the top.

2.14 Summary

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

Chapter 3

Existing Method

By considering recent experiment, where toffoli gate had been used, it can be easily understood that their minimal cost of sorting is two. choose a selection sort algorithm that uses row by row checking of mismatched input and output bits. Row by row checking and fixing can be efficiently done by reversible circuits, in our case by circuit implementations of essential functions. However, operations like dividing and sliding used in other sorting algorithms necessitates relatively larger circuits For example, a merge sort algorithm divides a target set into two subsets and sorts them separately. After subset sorting is completed, they are merged and the same procedure repeats until finding a final solution. Here, each of these operations, especially the dividing operation, is quite costly. Similarly, sliding process used in an insertion sort and bubble sort algorithms have sthe same handicap. Figure 7 illustrates differences between sorting algorithms by showing the number of required essential function implementations. For this specific example, the selection/merge/insertion/bubble algorithms require 2/4/6/6 reversible gate implementation, respectively. Moreover their sorting cost is also high. Therefore, overall circuit cost is high and not even fault tolerant. Motivated by these, we aim at reducing circuit cost and bit error problem.

3.1 Block diagram of Reversible Gate (Toffoli)

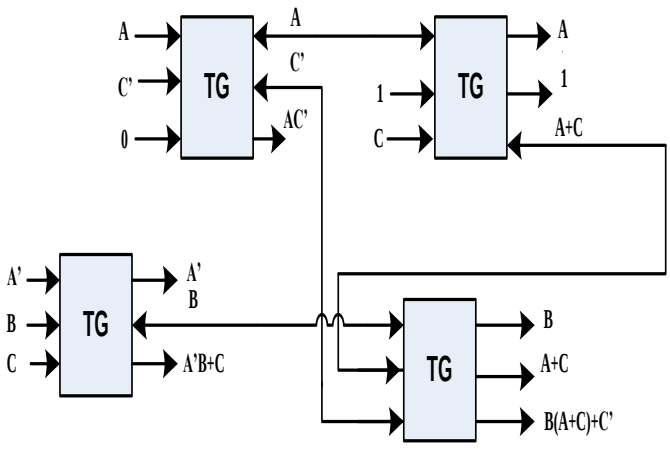


Fig 3.1: Block diagram of reversible 3×3 Toffoli Network for sorting

Using reversible logic in essential function from the recent research reversible (toffoli) gates have been constructed. There are 4 toffoli gates used in this diagram. From gates requisite outputs are received. There we get some garbage as well as used some constant inputs in those reversible (toffoli) gates.

3.2 Summary

This chapters demonstrate the working procedure of basic reversible Control Unit with the help of their structures. The design approaches towards the reversible Control Unit have also been presented here.

Chapter 4

Proposed Reversible Fault Tolerant And Sorting Method

4.1 Proposed Synthesis Process

In this section, we present the implementation of different type of sorting methods including bubble sort, merge sort, insertion sort, selection sort with respect to reversible fault tolerant logic. This is a two steps synthesis process. At first, a permutation based algorithm has been used to implement certain functions called "essential functions". As a second step, given target functions has been implemented by using the essential function implementations obtained in the first step. Moreover, uses of minimum number of reversible gates make this algorithm optimal. The overall synthesis approach is greedy as opposed to dynamic or optimal. It was a subject of concern whether to use garbage outputs or not for synthesis algorithm. Garbage outputs are used especially for synthesizing functions with high number of bits [18] that offers design flexibility and gate cost reduction at the expense of additional bit line [19]. Here, instead of garbage outputs minimum number of bit lines has been used. This synthesis algorithm is straightforward, fast and do not require additional bit lines. To implement our essential function fredkin gate has been used. The four fast sorting algorithms have proven the efficiency of the chosen reversible and fault tolerant fredkin gate. Sorting cost of it's essential function with four different sorting algorithm has been stored to determine overall circuit cost. After storing the essential function implementation sorting process has been developed. Following table (Table-1) shows the sorting process where f is the essential function which is used to find a target function F. Target function contains 8 rows in which 2 of them are mismatched row with different input and output values. Interchange the output rows is done pairwise to match them. Red line is essential function which indicate the interchange row. Implementation of ordering function can be arbitary. That means, if the order of f1 funtion is changed then F function can also be implemented.

Table 4.1: Essential function of proposed method

Input				Output			
a	b	c		A	B	C	Parity
0	0	0		0	0	0	E
0	0	1		0	0	1	O
0	1	0		0	1	0	O
0	1	1		0	1	1	E
1	0	0		1	0	0	O
1	0	1		1	1	0	E
1	1	0		1	0	1	E
1	1	1		1	1	1	O

In the first step, essential function has been implemented. Then from the essential function for each variables (a,b,c) using K-map equations have been formed. Using those equations block diagram for toffoli and fredkin have been developed. From the block diagrams and after soring the mismatched rows gate costs as well as sorting costs have been calculated. Finally, multiplying gate cost and sorting cost circuit cost is being calculated.

4.2 Block diagram of Reversible Gate (Toffoli) And Reversible Fault Tolerant (Fredkin) Network for Soritng

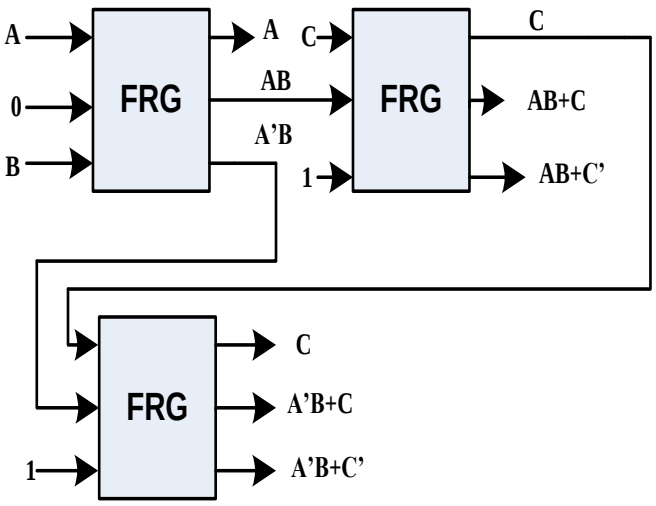


Fig 4.1: Fredkin gate

Using reversible logic in essential function from the proposed reversible and fault tolerant (fredkin) gates have been constructed. There are 3 fredkin gates used in this diagram. From gates requisite outputs are received. There we get some garbage output as well as used some constant input in those reversible (fredkin) gates.

