EFFICIENT IMPLEMENTATION OF MULTI-CONTROL TOFFOLI GATES IN LINEAR NEAREST NEIGHBOR ARRAYS

A Thesis by

Saman Daraeizadeh

Bachelor of Science, Zanjan University, Iran, 2007

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

May 2014

EFFICIENT IMPLEMENTATION OF MULTI-CONTROL TOFFOLI GATES IN LINEAR NEAREST NEIGHBOR ARRAYS

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science, with a major in Electrical Engineering.

____

Preethika Kumar, Committee Chair

____

Steven R. Skinner, Committee Member

____

James E. Steck, Committee Member

DEDICATION

To my wife, my daughter, my parents, my brother
and my sister

For there are these three things that endure: Faith, Hope and Love,
but the greatest of these is Love

ACKNOWLEGEMENT

I would like to express my sincere gratitude to my advisor and teacher, Dr. Preethika Kumar, without whom this would not be possible. Also, I would like to extend my gratitude to members of my committee, Dr. Steven R. Skinner and Dr. James E. Steck, for their helpful comments and suggestions.

My deepest expression of appreciation goes to my wife, Afsaneh. Her love, patience, support and motivation have made it possible for me to pursue this degree. I would also like to thank my mother for her prayers, my father for his support, my sister for her love and my brother for his kindness. Special thanks to my dear friends who have been with me during happiness and difficulties.

Above all, praise be to God, who has not rejected my prayers or withheld His love from me. I would like to thank Him for blessing me much more than I deserve.

ABSTRACT

Most promising implementations in quantum computing are based on Linear Nearest Neighbor (LNN) architectures, where qubits only interact with neighbors. Multi-control Toffoli gates are used in many quantum applications such as error correction and algorithms like Shor's factorization. Typically, to implement a multi-control Toffoli gate in an LNN architecture, additional operations called swap gates are required to bring the qubits adjacent to each other. This may increase the total number of quantum gates and computational overhead of the circuit. Here, we propose a new method to implement multi-control Toffoli gates in LNN arrays without using swap gates. The circuit reduction techniques discussed here are based on 3 lemmas. Using the lemmas, we show how to implement multi-control Toffoli gates in LNN arrays with different separations between the control and target qubits. The key feature of our scheme is to involve qubits other than control and target qubits to take part in gate operations. We call these qubits "auxiliary" qubits, and they are used in our gate decomposition protocols. Auxiliary qubits can be in any arbitrary states, $\alpha|0\ \ +\beta|1$, and are always restored back to their original states. Since we do not use swap gates to bring qubits adjacent to each other, compared to circuits using swap gates, the total number of gate operations used in our method is decreased, and the quantum cost is lowered.

In addition, for implementing multi-control Toffoli gate operations efficiently in LNN arrays, we also show how to extend our protocols to 2D arrays. Here, in addition to translating our gate reduction techniques directly from 1D to 2D, we use further simplification techniques for particular arrangements of qubits.

TABLE OF CONTENTS

Chapter Page

1. INTRODUCTION.. 1

1.1 Background..1
1.2 Quantum Gates... 3
1.2.1 Hadamard Gate.. 3
1.2.2 Pauli-X (or NOT) Gate... 4
1.2.3 Pauli-Y (or Y) Gate.. 4
1.2.4 Pauli-Z (or Z) Gate.. 5
1.2.5 Phase Shift Gates.. 5
1.2.6 Controlled-Unitary Gate... 6
1.2.7 Toffoli Gate.. 8
1.2.8 Swap Gate...10
1.3 Universal Gate Sets..10
1.4 LNN Architectures..11
1.5 Quantum Error Correction...12
1.6 Our Research... 13

2. EFFICIENT IMPLEMENTATION OF MULTI-CONTROL TOFFOLI GATES IN LINEAR NEAREST NEIGHBOR ARRAYS...15

2.1 Abstract..15
2.2 Keywords..15
2.3 Introduction...15
2.4 Definitions and Lemmas..19
2.4.1 Definitions.. 19
2.4.2 Lemmas...21
2.5 Implementing Toffoli Gates Between Non-Adjacent Qubits in Multi-Qubit Systems...25
2.5.1 Toffoli Gate in a 4 Qubit System.. 25
2.5.2 Toffoli Gate in an N Qubit System...26
2.6 Multi-Control Toffoli Gates... 28
2.6.1 Multi-Control Toffoli Gates in N Qubit Systems.......................................28
2.6.2 A $C^3(X)$ Gate in a 5 Qubit System.. 30
2.7 Multi-Control-Multi-Target Toffoli Gates... 30
2.7.1 Multi-Control-Multi-Target Toffoli Gates in N Qubit Systems................30
2.7.2 Multi-Control-Multi-Target Toffoli Gate in a 5 Qubit System.................31
2.8 Simulations... 32
2.9 Multi-Control Toffoli Gates in 2D Architectures...36
2.10 Conclusions..37
2.11 References...38

TABLE OF CONTENTS (continued)

Chapter Page

3. SINGLE-CONTROL-MULTI-TARGET TOFFOLI GATES IN LNN ARRAYS...........42

4. CONCLUSIONS.. 45

REFERENCES... 47

LIST OF TABLES

Table Page

1.1. State Table of the CNOT Gate....................7

1.2. State Table of the $C^1(Z)$ Gate…….................... 8

1.3. State Table of the Toffoli Gate.................... 9

1.4. State Table of the Swap Gate....................10

2.1. State Table of Figure 2.1.................... 22

LIST OF FIGURES

Figure Page

1.1 Circuit Diagram of the Hadamard Gate Applied on a Qubit…….. 4

1.2 Circuit Diagram of the NOT Gate Applied on a Qubit..4

1.3 Circuit Diagram of the Y Gate Applied on a Qubit.. 5

1.4 Circuit Diagram of the Z Gate Applied on a Qubit.. 5

1.5 Circuit Diagram of the Phase Shift Gate Applied on a Qubit..5

1.6 Circuit Diagram of the CNOT Gate in a 2 Qubit System...6

1.7 Circuit Diagram of the $C^1(Z)$ Gate in a 2 Qubit System.. 7

1.8 Circuit Diagram of the Toffoli Gate in a 3 Qubit System.. 9

1.9 Circuit Diagram of the Swap Gate in a 2 Qubit System.. 10

1.10 Encoding Code for the 5 Qubit Qubit System...13

2.1 Implementation of a Toffoli Gate in a 3 Qubit System..21

2.2 Decomposition of a $C^2(Z)$ Gate into Two $C^2(X)$ Gates and Two $C^1(Z)$ Gates..............24

2.3 Implementation of a Toffoli Gate in a 4 Qubit System..26

2.4 Implementation of a Toffoli Gate in an N Qubit System (Case 1)...................................27

2.5 Implementation of a Multi-Control Toffoli Gate in an N Qubit System..........................29

2.6 Implementation of a Multi-Control Toffoli Gate in a 5 Qubit system............................. 30

2.7 Decomposition of a Multi-Control-Multi-target Toffoli Gate in an N Qubit System... 31

2.8 Implementation of a Multi-Control-Multi-target Toffoli Gate in a 5 Qubit System... 32

2.9 Simulation Results for a Toffoli Gate in a 4 Qubit System..35

LIST OF FIGURES (continued)

Figure Page

2.10 Simulation Results for a Multi-Control-Multi-Target Toffoli Gate in a 5 Qubit System.......36

2.11 A 2D Nearest Neighbor Arrangement of Qubits.......37

3.1 Implementation of a Single-Control-Multi-Target Toffoli Gate in an N Qubit System.......43

3.2 Implementation of a 1-Control- 3-Target Toffoli Gate in a 5 Qubit System.......44

4.1 A controlled-Z Gate and a controlled-S Gate With a Shared Control Qubit.......46

LIST OF ABBRETIATIONS

LNN	Linear Nearest Neighbor
NNC	Nearest Neighbor Cost
CNOT	Controlled-NOT

CHAPTER 1

INTRODUCTION

1.1 Background

In 1985, David Deutsch at the University of Oxford described the first universal quantum computer. A quantum computer is a device that makes use of quantum-mechanical phenomena like superposition and entanglement to perform operations on data [1]. Unlike digital computers, where data is encoded into binary bits (digits), a quantum computer uses qubits. The difference between bits and qubits is that whereas a bit must be either 0 or 1, a qubit can be 0, 1, or a superposition of both. That is, a qubit can be in one of the two basis states, $|0\rangle$ or $|1\rangle$, or a linear superposition of the two states, where $|0\rangle$ and $|1\rangle$ are:

$$|0\rangle = \binom{1}{0};\ |1\rangle = \binom{0}{1} \tag{1.1}$$

The two basis states, $|0\rangle$ and $|1\rangle$, are orthonormal vectors. The general state of a qubit can be defined as:

$$|q\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1.2}$$

where,

$$\sqrt{|\alpha|^2 + |\beta|^2} = 1 \tag{1.3}$$

Accordingly, a qubit is a unit vector in a two-dimensional complex vector space known as the Hilbert space. A system with N qubits is described by a 2^N dimensional vector space. The simplest example of a two-state physical system is the spin state of an electron, where it can be found in state-up $|\uparrow\rangle$, state-down $|\downarrow\rangle$ or in the superposition of these two states. Other examples of qubits are the polarizations of a single photon, superconducting qubits, ion-trap qubits, nuclear magnetic resonance qubits and quantum dots [1-4].

A number of quantum computing models have been proposed, of which four are the most popular in terms of practical importance. They are:

I. Quantum Gate Array: in this model, a quantum computation is decomposed into a sequence of unitary operations realized on one/more qubits called quantum gates [43].

II. One-Way Quantum Computer: in this model, a quantum computation is decomposed into a sequence of one-qubit measurements applied to a highly entangled initial state or cluster state [48].

III. Adiabatic Quantum Computer: in this model, a quantum computation is broken down into a slow continuous transformation of an initial Hamiltonian into a final Hamiltonian, whose ground states contain the solution [49]. This model is also called "Quantum Annealing".

IV. Topological Quantum Computer: in this model, computation is decomposed into the braiding of anyons (particles which can have some quantum numbers of fractional values with respect to other elementary particles. For example, an anyon can be charged and the charge can be a fraction of $1\ e$ (electron charge)) in a 2D lattice [50].

Even though each of the 4 models have been shown to be equivalent to each other, the Quantum Gate Array (or Quantum Circuit Model) is the most popular. This model directly adapts from the classical computational model for classical computers where a discrete set of universal gates are used to implement all computations within the computer. Each logic gate transforms its input bits into one or more output bits in some deterministic fashion according to the definition of the gate. All computations within the computer are performed by arranging the gates in a sequence such that the outputs from earlier gates feed into the inputs of later gates. In classical computing, the NAND and NOR gates form single-gate universal sets, and all computations can be decomposed into a sequence of these gates. The Quantum Circuit Model for quantum computing is analogous to the classical approach of computing. Here, again, all computations within the quantum computer are broken down into a series of gate operations taken from a universal gate set. However, contrary, to gates used in classical computing, each quantum gate has to be reversible (since quantum gates are mathematically represented by “unitary matrices” which operate on qubits which are mathematically represented as “unit vectors”, whose lengths have to be preserved). We will now look at different quantum logic gates used to perform quantum operations under the Quantum Circuit Model, some of which are used for forming universal gate sets.

1.2 Quantum Gates

In quantum computing, the Hamiltonian is the operator corresponding to the total energy of the system. Every quantum system has its own Hamiltonian. Quantum gates are unitary operations generated by tuning the appropriate parameters of the system Hamiltonian [1-7]. Quantum gates are either single-qubit or multi-qubit gate operations. A single-qubit quantum gate is implemented on one qubit, and has a two-dimensional vector space. A multi-qubit quantum gate is realized on more than one qubit, and has a $2^K \times 2^K$ dimensional vector space, where “K” is the number of qubits involved. In the following, some of the common quantum gates are introduced briefly.

1.2.1 Hadamard Gate

The Hadamard gate acts on a single qubit, and it maps the basis state $|0$ to $\frac{|0 +|1}{\sqrt{2}}$ and $|1$ to $\frac{|0 -|1}{\sqrt{2}}$. It is presented by the Hadamard matrix:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.4}$$

Figure 1.1 shows the circuit diagram of the Hadamard gate. Here and throughout this thesis, the time evolution is from left to right. Suppose that the qubit is initially in the arbitrary state of $\alpha|0\ +\beta|1$. After the implementation of the Hadamard gate, the final state of the qubit will be $[\alpha/\sqrt{2}\,(|0\ +|1\ )+\beta/\sqrt{2}(|0\ -|1\ )]$.

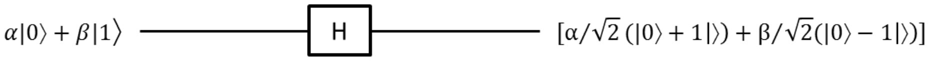
$\alpha|0\rangle+\beta|1\rangle$
H
$[\alpha/\sqrt{2}\,(|0\rangle+1|\rangle)+\beta/\sqrt{2}(|0\rangle-1|\rangle)]$

Figure 1.1 The circuit diagram of the Hadamard gate applied on an arbitrary qubit, $\alpha|0\ +\beta|1$

1.2.2 Pauli-*X* (or NOT) Gate

The Pauli-X gate acts on a single qubit, and it maps the basis state $|0$ to $|1$, and vice versa. It is presented by the Pauli-X matrix:

$$X=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \tag{1.5}$$

Figure 1.2 shows the circuit diagram of the NOT gate. Here, the initial state of the qubit is $\alpha|0\ +\beta|1$ and the final state will be $\alpha|1\ +\beta|0$.

