A Novel Approach to Design Reversible Fault Tolerant Barrel Shifter

by

Md.Shamsujjoha
Exam Roll: Curzon - 1601
Registration No: HA - 3765
Session: 2005-2006

A Thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF DHAKA

January 2011

Declaration

I, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by me under the supervision of Ahsan Raja Chowdhury, Assistant Professor, Department of Computer Science and Engineering, University of Dhaka. I also declare that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

21. 12. 2010

....

Ahsan Raja Chowdhury
Supervisor

Signature

....

Md.Shamsujjoha
Candidate

Abstract

Today's computer erases a bit of information each time it performs a logic operation, this erasure of bit dissipate heat. In this regard, reversible or information lossless logic has become an immensely promising technology not only in low power CMOS design and nano-technology based system, but also primal requirement for quantum computing. Bit error is also a big problem for both reversible and irreversible circuits. Fault tolerant circuitry is one of the optimistic solutions for bit error problem. We propose a new method to realize reversible fault tolerant unidirectional barrel shifter, which is capable of rotating multiple bits in a single cycle, thus attains great importance in logic and digital circuits. Proposed unidirectional design outperforms the existing work by number of gates, garbage output produced, delay and quantum cost. More interestingly, we have converted our proposed unidirectional design to a bidirectional one using some extra gate and an extra control input. Reversible design of bidirectional barrel shifter is proposed first time ever in the literature. Most importantly, proposed circuits are designed in a **Fault tolerant** mode which is also first ever proposed in the literature to our best knowledge.

Dedicated to Those People

Who are Honest, Spiritual and Patriot

Who Work for the Welfare of Human Being

Acknowledgements

First of all, I am thankful and expressing my gratefulness to Almighty who offers me His divine blessings, patient, mental and psychical strength to complete this thesis work.

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 thesis supervisor Ahsan Raja Chowdhury, Assistant Professor, Department of Computer Science and Engineering, University of Dhaka. His scholarly guidance, especially for his tolerance with my persistent bothers and unfailing support has helped my thesis achieve its desired goals.

CSEDU's laboratories for Computer Science and Engineering has provided an ideal working environment. I offer special thanks to the Lab assistants who have been particularly supportive and tolerant of my incessant requests. I am also indebted to the library of our department, from where this research began.

I would like to express my deep gratitude to Irina Hashmi and S. K. Mitra. Discussions with them on various topics on my work have helped me to enrich my knowledge and conception regarding this work.

I am highly grateful to my teachers and family members for their support and constant encouragement, which have always been a source of great inspiration for me.

Finally, I would like to thank all the researchers within this field whose profound papers, reports and journals have helped cultivate my interest in the subject of reversible, fault tolerant and quantum computing.

Contents

List of Figures

List of Tables

Contribution from the Thesis

The thesis has three principal contributions.

1. “A Novel Approach to Design Reversible Fault Tolerant Barrel Shifter” is submitted to the 21st edition of GLSVLSI conference.

2. “Reversible Fault Tolerant Barrel Shifter: A Novel Approach in Reversible Logic Synthesis” is going to be submitted at ACM Journal on Emerging Technologies in Computing Systems.

3. “A Novel Approach to Design Reversible Fault Tolerant Bidirectional Logarithmic Barrel Shifter” is going to be submitted in the Journal of ACM Transactions on Computer Systems .

I don't know anyone who has gotten to the top without hard work. That is the recipe. It will not always get you to the top, but it will get you pretty near.

–Margaret Thatcher, Britain's first female prime minister.

Chapter 1

Introduction

Modern VLSI enriched with CMOS technology has low power consumption. But the demand would make it more power hungry for larger architecture to reload the erased bits. Since conventional irreversible circuit erases bit each time it perform logic operation and erasure of single bit dissipate $KTln2$ joules of heat-energy where K is Boltzmann's constant of 1.38×10^{-23} and T is the absolute temperature of the environment. The amount of energy dissipation increases in direct propagation to the number of computed bits [1].

Reversibility means that each state in a computing system can be uniquely determined from its previous state. In Reversible circuit, number of output is equal to the number of input and there is a one to one correspondence between the vector of input and output, i.e. it can generate unique output vector from each input vector and vice versa [2, 3]. Thus reversible system doesn't allow information to be erased. Bennett showed that heat generated by reversible computer is exactly $KTln1$ which is logically zero [4].

1.1 Motivation Towards the Research

Reversible circuit can be viewed as a special case of quantum circuit because quantum evolution must be reversible [5]. Quantum computing is gaining popularity now a days, as it solved some exponentially hard problems in polynomial time [6]. Thus reversible design methods might simulate efficient and new methods of quantum circuit construction, resulting in much more powerful computers and computations in future.

Reversible Fault tolerant circuit preserves parity between input and output for all input-output sequences. Therefore solve's both problem of bit loss and bit error. Quantum technology is not the only one where reversibility and fault tolerance is used. Both of this gained remarkable interests over the past few years and it has found its applications in several technologies, such as optical, nano-technology and DNA technology [7].

On the other hand data shift and rotate operations are widely used in variable length encoding, arithmetic operations, bit-indexing and over all on fast calculations. Shifting and rotating data by using controlled inputs, barrel shifter is best and optimistic. In this consequence, barrel shifter which is capable of performing n-bit shifting and rotating of data in a single cycle, are normally used in embedded processors such as, digital signal processors [8], high performance processors [9], high-speed or low-power applications [10] etc. However, only two papers have been published so far on reversible barrel shifter [11, 12].

Thus this research focuses on designing (n, k) reversible fault tolerant barrel shifter, where n is the number of data bits and k in the number of control inputs equals to $log_2 n$.

1.2 Methodologies of the Research

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

- Understanding reversibility and its importance in low power circuitry.
- Understanding the basics of quantum computation and its synthesis.
- Understanding the basics of fault tolerance and its synthesis.
- Understanding various existing reversible and fault tolerant logic gates.
- Designing various reversible fault tolerant combinational circuits.
- Analyzing several irreversible barrel shifter and examining their structural properties.
- Analyzing the existing reversible barrel shifters, its design and working procedure, also its advantages and shortcomings.
- Inventing the idea for designing a fault tolerant barrel shifter circuitry.
- Contriving reversible fault tolerant barrel shifter.
- Performing a comparative study with the existing designs.

1.3 Organization of the Thesis

The organization of the thesis is as follows:

`Chapter 2` briefly discuss background and previous work relating to reversible and fault tolerant computing. The study includes understanding of the basic idea of reversible and fault tolerant logic, their importance, empathizing different basic reversible and fault tolerant gate and quantum realization of them.

`Chapter 3` introduces barrel shifters and shift rotate operations in details.

`Chapter 4` describes exiting design of unidirectional reversible barrel shifters.

`Chapter 5` depicts the design and working procedure of the proposed reversible fault tolerant unidirectional barrel shifter, its performance and comparison with existing one.

`Chapter 6` shows design and working procedure of the proposed reversible fault tolerant bidirectional barrel shifter. This chapter also presents performance of proposed bidirectional scheme and rate of change of parameters.

`Chapter 7` presents a conclusion which is drawn based on the study of proposed methods. Also future enhancement of proposed schemes is given here.

1.4 Summary

This chapter demonstrates motivation for the research on reversible and fault tolerant computing. The system of methods that is being followed throughout the research work is also discuss here. A brief elementary instructional text of remaining chapters is given in the last part of this chapter as well.

Chapter 2

Basic Definitions and Literature Overview

This chapter start with some basic definitions and literature overview on reversible logic for future reference. Formal illustration of common reversible logic gates and their equivalent quantum realization also given here.

2.1 Grandness of Reversible Logic

Maxwell and Szilard first ascertained the connection between a single bit and corresponding minimum quantity of entropy [13, 14]. Summary of their observation is, "a minimum amount of energy (***KT ln(m)***, where K is Boltzmann's constant of 1.38×10^{-23}, T is the absolute temperature of the environment and m is an integer number proportional to number of computed bits) must be dissipated during every elemental enactment of computation."In [1] Landauer proved that energy dissipation is only avoidable if the system is made reversible. In [4] Bennett showed that a reversible computation, in which no information is destroyed, dissipate an arbitrarily small amount of energy ($KT\,ln(1)$), which is logically zero. This observation entail's that, computing system can avoid energy dissipation if it is made reversible. In other words a computing system can operate having asymptotically zero energy loss if it is made reversible. Landauer and Bennett also showed that, irreversible system has a fundamental lower limit to the energy dissipated during a computation, proportional to the number of bit erased ($KT\,ln2$, *for each erased bit*). An irreversible system may store information which is produced during a computation rather than erasing it but doesn't provide a unique path from each state to its previous state. Energy used to store this information is unrecoverable unless the system is made reversible. Therefore to be fully reversible, a computing system must be reversible at all levels. Reversible computers must have a reversible power supply, which power's the reversible circuits made of reversible gates. Today's computer erases bit of information every time it performs a logic operation. This erasure is done very inefficiently and heat is dissipated for each bit erased. If we are to continue the revolution in computer hardware performance we must reduce these energy dissipation which occurs because of bit loss while performing logic operation. Although power dissipation due to information loss is negligible under current technologies but it will become a substantial part of energy dissipation by 2020 as a result of increased density in computer hardware, if Moore's Law continues to be in effect [15]. G. Moore predicted in [16] that,

> *"The number of transistors and resistors on a chip doubles every 18 months "*

Which signifies that, *"it would continue to improve at an exponential rate with the performance per unit cost increasing by a factor of 2 every 18 months or so"*. This assumptions has great importance lies in the technological necessity that, every future technology will have to be reversible in order to reduce power or any other new technology should be invented which has lees power consumption than conventional one. Otherwise overheating in a chip will demolish it.

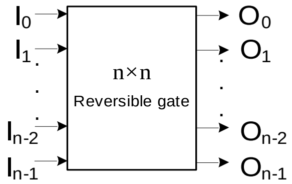

FIGURE 2.1: Block diagram of a $n \times n$ Reversible gate

2.2 Reversible Gate

A $n \times n$ reversible gate is a data stripe block which uniquely maps between input vector $I_v = (I_0, I_1, ..., I_{n-1})$ and output vector $O_v = (O_0, O_1, ..., O_{n-1})$ denoted as $I_v \leftrightarrow O_v$. Figure 2.1 shows the block diagram of a $n \times n$ reversible gate. Since reversible gate provide one to one mapping among all input and output for each input-output sequence, hence it doesn't loose information i.e. if we are able to design a circuit using only reversible gate it will provide a unique path from each stage to it's previous stage, thus entire circuit becomes information lossless, therefore will not dissipate any heat.

2.3 Fault Tolerant Gate

A Fault tolerant gate is a reversible gate which constantly preserve same parity between input and output [17], more specifically a $n \times n$ fault tolerant gate clarify the following properties between the input vector, $I_v = (I_0, I_1, ..., I_{n-1})$ and the output vector $O_v = (O_0, O_1, ..., O_{n-1})$.

$$I_0 \oplus I_1 \oplus ... \oplus I_{n-1} = O_0 \oplus O_1 \oplus ... \oplus O_{n-1}$$

Circuit consisting of only fault tolerant gates preserves parity. In other words if entire circuit is designed using only reversible fault tolerant gate then entire circuit will become information lossless as well as fault tolerant [18].

2.4 Quantum Computation and Reversibility

First proposed in the 1970s, quantum computing relies on quantum physics, taking advantage of certain quantum physics properties of atoms or nuclei that allow them to work together as quantum bits, or qubits. Qubits can perform certain calculations exponentially faster than conventional computers [19]. Quantum computers encode information as a series of quantum-mechanical states such as spin directions of electrons or polarized orientations of a photon that might represent as $|0 > or|1 >$, or might represent a superposition of the two values. Quantum computation uses matrix multiplication rather than Boolean operations and the information measurement is realized using qubits rather than bits [20].

Quantum computer demand underneath circuitry be reversible. This imposes extra costs when doing classical computations on a quantum computer, which can be necessary in subroutines of quantum computations [21]. So, calculating quantum cost of a circuit make sense. In the research field of Reversible logic synthesis and Quantum synthesis, minimizing quantum cost of the circuit is one of the major challenges, which can be accomplish by redesigning the logic. Every reversible or quantum circuit is built from 1×1 and 2×2 quantum primitives and its cost calculated as a total sum of 2×2 gates used since 1×1 gate cost nothing i.e. zero. All gates of the form 2×2 has equal quantum cost and the cost is unity i.e. one [22]. Since every reversible gate is a combination of 1×1 or 2×2 quantum gate, therefore the quantum cost of a reversible gate is the total number of 2×2 gates used as cost of 1×1 gate is zero. Hence to calculate total quantum cost of a reversible circuit, we need to calculate total number of 2×2 gates used.