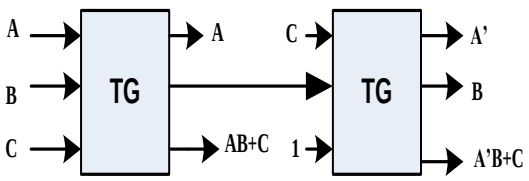


Fig 4.2: Toffoli gate

Using reversible logic in essential function from the proposed method, reversible (toffoli) gates have been constructed. There are 2 toffoli gates used in this diagram. From gates requisite outputs are received. There are some garbage as well as constant outputs used in those reversible (toffoli) gates.

4.3 Proposed Algorithm

a. Proposed Algorithm:

Algorithm1: Algorithm for proposed synthesis process:

Input: Mismatch row

Output: Sorted row

1. Begin
2. Select an essential function
3. do permutation circuit
4. Check
5. while(not match)
6. Store and go to step 1
7. End

4.4 Design of Sorting Algorithms for Proposed Essential Function

Four types of sorting algorithm has been used to assure the efficiency of proposed algorithm.

4.4.1 Selection Sort

Algorithm: Algorithm of selection sort for proposed reversible fault tolerant Fredkin Gate(FRG)

Input: An array A[1 ... n] of n elements.

Output: A[1... n] sorted in non-decreasing order.

1. Begin

2. for $i \leftarrow 1$ to $n - 1$

3. $k \leftarrow i$

4. for $j \leftarrow i + 1$ to n {Find the *i th* smallest element.}

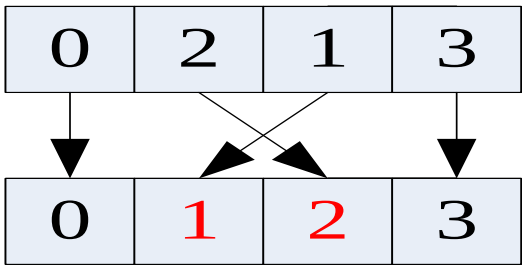
5. if $A[j] < A[k]$ then $k \leftarrow j$

6. end for

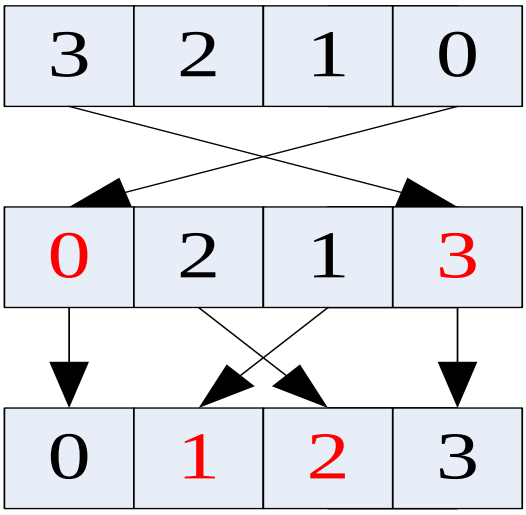
7. if $k \neq i$ then

8. SWAP $A[i]$ and $A[k]$

9. end for

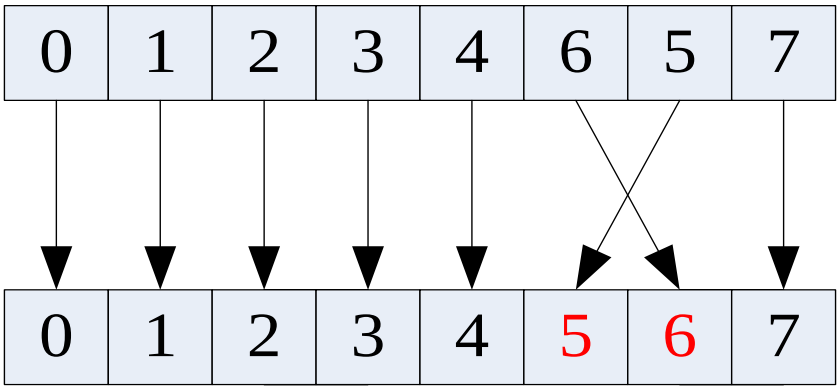


(a)

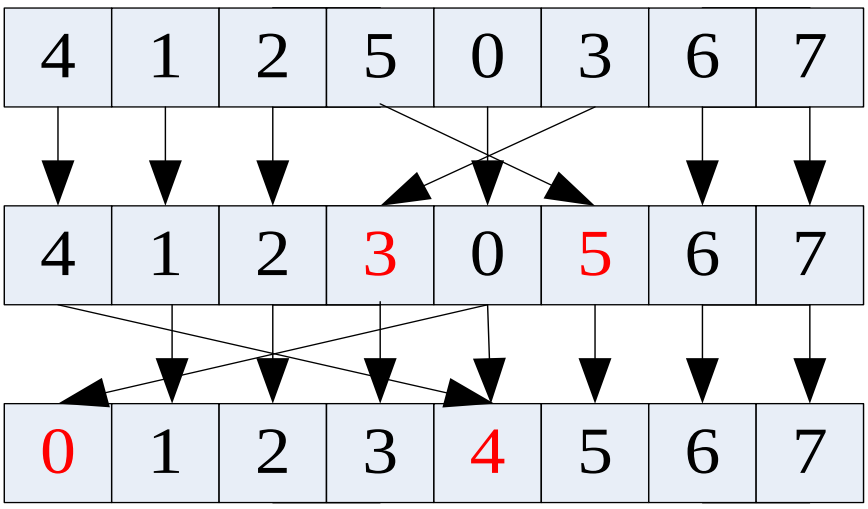


(b)

Fig: 4.3 Selection sorting of (a) Fredkin and (b) Toffoli gate for 4 bit



(a)



(b)

Fig: 4.4 Selection sorting of (a) Fredkin and (b) Toffoli gate 8 bit

Table 4.2: Selection sorting cost of proposed and existing 4 and 8 bit

Fredkin Gate		Toffoli Gate	
4 bit	8 bit	4 bit	8 bit
1	1	2	2

Here, from these figure 4.3 and figure 4.4 we get the selection sorting cost for fredkin which is 1 and for toffoli which is 2 for 4 bit and for 8 bit selection sorting cost for fredkin is 1 and for toffoli is 2. Table 4.2 shows that selection sorting cost of fredkin is lower than toffoli.

4.4.2 Merge Sort

Algorithm: Algorithm of merge sort for proposed reversible fault tolerant Fredkin Gate(FRG)

Input: An array A[1 ... n] of n elements.