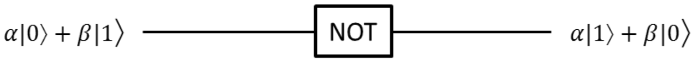
$\alpha|0\rangle+\beta|1\rangle$
NOT
$\alpha|1\rangle+\beta|0\rangle$

Figure 1.2 The circuit diagram of the NOT gate applied on an arbitrary qubit, $\alpha|0\ +\beta|1$

1.2.3 Pauli-*Y* (or *Y*) Gate

The Pauli-Y gate acts on a single qubit, and it maps $|0$ to $i|1$, and $|1$ to $-i|0$. It is presented by the Pauli-Y matrix:

$$Y=\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix} \tag{1.6}$$

Figure 1.3 shows the circuit diagram for the Y gate. If the initial state is $\alpha|0\ +\beta|1$, the final state will be $\alpha i|1\ -\beta i|0$.

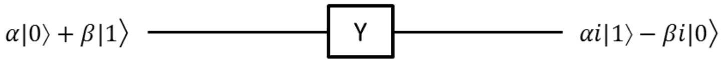
$\alpha|0\rangle + \beta|1\rangle$
Y
$\alpha i|1\rangle - \beta i|0\rangle$

Figure 1.3 The circuit diagram of the Y gate applied on an arbitrary qubit, $\alpha|0 \ + \beta|1$

1.2.4 Pauli-Z (or Z) Gate

The Pauli-Z gate acts on a single qubit, and it leaves the basis state $|0$ unchanged and maps $|1$ to $-|1$. It is presented by the Pauli-Z matrix:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1.7}$$

The circuit diagram of the Z gate is shown in the Figure 1.4. If the initial state of the qubit is $\alpha|0 \ + \beta|1$, the final state will be $\alpha|0 \ - \beta|1$.

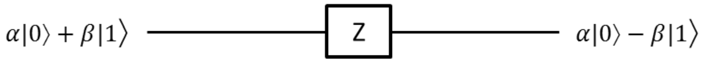
$\alpha|0\rangle + \beta|1\rangle$
Z
$\alpha|0\rangle - \beta|1\rangle$

Figure 1.4 The circuit diagram of the Z gate applied on an arbitrary qubit $\alpha|0 \ + \beta|1$

1.2.5 Phase Shift Gates

This is a family of single-qubit gates that leave the basis state $|0$ unchanged and map $|1$ to $e^{i\theta}|1$. They are represented by rotation matrices:

$$R_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{1.8}$$

where θ is the phase shift. Some common examples are T gate where $\theta = \frac{\pi}{8}$, S gate where $\theta = \frac{\pi}{4}$ and Z gate (Pauli-Z gate) where $\theta = \frac{\pi}{2}$. Figure 1.5 shows the circuit diagram of the phase shift gate. If the initial state of the qubit is $\alpha|0 \ + \beta|1$, the final state will be $\alpha|0 \ + \beta e^{i\theta}|1$.

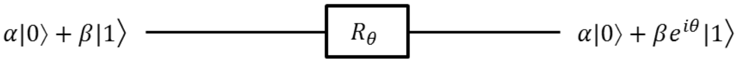
$\alpha|0\rangle + \beta|1\rangle$
R_θ
$\alpha|0\rangle + \beta e^{i\theta}|1\rangle$

Figure 1.5 The circuit diagram of the phase shift gate applied on an arbitrary qubit, $\alpha|0 \ + \beta|1$

1.2.6 Controlled-Unitary Gate

A controlled-unitary gate is a multi-qubit gate, where one or more qubits act as controls for a unitary gate operation. It operates a unitary gate on the target qubit only when all the control qubits are in the $|1\rangle$ state (a control in the $|1\rangle$ state is shown with a filled circle). Note that, in a controlled-unitary quantum gate the state of the control qubit that determines whether or not a special action is performed on the target qubit does not have to be $|1\rangle$; it can be $|0\rangle$ (or any other state) too. If it is $|0\rangle$, then the control qubit is represented as an "empty" circle (see Figure 1.10 to be discussed later)

An example of a controlled-unitary gate is the CNOT (controlled-NOT) gate, which is a two-qubit gate defined by the matrix:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.9}$$

Here, a NOT gate is applied on the target qubit only when the control is in the $|1\rangle$ state. The circuit diagram and the state table of the CNOT gate, where qubit 1 is the control and qubit 2 is the target, are shown in the Figure 1.6 and the Table 1.1, respectively.

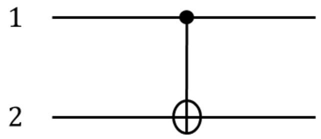

Figure 1.6 The circuit diagram of the CNOT gate where qubit 1 is the control and qubit 2 is the target

TABLE 1.1

STATE TABLE OF THE CNOT GATE

Initial state	**Final state**
$\vert 00$	$\vert 00$
$\vert 01$	$\vert 01$
$\vert 10$	$\vert 11$
$\vert 11$	$\vert 10$

Another example of a controlled-unitary gate is the controlled-Z gate, which is a two-qubit gate operation defined by the matrix:

$$C^1(Z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{1.10}$$

Under a controlled-Z gate, a phase of π is picked up by the target qubit only when both qubits are in the $|1$ state. The circuit diagram and the state table of the $C^1(Z)$ gate, where qubit 1 is the control and qubit 2 is the target, are shown in the Figure 1.7 and the Table 1.2, respectively.

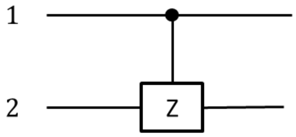

Figure 1.7 The circuit diagram of the $C^1(Z)$ gate where qubit 1 is the control and qubit 2 is the target

TABLE 1.2

STATE TABLE OF THE $C^1(Z)$ GATE

Initial state	**Final state**
$\|00$	$\|00$
$\|01$	$\|01$
$\|10$	$\|10$
$\|11$	$-\|11$

Another example is a $C^1(H)$ gate where an H gate is implemented on the target qubit when the control qubit is in the $|1$ state.

1.2.7 Toffoli Gate

The Toffoli gate acts on multiple qubits. It leaves the state of the target qubit unchanged if all the controls are not in the $|1$ state, and applies a NOT gate on the target when all the controls are in the $|1$ state. The Toffoli gate in a three qubit system, with qubits 1 and 2 as the controls and qubit 3 as the target, is defined by:

$$Toffoli = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (1.11)$$

Figure 1.8 shows the circuit diagram for the Toffoli gate where qubits 1 and 2 are the controls and qubit 3 is the target qubit. The state table of the Toffoli gate is shown in Table 1.3.

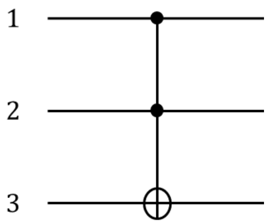

Figure 1.8 The circuit diagram of the Toffoli gate where qubit 1 and 2 are the controls and qubit 3 is the target

TABLE 1.3

STATE TABLE OF THE TOFFOLI GATE

Initial state	**Final state**
$\vert 000$	$\vert 000$
$\vert 001$	$\vert 001$
$\vert 010$	$\vert 010$
$\vert 011$	$\vert 011$
$\vert 100$	$\vert 100$
$\vert 101$	$\vert 101$
$\vert 110$	$\vert 111$
$\vert 111$	$\vert 110$

A Toffoli gate with more than 2 controls is called a multi-control Toffoli gate. Here, a NOT gate is performed on the target qubit only when all the controls are in the $|1$ state.

1.2.8 Swap Gate

The swap gate is a two-qubit gate operation, and it interchanges the states of two qubits. A swap gate for two qubits is defined by:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.12}$$

The circuit diagram and state table of the swap gate is shown in Figure 1.9 and Table 1.4, respectively.

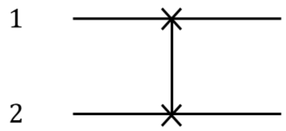

Figure 1.9 The circuit diagram of the swap gate

TABLE 1.4

STATE TABLE OF THE SWAP GATE

Initial state	**Final state**
$\|00$	$\|00$
$\|01$	$\|10$
$\|10$	$\|01$
$\|11$	$\|11$

1.3 Universal Gate Sets

Like classical computing, in quantum computing every multi-qubit gate operation can be decomposed into a sequence of gate operations belonging to a universal set. For example, one such gate set comprises the Hadamard, S and CNOT gates [1]. Another gate set consists of Hadamard, S and Toffoli gates [1]. In any gate set, the accuracy and efficiency with which we are able to perform multi-qubit operations depends on the accuracy and ease with which these gates can be implemented [19]. At the device level, it depends on the system Hamiltonian and the flexibility allowed in controlling the parameters of the system. At the circuit level, it depends on the physical layout of qubits, and the complexity of the control circuitry [19]. As such, techniques for efficiently decomposing multi-qubit operations into universal gates is an active research area [20-29].

1.4 LNN Architectures

In a quantum computer, there are different arrangements of qubits, and the most common one is the Linear Nearest Neighbor (LNN) architecture. An LNN is a one dimensional array where a qubit is coupled only to its two nearest neighbors, on either side of it. As such, only nearest neighbor qubits can interact with each other during gate operations. It has been shown that if a quantum algorithm can be implemented efficiently on an LNN array, it can be realized in many other quantum architectures as well [8]. To perform gate operations in LNN arrays, qubits involved in a gate operation are required to be brought adjacent to each other. This is typically accomplished by implementing swap gates to interchange the position of qubits along the array. In most quantum systems, each swap gate equates to three CNOT gates. As the number of swap gates increases, the total number of gate operations can increase. This can increase the quantum cost and computational overhead of the circuit. As such, many techniques for implementing gate operations efficiently in LNN arrays have been proposed. Template-matching techniques are proposed for multi-qubit gate combinations in LNN architectures [30]. Here, a cascade of reversible gates is replaced by a different cascade with the same functionality, and minimization rules are applied which reduce the number of gates. Exact synthesis methods have been proposed that construct quantum circuits with minimal quantum cost where the circuit synthesis problem is expressed as a sequence of Boolean satisfiability instances [30]. Reordering techniques have been constructed where by modifying the ordering of qubits, additional cost can be saved [30].

1.5 Quantum Error Correction

Irrespective of the layout of qubits, implementing any sophisticated quantum algorithm, like Shor's algorithm, involves several gate operations, which requires manipulating qubits through an external control circuitry. This makes the quantum system susceptible to noise and decoherence [31-34]. Therefore, large scale quantum computation is so difficult as to be practically impossible unless error correction methods are used [35-37]. The first quantum error correcting codes were discovered independently by Shor [38] and Steane [39]. Shor proved that 9 qubits could be used to protect a single qubit against general errors, while Steane described a general code construction using 7 qubits [45]. Later, a 5-qubit quantum error correction was introduced, which demonstrated successful gate implementation and error correction [40,41]. Figure 1.10 shows the encoding circuit for the 5-qubit code [46] which uses Hadamard, CNOT and multi-control Z gates. Note that multi-control Z gates shown in Figure 1.10 are interchangeable with multi-control Toffoli gates (we will prove this interchangeability by lemma 2). Also, as previously mentioned, note that a control qubit can be in the $|0$ state and is represented by an "empty" circle in the circuit. As such, since multi-control Toffoli gates are widely used in quantum error correction, efficient methods for implementing them in LNN arrays, on which most physical proposals of a quantum computer are built, need to be investigated.

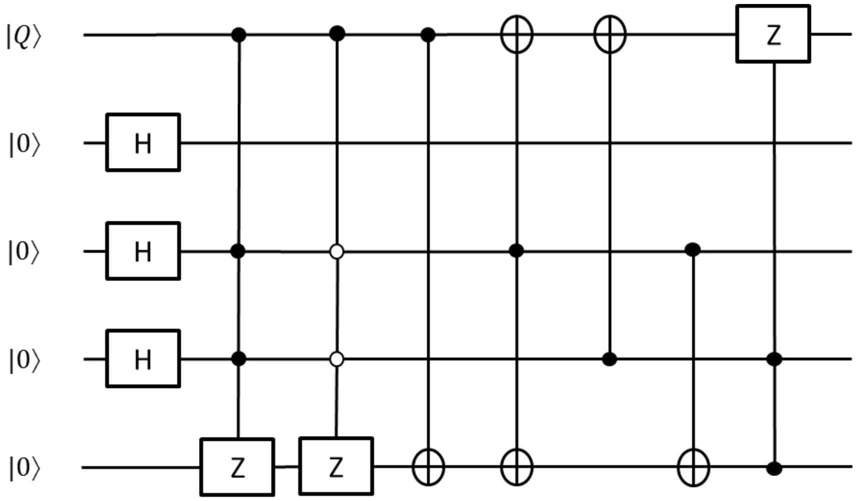

Figure 1.10. The encoding circuit for the 5-qubit code [46] which uses Hadamard, CNOT and multi-control Z gates

1.6 Our Research

Here, we propose a new method to implement multi-control Toffoli gates in LNN arrays without using swap gates. The circuit reduction techniques discussed here are based on 3 lemmas. The first lemma is based on a new gate, introducing in [42], called the $C^2(-I)$ gate. This gate allows us to perform a controlled-Z gate between two uncoupled next-to-NN qubits without having to bring them adjacent to each other. The second and third lemmas are derived from prior work in [43]. Using the lemmas, we first show how to implement Toffoli gates in LNN arrays with different separations between the two control qubits and the target qubit. We then extend our scheme to show how to implement multi-control Toffoli gates in LNN arrays with arbitrary separation between the controls and target. For each case, the final circuit only comprises of the least number of $C^2(-I)$, H and CNOT gates (all between NN qubits). The key feature of our scheme is to involve qubits other than control and target qubits to take part in gate operations. We call these qubits "auxiliary" qubits, which are simply qubits in arbitrary states, $\alpha|0\ \ +\beta|1$, in the LNN array. They do not have to be prepared in any special states, for instance, they are not ancillas, which are qubits in the $|0$ state. We simply use auxiliary qubits in our gate decomposition protocols, at the end of which, they are always restored back to their original states. Compared to circuits using swap gates, the total number of gate operations used in our method is decreased. As such, the quantum cost of the circuit is lowered.