2.5 Garbage Output

To maintain the reversibility of a reversible logic gate several dummy output signal(s) are needed to be produced in order to equalize the number of input to that of output. These signal(s) are commonly known as Garbage signal(s), Garbage output(s) or garbage bit(s). Heavy price is paid off for each garbage output [23, 24]. In other words, an output of a reversible logic gate which is not used as a primary output or input to another gate is known as garbage output. For example, to perform exclusive-or (XOR) operation between two input, we can use 2×2 reversible Feynman gate (FG) which produces an extra dummy output along with desire output signal to preserve reversibility. This extra output is the garbage output, denoted by P in Figure 2.2.

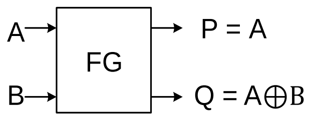

FIGURE 2.2: Reversible Feynman gate

2.6 Reversible Circuit Delay

In a logic circuit or reversible logic circuit, the path consists of most number of gates from any input to any output is known as critical path. There may be more than one critical path in a circuit. The delay of a logic circuit or reversible logic circuit is the delay of the critical path. This definition is based on the following two assumptions [23]:

- Each gate performs 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 input to the circuit are known before the computation begins. Which means that, internal structure and each operation of gate is known before the calculation.

From the above definition, delay of the logic circuit which consist only one gate is obviously one as this is the only gate from any input to output line. Therefore delay of Figure 2.2 is one.

2.7 Restrictions on Reversible Logic Synthesis

In terms of state mapping reversible circuitry is the only solution which can regain every computed stage through backward order. But designing reversible circuit using reversible gate need to maintain several following restrictions :

- More than one fan-out is not allowed. In other word if an input is needed more than once, then copying gate should be used to produced more copies of that input.
- Loop is strictly restricted i.e one can't use feedback while designing circuitry in reversible logic synthesis.

2.8 Design Criteria

Efficient designing of reversible circuit has following designing criteria :

- Minimize garbage outputs as much as possible i.e. use as many outputs of every reversible gate as possible.
- Do not create more constant input to gate unless it is absolutely required i.e. try to avoid constant input to gates.

- Use as less number of reversible gates as possible to achieve the goal.
- Try to minimize quantum cost and delay of the circuit as much as possible.

2.9 Common Reversible Gates

Following subsections illustrate some commonly used reversible gate, then next section presents their equivalent quantum realization.

2.9.1 Feynman Gate

Reversible Feynman gate is one of the most widely used reversible gate. The input vector, I_v and output vector, O_v for 2×2 Feynman gate (FG) is defined as follows [2]: $I_v = (A, B)$ and $O_v = (A, A \oplus B)$. Block diagram of Feynman gate is shown in Figure 2.2.

Normally Feynman gate is used as copying gate. As number of fan-out bounded to one in reversible logic synthesis, using feynman gate more than one copy of a input can be created. For example If input vector $I_v = (A, B = 0)$ then output vector of Feynman gate will be $O_v = (A, A)$ shown in Figure 2.3.

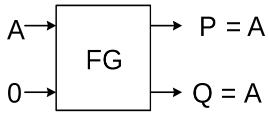

Figure 2.3: Feynman gate as copying gate

Table 2.1: Truth table of Feynman double gate

A	B	C	P	Q	R	Parity
0	0	0	0	0	0	Even
0	0	1	0	0	1	Odd
0	1	0	0	1	0	Odd
0	1	1	0	1	1	Even
1	0	0	1	1	1	Odd
1	0	1	1	1	0	Even
1	1	0	1	0	1	Even
1	1	1	1	0	0	Odd

2.9.2 Fault Tolerant Feynman Double Gate

Feynman double gate (F2G) is another popular reversible gate which is also a Fault tolerant gate according to [2, 18]. The input vector I_v and output vector O_v for 3 × 3 Feynman double gate (F2G) is defined as follows : $I_v = (A, B, C)$ and $O_v = (A, A \oplus B, A \oplus C)$. Figure 2.4 shows the block diagram of the reversible Feynman double gate. Parity preserving property of Feynman double gate can be determined from Table 2.1.

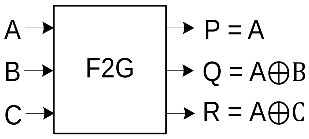

Figure 2.4: Block diagram of Fault tolerant Feynman double gate

Feynman double gate can also be used as copying gate like Feynman gate for any fault tolerant circuit. If $I_v = (A, B = 0, C = 0)$ then $O_v = (A, A, A)$ shown in figure 2.5.

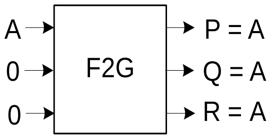

Figure 2.5: Feynman double gate as copying gate

2.9.3 Toffoli Gate

Another commonly used reversible gate is Toffoli gate. Input and output vectors for 3 × 3 Toffoli gate (TG) is defined as follows [3]: Input vector $I_v = (A, B, C)$ and output vector $O_v = (A, B, AB \oplus C)$. Block diagram of reversible Toffoli gate is shown in Figure 2.6.

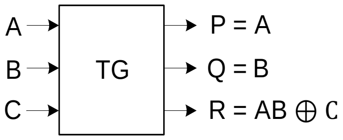

FIGURE 2.6: Block diagram of Toffoli gate

Using Toffoli gate one can implement logic operation such as, AND, NAND, NOT etc. Figures 2.7(a) and 2.7(b) shows how toffoli gate can be used as NAND and NOT gate respectively.

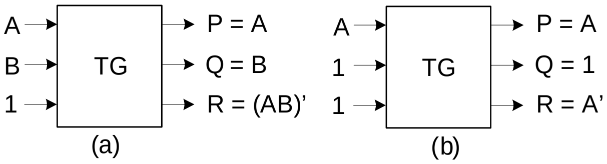

FIGURE 2.7: Toffoli gate as (a) NAND gate (b) NOT gate

2.9.4 Fault Tolerant Fredkin Gate

Fredkin gate is also extensively used by the researchers. Moreover this is one of only two 3 × 3 reversible gate which is Fault tolerant as well [18]. The input and output vectors for 3 × 3 Fredkin gate (FRG) are defined as follows [3]: $I_v = (A, B, C)$ and $O_v = (A, A'B \oplus AC, A'C \oplus AB)$. Its block diagram shown in Figure 2.8. Parity preserving property of this gate can be determined from Table 2.2.

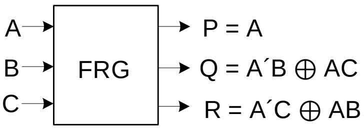

Figure 2.8: Block diagram of Fault tolerant Fredkin gate

Table 2.2: Truth table of Fredkin gate

A	B	C	P	Q	R	Parity
0	0	0	0	0	0	Even
0	0	1	0	0	1	Odd
0	1	0	0	1	0	Odd
0	1	1	0	1	1	Even
1	0	0	1	0	0	Odd
1	0	1	1	1	0	Even
1	1	0	1	0	1	Even
1	1	1	1	1	1	Odd

Being a fault tolerant gate Fredkin gate can be used to implement logical AND, OR, NOT operations. Fredkin gate can also be used to swap 2^{nd} and 3^{rd} input using 1st one as a control input, shown in Figures 2.9(a) and 2.9(b) respectively.

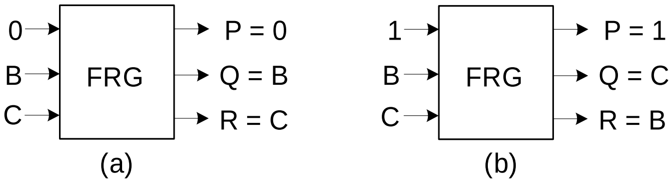

Figure 2.9: Fredkin gate as swapping gate

2.9.5 Peres Gate

The input and output vectors for 3 × 3 reversible Peres gate (Peres) are defined as follows [25]: $I_v = (A, B, C)$ and $O_v = (A, A \oplus B, AB \oplus C)$. Its block diagram is shown in Figure 2.10.

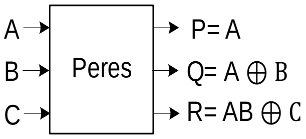

FIGURE 2.10: Block diagram of Peres gate

2.9.6 New Gate

Reversible New gate is also known as Khan gate, named after its inventor [26]. Input and output vectors for 3 × 3 New gate (NG) are defined as follows: $I_v = (A, B, C)$ and $O_v = (A, AB \oplus C, A'C' \oplus B')$ shown in Figure 2.11.

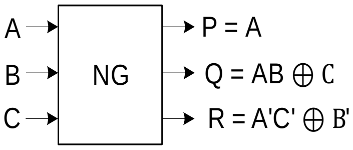

FIGURE 2.11: Block diagram of New gate

2.9.7 HNFG Gate

The input and output vectors for 4 × 4 reversible HNFG gate [27] are defined as follows: $I_v = (A, B, C, D)$ and $O_v = (A, A \oplus C, B, B \oplus D)$. Its block diagram shown in Figure 2.12.

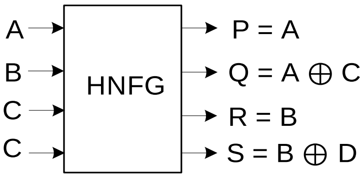

FIGURE 2.12: Block diagram of HNFG gate

2.10 Quantum Analysis of Reversible Gates

Reducing quantum cost from reversible circuit is always a challenging one and works are still going on in this area. Equivalent quantum realization of commonly used reversible and fault tolerant gate, is given in following subsections.

2.10.1 Feynman Gate

2×2 Feynman gate is also called controlled not (CNOT) gate, because if its input vector is set as, $I_v = (A = 1, B)$ then the output vector will be $O_v = (1, B')$. Every linear reversible function can be built by using only 2×2 Feynman gates and inverters. Since this is a 2×2 gate, according to our discussion on section 2.4 quantum cost of Feynman gate is one [7]. Quantum equivalent circuit of Feynman gate is shown in Figure 2.13.

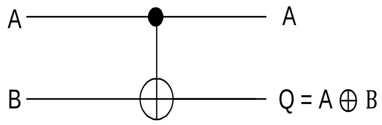

Figure 2.13: Quantum equivalent representation of reversible Feynman gate

2.10.2 Feynman Double Gate

3×3 Feynman double gate (F2G) work alike Feynman gate, but in a fault tolerant mode. Quantum equivalent circuit of Feynman double gate is shown in Figure 2.14. Quantum cost of Feynman double gate is 2 according to proposed method in [7].

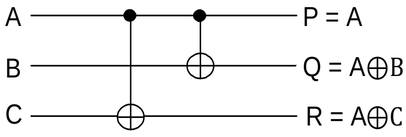

Figure 2.14: Quantum equivalent representation of F2G

2.10.3 Toffoli Gate

Equivalent Quantum realization 3×3 Toffoli gate shown in Figure 2.15. In Figure 2.15 V is a square root-of-NOT (SRN) gate and V^+ is its hermitian. Thus VV creates a unitary matrix of NOT gate. And $VV^+ = \mathrm{I}$, an identity matrix describing just a quantum wire [7]. Quantum cost of Toffoli gate is five as it is equivalent to five 2×2 gates.

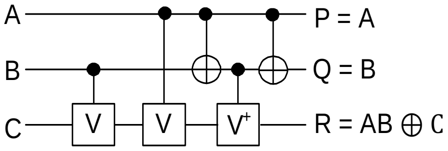

Figure 2.15: Quantum equivalent representation of Toffoli gate

2.10.4 Fredkin Gate

Quantum cost of Fredkin gate is same as Toffoli gate. Quantum equivalent representation of Fredkin gate is shown in Figure 2.16. In this figure, cost of each rectangle is equivalent to 2×2 quantum XOR gate, that is one. Thus total quantum cost of reversible fault tolerant Fredkin gate is five [7].

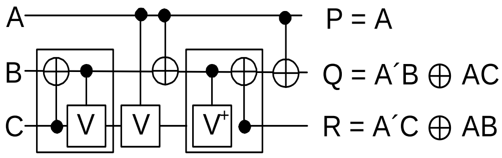

Figure 2.16: Quantum equivalent representation of FRG

2.10.5 Peres Gate

Reversible Peres gate can be realized with quantum cost 4. It is cheapest realization of a universal 3×3 permutation gate [28]. Quantum equivalent representation of Peres gate is shown in Figure 2.17. It is just like a Toffoli gate but without the last Feynman gate from right.

