Design Exploration and Application of Reversible Circuits in Emerging Technologies

Saurabh Kotiyal
University of South Florida, skotiyl@mail.usf.edu

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Scholar Commons Citation

This Thesis is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.
Design Exploration and Application of Reversible Circuits in Emerging Technologies

by

Saurabh Kotiyal

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Nagarajan Ranganathan, Ph.D.
Srinivas Katkoori, Ph.D.
Sanjukta Bhanja, Ph.D.
Kandethody Ramachandran, Ph.D.
Hao Zheng, Ph.D.

Date of Approval:
March 22, 2016

Keywords: Reversible logic, Optical Computing, Quantum cost, Ancilla inputs

Copyright © 2016, Saurabh Kotiyal
DEDICATION

To my parents, for all of their support and encouragement.
ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Nagarajan Ranganathan for providing me the opportunity for pursuing my research. As a mentor, Dr. Ranganathan has been instrumental in shaping my research career. I would always be thankful for his encouragement and constant support during the difficult times and showing confidence in me during those times. His valuable inputs and suggestions have a great influence on my research career, which resulted in transforming me into a better researcher. I would like to thank Dr. Himanshu Thapliyal for all of his valuable advices and suggestions throughout the years. I would also like to thank Dr. Srinivas Katkoori, Dr. Hao Zheng, Dr. Sanjukta Bhanja and Dr. Kandethody Ramachandran for taking the time to be in my doctoral committee and providing valuable suggestions to improve this manuscript. I am extremely grateful to the CSE department faculties, Dr. Rangachar Kasturi, Dr. Swaroop Ghosh, Dr. Dmitry Goldgof and Dr. Larry Hall, and CSE department staff for all their help and support.
# TABLE OF CONTENTS

LIST OF TABLES iv

LIST OF FIGURES vi

ABSTRACT x

CHAPTER 1 INTRODUCTION 1
   1.1 Motivation 3
   1.2 Contributions of Dissertation 7
   1.3 Outline of Dissertation 10

CHAPTER 2 BACKGROUND AND RELATED WORK 12
   2.1 Reversible Logic Gates 12
      2.1.1 Controlled-V and Controlled-$V^+$ Gate 12
      2.1.2 The NOT Gate 13
      2.1.3 Feynman Gate (CNOT Gate) 14
      2.1.4 Fredkin Gate 15
      2.1.5 Toffoli Gate 16
      2.1.6 Peres Gate 17
      2.2 Background on Mach-Zehnder Interferometer (MZI) Based Optical Reversible Computing 18
         2.2.1 All Optical Feynman Gate 19
         2.2.2 Mach-Zehnder Interferometer (MZI) Based 3x3 Toffoli Gate 20
         2.2.3 Mach-Zehnder Interferometer (MZI) Based 3x3 Peres Gate 21
   2.3 Background on Linear Optical Quantum Computing 23
      2.3.1 Linear Optical Quantum 3x3 Toffoli Gate 24

CHAPTER 3 MACH-ZEHNDER INTERFEROMETER BASED DESIGN OF ALL OPTICAL REVERSIBLE BINARY ADDER 27
   3.1 Proposed All Optical Reversible Gates 28
   3.2 Proposed All Optical Reversible Ripple Carry Adder with Input Carry 31
3.2.1 Steps of Proposed Methodology for Optical Reversible Adder 33
3.2.2 Theorem for Proposed Methodology of Optical Reversible Adder 35
3.2.3 Proof of Theorem for Proposed Methodology of Optical Reversible Adder 35
3.3 Delay and Optical Cost Analysis 36
  3.3.1 Comparison of n Bit Optical Reversible Ripple Carry Adders 37
3.4 Conclusion 38

CHAPTER 4 EFFICIENT REVERSIBLE NOR GATES AND THEIR MAPPING
IN OPTICAL COMPUTING DOMAIN 40
4.1 All Optical Reversible NAND Gates 41
  4.1.1 MZI Based Generalized Toffoli Gate (GTG) 41
  4.1.2 Generalized Linear Optical Quantum Computing Based Reversible Toffoli Gate 42
4.2 Proposed NOR Logic Based Reversible Gates and Their Mapping in Optical Domain 43
  4.2.1 Proposed Generalized Reversible NOR Gate for MZI Mapping 43
  4.2.2 Proposed Linear Optical Quantum Computing Based Reversible NOR Gate (LOQC-RNOR) 45
  4.2.3 Proposed Generalized Linear Optical Quantum Computing Based Reversible NOR Gate 49
4.3 Comparison of Proposed NOR Logic Based Reversible Gates with NAND Logic Based Reversible Gates Mapped in Optical Domain 51
  4.3.1 Comparison of 3-Input and 3-Output Gates 51
  4.3.2 Comparison of n-Input and n-Output Mach-Zehnder Interferometer Based Reversible Logic Gates 52
  4.3.3 Comparison of Proposed Reversible NOR Gate (LOQC-RNOR) with Existing Toffoli Gate Mapped in Linear Optical Quantum Computing Domain 54
4.4 Discussion and Conclusions 55

CHAPTER 5 DESIGN OF REVERSIBLE ADDER-SUBTRACTOR AND ITS MAPPING IN OPTICAL COMPUTING DOMAIN 57
5.1 Design of Unified Optical Reversible Adder-Subtractor 57
  5.1.1 Design of Optical Reversible Adder-Subtractor Based on Approach 1 58
  5.1.2 Design of Optical Reversible Adder-Subtractor Based on Approach 2 58
5.2 Comparison of n Bit Reversible Adder-Subtractor 59
5.3 Conclusion 63
CHAPTER 6 REVERSIBLE LOGIC BASED MULTIPLICATION COMPUTING UNIT USING BINARY TREE DATA STRUCTURE

6.1 Proposed Binary Tree-Based Design Methodology for NxN Reversible Multiplier 65
6.2 Design Example of a 4x4 Bit Reversible Multiplier Based on Proposed Methodology 70
6.3 Comparison of Proposed Reversible Multiplier Design with Existing Designs 74
   6.3.1 Comparison Study with Reversible Array Multipliers 76
6.4 Proposed Binary Tree-Based Design Methodology for Signed NxN Reversible Multiplier 78
6.5 Design Example of a Signed 5x5 Reversible Multiplier Based on Proposed Design Methodology 79
   6.5.1 Comparison of Proposed Signed 5x5 Reversible Multiplier Design with Existing Designs 86
6.6 Comparison of Proposed Signed Reversible Multiplier Design with Existing Designs 87
6.7 Conclusion 90

CHAPTER 7 LIMITATIONS AND CONCLUSION 91
REFERENCES 93
APPENDIX A: COPYRIGHT PERMISSIONS 102
ABOUT THE AUTHOR End Page
LIST OF TABLES

Table 1.1  Truth Table of Conventional and Reversible XOR Gates .......................... 2
Table 1.2  Ancilla Inputs and Garbage Outputs of a NxN Reversible Array Multiplier .......... 7
Table 2.1  Truth Table of Toffoli-Sign Gate ................................................................. 25
Table 2.2  Truth Table of Linear Optical Quantum 3x3 Toffoli Gate .............................. 26
Table 3.1  Truth Table of Optical Reversible Gate I (ORG-I) ........................................ 30
Table 3.2  Truth Table of Optical Reversible Gate II (ORG-II) ...................................... 30
Table 3.3  Optical Cost and Delay of All Optical Implementation of Reversible Gates ........ 37
Table 3.4  A Comparison of Reversible Ripple Carry Adder with Input Carry .................. 38
Table 3.5  Optical Cost Comparison of Reversible Ripple Carry Adders (with Input Carry) .. 38
Table 3.6  Delay (in Δ) Comparison of Reversible Ripple Carry Adders (with Input Carry) .. 39
Table 4.1  Truth Table of All Optical 3x3 MZI-RNOR Gate ......................................... 46
Table 4.2  Truth Table for Linear Optical Quantum Computing Based 3x3 Reversible NOR Gate (LOQC-RNOR) ................................................................. 49
Table 4.3  Optical Cost and Delay of 3x3 Reversible NAND Logic Gate and Proposed 3x3 MZI-RNOR Gate Mapped In MZI Switch ......................................................... 52
Table 4.4  Optical Cost and Delay Analysis of 3x3 Reversible Logic Gates by Implementing 13 Standard Boolean Functions ................................................................. 53
Table 4.5  Optical Cost and Delay Analysis of All Optical nxn Reversible NOR Logic Gates by Implementing 13 Standard Boolean Functions ......................................................... 54
Table 5.1 A Comparison of The Proposed Reversible Adder-Subtractor 60
Table 5.2 A Comparison of Reversible Adder-Subtractor Based on Approach 2 62
Table 5.3 Optical Cost Comparison of Reversible Adder-Subtractor Based on Approach 2 63
Table 5.4 Delay (in $\Delta$) Comparison of Reversible Adder-Subtractor Based on Approach 2 63
Table 6.1 A Comparison of 4x4 Reversible Multiplier 74
Table 6.2 A Comparison of NxN Reversible Multipliers 77
Table 6.3 A Comparison of Signed Reversible Multipliers 87
Table 6.4 A Comparison of Signed NxN Reversible Multiplier 89
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Conventional and Reversible XOR Gates</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Ancilla Inputs of Reversible Array Multiplier</td>
<td>7</td>
</tr>
<tr>
<td>1.3</td>
<td>Garbage Outputs of Reversible Array Multiplier</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>Controlled-V and Controlled-$V^+$ Gate</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>NOT Gate</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Feynman Gate (CNOT Gate), Its Quantum Implementation and Useful Properties</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>Fredkin Gate and Its Quantum Implementation</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>Toffoli Gate and Its Quantum Representation</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>Peres Gate and Its Quantum Implementation</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>Mach-Zehnder Interferometer (MZI) Based All Optical Switch</td>
<td>18</td>
</tr>
<tr>
<td>2.8</td>
<td>Feynman Gate and Its All Optical Implementation</td>
<td>20</td>
</tr>
<tr>
<td>2.9</td>
<td>3x3 Toffoli Gate and Its MZI Based Optical Implementation</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>(TG: Toffoli Gate, MZI: Mach-Zehnder Interferometer, BC: Beam Combiner, BS: Beam Splitter)</td>
<td></td>
</tr>
<tr>
<td>2.10</td>
<td>Peres Gate and Its All Optical Implementation</td>
<td>22</td>
</tr>
<tr>
<td>2.11</td>
<td>Qubit and Qutrit in Linear Optical Quantum Computing</td>
<td>23</td>
</tr>
<tr>
<td>2.12</td>
<td>Linear Optical Quantum Logic Gates and Unitary Matrix Representation</td>
<td>23</td>
</tr>
<tr>
<td>2.13</td>
<td>A Linear Optical Quantum Toffoli-Sign Gate</td>
<td>24</td>
</tr>
<tr>
<td>2.14</td>
<td>A Linear Optical Quantum Toffoli Gate</td>
<td>25</td>
</tr>
</tbody>
</table>
Figure 3.1 All Optical Implementations of Optical Reversible Gate I (ORG-I) and Optical Reversible Gate II (ORG-II) 29
Figure 3.2 Optical Reversible Gate I (ORG-I) and Optical Reversible Gate II (ORG-II) 29
Figure 3.3 Graphical Representation of Proposed Optical Reversible Gates and Design of An Optical Reversible Full Adder 31
Figure 3.4 Circuit Generation of Optical Reversible n Bit Adder Using Proposed Optical Reversible Gate I (ORG-I) and Optical Reversible Gate II (ORG-II) 32
Figure 3.5 Proposed Optical Reversible 4 Bit Adder 34
Figure 4.1 Generalized Toffoli Gate and Its MZI Based Optical Implementation (TG: Toffoli Gate, MZI: Mach-Zehnder Interferometer, BC: Beam Combiner, BS: Beam Splitter) 42
Figure 4.2 Generalized Linear Optical Quantum Computing Based Reversible Toffoli Gate 43
Figure 4.3 Generalized MZI-RNOR Gate and Its MZI Based Optical Implementation (MZI-RNORG: MZI-RNOR Gate, MZI: Mach-Zehnder Interferometer, BC: Beam Combiner, BS: Beam Splitter) 45
Figure 4.4 3x3 MZI-RNOR Gate and Its MZI Based Optical Implementation (MZI: Mach-Zehnder Interferometer, BC: Beam Combiner, BS: Beam Splitter) 47
Figure 4.5 Linear Optical Quantum Computing Based 3x3 Reversible NOR Gate (LOQC-RNOR) 48
Figure 4.6 A Linear Optical Quantum Computing, Based Reversible NOR Gate (LOQC-RNOR) 49
Figure 4.7 Transformation of The Proposed Linear Optical Quantum Computing Based Reversible NOR Gate (LOQC-RNOR) for Control Signal Values ($|C_2, C_1> = |0, 0>$)  * No Transformations are Performed at Stage (c) and (e), as Controlled-NOR Gate Has $C_1 = |0>$ as Control Value 50
Figure 4.8 Generalized Linear Optical Quantum Computing Based Reversible NOR Gate (LOQC-RNOR) 51
Figure 6.9  A Partial Product Generation Circuit for Signed N\times N Reversible Multiplier (G_0 to G_{2n-1}: Garbage Outputs) 81

Figure 6.10  Addition of Partial Products for a Signed 5\times 5 Multiplication (c10,c20,c30: Carry Outs) 82

Figure 6.11  A Binary Tree Representation of a Signed 5\times 5 Reversible Multiplier 83

Figure 6.12  Proposed Signed 5\times 5 Reversible Multiplier Design Using Reversible Ripple Adder (P_0 to P_{10}: Final Result; G_0 to G_{22}: Garbage Outputs) 84

Figure 6.13  A Partial Product Generation Circuit for Signed 5\times 5 Reversible Multiplier (G_0 to G_9: Garbage Outputs; p_0 to p_{24}: Partial Products) 84

Figure 6.14  A Signed 5\times 5 Reversible Multiplier Design Using Reversible Ripple Carry Adders with Input Carry (P_0 to P_{10}: Final Result; G_{10} to G_{32}: Garbage Outputs; p_0 to p_{24}: Partial Products) 85

Figure 6.15  Improvement of The Proposed Signed Reversible Multiplier Compared to Existing Signed Reversible Array Multiplier 89
ABSTRACT

The reversible logic has promising applications in emerging computing paradigms, such as quantum computing, quantum dot cellular automata, optical computing, etc. In reversible logic gates, there is a unique one-to-one mapping between the inputs and outputs. To generate a useful gate function, the reversible gates require some constant ancillary inputs called ancilla inputs. Also to maintain the reversibility of the circuits some additional unused outputs are required that are referred to as the garbage outputs. The number of ancilla inputs, the number of garbage outputs and quantum cost plays an important role in the evaluation of reversible circuits. Thus minimizing these parameters are important for designing an efficient reversible circuit. Reversible circuits are of highest interest in optical computing, quantum dot cellular automata and quantum computing. The quantum gates perform an elementary unitary operation on one, two or more two-state quantum systems called qubits. Any unitary operation is reversible in nature, and hence, quantum networks are also reversible, to conclude the quantum computers must be built from reversible logic components.

The main contribution of this dissertation is the design exploration and application of reversible circuits in emerging nanotechnologies. The emerging technologies explored in this work are 1) Optical quantum computing 2) Quantum computing.

The first contribution of this dissertation is Mach-Zehnder interferometer based design of all optical reversible binary adder. The all optical reversible adder design is based on two new optical reversible gates referred as optical reversible gate I (ORG-I) and optical reversible gate II (ORG-II) and the existing all optical Feynman gate. The two new reversible
gates ORG-I and ORG-II have been proposed and can implement a reversible adder with a reduced optical cost which is equal to the number of MZI switches required, less propagation delay, and with zero overhead in terms of number of ancilla inputs and the garbage outputs. The proposed all optical reversible adder design based on the ORG-I and ORG-II reversible gates are compared and shown to be better than the other existing designs of reversible adder proposed in the non-optical domain in terms of number of MZI switches, delay, the number of ancilla inputs and the garbage outputs. The proposed all optical reversible adder will be a key component of an all optical reversible arithmetic logical unit (ALU), that is a quite essential component in a wide variety of optical signal processing applications. In the existing literature, the NAND logic based implementation is the only known implementation available for reversible gates and its functions. There is a lack of research in the direction of NOR logic based implementation of reversible gates and functions. The second contribution of this dissertation is the design of NOR logic based n-input and n-output reversible gates, one of which can be efficiently mapped into optical computing using the Mach-Zehnder interferometer (MZI), while the other can be mapped efficiently in optical computing using the linear optical quantum gates. The proposed reversible NOR gates work as a corresponding NOR counterpart of NAND logic based Toffoli gates. The proposed optical reversible NOR logic gates can implement the reversible boolean logic functions with less number of linear optical quantum logic gates with reduced optical cost and propagation delay compared to the implementation using existing optical reversible NAND gates. It is illustrated that an optical reversible gate library having both optical Toffoli gate and the proposed optical reversible NOR gate is superior compared to the library containing only the optical Toffoli gate: (i) in terms of number of linear optical quantum gates when implemented using linear optical quantum computing (LOQC), (ii) in terms of optical cost and delay when implemented using the Mach-Zehnder interferometer. The third contribution of this dissertation is a binary tree-based design methodology for a NxN reversible multiplier. The proposed
binary tree-based design methodology for a NxN reversible multiplier performs the addition of partial products in parallel using the reversible ripple adders with zero ancilla bit and zero garbage bit; thereby, minimizing the number of ancilla and garbage bits used in the design. The proposed design methodology shows improvements in terms of number of ancilla inputs and garbage outputs compared to all the existing reversible multiplier designs. The methodology is also extended to the design of NxN reversible signed multiplier based on modified Baugh-Wooley multiplication methodology.
CHAPTER 1

INTRODUCTION

Reversible logic is a logic design style in which there is a one to one mapping between the input and the output vectors. According to [1], if a system is irreversible in nature then erasing a bit causes kTln2 joules of heat energy to be dissipated, where k is the Boltzmann’s constant and T is the absolute temperature of the environment. This kTln2 joule of heat energy won’t be dissipated, if a computation is performed reversibly based on reversible logic circuits [2]. The reversible logic has extensive applications in emerging technologies such as quantum computing, quantum dot cellular automata, optical computing, etc. [3, 4]. The major application of reversible logic lies in quantum computing. A quantum computer will be viewed as a quantum network (or a family of quantum networks) composed of quantum logic gates; each gate is performing an elementary unitary operation on one, two or more two-state quantum systems called qubits. Quantum networks must be built from reversible logical components [5].