In addition to implementing multi-control Toffoli gate operations efficiently in LNN arrays, we also show how to extend our protocols to 2D arrays. Here, in addition to translating our gate reduction techniques directly from 1D to 2D, additional simplification is possible for particular arrangements of qubits. To this end, we introduce a new gate, $C^4(-I)$ gate, analogous to the $C^2(-I)$ gate in 1D LNN arrays, using which we are able to implement a 3-control-1-target Toffoli gate directly, without any interactions between the four qubits involved. Thus, additional lowering of computational cost might be possible using our scheme in 2D and 3D arrays. This, however, is out of scope of this thesis, and can be pursued as a future work.

CHAPTER 2

EFFICIENT IMPLEMENTATION OF MULTI-CONTROL TOFFOLI GATES IN LINEAR NEAREST NEIGHBOR ARRAYS

S. Daraeizadeh and P. Kumar

2.1 Abstract

Most proposals for quantum computers are based on linear nearest neighbor (LNN) arrangements where qubits only interact with neighbors. Multi-control Toffoli gates are used in many quantum applications such as error correction, and algorithms like Shor's factorization. Typically, to implement a multi-control Toffoli gate in an LNN architecture, additional operations called swap gates are required to bring the qubits adjacent to each other. We propose a new method to implement multi-control Toffoli gates in LNN arrays without using swap gates. As such, compared to circuits using swap gates, the quantum cost of our circuit is much lower.

2.2 Keywords

Quantum, Qubit, Linear Nearest Neighbor Architecture, CNOT gate, Toffoli gate, multi-control, multi-target

2.3 Introduction

Quantum computing comprises a series of gate operations on qubits (quantum bits) [1]. A qubit is a unit of information, and a two-state quantum-mechanical system. Unlike a classical bit, a qubit can be in one of the two basis states, $|0$ or $|1$, or a linear superposition of the two states. Examples of qubits are the polarizations of a single photon, superconducting qubits, ion-trap qubits, nuclear magnetic resonance qubits, and quantum dots [1-4]. The special characteristic of the ability to exist in superposition states of qubits allows quantum computers to provide exponentially faster results than traditional computers for solving some problems like factoring numbers. In a quantum computer, there are different arrangements of qubits, and the most common one is the Linear Nearest Neighbor (LNN) architecture. An LNN is a one dimensional array where every qubit is coupled only to its two neighbors. The authors in [8] showed that if a quantum algorithm can be realized efficiently on an LNN architecture, it can be realized in many other architectures as well, making LNN architectures an active research area [8-18].

In every closed quantum system, the Hamiltonian is the operator corresponding to the total energy of the system. Quantum gates are unitary operations generated from the Hamiltonian. A quantum gate is implemented by tuning one or more appropriate controllable parameters of the system [1-7]. Like classical gates, quantum gates can be single-qubit or multi-qubit gate operations. For example, NOT gate, Hadamard gate and phase (/2, /4, etc.) gate are examples of single-qubit quantum gates. Swap gate, CNOT gate and Toffoli gate are examples of multi-qubit quantum gates, where two or more qubits take part in the gate operations.

Like classical computing, in quantum computing every multi-qubit gate operation can be decomposed into a sequence of gate operations belonging to a universal set. For example, one such gate set comprises the Hadamard, S (phase shift gate where $\theta = \frac{\pi}{8}$), T (phase shift gate where $\theta = \frac{\pi}{4}$) and CNOT gates [1]. Another gate set consists of Hadamard, S, and Toffoli gates [1]. In any gate set, the accuracy and efficiency with which we are able to perform multi-qubit operations depends on the accuracy and ease with which these gates can be implemented [19]. At the device level, it depends on the system Hamiltonian and the flexibility allowed in controlling the parameters of the system. At the circuit level, it depends on the physical layout of qubits and the complexity of the control circuitry [19]. As such, techniques for efficiently decomposing multi-qubit operations into universal gates is an active research area [20-29].

In a quantum computer, there are different arrangements of qubits, and the most common one is the Linear Nearest Neighbor (LNN) architecture. An LNN is a one dimensional array where a qubit is coupled only to its two nearest neighbors, on either side of it. As such, only nearest neighbor qubits can interact with each other during gate operations. It has been shown that if a quantum algorithm can be implemented efficiently on an LNN array, it can be realized in many other quantum architectures as well [8]. To perform gate operations in LNN arrays, qubits involved in a gate operation are required to be brought adjacent to each other. This is typically accomplished by implementing swap gates to interchange the position of qubits along the array. In most quantum systems, each swap gate equates of three CNOT gates. As the number of swap gates increases, the total number of gate operations can increase. This can increase the quantum cost and computational overhead of the circuit. As such, many techniques for implementing gate operations efficiently in LNN arrays have been proposed. Template-matching techniques are proposed for multi-qubit gate combinations in LNN architectures [30]. Here, a cascade of reversible gates is replaced by a different cascade with the same functionality, and minimization rules are applied which reduce the number of gates. Exact synthesis methods have been proposed that construct quantum circuits with minimal quantum cost where the circuit synthesis problem is expressed as a sequence of Boolean satisfiability instances [30]. Reordering techniques have been constructed where by modifying the ordering of qubits, additional cost can be saved [30].

Irrespective of the layout of qubits, implementing any sophisticated quantum algorithm, like Shor's algorithm, involves several gate operations, which requires manipulating qubits through an external control circuitry. This makes the quantum system susceptible to noise and decoherence [31-34]. Therefore, large scale quantum computation is so difficult as to be practically impossible unless error correction methods are used [35-37]. The first quantum error correcting codes were discovered independently by Shor [38] and Steane [39], which they used nine-qubit and seven-qubit error correcting codes respectively. Later, a five-qubit quantum error correction was introduced, which demonstrated successful gate implementation and error correction [40,41]. Since multi-control Toffoli gates (with more than two controls) are widely used in quantum error correction, efficient methods for implementing them in LNN arrays, on which most physical proposals of a quantum computer are built, need to be investigated.

Here, we propose a new method to implement multi-control Toffoli gates in LNN arrays without using swap gates. The circuit reduction techniques discussed here are based on 3 lemmas. The first lemma is based on a new gate, introducing in [42], called the $C^2(-I)$ gate. This gate allows us to perform a controlled-Z gate between two uncoupled next-to-NN qubits without having to bring them adjacent to each other. The second and third lemmas are derived from prior work in [43]. Using the lemmas, we first show how to implement Toffoli gates in LNN arrays with different separations between the two control qubits and the target qubit. We then extend our scheme to show how to implement multi-control Toffoli gates in LNN arrays with arbitrary separation between the controls and target. For each case, the final circuit only comprises of the least number of $C^2(-I)$, H, and CNOT gates (all between NN qubits). The key feature of our scheme is to involve qubits other than control and target qubits to take part in gate operations. We call these qubits "auxiliary" qubits, which are simply qubits in arbitrary states, $\alpha|0\ +\beta|1$, in the LNN array. They do not have to be prepared in any special states, for instance, they are not ancillas, which are qubits in the $|0$ state. We simply use auxiliary qubits in our gate decomposition protocols, at the end of which, they are always restored back to their original states. Compared to circuits using swap gates, the total number of gate operations used in our method is decreased. As such, the quantum cost of the circuit is lowered.

In addition to implementing multi-control Toffoli gate operations efficiently in LNN arrays, we also show how to extend our protocols to 2D arrays. Here, in addition to translating our gate reduction techniques directly from 1D to 2D, additional simplification is possible for particular arrangements of qubits. To this end, we introduce a new gate, $C^4(-I)$ gate, analogous to the $C^2(-I)$ gate in 1D LNN arrays, using which we are able to implement a 3-control-1-target Toffoli gate directly, without any interactions between the four qubits involved. Thus, additional lowering of computational cost might be possible using our scheme in 2D and 3D arrays. This, however, is out of scope of this research, and can be pursued as a future work.

The remainder of this paper is organized as follows. In section 2.4, definitions and lemmas are introduced. In section 2.5, implementation of a Toffoli gate in an N qubit system is explained. In sections 2.6 and 2.7, methods for implementing a multi-control Toffoli gate and a multi-control-multi-target Toffoli gate are described. In section 2.8, simulations are shown to validate our gate operations. In section 2.9, a technique for efficiently implementing a multi-control Toffoli gate in a 2-dimensional quantum system is shown. The conclusions are presented in section 2.10.

2.4 Definitions and Lemmas

Throughout the paper, the following definitions will be used:

2.4.1 Definitions

Nearest Neighbor Cost (NNC): The NNC of a two-qubit controlled-unitary gate with control placed at the c^th^ line and target at the t^th^ line is defined as the distance between the control and target, i.e., $|c - t - 1|$. The NNC of a circuit is the sum of the NNCs of its gates. Optimal NNC is zero, where all quantum gates are either 1- or 2-qubit gates performed on adjacent qubits. The NCC of a multi-qubit controlled-unitary gate is the maximum distance between any two qubits (controls or targets) involved in the gate operation.

Gate Count: The gate count of a circuit is the total number of gates used to implement the circuit.

Depth: The Depth is number of layers of gates. All gates in a layer are realized simultaneously.

Participant Qubit: A qubit which is involved in a gate operation either as a control or a target qubit.

Auxiliary Qubit: A qubit that is not a participant in a gate operation. An auxiliary qubit is a data qubit in any arbitrary quantum state, $\alpha|0\ +\beta|1$. If used in a gate operation, auxiliary qubits are always restored to their original state.

Definition 1: A $C^2(-iX)$ is a controlled-unitary matrix which implements the $-i(X)$ gate on the target qubit when the two controls are in the $|11$ state. Here, the gate operation can be defined by the following linear transformation:

$$C^2(-iX)|abc\ =\begin{cases}-i|ab'c\ ,\ if\ |ac\ =|11\\ \quad |abc\ ,\ if\ |ac\ \neq|11\end{cases}\tag{2.1}$$

where $|a$ and $|c$ are the states of the control qubits, $|b$ is the target qubit and $|b'$ is the complement state of qubit $|b$, for instance, if $|b\ =|0$, then $|b'\ =|1$, and vice versa. An X (or NOT) gate is defined by the following matrix:

$$X=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\tag{2.2}$$

Definition 2: A $C^2(-I)$ gate is defined by the following linear transformation:

$$C^2(-I)|abc\ =\begin{cases}-|abc\ ,\ if\ |ac\ =|11\\ \ |abc\ ,\ if\ |ac\ \neq|11\end{cases}\tag{2.3}$$

where $|a$ and $|c$ are the states of controls and $|b$ is the target qubit. From equation (2.1), we can see that two successive applications of a $C^2(-iX)$ gate equate to a $C^2(-I)$ gate.

Note that the $C^2(-I)$ gate in a 3-qubit system is equivalent to a $C^1(Z)$ gate between qubits 1 and 3 [42].

2.4.2 Lemmas

We will now introduce 3 lemmas that will form the basis for all our circuit reduction techniques. Lemma 1 is new, lemmas 2 and 3 are derived from the results in [43].

Lemma 1: A $C^2(X)$ or Toffoli gate between three qubits in an LNN array, where the target qubit is not in the middle position (Figure 2.1), can be replaced by a $C^2(-I)$ gate sandwiched between two controlled-Hadamard gates applied between the second and third qubits. A $C^2(-I)$ gate has two controls and one target [42]. A phase is picked up by the target only when the two controls are in the $|11$ state. A controlled-Hadamard gate implements a Hadamard gate on the target qubit when the control qubit is in the $|1$ state. The gate operation is defined as:

$$C^1(H) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{2.4}$$

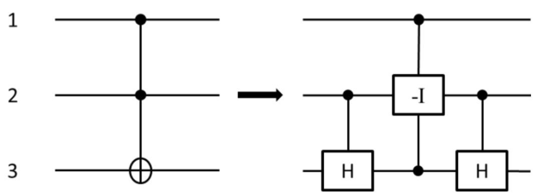

Figure 2.1. Implementation of a $C^2(X)$ or Toffoli gate in a 3-qubit LNN array between qubits 1, 2 and 3 with qubit 3 as the target

Proof: We will use a state table to show the proof (Table 1).

TABLE 2.1

STATE TABLE OF FIGURE 2.1

Initial state	**First $C^1(H)$ gate**	**$C^2(-I)$ gate**	**Second $C^1(H)$ gate**
$\vert 000$	$\vert 000$	$\vert 000$	$\vert 000$
$\vert 001$	$\vert 001$	$\vert 001$	$\vert 001$
$\vert 010$	$\frac{1}{\sqrt{2}}[\vert 01\ (\vert 0\ + \vert 1\ )]$	$\frac{1}{\sqrt{2}}[\vert 01\ (\vert 0\ + \vert 1\ )]$	$\vert 010$
$\vert 011$	$\frac{1}{\sqrt{2}}[\vert 01\ (\vert 0\ - \vert 1\ )]$	$\frac{1}{\sqrt{2}}[\vert 01\ (\vert 0\ - \vert 1\ )]$	$\vert 011$
$\vert 100$	$\vert 100$	$\vert 100$	$\vert 100$
$\vert 101$	$\vert 101$	$\vert 101$	$\vert 101$
$\vert 110$	$\frac{1}{\sqrt{2}}[\vert 11\ (\vert 0\ + \vert 1\ )]$	$\frac{1}{\sqrt{2}}[\vert 11\ (\vert 0\ - \vert 1\ )]$	$\vert 111$
$\vert 111$	$\frac{1}{\sqrt{2}}[\vert 11\ (\vert 0\ - \vert 1\ )]$	$\frac{1}{\sqrt{2}}[\vert 11\ (\vert 0\ + \vert 1\ )]$	$\vert 110$

By this method, the gate count and the depth of the circuit are 3. In [1], the authors show that a controlled-unitary gate operation, a $C^2(U)$ gate, with two NN controls and one target in a three qubit system, can be decomposed into five gate operations comprising CNOT and controlled-V

gates. Note that here, the controlled-V gate is the square root of the gate operation U. For realizing the same Toffoli gate in Figure 2.1 by using the method in [1], the gate count is 11. Therefore by using our method, the quantum cost has been improved by 72%.

