Research in Industrial Projects for Students

Sponsor

IBM

Final Report

Improving Quantum Circuits of Toffoli Gates

Student Members

Muye "Willers" Yang (Project Manager), *MIT*
willers@mit.edu

Drew Gao, *Stanford University*

Xinjie He, *Carnegie Mellon University*

James Woodcock, *Texas A&M University*

Academic Mentor

Micky Abir, abir@g.ucla.edu

Sponsoring Mentors

Dmitri Maslov, Dmitri.Maslov@ibm.com

Sebastian Hassinger, Sebastian.Hassinger@ibm.com

Aug 19, 2021

This project was jointly supported by IBM and NSF Grant ⟨DMS⟩ 1925919.

Abstract

The goal of our project is to study possible ways to decompose the Multiple Control Toffoli gate (TOF^n), a gate that is important to implement efficiently for near-term quantum computers. TOF^n gates are analogous to the classical $(n-1)$-input AND gates. They are used in various error correction schemes and function as an important primitive for decomposing arbitrary multiple qubit gates into physically implementable circuits. However, despite its ubiquity, optimal decomposition of TOF^n is unknown in general. Our work focuses on improving the quantum circuits that implement TOF^n with respect to the number of controlled-not (CNOT) gates and T gates, which are the most difficult elementary gates to implement physically.

In this report, we show $2n-2$ and $\frac{3}{2}n-1$ lower bounds for the CNOT cost of relative phase Toffoli gates (a generalization of multiple control Toffoli gates), implemented with read only, and read-write memory respectively. We also provide a proof showing that the known implementation of $RTOF^4$ is CNOT-optimal. Finally, we provide a systematic construction of TOF^n which improves on existing implementations of TOF^n in terms of CNOT-cost, T-cost and Ancilla-count.

Acknowledgments

The RIPS program is funded by NSF and several industrial corporations through Institute of Pure and Applied Math (IPAM). We do appreciate the generous funding and the environment provided by IPAM. The project was made possible by the efforts of IPAM staff. We would also like to express our gratefulness to Dr. Serna for her help in the presentations, Mr. Hassinger for his works to coordinate the project with IBM Quantum, and Mr. Abir for his help to promote the project. Finally, we would like to thank Dr. Maslov especially for his clarifications of directions and his suggestions on our research.

Contents

Chapter 1

Introduction

1.1 Background

The field of quantum computing lies at the natural intersection of the related fields of computer science, information theory, and quantum mechanics. Quantum computing, as the name implies, is fundamentally governed by the laws of quantum mechanics. These laws, first presented mathematically by John von Neumann in 1932 [22], state that while quantum systems can potentially occupy multiple states simultaneously, in a phenomena known as superposition, these superpositions of states collapse into a single state upon observation or measurement.

The field of information theory—in particular quantum information theory—aims to examine the ways in which quantum systems may be used to store and transfer information. Although it may be self-evident that a quantum system in a superposition state contains more information than a classical system in a non superposition state, due to the fact that superpositions collapse upon measurement it is less obvious how the information stored in a superposition state can actually be used. In fact, Holevo 1973 [13] demonstrated that the amount of information which could be retrieved from a quantum system is no more than the information which is given by a particular observation of the system. Furthermore, Wootters and Zurek 1982 [24] demonstrated that quantum systems could not be copied.

Although it remained unclear at this point how quantum systems could improve upon classical computers, Paul Benioff [4] proved that a classical Turing machine could be implemented using quantum systems. This was the first demonstration that quantum systems could be used to perform any computation a classical computer could. Following this result, in 1982, Richard Feynman speculated that controlled quantum systems could be used to help simulate more complex quantum systems—a task which classical computers were known to be unable to do [8]. Soon thereafter David Deutsch (1985) formalized the notion of a universal quantum computer, or quantum Turing machine, using many of the axioms still in use today [6]. To summarize this formalism, quantum bits (or qubits) are two state quantum systems which can be mathematically represented as a vector given by linear combination of these states. Quantum logic gates then act on these qubits by manipulating these quantum systems. These transformations can be mathematically represented by unitary matrices.

Despite the groundwork of quantum computation having been laid, there were no concrete examples of problems which had an efficient solution on quantum computers but no such solutions on classical computers until the advent of the Deutsch–Jozsa algorithm in 1992 [7]. Although this algorithm has little practical use, it motivated further research into

quantum algorithms and complexity. Perhaps the most well-known, as well as one of the first quantum algorithms to have a practical use, was Shor's factoring algorithm (1994) [20]. This algorithm allows quantum computers to factor large semiprime numbers in polynomial time. Since many commonly used encryption methods rely on the difficulty of factoring large numbers, Shor's algorithm, once physically realizable at a large enough scale to factor these numbers, poses a threat to computer security.

Today, many quantum algorithms which outperform their classical counterparts exist, including Grover's search algorithm which performs an unstructured search on a database in $O(\sqrt{n})$ time (opposed to the classical equivalent which takes $O(n)$ time)[10]. More theoretical work has also been done to evaluate the benefit that quantum computers have over classical computers. In 1993 Ethan Bernstein Umesh Vazirani defined the class of problems which could be solved by quantum computers in polynomial time and demonstrated that any problems which could be solved by classical computers in polynomial time also belong to the aforementioned class of problems [5]. Furthermore, results such as Shor's algorithm have since shown that some NP (problems with solutions that can be verified in polynomial time without known polynomial time classical solutions) can be solved in polynomial time, giving evidence that quantum computers are more powerful than classical computers.

Clearly, quantum computers have huge potential to revolutionize the world of computation; however, there remain some physical limitations which prevent quantum computers from outperforming their classical counterparts in most practical applications. Despite the abundance of research done in academia and by industry leaders such as IBM, the technology required to build quantum computers is still in its infancy. Due to the necessary small size, and often extreme low temperature of components necessary to access quantum effects, fabrication of such components is difficult and imperfect. Moreover, the sensitivity of these components to external interference means that quantum computers will require robust error correction which requires multiple physical qubits to implement one usable logical qubit. Due to these physical limitations, current quantum computers are limited to fewer than 100 qubits, which severely hinders the practical use of quantum computers.

More challenges arise when trying to implement particular quantum algorithms. Due to the difficulty of directly implementing arbitrary, high-fidelity operations on quantum systems with multiple qubits, current implementations of quantum computers use some finite set of 1 and 2 qubit quantum gates, known as a universal gate set to approximate arbitrary operations. One such universal gate set, and the universal gate set used for the the results of this paper is the Clifford + T gate set. Even within the Clifford + T gate set, we see that the sole 2 qubit gate, the controlled not ($CNOT$), is more error prone and takes more time to execute than other Clifford + T gates[19]. Among the 1 qubit gates, the T gate (defined in section 2) is also expensive to implement fault-tolerantly. Thus when designing quantum circuits with the Clifford + T library , it is important to optimize them with respect to $CNOT$ gate count and T gate count.