The input and output vector of an N-input and N-output reversible logic gate or NxN reversible logic gate can be represented as:

\[
I_u = I_1, I_2, I_3, ..., I_N \quad (1.1)
\]

\[
O_u = O_1, O_2, O_3, ..., O_N \quad (1.2)
\]

Here \(I_u\) and \(O_u\) represents the input and output vectors of a reversible logic gate. The conventional CMOS logic gates are irreversible in nature as the input vectors can not be regenerated directly by the output vectors. Thus erasing a bit or loss of information causes
kTln2 joules of heat energy [1] to be dissipated. However in reversible logic gates, there exists a unique one to one mapping between the input and output vectors. An irreversible XOR gate can be represented as shown in Fig. 1.1(a), where A, B and P are the inputs and outputs respectively.

![Conventional XOR Gate](image1)

![Reversible XOR Gate](image2)

Figure 1.1. Conventional and Reversible XOR Gates

In the conventional XOR gate, the inputs A and B are mapped to the output P as $P = A \oplus B$. Whereas in the reversible logic, an XOR gate can be represented as shown in Fig. 1.1(b), where A, B and P, Q are the input and outputs respectively. Here the mapping between the inputs and outputs can be represented as $P = A$ and $Q = A \oplus B$ and holds the property of unique input and output vector mapping property of reversibility. The reconstruction of input vectors from the output vectors can be seen in table 1.1(b). The table 1.1(a) shows the input and output vectors for a conventional XOR gate. In the table 1.1(a) it can be seen that for the output $P = 0$ the input vectors are $AB = 01, 10$, whereas in the table 1.1(b) each output vector corresponds to a unique input vector.

<table>
<thead>
<tr>
<th>Table 1.1. Truth Table of Conventional and Reversible XOR Gates</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Conventional XOR Gate</td>
</tr>
<tr>
<td>$A$</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
1.1 Motivation

The emerging technologies such as Quantum computing, quantum dot cellular automata (QCA), optical quantum computing has promising applications for reversible logic [5, 6, 4, 7, 8]. The reversible logic satisfies the quantum computing property of unique one to one mapping between the input and output vectors. It has been proven by the researchers that, if a computation is performed in an irreversible manner then the loss of each bit of information produces kTln2 Joules of heat energy [1]. In order to avoid this limit, from a thermodynamics point of view, Bennett proved that kTln2 joules of heat energy would not be dissipated if the computation is carried in a reversible manner [2]. Thus, from thermodynamic considerations, a firm lower limit on dissipation of $E_{\text{diss}} = kT \ln 2 \approx 18 \text{meV}$ (in room-temperature environment) is a necessity for conventional (irreversible) logic, even if reliability issues could be ignored. If the physical implementation of the logic can be physically reversible, in that case, reversible logic can be useful for designing non-dissipative circuits. The current CMOS technology can be considered as a practical implementation platform for implementing reversible circuits as CMOS is not physically reversible. In CMOS, the energy of a signal can be characterized as $E_{\text{sig}} = (1/2) CV^2$, here energy is directly proportional the voltage change and is much higher than the kTln2 limit.

In contrary, there exists emerging nanotechnologies such as Optical Computing, Quantum Cellular Automata (QCA) and Superconducting Flux Logic (SFL) family, etc., where the energy dissipation due the loss of information will be a significant factor of overall heat dissipation of the system [9, 10, 11, 12, 13, 14, 15]. Thus, in this dissertation our primary motivation for adopting reversible logic lies in the fact that, it can provide a logic design terminology for creating ultra-low power circuits beyond the limit of kTln2, that are applicable for the emerging nanotechnologies in which the energy dissipated due to the loss of information will be significant factor of the overall heat dissipation. For an example, in
Superconductor Flux Logic (SFL) family based nSQUID gates, the energy dissipation in conventionally logically irreversible design is close to few kTln2 per logic operation. Here by employing reversible at 4 K temperature, the energy dissipation per nSQUID gate per bit is measured below the thermodynamics limit of kTln2 [11]. Therefore, the reversible logic has been extensively investigated for its promising application in power-efficient nanocomputing [16, 17, 18].

A photon can provide unmatched high speed and can store the information in a signal of zero mass. These properties of photon have attracted the attention of researchers to implement the reversible logic gates in all optical domain. The optical implementation of reversible logic gates could be useful to overcome the limits imposed by conventional computing, and is also considered as an implementation platform for quantum computing [19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. In the existing literature there are two types of optical mapping of reversible logic gates: (i) based on the semiconductor optical amplifier (SOA) using the MachZehnder interferometer (MZI) switch [29, 4, 30]; (ii) based on linear optical quantum computation (LOQC) using linear optical quantum logic gates [23, 24, 25, 26, 27, 28]. In recent years, researchers have implemented several reversible logic gates in optical computing domain such as Feynman gate, Toffoli gate, Peres gate, Modified Fredkin gate and reversible NOR gates [29, 4, 30]. The optical implementation of reversible logic gates can be achieved using SOA-based MZI optical switches. The MZI-based implementation of reversible logic gates provides significant advantages such as high speed, low power, fast switching time.

In the existing literature, the most widely used implementation of reversible logic gates and the reversible boolean functions are the implementations using NAND logic. This is due to the lack of research in the direction of NOR logic based reversible logic gates and functions. In this work, we propose two NOR logic based n-input and n-output reversible gates one of which can be efficiently mapped in optical computing using the MachZehnder inter-
ferometer (MZI) while the other one can be mapped efficiently in optical computing using the linear optical quantum gates. The first reversible NOR gate is called as a MachZehnder interferometer based reversible NOR gate (MZI-RNOR), and the second reversible NOR gate is called as a linear optical quantum computing based reversible NOR gate (LOQC-RNOR). The proposed optical reversible NOR gates are useful for NOR logic based implementation of reversible boolean functions. The proposed MZI-RNOR gate can implement the reversible boolean functions with reduced optical cost and propagation delay compared to the implementation of reversible boolean functions using optical reversible NAND gates (NAND logic based reversible gates is all optical Toffoli gate) implemented using the MZI switch. The proposed LOQC-RNOR can implement the reversible boolean functions with a reduced number of linear optical quantum logic gates compared to the implementation of reversible boolean functions implemented using linear optical quantum reversible NAND gates (NAND logic based reversible gates is linear optical quantum Toffoli gate). As the proposed optical reversible NOR gates are n-input and n-output optical reversible NOR-based counterpart of NAND logic based n-input and n-output Toffoli gate, thus we have also illustrated the optical design of the n-input and n-output Toffoli gate. In this work, the optical cost of a reversible logic gate is defined as the number of MZI switches used in its all optical implementation [31]. We illustrated the advantages of proposed optical reversible NOR gates in terms of optical cost and delay by implementing the 13 standard boolean functions [32]. The 13 standard boolean functions proposed in [23] can represent all possible 256 combinations of three variable boolean functions. It is illustrated that an optical reversible gate library having both optical Toffoli gate and the proposed optical reversible NOR gate is superior compared to the library containing only the optical Toffoli gate: (i) in terms of number of linear optical quantum gates when implemented using linear optical quantum computing (LOQC) and (ii) in terms of optical cost and delay when implemented using the MachZehnder interferometer.
Further in the field of quantum computing, researchers have proposed the design of reversible array multiplier [33]. The array multiplier approach is based on the following two steps: 1) generate the partial products using the reversible partial product generation circuitry 2) perform the addition of partial products generated at step one using the array of reversible full adders and reversible half adders. In this work, we are assuming that step one which is responsible for the generation of partial product using the reversible partial product generation circuitry will be the same for all multipliers. A NxN reversible partial product generation circuitry requires $N^2$ ancilla inputs and generates $2N$ garbage outputs. The details of step one will be discussed in section 6.1. In the reversible array multiplier, the step two which requires the addition of partial products would need an $N$ reversible half adders and $(N^2 - 2N)$ reversible full adders. In the existing literature, Peres gate can work as a reversible half adder with one ancilla input and one garbage output. The design of reversible full adder will need one ancilla input and two garbage outputs. Thus, to design an NxN reversible array multiplier (without considering the ancilla inputs and garbage outputs of reversible partial product generation circuitry), the number of ancilla inputs can be computed as $AN = (N + (N^2 - 2 \times N))$ and the number of garbage outputs can be computed as $GO = (N + 2(N^2 - 2 \times N))$. Table 1.2 shows the number of ancilla inputs and garbage outputs in NxN reversible multiplier for various values of N (including reversible partial product generation circuitry). Figures 1.2 and 1.3 show the plot of ancilla inputs and garbage outputs in NxN reversible multiplier for various values of N. From Figures 1.2 and 1.3, it is very evident that the number of ancilla inputs and garbage outputs in NxN reversible array-based multiplier increases exponentially with N. This sets our main objective to design a new reversible multiplier based on binary tree architecture that is primarily optimized in terms of the number of ancilla inputs and garbage outputs. The details of our proposed binary tree-based reversible multiplier will be discussed in section 6.1.
Table 1.2. Ancilla Inputs and Garbage Outputs of a NxN Reversible Array Multiplier

<table>
<thead>
<tr>
<th>NxN</th>
<th>Ancilla Inputs</th>
<th>Garbage Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>8x8</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>16x16</td>
<td>496</td>
<td>496</td>
</tr>
<tr>
<td>32x32</td>
<td>2016</td>
<td>2016</td>
</tr>
<tr>
<td>64x64</td>
<td>8128</td>
<td>8128</td>
</tr>
<tr>
<td>128x128</td>
<td>32640</td>
<td>32640</td>
</tr>
<tr>
<td>256x256</td>
<td>130816</td>
<td>130816</td>
</tr>
<tr>
<td>512x512</td>
<td>523776</td>
<td>523776</td>
</tr>
<tr>
<td>1024x1024</td>
<td>2096128</td>
<td>2096128</td>
</tr>
</tbody>
</table>

Figure 1.2. Ancilla Inputs of Reversible Array Multiplier

1.2 Contributions of Dissertation

The key parameters to design an efficient reversible logic design are as follows:

- Reducing the number of constant input bits, also referred as the ancilla inputs
- Minimizing the number of garbage outputs
- Minimizing the quantum delay of the reversible circuit
Minimizing the cost of the circuit, here the cost is measured in terms of the number of 1x1 and 2x2 quantum gates required to design a circuit. While keeping above mentioned parameters in consideration, we have proposed efficient reversible logic design methodologies targeting different emerging nanotechnologies. The dissertation presents the following contributions towards the design exploration and application of reversible circuits in emerging technologies.

- The first contribution of this dissertation is the all optical implementation of an n bit reversible ripple carry adder for the first time in literature. The all optical reversible adder design is based on two new optical reversible gates referred as optical reversible gate I (ORG-I) and optical reversible gate II (ORG-II) and the existing all optical Feynman gate. The two new reversible gates ORG-I and ORG-II are proposed as they can implement a reversible adder with a reduced optical cost which is the measure of number of MZIs switches and the propagation delay, and with zero overhead in terms of number of ancilla inputs and the garbage outputs. The proposed all optical reversible adder design based on the ORG-I and ORG-II reversible gates are compared...
and shown to be better than the other existing designs of reversible adder proposed in the non-optical domain in terms of number of MZIs, delay, the number of ancilla inputs and the garbage outputs. The proposed all optical reversible ripple carry adder will be a key component of an all optical reversible ALU that can be applied in a wide variety of optical signal processing applications.

- The second contribution is proposing two NOR logic based n-input and n-output reversible gates one of which can be efficiently mapped in optical computing using the Mach-Zehnder interferometer (MZI) while the other one can be mapped efficiently in optical computing using the linear optical quantum gates. The proposed reversible NOR gates work as a corresponding NOR counterpart of NAND logic based Toffoli gates. The proposed optical reversible NOR logic gates can implement the reversible boolean logic functions with reduced number of linear optical quantum logic gates or reduced optical cost and propagation delay compared to their implementation using existing optical reversible NAND gates. It is illustrated that an optical reversible gate library having both optical Toffoli gate and the proposed optical reversible NOR gate is superior compared to the library containing only the optical Toffoli gate: (i) in terms of number of linear optical quantum gates when implemented using linear optical quantum computing (LOQC), (ii) in terms of optical cost and delay when implemented using the Mach-Zehnder interferometer.

- The third contribution of this dissertation is presenting a binary tree-based design methodology for a NxN reversible multiplier. The proposed binary tree-based design methodology for NxN reversible multiplier performs the addition of partial products in parallel using the reversible ripple adders with zero ancilla bit and zero garbage bit; thereby minimizing the number of ancilla and garbage bits used in the design. The proposed design methodology shows a 17.86% to 60.34% improvement in terms of
ancilla inputs, and 21.43% to 52.17% in terms of garbage outputs compared to all the existing reversible multiplier designs. The methodology is also extended to the design of NxN reversible signed multiplier based on modified Baugh-Wooley multiplication methodology.

- The other important contribution of this dissertation is proposing two design methodologies of reversible adder-subtractor to optimize the parameters of ancilla input bits, the number of garbage outputs, optical cost, and the delay. The proposed efficient design of optical reversible adder-subtractor will find promising applications in optical reversible computing and could form a key component of optical reversible digital processing circuits and architectures.

1.3 Outline of Dissertation

The organization of the dissertation is as follows: Chapter 1 provides the introduction to the proposed work. It also provides a brief review of quantum computing and reversible computing. The Chapter 2 provides the background and related work towards this proposed dissertation work. In this chapter, a brief review of reversible logic gates, Mach-Zehnder Interferometer (MZI) and Linear Optical Quantum Computing (LOQC) have been provided. In Chapter 3, the details on proposed Mech-Zehnder Interferometer based design of all optical reversible binary adder have been provided. A detailed delay and optical cost based analysis is also provided for proposed Mech-Zehnder Interferometer based design of all optical reversible binary adder. Chapter 4 presents the design of efficient reversible NOR gates and their mapping in optical computing domain. A comparison of proposed NOR logic based reversible gates with NAND logic based reversible gates is also provided in this chapter. The Chapter 5 provides details on the design of reversible adder-subtractor and its mapping
in optical computing domain. In Chapter 6, we have proposed the reversible logic based multiplication computing unit using binary tree data structure.
CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Reversible Logic Gates

In the existing literature, there exists several reversible gates such as the Feynman gate, Toffoli gate, and the Fredkin gate. The quantum cost of a reversible gate is the number of 1x1 and 2x2 reversible gates required to design a 3x3 reversible gate. The quantum cost of all 1x1 and 2x2 reversible gates are considered as unity [34, 35, 36]. The 3x3 reversible gates are implemented using the 1x1 NOT gate and 2x2 reversible gates such as Controlled-V and Controlled-V+ (V is a square-root of the NOT gate, and V+ is its hermitian), and the Feynman gate is also known as Controlled NOT gate (CNOT).

2.1.1 Controlled-V and Controlled-$V^+$ Gate

A controlled-V gate is shown in Fig. 2.1(a). In a controlled-V gate, when the value of control signal $A=0$, then the qubit B will pass through the controlled part unchanged, i.e., we will have $Q=B$. When the value of control signal $A=1$, then the unitary operation $V = \frac{i+1}{2} \left( \frac{1}{i} - i \right)$ is applied to input B, i.e., $Q=V(B)$. The controlled-$V^+$ gate is shown in Fig. 2.1(b). In the controlled-$V^+$ gate when the value of control signal $A=0$, then the qubit B will pass through the controlled part unchanged, i.e., we will have $Q=B$. When the value of control signal $A=1$, then the unitary operation $V^+ = V^{-1}$ is applied to the input B, i.e., $Q=V^+(B)$. 

12
The V and V+ quantum gates have the following properties:

\[ V \times V = NOT \]
\[ V \times V^+ = V^+ \times V = I \]
\[ V^+ \times V^+ = NOT \]

The property, as shown above, represents that, when two V gates are in series they will behave as an NOT gate. Similarly two V+ gate in series behaves as an NOT gate. A V gate in series with a V+ gate and vice versa forms an identity. The more details on V and V+ gate and its properties can be found in [5, 35].

2.1.2 The NOT Gate

An NOT gate is 1x1 gate represented as shown in Fig. 2.2. Since it is a 1x1 gate, its quantum cost is unity.
2.1.3 Feynman Gate (CNOT Gate)

The Feynman gate (FG) also known as the controlled-NOT gate (CNOT) is a 2-inputs and 2-outputs reversible gate with the mapping \((A, B)\) to \((P=A, Q=A \oplus B)\). Here \(A\) is the controlling input and \(B\) is the controlled input; \(P\) and \(Q\) are the two outputs of the gate. Since the Feynman gate is a 2x2 reversible gate, it has a quantum cost of 1. Figure 2.3(a) and 2.3(b) shows the block representation and equivalent quantum representation of the Feynman gate. The fanout is not allowed in a reversible logic-based design. Feynman gate is very helpful in this regard as it can be used for copying the signal thus avoiding the fanout issue in reversible logic-based design as shown in Fig. 2.3(c). The Feynman gate can also be used for generating the complement of a given input signal as shown in Fig. 2.3(d).

Figure 2.3. Feynman Gate (CNOT Gate), Its Quantum Implementation and Useful Properties
2.1.4 Fredkin Gate

The Fredkin gate is a 3x3 reversible logic gate, with three inputs and three outputs. Figure 2.4(a) shows the block representation of a Fredkin gate. The Fredkin gate maps inputs (A, B, C) to outputs (P=A, Q = \bar{A}B + AC, R = AB + \bar{A}C), where A, B, C are the inputs and P, Q, R are the outputs, respectively [37].

![Fredkin Gate Diagram](image)

A Fredkin gate can work as 2:1 MUX, as it can swap its other two inputs (B, C) depending on the value of its first input (A). Referring to the Fig. 2.4(a), the first input A works as a controlling input while the inputs B and C work as controlled inputs. Thus, when A=1 the inputs B and C will be swapped and results in the value of outputs as Q=C and R=B. For the value of controlling input A=0, the outputs P and Q will be directly connected to inputs A and B. Figure 2.4(b) shows the quantum implementation of a Fredkin gate with a quantum cost of 5 [35]. In Fig. 2.4(b) each dotted rectangle is equivalent to a 2x2 Feynman gate, and
the quantum cost of each dotted rectangle is considered as 1 [34]. The same assumption is used for calculating the quantum cost of the Fredkin gate in [35]. Thus, the quantum cost of a Fredkin gate is 5 as it consists of 2 dotted rectangles, 1 Controlled-V gate, and 2 CNOT gates. In this work, we have also followed the assumption by [34], and in our quantum cost calculations, the quantum cost of the Fredkin gate is considered as 5.

2.1.5 Toffoli Gate

The Toffoli gate (TG) is a 3X3 reversible logic gate with three inputs and three outputs. The inputs to outputs mapping of a Toffoli gate can be represented as $(A, B, C)$ to $(P = A, Q = B, R = A.B \oplus C)$, where $A, B, C$ are the inputs and $P, Q, R$ are the outputs of a Toffoli gate. Figure 2.5(a) shows the block diagram of a Toffoli gate.
A Toffoli gate has a quantum cost of 5, as it can be implemented using 2 V gates, 1 \( V^+ \) gate and 2 CNOT gates [38]. Figure 2.5(b) shows the quantum implementation of a Toffoli gate using V gates, \( V^+ \) gate and CNOT gates.