Output: A [1... n] sorted in non-decreasing order.

1.Begin

2. if length*(m)* <= *1*

3. return *m*

4. var list *left, right, result*

5. var integer *middle* = length *(m) / 2*

6. for each *x* in *m* up to *middle*

7. add *x to left*

8. for each *x* in *m* after or equal *middle*

9. add *x to right*

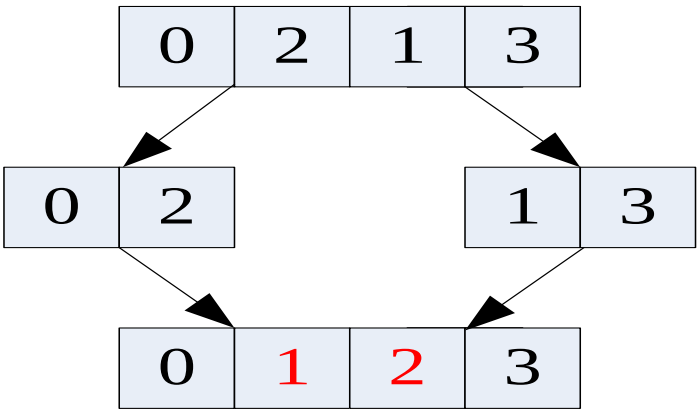
10. *left* = merge_sort *(left)*

11. *right*= merge_sort *(right)*

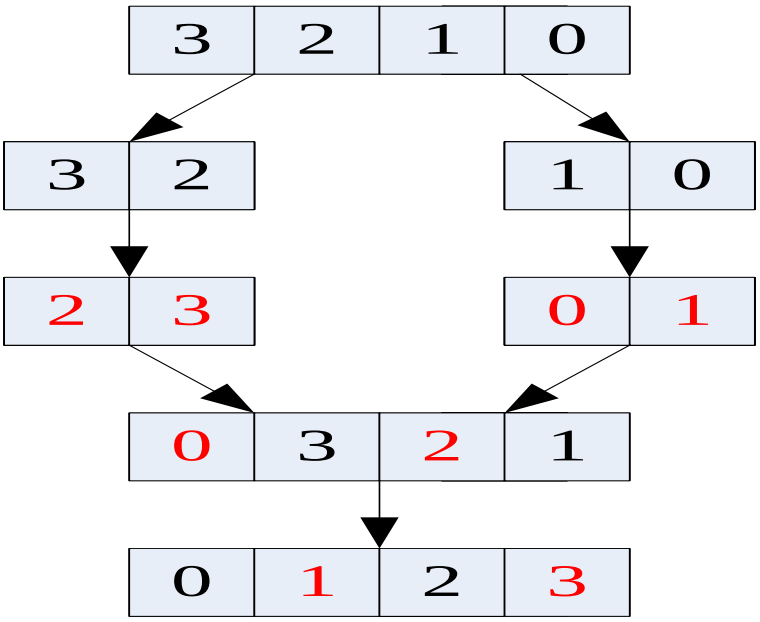
12. *result*= merge *(left, right)*

13. return *result*

14. End

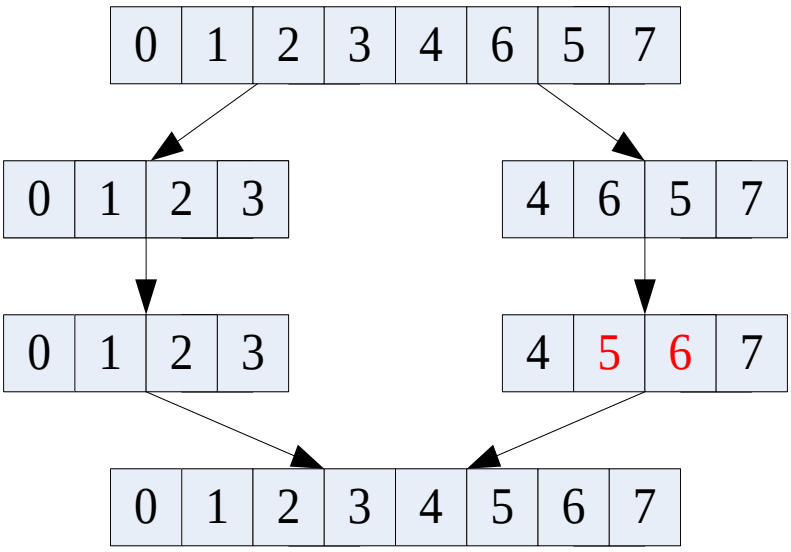


(a)

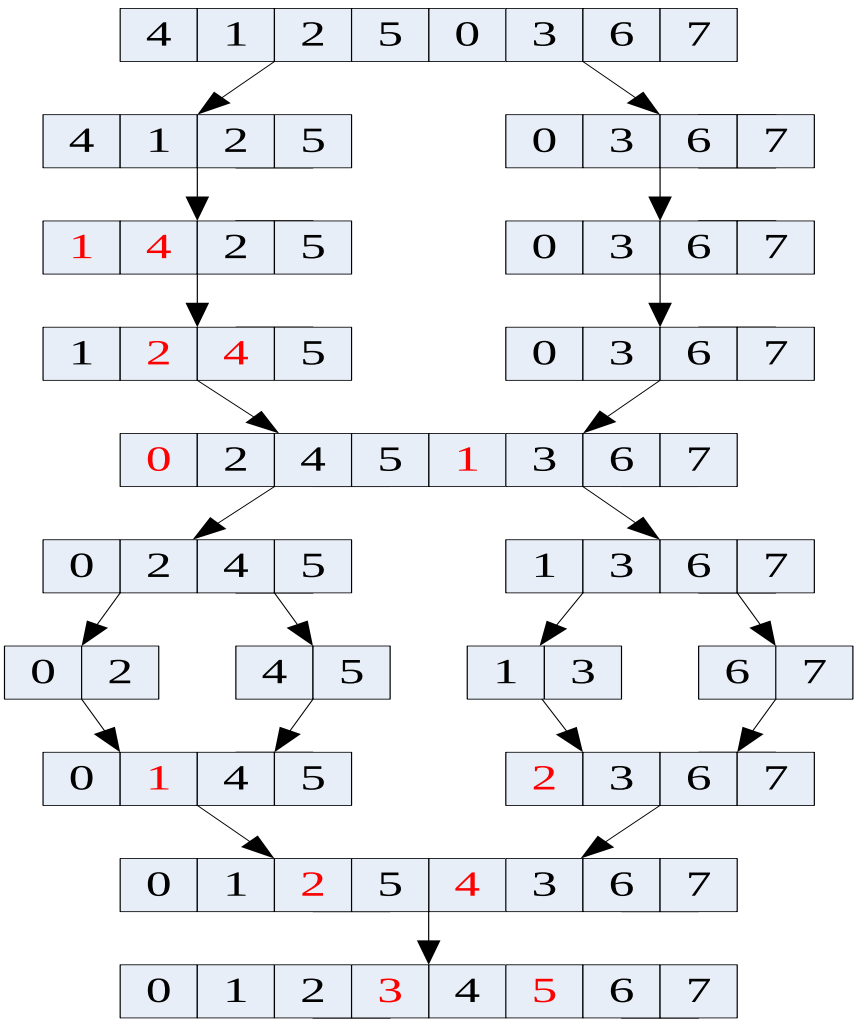


(b)