However, it is difficult to determine when a particular implementation of a quantum gate is optimal. This is especially the case for gates with the potential to entangle qubits (i.e. modify the system so that the state of one qubit cannot be described independently of the others). One of the simplest and most commonly encountered multiple qubit entangling quantum gates is the 3-qubit Toffoli gate (TOF), and its n-qubit generalizations TOF^n, which can be thought of as a logic gate that implements $(n-1)$-input AND, bit-wise product of $n-1$ control qubits, and writes the output to a target qubit. The Toffoli gate is significant in that it is universal for classical computation, and coupled with the Hadamard gate it is universal for quantum computation. It is also used in error correction schemes, and is an important primitive for decomposing multiple-input quantum gates into 1 and 2

qubit gate circuit implementations.

Despite its relative simplicity and ubiquity, optimal implementations of Toffoli gates beyond the $n = 3$ case are unknown, and current implementations are believed to be further optimizable. Thus, the primary focus of our work is to determine which current implementations of Toffoli gates are optimal as well as to find improved implementations of Toffoli gates which are not optimal. Due to the high frequency in which Toffoli gates are used as intermediate steps when decomposing a many-qubit quantum operation into Clifford + T gates, improved implementations of Toffoli gates will in turn result in more optimized implementations of various other quantum operations.

1.2 Overview

We approached the topic of optimizing the multiple control Toffoli gate from a few perspectives. First, we showed general lower bounds for Toffoli gate variants implemented with both read-only and read-write memory. We then go on to give explicit constructions for $RTOF^n$ and demonstrate how these constructions can be used to implement Toffoli gates which offers practical improvements.

In particular, we focus on studying the lower and upper bound of n-qubit relative phase Toffoli Gates ($RTOF^n$) on ancilla-free quantum circuits with read-only memory (ROM). The relative phase implementation of TOF can be a powerful tool for optimizing its physical realizations. Efficient constructions of $RTOF^n$ can be used directly to build efficient TOF^n [15] when given access to ancillary qubits. Thus lower and upper bounds on the $CNOT$-cost of $RTOF^n$ directly imply related bounds on the $CNOT$-cost of TOF^n.

We begin our investigation by studying the the cost of implementing the n-qubit Toffoli gate (TOF^n), which is drastically reduced with the access to even one ancillary qubit. TOF gates are ubiquitous in quantum circuits, and significant effort into finding its efficient realizations point to an interesting time-space trade-off. With $\lceil \frac{n-3}{2} \rceil$ ancillary qubits available, the best known implementation uses $6n - 12$ $CNOT$ gates [15]. However, when we limit the number of ancillary qubit to just one, the best known implementation takes $12 + \mathcal{O}(1)$ $CNOT$s, although neither of these bounds are proven to be tight.

This report will be structured as follows. In the following chapters, we will first introduce the mathematical formalism of quantum circuits, gates and qubits, as well as some helpful definitions and lemmas used throughout the report. Then, our main results are collected under two sections. We will first show, in Chapter 4, various lower bounds on the CNOT-cost of $RTOF^n$ in different settings. These will be the first set of lower bounds known for $RTOF^n$ in literature. In Chapter 5, we will present a construction of $RTOF^n$ which gives rise to a family of circuits that implements TOF^n more efficiently than previous best known constructions in terms of CNOT-cost, T-cost and ancilla-count. Finally, our conclusion in chapter 6 will discuss open problems and future works.

Chapter 2

Rigor and Vigor

2.1 Hilbert Space Basics

The methods of quantum computing, at least of what will be discussed here, are primarily linear algebra and group theory. These two domains are unified by the notion of a Hilbert space, which is the setting of quantum computing.

Definition 1. *Let $\mathcal{H}$ be a vector space over the complex numbers with an inner product $\langle , \rangle$. We say $\mathcal{H}$ is a Hilbert space when $\mathcal{H}$ is complete with respect to the norm $\|v\| = \sqrt{\langle v, v \rangle}$.*

The power of Hilbert spaces comes from their inner product as they give geometric intuition for what is otherwise hard to visualize. The inner product generalizes the concepts of angles, parallel lines, orthogonality, and distance. Another feature of Hilbert spaces which is missing in generic vector spaces is that a Hilbert space is always naturally isomorphic to its dual.

Definition 2. *Let $\mathcal{H}$ be a Hilbert space. Then let $\mathcal{H}^\dagger$ denote the collection of continuous linear transformations from $\mathcal{H}$ to $\mathbb{C}$.*

The fact that $\mathcal{H}$ and $\mathcal{H}^\dagger$ are isometrically isomorphic is best seen using bra-ket notation. Given a fixed Hilbert space $\mathcal{H}$ let $|\psi\rangle$ denote a vector in $\mathcal{H}$ where ψ could be any handy index, such as Boolean bits. Inspired by the inner product, we can then define an element of $\langle\psi| \in \mathcal{H}^\dagger$ given by $|\varphi\rangle \mapsto \langle\psi|\varphi\rangle$, the inner product of $\langle\psi|$ and $\langle\varphi|$. The Riez-Representation theorem establishes that this correspondence between $\mathcal{H}$ and $\mathcal{H}^\dagger$ is bijective and in fact an isometric isomorphism.

Since Hilbert spaces is not only a vector space, it is also a metric space, which is why it is important to consider the linear automorphisms of $\mathcal{H}$ which are also an isometry. In fact, these linear transformations deserve a name.

Definition 3. *Let $\mathcal{H}$ be a Hilbert space and $T : \mathcal{H} \to \mathcal{H}$ a linear transformation. If for all $|\psi\rangle \in \mathcal{H}$, $\|T|\psi\rangle\| = \||\psi\rangle\|$, then T is said to be a unitary transformation.*

There is a more useful and purely algebraic characterization of unitary transformations. This characterization relies on the concept of the adjoint of a linear transformation.

Definition 4. *Let $\mathcal{H}$ be a Hilbert space and $T : \mathcal{H} \to \mathcal{H}$ a linear transformation. The adjoint of T, denoted $T^\dagger$, is the unique linear map satisfying $\langle Tv, w \rangle = \langle v, T^\dagger w \rangle$.*

Here are some useful facts about the adjoint.

Proposition 1. *Let $\mathcal{H}$ be a Hilbert space, $S, T : \mathcal{H} \to \mathcal{H}$ are linear transformations, and $\alpha, \beta \in \mathbb{C}$. Then*

- *There exists a unique adjoint for both T and S,*
- $T^{\dagger\dagger} = (T^\dagger)^\dagger = T$,
- $(\alpha S + \beta T)^\dagger = \overline{\alpha} S^\dagger + \overline{\beta} T^\dagger$,
- $(TS)^\dagger = S^\dagger T^\dagger$.

Proposition 2. *Let $\mathcal{H}$ be a Hilbert space, then the collection of unitary matrices is precisely the set of invertible transformations whose adjoint is its inverse, i.e. $TT^\dagger = T^\dagger T = I$.*

The collection of unitary transformations of a Hilbert space $\mathcal{H}$ form a group called the unitary group of $\mathcal{H}$ and is denoted by $\mathcal{U}(\mathcal{H})$.