Lemma 2: A $C^n(X)$ gate in an LNN array comprising N qubits, where n is the number of controls and $n < N$, can be replaced by a $C^n(Z)$ gate sandwiched between two Hadamard gates applied on the target qubit:

$$C^n(X) = H_k(C^n(Z))H_k \tag{2.5}$$

where H_k is a $N \times N$ matrix defined by:

$$H_k = I^{k-1} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \otimes I^{N-k} \tag{2.6}$$

where "k" is the position of the target qubit, and $1 \leq k \leq N$.

Proof: Assume that the target is in the $\alpha|0\ + \beta|1\ $ state and all the controls are in the $|1$ state. By applying the first Hadamard gate, the state of the target becomes $(\frac{1}{\sqrt{2}})(\alpha(|0\ + |1\ ) + \beta(|0\ - |1\ ))$. Next, the $C^n(Z)$ gate is implemented on the target and the state of the target is altered to $(\frac{1}{\sqrt{2}})(\alpha(|0\ - |1\ ) + \beta(|0\ + |1\ ))$. Finally, after applying the last Hadamard gate, the state of the target becomes $\alpha|1\ + \beta|0\ $. Therefore, an X (or NOT) gate is applied on the target qubit when all the controls are in the $|1\ $ state. If some or all of the control qubits are not in the $|1\ $ state, two successive H_k gates are applied on the target qubit, wherein the state of the qubit remains unchanged (since H_k is self-inverse).

Corollary 1: A $C^n(Z)$ gate in an LNN array comprising N qubits can be replaced by a $C^n(X)$ gate sandwiched between two Hadamard gates applied on the target qubit:

$$C^n(Z) = H_k(C^n(X))H_k \tag{2.7}$$

Proof: Equation (2.7) can be derived by applying two Hadamard gates on either side of equation (2.5):

$$H_k[C^n(X)]H_k = H_k[H_k(C^n(Z))H_k]H_k \tag{2.8}$$

Since, the Hadamard gate is a self-inverse unitary matrix, we have $(H_\mathrm{k})^2 = I$.

Lemma 3: A $C^n(Z)$ gate in an N qubit system, where $N \geq n + 2$, can be decomposed into two $C^{n-w}(X)$ gates and two $C^{w+1}(Z)$ gates, where $0 \leq w < n$ and all four controlled-unitary operations use the same auxiliary as the target qubit. The advantage of using this lemma repeatedly is that non-NN controlled-unitary gates can be broken down into NN controlled unitary gates by using auxiliary qubits.

Proof: Figure 2.2 shows a $C^2(Z)$ gate in an N qubit system, where qubits a and b are the controls and qubit d is target, and $N \geq 4$. For simplicity, qubits a and b are chosen to be adjacent to each other. The $C^2(Z)$ gate is decomposed into two $C^2(X)$ (or Toffoli) gates and two $C^1(Z)$ gates, where all operations use qubit c as the auxiliary qubit. Here, $n = 2$ and $w = 0$.

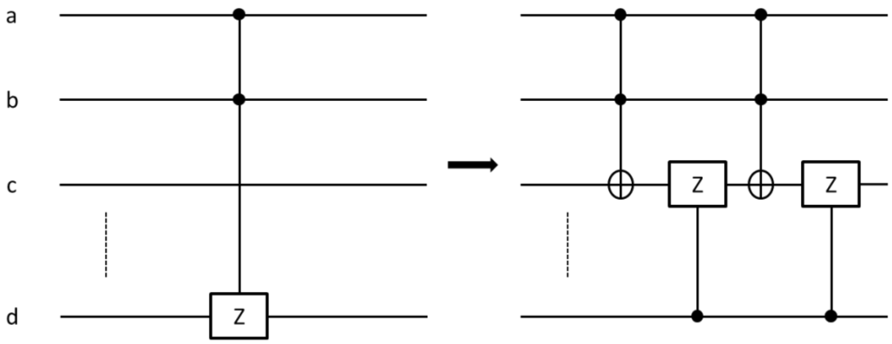

Figure 2.2. Decomposition of a $C^2(Z)$ gate in an N-qubit LNN array into two $C^2(X)$ (or Toffoli) gates and two $C^1(Z)$gates

We assume $|a$, $|b$ and $|d$ are all in the $|1$ state. For the first Toffoli gate we have:

$$|a, b, c, d \quad \rightarrow |a, b, c', d$$

where $|c'$ is the complement state of qubit $|c$. For the second gate we have:

$$|a,b,c',d \;\rightarrow\; |a,b,c',d.(-1)^{c'.d}$$

where $(-1)^{c'.d}$ is the $C^1(Z)$ gate between $|c'$ and $|d$. For the third gate we have:

$$|a,b,c',d.(-1)^{c'.d} \;\rightarrow\; |a,b,c,d.(-1)^{c'.d}$$

where $|c$ is the complement state of qubit $|c'$. Finally, for the last gate we have:

$$|\,a,b,c,d.(-1)^{c'.d} \;\rightarrow\; |\,a,b,c,d.(-1)^{c'.d}.(-1)^{c.d}$$

where $(-1)^{c'.d}.(-1)^{c.d}$ yields (-1), since either $c'.d = 0$ or $c.d = 0$. Therefore, the final state becomes $|a,b,c,-d$. If $|a$, $|b$ or $|c$ are not all in the $|1$ state, the output remains unchanged.

2.5 Implementing Toffoli Gates between Non-Adjacent Qubits in Multi-Qubit Systems

We will now show how to use the results of section 2.4 to implement Toffoli gates in multi-qubit systems. In this section, we are only concerned with the 3-qubit Toffoli gate which has 2 controls and 1 target. We show how to implement this gate in an N-qubit system ($N > 3$) when the control and target qubits are not adjacent to each other (NCC$\neq 0$) without having to use swap gates. In subsequent sections, we will deal with Toffoli gates with more than 2 controls (multi-control Toffoli gates) and more than 1 target (multi-control-multi-target Toffoli gates).

2.5.1 Toffoli Gate in a 4 Qubit System

Figure 2.3 shows our method for implementing a Toffoli gate in a four qubit system, where two of the participants are not adjacent. By lemma 2, the Toffoli gate is realized by sandwiching a $C^2(Z)$ gate between two H gates applied on the target qubit. By lemma 3, the $C^2(Z)$ gate is replaced by two Toffoli gates between qubits 1, 2 and 3 with qubit 2 as the target and two $C^1(Z)$ gates between qubits 2 and 4. Note that qubit 2 is the auxiliary qubit. If the Toffoli gate is between qubits 1, 2 and 4, where qubits 1 and 2 are controls and qubit 4 is the target, we use qubit 3 as the auxiliary qubit for the decomposition of the Toffoli gate. Since the

$C^1(Z)$ gate is applied only when all the participants are in the $|1$ state, it does not matter what qubit is chosen as the control or the target qubit. Each of the two $C^1(Z)$ gates between qubits 2 and 4 will be replaced by two $C^2(-I)$ gates.

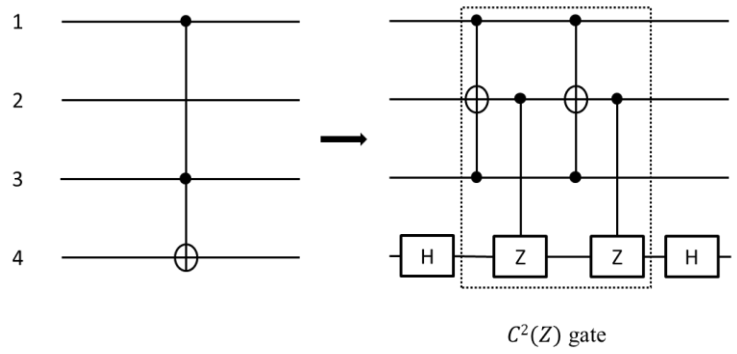

Figure 2.3. Implementation of a Toffoli gate in a 4 qubit system using qubit 2 as the auxiliary qubit. If the Toffoli gate is between qubits 1, 2 and 4, where qubits 1 and 2 are controls and qubit 4 is the target, we use qubit 3 as the auxiliary qubit for the decomposition of the Toffoli gate

Note that if the method in [1] is used to realize the Toffoli gate in a four qubit system, wherein swap gates are applied to bring qubits adjacent to each other, the gate count is 17. Using our method, the same Toffoli gate can be implemented by 6 elementary gates. The quantum cost is improved by 64%.

2.5.2 Toffoli Gate in an N Qubit System

Figure 2.4 shows a Toffoli gate with three participants in an N qubit system, where qubits p, q and r are the participants. Here and throughout, p is the 1st qubit, r is the Nth qubit and q is a qubit between them. As before, to perform a Toffoli gate, a $C^2(Z)$ gate is realized on these three participants which is sandwiched between two H gates applied on the target qubit. Qubit m shows the position of the auxiliary qubit. In our method, we divide qubits into two major groups, “I” and “II”; where each group comprises “m” and “$(N - m) + 1$” qubits, respectively. If N is odd, we choose “$m = (N + 1)/2$”, and if N is even, we choose “$m = (N +$

$2)/2$". In either case, if $m = q$, we change $m = q + 1$. Depending on whether participant q is in group "I" (case 1) or in group "II" (case 2), each $C^2(Z)$ gate is decomposed into four gates. In case 1, the $C^2(Z)$ gate is replaced by two Toffoli gates between p, q and m, with m as the target, and two $C^1(Z)$ gates between m and r. In case 2, the $C^2(Z)$ gate is substituted by two Toffoli gates between r, q and m (m as the target), and two $C^1(Z)$ gates between m and p. In Figure 2.4, the two Toffoli gates implemented between qubits p, q and m (group "I") need not be between nearest neighbor qubits (NNC≠ 0). If the Toffoli gates are not of the form of Figure 2.3, further decomposition into sub-groups are needed.

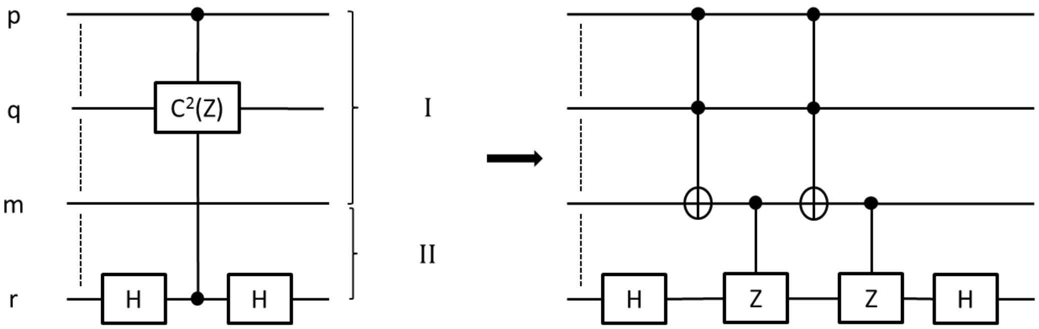

Figure 2.4. Implementation of a Toffoli gate in an N qubit system (case 1) where participant q is in group "I"

This process of decomposition is carried out until all Toffoli gate operations are of the form of Figure 2.1. Likewise, in each group/subgroup, if the $C^1(Z)$ gates are such that the separation between the control and target qubits is greater than one (NNC > 1), each gate will have to be broken down into subgroups (if NNC $= 1$, a $C^1(Z)$ gate is replaced by a $C^2(-I)$ gate). In [42], the author shows how to implement a $C^1(Z)$ gate between non-NN qubits. For P 7, where P is the number of qubits, the total gate count for the $C^1(Z)$ gate is shown to be [42]:

$$G_P = (20 + (P - 7) \times 6\,; P \geq 7 \qquad (2.9)$$

Note that, here $P = (N - m) + 1$.

2.6 Multi-Control Toffoli Gates

In this section, we introduce a new method of realizing multi-control Toffoli gates, the $C^n(X)$ gate, where $n \geq 3$ in N qubit LNN architectures. We then show an example of implementing a $C^3(X)$ in a five-qubit system.

2.6.1 Multi-Control Toffoli Gates in *N* Qubit Systems

Figure 2.5(a) (the first plot) shows a $C^n(X)$ gate in an N qubit system, where $n+1$ qubits are the participants and $n+1 \leq N-1$. For realizing multi-control Toffoli gates, at least one additional qubit, other than the participants in the circuit, is required. As shown in Figure 2.5(a) (the second plot) and based on lemma 2, any $C^n(X)$ gate can be replaced by a $C^n(Z)$ gate sandwiched between two Hadamard gates applied on the target qubit. Based on lemma 3, the $C^n(Z)$ gate is decomposed into two $C^{n-w}(X)$ gates and two $C^{w+1}(Z)$ gates, where all four gate operations use qubit m as the target qubit. For the $C^{n-w}(X)$ and $C^{w+1}(Z)$ gates, if $n-w \geq 3$ or $w+1 \geq 3$, further decompositions based on lemma 2 and 3 are needed. These decompositions go on until $C^2(X)$ and $C^2(Z)$ gates are extracted. Then, the method explained in section 2.5 can be used. Figure 2.5(b) shows the steps of decomposition.