Fig 4.5: Merge sorting of (a) Fredkin and (b) Toffoli gate for 4 bit



(a)



(b)

Fig 4.6: Merge sorting of (a)Fredkin and (b) Toffoli gate for 8 bit

Table 4.3: Merge sorting cost of proposed and existing 4 and 8 bit

Fredkin Gate		Toffoli Gate	
4 bit	8 bit	4 bit	8 bit
1	1	4	7

Here, from these figure 4.5 and figure 4.6 we get the merge sorting cost for fredkin which is 1 and for toffoli which is 4 for 4 bit and for 8 bit merge sorting cost for fredkin is 1 and for toffoli is 7. Table 4.3 shows that merge sorting cost of fredkin is lower than toffoli.

4.4.3 Insertion Sort

Algorithm: Algorithm of insertion sort for proposed reversible fault tolerant Fredkin Gate (FRG)

Input: An array A [1 ... n] of n elements.

Output: A [1... n] sorted in non-decreasing order.

1. Begin

2. for *j = 1 to n-1*

3. *key = A[j]*

4. *i = j – 1*

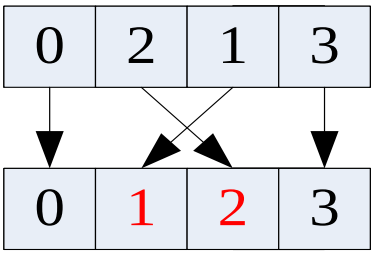
5. while *i >= 0* & A *[i] > key*

6. *A [i +1] = A[i]*

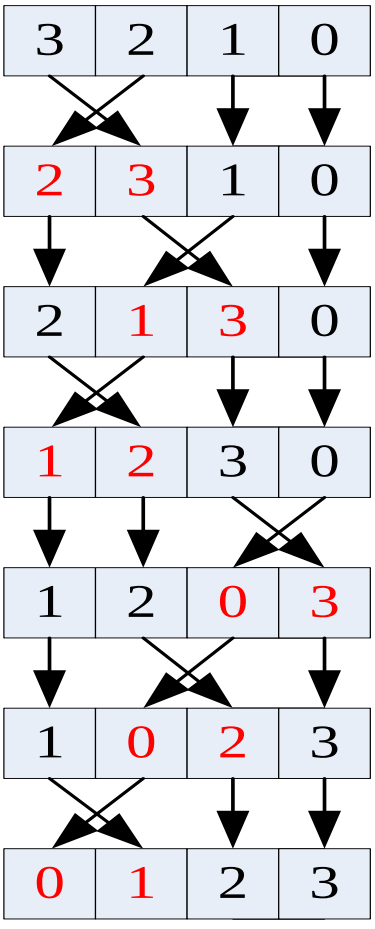
7. *i = i -1*

8. *A [i +1]* = key

9. End

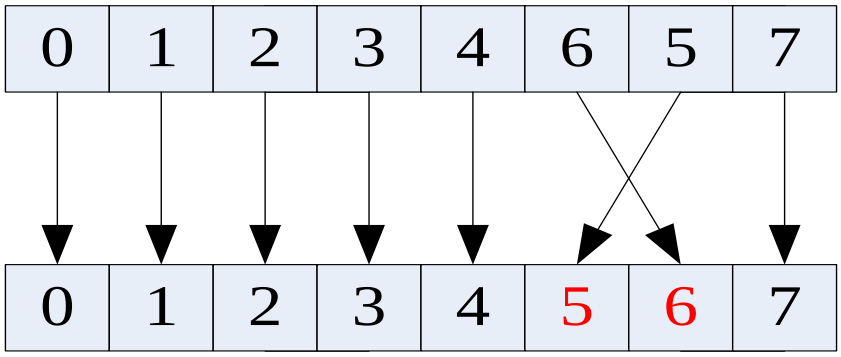


(a)

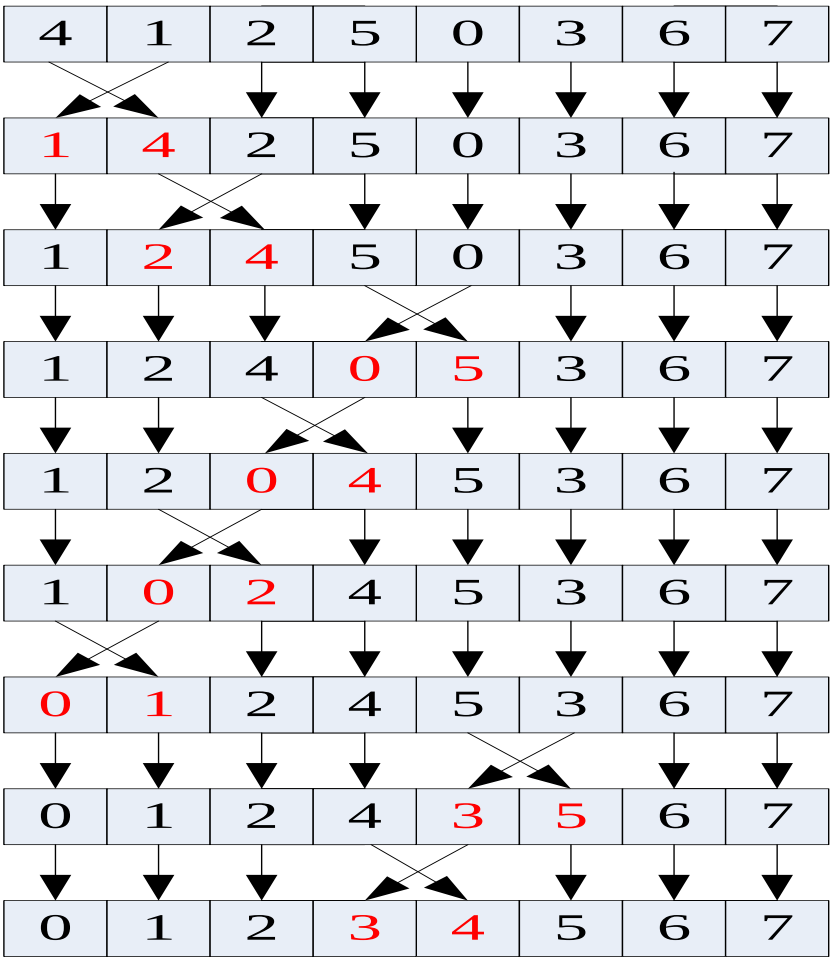


(b)