Hilbert space theory is plenty of fun for the whole family, but we no longer need such generality. The prototypical Hilbert spaces are $\mathbb{C}^n$ for some $n \in \mathbb{N}$ and these are where quantum computing takes place. The inner product on $\mathbb{C}^n$ is given by

$$\langle v, w \rangle = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} \begin{pmatrix} \overline{w_1} \\ \vdots \\ \overline{w_n} \end{pmatrix} = v_1\overline{w_1} + \cdots + v_n\overline{w_n}.$$

Because of this, if $|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}$, then $\langle\psi| = \begin{pmatrix} \overline{\alpha_1} & \cdots & \overline{\alpha_n} \end{pmatrix}$. Since the norm $\|\psi\|^2 = \langle\psi|\psi\rangle$, then $\|\psi\|^2 = \sum_{i=1}^n |\alpha_i|^2$, which is the standard norm on $\mathbb{C}^n$.

We also have a convenient characterization of the adjoint of linear transformations from $\mathbb{C}^n \to \mathbb{C}^n$, since these correspond directly with complex $n \times n$ matrices. We will treat linear transformations and matrices as interchangeable objects from here on out.

Proposition 3. *Let $U \in \mathbb{C}^{n\times n}$, then the adjoint of U is the Hermitian conjugate of U, i.e. for all $1 \le i, j \le n$, $(U^\dagger)_{ij} = \overline{(U)_{ji}}$.*

We let $\mathcal{U}(n)$ be the unitary group of $\mathbb{C}^n$.

Definition 5. *The special unitary group is $\mathcal{SU}(n) = \{U \in \mathcal{U}(n) : \det(U) = 1\}$.*

This group is important to us for two reasons. First, it is not hard to convince yourself that all elements of $\mathcal{U}(n)$ can be expressed as a product of a unit scalar and a special unitary matrix. Second, $\mathcal{SU}(2)$ can be parameterized nicely which allows for simpler calculations.

Proposition 4. $\mathcal{SU}(2) = \left\{ \begin{pmatrix} x & y \\ -\overline{y} & \overline{x} \end{pmatrix} : x, y \in \mathbb{C},\ |x|^2 + |y|^2 = 1 \right\}$.

It is easy to see that the mapping $(x, y) \mapsto \begin{pmatrix} x & y \\ -\overline{y} & \overline{x} \end{pmatrix}$ is injective and surjective. Because of this, we can save time on writing out the whole matrix and represent $U \in \mathcal{SU}(2)$ as $[x, y]$. Since the mapping is a bijection we can define $[x, y][u, v] = [xu - y\overline{v}, xv + y\overline{u}]$.

2.2 The Tensor Product

What we have built up so far would suffice to study 1-qubit quantum systems, but unfortunately the complexity of the system grows exponentially as the number of qubits increase. To study these systems we need to appeal to Tensor products.

Definition 6. *Let V and W be complex vector spaces. Then the tensor product $V \otimes W$ is generated by elements of the form $v \otimes w$ with $v \in V$ and $w \in W$ subject to the relations*

- $v \otimes w_1 + v \otimes w_2 = v \otimes (w_1 + w_2)$,
- $(v_1 + v_2) \otimes w = v_1 \otimes w + v_2 \otimes w$,
- *For all $\alpha \in \mathbb{C}$, $(\alpha v) \otimes w = v \otimes (\alpha w) = \alpha(v \otimes w)$.*

For finite dimensional spaces, such as the ones we carry about, this product is very well behaved, despite being counter-intuitive.

Proposition 5. *Let V and W be complex vector spaces of dimensions n and m, with bases given by $\{e_1, \cdots, e_n\}$ and $\{f_1, \cdots, f_m\}$. Then $\{e_i \otimes f_j : 1 \leq i \leq n,\ 1 \leq j \leq m\}$ is a basis for $V \otimes W$.*

Fortunately, when we restrict our attention to $\mathbb{C}^n$ the tensor product of elements can be quite easy to calculate. First observe that $\mathbb{C}^n \otimes \mathbb{C}^n = \mathbb{C}^{n \times n}$, so intuitively, the tensor product of two vectors of the same dimension should result in a vector of length n^2.

Proposition 6. *Let $v, w \in \mathbb{C}^n$, then* $v \otimes w = \begin{pmatrix} v_1 w \\ \vdots \\ v_n w \end{pmatrix}$.

An important thing to note is that if v and w are both unit vectors, then $v \otimes w$ is as well. Recall that the space of $n \times n$ complex matrices is also a vector space so their is a notion of tensor products of matrices, except this product is called the Kroenecker product.

Definition 7. *Let $U \in \mathbb{C}^{m \times n}$ and $V \in \mathbb{C}^{p \times q}$, then $U \otimes V \in \mathbb{C}^{mp \times nq}$ and*

$$U \otimes V = \begin{pmatrix} u_{11}V & \cdots & u_{1n}V \\ \vdots & & \vdots \\ u_{m1}V & \cdots & u_{mn}V \end{pmatrix}.$$

The well behaved nature of the Kronecker product is partly responsible for making it possible to study quantum computing.

Proposition 7. *Let $A \in \mathbb{C}^{m \times n}, B \in \mathbb{C}^{n \times k}$, $U \in \mathbb{C}^{p \times q}, V \in \mathbb{C}^{q \times r}$, $x \in \mathbb{C}^n$ and $y \in \mathbb{C}^q$. Then*

- $(A \otimes U)(B \otimes V) = AB \otimes UV$,
- $(A \otimes U)(x \otimes y) = (Ax) \otimes (Uy)$,
- $(A \otimes U)^\dagger = A^\dagger \otimes U^\dagger$,
- $(A \otimes U)^{-1} = A^{-1} \otimes U^{-1}$.
- *If A and U are both unitary then $A \otimes U$ is as well.*

These properties are what makes studying quantum circuits so fruitful, but this will be shown in the next section. Since we have now built up enough groundwork to discuss particular gates and how to read circuits, we will move onto the basics of quantum computing.

2.3 Quantum Computing Basics

A n-qubit quantum computer takes in n-qubits and applies gates to these qubits and will then output an n-qubit after a measurement is applied to the quantum computer. This is the typical architecture of a quantum algorithm, yet there are exceptions.

Definition 8. *A qubit is a unit vector in $\mathbb{C}^2$ and the state of a single qubit quantum computer is a qubit. More generally, an n-qubit is a unit vector in $(\mathbb{C}^2)^{\otimes n}$ and the state of a n-qubit quantum computer is a n-qubit.*

As stated earlier, if $|\psi\rangle$ and $|\varphi\rangle$ are qubits then $|\psi\rangle \otimes |\varphi\rangle$ is a 2-qubit. The typical way to represent a 1-qubit is as a normalized superposition (linear combination) of $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. More explicitly, the set of states a 1-qubit can be is $\{\alpha|0\rangle + \beta|1\rangle : |\alpha|^2 + |\beta|^2 = 1\}$. This is where the computational aspect of quantum computing is apparent. The single bits 0 and 1 correspond to the orthonormal basis $|0\rangle$ and $|1\rangle$ and bit strings of length n correspond to the n-fold tensor product of $|0\rangle$ and $|1\rangle$.