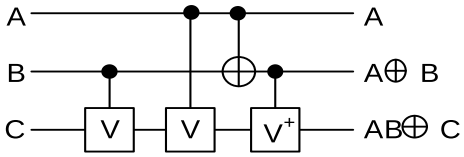

FIGURE 2.17: Quantum equivalent representation of Peres gate

2.10.6 New Gate

Reversible New gate can be realized with quantum cost 11. It is one of most expensive quantum realization of a 3×3 permutation gate. Quantum equivalent representation of reversible New gate is shown in Figure 2.18 [26].

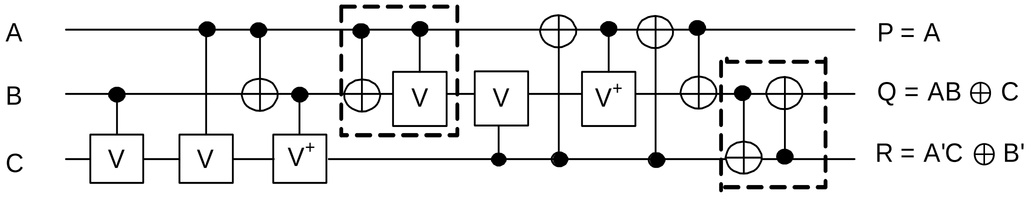

FIGURE 2.18: Quantum equivalent representation of New gate

2.10.7 HNFG Gate

Reversible HNFG gate can be realized with quantum cost 2 [27]. It is cheapest realization of a 4×4 permutation gate. HNFG gate is very close to Feynman gate but it incurs an additional quantum cost. Quantum equivalent representation of reversible HNFG gate is shown in Figure 2.19.

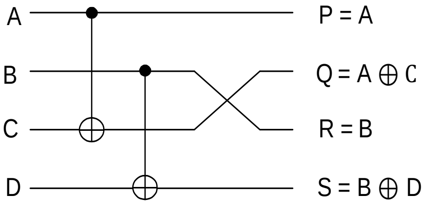

Figure 2.19: Quantum equivalent representation of HNFG gate

2.11 Summary

A brief literature overview, and the related terminologies regarding reversible and Fault tolerant logic synthesis are presented in this chapter. Definition of several existing reversible and fault tolerant logic gates devoted here as well. Equivalent quantum representation of different reversible and fault tolerant logic gates have also been depicted in this chapter.

Chapter 3

Introduction to Barrel Shifter and Shift Rotate Operations

Barrel shifter is a fundamental part of digital system which has n input-output lines for data transmission and k control inputs, where k=$log_2 n$ [29]. Since multiple and variable bit shifting are more desirable than single bit operations, several irreversible barrel shifters have been proposed. Barrel shifters can be unidirectional which performs shift-rotate to a direction (left or right), or Bidirectional that are capable of shifting-rotating to both directions (left and right). Various shift and rotate operations, common design choice for barrel shifters, their structures and working procedures is instanced in following sections.

3.1 Shift and Rotate Operations

Shift and rotate operations are indispensable elements in most processors. They are utilitarian on their own, but also used in multiplication and division modules. Shift operation can be further broken down into logical shift and arithmetic shift. All this operations can have left or right as a direction.

3.1.1 Logical Shift

In binary computer, left logical shift has same effect as multiplication by 2, and a right logical shift has same effect as a division by 2. Left logical shift operation moves each bit in a bit string one position to the left e.g. bit one moves into bit position zero, bit two moves into bit position one and so on. Right logical shift operation moves each bit in a bit string one position to the right e.g. bit zero moves into bit position one, bit one moves into bit position two and so on [30]. However, in both of this two cases zero is inserted in empty position. Figures 3.1(a) and 3.1(b) represent left and right logical shift for respectively.

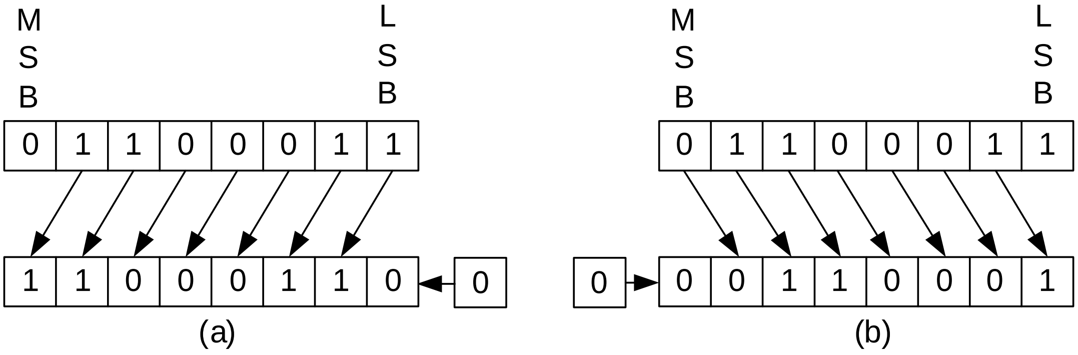

FIGURE 3.1: (a)Left Logical shift. (b)Right Logical shift

3.1.2 Arithmetics Shift

There is no difference between left arithmetic shift and left logical shift. Right arithmetic shift is almost same as right logical shift with the difference that, it always keeps its sign bit intact i.e. zeros are inserted into the empty positions except the first one, which keeps the sign bit and it remains unchanged [31]. Hence arithmetic shift is suitable for signed two's complement binary numbers and logical shift is suitable for unsigned binary numbers. Figures 3.2(a) and 3.2(b) demonstrate left and right arithmetic shift respectively.

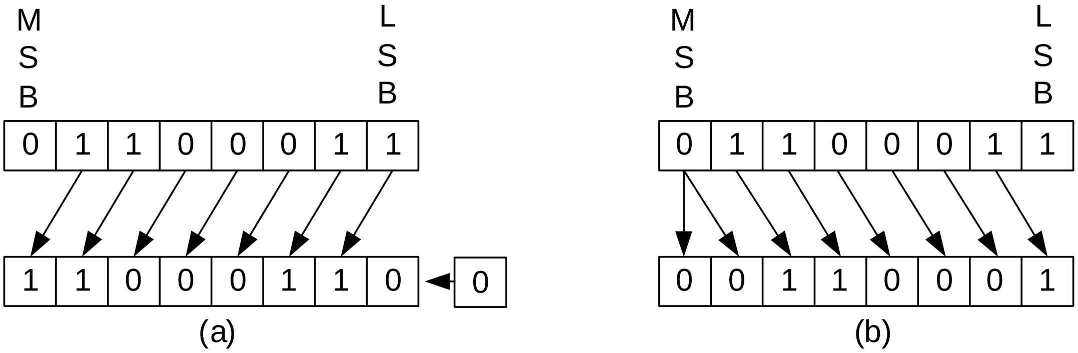

FIGURE 3.2: (a)Left Arithmetic shift. (b)Right Arithmetic shift.

3.1.3 Rotate Operation

Rotate operations is similar to shift operations with the difference that rotate operation is circular, with the bits shifted out one end returning on the other end [32]. Figures 3.3(a) and 3.3(b) depict left and right rotate operations respectively.

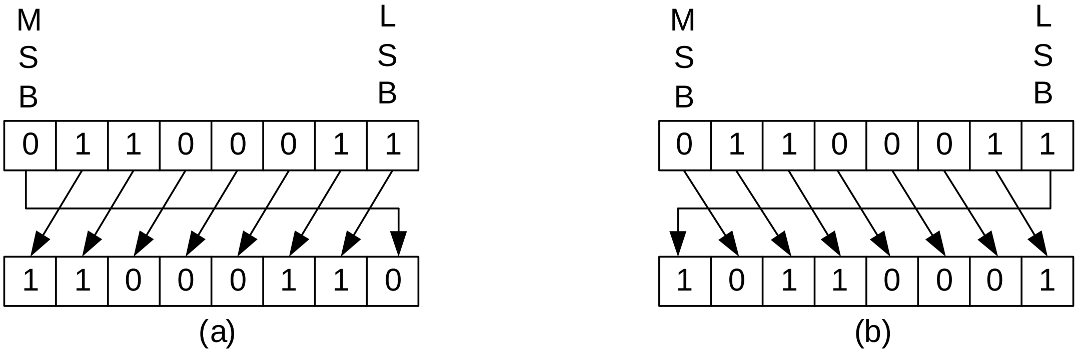

FIGURE 3.3: (a)Left rotate operation. (b)Right rotate operation.

In all the above figures of shift-rotate operations, a single bit operation is depicted, two bit operation means, perform a operation on initial input get intermediate result then again perform another operation on that intermediate result to get final result. n bit operation means do this n times. Table 3.1 summarize all the shift rotate operations. In this table we consider our input is $I_0I_1I_2I_3I_4I_5I_6I_7$, Where each $I_j\,(j = 0\,to\,7)$ represent's a bit.

Table 3.1: Shift and rotate operations

Operations	Output
3-bit left logical shift	$I_3I_4I_5I_6I_7000$
4-bit right logical shift	$0000I_0I_1I_2I_3$
2-bit left arithmetic shift	$I_2I_3I_4I_5I_6I_700$
5-bit right arithmetic shift	$I_00000I_0I_1I_2$
6-bit left rotation	$I_6I_7I_0I_1I_2I_3I_4I_5$
3-bit right rotation	$I_5I_6I_7I_0I_1I_2I_3I_4$

3.2 Basic Irreversible Barrel Shifters

Barrel shifter is a part of a microprocessor CPU can typically specify direction of shift, type of shift and amount of shift [33]. Barrel shifter which perform left rotate operation, shifted out the most significant bit (MSB) end of the register are shifted back into the least significant bit (LSB) end of the register. For example, suppose an eight-bit barrel shifter rotate data by three positions to the left in a single clock cycle, then if the original data was 10101110, one clock cycle later the result will be 01110101 [34].

3.2.1 Logarithmic Shifter

An n-bit unidirectional logarithmic barrel shifter has n input bits and total of k *(k = log_2n)* stages which are controlled by k control bits shown in Figure 3.4.

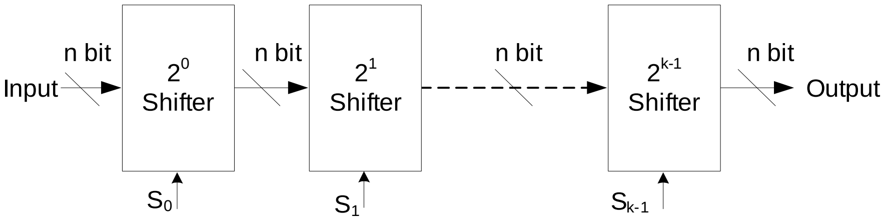

Figure 3.4: Block diagram of logarithmic unidirectional barrel shifter

TABLE 3.2: Working procedure of unidirectional logarithmic barrel shifter

Control Signal	Output
S_0 = Low, all the other control signals is Low	$I_0\, I_1\, I_2\, I_3 \ldots I_{n-3}\, I_{n-2}\, I_{n-1}$
S_0 = High, all the other control signals is Low	$I_1\, I_2\, I_3 \ldots I_{n-3}\, I_{n-2}\, I_{n-1}\, I_0$
S_1 = Low, all the other control signals is Low	$I_0\, I_1\, I_2\, I_3 \ldots I_{n-3}\, I_{n-2}\, I_{n-1}$
S_1 = High, all the other control signals is Low	$I_2\, I_3 \ldots I_{n-3}\, I_{n-2}\, I_{n-1}\, I_0\, I_1$
(S_0, S_1) = High, all the other control signals is Low	$I_3 \ldots I_{n-3}\, I_{n-2}\, I_{n-1}\, I_0\, I_1\, I_2$

Control bit S_j $(where\ j\ =\ 0\ to\ k-1)$ of each stage determines whether to shift or not to shift the input data for that stage. If the control bit S_j is set to high then jth stage will shift the input 2^j times otherwise the input will remain unchanged. Working procedure of unidirectional logarithmic shifter is shown in Table 3.2. In this table we consider our unidirectional logarithmic shifter perform left rotation while control input (S_j) is set to high, and our input vector is, $I_0\, I_1\, I_2\, I_3 \ldots I_{n-3}\, I_{n-2}\, I_{n-1}$. Bidirectional logarithmic shifter is same as unidirectional one with the extension that it has another control signal which determined the direction of shift and rotate on each stage. Its block diagram is shown in Figure 3.5, from where we access that now each stage shift to a specific direction depending on the value direction determiner control signal (S_{LR}). Also stages shift only when control signal set to high. This means whatever the value of S_{LR}, shifting will be done only when S_j is set to high. Thus if S_{LR} is set to left and S_0 set to high then stage 0 will shift the input 2^0 times left and whatever the value of S_{LR}, if S_0 set to low then stage 0 will not perform any shifting [35].