Fig 4.7 : Insertion sorting of (a) Fredkin and (b) Toffoli gate for 4 bit



(a)



(b)

Fig 4.8 : Insertion sorting of (a) Fredkin and (b) Toffoli gate for 8 bit

.

Table 4.4: Insertion sorting cost of proposed and existing 4 and 8 bit

Fredkin Gate		Toffoli Gate	
4 bit	8 bit	4 bit	8 bit
1	1	6	8

Here, from these figure 4.7 and figure 4.8 we get the insertion sorting cost for fredkin which is 1 and for toffoli which is 6 for 4 bit and for 8 bit insertion sorting cost for fredkin is 1 and for toffoli is 8. Table 4.4 shows that insertion sorting cost of fredkin is lower than toffoli.

4.4.3 Bubble Sort

Algorithm: Algorithm of bubble sort for proposed reversible fault tolerant Fredkin Gate(FRG)

Input: An array A[1 ... n] of n elements.

Output: A[1... n] sorted in non-decreasing order.

1. Begin

2. Set *flag = false*

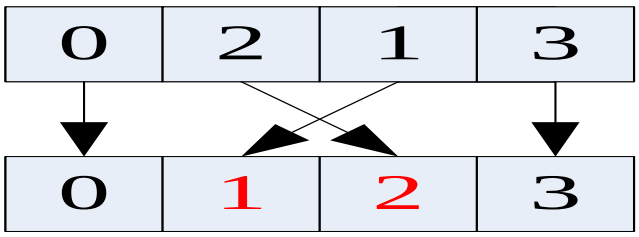
3. Traverse the array and compare pairs of two consecutive elements

2.1 If *E1 =< E2* then OK (do nothing)

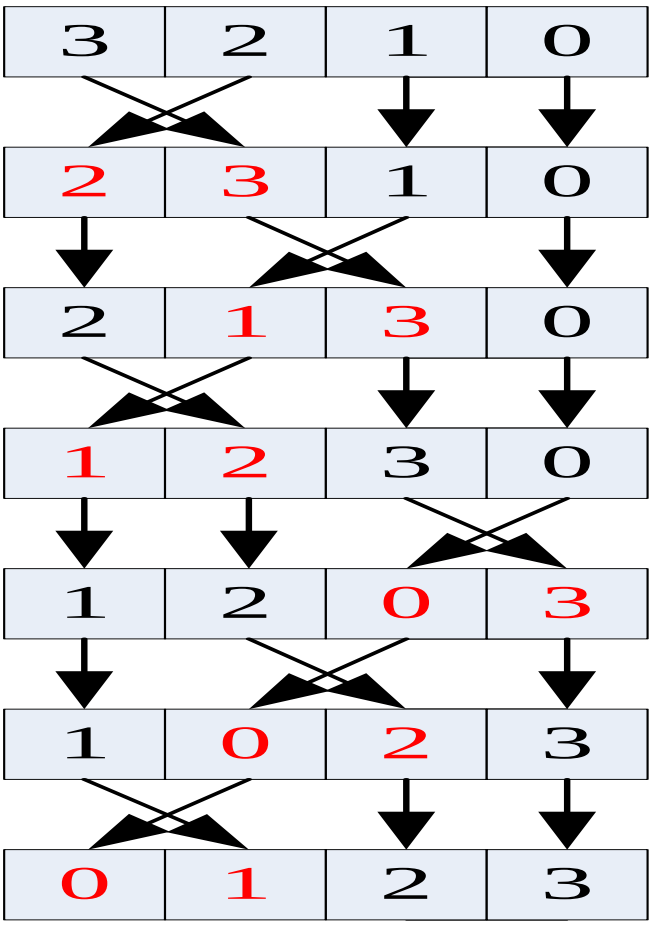
2.2 If *E1 > E2* then Swap*(E1, E2)* and set *flag = true*

4. repeat 1. and 2. while *flag=true*.

5. End

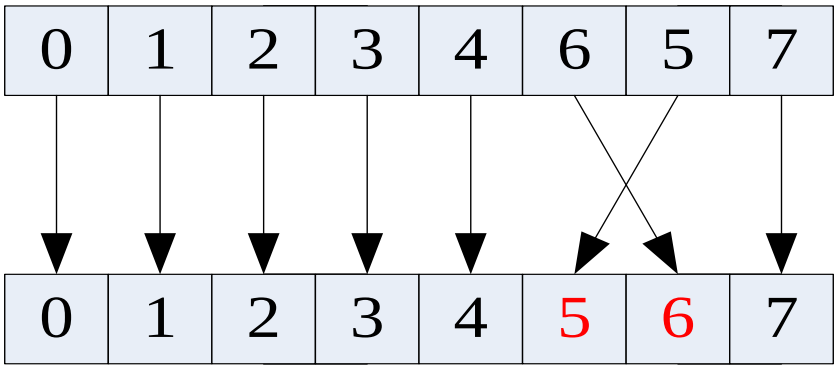


(a)