Definition 9. *The computational basis for $(\mathbb{C}^2)^{\otimes n}$ is the collection $\{|x_1\rangle \otimes \cdots \otimes |x_n\rangle : x_i = 0$ or $x_i = 1\}$. Given a basis element $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$, we represent it as $|x_1 \cdots x_n\rangle$*

For example, $|0\rangle \otimes |1\rangle \otimes |0\rangle = |010\rangle$. The column vector which $|x_1 \cdots x_n\rangle$ represents is 0 on all components except for the component whose base two representation is $x_1 \cdots x_n$.

To measure the state of a quantum computer is the only irreversible operation in any quantum algorithm. We will first show how measurements work in the simple case of a single qubit quantum system. Suppose we have a qubit with a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then a measurement of $|\psi\rangle$ yields $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. This is well defined since the sum of these probabilities is 1 and both are positive. More generally, given an n-qubit with a state $|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$, then the probability of measuring the state to yield $|x\rangle$ is $|\alpha_x|^2$.

Recall that for any nonzero complex number α, there exists a positive real r and $\theta \in [0, 2\pi)$ such that $\alpha = re^{i\theta}$. This decomposition is unique and θ is the phase of α and r is the magnitude. Observe how the only relevant property of the coefficients of the computational basis representation of a state is the magnitude of the coefficient and not its phase. This important fact allows us to reduce the cost by implementing a circuit which has all the correct magnitudes but may differ with the phases.

Another key concept is entanglement.

Definition 10.

- *A state $|\psi\rangle$ of an n-qubit quantum computer is a product state if $|\psi\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$. Otherwise, $|\psi\rangle$ is said to be entangled.*
- *A unitary matrix $U \in \mathcal{U}(n)$ is said to be separable if $U = U_1 \otimes \cdots \otimes U_n$ and is said to be partially separable if $U = U_1 \otimes \cdots \otimes U_m$ for some $m \leq n$.*
- *$U \in \mathcal{U}(n)$ is said to be piece-wise separable if $U = VW$ where $V, W \in \mathcal{U}(n)$ are both partially separable.*

Definition 11. *We define a set of gates by the matrices they represent*

- *Rotation around x-axis:* $R_x(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$

- *Rotation around y-axis:* $R_y(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$
- *Rotation around z-axis:* $R_z(\theta) = \begin{pmatrix} e^{\frac{i\theta}{2}} & 0 \\ 0 & e^{-\frac{i\theta}{2}} \end{pmatrix}$
- *Pauli-X Gate:* $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- *Pauli-Y Gate:* $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
- *Pauli-Z Gate:* $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
- *CNOT Gate:* $CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ [1]
- *Hadamard Gate (H Gate):* $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- *T Gate:* $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}$

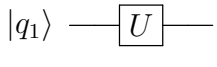
Definition 12. *We define the Controlled-Z Gate by the linear transformation it performs:* $CZ(a,b) : |a,b\rangle \mapsto (-1)^{ab}|a,b\rangle$. *Note that* $CZ(a,b) = CZ(b,a)$

Definition 13. *We define the Toffoli Gate by the linear transformation it performs:* $TOF(a,b,c) : |a,b,c\rangle \mapsto |a,b,ab \oplus c\rangle$

Definition 14. *We define* TOF^n *by the linear transformation it performs:* $TOF^n(x_1,...,x_{n-1},y) : |x_1,...,x_{n-1},y\rangle \mapsto |x_1,...,x_{n-1},x_1 \cdots x_{n-1} \oplus y\rangle$. *The* $n-1$ *qubits* $x_1,...,x_{n-1}$ *are called control qubits and* y *is called target qubit. Hence, the Toffoli Gate we defined in the previous definition (Definition 13) is a* TOF^3 *Gate with* 2 *controls and* 1 *target. the CNOT gate could be also treated as a special case of Toffoli Gate with 1 control qubit and 1 target qubit.*

2.4 Circuits

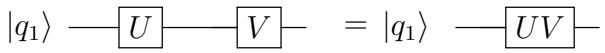
Circuits are used to visualize and manipulate quantum algorithms. The simplest circuit, besides the trivial one, is



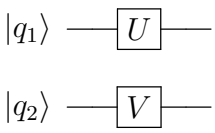
Circuits are read left to right and what the input is a qubit and as the qubit moves along the wire the gates which are on the wire are applied to the qubit. In this case, $|q_1\rangle$ is fed into the wire from the left and the output on the right would be the state, $U|q_1\rangle$.

[1] (In this report, $CNOT$ and CX are used interchangeably)

Because of this we have the following identity,



Things become more complicated with a 2-qubit algorithm. Suppose have the following circuit,



This is equivalent to the matrix form $(U|q_1\rangle) \otimes (V|q_2\rangle) = (U \otimes V)(|q_1\rangle \otimes |q_2\rangle)$. This works when both our state and and gates are separable. In the case of the $CNOT$ gate, things are not so easy. The $CNOT$ gate has a circuit symbol

Where the qubit is called the control qubit and the bottom qubit is called the target qubit. The $CNOT$ gate is not separable and so this gate entangles the two qubits.

Some gates are more difficult to implement than others. The TOF gate for example requires multiple $CNOT$ gates which are themselves difficult to implement. Hence, cost saving implementations of TOF are highly desirable even if some of the information is muddled. This is the motivating concept behind relative phase.

Definition 15. *Let A and B be matrices of the same dimension. We say that A is a relative phase of B if for all i, j, $|A_{ij}| = |B_{ij}|$. We say that A is a global phase of B if $A = \alpha B$ for some α with $|\alpha| = 1$.*

We denote the family of gates that are relative phases of TOF^n by $RTOF^n$. When we restrict our attention to $\mathcal{U}(n)$, relative and global phase both determine equivalence relations by A is equivalent to B if A is a relative, respectively a global, phase of B. But only equivalence up to global phase determines a congruence relation. This allows for great syntactic manipulation with global phase.

We define the following circuit primitives which we will use throughout this paper:

Definition 16. *We define a shorthand notation for the Margolus gate as in Figure 2.1. Note that Margolus Gate is a $RTOF^3$.*

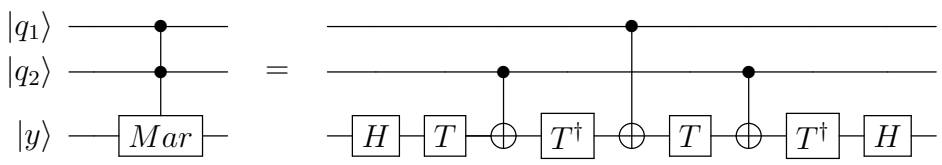


Figure 2.1: The Margolus Gate

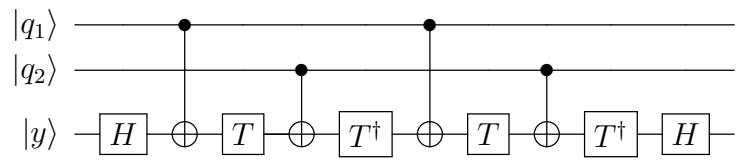


Figure 2.2: The CCiX Gate

Definition 17. *The CCiX gate is implemented as in Figure 2.2*

Definition 18. *We refer to the RTOF*[4] *gate presented by Maslov in [15] as the Mgate. The circuit implementation of Mgate is shown in Figure 2.3*