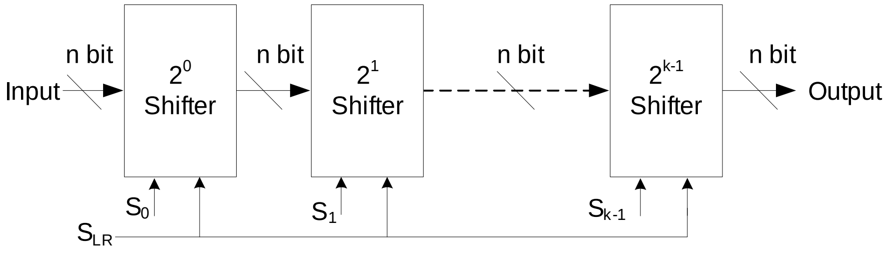

FIGURE 3.5: Block diagram of logarithmic bidirectional barrel shifter

3.2.2 Array Shifter

Array shifters are more faster than logarithmic shifter but they take more area and has a complex structure. An array shifter is been proposed in [36]. An Array shifter with 32 bit data value and 5 bit control input is shown in Figure 3.6.

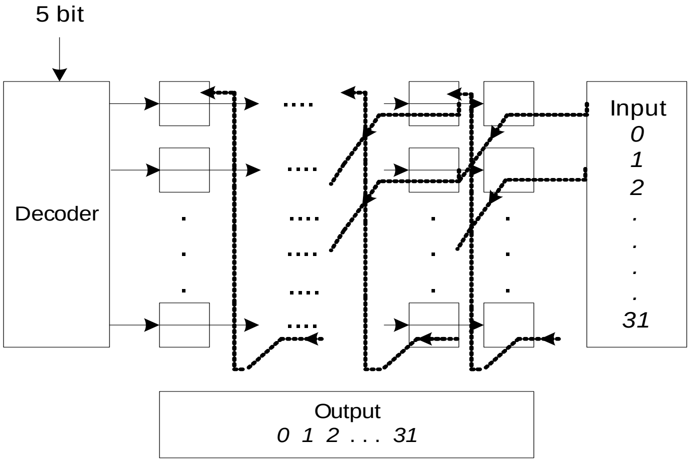

FIGURE 3.6: Block diagram of array shifter

3.2.3 Combinational Shifter

Architecture of a combinational shifter is shown in Figure 3.7. It shifts by transmitting an n-bit slice of the $2n$ data bits to the outputs. This shifter with n-bit input and with n output bits is built from $2n$ vertical by n horizontal array of cells, each of which has a single transistor and few wires. Control line values are set so that exactly one is high, which turns on all the transmission gates on that column. Transmission gates connect the diagonal input wires to the horizontal output wires. When a column is turned on, all the inputs are shunted to the outputs. The length of the shift is determined by the position of the selected column, further to the right it is, the greater the distance the input bits have traveled upward before being shunted to the output [37].

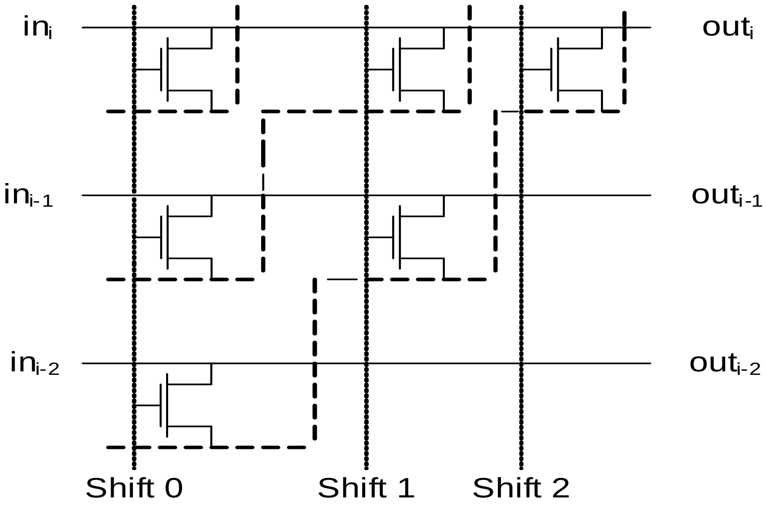

Figure 3.7: Block diagram of combinational shifter

3.3 Summary

In this chapter, basic shift and rotate operations are discussed. Then the related works on several irreversible barrel shifters are reviewed. Working procedure of Logarithmic shifter and combinational shifter is being illustrated as well.

Chapter 4

Existing Reversible Barrel Shifters

Two paper have been publish so far on reversible barrel shifter both of which are unidirectional shifters. First paper on reversible unidirectional barrel shifter was proposed in 2007 at [11]. This design consists of multiplexer. Recently in 2010 efficient design of unidirectional reversible barrel shifter was proposed at [12].

4.1 Architecture

Design approach from [12] greatly outperforms the work of [11]. The existing reversible barrel shifter from [11] and [12] with 4-bit data value and 2-bit control value is shown in Figures 4.1 and 4.2

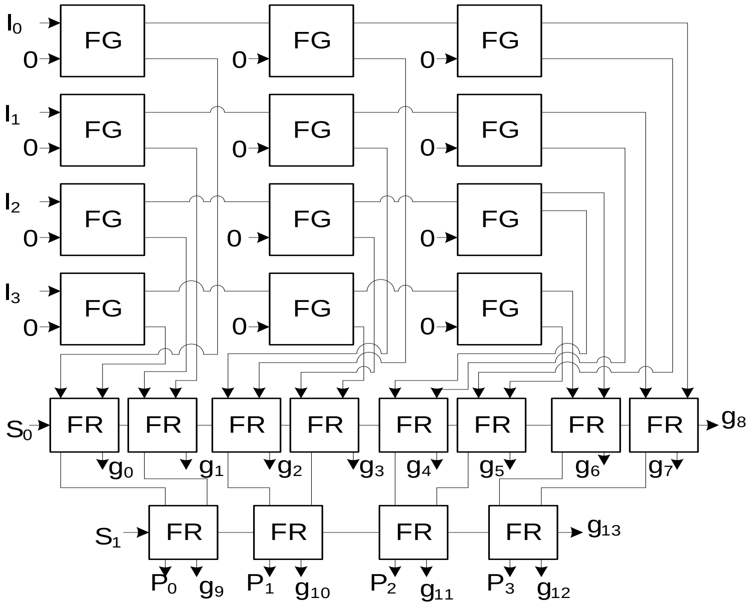

Figure 4.1: Existing (4, 2) reversible unidirectional barrel shifter [11]

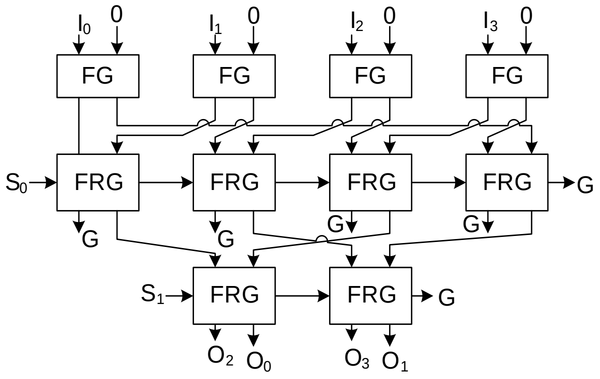

Figure 4.2: Existing (4, 2) reversible unidirectional barrel shifter [12]

4.2 Properties of Existing Design

As mentioned earlier, latest work on reversible barrel shifter from [12] outperforms the work of [11]. So we are exclusively illustrating the properties of [12]. According to [12], shifter with n-bit data value and k-bit shift value have the following properties.

- Number of Fredkin gates, $FR = n(k-1) + n/2$
- Number of Feynman gates, $FG = n(k-1)$
- Number of garbage outputs produced, $GO = n(k-1) + k$

Table 4.1 express the summary of number of gate used, total number of Garbage output, Delay and Quantum cost of the design proposed in [12].

Table 4.1: Existing reversible unidirectional barrel shifter properties [12]

(n, k)	FRG [2]	FG [3]	Delay [4]	QC [5]	GO [6]
(4, 2)	6	4	6	34	6
(8, 3)	20	16	14	116	19
(16, 4)	56	48	28	328	52
(32, 5)	144	128	52	848	133

4.3 Shortcomings of Existing Design

Existing shifter from [12] is the 2^{nd} proposed reversible barrel shifter in the literature. However existing design proposed in [12] exhibits several design complexities. Principal problems with the design proposed in [12] are following:

- Design architecture from [12] has complex supernumerary.
- In-terms of required number of gates design of [12] do something to a greater degree than the design of [11], but even than design of [12] uses large number of reversible gates. As a result has prominent delay and large quantum cost.

[2] Required number of Fredkin gates

[3] Required number of Feynman gates

[4] Delay of the circuit

[5] Total Quantum cost of the circuit

[6] Total number of Garbage output

- Design from [12] produces huge number of garbage outputs. Since heavy price is paid off for each garbage output, hence in that sense this design is too costly.

Moreover none of the existing reversible unidirectional barrel shifters has the fault tolerant capacity. Thus there is great scope for improving the circuit complexity by reducing the number of gates, garbage outputs, delay and quantum cost. Also design in a fault tolerant mode will provide extra supremacy.

4.4 Summary

In this chapter related works on existing two reversible barrel shifters are reviewed. Then their structure are shown. Properties and problem with the latest existing design have also been depicted.

Chapter 5

Proposed Reversible Fault Tolerant Unidirectional Barrel Shifter

Full barrel shifter are often on the critical path, which directed most researchers towards speed optimization. Speed optimization actually depends on efficient designing. While designing reversible fault tolerant barrel shifter we focus on how to improve the circuit complexity proposed in [12] in-terms of gate, delay, quantum cost and garbage output produces. Also how to design in a fault tolerant mode. For this purpose we use only fault tolerant Fredkin gate, therefore according to our discussion on section 2.3 and 2.9.4 entire circuit becomes fault tolerant. Design and working procedure of Reversible Fault Tolerant Unidirectional Logarithmic Barrel Shifter is being depicted with an illustration in the nest few sections, followed by some theoretical explanation. In the last part of this chapter a comparative study between the existing design [12] and proposed design is given to access the performance of proposed scheme (as [12] already outperform the work of [11] nothing is presented anymore regarding [11]).

5.1 Design Procedure of Proposed Unidirectional Shifter

While designing Reversible Fault Tolerant Unidirectional Logarithmic Barrel Shifter we follow the adaptive structure of logarithmic shifter e.g. 1^{st} stage shift/rotate 2^0 bit, 2^{nd} stage shift/rotate 2^1 bits, 3^{rd} stage shift/rotate 2^2 bits and so on. Shifting/rotation will occur only when control signal is set to high. If any stage's control signal is set to low then instead of shifting, that stage will just pass the input to its next. Thus if no control signal is set to high final output will remain same as initial input. Adaptive structure of proposed (8, 3) unidirectional barrel shifter is shown in Figure 5.1. In this Figure circles are represented as signal propagation unit which can be realized by single or part of Fredkin gate except final stage. Final stage generates output as sequence that it receives from earlier stage. Based on this adaptive structure we are driving the Algorithm for proposed unidirectional shifter (shown in Algorithm 1).