### 2.1.6 Peres Gate

The Peres gate is a 3X3 reversible gate with a mapping between the inputs A, B, C and the outputs P, Q, R as \((A, B, C) \rightarrow (P = A, Q = A \oplus b, R = A.B \oplus C)\) [39]. A block representation of the Peres gate is shown in Fig. 2.6(a) and Fig. 2.6(b) shows the quantum representation of the Peres gate. The quantum implementation of Peres gate requires 2 \( V^+ \) gates, 1 V gate and 1 CNOT gate. Thus, the quantum cost of a Peres gate can be calculated as 4.

![Peres Gate and Its Quantum Implementation](image)

Figure 2.6. Peres Gate and Its Quantum Implementation
2.2 Background on Mach-Zehnder Interferometer (MZI) Based Optical Reversible Computing

In recent years, the Mach-Zehnder interferometer (MZI) based optical switch has drawn the interest of many researchers in the field of all optical reversible logic [4, 29, 30, 40]. A design of all optical MZI switch is shown in Fig. 2.7(a). An MZI based all optical switch can be designed using 2 Semiconductor optical amplifier (SOA-1, SOA-2) and two couplers (C-1, C-2). The operating principle of MZI based all optical switch can be explained as follows:

In Mach-Zehnder Interferometer (MZI), there are two input ports A and B and two output ports called as bar port and cross port, respectively as shown in Fig.2.7(a). At the input ports, the optical signal coming from port B is considered as the control signal (\(\lambda_2\)) and the optical signal coming from port A is considered as the incoming signal (\(\lambda_1\)).

![Diagram](image)

(a) Semiconductor Optical Amplifier Based Mach-Zehnder Interferometer (MZI)

![Diagram](image)

(b) Mach-Zehnder Interferometer

Figure 2.7. Mach-Zehnder Interferometer (MZI) Based All Optical Switch
The working of the MZI can be explained as follows: (i) When there is an incoming signal at port A and the control signal at port B then there is a light present at the output bar port and there is no light present at the output cross port (ii) In the absence of control signal at input port B and incoming signal at input port A, then the outputs of MZI are switched and results in the presence of light at the output cross port and no light at the bar port. In our work, we have considered no light or absence of light as the binary value 0. The above behavior of MZI based all optical switch can be represented as boolean functions having inputs to outputs mapping as \((A, B)\) to \((P=AB, Q = A\overline{B})\), where \(A\) (incoming signal), \(B\) (control signal) are the inputs of Mach-Zehnder Interferometer (MZI) and \(P\) (Bar Port), \(Q\) (Cross Port) are the outputs of Mach-Zehnder Interferometer (MZI), respectively. A block representation of MZI based all optical switch is shown in Fig. 2.7(b). In our work the optical cost and the delay \((\Delta)\) of MZI based all optical switch is considered as a unity.

### 2.2.1 All Optical Feynman Gate

The Feynman gate (FG) is a 2 inputs and 2 outputs reversible gate. It has the mapping \((A, B)\) to \((P=A, Q= A\oplus B)\) where \(A, B\) are the inputs and \(P, Q\) are the outputs, respectively. The Feynman is also referred as the Controlled-Not gate (CNOT), as for the value of controlled input \(A=1\) the output generated at \(Q\) will be the complement of input \(B\) that is \(Q=\overline{B}\). A Feynman gate can be implemented using 2 MZI based all optical switches, 2 beam combiners (BC) and 2 beam splitters (BS) in all optical reversible computing \([4]\). The beam combiner (BC) simply combines the optical beams, while the beam splitter simply splits the optical beams into two optical beams. Hence, researchers do not consider them in the optical cost and the delay calculation in all optical quantum computing domain \([41, 42]\). Figure 2.8(a) and 2.8(b) shows the block representation and the all optical implementation of the Feynman gate. From the Fig. 2.8, it can be seen that the Feynman gate can be implemented using 2 MZI based optical switches. Hence, the optical cost of Feynman gate is
considered as 2. In the all optical implementation of the Feynman gate, two MZIs switches works in parallel thus the delay of the optical Feynman gate is considered as $1 \Delta$.

![Figure 2.8. Feynman Gate and Its All Optical Implementation](image)

### 2.2.2 Mach-Zehnder Interferometer (MZI) Based 3x3 Toffoli Gate

Toffoli gate is the most popular universal reversible gate, as it can function as an NAND gate. Consider a 3-input and 3-output Toffoli gate referred as TG in Fig. 2.9(a). The inputs to outputs mapping of a 3x3 TG is $(A, B, C)$ to $(P = A, Q = B, R = A.B \oplus C)$, where $A$, $B$, $C$ are the inputs and $P$, $Q$, $R$ are the outputs, respectively [4]. When $C=1$, the outputs of the Toffoli gate transform as $P = A$, $Q = B$, $R = A.B \oplus 1 = \overline{A.B}$.

Figure 2.9(b) represents the Toffoli gate working as an NAND gate when the value of input signal $C$ is set to one. An all optical 3x3 Toffoli gate can be implemented using 3 MZI based all optical switches, 1 beam combiner (BC) and 4 beam splitters [4]. Figure 2.9(a) and Fig. 2.9(c) shows the block representation and all optical implementation of Toffoli gate, respectively. The optical cost of the Toffoli gate is considered as 3, as the Toffoli gate can...
be implemented using 3 MZI based all optical switches. The Toffoli gate has a delay of 2∆, as two MZI switches out of three MZI switches works in parallel. *Following the above idea of 3x3 Toffoli gate, it can be easily followed that a generalized n-inputs and n-outputs Toffoli gate can work as an n-1 input NAND gate by hard wiring the value of last input as 1.*

### 2.2.3 Mach-Zehnder Interferometer (MZI) Based 3x3 Peres Gate

The Peres gate is a 3x3 reversible logic gate with the inputs to outputs mapping as \((A, B, C)\) to \((P = A, Q = A \oplus B, R = A.B \oplus C)\), where A, B, C are the inputs and P, Q, R
are the outputs respectively [30]. An all optical Peres gate can be implemented using 4 MZI based switches, 5 beam splitters (BS) and 3 beam combiners (BC). Figure 2.10(a) and Fig. 2.10(b) shows the block representation and the all optical implementation of a Peres gate, respectively. The optical cost of Peres gate is 4, as the all optical implementation of Peres gate requires 4 MZI based switches. The delay of a Peres gate is $2\Delta$, as two MZI switches work in parallel with the two other MZI switches.

Figure 2.10. Peres Gate and Its All Optical Implementation (PG: Peres Gate, MZI: Mach-Zehnder Interferometer, BC: Beam Combiner, BS: Beam Splitter)
2.3 Background on Linear Optical Quantum Computing

The linear optical quantum computing uses the photons to encode the information. The information stored in the linear optical quantum computer are in the form of qubits and qutrits. A qubit has two possible logical states referred as $|0>$ and $|1>$, while a qutrit has three logical states represented as $|0>$, $|1>$, $|2>$. The representation of a qubit and qutrit in linear optical quantum computing is shown in Fig. 2.11.

$$|0\rangle = |0,1\rangle \Rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |1\rangle = |1,0\rangle \Rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

(a) Qubit and matrix representation

Qutrit states $\Rightarrow$ $|0\rangle, |1\rangle, |2\rangle \Rightarrow \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$

(b) Qutrit and matrix representation

Figure 2.11. Qubit and Qutrit in Linear Optical Quantum Computing

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(a) Controlled-Z Gate  (b) Controlled-NOT Gate

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

(c) Hadamard Gate  (d) NOT Gate ($X$)  (e) $X_a$ Gate

Figure 2.12. Linear Optical Quantum Logic Gates and Unitary Matrix Representation
The basic linear optical quantum logic gates that perform the logical operation on the qubits and qutrits are NOT gate, Hadamard gate, \( X_a \) gate, Controlled-NOT (CNOT) gate and Controlled-Z gate. Figure 2.12 shows the linear optical quantum logic gates and their respective unitary matrix representation.

2.3.1 Linear Optical Quantum 3x3 Toffoli Gate

In the existing literature, the researchers have proposed the implementation of 3x3 Toffoli gate in linear optics quantum computing [26, 27, 28]. The existing design of a linear optical quantum 3x3 Toffoli gate is based on the design of a linear optical quantum Toffoli-sign gate. The linear optical quantum Toffoli-sign gate is a three-qubit quantum gate that imposes a sign shift to one of the logical states based on the values of two control signals or qubits [26, 27, 28]. An implementation of a linear optical quantum Toffoli-sign gate using the three two-qubit gates is shown in Fig. 2.13.

![Figure 2.13. A Linear Optical Quantum Toffoli-Sign Gate](image)

The linear optical quantum implementation of the Toffoli-sign gate has the target bit as a qutrit with the logical states as \(|0>\), |1>, |2>\). The linear optical quantum logic gate represented as \( X_a \) in Fig.2.13 works on a qutrit and performs the swap operation between the logic state \(-0\rangle\) and \(-2\rangle\), this gate leaves the logic state \(-1\rangle\) unchanged. The unitary matrix of the quantum logic gate \( X_a \) is shown in Fig. 2.12. The unitary matrix of the Hadamard
gate and the controlled-Z gate used in the linear optical quantum implementation of Toffoli-sign gate is also shown in Fig. 2.12. The truth Table of the inputs to outputs mapping of a Toffoli-sign gate is shown in Table 2.1. From Table 2.1 it can be seen that the Toffoli-sign gate performs a sign shift only for the input state $|C_2, C_1, T \rangle = |1, 0, 1 \rangle$.

Table 2.1. Truth Table of Toffoli-Sign Gate

<table>
<thead>
<tr>
<th>No.</th>
<th>Control($C_2, C_1$)</th>
<th>Target($T$)</th>
<th>Toffoli-Sign Gate ($C_2, C_1, T$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$</td>
<td>0, 0 \rangle$</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>$</td>
<td>0, 0 \rangle$</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>$</td>
<td>0, 1 \rangle$</td>
<td>$</td>
</tr>
<tr>
<td>4</td>
<td>$</td>
<td>0, 1 \rangle$</td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>$</td>
<td>1, 0 \rangle$</td>
<td>$</td>
</tr>
<tr>
<td>6</td>
<td>$</td>
<td>1, 0 \rangle$</td>
<td>$</td>
</tr>
<tr>
<td>7</td>
<td>$</td>
<td>1, 1 \rangle$</td>
<td>$</td>
</tr>
<tr>
<td>8</td>
<td>$</td>
<td>1, 1 \rangle$</td>
<td>$</td>
</tr>
</tbody>
</table>

A linear optical quantum Toffoli gate can be implemented using the Toffoli-sign gate by applying two additional Hadamard gates before and after the target qutrit and by applying additional two NOT gate ($X$ gates) before and after second control qubit ($C_1$) as shown in Fig. 2.14 [26, 27, 28].

![Figure 2.14. A Linear Optical Quantum Toffoli Gate](image)

The truth table representation of the linear optical quantum Toffoli gate using the Toffoli-sign gate is shown in Table 2.2. From Table 2.2, it can be seen that the linear optical quantum Toffoli gate performs a bit flip on the target qutrit only for the input state $|C_2, C_1 \rangle = |1, 1 \rangle$. The linear optical quantum Toffoli gate implementation as shown in Fig. 2.14 requires 3 two-
qubit gates (2 CNOT gates and 1 Controlled-Z gate) and 6 one-qubit gates (2 $X_a$ gates, 2 Hadamard gates and 2 NOT gates ($X$ gates)).

Table 2.2. Truth Table of Linear Optical Quantum 3x3 Toffoli Gate

<table>
<thead>
<tr>
<th>No.</th>
<th>Control($C_2, C_1$)</th>
<th>Target($T$)</th>
<th>Toffoli Gate ($C_2, C_1, T = C_2 \cdot C_1 \oplus T$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$</td>
<td>0, 0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>$</td>
<td>0, 0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>$</td>
<td>0, 1\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>4</td>
<td>$</td>
<td>0, 1\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>$</td>
<td>1, 0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>6</td>
<td>$</td>
<td>1, 0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>7</td>
<td>$</td>
<td>1, 1\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>8</td>
<td>$</td>
<td>1, 1\rangle$</td>
<td>$</td>
</tr>
</tbody>
</table>
CHAPTER 3

MACH-ZEHNDER INTERFEROMETER BASED DESIGN OF ALL OPTICAL REVERSIBLE BINARY ADDER

1 Optical implementation of reversible logic gates is gaining the attention of researchers as a photon can provide the unmatched high speed and can have the information stored in a signal of zero mass. Reversible logic gates implemented in optical computing can be useful to overcome the limits imposed by conventional computing with minimal energy dissipation, and is also considered one of the feasible alternatives to implement quantum computing [20, 19, 21, 22]. Recently, researchers have also implemented reversible logic gates such as Feynman gate, Toffoli gate, Peres gate and Modified Fredkin gate using semiconductor optical amplifier (SOA)-based Mach-Zehnder interferometer (MZI) optical switch due to its significant advantages such as high speed, low power, fast switching time and ease in fabrication [29, 4, 30]. In this work, we present the all optical implementation of an n bit reversible ripple carry adder for the first time in literature. The all optical reversible adder design is based on two new optical reversible gates referred as optical reversible gate I (ORG-I) and optical reversible gate II (ORG-II) and the existing all optical Feynman gate. The two new reversible gates ORG-I and ORG-II are proposed as they can implement a reversible adder with a reduced optical cost which is the measure of number of MZIs switches and the propagation delay, and with zero overhead in terms of number of ancilla inputs and the

\footnote{Portions of this chapter were published in the Design Automation Test in Europe Conference Exhibition (DATE), 2012 [31]. Permission is included in Appendix A.}
garbage outputs. In any reversible circuit, the ancilla inputs and the garbage outputs are considered as overhead and need to be minimized, as more the number of ancilla inputs and the garbage outputs more will be the number of the I/O pins in the circuit [43]. The proposed all optical reversible adder design based on the ORG-I and ORG-II reversible gates are compared and shown to be better than the other existing designs of reversible adder proposed in the non-optical domain in terms of number of MZIs, delay, number of ancilla inputs and the garbage outputs. The proposed all optical reversible ripple carry adder will be useful to design an all optical reversible ALU that can be applied in a wide variety of optical signal processing applications.

3.1 Proposed All Optical Reversible Gates

In this work, we propose two new optical reversible gates that are most efficient to design an all optical reversible ripple carry adder with input carry. The proposed all optical reversible gates are referred as optical reversible gate I (ORG-I) and optical reversible gate II (ORG-II). The block representation of optical reversible gate I and optical reversible gate II are shown in Fig. 3.2(a) and 3.2(b), respectively. The optical reversible gate I (ORG-I) is a 3 inputs and 3 outputs reversible gate with mapping of inputs (A, B, C) to outputs as \( P = AB + (A \oplus B)C, \quad Q = A \oplus B, \quad R = A\bar{B} + (A \oplus B)C \). The optical implementation of ORG-I is shown in Fig. 3.1(a) using 3 MZI based optical switch (MZI), 4 beam splitters (BS) and 3 beam combiners (BC). The optical reversible gate II is a 3 inputs and 3 outputs reversible gate with mapping between the inputs (A, B, C) and outputs (P, Q, R) as (A, B, C) to \( P = A\bar{B} + BC, \quad Q = BC + \bar{A}B, \quad R = AB + BC \). The optical implementation of an ORG-II is shown in Fig. 3.1(b) using 3 MZI based switches (MZI), 4 beam splitters (BS) and 3 beam combiners (BC). The truth table representation of optical reversible gate I and optical reversible gate II are shown in Tables 3.1 and 3.2, respectively. As the cost of an all optical reversible logic gate is the number of MZI based optical switches, hence the optical
The cost of ORG-I and ORG-II will be considered as 3. In ORG-I out of 3 MZI switches, 2 MZI switches work in parallel. Thus, its delay is considered as $2 \Delta$. The optical implementation of ORG-II has all three MZI based optical switches working in parallel, thus has a delay of $1 \Delta$.

(a) All optical Implementation of Optical Reversible Gate I (ORG-I)

(b) All optical Implementation of Optical Reversible Gate II (ORG-II)

Figure 3.1. All Optical Implementations of Optical Reversible Gate I (ORG-I) and Optical Reversible Gate II (ORG-II)

(a) Optical Reversible Gate I (ORG-I)

(b) Optical Reversible Gate II (ORG-II)

Figure 3.2. Optical Reversible Gate I (ORG-I) and Optical Reversible Gate II (ORG-II)

The graphical representation of optical reversible gate I (ORG-I) and optical reversible gate II (ORG-II) are shown in Figs. 3.3(a) and 3.3(b). As the proposed gates are new and
Table 3.1. Truth Table of Optical Reversible Gate I (ORG-I)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.3. Graphical Representation of Proposed Optical Reversible Gates and Design of An Optical Reversible Full Adder

3.2 Proposed All Optical Reversible Ripple Carry Adder with Input Carry

The optical reversible ripple carry adder with input carry \((c_0)\) is designed without any ancilla inputs and the garbage outputs, and with less optical cost and reduced delay compared to the existing non-optical reversible ripple carry adder design approaches \([44, 45, 46]\). The comparison is done with non-optical reversible ripple carry adders as the proposed work of the design of optical reversible adder is the first attempt in the direction of implementing reversible adder in optical computing. Consider the addition of two \(n\) bit numbers \(a_i\) and \(b_i\) stored at memory locations \(A_i\) and \(B_i\), respectively, where \(0 \leq i \leq n-1\). The input carry \(c_0\) is stored at memory location \(A_{-1}\). Further, consider that memory location \(A_n\) is initialized with \(z \in \{0, 1\}\). At the end of the computation, the memory location \(B_i\) will have \(s_i\), while the location \(A_i\) keeps the value \(a_i\) for \(0 \leq i \leq n-1\). Further, at the end of the computation, the additional location \(A_n\) that initially stores the value \(z\) will have the value \(z \oplus s_n\), and the memory location \(A_{-1}\) keeps the input carry \(c_0\) (here \(s_n\) represents the output carry of the n
bit reversible ripple carry adder). Thus $A_n$ will have the value of $s_n$ when $z=0$. Here, $s_i$ is the sum bit produced and is defined as:

\[
s_i = \begin{cases} 
  a_i \oplus b_i \oplus c_i & \text{if } 0 \leq i \leq n - 1 \\
  c_n & \text{if } i = n
\end{cases}
\]  

(3.1)

where $c_i$ is the carry bit and is defined as:

\[
c_i = \begin{cases} 
  c_0 & \text{if } i = 0 \\
  a_{i-1}b_{i-1} \oplus b_{i-1}c_{i-1} \oplus c_{i-1}a_{i-1} & \text{if } 1 \leq i \leq n
\end{cases}
\]  

(3.2)

Figure 3.4. Circuit Generation of Optical Reversible $n$ Bit Adder Using Proposed Optical Reversible Gate I (ORG-I) and Optical Reversible Gate II (ORG-II)
The generalized methodology of designing the n bit optical reversible ripple carry adder with input carry is shown in Fig. 3.4. The methodology is explained below along with an illustrative example of 4-bit optical reversible ripple carry adder. The illustrative example of 4 bit all optical reversible ripple carry adder is shown in Fig. 3.5 that can perform the addition of two 4 bit numbers \( a = a_0...a_3 \) and \( b = b_0...b_3 \), and has the input carry \( c_0 \). The details of the proposed approach that uses optical reversible gate I (ORG-I) and optical reversible gate II (ORG-II) along with Feynman gates to minimize the optical cost and the delay with zero overhead in terms of garbage outputs and the ancilla inputs is discussed below.