The worst case is when there is not any auxiliary qubit among participants to implement further multi-control gates in the circuit. For example, in Figure 2.5(b), no auxiliary qubit is left to implement $C^{n-w}(X)$ gate between qubits p and m with qubit m as the target. In this case, the first available qubit next to the target, qubit q, which is not a participant for the desired gate, is used as an auxiliary qubit for the decomposition. Figure 2.5(c) shows how the auxiliary qubit is used and brought back to its initial state.

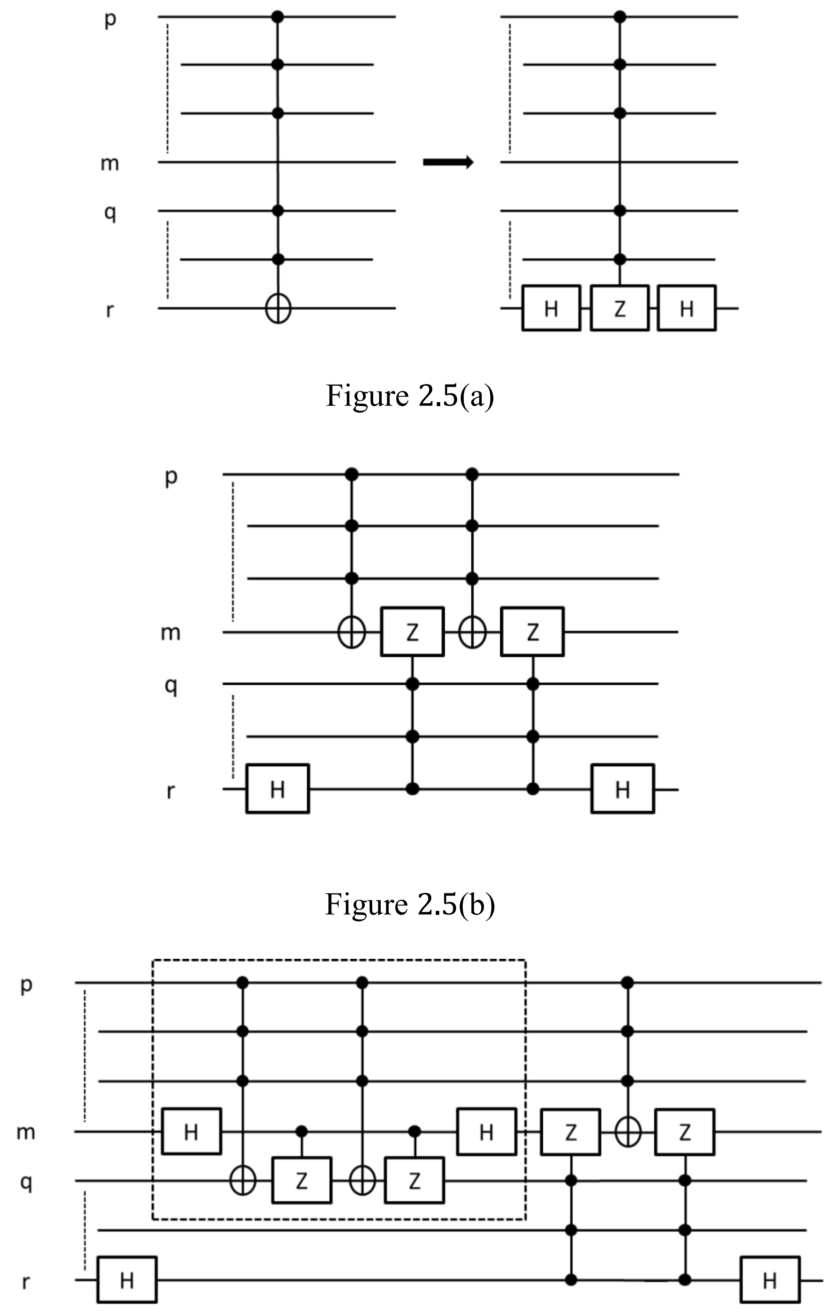

Figure 2.5(a)

Figure 2.5(b)

Figure 2.5(c)

Figure 2.5. Implementation of a multi-control Toffoli gate in an N qubit system. (a) Using lemma 2 to decompose the multi-control Toffoli gate into multi-control Z gate sandwiched between Hadamard gates on the target. (b) Using lemma 3 to break down the multi-control Z gate into two $C^{n-w}(X)$ gates and two $C^{w+1}(Z)$ applied on the auxiliary qubit, "m". (c) Implementation of a multi-control Toffoli gate without any auxiliary qubit among participants (the worst case), where qubit q is used as an auxiliary qubit

2.6.2 A $C^3(X)$ Gate in a 5 Qubit System

Figure 2.6 shows an example of multi-control Toffoli gate in a 5 qubit system, where qubits 1, 2 and 4 are the controls and qubit 5 is the target qubit. Here, qubit 3 is the auxiliary qubit and participates in gate operations.

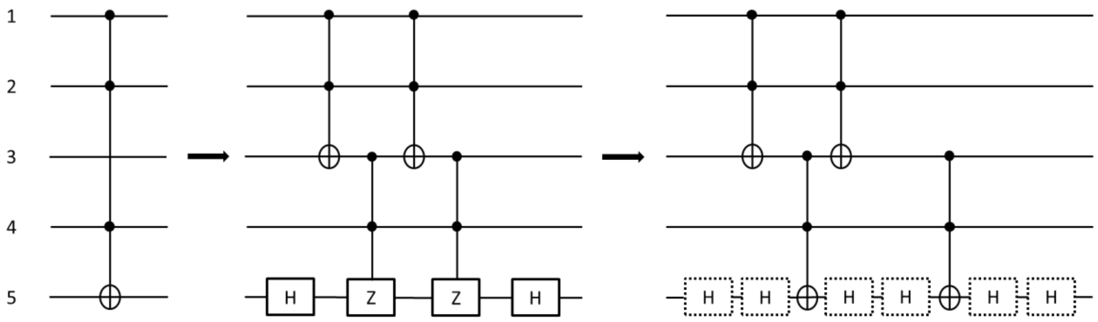

Figure 2.6. Implementation of a multi-control Toffoli gate in a 5 qubit system where qubits 1, 2 and 4 are the controls and qubit 5 is the target

Further simplification is possible if corollary 1 is used. Each of the two $C^2(Z)$ gates can be replaced by a $C^2(X)$ (or Toffoli) gate and two Hadamard gates. Finally, the two successive H gates are cancelled out (shown by dotted boxes), and the gate count is 12.

2.7 Multi-Control-Multi-Target Toffoli Gates

In this section, we propose a new circuit reduction technique for implementing a multi-control-multi-target Toffoli gate in N qubit LNN arrays. We then show an example of realizing a two-control-two-target Toffoli gate in a five qubit system.

2.7.1 Multi-Control-Multi-Target Toffoli Gates in N Qubit Systems

In Figure 2.7 (the first plot), a multi-control-multi-target Toffoli gate in an N qubit system is shown. Here, n and l are the numbers of controls and targets respectively, and $n + l \leq N$. The key point is to decompose the desired gate into l different multi-control Toffoli gates and simplify the circuit.

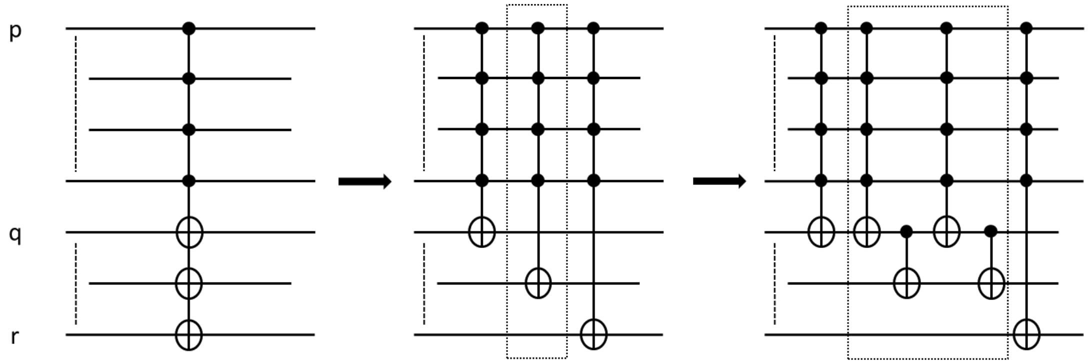

Figure 2.7. Decomposition of a multi-control-multi-target Toffoli gate into a sequence of multi-control Toffoli gates, and simplifying the circuit

The first step is to decompose the multi-control-multi-target gate into a sequence of multi-control Toffoli gates by taking out the target qubits (the second figure in Figure 2.7). By using the concepts of implementing multi-control Toffoli gates discussed in section 2.6, further simplifications are possible. As shown in the third plot of Figure 2.7, the second multi-control Toffoli gate (the dotted box) is decomposed into two multi-control Toffoli gates and two CNOT gates, using qubit q as the auxiliary qubit. Note that the first two gates are identical and can be cancelled out. The main advantage of this method is that no auxiliary qubit is required.

2.7.2 Multi-Control-Multi-Target Toffoli Gate in a 5 Qubit System

As an example of a multi-control-multi-target Toffoli gate, Figure 2.8(a) shows a two-control-two-target Toffoli gate, where qubits 1 and 3 are the controls and qubits 2 and 5 are the targets. The desired gate is decomposed into two Toffoli gates. The first Toffoli gate is between qubits 1, 2 and 3, where qubits 1 and 3 are the controls and qubit 2 is the target. The second Toffoli gate is between qubits 1, 3 and 5, where qubits 1 and 3 are the controls and qubit 5 is the target. (The second Toffoli gate is replaced by: a Hadamard gate on qubit 5, a Toffoli gate between qubits 1, 2 and 3, where qubits 1 and 3 are the controls and qubit 2 is the target, a controlled-Z gate between qubit 2 and 5, a Toffoli gate between qubits 1, 2 and 3, where qubits

1 and 3 are the controls and qubit 2 is the target, a controlled-Z gate between qubits 2 and 5 and a Hadamard gate on qubit 5). As it is shown in the Figure 2.8(a), two successive identical Toffoli gates are eliminated (the gate is self-inverse). Figure 2.8(b) shows the final circuit where the gate count is 11.

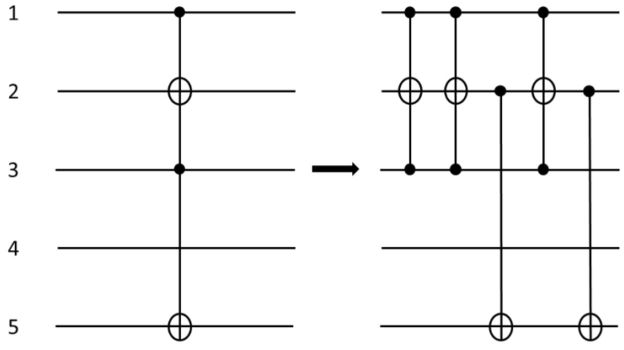

Figure 2.8(a)

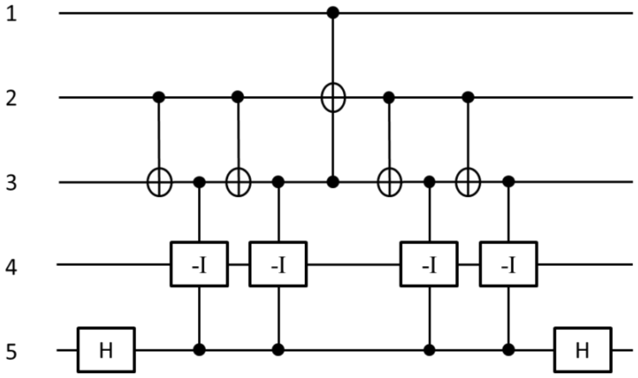

Figure 2.8(b)

Figure 2.8. Implementation of a multi-control-multi-target Toffoli gate in a 5 qubit system. (a) Decomposition of the desired gate into two Toffoli gates, where the second gate is broken down by using the concepts of realizing multi-control Toffoli gates discussed in section 2.6. (b) The final circuit comprising NN gates

2.8 Simulations

As an example, for our simulations, we consider an Ising-coupled LNN system of N qubits. The Hamiltonian, H_N, where “N” represents the number of qubits in the system, is:

$$H_N = \sum_{i=1}^{N}(\Delta_i\sigma_{Xi} + \varepsilon_i\sigma_{Zi}) + \sum_{i=1}^{N-1}(\ {}_{i,i+1}\sigma_{Zi}\sigma_{Zi+1}) \tag{2.10}$$

Here, the terms Δ_i and ε_i for $i = 1, 2 \ldots N$, are the tunneling and bias parameters of the i^{th} qubit Q_i, respectively, and ${}_{i,i+1}$ is the coupling parameter between qubits Q_i and Q_{i+1}. Also, σ_{Xi} and σ_{Zi} for $i = 1, 2, \ldots., N$, are the Pauli matrices corresponding to qubit Q_i:

$$\sigma_{Xi} = I^{i-1}\otimes\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\otimes I^{N-i} \tag{2.11}$$

$$\sigma_{Zi} = I^{i-1}\otimes\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\otimes I^{N-i} \tag{2.12}$$

Ising type interactions are diagonal interactions in the computational basis, which are typical of Josephson-junction qubits [2,44]. Our simulation results were run in MATLAB where we examined the evolution of the system described by the Hamiltonian, H_N, by integrating the Schrödinger equation with respect to time. The Schrödinger equation is defined as:

$$\frac{ih}{2\pi}\frac{\partial}{\partial t}|\psi\ = H_N|\psi \tag{2.13}$$

where $|\psi$ is the state of the system and "h" is the Planck's constant. In our simulations, we normalized the Plank's constant to 1. The parameters for the Hamiltonian depends upon the gate operation being realized and were based on results presented in [6], using a pulsed-bias technique in superconducting qubits.