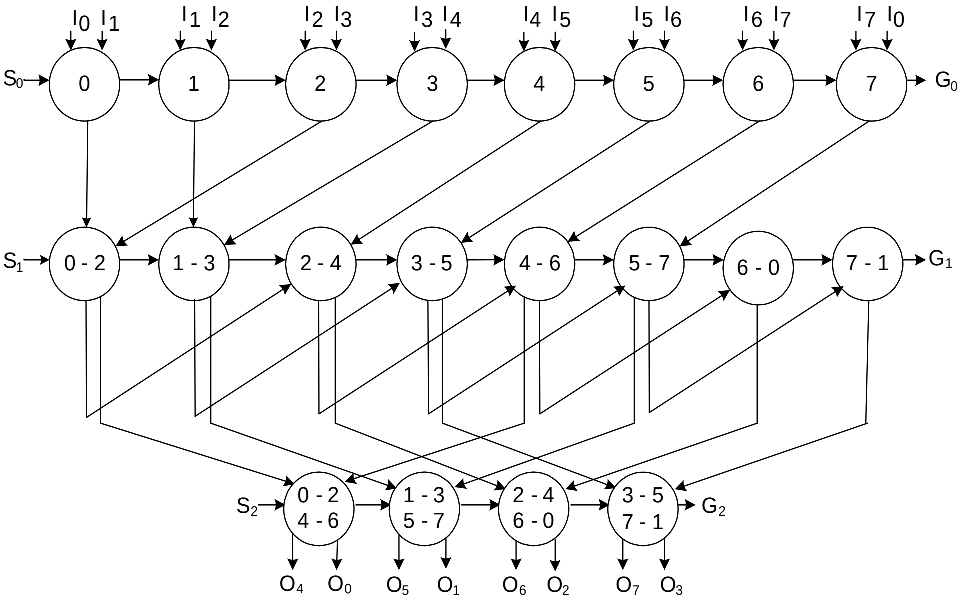

Figure 5.1: Adaptive structure of proposed (8, 3) unidirectional barrel shifter

Algorithm 1: Shift/rotate operation of proposed (n, k) unidirectional shifter

```
Input  : Data input set $I(I_0, I_1, ..., I_{n-1})$
         Control input set $S(S_0, S_1, ..., S_{k-1})$, where $k = log_2 n$
Output: Output data set $O(O_0, O_1, ..., O_{n-1})$
begin
    for $j \leftarrow 0$ to $k - 1$ do
        if $S_j = 1$ then
            $I \longleftarrow$ Shifted/Rotated $I$, $2^j$ $Times$
        end if
        else
            $I \longleftarrow I$
        end if
    end for
    return $O \longleftarrow I$
end
```

Idea behind the Algorithm 1 comes from the key property of unidirectional logarithmic shifter and the key property is, whether to shift or not. Whenever control bit S_j is set to high then the stage j ($j = 0$ *to* $k - 1$) will shift/rotate input 2^j times otherwise the input will remain same as previous stage, this instruction will execute $log_2 n$ times since there are total of $log_2 n$ stages.

Figures 5.2 and 5.3 shows proposed (4, 2) and (8, 3) Reversible Fault Tolerant Unidirectional Logarithmic Barrel Shifter respectively. Both the designs seem almost similar block representation of Figure 5.1, but the reversible fault tolerant Fredkin Gate (FRG) is used and especially some intelligent control of input output mappings are applied in the proposed design.

From Figures 5.2 and 5.3 we assess that, proposed (4, 2) and (8, 3) unidirectional shifter uses a total of 5 and 17 Fredkin Gates, produces 2 and 3 Garbage outputs and their Quantum cost are 25 and 85, respectively. Also Delay of the designs are 5 and 12 as there exists maximum of 5 and 12 gate between any input to any output respectively.

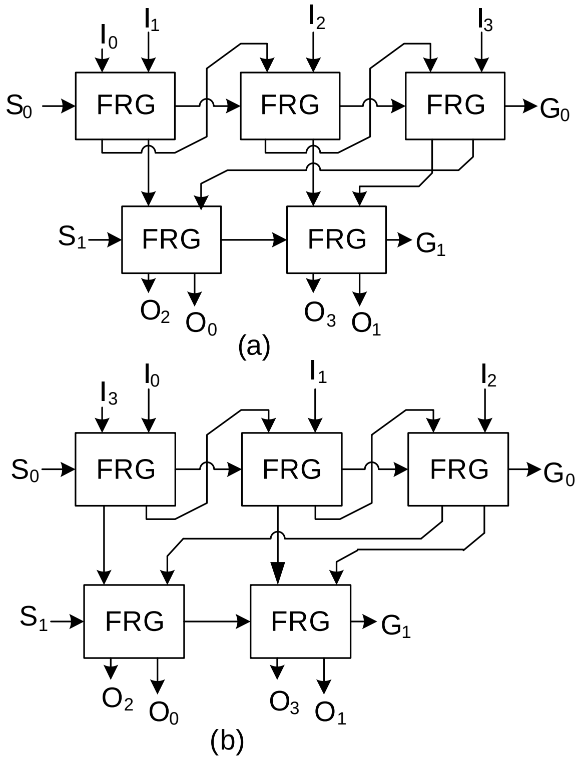

Figure 5.2: Proposed (4, 2) reversible fault tolerant unidirectional barrel shifter (a) Circuit for left rotation (b) Circuit for right rotation

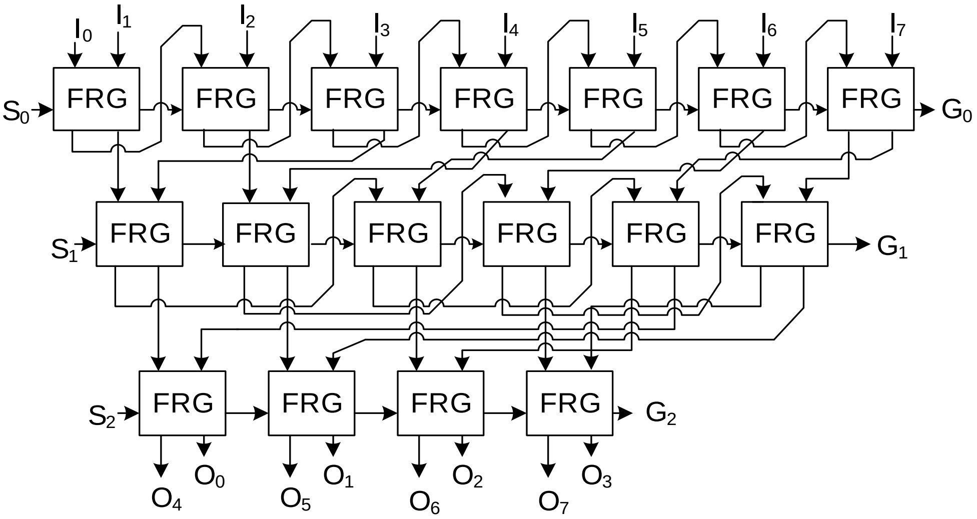

Figure 5.3: Proposed (8, 3) reversible fault tolerant unidirectional barrel shifter, circuit for left rotation

5.2 Working Procedure of Proposed Unidirectional Shifter

Working procedure of proposed unidirectional shifter is as follow: Each stage of Fredkin gate shifts the input according to the value of control input (S_j). Assume (4, 2) shifter takes $I_0I_1I_2I_3$ as data inputs and S_0 and S_1 as control inputs. If the control input $S_0 = 1$ and $S_1 = 1$ then data input will be rotated $2^0 + 2^1 = 3$ times to the left (assume circuit are for left rotation). Sequence of the rotate operation will be $I_1I_2I_3I_0$ for the first stage and $I_3I_0I_1I_2$ for the next, total of 3 bit rotation. On the other hand, for the control input $S_0 = 0$ and $S_1 = 0$, the input sequence will remain unchanged. Thus, each Fredkin gate chooses between two input lines it receives and performs the appropriate operation according to the select input of that particular stage. Working procedure of proposed (4, 2) unidirectional shifter is shown in Figure 5.4 (here circuit for left rotation are considered).

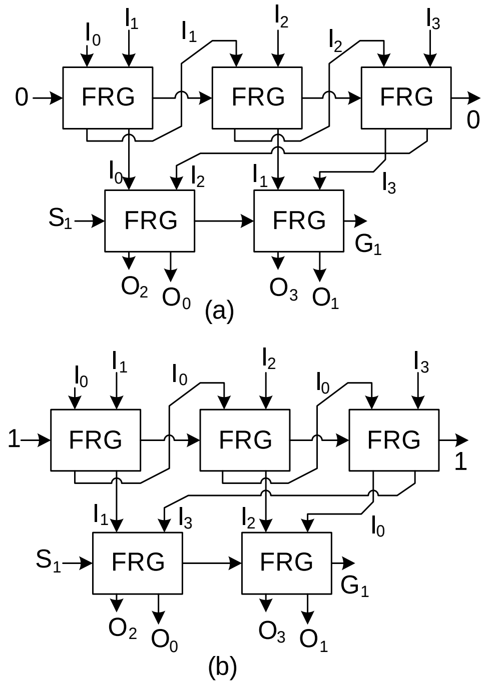

FIGURE 5.4: Working procedure of proposed (4, 2) unidirectional shifter (a) 1^{st} control input is set to low (b) 1^{st} control input is set to high

From Figure 5.4.(a) we access that, 1^{st} Fredkin gate of 1^{st} row will select input I_0, the 2^{nd} one will select I_1 and 3^{rd} one will select I_2 when 1^{st} control input (S_0) is set to low, also I_3 will generate from another output of 3^{rd} Fredkin gate. Finally selected four outputs will be used as inputs of 1^{st} and 2^{nd} Fredkin gates at 2^{nd} row which are controlled by 2^{nd} control input S_1. On the other hand if 1st control input is set to high then the selection sequence will be I_1, I_2, I_3 and I_0 by 1^{st}, 2^{nd} and 3^{rd} Fredkin gates at 1^{st} row respectively (shown in figure 5.4 (b)).

5.3 Evaluation of Proposed Unidirectional Design

From the above explanation and designing procedure several important properties of the proposed (n, k) reversible fault tolerant unidirectional logarithmic barrel shifter is demonstrated below.

Theorem 1: Let **FR** is the required number of the Fredkin gate for proposed (n, k) unidirectional shifter, then

$$\mathbf{FR} = n(k-1)+1$$

Proof: An (n, k) Shifter has k stages and total of n input bits. According to our design procedure gates required for each j^{th} stage is (n-2^j) except the last one, last stage always required $n/2$ number of gates. So total number of gate required are,

$$\begin{aligned} & (n-2^0)+(n-2^1)+(n-2^2)+\cdots+(n-2^{k-2})+n/2 \\ = \; & (n-2^0)+(n-2^1)+(n-2^2)+\cdots+(n-2^{k-2})+(n-n/2) \\ = \; & nk-(2^0+2^1+2^2+\cdots+2^{k-2}+2^{k-1}) \;\; [\because n=2^k] \end{aligned}$$

$$
\begin{aligned}
&= nk - \frac{2^0(2^k - 1)}{(2-1)} \\
&= n(k-1) + 1 \;\; [\because n = 2^k]
\end{aligned}
$$

So, total number of Fredkin gate required for proposed $(n,\ k)$ *unidirectional barrel shifter is* $n(k-1) + 1\Box$

Lemma 1: Let **QC** is the total quantum cost of proposed $(n,\ k)$reversible Fault tolerant unidirectional barrel shifter then,

$$\mathbf{QC} = 5n(k-1) + 5$$

Proof: In Theorem 1, we prove that proposed $(n,\ k)$ reversible fault tolerant unidirectional barrel shifter need total of $(n(k-1)+1)$ Fredkin gates. According to subsection 2.10.4 Quantum cost (**QC**) of each Fredkin gate is 5, hence total quantum cost is,

$$
\begin{aligned}
& 5 \times (n(k-1) + 1) \\
=\; & 5n(k-1) + 5
\end{aligned}
$$

Therefore total Quantum cost of proposed $(n,\ k)$*reversible fault tolerant unidirectional barrel shifter is* $5n(k-1) + 5\,\Box$

Theorem 2: Let **DL** is the total delay for proposed $(n,\ k)$ unidirectional shifter, then

$$\mathbf{DL} = \frac{3n}{2} + k - 3$$

Proof: The Delay of a logic circuit is the delay of the critical path. According to our design procedure, except last stage, critical path always consist $(2^{j+1} + 1)$ number of gate for each j^{th} stage (where j = 0 to $k-1$) and if j^{th} stage is the last one then the number of gate is in critical path for this stage is 2^j. So total delay of proposed $(n,\ k)$ reversible fault tolerant unidirectional design is,

$$\begin{aligned}
& (2^1+1)+(2^2+1)+(2^3+1)+\cdots+(2^{k-2}+1)+(2^{k-1}+1)+2^{k-1} \\
= \;& 1(k-1)+(2^1+2^2+2^3+\cdots+2^{k-2}+2^{k-1})+2^{k-1} \\
= \;& (k-1)+\frac{2^1(2^{k-1}-1)}{(2-1)}+2^{k-1} \\
= \;& (k-1)+2^k-2+2^{k-1} \\
= \;& (k-1)+n-2+\frac{n}{2}[\because n=2^k] \\
= \;& \frac{3n}{2}+k-3
\end{aligned}$$

Therefore total Delay of proposed $(n,\ k)$ reversible Fault tolerant unidirectional logarithmic barrel shifter is $3n/2+k-3$ □

Theorem 3: Let **GO** is the total number of the garbage output produce by proposed $(n,\ k)$ unidirectional shifter, then

$$\mathbf{GO}=k$$

Proof: Proposed $(n,\ k)$ shifter has total of k stages and n input bits. According to our design procedure each stage produce only one garbage output, since there are total of k stages, hence total number of the garbage output produced by proposed $(n,\ k)$ unidirectional shifter is,

$$\sum_{i=1}^{k} 1_i = k$$

Therefore total number of Garbage output produced by proposed $(n,\ k)$ shifter is k□

5.4 Performance Analysis of Proposed Unidirectional Design

In terms of different parameters comparison between existing shifter from [12] and the proposed one is shown in Tables 5.1, 5.2, 5.3 and 5.4 [7].