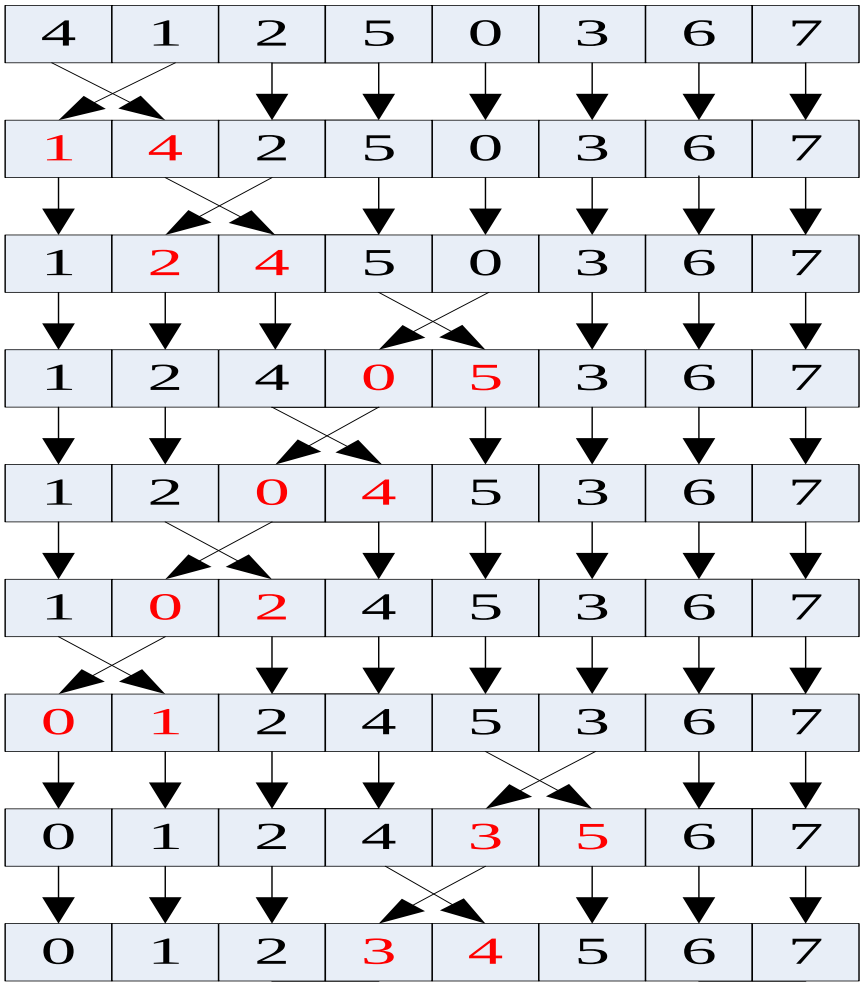


(b)

Fig 4.9 : Bubble sorting of (a) Fredkin and (b) Toffoli gate for 4 bit



(a)



(b)

Fig 4.10 : Bubble sorting of (a) Fredkin and (b) Toffoli gate for

Table 4.5: Bubble sorting cost of proposed and existing 4 and 8 bit

Fredkin Gate		Toffoli Gate	
4 bit	8 bit	4 bit	8 bit
1	1	6	8

Here, from these figure 4.9 and figure 4.10 we get the bubble sorting cost for fredkin which is 1 and for toffoli which is 6 for 4 bit and for 8 bit bubble sorting cost for fredkin is 1 and for toffoli is 8. Table 4.5 shows that bubble sorting cost of fredkin is lower than toffoli.

From these above sorting diagrams we observe that, sorting cost of fredkin is always 1 which we prove with 4 bit and 8 bit essential function. Applying this same method for 16 bit it is obtained that, this value still remain 1. From these research, it is clear that for n bit essential function the value will be the same. On the other hand, sorting cost for toffoli is always varying and which never less than proposed method. This prove the excellence of our research work.

4.5 Sorting Cost:

Proposed and existing in both cases, applying sorting algorithm it is assured that sorting cost of fredkin is always 1.Which is less than not only from existing (toffoli) but also all reversible circuit. This can be considered as the minimum cost. Despite, proposed paper deals with much gate the circuit cost is remain minimum.

4.6 Summary

This chapter detailed the design and working procedures of the proposed reversible fault tolerant Control Unit. Here, several lower bounds on the number of garbage outputs, constant inputs and quantum cost of fault tolerant logic circuits are proposed for Control Unit. Finally construction procedure and algorithm are proposed for the implementation of proposed control Unit. It has also been evidenced that the proposed components are optimized greatly from the existing components.

Chapter 5

Performance Evaluation of the Proposed Method

In the previous chapter, the performance of the components of reversible fault tolerant circuit and their sorting process and the existing components of reversible circuit and sorting process been shown. This chapter provides the overall performances of the proposed reversible fault tolerant scheme with the existing reversible methods. The performance of the proposed method is evaluated by the required number of gates, garbage outputs, constant inputs, quantum cost and sorting cost.

5.1 Performance of proposed algorithms

According to proposed, circuit cost of fredkin is less than existing (toffoli). from the following tables it can be perfectly verified.

Table 5.1: Essential function of existing method

Input				Output		
c	b	a		C	B	A
0	0	0		1	0	0
0	0	1		0	0	1
0	1	0		0	1	0
0	1	1		1	0	1
1	0	0		0	0	0
1	0	1		0	1	1
1	1	0		1	1	0
1	1	1		1	1	1