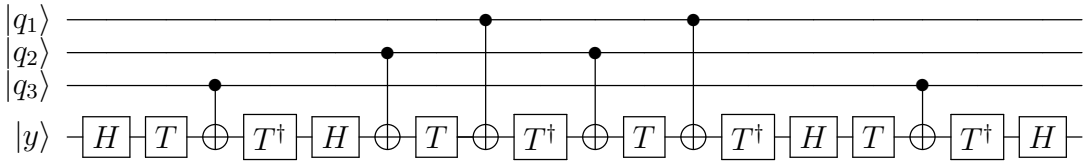
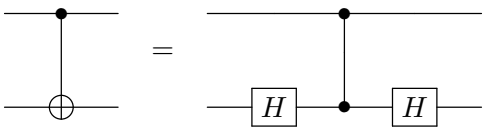


Figure 2.3: The Mgate

Definition 19. *For an arbitrary gate P that acts on qubits $q_1, ..., q_n$, we use this notation $P^{(q_1,...,q_n)}$ to specify that P operates on qubits $q_1, ..., q_n$. For example, if l is a qubit, we denote $Z^{(l)}$ to be a Z gate that acts on qubit l. Similarly, we define $CZ^{(i,j)}$ to denote that the CZ gate acts on qubit i and qubit j. For CX gate (i.e. CNOT gate), we use this notation $CX^{(i;j)}$ to denote that i is the control qubit and j is the target qubit. (Notice the semicolon in the superscript of $CX^{(i;j)}$ separates the control from the target. The qubit appear before the semicolon is the control and the qubit appear after the semicolon is the target)*

Remark 1. *$CX^{(i;j)}$ can be obtained by conjugating the $CZ^{(i;j)}$ with Hadamard Gates on both side, and similarly, $CZ^{(i;j)}$ can be obtained by conjugating the $CX^{(i;j)}$ with Hadamard Gates on both side.*



Remark 2. *Note that the CNOT gate together with single qubit unitary gates are universal. Hence, when we are considering circuit designs, we could restrict ourselves to the set of single qubit unitary gates and the CNOT gate. Furthermore, by Remark* **??**, *one can see that CZ gate together with single qubit unitary gates are also universal. Hence, we could similarly restrict ourselves to single qubit unitary gates and the CZ gate when considering circuit designs.*

Definition 20. *We define the notion of a CX circuit to be a circuit $\mathcal{L}$ such that CNOT gates are the only entangling gate between mulitple qubits. Similarly, define a CZ circuit to be a circuit $\mathcal{C}$ such that CZ gates are the only entangling gate between multiple qubits.*

Definition 21. *We define two notions of load (load factors) here.*

- *Let $\mathcal{L}$ be a CX circuit. Let l be a qubit in $\mathcal{L}$. We define the load (load factor) of l to be the number of $CX^{(l,\star)}$ gates (i.e. The number of CX gates that have l as the control qubit).*

- *Let $\mathcal{C}$ be a CZ circuit. Let q be a qubit in $\mathcal{C}$. We define the load (load factor) of q to be the number of CZ gates that are incident to q.*

Note that the notion of load (load factor) is different in different contexts.

Definition 22 (Ancillary Qubits)**.** *The ancilla in a circuit are extra qubits that help the computation. They are not involved in the logical operations, but they give extra space to perform computations. We also distinguish two types of ancilla.*

- *Clean Ancilla: clean ancilla are initially assumed to be all in states $|0\rangle$ before computations, and they should be returned to states $|0\rangle$ after the computations.*

- *Dirty Ancilla: dirty ancilla can be in any arbitrary states before computations, and they should be return to their initial states after the computations.*

Definition 23. *We define the Read-Only-Memory Model (ROM) and the Read-Write-Memory Model (R-W) with respect to TOF^n and $RTOF^n$. Let $\mathcal{C}$ be a circuit that computes TOF^n or $RTOF^n$ with $q_1, ..., q_{n-1}$ as control qubits and q_n as target qubit.*

- *We say that $\mathcal{C}$ is in Read-Only-Memory Model (ROM) if we never change the states of $q_1, ..., q_{n-1}$ in $\mathcal{C}$.*

- *We say that $\mathcal{C}$ is in Read-Write Memory Model (R-W) if we are allowed to change the states of $q_1, ..., q_{n-1}$ in $\mathcal{C}$.*

Definition 24. *Let A be a 2×2 matrix. We say A is sparse if A is a diagonal matrix or an anti-diagonal matrix.*

Definition 25. *We say that qubit l in a circuit $\mathcal{C}$ is sparse if all the single qubit unitaries on l can be assumed as diagonals.*

Remark 3. *If qubit l in circuit $\mathcal{C}$ is sparse and circuit $\mathcal{C}$ has at least two qubits, we know that all the single qubit unitaries on l are diagonals. Furthermore, we can merge all the diagonals into one diagonal matrix without affecting the computation.*

Definition 26. *We define $-q_i = \{q_1, ..., q_n\} \setminus \{q_i\}$. Hence, if we write $P^{(-q_i)}$, we mean that P operates on the set of qubits $\{q_1, ..., q_{i-1}, q_{i+1}, ..., q_n\}$.*

Definition 27. *Suppose $\mathcal{L}$ is a n-qubit network such that the only single qubit unitary acting on qubit l is a diagonal matrix D. Furthermore, assume that the CZ gates incident to l are $CZ^{(l,q_1)}, ..., CZ^{(l,q_n)}$ ($q_1, ..., q_n$ are not necessarily different). In other words, the circuit can be written as $D^{(l)} S_1^{(-l)} CZ^{(l,q_1)} S_2^{(-l)} CZ^{(l,q_2)} ... CZ^{(l,q_m)} S_m^{(-l)}$.*

By removing $l = |0\rangle$, we mean that we consider the circuit $\mathcal{C}_0$ in the $n-1$-qubit network without l defined by $\langle 0|D|0\rangle S_1^{(-l)} S_2^{(-l)} ... S_m^{(-l)}$. In other words, it is the sub-circuit of $\mathcal{L}$ if we input $q_0 = |0\rangle$ into the original circuit $\mathcal{L}$.

By removing $l = |1\rangle$, we mean that we consider the circuit $\mathcal{C}_1$ in the $n-1$-qubit network without l defined by $\langle 1|D|1\rangle S_1^{(-l)} Z^{(l)} S_2^{(-l)} Z^{(l)} ... Z^{(l)} S_m^{(-l)}$.