Table 5.1: Comparative study of reversible unidirectional barrel shifters in terms of required number of gates

(n, k)	Proposed Design	Existing Design[12]
(4, 2)	5	10
(8, 3)	17	36
(16, 4)	49	104
(32, 5)	129	272

Table 5.2: Comparative study of reversible unidirectional barrel shifters in terms of Delay of the circuit

(n, k)	Proposed Design	Existing Design[12]
(4, 2)	5	6
(8, 3)	12	14
(16, 4)	25	28
(32, 5)	50	52

Table 5.3: Comparative study of reversible unidirectional barrel shifters in terms of Quantum cost of the circuit

(n, k)	Proposed Design	Existing Design[12]
(4, 2)	25	34
(8, 3)	85	116
(16, 4)	245	328
(32, 5)	645	848

[7] Design from [12] already outperforms the work of [11]. So to compare with our proposed scheme we consider only the work of [12].

Table 5.4: Comparative study of reversible unidirectional barrel shifters in terms of total number of Garbage output

(n, k)	Proposed Design	Existing Design[12]
(4, 2)	2	6
(8, 3)	3	19
(16, 4)	4	52
(32, 5)	5	133

From above tables we can conclude that, proposed method outperforms the most efficient existing design of reversible barrel shifter proposed in [12] by all the parameters of reversible logic synthesis. All these comparison also proves that the proposed design method performs much expeditiously than the existing one and performance improves as the size of the circuit grows. Moreover proposed circuit design in Fault tolerant mode which provide deftness.

5.5 Summary

This chapter demonstrate design layout, designing algorithm, working procedure and the properties of the proposed Reversible Fault Tolerant Unidirectional Barrel Shifter. Comparative study is also given to compare the performance of the proposed design with existing design, which proves overall performance of the circuit is much improved in proposed design. Moreover proposed scheme is entirely Fault tolerant which is first ever proposed in the literature to the best of our knowledge.

Chapter 6

Proposed Reversible Fault Tolerant Bidirectional Barrel Shifter

Most interesting characteristic of proposed unidirectional shifter is that it can easily be converted into a bidirectional shifter using some extra Fredkin gate in addition with an extra control input which actually define the direction of shift-rotate. Designing and working procedure of Novel Reversible Fault Tolerant Bidirectional Barrel Shifter is being depicted with an illustration in the next few sections, followed by some theoretical explanation. Four graphical representation is given in the last part of this chapter from where we can access the performance of proposed schemes.

6.1 Design Procedure of Proposed Bidirectional Shifter

While designing reversible fault tolerant bidirectional barrel shifter, concentration is given on stage by stage output of proposed unidirectional shifter. Also massive engrossment is paid for calculating the difference between output of the left rotation and right rotation stage by stage i.e. for all combination of control sequence, what is the output of proposed unidirectional shifter at each stage if the circuit is for left rotation and what is the output of proposed unidirectional shifter at each stage if the circuit is for right rotation. However in conversion of unidirectional shifter to bidirectional shifter, we extend each stage of proposed unidirectional shifter to two except the last one and tell these stages when to perform rotation and when not. Combination of this two stages work as a single stage. As a result we have total of (k-1) stages each of which consist two stages and the last one. 1^{st} stage of these two stages perform left rotation and 2^{nd} one perform right rotation. If 1^{st} stage performs rotation, 2^{nd} one do nothing but passes the output to the next and vice versa. Algorithm 2 demonstrate this designing structure.

Procedure of Algorithm 2 is almost same as Algorithm 1 with the extension that, in Algorithm 2 shifting/rotation is depended not only on control input S_j but also on direction determiner control input S_{LR}, which actually define the direction of shift-rotate. Working process of Algorithm 2 is as follows: if direction determiner control signal is set to left (assume, for left and right rotation directional determiner control signal is set to low and high respectively i.e. $S_{LR} = 0$ for left rotation and $S_{LR} = 1$ for right rotation) and control bit S_j is set to high and then the stage j will shift the input 2^j times left. On the other hand if direction determiner control signal is set to right and control bit S_j is set to high and then the stage j will shift the input 2^j times right. Otherwise the input will remain same as previous stage. Alike Algorithm 1 this intrusions executes $log_2 n$ times.

Algorithm 2: Shift/rotate operation of proposed bidirectional shifter

```
input  : Data input set $I(I_0, I_1, ..., I_{n_1})$
         Control input set $S(S_0, S_1, ..., S_{k-1})$, where $k = log_2n$
         Directional determiner control signal $S_{LR}$
output: Output data set $O(O_0, O_1, ..., O_n - 1)$
begin
    for $j \leftarrow 0$ to $k - 1$ do
        if $S_j$=1 & $S_{LR}$ = 0 then
            $I \longleftarrow$ Shifted/Rotated $I$, $2^j$ Times Left
        end if
        else if $S_j$=1 & $S_{LR}$ = 1 then
            $I \longleftarrow$ Shifted/Rotated $I$, $2^j$ Times Right
        end if
        else
            $I \longleftarrow I$
        end if
    end for
    return $O \longleftarrow I$
end
```

Nevertheless according to our bidirectional design procedure entire circuit performs left rotation while direction determiner control signal (S_{LR}) is set as left (low in our given circuit below) otherwise right rotation, but both of this rotation also require stage's control signal (S_j) is set to hight. If stage's control signal (S_j) is set to low, then no rotation will take palace. Thus this design ensure's that the structure of logarithmic shifter remain untouched. We do not need any extra stage for last one in this conversion, as last stage rotate n bits data input by $n/2$ bits, if that stage's control input is set to high. According to our discussion on subsection 3.1.3 we experience that the left rotation or the right rotation of n bits data input by $n/2$ bits has same outcome. Hence we don't need that extra stage for last one in this conversion procedure.

Figures 6.1 and 6.2 shows proposed (4, 2) and (8, 3) reversible fault tolerant bidirectional logarithmic barrel shifter respectively.

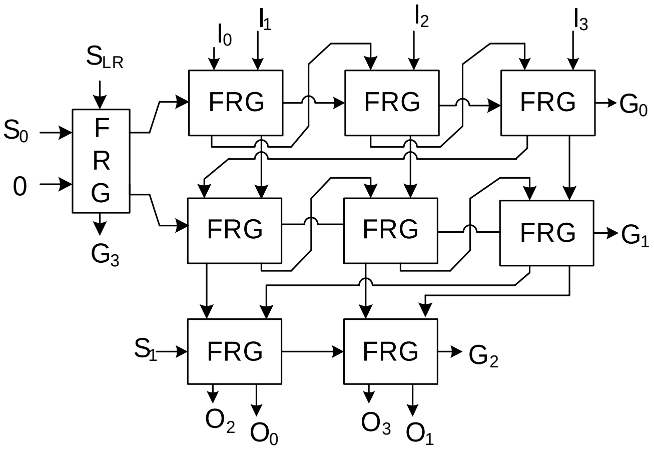

Figure 6.1: Proposed (4, 2) reversible fault tolerant bidirectional barrel shifter

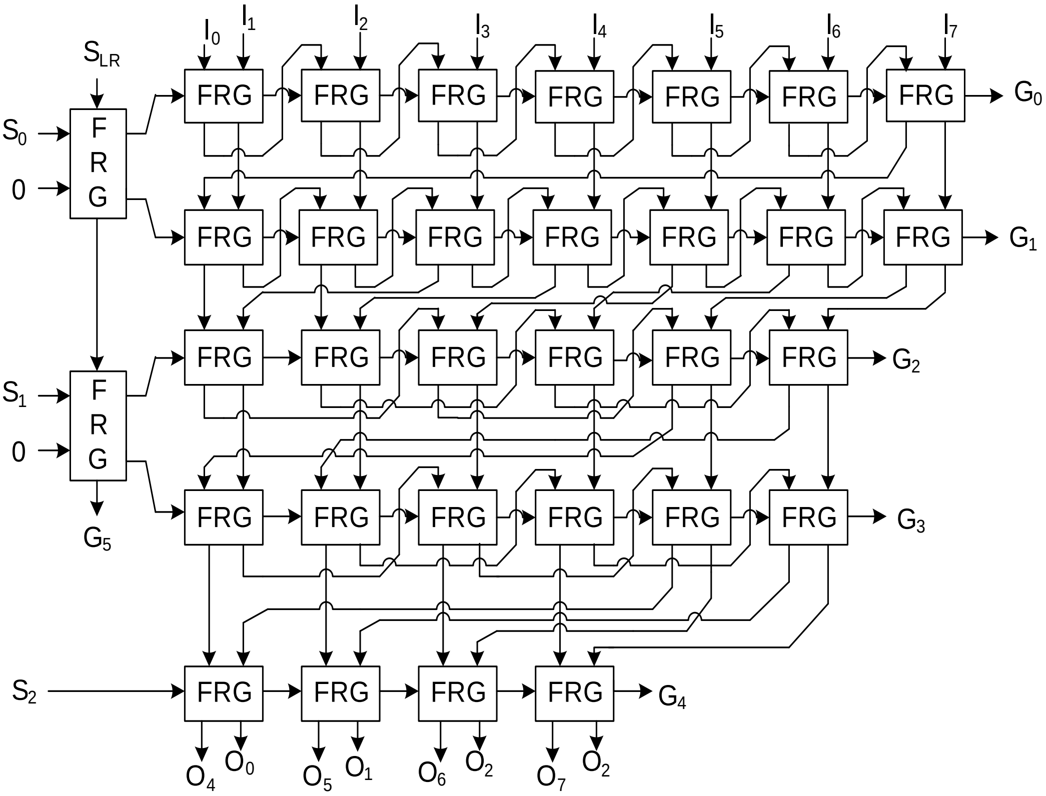

Figure 6.2: Proposed (8, 3) reversible fault tolerant bidirectional barrel shifter

6.2 Working Procedure of Proposed Bidirectional Shifter

Bidirectional design is the extension of the proposed unidirectional shifter, although intelligent control of input output mappings are applied in each stage. It's working procedure in shown in Figure 6.3.

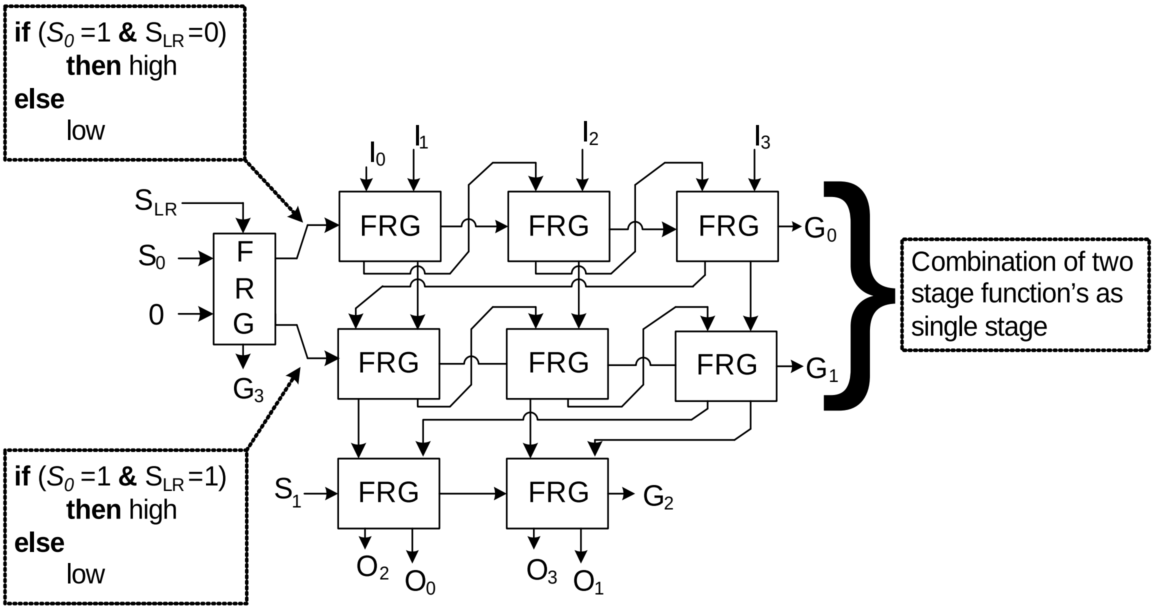

FIGURE 6.3: Working procedure of Proposed (4, 2) bidirectional shifter

From above figure we access that an extra Fredkin gate is needed to tell each stage (which is actually a combination of two stage among which 1^{st} one perform left rotation and 2^{nd} one perform right rotation) whether it should perform left rotation or right rotation. One of this two stages become active at a time. In other words, designing procedure is such that if one of this two stage perform rotation other one don't do anything other than passing output to the next stage. And both of them receives a low signal if S_j is low. So we can conclude that this design follows the adaptive nature of logarithmic shifter. Circuit of Figure 6.3 performs left rotation when $S_{LR} = 0$ and $S_j = 1$, and right rotation when $S_{LR} = 1$ and $S_j = 1$, in all other cases it does not perform any rotation.