Suppose we want to simulate a Toffoli gate in a four qubit system (Figure 2.3), where qubit 1 and 3 are the controls and qubit 4 is the target. To implement gate operations, we use the analytical solutions presented in [6,19] to tune the parameter values (pulsed-bias scheme). We assumed all the qubits to be identical in design, and fixed the tunneling parameter $\Delta = \Delta_1 = \Delta_2 = \Delta_3 = \Delta_4 =$ 25 MHz. For the first gate, Hadamard gate on qubit 4, we fix the couplings ${}_{1,2} = {}_{2,3} = {}_{3,4} =$ 0 Hz, $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 =$ 4 GHz, and $\varepsilon_4 = \Delta =$ 25 MHz for $T_1 =$ 7 nanoseconds. For the second gate, a Toffoli gate between qubits 1, 2 and 3 where qubit 2 is the target, we tune $_{1,2} = {}_{2,3} = 1$ GHz, $_{3,4} = 0$ Hz, $\varepsilon_1 = \varepsilon_3 = \varepsilon_4 = 4$ GHz, and $\varepsilon_2 = {}_{1,2} + {}_{2,3} = 2$ GHz for $T_2 = \frac{1}{4\Delta} = 10$ nanoseconds. For the third gate, $C^2(-I)$ gate between qubits 2, 3 and 4 where qubit 3 is the target, we fix $_{1,2} = 0$ Hz, $_{2,3} = {}_{3,4} = 1$ GHz, $\varepsilon_1 = \varepsilon_2 = \varepsilon_4 = 4$ GHz, and $\varepsilon_3 = {}_{2,3} + {}_{3,4} = 2$ GHz for $T_3 = \frac{1}{2\Delta} = 20$ nanoseconds. For the fourth, fifth and sixth gates, we use the same parameters obtained for the first, second and third gates, respectively. Note that the time period for the last Hadamard gate is $T_6 = 3 \times T_1 = 21$ nanosecond, in order to cancel out the global phase of "$\frac{\pi}{2}$" resulting from the first Hadamard gate [42]. Figure 2.9 shows the simulation results when the initial state is $|1110$. The figure shows the probabilities of each qubit. The final state is $|1111$. As another example, Figure 10 shows the simulation results for a multi-control-multi-target gate in a five qubit system (Figure 8(b)), with qubits 1 and 3 as the controls and qubits 2 and 5 as the targets. The initial state of the system is $|10101$, and the final state is $|11100$, where the states of the targets are inverted from $|0$ to $|1$ and vice versa since the control qubits are in the $|1$ state.

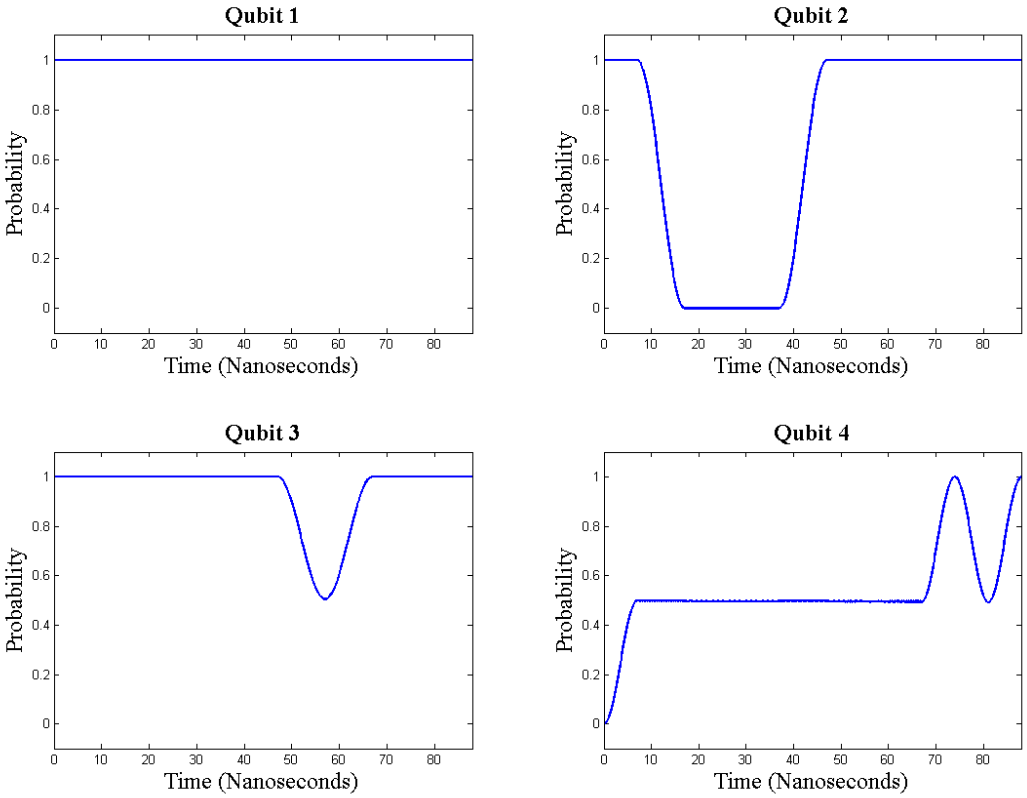

Figure 2.9. Simulation results for the Toffoli gate discussed in section 2.5 with qubits 1 and 3 as the controls and qubit 4 as the target

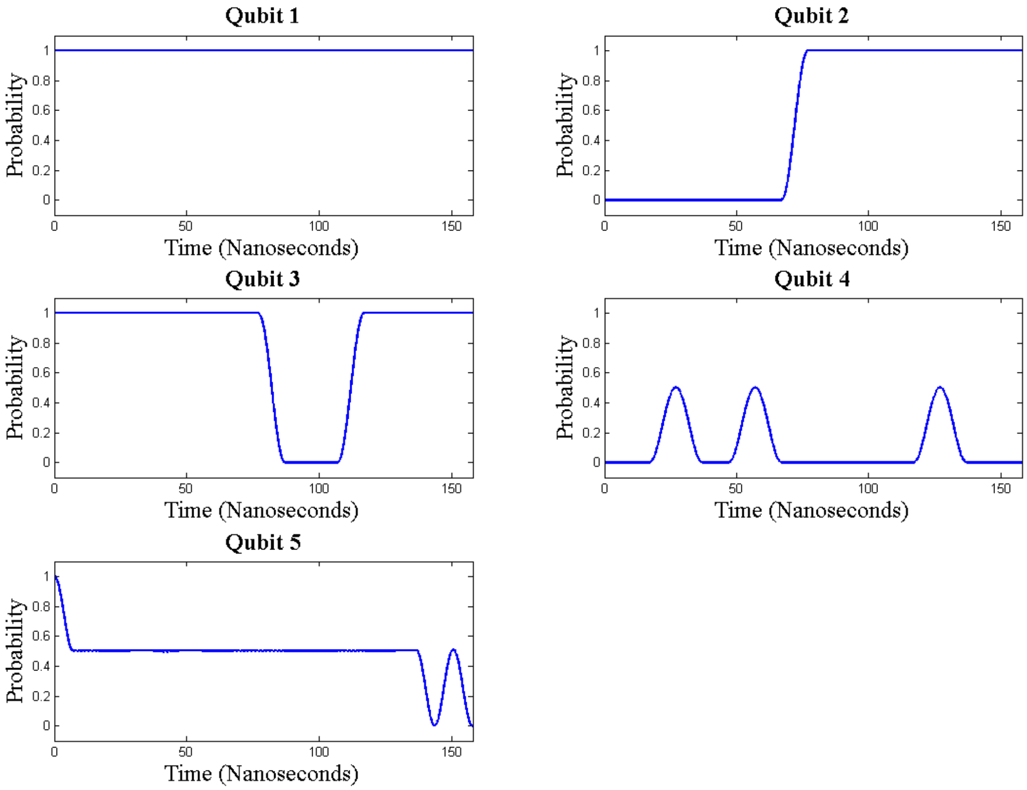

Figure 2.10. Simulation results for the multi-control-multi-target Toffoli gate discussed in section 2.7 with qubits 1 and 3 as the controls and qubits 2 and 5 as the targets

2.9 Multi-Control Toffoli Gates in 2D Architectures

All the reduction techniques discussed for 1D LNN arrays can be extended to 2D and 3D arrays. However, further simplification can be achieved in 2D and 3D arrays for specific arrangements of qubits. This can be done by implementing a $C^P(-I)$ gate, analogous to the $C^2(-I)$ gate, where P is the number of qubits coupled to an auxiliary qubit. As an example, consider Figure 2.11, which shows a 2D NN arrangement of qubits. Here, consider qubit D, which is directly coupled to its 4 neighbors. Suppose we want to implement a multi-control Toffoli gate ($C^3(X)$ gate) with qubits A, B and C as the controls and qubit E as the target. Since qubit D is coupled to all 4 participants (A, B, C and E), we can implement a $C^4(-I)$ gate between qubits A through E with qubits A, B, C, E as the controls, and qubit D as the target. Analogous to the $C^2(-I)$ gate, the $C^4(-I)$ gate is defined as:

$$C^4(-I)|ABCDE \; = \begin{cases} -|ABCDE \;,\; if\; |ABCE \; = |1111 \\ \;\;|ABCDE \;,\; if\; |ABCE \; \neq |1111 \end{cases} \tag{2.14}$$

where $|A$, $|B$, $|C$, $|D$ and $|E$ are the states of qubits. When all qubits are in the $|1$ state, a phase is applied on the target qubit.

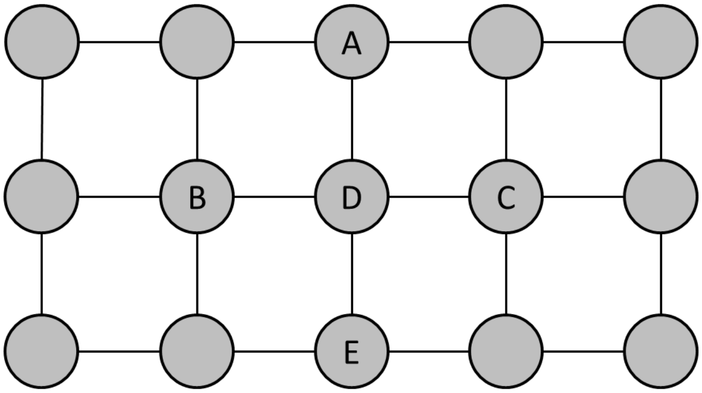

Figure 2.11. A 2D nearest neighbor arrangement of qubits. Each qubit only interacts with its adjacent neighbors

Now, if the $C^4(-I)$ gate is sandwiched between two Hadamard gates applied on qubit E, the desired $C^3(X)$ gate is realized. Using schemes like the pulsed-bias scheme [6], the $C^4(-I)$ gate can be implemented in a single step by making the bias on qubit D equal to the sum of all the 4 coupling terms for a time interval $T = 1/2\Delta$. Thus, in addition to extending the results of 1D LNN arrays to 2D arrays, further gate simplifications can be realized in 2D architectures for certain specific arrangements of qubits.

2.10 Conclusions

Quantum computing is not feasible without quantum error correction. Multi-control Toffoli gates are widely used in quantum error correcting codes, and efficient methods for implementing them in LNN arrays, on which most physical proposals of a quantum computer are built, need to be investigated. In this research, a new method to implement multi-control Toffoli gates in LNN arrays was presented. To this end, we introduced 3 lemmas that formed the basis for all our circuit reduction techniques. Using the lemmas, we first showed how to implement Toffoli gates in LNN arrays with different separations between the two control qubits and the target qubit. We extended our scheme to show how to implement multi-control Toffoli gates (with more than 2 controls) in LNN arrays with arbitrary separation between the controls and target. We used auxiliary qubits in our gate decomposition protocols, at the end of which, they were always restored back to their original states. In addition, for implementing multi-control Toffoli gate operations efficiently in LNN arrays, we showed how to extend our protocols to 2D arrays. We introduced a new gate, $C^4(-I)$ gate, analogous to the $C^2(-I)$ gate in 1D LNN arrays, using which we were able to implement a three-control-one-target Toffoli gate directly, without any interactions between the four qubits involved. The advantage of our work was that swap gates were not used to bring the qubits adjacent to each other. Instead, qubits other than control and target qubits were used to participate in gate operations. This decreased the gate count and overall computational overhead of our circuit.

2.11 References

[1] Nielsen, M. A., Chuang, I. L., "Quantum Computation and Quantum Information," *Cambridge University Press*, Cambridge, 2001.

[2] Makhlin, Y., Schon, G., Shnirman, A., "Quantum State Engineering with Josephson-junction Devices," *Rev. Mod. Phys.*, Vol. 73, 2001, pp. 357-400.

[3] Cirac, J. J., Zoller, P., "A Scalable Quantum Computer with Ions in an Array of Microtraps," *Nature*, 404, 2000, pp. 579-581.

[4] Gershenfeld, N. A., Chuang, I. L., "Bulk Spin Resonance Quantum Computation," *Science*, 275, 1997, pp. 350-356.

[5] Wei, L. F., Liu, Y., Nori, F., "Quantum Computation with Josephson Qubits Using a Current-biased Information Bus," *Phys. Rev. B*, Vol. 71, 2005, pp. 134506.1-134506.12.

[6] Gagnebin, P. K., Skinner, S. R., Behrman, E. C., Stek, J. E., Zhou, Z., Han, S., "Quantum Gates Using a Pulsed Bias Scheme," *Phys. Rev. A*, Vol. 72, 2005, pp. 042311-042320.