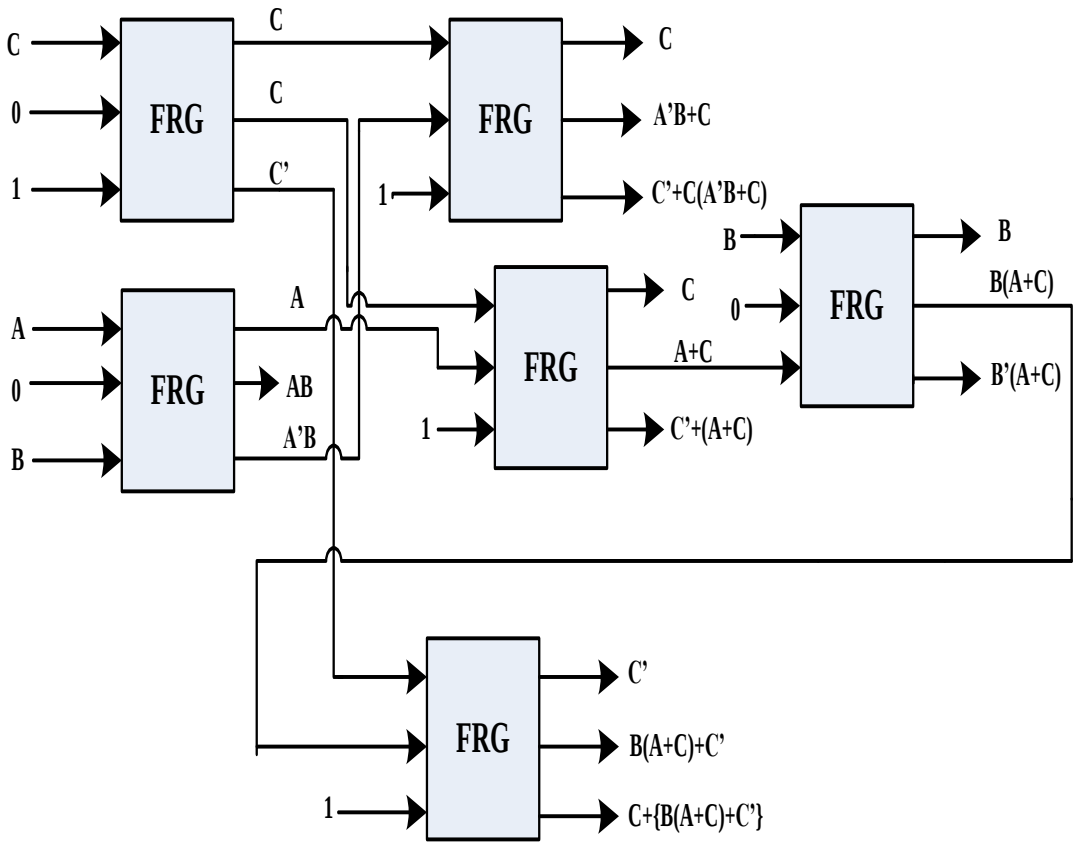


Fig 5.1 : Existing essential function realization with fredkin gate

Using reversible logic in essential function from the from the recent research reversible and fault tolerant (fredkin) gates have been constructed. There are 6 fredkin gates used in this diagram. From gates requisite outputs are received. There we get some garbage output as well as used some constant input in those reversible (fredkin) gates.

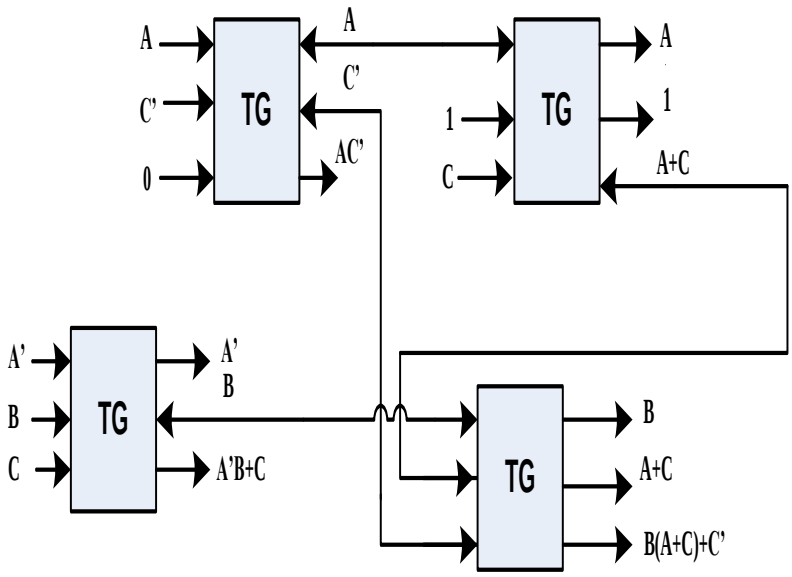


Fig 5.2 : Existing essential function realization with toffoli gate

Using reversible logic in essential function from the recent research reversible (toffoli) gates have been constructed. There are 4 toffoli gates used in this diagram. From gates requisite outputs are received. There are some garbage as well as constant outputs used in those reversible (toffoli) gates.

Table 5.2 : Sorting cost of existing and proposed circuit for 4 bit

Sorting	**Toffoli Cost**	**Fredkin Cost**
Selection	**2 × 4 = 8**	**1 × 6 = 6**
Merge	**4 × 4 = 16**	**1 × 6 = 6**
Insertion	**6 × 4 = 24**	**1 × 6 = 6**
Bubble	**6 × 4 = 24**	**1 × 6 = 6**

Table 5.3 : Sorting cost of existing and proposed circuit for 8 bit

Sorting	**Toffoli Cost**	**Fredkin Cost**
Selection	**$2 \times 4 = 8$**	**$1 \times 6 = 6$**
Merge	**$7 \times 4 = 28$**	**$1 \times 6 = 6$**
Insertion	**$8 \times 4 = 32$**	**$1 \times 6 = 6$**
Bubble	**$8 \times 4 = 32$**	**$1 \times 6 = 6$**

This table shows that, for toffoli the sorting cost of selection, merge, insertion and bubble sorting for 4 bit are respectively 2,4,6,6 and for fredkin, cost are respectively 1,1,1,1. And for 8 bit the corresponding sorting costs are 2,7,8,8 for toffoli and 1,1,1,1 for fredkin respectively. From the gate diagram it can be obtained, in toffoli the gate cost is 4 where in fredkin the cost is 6. For all action fredkin circuit cost is 6 where for every sorting the circuit cost of toffoli can vary. Even in Feymen Double gate the circuit cost is minimum for selection sorting which is optimal from fredkin but in merge, insertion, bubble cost is too high. That is why, Feymen Double gate is locally optimal but not globally. But fredkin is globally optimal and also for reversible circuit fredkin is optimal.

5.2 Summary

Fredkin gate is a fault tolerant reversible gate which allows us to recover from bit loss through unique mapping between inputs and outputs. Additionally fault tolerant property detect ant faulty signal from primary output. In previous chapter we show that our sorting cost is lowest which we prove with four different sorting process. That is why our circuit cost will be always lower than other previous research works.

Chapter 6

Conclusions

6.1 Conclusion

Reversible fault tolerant circuit reclaims bit loss through one-to-one mapping for all possible combination and dissipates zero power and it can detect faulty signal from its primary outputs by reversible parity preserving gates while conventional circuits misuses power by losing bits of information. In order to operate large volume of data using sorting algorithm is quite progressive. The proposed algorithm redacts sorting among optimal implementations of designed essential functions to implement the reversible fault tolerant boolean function and also considers both gate and quantum level circuit costs. The proposed scheme is evaluated with respect to existing approaches which showed that the proposed method performs much and are much more scalable. Comparing existing approaches it is showed that the proposed method performs better and more effective.

6.2 Future Work

In future we will work with quick sort and will try to optimize our proposed sorting cost as if we will also get the equivalent sorting cost 1. For each fredkin gate quantum cost is 5. Since our proposed circuit has 3 fredkin gate therefore, our quantum cost 15. But in existing network quantum cost is 10 as they use 2 toffoli gate. In this consequence, we will work on this to optimize the quantum cost as well.