### 3.2.1 Steps of Proposed Methodology for Optical Reversible Adder

- **Step 1** has the following two sub-steps:
  
  - For \( i=0 \) to \( n-1 \): At a pair of locations, \( A_i \), \( B_i \) and \( C_i \) apply the optical reversible gate I (ORG-I) gate such that the location \( A_i \) and \( B_i \) are passed to the inputs \( A \), \( B \) respectively, of the optical reversible gate I (ORG-I).

  For \( i=0 \) the input carry \( C_{in} \) is passed to the input \( C_0 \) of optical reversible gate I, while for the value of \( i=1 \) to \( n-1 \) the input carry is propagated in the circuit. The propagated input carry is generated at the location \( A_i \) of optical reversible gate I in the circuit and passed to location \( C_i \) for \( i=1 \) to \( n-1 \) as \( C_{i+1} = A_{i+1} \). The input \( A_i \) is transformed to \( A_iB_i + (A_i \oplus B_i)C_i \) and is passed to the input \( C_{i+1} \) as propagated carry. So in our case the value of \( C_{i+1} = A_i \) for \( i=1 \) to \( n-1 \). The input \( B_i \) is transformed as \( A_i \oplus B_i \) and input \( C_i \) is transformed as \( A_i \overline{B_i} + (A_i \oplus B_i)C_i \).

  - Further for \( i=n \): apply the optical Feynman gate at pair of locations \( A_{n-1}, A_n \).

- **For \( i=0 \) to \( n-1 \):**

  At pair of locations \( A_i \), \( B_i \) and \( C_i \) apply the optical reversible gate II (ORG-II) gate.

33
such that the location $A_i$ and $B_i$ are passed to the inputs $A$, $B$ respectively, of the optical reversible gate II (ORG-II).

Figure 3.5. Proposed Optical Reversible 4 Bit Adder

For $i=n-1$ the propagated input carry $C_{n-1}$ generated from previous step from optical reversible gate I is passed to the input $C_{n-1}$ of optical reversible gate II, while for the value of $i=n-2$ to 0 the input carry is propagated further in the circuit. The propagated input carry is generated at the location $A_i$ of optical reversible gate II in the circuit and passed to location $C_i$ for $i=n-1$ to 0 as $C_i = A_{i+1}$. The input $A_i$ is transformed to $A_i\bar{B}_i + B_i\bar{C}_i$ and is passed to the input $C_{i-1}$ as propagated carry for $i=n-1$ to 0. So in our case the value of $C_{i-1}$ can be represented as $A_i$ for $i=n-1$ to 0. The input $B_i$ is transformed as $\bar{B}_iC_i + \bar{A}_iB_i$ and input $C_i$ is transformed as $A_iB_i + \bar{B}_iC_i$. After this step we will have the complete working design of the optical reversible adder an example of which is shown for addition of 4 bit numbers in Fig.3.5.
3.2.2 Theorem for Proposed Methodology of Optical Reversible Adder

Let a and b are two n bit binary numbers represented as \( a_i \) and \( b_i \), \( c_0 \) is the input carry \((c_0)\), and \( z \in \{0, 1\} \) is the another 1 bit input, where \( 0 \leq i \leq n - 1 \), then the proposed design steps of methodology result in the ripple carry adder circuit that works correctly. The proposed design methodology designs an n bit adder circuit that produces the sum output \( s_i \) at the memory location where \( b_i \) is initially stored while the location where \( a_i \) is initially stored is restored to the value \( a_i \) for \( 0 \leq i \leq n - 1 \). Further, the memory location where \( z \) is initially stored transforms to \( z \oplus s_n \), and the memory location where the input carry \( c_0 \) is initially stored is restored to the value \( c_0 \).

3.2.3 Proof of Theorem for Proposed Methodology of Optical Reversible Adder

The proposed approach will make the following changes on the inputs that are illustrated as follows:

- **Step 1:** The step 1 of the proposed approach transforms the input \( A_i \), \( B_i \) and \( C_i \) states to

\[
\left( \bigotimes_{i=0}^{n-1} |a_i b_i + (a_i \oplus b_i)c_i\rangle |a_i \oplus b_i\rangle |a_i b_i + (a_i \oplus b_i)c_i\rangle |z \oplus s_n\rangle \right)
\]

where for \( i = 0 \), \( c_i = c_{in} \) else \( c_i = a_i - 1 \).

For illustrative purpose, the transformation of the input states of a 4 bit all optical reversible adder circuit after step 1 is shown in Fig.3.5.

- **Step 2:** The step 2 of the proposed approach transforms the input states to

\[
\left( \bigotimes_{i=0}^{n-1} |a_i\rangle |s_i\rangle |c_0\rangle |z \oplus s_n\rangle \right)
\]

For illustrative purpose, the transformation of the input states of a 4 bit all optical reversible adder circuit after step 2 is shown in Fig.3.5.
Thus, we can see that the proposed two-step will produce the sum output $s_i$ at the memory location where $b_i$ is stored initially while the location where $a_i$ is stored initially will be restored to the value $a_i$ for $0 \leq i \leq n - 1$. The memory location where $z$ is stored will have $z \oplus s_n$ and the memory location where the input carry $c_0$ was stored initially will be restored to the value $c_0$. This proves the correctness of the proposed methodology of designing the all optical reversible ripple carry adder with input carry.

### 3.3 Delay and Optical Cost Analysis

The proposed optical reversible ripple carry adder with input carry can be designed by following the two steps described previously. The delay and optical cost analysis of the proposed adder are performed by analyzing the steps involved in the design of all optical reversible ripple carry adder with input carry.

- **Step 1** of the proposed methodology requires $n$ optical reversible gate I (ORG-I), that works in series thus in this step the proposed design of all optical reversible adder will have the optical cost of $3n$ and delay of $2n\Delta$. Step 1.b has a CNOT gate that is used to generate the carry out, and it contributes the optical cost of 1 and delay of $1\Delta$. Thus, the total optical cost of Step 1 is $3n + 1$ and delay of $2n + 1\Delta$.

- **Step 2** of the proposed methodology requires $n$ optical reversible gate II (ORG-II) working in series thus in this step the proposed design of all optical reversible adder will have the optical cost of $3n$ and delay of $n\Delta$.

The total optical cost of the proposed optical reversible ripple carry adder with input carry can be summed up as $3n + 1 + 3n = 6n + 1$. The propagation delay of the proposed design will be $(2n + 1)\Delta + n\Delta = (3n + 1)\Delta$. 
3.3.1 Comparison of n Bit Optical Reversible Ripple Carry Adders

There are various existing non-optical reversible designs of n bit ripple carry adders in the literature such as the design in [44, 45, 46]. Thus, we are summarizing the optical cost and the delay of the various reversible gates used in the existing work in Table 3.3. The optical cost and the delay summarized in Table 3.3 will be used for comparison of the proposed design of the optical reversible n bit adder with the existing non-optical reversible n bit adders. The Table 3.4 illustrates the comparison of the proposed design with the existing designs proposed in [44, 45, 46].

Table 3.3. Optical Cost and Delay of All Optical Implementation of Reversible Gates

<table>
<thead>
<tr>
<th>Gate</th>
<th>Optical Cost</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feynman Gate [4]</td>
<td>2</td>
<td>1Δ</td>
</tr>
<tr>
<td>Fredkin Gate</td>
<td>2</td>
<td>1Δ</td>
</tr>
<tr>
<td>Peres Gate [30]</td>
<td>4</td>
<td>2Δ</td>
</tr>
<tr>
<td>Toffoli Gate [4]</td>
<td>3</td>
<td>2Δ</td>
</tr>
<tr>
<td>TR Gate</td>
<td>4</td>
<td>2Δ</td>
</tr>
</tbody>
</table>

The Table 3.4 shows that the proposed optical design of the reversible n bit adder excels the existing non-optical reversible designs of n bit adder in terms of optical cost and delay and have zero overhead in terms of number of ancilla inputs and the garbage outputs. From Table 3.5 it can be seen that the proposed design of the optical reversible carry adder achieves the improvement ratios ranging from 12.50% to 24.91%, 63.97% to 66.65%, 60.80% to 66.63% and 66.67% to 68.41% compared to the designs presented in designs of [45, 44, 46] in terms of optical cost. From Table 3.6, it can be seen that the proposed design of optical reversible ripple carry adder achieves the improvement ratios ranging from 0.03% to 3.85%, 24.24% to 24.99%, 25.07% to 32.43% and 25.05% to 30.56% in terms of delay compared to the designs presented in [45, 44, 46], respectively.
Table 3.4. A Comparison of Reversible Ripple Carry Adder with Input Carry

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancilla Inputs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Garbage Outputs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Optical Cost</td>
<td>8n-8</td>
<td>18n-8</td>
<td>18n-19</td>
<td>19n-5</td>
<td>6n+1</td>
</tr>
<tr>
<td>Delay Δ</td>
<td>3n+2</td>
<td>4n+1</td>
<td>4n+5</td>
<td>4n+4</td>
<td>3n+1</td>
</tr>
</tbody>
</table>

1 is the design in [45], 2 is the design 1 in [44], 3 is the design 2 in [44], 4 is the design 2 in [46]

Table 3.5. Optical Cost Comparison of Reversible Ripple Carry Adders (with Input Carry)

<table>
<thead>
<tr>
<th>Bits</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>% Imp. w.r.t 1</th>
<th>% Imp. w.r.t 2</th>
<th>% Imp. w.r.t 3</th>
<th>% Imp. w.r.t 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>56</td>
<td>136</td>
<td>125</td>
<td>147</td>
<td>49</td>
<td>12.50</td>
<td>63.97</td>
<td>60.80</td>
<td>66.67</td>
</tr>
<tr>
<td>16</td>
<td>120</td>
<td>280</td>
<td>269</td>
<td>299</td>
<td>97</td>
<td>19.17</td>
<td>65.36</td>
<td>63.94</td>
<td>67.56</td>
</tr>
<tr>
<td>32</td>
<td>248</td>
<td>568</td>
<td>557</td>
<td>603</td>
<td>193</td>
<td>22.18</td>
<td>66.02</td>
<td>65.35</td>
<td>67.99</td>
</tr>
<tr>
<td>64</td>
<td>504</td>
<td>1144</td>
<td>1133</td>
<td>1211</td>
<td>385</td>
<td>23.61</td>
<td>66.35</td>
<td>66.02</td>
<td>68.21</td>
</tr>
<tr>
<td>128</td>
<td>1016</td>
<td>2296</td>
<td>2285</td>
<td>2427</td>
<td>769</td>
<td>24.31</td>
<td>66.51</td>
<td>66.35</td>
<td>68.31</td>
</tr>
<tr>
<td>256</td>
<td>2040</td>
<td>4600</td>
<td>4589</td>
<td>4859</td>
<td>1537</td>
<td>24.66</td>
<td>66.59</td>
<td>66.51</td>
<td>68.37</td>
</tr>
<tr>
<td>512</td>
<td>4088</td>
<td>9208</td>
<td>9197</td>
<td>9723</td>
<td>3073</td>
<td>24.83</td>
<td>66.63</td>
<td>66.59</td>
<td>68.39</td>
</tr>
<tr>
<td>1024</td>
<td>8184</td>
<td>18424</td>
<td>18413</td>
<td>19451</td>
<td>6145</td>
<td>24.91</td>
<td>66.65</td>
<td>66.63</td>
<td>68.41</td>
</tr>
</tbody>
</table>

* 1 is the design in [45], * 2 is the design 1 in [44], * 3 is the design 2 in [44]
* 4 is the design 2 in [46], * 5 is the proposed design

3.4 Conclusion

In this work, we have presented a new design of optical reversible ripple carry adder using proposed optical reversible gate I and optical reversible gate II. The proposed design of all optical reversible ripple carry adder is primarily optimized for the optical cost and the delay and has zero overhead in terms of number of ancilla inputs and garbage output compared to the existing counterparts. We conclude that the use of the proposed new optical reversible gates for the design of optical reversible ripple carry adder can be very much beneficial in minimizing the optical cost and the delay along with number of ancilla inputs and garbage output. The proposed optical reversible adder design is functionally verified at the logical level by creating a Verilog library of optical reversible gates using Verilog modules of Mach-
### Table 3.6. Delay (in ∆) Comparison of Reversible Ripple Carry Adders (with Input Carry)

<table>
<thead>
<tr>
<th>Bits</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>% Imp. w.r.t 1</th>
<th>% Imp. w.r.t 2</th>
<th>% Imp. w.r.t 3</th>
<th>% Imp. w.r.t 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>26</td>
<td>33</td>
<td>37</td>
<td>36</td>
<td>25</td>
<td>3.85</td>
<td>24.24</td>
<td>32.43</td>
<td>30.56</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>65</td>
<td>69</td>
<td>68</td>
<td>49</td>
<td>2.00</td>
<td>24.62</td>
<td>28.99</td>
<td>27.94</td>
</tr>
<tr>
<td>32</td>
<td>98</td>
<td>129</td>
<td>133</td>
<td>132</td>
<td>97</td>
<td>1.02</td>
<td>24.81</td>
<td>27.07</td>
<td>26.52</td>
</tr>
<tr>
<td>64</td>
<td>194</td>
<td>257</td>
<td>261</td>
<td>260</td>
<td>193</td>
<td>0.52</td>
<td>24.90</td>
<td>26.05</td>
<td>25.77</td>
</tr>
<tr>
<td>128</td>
<td>386</td>
<td>513</td>
<td>517</td>
<td>516</td>
<td>385</td>
<td>0.26</td>
<td>24.95</td>
<td>25.53</td>
<td>25.39</td>
</tr>
<tr>
<td>256</td>
<td>770</td>
<td>1025</td>
<td>1029</td>
<td>1028</td>
<td>769</td>
<td>0.13</td>
<td>24.98</td>
<td>25.27</td>
<td>25.19</td>
</tr>
<tr>
<td>512</td>
<td>1538</td>
<td>2049</td>
<td>2053</td>
<td>2052</td>
<td>1537</td>
<td>0.07</td>
<td>24.99</td>
<td>25.13</td>
<td>25.10</td>
</tr>
<tr>
<td>1024</td>
<td>3074</td>
<td>4097</td>
<td>4101</td>
<td>4100</td>
<td>3073</td>
<td>0.03</td>
<td>24.99</td>
<td>25.07</td>
<td>25.05</td>
</tr>
</tbody>
</table>

* 1 is the design in [45], * 2 is the design 1 in [44], * 3 is the design 2 in [44]
* 4 is the design 2 in [46], * 5 is the proposed design

Zehnder interferometer, beam combiner, and the beam splitter. The proposed efficient design of optical reversible ripple carry adder will find promising applications in optical reversible computing and could form a key component of optical reversible digital processing circuits and architectures.
CHAPTER 4

EFFICIENT REVERSIBLE NOR GATES AND THEIR MAPPING IN

OPTICAL COMPUTING DOMAIN

1 2 In the existing literature there exists two types of optical mapping of reversible logic gates: (i) mapping based on a semiconductor optical amplifier (SOA) using a Mach-Zehnder interferometer (MZI) switch; (ii) mapping based on linear optical quantum computation (LOQC) using linear optical quantum logic gates. In reversible computing, the NAND logic based reversible gates and design methodologies based on them are widely popular. The NOR logic based reversible gates and design methodologies based on them are still unexplored. In this work, we propose two NOR logic based n-input and n-output reversible gates one of which can be efficiently mapped in optical computing using the Mach-Zehnder interferometer (MZI) while the other one can be mapped efficiently in optical computing using the linear optical quantum gates. The proposed reversible NOR gates work as a corresponding NOR counterpart of NAND logic based Toffoli gates. The proposed optical reversible NOR logic gates can implement the reversible boolean logic functions with a reduced number of linear optical quantum logic gates or reduced optical cost and propagation delay compared to their implementation using existing optical reversible NAND gates. It is illustrated that an optical reversible gate library having both optical Toffoli gate and the proposed optical reversible

1Portions of this chapter were published in the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2012 [47]. Permission is included in Appendix A.
2Portions of this chapter were published in the Microelectronics Journal, 2014 [48]. Permission is included in Appendix A.
NOR gate is superior compared to the library containing only the optical Toffoli gate: (i) in terms of number of linear optical quantum gates when implemented using linear optical quantum computing (LOQC), (ii) in terms of optical cost and delay when implemented using the Mach-Zehnder interferometer.

4.1 All Optical Reversible NAND Gates

The Toffoli gate is the widely popular NAND logic gate based reversible gate. The generalized Toffoli gate (GTG) has n-input and n-output with corresponding mapping as $(x_0, x_1, x_2, \ldots, x_{n-2}, x_{n-1}, x_n)$ to $(x_0, x_1, x_2, \ldots, x_n-3, x_n-2, x_0, x_1, x_2, \ldots, x_n-3, x_n-2 \oplus x_{n-1})$, where $x_0, x_1, x_2, \ldots, x_{n-3}, x_{n-2}, x_{n-1}$ are the inputs and $(x_0, x_1, x_2, \ldots, x_n-3, x_n-2, x_0, x_1, x_2, \ldots, x_n-3, x_n-2 \oplus x_{n-1})$ are the outputs, respectively[49]. Figure 4.1(a) shows the block diagram of a generalized Toffoli gate. The optical implementations of Toffoli gate using Mach-Zehnder interferometer and linear optical quantum computing are discussed below.