6.3 Evaluation of Proposed Bidirectional Design

For n bits of data input there are total of k+1 control signal for bidirectional shifter. Comparing with the design structure of unidirectional shifter we access that, in bidirectional design there is one extra control signal. This extra control signal is actually determines the direction of shift-rotate. From the above explanation and designing procedure several important properties of the proposed (n, *k+1*) Reversible Fault Tolerant Bidirectional Barrel Shifter is demonstrated below.

Theorem 4: Let **F** is the required number of the Fredkin gate for proposed (n, *k+1*) bidirectional shifter, then

$$\mathbf{F} = \frac{1}{2}(4nk - 5n) + (k + 1)$$

Proof: Proposed (n, k+1) bidirectional shifter has (k-1) extra stage than proposed unidirectional shifter. In Theorem 1 we proved that we need (n-2^j) Fredkin gate for each j^{th} stage. This property remains unchanged for proposed bidirectional design as well. In bidirectional design we also need *(*k-1) extra Fredkin gate for this stages, to propagate the direction determiner control signal (S_{LR}). Also property of last stage remain unchanged i.e. last stage is always required $n/2$ number of Fredkin gate. In other words, there are total of 2(*k-1*) stages each of which required (n-2^j) number of gate (where $j\ =\ 0\,to\,k-1$), $n/2$ number of gate for last stage and (k-1) Fredkin gate for propagating direction determiner control signal. So total number of Fredkin gate required for proposed (n, k+1) reversible fault tolerant bidirectional barrel shifter is,

$$\begin{aligned} & 2\{(n-2^0)+(n-2^1)+(n-2^2)+\cdots+(n-2^{k-2})\}+\frac{n}{2}+(k-1) \\ = \; & \{(2n-2^1)+(2n-2^2)+(2n-2^3)+\cdots+(2n-2^{k-1})\}+\frac{n}{2}+(k-1) \\ = \; & 2n(k-1)-(2^1+2^2+2^3+\cdots+2^{k-1})+\frac{n}{2}+(k-1) \end{aligned}$$

$$
\begin{aligned}
&= (2n+1)(k-1) - \frac{2^1(2^{k-1}-1)}{(2-1)} + \frac{n}{2} \\
&= (2n+1)(k-1) - 2^k + 2 + \frac{n}{2} \\
&= (2n+1)(k-1) - n + 2 + \frac{n}{2}\ [\because 2^k\ = n] \\
&= 2nk - 2n + k - 1 - \frac{n}{2} + 2 \\
&= \frac{1}{2}(4nk-5n) + (k+1)
\end{aligned}
$$

Therefore total number of Fredkin gate required for proposed (n*, k+1*) *bidirectional shifter is* $1/2(4nk-5n)+(k+1)$ □

Lemma 2: Let **Q** is the total quantum cost of proposed (n, k+1) reversible fault tolerant bidirectional shifter, then

$$\mathbf{Q} = \frac{5}{2}(4nk-5n) + 5(k+1)$$

Proof: In Theorem 4, we prove that for proposed (n, $k+1$) bidirectional shifter total of $1/2(4nk-5n)+(k+1)$ Fredkin gates are needed. Quantum cost of each Fredkin gate is 5 according to subsection 2.10.4, hence total quantum cost of proposed bidirectional (n, k+1) reversible fault tolerant bidirectional shifter is,

$$
\begin{aligned}
& 5 \times \{\frac{1}{2}(4nk-5n) + (k+1)\} \\
&= \frac{5}{2}(4nk-5n) + 5(k+1)
\end{aligned}
$$

Therefore total Quantum cost of proposed (n*, k+1*) *reversible fault tolerant bidirectional barrel shifter is* $5/2(4nk-5n)+5(k+1)$ □

Theorem 5: Let **D** is the total delay for proposed (n, k+1) Reversible Fault Tolerant Bidirectional Barrel Shifter, then

$$\mathbf{D} = \frac{1}{2}(6k + 5n) - 7$$

Proof: In (n, *k+1*) bidirectional shifter there are total of k stages among which (*k-1*) of them are combination of two stages. Thus total number of stage is $2(k-1)$ plus the last stage. According to our design procedure of, except last stage, critical path always consist $(2^{j+1} + 1)$ number of gate for each j^{th} stage (where$j \ = \ 0\, to\, k-1$) and if j^{th} stage is the last one then the number of gate is in critical path for this stage is 2^j. Also we need (*k-1*) Fredkin gate for the purpose of propagating direction determiner control signal to these stages. Therefore total delay for an (n, *k+1*) Reversible Fault Tolerant Bidirectional Barrel Shifter is,

$$\begin{aligned}
& 2\{(2^1+1)+(2^2+1)+(2^3+1)+\cdots+(2^{k-2}+1)+(2^{k-1}+1)\}+2^{k-1}+(k-1) \\
= \ & \{(2^2+2)+(2^3+2)+(2^4+2)+\cdots+(2^{k-1}+2)+(2^k+2)\}+2^{k-1}+(k-1) \\
= \ & 2(k-1)+(2^2+2^3+2^4+\cdots+2^{k-1}+2^k)+2^{k-1}+(k-1) \\
= \ & 3(k-1)+\frac{2^2(2^{k-1}-1)}{(2-1)}+2^{k-1} \\
= \ & 3k-3+2^{k+1}-2^2+2^{k-1} \\
= \ & 3k-3+2n-4+\frac{n}{2} \ [\because 2^k \ = \ n] \\
= \ & 3k+\frac{5n}{2}-7 \\
= \ & \frac{1}{2}(6k+5n)-7
\end{aligned}$$

Thus total Delay of proposed (n, *k+1*) *reversible fault tolerant bidirectional barrel shifter is* $1/2(6k+5n)-7$□

Theorem 6: Let **G** is the total number of the garbage output produced by proposed $(n,\ k{+}1)$ bidirectional shifter, then

$$\mathbf{GO} = 2k$$

Proof: Proposed $(n,\ k{+}1)$ bidirectional shifter has total of k stages, (*k-1*) of them are combination of two stages. So total number of stage is $2k - 1$. According to our design procedure each stage produce only one garbage output. Also direction determiner control signal produces another one garbage output, thus total number of garbage output are,

$$\sum_{i=1}^{2k-1} 1_i + 1 = 2k - 1 + 1 = 2k$$

Thus total Garbage output produced by proposed (n, k+1) reversible fault tolerant bidirectional barrel shifter is $2k$ □

6.4 Performance Analysis of Proposed Bidirectional Design

Reversible bidirectional barrel shifter is first ever proposed in the literature which is also fault tolerant. Bidirectional property is unique in feature. One bidirectional design can be thought of two distinct unidirectional designs. This is the first time ever proposed reversible bidirectional barrel shifter in the literature which is also a Fault tolerant. A summarized study of proposed Reversible Fault Tolerant Bidirectional Barrel Shifter is given in table 6.1.

Table 6.1: Summarized study of bidirectional barrel shifter in terms required of number of gates, garbage outputs, delay and quantum cost

(n, k)	GT[8]	GO[9]	DE[10]	QC[11]
(4, 2)	9	4	6	45
(8, 3)	32	6	22	160
(16, 4)	93	8	45	465
(32, 5)	246	10	88	1230

Figures 6.4, 6.5, 6.6 and 6.7 depict the rate of change of required number of gate, Garbage output produced, Delay and Quantum cost of proposed bidirectional scheme.

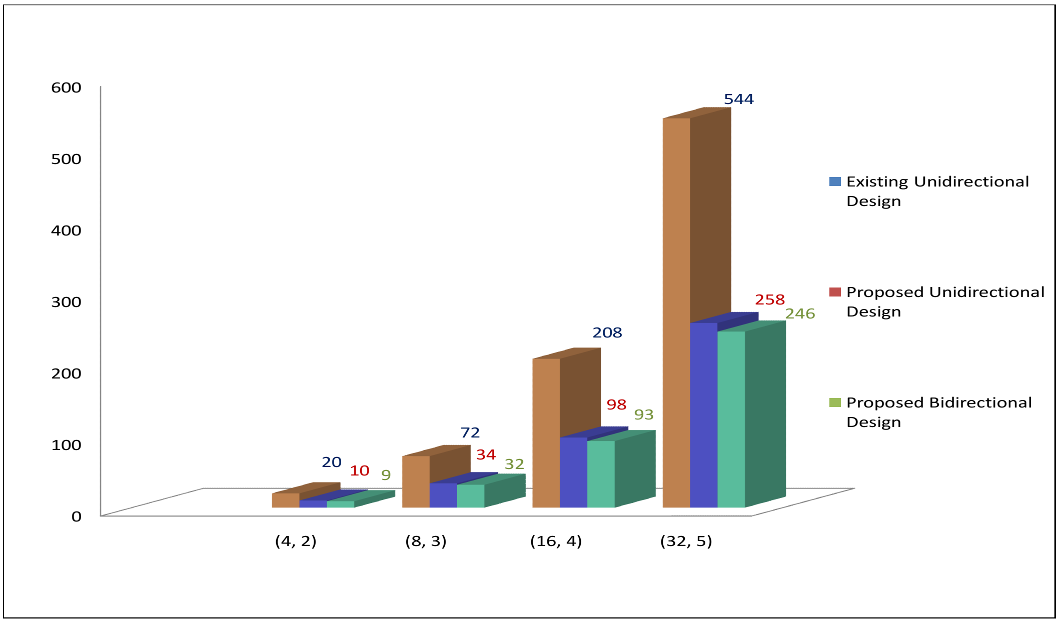

Figure 6.4: Comparative study of proposed designs and existing design [12] in-terms of required number of gate

[8] Required number of gates
[9] Number of Garbage output produced
[10] Delay of the circuit
[11] Total Quantum cost of the circuit

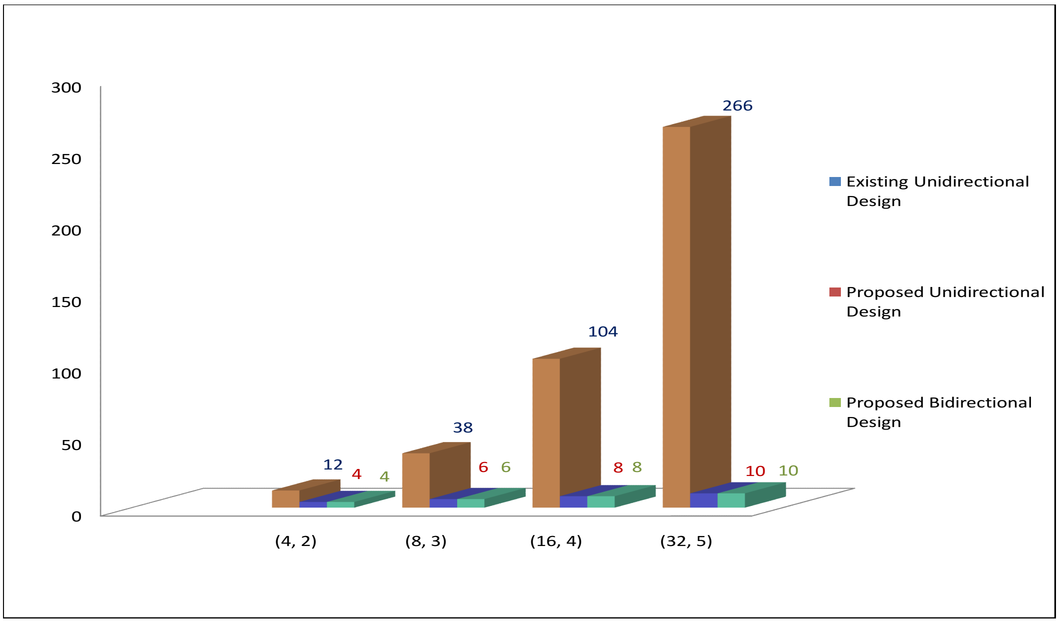

FIGURE 6.5: Comparative study of proposed designs and existing design [12] in-terms of garbage output produced