Remark 4. *Note that if $\mathcal{L}$ computes U. Let l be a sparse qubit and assume U commutes with $Z^{(l)}$. In other words, U can be written as $|0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes U_1$ Then the circuit $\mathcal{C}_0$ obtained from removing $l = |0\rangle$ computes U_0 and the circuit $\mathcal{C}_1$ obtained from removing $l = |1\rangle$ computes U_1.*

Chapter 3

Lower Bounds and Optimality

In this chapter, we will show $2n-2$ and $\frac{3}{2}n-1$ lower bounds on the $CNOT$-cost of ancilla free implementations of $RTOF^n$ with ROM and read-write memory respectively. Some corollaries include the optimality of some implementations of $RTOF^n$. We conjecture $3n-6$ to be the lower bound on the $CNOT$-cost of ancilla free implementations of $RTOF^n$ with ROM, and record our attempts in proving the conjecture as well as some partial results. Furthermore, we will also include a proof that $RTOF^4$ given by Maslov in 2016 [15] is optimal in $CNOT$-count.

3.1 $CNOT$-cost of $RTOF^n$ in ROM

We prove the following lower bound on the number of $CNOT$ gates required to implement $RTOF^n$ with ROM.

Theorem 1. *Any relative phase Toffoli gate with n-1 read-only control qubits and one target qubit cannot be implemented with fewer than* $2n-2$ $CNOT$ *gates for* $n \geq 4$

The proof of this theorem consists of two parts. We will first show a lower bound of $2n-3$, and then show that a circuit with exactly $2n-3$ $CNOT$s cannot implement $RTOF^n$.

3.1.1 Part 1: $2n-3$ $CNOT$s are necessary

To aid us in our proof we first make the following observation:

Observation 1. *Each qubit of the control group must have a load of at least* 1 *in order to implement a relative phase Toffoli.*

To see this, we assume for contradiction that we have a ROM $RTOF^n$ implementation with some qubit l of load 0. Then only single qubit unitaries act on l, which means the circuit is separable with respect to systems $\{l\}$ and $Q\backslash\{l\}$, contradicting the behavior of $RTOF^n$.

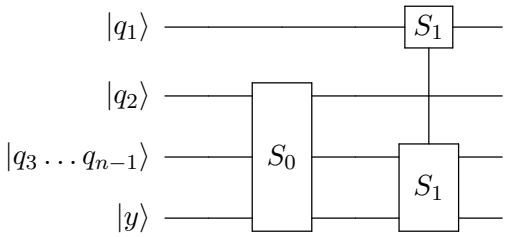
Observation 2. *Given a qubit* q_1 *in circuit* $\mathcal{C}$ *with Clifford + T gates. If all single qubit gates acting on* q_1 *is sparse and no CNOTs target* q_0*, then there exists a subcircuit* $C|_{|q_1\rangle}$ *on* $q_2, \ldots, q_n$ *which implements the action of* $\mathcal{C}$ *for each input state of* q_1*.*

Let input state be $|0\rangle$ or $|1\rangle$. Since all gates acting on q_1 is sparse, the state of q_1 will always be either $|0\rangle, |1\rangle$ with no superposition. In which case all $CNOT$s (if any) controlled by q_1 is equivalent to I or X on other qubits respectively, which gives the desired subcircuit. It follows by linearity that the observation hold for arbitrary input states.

Now we proceed with part one of the proof.

Lemma 1. *A piece-wise separable circuit cannot implement $RTOF^n$ in ROM for $n \geq 3$*

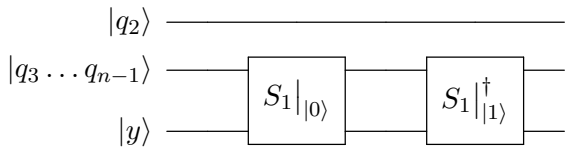
Proof. Note that any piece-wise separable circuit can be represented by the following diagram for $n \geq 3$:



Let $U_C = (\mathbb{I}^{(q_2)} \otimes S_1) \times (\mathbb{I}^{(q_1)} \otimes S_0)$ be the unitary operator describing the action of the previous circuit. Fixing $q_1 = |0\rangle$, we note that the sub-circuit consisting qubits $Q \backslash q_1$ implements the identity up to a relative phase, written as $U_0 = (\mathbb{I}^{(q_2)} \otimes S_1\big|_{|0\rangle})S_0 \cong \mathbb{I}$. Note this is possible by observation 2, where the conditions are trivially satisfied with ROM. Similarly, fixing $q_1 = |1\rangle$, we have a sub-circuit $U_1 = (\mathbb{I}^{(q_2)} \otimes S_1\big|)_{|1\rangle}S_0 \cong TOF^{n-1}$. We can compose a new circuit by applying the inverse of U_0 to U_1:

$$\begin{aligned} U_1 U_0^\dagger &= (\mathbb{I}^{(q_2)} \otimes S_1\big|_{|1\rangle})S_0[(\mathbb{I}^{(q_2)} \otimes S_1\big|_{|0\rangle})S_0]^\dagger \\ &= \mathbb{I}^{(q_2)} \otimes (S_1\big|_{|1\rangle} \times S_1\big|_{|0\rangle}^\dagger) \\ &\cong RTOF^{n-1} \times \mathbb{I}^{-1} = RTOF^{n-1}, \end{aligned}$$

which would be absurd, since q_2 have load 0, as illustraded below.



□

Remark 5. *Note that a similar proof also works for R-W Memory Model if one of the control qubits is sparse. We don't include the proof here because the idea is the same but the proof is more laborious.*

Corollary 1. *The number of $CNOT$ gates necessary to implement $RTOF^n$ with ROM is at least $2n-3$ when $n \geq 3$*

Proof. Assume for contradiction, there exists a circuit that implements $RTOF^n$ with less than $2n-3$ $CNOT$ gates
Then there are at least two qubits q_i, q_j with load factor 1 in the circuits. Note that q_i, q_j create a piece-wise separable circuit that was defined above, but by theorem 1, piece-wise separable circuits could never implement $RTOF^n$, resulting in a contradiction. □

Corollary 2. *Up to permutation of control qubits, and using ROM the only possible $CNOT$-structure of $RTOF^n$ with $2n-3$ $CNOT$ gates has load factor 1 on q_1, and load factor 2 on all the other control qubits $q_2, ..., q_{n-1}$. Furthermore, for each control qubit q_i with $2 \leq i \leq n-1$, there is a $CNOT$ with control on q_i operating before the $CNOT$ gate on q_1, and a $CNOT$ with control on q_i operating after the $CNOT$ gate on q_1*

Proof. Note that when the number of $CNOT$ gates in an $n-$qubit ROM Model is $2n-3$, there has to be at least one control qubit with load factor 1. Indeed, there is exactly one control qubit with load factor 1 (WLOG, call it q_1), and all the other control qubits have load factor 2.

Assume for contradiction, there is a qubit q_i, $2 \leq i \leq n-1$ with load factor 2 such that all the $CNOT$ gates on q_i operates either before the $CNOT$ gate on q_1 or after the $CNOT$ gate on q_1, then we have a piece-wise separable circuit where the piece-wise separability occurs on q_1 and q_i, a contradiction. □

We now conclude a lower bound of $2n-3$ on the $CNOT$-cost of $RTOF^n$ in ROM. To complete the proof, we need to show that no possible configurations of $2n-3$ $CNOT$ gates can implement $RTOF^n$.