4.1.1 MZI Based Generalized Toffoli Gate (GTG)

An all optical generalized Toffoli gate can be implemented using n MZI based all optical switches, 1 beam combiner and n+1 beam splitters, where n represents the number of inputs of a generalized Toffoli gate. Fig. 4.1(b) shows the all optical implementation of an 8x8 Toffoli gate as an example to illustrate the all optical implementation of a generalized Toffoli gate. The optical cost of generalized Toffoli gate of n-input an n-output is considered as n as it can be implemented using n MZI based switches. A generalized Toffoli gate with n-input and n-output will have a delay of $(\lceil \log(n - 1) \rceil + 1)\Delta$. 

41
4.1.2 Generalized Linear Optical Quantum Computing Based Reversible Toffoli Gate

The generalized Toffoli gate (GTG) has n-input and n-output with corresponding mapping as \((C_n,C_{n-1},C_{n-2},\ldots,C_2,C_1,T)\) to \((C_n,C_{n-1},C_{n-2},\ldots,C_2,C_1,C_{n-1},C_{n-2},\ldots,C_2,C_1 \oplus T)\), where \(C_n,C_{n-1},C_{n-2},\ldots,C_2,C_1\) are the control qubits and \(T\) is the target signal, respectively[49]. A linear optical quantum computing based implementation of generalized Toffoli gate can be implemented by applying two additional Hadamard gates before and after the target signal.
and by applying additional two NOT gates ($X$ gates) before and after the control qubits ($C_{n-1}$ to $C_1$), except the first control qubit $C_n$ as shown in Fig. 4.2 [26, 27, 28]. The target signal is a $n$ level information carrier, where $n$ is the number of control qubits in a generalized linear optical quantum computing based reversible Toffoli Gate. A generalized linear optical quantum Toffoli gate requires $6n - 3$ number of linear optical quantum gates, where $n$ is the number of control qubit.

![Diagram of generalized linear optical quantum computing based reversible Toffoli Gate](image)

Figure 4.2. Generalized Linear Optical Quantum Computing Based Reversible Toffoli Gate

### 4.2 Proposed NOR Logic Based Reversible Gates and Their Mapping in Optical Domain

In this work, we propose two NOR logic based $n$-input and $n$-output reversible gates one of which can be efficiently mapped in optical computing using the Mach-Zehnder interferometer (MZI) while the other one can be mapped efficiently in optical computing using the linear optical quantum gates. The first reversible NOR gate is called as Mach-Zehnder interferometer based reversible NOR gate (MZI-RNOR), and the second reversible NOR gate is called as linear optical quantum computing based reversible NOR gate (LOQC-RNOR).

#### 4.2.1 Proposed Generalized Reversible NOR Gate for MZI Mapping

The proposed generalized reversible NOR gate (MZI-RNOR) works as a replacement of the existing NAND based generalized Toffoli gate when implemented in MZI based optical computing. The generalized MZI-RNOR gate helps in performing the NOR based implemen-
tation of reversible boolean functions with reduced optical cost and delay. The all optical
generalized MZIRNOR gate has an input to output mapping as \((x_0, x_1, x_2, ..., x_{n-3}, x_{n-2}, x_{n-1})\) to \((x_0, x_1, x_2, ..., x_{n-3}, x_{n-2}, x_0 + x_1 + x_2 + ... + x_{n-3} + x_{n-2} \oplus x_{n-1})\), where \(x_0, x_1, x_2, ..., x_{n-3}, x_{n-2}, x_{n-1}\) are the inputs and \((x_0, x_1, x_2, ..., x_{n-3}, x_{n-2}, x_0 + x_1 + x_2 + ... + x_{n-3} + x_{n-2} \oplus x_{n-1})\) are the outputs, respectively.

Consider a 3x3 MZI-RNOR gate having inputs to outputs mapping as \((A, B, C)\) to \((P = A, Q = B, R = (A + B) \oplus c)\), where \(A, B, C\) are the inputs and \(P, Q, R\) are the outputs, respectively. Figure 4.4(a) and Fig. 4.4(c) shows the block diagram and the all optical implementation of all optical 3x3 MZI-RNOR gate, respectively. The truth table of all optical 3x3 MZI-RNOR gate is shown in Table 4.1. The all optical MZI-RNOR gate will work as an NOR gate when the value of input signal \(C\) is 1. When \(C=1\), the outputs of the MZI-RNOR gate transforms as \(P = A, Q = B\) and \(R = (A + B) \oplus 1 = A + B\). The all optical MZI-RNOR gate working as an NOR gate is shown in Fig. 4.4(b). The all optical MZI-RNOR gate can be implemented using 2 MZI based switches, 4 beam splitters (BS) and 2 beam combiners (BC). The optical cost of the MZI-RNOR gate is considered as 2 since its optical implementation requires 2 MZI based switches. The all optical MZI-RNOR gate has a delay of \(1\Delta\) as in its optical design two MZI switches works in parallel. Following the above idea of 3x3 MZI-RNOR gate, it can be easily followed that a generalized \(n\)-input and \(n\)-output MZI-RNOR gate can work as an \(n-1\) input NOR gate by hardwiring the value of last input as 1.

An all optical generalized MZI-RNOR gate can be implemented using 2 MZI based all optical switches, 2 beam combiners and \(n+1\) beam splitters. Figure 4.3(a) shows the block diagram of an all optical generalized MZI-RNOR gate and Fig. 4.3(b) shows the all optical implementation of an 8x8 generalized MZI-RNOR gate, respectively. The all optical cost of the generalized MZI-RNOR gate is considered as 2 as the generalized MZI-RNOR gate can be implemented using 2 MZI based switches. The generalized MZI-RNOR gate has a
Figure 4.3. Generalized MZI-RNOR Gate and Its MZI Based Optical Implementation (MZI-RNORG: MZI-RNOR Gate, MZI: Mach-Zehnder Interferometer, BC: Beam Combiner, BS: Beam Splitter)

delay of $1\Delta$ as 2 MZI switched works in parallel in the all optical implementation of the generalized MZI-RNOR gate.

4.2.2 Proposed Linear Optical Quantum Computing Based Reversible NOR Gate (LOQC-RNOR)

In this work, we have proposed the linear optical quantum implementation of the reversible NOR gate using the Toffoli-sign gate [26, 27, 28]. Figure 4.5(a) shows a block diagram of the linear optical quantum computing based reversible NOR gate (LOQC-RNOR).
The LOQC-RNOR gate has inputs to outputs mapping as \((C_2, C_1, T)\) to \((C_2, C_1, \overline{C_2C_1} \oplus T)\), where \(C_2, C_1\) are the control signals and \(T\) is the target signal. As oppose to the linear optical quantum computing based Toffoli gate, the linear optical quantum computing based reversible NOR gate flips the state of the target bit only when the values of both control signals are \(C_2 = \ket{0}, C_1 = \ket{0}\) or both the control signals are in state \(\ket{0}\). The working of the proposed linear optical quantum computing based reversible NOR gate as a logical NOR gate and a logical OR gate are shown in Figs.4.5(b) and 4.5(c). A truth table representation of the LOQCRNOR gate is shown in Table 4.2. From the Table 4.2, it can be seen that the state of the target signal flips only in the case when both the control signals are in state \(\ket{0}\) else the target signal remains in the same state. The proposed linear optical quantum computing based reversible NOR gate works as a logical NOR gate when the input signal \(T\) is kept at state \(\ket{0}\), that leads to the input to output transformations as \((C_2, C_1, \ket{1})\) to \((C_2, C_1, \overline{C_2 + C_1})\), here the target signal transformation results in the logical NOR operation on control qubits \((C_2, C_1)\). When the target signal is kept constant at state \(\ket{0}\) the proposed linear optical quantum computing based reversible NOR gate works as a logical OR gate, the input to output transformations in this case are \((C_2, C_1, \ket{0})\) to \((C_2, C_1, C_2 + C_1)\), here the target signal transformation results in the logical OR operation on control qubits \((C_2, C_1)\).
Figure 4.4. 3x3 MZI-RNOR Gate and Its MZI Based Optical Implementation (MZI: Mach-Zehnder Interferometer, BC: Beam Combiner, BS: Beam Splitter)

A linear optical quantum implementation of the proposed reversible NOR gate using the linear optical quantum Toffoli-sign gate is shown in Fig. 4.6. A linear optical quantum Toffoli-sign gate works as TNOR gate when two $X$ gates are applied before and after first control qubit ($C_2$) and two Hadamard gates are applied before and after the target qutrit as shown in Fig. 4.6. The linear optical quantum implementation of the reversible NOR gate can be divided into seven stages ((a),(b),(c),(d),(e),(f),(g)) after applying all the linear optical quantum transformations at each stage, the value of all the output signals can be observed at stage (g) for the correctness of the design. The two control qubits are referred as $C_2,C_1$. 

47
can have two possible states $|0\rangle$ or $|1\rangle$. In the proposed linear optical quantum computing based reversible NOR gate the target signal is a qutrit and can have three possible states $|0\rangle$, $|1\rangle$ and $|2\rangle$. The representation of a qubit and qutrit in linear optical quantum computing is shown in Fig. 2.11. An illustration of the linear optical quantum transformations performed for the proposed linear optical quantum computing based reversible NOR gate for control signal states $|C_2, C_1\rangle = |0, 0\rangle$ are shown in Fig. 4.7 as a proof of the correctness of the design. The linear optical quantum transformations performed for all eight stages are shown in Fig. 4.7.

The proposed linear optical quantum computing based reversible NOR gate as shown in Fig. 4.6 requires 3 two-qubit gates (2 CNOT gates and 1 Controlled-Z gate) and 6 one-qubit
Table 4.2. Truth Table for Linear Optical Quantum Computing Based 3x3 Reversible NOR Gate (LOQC-RNOR)

<table>
<thead>
<tr>
<th>No.</th>
<th>Control($C_2, C_1$)</th>
<th>Target($T$)</th>
<th>Toffoli Gate ($C_2, C_1, T = C_2 \cdot C_1 \oplus T$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$</td>
<td>0,0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>$</td>
<td>0,0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>$</td>
<td>0,1\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>4</td>
<td>$</td>
<td>0,1\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>$</td>
<td>1,0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>6</td>
<td>$</td>
<td>1,0\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>7</td>
<td>$</td>
<td>1,1\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>8</td>
<td>$</td>
<td>1,1\rangle$</td>
<td>$</td>
</tr>
</tbody>
</table>

Figure 4.6. A Linear Optical Quantum Computing, Based Reversible NOR Gate (LOQC-RNOR)

That is equal to the number of linear optical quantum logic gates required to implement linear optical quantum Toffoli gate.

4.2.3 Proposed Generalized Linear Optical Quantum Computing Based Reversible NOR Gate

The proposed generalized linear optical quantum computing based reversible NOR Gate has n-input and n-output with corresponding mapping as $(C_n, C_{n-1}C_{n-2}..., C_2, C_1, T)$ to
Figure 4.7. Transformation of The Proposed Linear Optical Quantum Computing Based Reversible NOR Gate (LOQC-RNOR) for Control Signal Values ($|C_2, C_1\rangle = |0, 0\rangle$)

* No Transformations are Performed at Stage (c) and (e), as Controlled-NOR Gate Has $C_1 = |0\rangle$ as Control Value

\[
C_2 = |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad C_1 = |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}
\]

\[
T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}
\]

(a) (b) (c) (d)

\[
C_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}
\]

\[
T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}, \quad T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}
\]

(f) (g) (h)

\[
(C_n, C_{n-1}C_{n-2}...C_2, C_1, \overline{C_nC_{n-1}C_{n-2}...C_2C_1} \oplus T), \quad \text{where } C_n, C_{n-1}C_{n-2}..., C_2, C_1 \text{ are the control signals and } T \text{ is the target signal, respectively.}
\]

A linear optical quantum computing based implementation of generalized Toffoli NOR gate can be implemented by applying two additional Hadamard gates before and after the target signal and by applying additional two $X$ gates before and after the first control qubit $C_n$ of a generalized Toffoli-sign gate as shown in Fig. 4.8. The target signal is a $n$ level information carrier, where $n$ is the number of control qubits in a proposed generalized linear optical quantum computing based reversible NOR Gate. A generalized linear optical quantum computing based reversible NOR Gate requires $4n + 1$ number of linear optical quantum gates, where $n$ is the number of control qubits.
4.3 Comparison of Proposed NOR Logic Based Reversible Gates with NAND Logic Based Reversible Gates Mapped in Optical Domain

The proposed reversible NOR gate can provide an NOR-based implementation of reversible boolean functions. In the existing literature, the Toffoli gates are used for NAND-based implementation of reversible boolean functions. The proposed reversible NOR gate has advantages compared to the existing reversible NAND gates in terms of optical cost and delay for the implementation based on Mach-Zehnder interferometer. When mapped in linear optical quantum computing, the proposed reversible NOR gate provides an advantage in terms of number of linear optical quantum logic gates.

4.3.1 Comparison of 3-Input and 3-Output Gates

In the existing literature, the 3-input and 3-output NAND logic based Toffoli gate is mostly used. Hence, we are comparing our proposed MZI-RNOR gate and with its 3x3 Toffoli counterpart. Table 4.3 shows the optical cost and delay analysis of existing 3x3 reversible NAND gates and the proposed 3x3 reversible NOR gates mapped in MZI switch. The proposed MZI-RNOR gate (Optical cost=2, Delay=1∆) has reduced optical cost and delay compared to the existing Toffoli gate that has the optical cost of 3 and delay of 2∆.

In the existing literature, researchers have proposed 13 standard boolean functions that can represent all 256 possible combinations of three variable boolean function[32]. To illus-
Table 4.3. Optical Cost and Delay of 3x3 Reversible NAND Logic Gate and Proposed 3x3 MZI-RNOR Gate Mapped In MZI Switch

<table>
<thead>
<tr>
<th></th>
<th>Optical Cost</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toffoli Gate [4]</td>
<td>3</td>
<td>2Δ</td>
</tr>
<tr>
<td>Proposed MZI-RNOR Gate</td>
<td>2</td>
<td>1Δ</td>
</tr>
</tbody>
</table>

trate the advantages of proposed 3x3 reversible NOR gate (MZI-RNOR) in terms of optical cost and delay, 13 standard boolean functions are implemented using the MZI-RNOR as well as the existing 3x3 reversible NAND gate. The comparison study is shown in Table 4.4. The table 4.4 provides the optical cost and delay analysis of all 13 standard boolean functions using (i) existing 3x3 reversible NAND logic gates (MZI based 3x3 Toffoli gate), (ii) proposed 3x3 MZIRNOR gate, (iii) using combination of MZI based 3x3 Toffoli gate and the proposed 3x3 MZIRNOR gate. The implementation of 13 standard boolean functions using proposed 3x3 MZI-RNOR gates has 20.57% improvement in terms of total optical cost, and 50.52% improvement in terms of total delay compared to the implementation using existing MZI based 3x3 Toffoli gate. The combination of MZI based 3x3 Toffoli gate and the proposed 3x3 MZI-RNOR gate shows the 37.32% improvement in terms of optical cost and 26.80% improvement in terms of delay compared to the implementation using only MZI based 3x3 Toffoli gate.

4.3.2 Comparison of n-Input and n-Output Mach-Zehnder Interferometer Based Reversible Logic Gates

The proposed nxn MZI-RNOR gate shows significant advantages in terms of optical cost and optical delay compared to MZI based nxn Toffoli gate. We can observe that the optical cost of MZI based nxn Toffoli gate grows linearly (proportional to n) while the optical cost of the nxn MZI-RNOR gate remains constant at 2. The delay of the MZI based nxn Toffoli gate grow logarithmically $(\lceil \log(n - 1) \rceil + 1)\Delta$, while the delay of the proposed nxn MZI-RNOR gate remains constant as 1. This shows that the proposed all optical MZI-RNOR
Table 4.4. Optical Cost and Delay Analysis of 3x3 Reversible Logic Gates by Implementing 13 Standard Boolean Functions

<table>
<thead>
<tr>
<th>No.</th>
<th>Standard Function</th>
<th>Using Toffoli gate</th>
<th>Using proposed MZI-RNOR gate</th>
<th>Using Toffoli gate and proposed MZI-RNOR gate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>OC*</td>
<td>Delay</td>
<td>OC*</td>
</tr>
<tr>
<td>1</td>
<td>$F = ABC$</td>
<td>6</td>
<td>4Δ</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>$F = AB$</td>
<td>3</td>
<td>2Δ</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>$F = ABC + ABC$</td>
<td>19</td>
<td>10Δ</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>$F = ABC + ABC$</td>
<td>21</td>
<td>9Δ</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>$F = AB + BC$</td>
<td>9</td>
<td>6Δ</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>$F = AB + ABC$</td>
<td>16</td>
<td>9Δ</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>$F = ABC + ABC + ABC$</td>
<td>30</td>
<td>15Δ</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>$F = A$</td>
<td>3</td>
<td>2Δ</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>$F = AB + BC + AC$</td>
<td>15</td>
<td>10Δ</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>$F = AB + BC$</td>
<td>11</td>
<td>5Δ</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>$F = AB + BC + ABC$</td>
<td>24</td>
<td>9Δ</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>$F = AB + AB$</td>
<td>13</td>
<td>5Δ</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>$F = ABC + ABC + ABC + ABC$</td>
<td>39</td>
<td>11Δ</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>209</strong></td>
<td><strong>97Δ</strong></td>
<td><strong>166</strong></td>
</tr>
</tbody>
</table>

% improvement w.r.t. implementation using existing MZI based Toffoli gate: 20.57% OC, 50.52% Delay

*OC: Optical Cost

A Toffoli gate is superior compared to MZI based Toffoli gate as its optical cost and delay remains constant irrespective of the value of n. Next, we implemented 13 standard boolean functions that can represent all 256 possible combinations of three variable boolean function[32] using three different libraries (i) containing only MZI based nxn Toffoli gate, (ii) containing only nxn MZI-RNOR gate, and (iii) containing both MZI based nxn Toffoli and the proposed MZI-RNOR gate. The mapping results are shown in Table 4.5. The results show that the library containing only MZI-RNOR gate achieves 21.54% improvement in terms of optical cost and 50.64% in terms of optical delay compared to the library containing only MZI based nxn Toffoli gate. Further, the library containing both MZI based nxn Toffoli and the proposed MZI-RNOR gate achieves better results in terms of optical cost and delay. The library containing both MZI based nxn Toffoli and the proposed MZI-RNOR gate achieves 24.10% improvement in terms of optical cost and 35.06% in terms of optical delay compared to the library containing only MZI based nxn Toffoli gate. This shows that a hybrid library...
containing \( n \times n \) Toffoli and \( n \times n \) MZI-RNOR gates will produce an efficient design in terms of optical cost and delay.