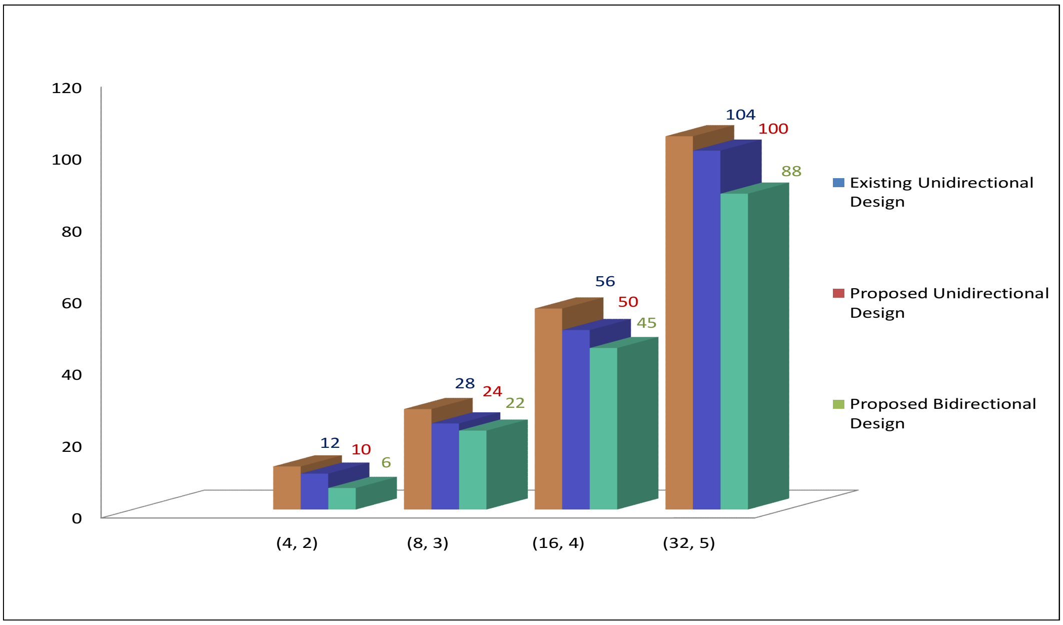

FIGURE 6.6: Comparative study of proposed designs and existing design [12] in-terms of delay of the circuit

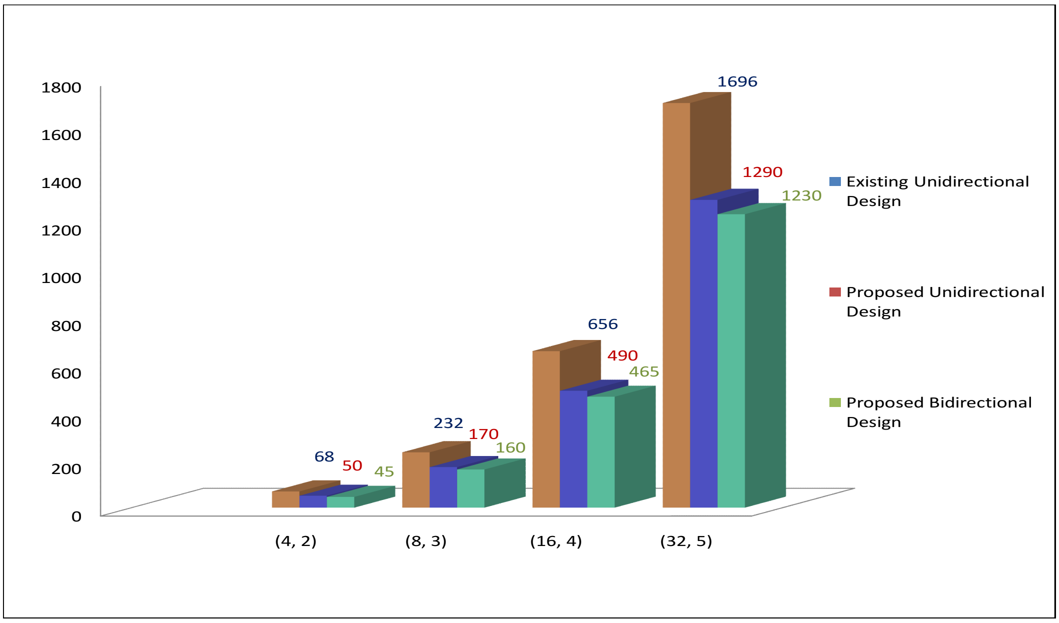

FIGURE 6.7: Comparative study of proposed designs and existing design [12] in-terms of Quantum cost of the circuit

In this above figures, to compare with our proposed bidirectional design we multiply each parameter of unidirectional designs by 2, since a bidirectional design can be consider as two separate unidirectional design. From the above figures we access the rate of change of required number gate, Garbage output produced, Delay and Quantum cost of proposed designs and existing design [12]. All this comparison proves that the proposed design method performs much efficiently than the existing one and performance improves as the size of the circuit grows.

6.5 Summary

In this chapter, designing procedure, working procedure and the properties of the proposed Reversible Fault Tolerant Bidirectional Barrel Shifter is being illustrated. Theoretical explanation and comparative study also given here for the purpose of accessing the performance of proposed bidirectional scheme.

Chapter 7

Overall Discussion

7.1 Conclusion

This thesis presents the design methodology of a novel logarithmic reversible fault tolerant barrel shifter, both in unidirectional and bidirectional mode. Only two significant work has been published so far in this regard but none of which were designed in a fault tolerant mode. The first phase of this research developed the new approach for designing the reversible fault tolerant unidirectional barrel shifter which was the main task to perform. The second phase of this research developed the automatic conversion procedure of proposed unidirectional design into bidirectional design. Third phase was to estimate the circuit performance in terms of total number of gates and garbage outputs. Then the research work includes calculating the Quantum cost and delay of the circuit. Proposed circuit surpasses the existing one in every significant terms of reversible circuit designing, also reversible design of bidirectional shifter is first ever proposed in the literature. Most importantly, proposed circuits are designed in a fault tolerant mode, and being a fault tolerant circuit, it outperforms all the existing no-fault tolerant circuit in all parameters.

7.2 Future work

At present the circuit is capable of performing bidirectional rotate. It can be enhanced towards multi-operation shift-rotate such as rotate operation, arithmetic and logical shift in a single cycle within a single circuit while preserving bi-directionality. In that case more control signal may be needed. The proposed designs of reversible circuits are a theoretical entity that is not yet implemented. The future plan of this study is to realize the proposed designs practically.

Bibliography

[1] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Research and Development*, vol. 5, pp. 183–191, 1961.

[2] R. Feynman, "Quantum mechanical computers," *Opt. News*, 1985.

[3] E. Fredkin and T. Toffoli, "Conservative logic," *Int'l J. Theoretical Physics*, vol. 21, pp. 219–253, 1982.

[4] C. H. Bennett, "Logical reversibility of computation," *IBM J. Research and Development*, vol. 17, pp. 525–532, 1973.

[5] M. A. Nielsen, I. Chuang, and L. K. Grover, "Quantum computation and quantum information," *American Journal of Physics*, vol. 70, no. 558, 2002.

[6] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Reversible logic circuit synthesis," *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design, ACM New York, USA*, pp. 353–360, 2002.

[7] M. Perkowski, "A hierarchical approach to computer-aided design of quantum circuits," *6th International Symposium on Representations and Methodology of Future Computing Technology*, pp. 201–209, 2003.

[8] D. G. Voyiatzis and A. Paschalis, "Accumulator-based test generation for robust sequential fault testing in dsp cores in near-optimal time," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 1079–1086, 2005.

[9] P. Metzgen, "Optimizing a high performance 32- bit processor for programmable logic," *Proceedings of International Symposium on System-on-Chip*, 2004.

[10] T. Conway, "Galois field arithmetic over gf(pm) for high-speed/low-power error-control applications," *IEEE Transactions on Circuits and Systems*, vol. 51, 2004.

[11] S. Gorgin and A. Kaivani, "Reversible barrel shifters computer system and applications," in *International Conference on Computer Systems and Applications AICCSA '07 IEEE/ACS*, pp. 479–483, 2007.

[12] I. Hashmi and H. M. H. Babu, "An efficient design of a reversible barrel shifter," in *23rd International Conference on VLSI Design Bangalore India*, pp. 93–98, 2010.

[13] J. C. Maxwell, *Theory of heat.* Longmans, Green and co, 2nd ed., 1902.

[14] L. Szilard, "On the decrease of entropy in a thermodynamic system by the intervention of intelligent beingsmaxwell's demon 2 entropy," *Classical and quantum information computing*, p. 110, 2003.

[15] H. Chau and F. Wilczek, "Simple realization of the fredkin gate using a series of two-body operators," *Physical review letters*, vol. 75, no. 4, pp. 748–750, 1995.

[16] G. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, pp. 82–85, April 1998.

[17] T. S. K. Rekha, S. Sasi and K. Jacob, "Fault tolerant error coding and detection using reversible gates," *IEEE TENCON 2007 Oct 30 - Nov 2 2007 Taiwan*, 2007.

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

[19] J. B. Altepeter, "A tale of two qubits: how quantum computers work." Available at, $http://arstechnica.com/science/guides/2010/01/a-tale-of-two-qubits-how-quantum-computers-work.ars$, (last visited Aug 10, 2010).

[20] A. Peres, "Reversible logic and quantum computers," *Phys. Rev. A*, vol. 32, pp. 3266–3276, Dec 1985.

[21] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *Society for Industrial and Applied Mathematics, SIAM*, vol. 41, no. 2, pp. 303–332, 1999.

[22] A. K. Biswas, A. R. Chowdhury, and H. M. Hasan Babu, "Efficient approaches for designing reversible binary coded decimal adders," *Microelectronics Journal*, vol. 39, pp. 1693–1703, 2008.

[23] H. M. Hasan Babu, M. R. Islam, A. R. Chowdhury, and S. M. A. Chowdhury, "Reversible logic synthesis for minimization of full-adder circuit," *IEEE Conference on Digital System Design 2003, Euro-Micro'03, Belek, Antalya, Turkey*, pp. 50–54, 2003.

[24] D. Maslov and G. W. Dueck, "Garbage in reversible designs of multiple output functions," *NSERC (CANADA).*

[25] A. Peres, "Reversible logic and quantum computers," *Physical Review*, vol. 32, no. 558, pp. 3266–3276, 1985.

[26] M. H. A. Khan and M. Perkowski, "Logic synthesis with cascades of new reversible gate families," *6th International Symposium on Representations*

and Methodology of Future Computing Technology (Reed-Muller), pp. 43–55, 2003.

[27] M. Haghparast and K. Navi, “A novel reversible bcd adder for nanotechnology based systems,” *American Journal of Applied Sciences*, vol. 5, no. 3, pp. 282–288, 2008.

[28] G. Yang, X. Song, W. N. N. Hung, and M. A. Perkowski, “Bi-direction synthesis for reversible circuits,” *IEEE Computer Society Annual Symposium on VLSI, 2005. Proceedings*, pp. 14–19, 2005.

[29] Hades, “Barrel-shifter 8 bit.” Abvialabe at, *http : //tams−www.informatik.uni − hamburg.de/applets/hades/webdemos/*10 − *gates/*60 − *barrel/shifter*8*.html*, (last visited Jan 26, 2010).

[30] Wikipedia, “Logical shift.” Abvialabe at, *http : //en.wikipedia.org wiki/Logical_shift*, (last visited May 3, 2010).

[31] Wikibooks, “Microprocessor design/shift and rotate blocks.” Abvialabe at, *http : //en.wikibooks.org/wiki/Microprocessor_Design/Shift_and_Rotate _Blocks*, (last visited May 3, 2010).

[32] Webster, “Art of assembly : Shifts and rotates.” Abvialabe at, *http : //webster.cs.ucr.edu/AoA/Windows/HTML/DataRepresentationa*6*.html*, (last visited May 3, 2010).

[33] M. R. Pillmeier, M. J. Schulte, and E. G. Walters, “Design alternatives for barrel shifters advanced signal processing, algorithms, architectures, and implementations XII,” *Proceedings of the SPIE*, vol. 4791.

[34] T. Thomson and H. Tam, “Barrel shifter,” *U.S. Patent 5,652,718*, July 1997.

[35] A. Yamaguchi, “Bidirectional shifter,” *U.S. Patent 5,262,971*, November 1993.

[36] K. Eshraghian and N. H. E. Weste, *Principles of CMOS VLSI design: a systems perspective.* Addison Wesley, 2nd ed., 1994.

[37] W. Wolf, *Modern VLSI design System-on-Chip Design.* Prentice Hall, 3rd ed., 2002.