[7] Majer, J. B., Paauw, F. G., ter Harr, A. C. J., Harmans, C. J. P. M., Mooji, J. E., "Spectroscopy on Two Coupled Superconducting Flux Qubits," *Phys. Rev. Lett.*, 94, 2005, pp. 090501.1-090501.4.

[8] Maslov, D., "Linear Depth Stabilizer and Quantum Fourier Transformation Circuits with no Auxiliary Qubits in Finite Neighbor Quantum Architectures," *Phys. Rev. A*, Vol. 76, 2007, pp. 052310-052317.

[9] Shende, V. V., Bullock, S. S., Markov, I. L., "Synthesis of Quantum-logic Circuits," *IEEE Trans. CAD*, 25(6), 2006, pp. 1000-1010.

[10] Cheung, D., Maslov, D., Severini, S., "Translation Techniques Between Quantum Circuit Architectures," *Workshop on Quantum Information Processing*, 2007.

[11] Takahashi, Y., Kunihiro, N., Ohta, K., "The Quantum Fourier Transform on a Linear Nearest Neighbor Architecture," *Quantum Information and Computation*, Vol. 7, 2007, pp. 383-391.

[12] Kutin, S. A., "Shor's Algorithm on a Nearest-neighbor Machine," *Asian Conference on Quantum Information Science*, 2007.

[13] Choi, B. S., Van Meter, R., "On the Effect of Interaction Distance on Quantum Addition Circuits," *ACM Journal on Emerging Technologies in Computing Systems*, 7, No. 3, 11, 2011.

[14] Fowler, A. G., Hill, C. D., Hollenberg, L. C. L., "Quantum Error Correction on Linear Nearest Neighbor Qubit Arrays," *Phys. Rev. A*, Vol. 69, 2004, pp. 042314.1-042314.4.

[15] Möttönen, M., Vartiainen, J. J., "Decompositions of General Quantum Gates," *Trends in Quantum Computing Research*, NOVA Publishers, New York, 2006, Chapter 7.

[16] Chakrabarti, A., Sur-Kolay, S., "Nearest Neighbor Based Synthesis of Quantum Boolean Circuits," *Engineering Letters*, 15, 2007, pp. 356-361.

[17] Khan, M. H. A., "Cost Reduction in Nearest Neighbor Based Synthesis of Quantum Boolean Circuits," *Engineering Letters*, 16, 2008, pp. 1-5.

[18] Hirata, Y., Nakanishi, M., Yamashita, S., Nakashima, Y., "An Efficient Method to Convert Arbitrary Quantum Circuits to Ones on a Linear Nearest Neighbor Architecture," *International Conference on Quantum, Nano and Micro Technologies*, 2009, pp. 26-33.

[19] Kumar, P., Skinner, S. R., "Using Non-ideal Gates to Implement Universal Quantum Computing Between Uncoupled Qubits," *Quantum Information Processing*, Vol. 12 Issue 2, 2013.

[20] Maslov, D., Dueck, G. W., Miller, D. M., Negrevergne, C., "Quantum Circuit Simplification and Level Compaction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 27, 2008, pp. 436.

[21] Bullock, S. S., Markov, I. L., "Smaller Circuits for Arbitrary n-Qubit Diagonal Computations," *Quantum Information and Computation*, Vol. 4, 2004, pp. 27.

[22] Song, G., Klappenecker, A., "Optimal Realizations of Controlled Unitary Gates," *Quantum Information and Computation*, Vol. 3, 2003, pp. 139-155.

[23] Vatan, F., Williams, C., "Optimal Quantum Circuits for General Two-qubit Gates," *Physical Review A*, Vol. 69, 2004, pp. 032315.

[24] Maslov, D., Saeedi, M., "Reversible Circuit Optimization via Leaving the Boolean Domain," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, 2011, pp. 806-816.

[25] Banerjee, A., Pathak, A., "An Algorithm for Minimization of Quantum Cost," *Applied Mathematics and Information Sciences*, Vol. 6, 2012, pp. 157-165.

[26] Wille, R., Groβe, D., Dueck, G. W., Dreschler, R., "Reversible Logic Synthesis with Output Permutation," *International Conference on VLSI Design*, 2009, pp. 189-194.

[27] Hung, W. N. N., Song, X., Yang, G., Jang, Y., Perkowski, M., "Optimal Synthesis of Multiple Output Boolean Functions Using a Set of Quantum Gates by Symbolic Reachability Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, 2006, pp. 1652-1663.

[28] Gupta, P., Agarwal, A., Jha, N. k., "An Algorithm for Synthesis of Reversible Logic Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, 2006, pp. 2317-2330.

[29] Iwama, K., Kambayashi, Y., Yamashita, S. "Transformation-rules for Designing CNOT-based Quantum Circuits," *Proceedings of the Design Automation Conference*, New Orleans, Louisiana, USA, June 2002.

[30] Saeedi, M., Wille, R., Drechsler, R., "Synthesis of Quantum Circuits for Linear Nearest Neighbor Architectures," *Quantum Information Processing*, Vol. 10, 2011, pp. 355-377.

[31] Unruh W. G., “Maintaining Coherence in Quantum Computers,” *Phys. Rev. A*, Vol. 51, 1995, pp. 992-997.

[32] Chuang, I. L., Vandersypen, L. M. K., Zhou, X., Leung, D. W., Lloyd, S., “Experimental Realization of a Quantum Algorithm,” *Royal Society A*, 452, 1996, pp. 567-584.

[33] Proc., R., “Decoherence Limits to Quantum Computation Using Trapped Ions,” *Royal Society A*, 453, 1997, pp. 2017-2041.

[34] Chuang, I. L., Laflamme, R., Shor, P. W., Zurek, W. H., “Quantum Computers, Factoring, and Decoherence,” *Science*, 270, 1995, pp. 1633-1635.

[35] Preskill, J., “Reliable Quantum Computers,” *Royal Society A*, 454, 1998, pp. 385-410.

[36] Lo, H. K., “Introduction to Quantum Computation and Information,” *World Scientific*, Singapore, 1998.

[37] Steane, A. M., “Quantum Computing,” *Rep. Prog. Phys.*, 61, 1998, pp. 117-173.

[38] Shor P. W., “Scheme for Reducing Decoherence in Quantum Computer Memory,” *Phys. Rev. A*, Vol. 52, 1995, pp. R2493-R2496.

[39] Steane, A. M., “Efficient Fault-tolerant Quantum Computing,” *Phys. Rev. Lett.*, Vol. 77, 1996, pp. 793-767.

[40] Laflamme, R., Miquel, C., Paz, J. P., Zurek, W. H., “Perfect Quantum Error Correcting Code,” *Phys. Rev. Lett.*, Vol. 77, 1996, pp. 198.

[41] Bennett, C. H., DiVincenzo, D. P., Smolin, J. A., Wootters, W. K., “Mixed-state Entanglement and Quantum Error Correction,” *Phys. Rev. A*, Vol. 54, 1996, pp. 3824-3851.

[42] Kumar, P., “Efficient Quantum Computing Between Remote Qubits in Linear Nearest Neighbor Architectures,” *Quantum Information Process*, 12, 2013, pp. 1737-1757.

[43] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., Weinfurter, H., “Elementary Gates for Quantum Computation,” *Phys. Rev. A*, Vol. 52, 1995, pp. 3457-3467.

[44] Schuch, N., Siewert, J., “Natural Two-qubit Gate for Quantum Computation Using the XY Interaction,” *Phys. Rev. A*, Vol. 67, 2003, pp. 032301-032308.

CHAPTER 3

SINGLE-CONTROL-MULTI-TARGET TOFFOLI GATES IN LNN ARRAYS

Our method can be used for implementing single-control-multi-target Toffoli gates in an N qubit system. Here, if the total number of target qubits is l, then there are $l+1$ participants in the circuit. The limit for this method is the availability of at least one auxiliary qubit in the circuit, which requires $l+1 \leq N-1$. Figure 3.1 shows a single-control-multi-target Toffoli gate where qubit p is the control. If we use swap gates to apply this gate operation, $2 \times (N-2)$ swap gates and l CNOT gates are needed, and the quantum cost is $6 \times (N-2) + l$ CNOT gates (1 swap gate equals 3 CNOT gates).

To apply a single-control-multi-target Toffoli gate, the following are the procedures:

Step 1: a CNOT gate is applied on the auxiliary qubit, m, with qubit p as the control qubit.

Step 2: Implement CNOT gates on all the target qubits with qubit m as the control qubit. To decrease the depth, two CNOT gates can be realized in parallel.

Step 3: Repeat step 1.

Step 4: Repeat step 2.

For applying any remote CNOT gate with one control and one target qubit in the circuit, we use circuit reduction techniques discussed in [42]. The advantage of our method is that a pair of CNOT gates can be implemented in parallel. This can reduce the gate count of the system. The disadvantage of this method is when the NCC of qubits p and m is less than $N/2$ (qubit m is not in the middle of the circuit), the gate count increases, and compared to circuits with swap gates, this method may no longer be efficient, and new techniques need to be investigated (which is beyond the scope of this thesis). In the following, a circuit to implement a single-control-multi-target Toffoli gate in a 5 qubit system is shown.

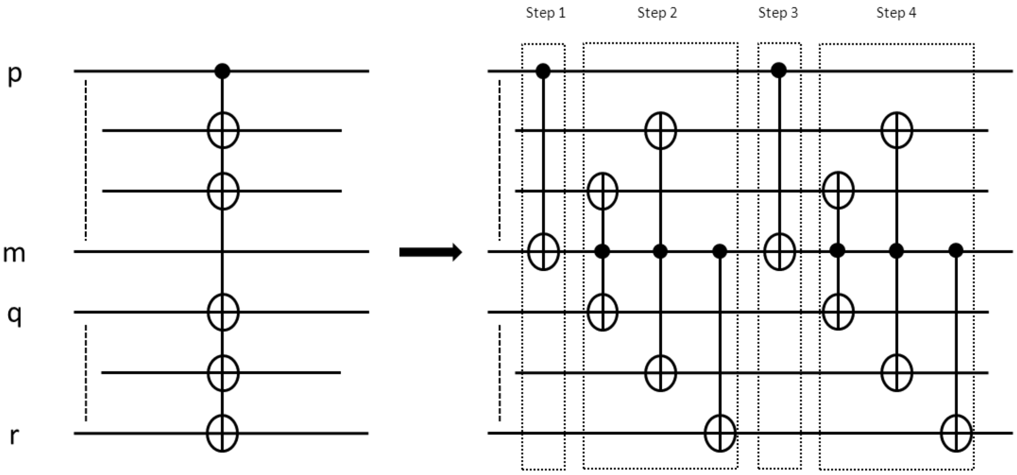

Figure 3.1. Implementation of a single-control-multi-target Toffoli gate in an N qubit system

Note that in the circuit reduction technique shown in Figure 3.1, the auxiliary qubit is located in the middle of the LNN array. The gate count may vary depending upon the location of the auxiliary qubit. Finding the most efficient circuit as a function of the position of the auxiliary qubit has been left as a future work.

As an example, Figure 3.2(a) shows a one-control-three-target Toffoli gate with four participants where qubit 1 is the control and qubits 2, 4 and 5 are the targets in the circuit. Qubit 3 is in the middle and can be used as the auxiliary qubit. To apply the desired gate, first a CNOT gate is implemented between qubits 1 and 3, where qubit 3 is the target. Then, two CNOT gates are applied on qubits 2 and 4, where qubit 3 is the control. Another CNOT gate is implemented between qubits 3 and 5, where qubit 5 is the target. Finally, we repeat all the steps as follows: one CNOT gate implemented between qubits 1 and 3, where qubit 3 is the target; two CNOT gates applied on qubits 2 and 4, where qubit 3 is the control, and one CNOT gate between qubits 3 and 5, where qubit 5 is the target.

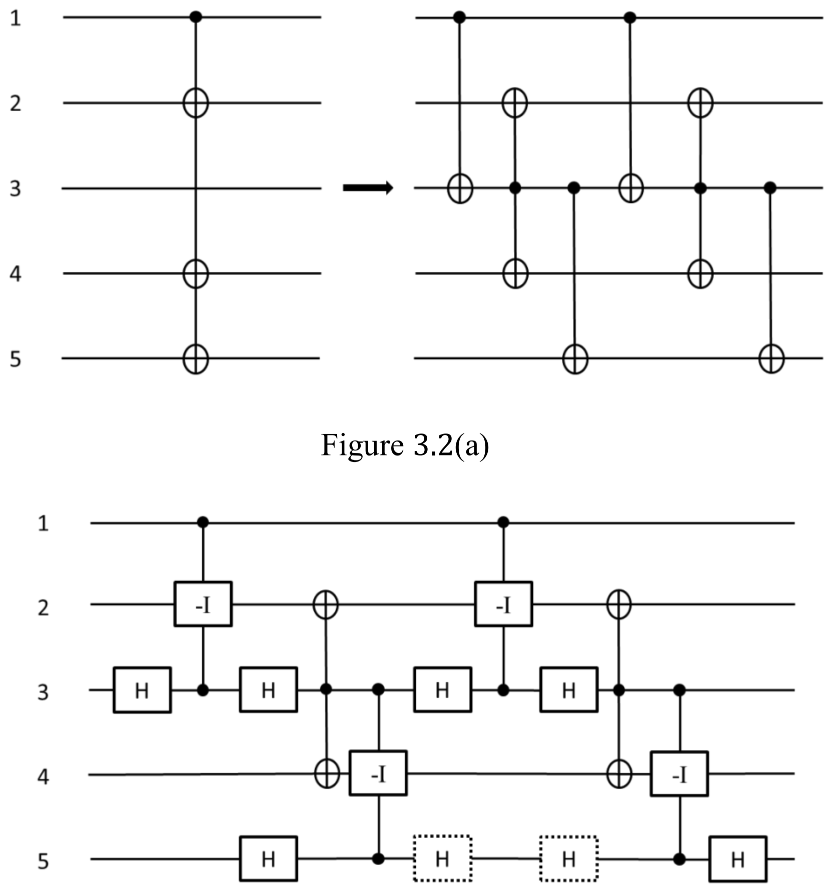

Figure 3.2(a)

Figure 3.2(b)

Figure 3.2. Implementation of a one-control-three-target Toffoli gate in a 5 qubit system. (a) Using steps 1, 2, 3 and 4 to decompose the desired gate into a sequence of CNOT gates. (b) The final circuit comprising universal gates

Figure 3.2(b) shows the final circuit with the elementary gate operations. In [42], the author showed how to implement a CNOT gate between two next-to-near qubits using three elementary gates. Note that two successive Hadamard gates applied on qubit 5 cancel out each other. The gate count for this single-control-multi-target Toffoli gate is 14, and the depth is 11. Using swap gates will increase the gate count to $6 \times (5 - 2) + 3 = 21$. In this example, if the auxiliary qubit is not in the middle of the circuit (was either qubit 2 or qubit 4), by using our method, the gate count and depth increase to 20 and 14, respectively.