4.3.3 Comparison of Proposed Reversible NOR Gate (LOQC-RNOR) with Existing Toffoli Gate Mapped in Linear Optical Quantum Computing Domain

As illustrated earlier the generalized \((n\text{-inputs and } n\text{-outputs})\) linear optical quantum Toffoli gate requires \(6n - 3\) number of linear optical quantum gates, while the proposed generalized linear optical quantum computing based reversible NOR gate (LOQC-RNOR) requires \(4n + 1\) number of linear optical quantum gates, where \( n \) is the number of control qubits. A comparison plot of the number of linear optical quantum gates required to design a generalized LOQC-RNOR gate and generalized linear optical quantum computing based Toffoli gate is shown in Fig. 4.10 for various values of \( n \) (number of qubits). For a large value of \( n \) up to 33% less number of linear optical quantum, gates are needed in the proposed
L0QC-RNOR gate compared to the Toffoli gate. Hence, the proposed generalized L0QC-RNOR gate provides a significant advantage in terms of number of linear optical quantum logic gates compared to design of generalized linear optical quantum computing based Toffoli gate.

![Diagram of three-control-qubit linear optical quantum LOQC-RNOR gate and Toffoli gate](image)

(a) Three-control-qubit linear optical quantum LOQC-RNOR gate

(b) Three-Control-Qubit Linear Optical Quantum Toffoli Gate

Figure 4.9. Three-Control-Qubit Linear Optical Quantum LOQC-RNOR and Toffoli Gate (*Here $X_b$ Gate Swaps The State $|1\rangle$ and $|3\rangle$, While The $X_a$ Gate Swaps The Information Between State $|0\rangle$ and $|2\rangle$; The Target Is a Four Level Information Carrier)

### 4.4 Discussion and Conclusions

In this work, we have proposed two new reversible NOR gates. The first reversible NOR gate is useful for mapping in MZI based optical computing while the second reversible NOR gate is useful for mapping in linear optical quantum computing. The proposed reversible NOR gates can work as a replacement for NAND logic based reversible Toffoli gates, and can
Figure 4.10. Comparison of Number of Linear Optical Quantum Gates Required in Proposed LOQC-RNOR Gate Versus Toffoli Gate for Various Values of Number of Qubits

also be used along with Toffoli gates to design efficient reversible circuits. It is illustrated that the proposed optical reversible NOR gates have significant advantages over existing reversible NAND gates in terms of optical cost, delay, and number of linear optical quantum gates. We proved that a hybrid reversible library containing both nxn Toffoli gate and proposed nxn reversible NOR gate would be the ideal choice to design efficient optical reversible circuits. In conclusion, this work advances the state of the art of reversible optical circuits by providing NOR logic based reversible gates as an alternative to NAND logic based reversible gates.
In this work, we have proposed the design methodologies based on (i) \( a - b = \overline{a} + b \), and (ii) \( a - b = a + \overline{b} + 1 \), to design a reversible adder-subtractor, that is controlled by the control signal to perform addition or subtraction operation. The proposed design of reversible adder-subtractor is based on the design of efficient ripple carry adder proposed in this dissertation earlier. The proposed optical reversible adder-subtractor will be a key component of an optical reversible arithmetic logical unit (ALU) that can be applied in a wide variety of optical signal processing applications. The proposed design of reversible adder-subtractor shows significant improvements in terms of number of ancilla inputs, garbage outputs, optical cost and delay.

5.1 Design of Unified Optical Reversible Adder-Subtractor  

In this section, we discuss the design of unified optical reversible adder-subtractor using the proposed design of optical reversible ripple carry adder with input carry that can work as an adder as well as a subtractor depending on the value of the control signal (ctrl).
5.1.1 Design of Optical Reversible Adder-Subtractor Based on Approach 1

The approach 1 to design an n bit reversible adder-subtractor uses the property $a - b = \overline{a} + \overline{b}$. Using this approach of $a - b = \overline{a} + \overline{b}$, an n bit reversible adder-subtractor can be designed by using an n bit reversible ripple carry adder with input carry ($C_{in}$). By using the control signal, the reversible n bit subtraction operation can be performed based on n bit reversible ripple carry adder by complementing the input a at the start, and finally complementing the sum produced at the end as illustrated in Fig. 5.1. The complementing of the required inputs and outputs can be performed using the Feynman gates (CNOT). The optical reversible n bit adder-subtractor can be designed based on the proposed optical ripple carry adder with input carry by complementing the input a at the start, and finally complementing the sum produced at the end and hardwiring the input carry $C_{in}$ as 0. The optical reversible ripple carry adder with the input carry has the optical cost of $(6n + 1)$ and delay of $(3n + 1)\Delta$. As the CNOT gate has the optical cost of 2 and delay of $1\Delta$, the proposed design of n bit optical reversible adder-subtractor has the optical cost of $10n + 1$ and the delay as $(5n + 1)\Delta$.

5.1.2 Design of Optical Reversible Adder-Subtractor Based on Approach 2

Another possible approach to design an n bit reversible adder-subtractor that can add and subtract two n bit numbers a and b is to use the property $a - b = a + \overline{b} + 1$ to perform the subtraction operation. Thus, an n bit reversible adder-subtractor can be designed by using the n bit optical reversible ripple carry adder with input carry that performs the operation $a + b + C_{in}$ where a and b are n bit numbers and $C_{in}$ is the input carry. Here the input carry $C_{in}$ will be used as the control signal referred as $ctrl$. The complementing of the input b at the start can be performed by using the CNOT gate using control signal $ctrl$ that controls the complementing of the input b at the start and can also add 1 to the adder to perform the
Figure 5.1. Proposed Optical Reversible n Bit Adder-Subtractor Based on Approach 1

n bit subtraction. The design of n bit optical reversible adder-subtractor based on approach 2 is illustrated in Fig. 5.3. As illustrated in Fig. 5.3 if ctrl=1 the design will complement the controlled input b and add 1 and will work as an n bit reversible subtractor. Otherwise, when ctrl=0 the controlled input b is passed as such and the design will work as n bit reversible full adder. The design of the proposed n bit optical reversible adder-subtractor based on the optical reversible ripple carry adder with input carry is illustrated in Fig. 5.4. The proposed n bit optical reversible adder-subtractor has the optical cost of $8n + 1$ and optical delay of $(4n + 1)\Delta$ and is designed without any ancilla input and the garbage output.

5.2 Comparison of n Bit Reversible Adder-Subtractor

As discussed above the n bit optical reversible adder-subtractor based on approach 1 has the optical cost of $10n + 1$, delay of $(5n + 1)\Delta$, 1 ancilla input and 0 garbage output. The design of n bit optical reversible adder-subtractor based on approach 2 has the optical cost
Figure 5.2. Proposed Optical Reversible n Bit Adder-Subtractor Based on The Optical Ripple Carry Adder with Input Carry

of $8n + 1$, delay of $(4n + 1)\Delta$, 0 ancilla input and 0 garbage output. Thus, design based on approach 2 is most efficient among the proposed optical reversible adder-subtractor designs as it has 0 ancilla inputs and 0 garbage outputs and has less optical cost and delay. The results show that the design approach 2 will provide an efficient way of designing an n bit reversible adder-subtractor using the proposed optical reversible ripple carry adder with input carry. All the results are summarized in Table 5.1.

Table 5.1. A Comparison of The Proposed Reversible Adder-Subtractor

<table>
<thead>
<tr>
<th></th>
<th>Ancilla Inputs</th>
<th>Garbage Outputs</th>
<th>Optical Cost</th>
<th>Optical Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design based on Approach 1</td>
<td>1</td>
<td>0</td>
<td>$10n+1$</td>
<td>$5n+1$</td>
</tr>
<tr>
<td>Design based on Approach 2</td>
<td>0</td>
<td>0</td>
<td>$8n+1$</td>
<td>$4n+1$</td>
</tr>
</tbody>
</table>
Figure 5.3. Proposed Optical Reversible n Bit Adder-Subtractor Based on Approach 2

Since the proposed approach 2 for designing the reversible adder-subtractor is efficient compared to proposed approach 1, we are only showing the comparison of reversible adder-subtractor design based on approach 2 with the existing design approaches (existing reversible adder designs if implemented as a reversible adder unit in approach 2). Table 5.2 illustrate the comparison of the proposed optical reversible adder-subtractor design based on approach 2 \((a - b = a + \bar{b} + 1)\) with the existing design approaches (existing reversible adder designs if implemented as a reversible adder unit in approach 2) proposed in [44, 45, 46]. The Table 5.2 shows that the proposed optical design of optical reversible n bit adder-subtractor based on approach 2 excels the designs of reversible adder-subtractor using existing non-optical reversible designs of n bit adder in terms of optical cost and delay. From Table 5.3 it can be seen that the proposed design of the optical reversible adder-subtractor achieves the improvement ratios ranging from 9.72% to 19.93%, 57.24% to 59.98%, 53.90% to 59.96% and 60.12% to 61.89% compared to the existing design approaches (existing reversible adder designs if implemented as a reversible adder unit in approach 2) proposed in [44, 45, 46] in terms of optical cost. Also from Table 5.4, it can be seen that the proposed design of
optical reversible adder-subtractor achieves the improvement ratios ranging from 0.02% to 2.94%, 19.51% to 20.00%, 20.06% to 26.67% and 20.04% to 25.00% in terms of delay compared to the existing design approaches (existing reversible adder designs if implemented as a reversible adder unit in approach 2) proposed in [44, 45, 46], respectively.

Table 5.2. A Comparison of Reversible Adder-Subtractor Based on Approach 2

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancilla Inputs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Garbage Outputs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Optical Cost</td>
<td>10n-8</td>
<td>20n-8</td>
<td>20n-19</td>
<td>21n-5</td>
<td>6n+1</td>
</tr>
<tr>
<td>Delay $\Delta$</td>
<td>4n+2</td>
<td>5n+1</td>
<td>5n+5</td>
<td>5n+4</td>
<td>3n+1</td>
</tr>
</tbody>
</table>

1 is the design in [45], 2 is the design 1 in [44]
3 is the design 2 in [44], 4 is the design 2 in [46]
5.3 Conclusion

In this work, we have presented two new designs of reversible adder-subtractors. The proposed design methodologies of reversible adder-subtractor have been illustrated to optimize the parameters of ancilla input bits, the number of garbage outputs, optical cost and the delay. We conclude that the use of the proposed new reversible adder-subtractor can be very much beneficial in minimizing the optical cost and the delay along with number of ancilla inputs and garbage output. The proposed efficient design of optical reversible adder-
subtractor will find promising applications in optical reversible computing and could form a key component of optical reversible digital processing circuits and architectures.
CHAPTER 6

REVERSIBLE LOGIC BASED MULTIPLICATION COMPUTING UNIT

USING BINARY TREE DATA STRUCTURE

1 2 In this work, we have proposed the binary tree-based design methodology for the signed as well as unsigned reversible multipliers that are scalable (generalized for a NxN reversible multiplier design). The proposed design methodology primarily optimizes the number of ancilla inputs and garbage outputs. This is because the proposed binary tree-based design methodology for NxN reversible multiplier performs the addition of partial products in parallel using the reversible ripple adders with zero ancilla bit and zero garbage bit, thereby minimizing the number of ancilla and garbage bits used in the design. The beauty of the proposed reversible multiplier is that it is generic in nature as any existing or future reversible ripple carry adder can be embedded to design an efficient multiplier optimizing the number of ancilla inputs and garbage outputs.

6.1 Proposed Binary Tree-Based Design Methodology for NxN Reversible Multiplier

A binary tree-based design methodology for a NxN reversible multiplier has been proposed in this work. In general, a NxN bit multiplication operation first requires the gener-

1 Portions of this chapter were published in the 27th International Conference on VLSI Design and 13th International Conference on Embedded Systems, 2014 [51]. Permission is included in Appendix A.
2 Portions of this chapter were published in the Journal of Supercomputing, 2015 [52]. Permission is included in Appendix A.
ation of partial products and after that, the summation of partial products are performed to generate the final multiplication output. We propose a design methodology based on the binary tree structure for the addition of partial products in parallel. The binary tree representation of a NxN reversible multiplier design using the proposed design methodology is shown in Fig. 6.1. In Fig. 6.1, the leaf nodes \( p_0, p_1, p_2, \ldots, p_{N-3}, p_{N-2}, p_{N-1} \) represent the partial products that are generated by the reversible partial product generation circuitry. The root node \( (S) \) represents the final multiplication result. The primary advantage of the proposed methodology is that it employs reversible ripple adders with no ancilla bit and garbage bit for addition \([53, 54, 55, 44, 56, 57, 58]\) at all the levels that are responsible for the addition of either partial products or the partial sums in parallel. This significantly reduces the ancilla and the garbage bits overheads. Figure 6.1 plus (+) represents the nodes working as the reversible ripple adders with zero ancilla bit and zero garbage bit. The partial sums generated at each stage are represented as \( ps_0, ps_1, ps_2, ps_3, \ldots, ps_{N/2^l-3}, ps_{N/2^l-2}, ps_{N/2^l-1} \), where \( l = 1, 2, 3, \ldots, \log_2 N \) represents the level of binary tree. Since the proposed multiplier design is based on the binary tree, it has \( l = \log_2 N \) levels. At each level, the number of reversible ripple adders with zero ancilla bit and zero garbage bit required to perform the addition of the adjacent partial products or the adjacent partial sums in parallel can be computed as \( N/2^l \), where \( l \) represents the levels of the binary tree. The width of the reversible ripple adders with no ancilla bit and garbage bit required at each level can be computed as \( N + l - 2^{l-1} - 2 \), where \( l \) represents the respective levels of the binary tree.

The proposed design methodology for a NxN reversible multiplier is summarized in the following steps:

- **Step 1:** The \( N \) partial products are generated using the reversible partial product generation circuit as shown in Fig. 6.2. The reversible partial product generation circuit is implemented using the Toffoli gates to minimize the number of ancilla inputs and garbage outputs. The partial product generation circuit as shown in Fig. 6.2
requires an \( N^2 \) Toffoli gates to generate an \( N \) partial products for a \( N \times N \) reversible multiplier. This step is similar to the reversible multiplier designs proposed in the existing literature [59, 60, 61].

- **Step 2:** As illustrated earlier in Fig. 6.1, a binary tree representation of an \( N \times N \) reversible multiplier has \( l = \log_2 N \) levels. This step is responsible for the addition of adjacent partial products or adjacent partial sums in parallel using reversible ripple adders that have zero ancilla bit and zero garbage bit. This step can further be divided into \( l = \log_2 N \) stages, wherein each stage, the adjacent partial products or the adjacent partial sums are added in parallel using reversible ripple adders that have zero ancilla bit and zero garbage bit. The number of reversible ripple adders with zero ancilla bit and zero garbage bit required at each stage can be computed as \( N/2^l \) and the width
of reversible ripple adders with no ancilla bit and garbage bit can be computed as $N + l - 2^{l-1} - 2$-bit, where $l$ represents the respective stage:

- **Stage I ($l = 1$):** In the first stage, the addition of adjacent partial products are performed using the reversible ripple adders. The first stage requires ($N/2^1 = N/2$) number of ($N+1+2^{1-1}-2 = N$)-bit reversible ripple adders to perform the addition of $N$ adjacent partial products in parallel. After performing the addition of $N$ partial products using $N/2$ reversible ripple adders, this stage generates $N/2$ number of partial sums.
• Stage II ($l = 2$): This stage performs the addition of $N/2$ adjacent partial sums in parallel using the reversible ripple adders. This stage requires $(N/2^2 = N/4)$ number of $(N + 2 + 2^{2-1} + 2 = N + 2)$-bit reversible ripple adders. In this stage, the addition of $N/2$ partial sums generates further $N/4$ partial sums.

• Stage III to Stage $\log_2 N - 1$ ($l = 3$ to $\log_2 N - 1$): The stages from stage III to stage $\log_2 N - 1$ also use the reversible ripple adders to perform the addition of adjacent partial sums in parallel and follow the same methodology used in the previous stages to generate the partial sums.

• Stage $\log_2 N$ ($l = \log_2 N$): This stage is responsible for the addition of adjacent partial sums generated by the previous stage (Stage $\log_2 N - 1$). At this stage, a reversible ripple adder ($N/2^{\log_2 N} = N/N = 1$) is required to perform the addition of partial sums. The size of reversible ripple adder required in this stage can be computed as $(N + l + 2^{l-1} - 2)$-bit. The reversible ripple adder used in this stage generates the final multiplication result.

\[
\begin{array}{cccccc}
\text{X} & x_3 & x_2 & x_1 & x_0 \\
\text{y_3} & y_2 & y_1 & y_0 \\
\hline
\text{P_0} & x_3 y_0 & x_2 y_0 & x_1 y_0 & x_0 y_0 \\
\text{P_1} & x_3 y_1 & x_2 y_1 & x_1 y_1 & x_0 y_1 & \text{X} \\
\text{P_2} & x_3 y_2 & x_2 y_2 & x_1 y_2 & x_0 y_2 & \text{X} & \text{X} \\
\text{P_3} & x_3 y_3 & x_2 y_3 & x_1 y_3 & x_0 y_3 & \text{X} & \text{X} \\
\hline
\text{PS_0} & c_{10} & p_{S_1} & p_{S_2} & p_{S_3} & p_{S_4} & x_0 y_0 \\
\text{PS_1} & c_{20} & p_{S_{13}} & p_{S_{12}} & p_{S_{11}} & p_{S_{10}} & x_0 y_2 & \text{X} & \text{X} \\
\hline
\text{S_8} & S_7 & S_6 & S_5 & S_4 & S_3 & S_2 & S_1 & S_0 \\
\end{array}
\]

Figure 6.3. Addition of Partial Products for a 4x4 Multiplication ($c_{10}, c_{20}$: Carry Outs)
6.2 Design Example of a 4x4 Bit Reversible Multiplier Based on Proposed Methodology

An illustration of our proposed binary tree-based design methodology for a 4x4 reversible multiplier design is shown in Fig. 6.3. A 4x4 bit multiplication operation requires 4 partial products to be generated and the the addition of adjacent partial products or partial sums is performed in parallel to generate the final multiplication result. The binary tree representation of the 4x4 reversible multiplier is shown in Fig. 6.4, here the leaf nodes \((P_0, P_1, P_2, P_3)\) represent the 4 partial products and the root node \((S)\) represents the final multiplication result. The binary tree representation of a 4x4 multiplier has \(l = \log_2 4 = 2\) levels. The nodes between the root node and the leaf nodes with a plus (+) sign represent the reversible ripple adders required to perform the addition. The partial sums generated after the addition at level 2 is represented as \(PS_0, PS_1\) in Fig. 6.4. The design of a 4x4 reversible multiplier using the proposed design methodology can be illustrated in the following steps:
• Step 1: The 4 partial products can be generated by using the reversible partial product generation circuit using the Toffoli gates. An example of an N-bit reversible partial product generation circuit is shown in Fig. 6.2. To generate the 4 partial products $4^2 = 16$ number of Toffoli gates are required.