References

[1] R. Landauer, “Irreversibility and heat generation in the computing process,” *IBM J. Res. Dev.*, vol. 5, no. 3, pp. 183–191, Jul. 1961. [Online]. Available: http://dx.doi.org/10.1147/rd.53.0183

[2] C. H. Bennett, “Logical reversibility of computation,” *IBM J. Res. Dev.*, vol. 17, no. 6, pp. 525–532, Nov. 1973. [Online]. Available: http://dx.doi.org/10.1147/rd.176.0525

[3] M. Shamsujjoha, H. M. Hasan Babu, and L. Jamal, “Design of a compact reversible fault tolerant field programmable gate array: A novel approach in reversible logic synthesis,” *Microelectronics Journal*.

[4] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, “Strengths and weaknesses of quantum computing,” *SIAM J. Comput.*, vol. 26, no. 5, pp. 1510–1523, Oct. 1997. [Online]. Available: http://dx.doi.org/10.1137/S0097539796300933

[5] 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, Chengdu, China, Jun. 2011, pp. 103–107.

[6] M. Shamsujjoha, H. M. Hasan Babu, L. Jamal, and A. R. Chowdhury, “Design of a fault tolerant reversible compact unidirectional barrel shifter,” in *Proceedings of the 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems*, ser. VLSID ’13. Washington, DC, USA: IEEE

Computer Society, 2013, pp. 103–108. [Online]. Available: http://dx.doi.org/10.1109/VLSID.2013.171

[7] M. Shamsujjoha and H. M. Babu, Hasan Babu, "A low power fault tolerant reversible decoder using mos transistors," in *Proceedings of the 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems*, ser. VLSID '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 368–373. [Online]. Available: http://dx.doi.org/10.1109/VLSID.2013.216

[8] S. N. Mahammad and K. Veezhinathan, "Constructing online testable circuits using reversible logic," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, pp. 101–109, 2010.

[9] L. Jamal, M. Shamsujjoha, and H. M. Hasan Babu, "Design of optimal reversible carry look-ahead adder with optimal garbage and quantum cost," *International Journal of Engineering and Technology*, vol. 2, pp. 44–50, 2012. [Online]. Available: http://iet-journals.org/archive/2012/ jan*n* vol*n* 2*n* no*n* 1/349421324456832.pdf

[10] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. Hasan Babu, "Efficient approaches for designing reversible binary coded decimal adders," *Microelectron. J.*, vol. 39, no. 12, pp. 1693–1703, Dec. 2008.

[11] K. Morita, "Reversible computing and cellular automata—a survey," *Theor. Comput. Sci.*, vol. 395, no. 1, pp. 101–131, Apr. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2008.01.041

[12] M. Mohammadi and M. Eshghi, "On figures of merit in reversible and quantum logic designs," *Quantum Information Processing*, vol. 8, no. 4, pp. 297–318, Aug. 2009.

[13] M. S. Islam, M. M. Rahman, Z. Begum, M. Z. Hafiz, and A. A. Mahmud, "Synthesis of fault tolerant reversible logic circuits," *CoRR*, vol. abs/1008.3340, pp. 1–4, 2010.

[14] B. Parhami, "Fault tolerant reversible circuits," Proc. of 40th Asimolar Conf. Signals, Systems, and Computers. Pacific Grove, CA, pp. 1726– 1729, 2006.

[15] 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.

[16] Antoine Berut, Artak Arakelyan, Artyom Petrosyan, Sergio Ciliberto, Raoul Dillenschnei- der, and Eric Lutz. (2012). Experimental verification of landauer/'s principle linking information and thermodynamics. Nature, 483(7388):187–189.

[17] W. N. N. Hung, X. Song, G. Yang, J. Yang, and M. A. Perkowski, "Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis," IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 25, no. 9, pp. 1652–1663, 2006.

[18] Robert Wille and Rolf Drechsler. (2009). Bdd-based synthesis of reversible logic for large functions. In Proceedings of the 46th Annual Design Automation Conference, pages 270–275. ACM.

[19] Robert Wille, Mathias Soeken, D Michael Miller, and Rolf Drechsler. (2014). Trading off circuit lines and gate costs in the synthesis of reversible logic. Integration, the VLSI Journal, 47(2):284–294.

[20] H. M. Hasan Babu, R. Islam, A. R. Chowdhury, and S. M. A. Chowdhury, "Reversible logic synthesis for minimization of full-adder circuit," *Digital Systems Design, Euromicro Symposium on*, pp. 50–53, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=942792. 943112

[21] S. N. Mahammad and K. Veezhinathan, "Constructing online testable circuits using reversible logic," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, pp. 101–109, 2010.

[21] 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.

[22] M. Nielsen and I. Chuang, Quantum computation and quantum information. New York, NY, USA: Cambridge University Press, 2000.

[23] M. P. Frank, "The physical limits of computing," Computing in Science and Engg., vol. 4, no. 3, pp. 16–26, May 2002. [Online]. Available:

http://dx.doi.org/10.1109/5992.998637

[24] M. Perkowski, "Reversible computation for beginners," 2000, lecture series, 2000, Portland state university. [Online]. Available:

http://www.ee.pdx.edu/mperkows

[25] D. Maslov, G. W. Dueck, and N. Scott, "Reversible logic synthesis benchmarks page," 2005. [Online]. Available: http://webhome.cs.uvic. ca/~dmaslov

[26] D. M. Miller, D. Maslov, and G. W. Dueck, “A transformation based algorithm for reversible logic synthesis,” in Proceedings of the 40th annual Design Automation Conference, ser. DAC ’03. New York, NY, USA: ACM, 2003, pp. 318–323. [Online]. Available:

http://doi.acm.org/10.1145/775832.775915

[27] R. K. James, T. K. Shahana, K. P. Jacob, and S. Sasi, “Fault tolerant error coding and detection using reversible gates,” IEEE TENCON, 2007, 2007.

[28] J. Mathew, J. Singh, A. A. Taleb, and D. K. Pradhan, “Fault tolerant reversible finite field arithmetic circuits,” in Proceedings of the 2008 14th IEEE International On-Line Testing Symposium, ser. IOLTS ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 188–189. [Online]. Available: http://dx.doi.org/10.1109/IOLTS.2008.35

[29] M. Shamsujjoha, H.M.H. Babu, A Low Power Fault Tolerant Reversible DecoderUsing MOS Transistor, in: 26th International Conference on VLSI Design andthe 12th International Conference on Embedded Systems, VLSID 2013, Pune,India, 2013, pp. 368–373.