CHAPTER 4

CONCLUSIONS

Efficient quantum algorithms rely on large scale quantum interference, which is sensitive to imprecision in the computer and to unwanted coupling between the computer and the rest of the world (noise and decoherence). Therefore, large scale quantum computation is practically impossible unless error correction methods are used. Multi-control Toffoli gates are widely used in quantum error correcting codes, and efficient methods for implementing them in LNN arrays, on which most physical proposals of a quantum computer are built, need to be investigated. In this thesis, a new method to implement multi-control Toffoli gates in LNN arrays was presented. To this end, we introduced 3 lemmas that formed the basis for all our circuit reduction techniques. Using the lemmas, we first showed how to implement Toffoli gates in LNN arrays with different separations between the two control qubits and the target qubit. We extended our scheme to show how to implement multi-control Toffoli gates (with more than 2 controls) in LNN arrays with arbitrary separation between the controls and target. We also proposed a new technique to realize a single-control-multi-target Toffoli gate in N qubit LNN arrays. Throughout this research, we used auxiliary qubits in our gate decomposition protocols, at the end of which, they were always restored back to their original states. In addition, for implementing multi-control Toffoli gate operations efficiently in LNN arrays, we showed how to extend our protocols to 2D arrays. We introduced a new gate, $C^4(-I)$ gate, analogous to the $C^2(-I)$ gate in 1D LNN arrays, using which we were able to implement a three-control-one-target Toffoli gate directly, without any interactions between the four qubits involved. The advantage of our work was that swap gates were not used to bring the qubits adjacent to each other. Instead, qubits other than control and target qubits were used to participate in gate operations. This decreased the gate count and overall computational overhead of our circuit.

One of the future works is to extend our method to implement multi-control Toffoli gates efficiently in 3D LNN architectures. Another area of research is finding the most efficient circuit in realizing single-control-multi-target Toffoli gates as a function of the position of the auxiliary qubit. In addition, implementation of two different controlled-unitary gate operations applied on two targets with the same control qubit can be investigated. For instance, in [47], the authors showed a method of generalization of Kitaev's phase estimation algorithm, where a controlled-*Z* and a controlled-*S* gate operations are applied on two targets with a shared control qubit. Figure 4.1 shows an example of the gate operation in [47]. A future work would be to investigate how to realize such gate operations efficiently in the minimal number of steps.

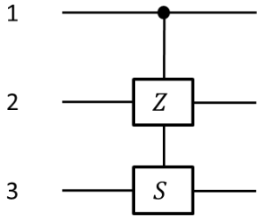

Figure 4.1. A controlled-*Z* gate and a controlled-*S* gate with a shared control qubit

REFERENCES

REFERENCES

[1] Nielsen, M. A., Chuang, I. L., "Quantum Computation and Quantum Information," *Cambridge University Press*, Cambridge, 2001.

[2] Makhlin, Y., Schon, G., Shnirman, A., "Quantum State Engineering with Josephson-junction Devices," *Rev. Mod. Phys.*, Vol. 73, 2001, pp. 357-400.

[3] Cirac, J. J., Zoller, P., "A Scalable Quantum Computer with Ions in an Array of Microtraps," *Nature*, 404, 2000, pp. 579-581.

[4] Gershenfeld, N. A., Chuang, I. L., "Bulk Spin Resonance Quantum Computation," *Science*, 275, 1997, pp. 350-356.

[5] Wei, L. F., Liu, Y., Nori, F., "Quantum Computation with Josephson Qubits Using a Current-biased Information Bus," *Phys. Rev. B*, Vol. 71, 2005, pp. 134506.1-134506.12.

[6] Gagnebin, P. K., Skinner, S. R., Behrman, E. C., Stek, J. E., Zhou, Z., Han, S., "Quantum Gates Using a Pulsed Bias Scheme," *Phys. Rev. A*, Vol. 72, 2005, pp. 42311-42320.

[7] Majer, J. B., Paauw, F. G., ter Harr, A. C. J., Harmans, C. J. P. M., Mooji, J. E., "Spectroscopy on Two Coupled Superconducting Flux Qubits," *Phys. Rev. Lett.*, 94, 2005, pp. 090501.1-090501.4.

[8] Maslov, D., "Linear Depth Stabilizer and Quantum Fourier Transformation Circuits with no Auxiliary Qubits in Finite Neighbor Quantum Architectures," *Phys. Rev. A*, Vol. 76, 2007, pp. 052310-052317.

[9] Shende, V. V., Bullock, S. S., Markov, I. L., "Synthesis of Quantum-logic Circuits," *IEEE Trans. CAD*, 25(6), 2006, pp. 1000-1010.

[10] Cheung, D., Maslov, D., Severini, S., "Translation Techniques Between Quantum Circuit Architectures," *Workshop on Quantum Information Processing*, 2007.

[11] Takahashi, Y., Kunihiro, N., Ohta, K., "The Quantum Fourier Transform on a Linear Nearest Neighbor Architecture," *Quantum Information and Computation*, Vol. 7, 2007, pp. 383-391.

[12] Kutin, S. A., "Shor's Algorithm on a Nearest-neighbor Machine," *Asian Conference on Quantum Information Science*, 2007.

[13] Choi, B. S., Van Meter, R., "On the Effect of Interaction Distance on Quantum Addition Circuits," *ACM Journal on Emerging Technologies in Computing Systems*, 7, No. 3, 11, 2011.

[14] Fowler, A. G., Hill, C. D., Hollenberg, L. C. L., “Quantum Error Correction on Linear Nearest Neighbor Qubit Arrays,” *Phys. Rev. A*, Vol. 69, 2004, pp. 042314.1-042314.4.

[15] Möttönen, M., Vartiainen, J. J., “Decompositions of General Quantum Gates,” *Trends in Quantum Computing Research*, NOVA Publishers, New York, 2006, Chapter 7.

[16] Chakrabarti, A., Sur-Kolay, S., “Nearest Neighbor Based Synthesis of Quantum Boolean Circuits,” *Engineering Letters*,15, 2007, pp. 356-361.

[17] Khan, M. H. A., “Cost Reduction in Nearest Neighbor Based Synthesis of Quantum Boolean Circuits,” *Engineering Letters*, 16, 2008, pp. 1-5.

[18] Hirata, Y., Nakanishi, M., Yamashita, S., Nakashima, Y., “An Efficient Method to Convert Arbitrary Quantum Circuits to Ones on a Linear Nearest Neighbor Architecture,” *International Conference on Quantum, Nano and Micro Technologies*, 2009, pp. 26-33.

[19] Kumar, P., Skinner, S. R., “Using Non-ideal Gates to Implement Universal Quantum Computing Between Uncoupled Qubits,” *Quantum Information Processing*, Vol. 12 Issue 2, 2013.

[20] Maslov, D., Dueck, G. W., Miller, D. M., Negrevergne, C., “Quantum Circuit Simplification and Level Compaction,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 27, 2008, pp 436-444.

[21] Bullock, S. S., Markov, I. L., “Smaller Circuits for Arbitrary n-Qubit Diagonal Computations,” *Quantum Information and Computation*, Vol. 4, 2004, pp. 27-47.

[22] Song, G., Klappenecker, A., “Optimal Realizations of Controlled Unitary Gates,” *Quantum Information and Computation*, Vol. 3, 2003, pp. 139-155.

[23] Vatan, F., Williams, C., “Optimal Quantum Circuits for General Two-qubit Gates,” *Physical Review A*, Vol. 69, 2004, pp. 032315.

[24] Maslov, D., Saeedi, M., “Reversible Circuit Optimization via Leaving the Boolean Domain,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, 2011, pp. 806-816.

[25] Banerjee, A., Pathak, A., “An Algorithm for Minimization of Quantum Cost,” *Applied Mathematics and Information Sciences*, Vol. 6, 2012, pp. 157-165.

[26] Wille, R., Groβe, D., Dueck, G. W., Dreschler, R., “Reversible Logic Synthesis with Output Permutation,” *International Conference on VLSI Design*, 2009, pp. 189-194.

[27] Hung, W. N. N., Song, X., Yang, G., Jang, Y., Perkowski, M., “Optimal Synthesis of Multiple Output Boolean Functions Using a Set of Quantum Gates by Symbolic Reachability Analysis,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, 2006, pp. 1652-1663.

[28] Gupta, P., Agarwal, A., Jha, N. k., “An Algorithm for Synthesis of Reversible Logic Circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, 2006, pp. 2317-2330.

[29] Iwama, K., Kambayashi, Y., Yamashita, S. “Transformation-rules for Designing CNOT-based Quantum Circuits,” *Proceedings of the Design Automation Conference*, New Orleans, Louisiana, USA, June 2002.

[30] Saeedi, M., Wille, R., Drechsler, R., “Synthesis of Quantum Circuits for Linear Nearest Neighbor Architectures,” *Quantum Information Processing*, Vol. 10, 2011, pp. 355-377.

[31] Unruh W. G., “Maintaining Coherence in Quantum Computers,” *Phys. Rev. A*, Vol. 51, 1995, pp. 992-997.

[32] Chuang, I. L., Vandersypen, L. M. K., Zhou, X., Leung, D. W., Lloyd, S., “Experimental Realization of a Quantum Algorithm,” *Royal Society A*, 452, 1996, pp. 567-584.

[33] Proc., R., “Decoherence Limits to Quantum Computation Using Trapped Ions,” *Royal Society A*, 453, 1997, pp. 2017-2041.

[34] Chuang, I. L., Laflamme, R., Shor, P. W., Zurek, W. H., “Quantum Computers, Factoring, and Decoherence,” *Science*, 270, 1995, pp. 1633-1635.

[35] Preskill, J., “Reliable Quantum Computers,” *Royal Society A*, 454, 1998, pp. 385-410.

[36] Lo, H. K., “Introduction to Quantum Computation and Information,” *World Scientific*, Singapore, 1998.

[37] Steane, A. M., “Quantum Computing,” *Rep. Prog. Phys.*, 61, 1998, pp. 117-173.

[38] Shor P. W., “Scheme for Reducing Decoherence in Quantum Computer Memory,” *Phys. Rev. A*, Vol. 52, 1995, pp. R2493-R2496.

[39] Steane, A. M., “Efficient Fault-tolerant Quantum Computing,” *Phys. Rev. Lett.*, Vol. 77, 1996, pp. 793-767.

[40] Laflamme, R., Miquel, C., Paz, J. P., Zurek, W. H., “Perfect Quantum Error Correcting Code,” *Phys. Rev. Lett.*, Vol. 77, 1996, pp. 198.

[41] Bennett, C. H., DiVincenzo, D. P., Smolin, J. A., Wootters, W. K., “Mixed-state Entanglement and Quantum Error Correction,” *Phys. Rev. A*, Vol. 54, 1996, pp. 3824-3851.

[42] Kumar, P., “Efficient Quantum Computing Between Remote Qubits in Linear Nearest Neighbor Architectures,” *Quantum Information Process*, 12, 2013, pp. 1737-1757.

[43] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., Weinfurter, H., “Elementary Gates for Quantum Computation,” *Phys. Rev. A*, Vol. 52, 1995, pp. 3457-3467.

[44] Schuch, N., Siewert, J., “Natural Two-qubit Gate for Quantum Computation Using the XY Interaction,” *Phys. Rev. A*, Vol. 67, 2003, pp. 032301-032308.

[45] Steane A. M., “A Tutorial on Quantum Error Correction,” *IOS Press*, 2006, pp. 1-32.

[46] Zhang, J., Laflamme, R., Suter, D., “Experimental Implementation of Encoded Logical Qubit Operations in a Perfect Quantum Error Correcting Code,” *Phys. Rev. Lett.*, 109, 2012, pp. 100503.1-100503.5.

[47] Higgins, B. L., Berry, D. W., Bartlett, S. D., Wiseman, H. M., Pryde, G. J., “Entanglement-free Heisenberg-limited Phase Estimation,” *Nature*, 450, 2007, pp. 393-396.

[48] Raussendorf, R., Browne, D. E., Briegel, H. J., “A One-way Quantum Computer,” *Phys. Rev. Lett.*, 86, 2001, pp. 5188-5191.

[49] Das, A., Chakrabarti, B. K., “Colloquium: Quantum Annealing and Analog Quantum Computation,” *Rev. Mod. Phys.*, 80, 2008, pp. 1061-1081.

[50] Nayak, C., Simon, S. H., Stern, A., Freedman, M., Sarma, S. D., “Non-abelian Anyons and Topological Quantum Computation,” *Rev. Mod. Phys.*, 80, 2008, pp.1083-1159.