• Step 2: In this step, the addition of 4 adjacent partial products in parallel are performed by using the reversible ripple adders. As illustrated earlier the 4x4 reversible multiplier design can be further divided into $(l = \log_24 = 2)$ stages, where $l$ represents the respective stage:

• Stage I ($l = 1$): In this stage, the additions of adjacent partial products are performed by reversible ripple adders. The addition of 4 partial products requires $(4/2^1 = 4/2 = 2)$ reversible ripple adders working in parallel that generates 2 partial sums. This stage requires $(4 + 1 + 2^{1-1} - 2 = 4)$-bit reversible ripple adders to perform the addition of 4 partial products in parallel.

• Stage II ($l = 2$): This stage requires one reversible ripple adder $(2/2^{2-1} = 1)$ to perform the addition of 2 adjacent partial sums generated by the first stage. This stage requires a $(4 + 2 + 2^{2-1} - 2 = 6)$-bit reversible ripple adder to perform the addition operation of 2 adjacent partial sums and generate the final multiplication output.

Figure 6.5 shows step 2 of the proposed design methodology for a 4x4 reversible multiplier. Figure 6.5 also shows the addition of partial products and partial sums in two stages referred to as Stage I and Stage II, wherein stage I, two 4 bit reversible ripple adders work in parallel to perform the addition of 4 partial products, while in the second stage, a 6 bit reversible ripple adder is used to perform the addition of partial sums.

In the existing literature, researchers have proposed several designs of reversible ripple adders [46, 56, 44, 58, 62]. The proposed reversible multiplier methodology is generic in
nature, and any N-bit reversible ripple carry adder with input carry having zero ancilla inputs, and zero garbage outputs can be used in the design. For illustration purposes, we have used the design of reversible ripple adder with input carry proposed in [46], as it requires zero ancilla inputs and produces zero garbage outputs. The design of a 4x4 reversible multiplier using reversible ripple adder with input carry is shown in Fig. 6.6. In Fig. 6.6, the partial product generation circuit is implemented using the Toffoli gates. The partial product generation circuit requires 16 Toffoli gates. As illustrated earlier, the first stage requires two 4-bit reversible ripple adders with input carry working in parallel. The reversible ripple adders with input carry used in the first stage perform the addition of adjacent partial products in parallel and generates two partial sums. In the second stage, a 6-bit reversible ripple adder with input carry is required for the addition of 2 partial sums. The 6-bit reversible ripple adder with input carry used in the second stage generates the final multiplication output.
Figure 6.6. A 4x4 Reversible Multiplier Design Using Reversible Ripple Adders with Input Carry (G₀ to G₂₁: Garbage Outputs; P₀ to P₈: Final Result)
6.3 Comparison of Proposed Reversible Multiplier Design with Existing Designs

The proposed binary tree-based design methodology for NxN reversible multiplier performs the addition of partial products in parallel using the reversible ripple adders with input carry. In the existing literature, researchers have proposed several designs of reversible multipliers [59, 60, 61, 63]. The existing designs of reversible multipliers have significant overheads in terms of the number of ancilla inputs and garbage outputs. The proposed binary tree-based design methodology for NxN reversible multiplier performs the addition of partial products in parallel using reversible ripple adders with input carry and provides significant advantages in terms of the number of ancilla inputs and the number of garbage outputs. The proposed binary tree-based design methodology can be divided into stages, and the number of stages can be determined by \( l = \log_2 N \), where \( l \) represents the respective stage. At each stage of the proposed design methodology, the number of ancilla inputs can be computed as \( AN = N/2^l \times (2^{l-1} + 1) \) and the number of garbage outputs can be computed as \( GO = N/2^l \times (N + 2^{l-1} + l - 2) \), where \( l \) represents the respective level. The N-bit partial product generation circuit as shown in Fig. 6.2 requires \( N^2 \) ancilla inputs and generates \( 2 \times N \) garbage outputs.

<table>
<thead>
<tr>
<th>Ancilla Inputs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>% Imp. w.r.t 1</th>
<th>% Imp. w.r.t 2</th>
<th>% Imp. w.r.t 3</th>
<th>% Imp. w.r.t 4</th>
<th>% Imp. w.r.t 5</th>
<th>% Imp. w.r.t 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>56</td>
<td>52</td>
<td>28</td>
<td>28</td>
<td>30</td>
<td>23</td>
<td></td>
<td>60.34</td>
<td>58.93</td>
<td>55.76</td>
<td>17.86</td>
<td>17.86</td>
<td>23.33</td>
</tr>
<tr>
<td>Garbage Outputs</td>
<td>46</td>
<td>44</td>
<td>40</td>
<td>28</td>
<td>28</td>
<td>34</td>
<td>22</td>
<td></td>
<td>52.17</td>
<td>50.00</td>
<td>45.00</td>
<td>21.43</td>
<td>21.43</td>
</tr>
</tbody>
</table>

| * 1 is the design in [59], * 2 is the design in [60], * 3 is the design in [44] |
| * 4 is the design in [61], 5 is the design 2 in [61], 6 is the design in [63], * 7 is the proposed design |

The existing designs of reversible multipliers available in the literature are limited to the designs of 4x4 reversible multiplier designs. Thus, in this work, we have shown the comparison of 4x4 reversible multiplier using the proposed design methodology. The analysis of a 4x4 reversible multiplier is performed by analyzing the required number of ancilla inputs.
and generated garbage outputs at each step. In the following steps, we are illustrating the
design cost of our proposed 4x4 reversible multiplier:

- **Step 1:** As shown in Fig. 6.6, the partial product generation circuit is implemented
  by Toffoli gates. The implementation of reversible partial product generation circuit
  using Toffoli gates require $4^2 = 16$ Toffoli gates. The number of ancilla inputs required
  in this step are $AN = 4^2 = 16$ and the total number of garbage outputs generated by
  the reversible partial product generation circuit is $GO = 2 \times 4 = 8$.

- **Step 2:** In this step, the addition of 4 partial products are performed in parallel by using
  the reversible ripple adders with input carry. As illustrated earlier, a 4x4 reversible
  multiplier design can be further divided into two ($l = \log_2 4 = 2$) stages.

  - **Stage I ($l = 1$):** The first stage requires two 4-bit reversible ripple adders with
    input carry working in parallel. The number of ancilla inputs required at this
    stage can be computed as $AN = (4/2^1 \times (2^1 - 1 + 1)) = 4$. The number of garbage
    outputs at the first stage can be computed as $GO = (4/2^1 \times (4 + 2^1 - 1 + 1 - 2)) = 8$.

  - **Stage II ($l = 2$):** The second stage of 4x4 reversible multiplier requires a 6-bit
    reversible ripple adder with input carry to perform the addition of partial sums
    generated by the first stage. The number of ancilla inputs required at this
    stage can be computed as $AN = (4/2^2 \times (2^2 - 1 + 1)) = 3$. The number of garbage outputs
    generated at this stage can be computed as $GO = (4/2^2 \times (4 + 2^2 - 1 + 2 - 2)) = 6$.

The total number of ancilla inputs required for designing a 4x4 reversible multiplier using
the proposed binary tree-based design methodology can be computed as $AN_{4x4} = Number of
ancilla inputs required by the reversible partial product generation circuit + number of ancilla
inputs required at Stage I + number of ancilla inputs required at Stage II = 16 + 4 + 3 = 23$. The total number of garbage outputs generated by the design of 4x4 reversible multiplier
using the proposed design methodology can be computed as $GO_{4x4} = \text{Number of garbage outputs generated by the reversible partial product generation circuit} + \text{number of garbage outputs generated by the Stage I} + \text{number of garbage outputs generated by the Stage II} = 8+8+6 = 22$.

The comparison study of 4x4 reversible multiplier using the proposed design methodology is shown in Table 6.1. The proposed design methodology shows significant improvement in terms of ancilla inputs and garbage outputs compared to the existing designs of reversible multiplier designs [59, 60, 61, 63].

The design of a 4x4 reversible multiplier using the proposed design methodology shows an improvement of (17.86 to 60.34)% in terms of ancilla inputs and (21.43 to 52.17)% improvement in terms of garbage outputs compared to the existing designs of reversible multipliers [59, 60, 61, 63].

### 6.3.1 Comparison Study with Reversible Array Multipliers

As illustrated earlier in the existing array multiplier-based design of $N\times N$ reversible multiplier, the number of ancilla inputs can be computed as $AN = (N + (N^2 - 2 \times N))$ and the number of garbage outputs can be computed as $GO = (N + 2(N^2 - 2 \times N))$. Table 6.2 provides the comparison of existing array multiplier-based design of reversible multiplier with the proposed binary tree-based reversible multiplier in terms of the number of ancilla inputs and garbage outputs for various values of $N$. For comparison purposes, we have used the same reversible partial product generation circuit implemented using the Toffoli gates as we used in the proposed tree-based reversible multiplier design. The proposed design methodology shows significant improvements in terms of ancilla inputs and garbage outputs compared to the reversible array multiplier design [33]. The design of a reversible multiplier using the proposed design methodology shows an improvement of (17.86 to 49.68)% in terms of ancilla inputs and (21.43 to 49.68)% improvement in terms of garbage outputs compared
to the existing reversible array multipliers [33]. It is to be noted that the existing reversible array multiplier design in [33] is a 5x5-bit design, to have the comparison with the proposed methodology, we have scaled it for a NxN reversible array multiplier design. A plot of the % improvement in terms of the number of ancilla inputs and % improvement in terms of the number of garbage outputs is shown in Fig. 6.7 for various values of N.

Table 6.2. A Comparison of NxN Reversible Multipliers

<table>
<thead>
<tr>
<th>NxN</th>
<th>Ancilla Inputs</th>
<th>Garbage Outputs</th>
<th>Ancilla Inputs</th>
<th>Garbage Outputs</th>
<th>Ancilla Inputs</th>
<th>Garbage Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4</td>
<td>28</td>
<td>28</td>
<td>23</td>
<td>22</td>
<td>17.86</td>
<td>21.43</td>
</tr>
<tr>
<td>8x8</td>
<td>120</td>
<td>120</td>
<td>83</td>
<td>81</td>
<td>30.83</td>
<td>32.50</td>
</tr>
<tr>
<td>16x16</td>
<td>496</td>
<td>496</td>
<td>303</td>
<td>300</td>
<td>38.91</td>
<td>39.52</td>
</tr>
<tr>
<td>32x32</td>
<td>2016</td>
<td>2016</td>
<td>1135</td>
<td>1131</td>
<td>43.70</td>
<td>43.90</td>
</tr>
<tr>
<td>64x64</td>
<td>8128</td>
<td>8128</td>
<td>4351</td>
<td>4346</td>
<td>46.47</td>
<td>46.53</td>
</tr>
<tr>
<td>128x128</td>
<td>32640</td>
<td>32640</td>
<td>16959</td>
<td>16953</td>
<td>48.04</td>
<td>48.06</td>
</tr>
<tr>
<td>256x256</td>
<td>130816</td>
<td>130816</td>
<td>66815</td>
<td>66808</td>
<td>48.92</td>
<td>48.93</td>
</tr>
<tr>
<td>512x512</td>
<td>523776</td>
<td>523776</td>
<td>264959</td>
<td>264951</td>
<td>49.41</td>
<td>49.42</td>
</tr>
<tr>
<td>1024x1024</td>
<td>2096128</td>
<td>2096128</td>
<td>1054719</td>
<td>1054710</td>
<td>49.68</td>
<td>49.68</td>
</tr>
</tbody>
</table>

* 1 is the design in [33], * 2 is the proposed design

Figure 6.7. % Improvement of The Proposed Reversible Multiplier Compared to Existing Reversible Array Multiplier

77
6.4 Proposed Binary Tree-Based Design Methodology for Signed NxN Reversible Multiplier

In the existing literature, researchers have proposed several signed reversible multiplier designs based on Wallace tree and modified Baugh-Wooley multiplication design styles [64, 33]. In this work, we have proposed the binary tree-based design methodology for signed reversible multiplier based on modified Baugh-Wooley multiplication algorithm. The modified Baugh-Wooley multiplication is a widely used multiplication technique for the signed binary numbers. The modified Baugh-Wooley multiplication algorithm uses the 2’s complement representation for the multiplication of the signed numbers. The partial product generation of NxN-bit signed multiplication using the modified Baugh-Wooley algorithm consists of three steps: 1) the most significant bit (MSB) of the first N-1 partial products are inverted or complemented; 2) the last partial product or the Nth partial product has all the bits inverted or complemented except the most significant bit (MSB); 3) finally two '1' are added to the Nth column and 2N-1 column of the partial products.

An example of a 4x4 bit 2's complement signed multiplication using modified Baugh-Wooley methodology is shown in Fig. 6.8(a), where \( x_3 \) and \( y_3 \) are the signed bits. To apply our binary tree-based design methodology, we have rearranged the partial products generated from the modified Baugh-Wooley multiplier as shown in Fig. 6.8(b). Fig. 6.8(b) also shows how the partial products will be regrouped together for addition using 4-bit reversible ripple carry adders and their final addition using the reversible ripple carry adders to generate the final multiplication result.

For the proposed signed NxN reversible multiplier, the reversible partial product generation circuitry can be implemented by using the Toffoli gates to reduce the number of ancilla inputs and garbage bits. The reversible partial product generation circuitry for the proposed binary tree-based signed NxN reversible multiplier is shown in Fig. 6.9. As illustrated in
Fig. 6.9, a signed NxN reversible multiplier requires \((N)^2\) Toffoli gates to generate \(N\) partial products. The binary tree-based design methodology for a signed NxN reversible multiplier requires \(l = \log_2 N\) levels, where at each level the summation of adjacent partial products or partial sums is performed in parallel. The summation of adjacent partial products is performed by using the reversible ripple carry adders.

The numbers and the widths of reversible ripple carry adders required at each level of the binary tree are summarized as follows:

- **Number of adders**: At each level of binary tree-based signed NxN reversible multiplier design, the number of reversible ripple carry adders required can be computed as \(N/2^l\), where \(l\) represents the respective level.

- **Width of adders**:
  - For levels \(l = 1\) to \((\log_2 N - 1)\) the proposed binary tree-based design methodology for signed NxN reversible multiplier requires \(N/2^l - 1\) number of \(N+l+2^{l-1} - 2\)-bit reversible ripple carry adders and one \(N+l+2^{l-1} - 1\)-bit reversible ripple carry adder.
  - At the last level \(l = \log_2 N\), a \(N+l+2^{l-1} - 1\)-bit reversible ripple carry adder is required for the addition of partial sums that generates the final multiplication result.

### 6.5 Design Example of a Signed 5x5 Reversible Multiplier Based on Proposed Design Methodology

In the existing literature [64, 33], the designs of signed reversible multiplier are of 5x5 bits. Thus, to have the comparison of the proposed design methodology with the existing
\[ x_3 \quad x_2 \quad x_1 \quad x_0 \\
\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
y_3 \quad y_2 \quad y_1 \quad y_0 \]

\begin{align*}
(a) & \quad \text{Partial Products} \\
& \quad \{ \cdots \}
\end{align*}

\begin{align*}
& x_3y_0 \quad x_2y_0 \quad x_1y_0 \quad x_0y_0, \\
& x_3y_1 \quad x_2y_1 \quad x_1y_1 \quad x_0y_1, \\
& \text{X} \\
& x_3y_2 \quad x_2y_2 \quad x_1y_2 \quad x_0y_2, \quad \text{X} \\
& x_3y_3 \quad x_2y_3 \quad x_1y_3 \quad x_0y_3.
\end{align*}

\begin{align*}
(b) & \quad \text{Partial Products} \\
& \quad \{ \cdots \}
\end{align*}

\begin{align*}
& 1 \\
& \text{X} \\
& 1
\end{align*}

\begin{align*}
& \text{Partial Sums} \\
& \{ \cdots \}
\end{align*}

\begin{align*}
& c_{10} \quad p_{s_{13}} \quad p_{s_{12}} \quad p_{s_{11}} \quad p_{s_{10}} \quad x_0y_0, \\
& 1 \quad x_3y_3 \quad x_2y_3 \quad x_1y_3 \quad x_0y_3.
\end{align*}

\begin{align*}
& P_9 \quad P_8 \quad P_7 \quad P_6 \quad P_5 \quad P_4 \quad P_3 \quad P_2 \quad P_1 \quad P_0
\end{align*}

Figure 6.8. Addition of Partial Products for a Signed 4x4 Multiplication (c10,c20: Carry Outs)

Designs, the design of signed 5x5 reversible multiplier based on the proposed binary tree-based design methodology is illustrated. The signed 5x5 bit Baugh-Wooley multiplication strategy is illustrated in Fig. 6.10(a). Figure 6.10(a) also illustrates that the signed 5x5 reversible multiplication requires 5 partial products to be generated along with the addition of two 1’s at the 5th and 9th column of partial products using the modified Baugh-Wooley multiplication. We are illustrating the grouping of the partial products or the partial sums using the proposed binary tree-based design methodology in Figs. 6.10(b), (c), (d), so that the reversible carry adders can be used to minimize the number of ancilla and the garbage bits.