3.1.2 Part 2: Exactly $2n-3$ $CNOT$s are not sufficient

To aid the second part of our proof, we show the following lemma. The proof to these lemmas can be found in the Appendix.

Lemma 2. *Let D be a diagonal matrix and U a unitary 2×2 matrix. If $X = UDU^\dagger$, where X is the Pauli-X gate, then $U = HZ$ up to a global phase.*

Lemma 3. *Suppose $(*)$ is a n-qubit ROM circuit with m $CNOT$s on the target qubit and the only other gates are single qubit unitaries acting on the target qubit. We denote the unitary before the first $CNOT$ as U_1 and the unitary after the first $CNOT$ as U_2 and so on until U_{m+1} after the final $CNOT$. Two additional assumptions are required.*

1. *There exists a qubit q_k with only one related $CNOT$.*
2. *There exists a second qubit q_l such that either to the left or to the right of the q_k, q_l is the control bit of a $CNOT$ gate acting on the target bit and is the only such $CNOT$ gate on that wire on that side of the q_k $CNOT$.*

Then $(*)$ cannot implement $RTOF^n$.

We observe that by corollary 2, any possible configurations of $2n-3$ $CNOT$ gates satisfy the conditions of theorem 3, and thus cannot implement $RTOF^n$.

The proof of theorem 1 is complete.

3.2 $CNOT$-cost of $RTOF^n$ in RW

For Read-Write Memory model in general, conditions of observation 2 is often not satisfied. For this reason, the proof we presented for theorem 1 cannot be extended and give us a similar bound. However, a lower bound can still be proved using a corollary of Markov and Shende's result.[18]

Theorem 2. *Any $RTOF^n$ circuit with Read-Write memory input requires at least $\frac{3}{2}n - 1$ $CNOT$ gates to implement.*

Proof. We proceed by induction on n. For the base case, we will prove in section 3.4 that $RTOF^4$ requires at least $6 > 5 = \frac{3}{2}4 - 1$ $CNOT$ gates to implement. The inductive hypothesis states that $RTOF^n$ requires at least $\frac{3}{2}n - 1$ $CNOT$ gates to implement. To show that $RTOF^{n+1}$ requires at least $\frac{3}{2}(n+1)-1$ $CNOT$s we consider a $n+1$ qubit circuit $\mathcal{C}$ with at most $\frac{3}{2}(n+1) - 2$ $CNOT$ gates. We now consider the generalized load, defined as the number of $CNOT$ gates target by or controlled by a given qubit. This induces a maximum total load of $3n - 1$ across $n + 1$ qubits. Note that the target qubit has to have at least load 1. Hence, the sum of loads across all the control qubits is at most $3n - 2$. Therefore, either two control qubits have loads 1, or there is a control qubit of load 2. In the former case, the circuit $\mathcal{C}$ is piece-wise separable. In the latter case, without loss of generality, the qubit is l. Note that we can assume that l is sparse by the result of Corollary 13 in [18].We remove $l = |1\rangle$ to obtain a circuit implementation of $RTOF^n$ which has less than $\frac{3}{2}(n + 1) - 1 - 2 = \frac{3}{2}n - \frac{3}{2} < \frac{3}{2}n - 1$ $CNOT$ gates, contradiction. □

3.3 Conjectures on $CNOT$-cost of $RTOF^n$

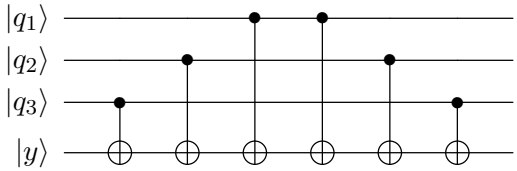
For the non-relative phase TOF^n case the following bound follows as a corollary of [18]:

Theorem 3. *Any ancillae free read-only memory implementation of TOF^n requires at least $3n - 6$ $CNOT$ gates.*

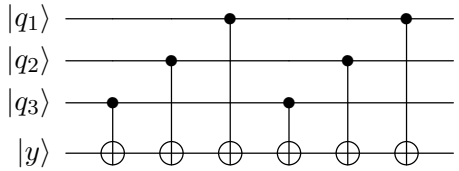
Proof. We proceed by induction on n. The base case follows from the fact that $3n-6 \leq 2n$ for $n \leq 6$. If we assume TOF^n cannot be implemented with $< 3n - 6$ $CNOT$ gates for some n, the existence of a TOF^{n+1} with fewer than $3n - 3$ gates would imply the existence of a TOF^n with fewer than $3n - 6$ gates as [18] guarantees a qubit of load ≥ 3 in the implementation of TOF^{n+1}. This qubit can be fixed at $|1\rangle$ resulting in a subcircuit implementing TOF^n with less than $3n - 6$ $CNOT$ gates. □

The following is an attempt at a proof of the same bound for $RTOF^n$ which relies on the following assumptions which we hope to prove with future work.

1. A circuit with the following $CNOT$ configuration cannot implement $RTOF$:



2. A circuit with the following $CNOT$ configuration cannot implement $RTOF$:



3. $RTOF^5$ requires at least 9 $CNOT$ gates to implement.

We can give the following conjectures if the assumptions hold.

Conjecture 1. *If the above assumption hold, then any circuit implementing $RTOF^n$ with $n \geq 6$ must have at least one control qubit of load at least 3.*

Proof. Assume for contradiction that we have a circuit C which implements $RTOF^n$ with all control qubits having load < 3. If more than one qubits have load 1, then the circuit is piecewise separable and cannot implement $RTOF^n$. Therefore, at most one qubit have load 1. Then it follows from Lemma 3 that each control qubit of C must have load 2. We also know that $CNOT$ gates must also appear in two groups such that the first $n-1$ $CNOT$s are controlled by each of the control qubits exactly once, since any configuration which doesn't satisfy this condition must be piecewise-separable and thus cannot implement $RTOF^n$.

Without loss of generality, we may arrange the wires such that the first $n-1$ $CNOT$s are given by $CX^{(q_{n-1};y)}, CX^{(q_{n-2};y)}, \ldots, CX^{(q_1;y)}$ and the last $n-1$ $CNOT$s are given by $CX^{(q_{\sigma(n-1)};y)}, CX^{(q_{\sigma(n-2)};y)}, \ldots, CX^{(q_{\sigma(1)};y)}$, where σ is a permutation of $n-1$ elements. Now, for $n \geq 6$, the Erdős–Szekeres theorem tells us that for all $\sigma \in S_{n-1}$ we must be able to identify $i < j < k$ such that $\sigma(i) < \sigma(j) < \sigma(k)$ or $\sigma(i) > \sigma(j) > \sigma(k)$. Letting $q_l = |1\rangle$ for $l \notin \{i, j, k\}$ and reducing $CNOT$s controlled by any q_l to its single qubit $X^{(y)}$ equivalent, we see that we are left with a subcircuit C' on $\{q_\sigma(i), q_\sigma(j), q_\sigma(k), y\}$. If $\sigma(k) < \sigma(j) < \sigma(i)$, then Assumption 1 tells us that C' cannot implement $RTOF^4$. Otherwise if $\sigma(k) > \sigma(j) > \sigma(i)$, then Lemma Assumption 2 tells us that C' cannot implement $RTOF^4$. Either way this is a contradiction since fixing m control qubits of $RTOF^n$ at $|1\rangle$ results in a subcircuit which implements $RTOF^{n-m}$. □