The binary tree representation of the signed 5x5 reversible multiplier based on proposed methodology is illustrated in Fig. 6.11. Fig. 6.11 illustrates that the proposed binary tree-
based design methodology for the signed 5x5 reversible multiplication requires $l = \lceil \log_5 \rceil = 3$ levels. The complete design is illustrated in Fig. 6.12 using the reversible ripple adders. As it can be seen in Fig. 6.12, the design of signed 5x5 reversible multiplier requires 3 levels where: 1) level 1 requires two 5-bit reversible ripple carry adders; 2) level 2 requires a 7-bit reversible ripple carry adder; 3) level 3 requires a 6-bit reversible ripple carry adder. The partial product generation circuitry for signed 5x5 reversible multiplier is shown in Fig. 6.13. The equivalent reversible gate representation of the complete design is illustrated in Fig. 6.14 using the reversible ripple carry adders with input carry.
As illustrated earlier, the proposed binary tree-based design methodology for the signed 5x5 reversible multiplier initially requires the partial products to be generated and later the summation of partial products are performed in parallel using the reversible ripple adders. The partial product generation circuitry for a signed 5x5 reversible multiplier is shown in Fig. 6.12. In Fig. 6.12, it can be seen that the partial products can be generated using the Toffoli gates, and it requires $N^2 = 5^2 = 25$ number of Toffoli gates. The reversible partial product generation circuitry requires $N^2 = 5^2 = 25$ ancilla inputs and generates $2 \times N = 2 \times 5 = 10$ number of garbage outputs. The summation of partial products using

\[
\begin{array}{cccccc}
X & x_4 & x_3 & x_2 & x_1 & x_0 \\
y_4 & y_3 & y_2 & y_1 & y_0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\bar{x}_4\bar{y}_0 & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
\bar{x}_4\bar{y}_1 & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 \\
x_4y_2 & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 \\
x_4y_3 & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 \\
x_4y_4 & x_3y_4 & x_2y_4 & x_1y_4 & x_0y_4 \\
\end{array}
\]

(a)

\[
\begin{array}{cccccc}
\bar{x}_4\bar{y}_0 & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
\bar{x}_4\bar{y}_1 & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 \\
x_4y_2 & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 \\
x_4y_3 & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 \\
x_4y_4 & x_3y_4 & x_2y_4 & x_1y_4 & x_0y_4 \\
\end{array}
\]

(b)

\[
\begin{array}{cccccc}
\bar{x}_4\bar{y}_0 & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
\bar{x}_4\bar{y}_1 & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 \\
x_4y_2 & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 \\
x_4y_3 & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 \\
x_4y_4 & x_3y_4 & x_2y_4 & x_1y_4 & x_0y_4 \\
\end{array}
\]

(c)

\[
\begin{array}{cccccc}
\bar{x}_4\bar{y}_0 & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
\bar{x}_4\bar{y}_1 & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 \\
x_4y_2 & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 \\
x_4y_3 & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 \\
x_4y_4 & x_3y_4 & x_2y_4 & x_1y_4 & x_0y_4 \\
\end{array}
\]

(d)
the reversible ripple adders is shown in Fig. 6.12. The design of a signed 5x5 reversible multiplier requires $l = \lceil \log_5 \rceil = 3$ levels. This step can further be divided into three stages, 1) Stage I requires two 5-bits reversible ripple carry adders for the addition of four partial products in parallel and generates 2 partial sums as shown in Fig. 6.12. This stage requires 4 ancilla inputs and generates 10 garbage outputs. 2) stage II requires one 7-bit reversible ripple carry adder for the addition partial sums generated in the previous stage. This stage requires 3 ancilla inputs and generates 7 garbage outputs as shown in Fig. 6.12. 3) stage III requires a 6-bit reversible ripple carry adder for the addition of partial product and partial sum and generates the final multiplication result as shown in Fig. 6.12. This stage requires 2 ancilla inputs and generates 6 garbage outputs.

In the existing literature, researchers have proposed several designs of reversible ripple adders [46, 56, 44, 58, 62]. In this work, we have used the design of reversible ripple adder with input carry proposed in [46], as it requires zero ancilla inputs and produces zero garbage
Figure 6.12. Proposed Signed 5X5 Reversible Multiplier Design Using Reversible Ripple Adder (P0 to P10: Final Result; G0 to G22: Garbage Outputs)

Figure 6.13. A Partial Product Generation Circuit for Signed 5x5 Reversible Multiplier (G0 to G9: Garbage Outputs; p0 to p24: Partial Products)
Figure 6.14. A Signed 5x5 Reversible Multiplier Design Using Reversible Ripple Carry Adders with Input Carry ($P_0$ to $P_{10}$: Final Result; $G_{10}$ to $G_{32}$: Garbage Outputs; $p_0$ to $p_{24}$: Partial Products)
outputs. The design of a signed 5x5 reversible multiplier using reversible ripple adder with input carry is shown in Fig. 6.14. The partial products required for the signed 5x5 reversible multiplier are generated by the reversible partial product generation circuit and is shown in Fig. 6.13. The design of a signed 5x5 reversible multiplier as shown in Fig. 6.14 requires 2 reversible ripple carry adders with input carry at the first stage, one reversible ripple carry adder with input carry in the second stage, and in the third stage it requires another reversible ripple carry adder with input carry. The proposed design of signed 5x5 reversible multiplier using the reversible ripple carry adders with input carry significantly reduces the number of ancilla inputs and garbage outputs compared to the existing designs of signed reversible multipliers.

6.5.1 Comparison of Proposed Signed 5x5 Reversible Multiplier Design with Existing Designs

In the existing literature, the researchers have proposed several designs of signed reversible multipliers [64, 33]. The existing design of signed reversible multipliers are limited to the design of signed 5x5 reversible multipliers. In this work, we proposed a binary tree-based design methodology for signed NxN reversible multiplier. The analysis of the proposed signed 5x5 reversible multiplier shown in Fig. 6.14 is performed by analyzing the number of ancilla inputs and number of garbage outputs generated at each step of the proposed design methodology.

In the proposed design, the total number of ancilla input required to design a signed 5x5 reversible multiplier using the proposed design methodology can be computed as $AN_{5x5\text{signed}}$

$= \text{Number of ancilla inputs required by the reversible partial product generation circuitry} + \text{Number of ancilla inputs required for the summation of partial products using the reversible ripple adders (Stage I + Stage II + Stage III)} = 25 + 4 + 3 + 2 = 34$. The total number of garbage outputs generated to design a signed 5x5 reversible multiplier using the proposed
design methodology can be computed as \( GO_{5 \times 5 \text{ signed}} = \text{Number of garbage outputs generated by the reversible partial product generation circuitry} + \text{Number of garbage outputs generated by the summation of partial products using the reversible ripple adders (Stage I + Stage II + Stage III)} = 10 + 10 + 7 + 6 = 33. \)

Table 6.3. A Comparison of Signed Reversible Multipliers

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>% Imp. w.r.t 1</th>
<th>% Imp. w.r.t 2</th>
<th>% Imp. w.r.t 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancilla Inputs</td>
<td>56</td>
<td>46</td>
<td>70</td>
<td>34</td>
<td>39.28</td>
<td>26.09</td>
<td>51.43</td>
</tr>
<tr>
<td>Garbage Outputs</td>
<td>56</td>
<td>46</td>
<td>70</td>
<td>33</td>
<td>41.07</td>
<td>28.26</td>
<td>52.86</td>
</tr>
</tbody>
</table>

\* 1 is the design 1 of signed 5x5 reversible multiplier in [64], * 2 is the design 2 of signed 5x5 reversible multiplier in [64], * 3 is the design of signed 5x5 reversible multiplier in [33], * 4 is the proposed design.

The comparison of the proposed signed 5x5 reversible multiplier with the existing signed reversible multipliers is shown in Table 6.3. The proposed design methodology for reversible signed multiplier shows significant improvements in terms of ancilla inputs and garbage outputs compared to the existing designs of signed reversible multipliers. The design of signed 5x5 reversible multiplier using the proposed design methodology shows an improvement of (26.09 to 51.43)% in terms of ancilla inputs and (28.26 to 52.86)% improvements in terms of garbage outputs compared to the existing designs of signed reversible multiplier [64, 33].

6.6 Comparison of Proposed Signed Reversible Multiplier Design with Existing Designs

As explained earlier, the implementation of a signed NxN reversible multiplier using the proposed binary tree-based design methodology requires \( l = \log_2 N \) stages. At each stage, the number of ancilla inputs can be computed as \( AN = (N/2^l \ast (2^{l-1} + 1)) + 2 \) for \( l = 1 \) and \( AN = (N/2^l \ast (2^{l-1} + 1)) + 1 \) for \( l = 2, 3, 4, \ldots, \log_2 N \), and the number of garbage outputs can be computed as \( GO = (N/2^l \ast (N + 2^{l-1} + l - 2)) + 1 \) where \( l = 1, 2, 3, \ldots, \log_2 N \) represents...
the respective level (These generalized formulas are derived for the values of N where N is the power of 2).

In the existing literature, researchers have proposed the design of signed reversible multiplier based on Baugh-Wooley multiplication [64]; however, the proposed design is limited to the design of a signed 5x5 reversible multiplier. The existing design of the signed reversible multiplier proposed in [64] is limited to the design of a signed 5x5 reversible multiplier. To have the comparison with the proposed methodology, we have to scale the design for a signed NxN reversible array multiplier design. As illustrated earlier throughout this work, the generation of partial products using the reversible partial product generation circuitry remains the same for all the signed reversible multiplier designs. A signed NxN reversible partial product generation circuitry requires \(N^2\) ancilla inputs and generates \(2 \times N\) garbage outputs as shown in Fig. 6.9. To perform the array multiplier-based addition of partial products using the modified Baugh-Wooley multiplication, it requires an \(N\) reversible half adders and \((N^2 - 2 \times N + 1)\) reversible full adders. A Peres gate can work as a reversible half adder with one ancilla input and one garbage output. The design of reversible full adder will need one ancilla input and two garbage outputs. Thus, to design a signed NxN reversible array multiplier based on Baugh-Wooley multiplication requires \(AN = N + N^2 - 2 \times N + 1\) number of ancilla inputs and \(GO = N + 2 \times (N^2 - 2 \times N + 1)\) number of garbage outputs. Table 6.4 shows the comparison of the proposed binary tree-based design methodology for signed NxN reversible multiplier with the existing modified Baugh-Wooley-based signed NxN reversible multiplier in terms of the number of ancilla inputs and the number of garbage outputs for various values of N. The implemented reversible partial product generation circuitry using the Toffoli gates is used for partial product generation in both the signed reversible multipliers for comparison purposes. The proposed design methodology shows significant improvement in terms of ancilla inputs and garbage outputs compared to the existing design of signed reversible multiplier based on Baugh-Wooley multiplication [64]. The design of signed
reversible multiplier based on the proposed design methodology shows an improvement of (10.34 to 49.68) % in terms of ancilla inputs and (20 to 49.68) % improvement in terms of garbage outputs compared to the existing designs of signed reversible array multipliers [64]. Figure 6.15 shows a plot of % improvement in terms of the number of ancilla inputs and % improvement in terms of the number of garbage outputs for various values of N.

Table 6.4. A Comparison of Signed NxN Reversible Multiplier

<table>
<thead>
<tr>
<th>NxN</th>
<th>Ancilla Inputs</th>
<th>Garbage Outputs</th>
<th>% Imp. w.r.t 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4</td>
<td>29</td>
<td>30</td>
<td>10.34</td>
</tr>
<tr>
<td>8x8</td>
<td>121</td>
<td>122</td>
<td>28.10</td>
</tr>
<tr>
<td>16x16</td>
<td>497</td>
<td>498</td>
<td>38.03</td>
</tr>
<tr>
<td>32x32</td>
<td>2017</td>
<td>2018</td>
<td>43.43</td>
</tr>
<tr>
<td>64x64</td>
<td>8129</td>
<td>8130</td>
<td>46.39</td>
</tr>
<tr>
<td>128x128</td>
<td>32641</td>
<td>32642</td>
<td>48.02</td>
</tr>
<tr>
<td>256x256</td>
<td>130817</td>
<td>130818</td>
<td>48.92</td>
</tr>
<tr>
<td>512x512</td>
<td>523777</td>
<td>523778</td>
<td>49.41</td>
</tr>
<tr>
<td>1024x1024</td>
<td>2096129</td>
<td>2096130</td>
<td>49.68</td>
</tr>
</tbody>
</table>

* 1 is the design in [64], * 2 is the proposed design

Figure 6.15. Improvement of The Proposed Signed Reversible Multiplier Compared to Existing Signed Reversible Array Multiplier
6.7 Conclusion

In this work, we have proposed a binary tree-based design methodology for NxN reversible multiplier for unsigned as well as signed multiplication. The proposed design methodology performs the addition of partial products in parallel by using the reversible ripple adders with zero ancilla and garbage bits. The unique binary tree approach coupled with the use of reversible ripple adders having no ancilla and garbage bits reduces the number of ancilla and garbage bits in the proposed reversible multiplier design. The proposed binary tree-based design methodology shows significant advantages in terms of number of ancilla and garbage bits compared to existing designs, as well as, array and Baugh-Wooley multiplication-based design styles. The proposed design methodology can be integrated as a part of an optimized reversible alu in terms of ancilla and garbage bits.
CHAPTER 7
LIMITATIONS AND CONCLUSION

Reversible logic has promising application in the quantum computing, optical computing and quantum dot cellular automata computing. Reversible arithmetic units such as adders, subtractors, multipliers which form the essential component of a computing system have also been designed in binary as well as ternary logic such as in [53, 54, 55, 65, 66, 67, 60, 64]. In [44], researchers have designed the reversible ripple carry adder having no input carry with one ancilla input bit. In [56, 58], the researchers have investigated new designs of the reversible ripple carry adder with no ancilla input bit and improved delay. A comprehensive survey of reversible arithmetic circuits can be found in [62]. The design of BCD adders and subtractors have also been attempted. The researchers have investigated the design of BCD adders and subtractors in which parameters such as the number of reversible gates, garbage outputs, quantum cost, transistors, etc. are considered for optimization [68, 55, 66, 67, 69].

In recent years, researchers have emphasized finding reversible logic applications in promising emerging technologies. In past, several emerging technologies have been investigated for design exploration and application of reversible logic. In this work, we have explored design and implementing of reversible logic circuits for quantum computing as well as optical computing. The designs and methodologies of reversible circuits proposed in this dissertation show an improvement towards the advancement of reversible circuit design for quantum as well as optical computing. In this dissertation, we have explored the design methodologies for efficient reversible circuits considering the design metrics of ancilla inputs, garbage out-
puts and quantum cost. Some of the future direction to improve this dissertation work and some interesting research ideas are listed as follows:

- The field of linear optical quantum computing explored in this dissertation work is limited to the design of an efficient reversible NOR logic based gates. It would be interesting to investigate further in this direction of synthesizing reversible circuits by using a combination of proposed optical NOR gates and existing optical NAND gates.

- The design of reversible multiplier proposed in this work is limited to integer multiplication, designing efficient reversible floating point multipliers would be an interesting field to investigate.

- Investigation in the field of basic reversible logic gates for more optimized parameters such as ancilla inputs, garbage outputs and quantum or optical cost.

- Development of reversible logic circuits for floating point arithmetic.

- There is less work in the direction of reversible logic-based memory units, such as registers, SRAM, etc.

- Development of reversible computer architecture, reversible CPU and reversible instruction set architecture.

- Investigation in the field of testing of a reversible circuit in the quantum space.

- Development of the synthesis tools for reversible circuits.
REFERENCES


 modes of the reversible josephson-junction logic circuits. *IEEE Transactions on Applied 

 Progress towards reversible computing with nsquid arrays. *IEEE Transactions on Ap-

 786, June 2011.

[12] S Frost-Murphy, M Ottavi, M Frank, and E DeBenedictis. On the design of reversible 

[13] Natalie Kostinski, Mable P. Fok, and Paul R. Prucnal. Experimental demonstration of 

 application to qca. In *Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on*, 


 W.Haensch. Practical strategies for power-efficient computing technologies. *Proc. of the 


APPENDIX A: COPYRIGHT PERMISSIONS

This copyright permission is for the paper published in the Design Automation Test in Europe Conference (DATE), 2012 [31]. Portions of this paper appear in Chapter 3.

Title: Mach-Zehnder interferometer based design of all optical reversible binary adder
Conference: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012
Author: S. Kotiyal; H. Thapliyal; N. Ranganathan
Publisher: IEEE
Date: 12-16 March 2012
Copyright © 2012, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication].
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to [http://www.ieee.org/publications_standards/publications/rights/rights_link.html] to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
This copyright permission is for the paper published in the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2012 [47]. Portions of this paper appear in Chapter 4.
This copyright permission is for the paper published in the Microelectronics Journal, 2014 [48]. Portions of this paper appear in Chapter 4.
This copyright permission is for the paper published in the Transactions on Computational Science XXIV: Special Issue on Reversible Computing, 2014 [50]. Portions of this paper appear in Chapter 5.

<table>
<thead>
<tr>
<th>Title:</th>
<th>Design of Reversible Adder-Subtractor and its Mapping in Optical Computing Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author:</td>
<td>Saurabh Kotiyal</td>
</tr>
<tr>
<td>Publication:</td>
<td>Springer eBook</td>
</tr>
<tr>
<td>Publisher:</td>
<td>Springer</td>
</tr>
<tr>
<td>Date:</td>
<td>Jan 1, 2014</td>
</tr>
</tbody>
</table>

Thank you very much for your order.

This is a License Agreement between Saurabh Kotiyal ("You") and Springer ("Springer"). The license consists of your order details, the terms and conditions provided by Springer, and the payment terms and conditions.

Get the printable license.

<table>
<thead>
<tr>
<th>License Number</th>
<th>3838690046863</th>
</tr>
</thead>
<tbody>
<tr>
<td>License date</td>
<td>Mar 30, 2016</td>
</tr>
<tr>
<td>Licensed content</td>
<td>Design of Reversible Adder-Subtractor and its Mapping in Optical Computing Domain</td>
</tr>
<tr>
<td>publisher</td>
<td>Springer</td>
</tr>
<tr>
<td>Licensed content</td>
<td>Design of Reversible Adder-Subtractor and its Mapping in Optical Computing Domain</td>
</tr>
<tr>
<td>publication</td>
<td>Springer</td>
</tr>
<tr>
<td>Licensed content date</td>
<td>Jan 1, 2014</td>
</tr>
<tr>
<td>Type of Use</td>
<td>Thesis/Dissertation</td>
</tr>
<tr>
<td>Portion</td>
<td>Full text</td>
</tr>
<tr>
<td>Number of copies</td>
<td>1</td>
</tr>
<tr>
<td>Author of this Springer article</td>
<td>Yes and you are a contributor of the new work</td>
</tr>
<tr>
<td>Title of your thesis / dissertation</td>
<td>Design Exploration and Application of Reversible Circuits in Emerging Technologies</td>
</tr>
<tr>
<td>Expected completion date</td>
<td>May 2016</td>
</tr>
<tr>
<td>Estimated size(pages)</td>
<td>120</td>
</tr>
<tr>
<td>Total</td>
<td>0.00 USD</td>
</tr>
</tbody>
</table>

Copyright © 2016 Copyright Clearance Center, Inc. All Rights Reserved.

Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Rightslink® by Copyright Clearance Center 
https://s100.copyright.com/AppDispatchServlet
This copyright permission is for the paper published in the 27th International Conference on VLSI Design and 13th International Conference on Embedded Systems, 2014 [51]. Portions of this paper appear in Chapter 6.
This copyright permission is for the paper published in the Journal of Supercomputing, 2015 [52]. Portions of this paper appear in Chapter 6.
ABOUT THE AUTHOR

Saurabh Kotiyal received the B.E. degree in computer Science from the University of Rajasthan, Jaipur, India, in 2006, and the Master’s degree in computer engineering from the University of South Florida, Tampa, in 2012. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering, University of South Florida, Tampa. He was a Software Engineer with 3DPLM Software Solution Ltd., Mumbai, India, from 2007 to 2009. His current research interests include reversible logic, quantum computing and high-speed low-power VLSI arithmetic and algorithms. He has authored or coauthored more than 11 articles in refereed conferences and journals. He was a recipient of 2013 Upsilon Pi Epsilon Award and 2011 Richard E. Merwin Student Scholarship by IEEE Computer Society for demonstrating outstanding involvement in an IEEE Computer Society Student Chapter, excellence in academic achievement, and promise in future efforts.