Conjecture 2. *If the above assumptions hold, then ancilla free read-only memory implementation of $RTOF^n$ for $n \geq 5$ requires at least $3n-6$ $CNOT$ gates.*

Proof. We proceed by induction on n. We assume our base case and the inductive hypothesis. Now assume for contradiction that we have a circuit C implementing $RTOF^{n+1}$ with fewer than $3(n+1)-6$ $CNOT$ gates. Lemma 1 tells us that C must have a qubit of load at least 3. Fixing this qubit at $|1\rangle$ results in a sub-circuit which implements $RTOF^n$ with fewer than $3n-6$ $CNOT$ gates, thus violating the inductive hypothesis. □

Although we are unable to verify these assumptions for general $RTOF^n$, we have proved a couple potentially useful partial results which we present below. In fact, assumption 1 and 2 hold for special type $RTOF^n$ with respect to the target[1].

First, we observe that if A is a relative phase of TOF, then $A = DTOF$ where D is a unitary diagonal.

Proposition 8. *Circuits with $CNOT$ structures in assumption 1 and 2 cannot implement a special type relative phase Toffoli gate, $SRTOF^4$, where $SRTOF^4 = DTOF^4$ and $D = D_1 \otimes I$.*

Proof. Since $RTOF^4$ is a 16×16 matrix, then $D_1 = diag\{\alpha_1, \cdots, \alpha_8\}$, so

$$D = \{\alpha_1, \alpha_1, \alpha_2, \alpha_2, \cdots, \alpha_8, \alpha_8\}.$$

Consider the first circuit and observe that when we fix the first three qubits as $|000\rangle$ and $|100\rangle$, we have,

$$\alpha_1 I = U_7 U_6 U_5 U_4 U_3 U_2 U_1 \text{ and,}$$

[1]The definition of special type $RTOF$ is given by Maslov in 2016 [15].

$$\alpha_5 I = U_7U_6U_5XU_4XU_3U_2U_1,$$

hence $U_4 = \alpha_1\alpha_5^{-1}XU_4X$. If we now fix the first three qubits as $|011\rangle$ and $|111\rangle$, we get the following equations,

$$U_7XU_6XU_5U_4U_3XU_2XU_1 = \alpha_4 I \text{ and,}$$
$$U_7XU_6XU_5XU_4XU_3XU_2XU_1 = \alpha_8 X$$

and since $U_4 = XU_4X$, we have $\alpha_1^{-1}\alpha_4\alpha_5 I = X$, i.e. $1 = 0$. Hence, this circuit cannot implement TOF.

The proof for the second circuit will proceed similarly. By fixing these qubits we will get the following constraints on the unitaries.

- $|100\rangle \implies \alpha_5 I = U_7XU_6U_5U_4XU_3U_2U_1,$
- $|011\rangle \implies \alpha_4 I = U_7U_6XU_5XU_4U_3XU_2XU_1,$
- $|010\rangle \implies \alpha_3 I = U_7U_6XU_5U_4U_3XU_2U_1,$
- $|101\rangle \implies \alpha_6 I = U_7XU_6U_5XU_4XU_3U_2XU_1,$
- $|001\rangle \implies \alpha_2 I = U_7U_6U_5XU_4U_3U_2XU_1,$
- $|110\rangle \implies \alpha_7 I = U_7XU_6XU_5U_4XU_3XU_2U_1$
- $|111\rangle \implies \alpha_8 X = U_7XU_6XU_5XU_4XU_3XU_2XU_1.$

As a consequence of this we have that,

$$I = U_7U_6U_5XU_4XU_4^\dagger XU_4XU_3U_2U_1 \implies U_4 = XU_4XU_4^\dagger XU_4X$$
$$\implies XU_4X = U_4XU_4^\dagger XU_4$$

This is equality up to a global phase, and since that is a congruence relation this is well defined. Similarly, we have

$$I = U_7XU_6XU_5U_4XU_4^\dagger XU_4U_3XU_2U_1 \implies U_7XU_6XU_5XU_4XU_3XU_2XU_1 = X,$$

since I and X are not equal up to a global phase we have a contradiction.

□

This proof is not replicable when we consider $RTOF$ since equality up to a relative phase is not a congruence relation, whereas equivalence up to a global phase is a congruence relation. We have been working our way around that by making use of two lemmas, the proof of which are included in the Appendix.

Lemma 4. *Let D_1 and D_2 be special unitary diagonals and U a unitary matrix such that $D_1 = UD_2U^\dagger$, all 2×2. Then one of the following must be true,*

1. *U is diagonal and $D_1 = D_2$,*
2. *U is anti-diagonal and $D_1 = D_2^\dagger$,*
3. *$D_1 = D_2 = \pm I$.*

Lemma 5. *Let D be a special unitary diagonal, U a unitary, and A a special unitary anti-diagonal. If $A = UDU^\dagger$ then $D = \pm iZ$.*

Furthermore, we also have some numerical evidence supporting these assumptions. We have written an exhaustive search that generates circuits with those specified $CNOT$ skeletons and arbitrary Clifford gates in between. In this search, T gates are added only in conjugate pairs around $CNOT$s, which changes the $CNOT$ to another Clifford Operation controlled-XS. We keep this heuristic to make computations efficient, as multiplying Clifford gates are inexpensive since we can find an explicit finite multiplication table. Indeed, in all efficient $RTOF$ circuits known, T gates always appear in conjugate pairs around $CNOT$s.

Conducting the search on the $CNOT$ structure in assumption 1 and 2 confirms that no circuit generated this way can implement $RTOF^4$. Furthermore, if we conduct the search with a smaller set of gates I, X, Y, X, H, S, no circuit generated this way can implement $RTOF^5$ with 8 $CNOT$s.

Furthermore, we notice that Assumption 3 is in fact the least significant assumption, since a $3n - const$ bound could still exist even if assumption 3 does not hold.

3.4 Optimality

The lower bound given by theorem 1 implies that $RTOF^4$ cannot be implemented in ROM with less than 6 $CNOT$s. Since the $Mgate$ uses exactly 6 $CNOT$s, an easy corollary is the optimality of the $Mgate$ in ROM. Similarly, by Corollary 1, the Margolus gate is also optimal in ROM.

One could hope that there might exists a better implementation of $RTOF^3$ and $RTOF^4$ in Read-Write memory model, since the ROM restriction is quite stringent. However, surprisingly, the optimality of Margolus gate in Read-Write model is well-known [21], suggesting that for the purpose of constructing $RTOF^n$, ROM and R-W could be equivalent. To add on to Song's result, we show a proof (in Appendix) that the $Mgate$ is in fact optimal in both computational models as well.

Theorem 4. *6 $CNOT$s are required to implement any Relative Phase Toffoli-4*

The optimality of $RTOF^4$ in general computational models is a significant result since the current best known implementations of TOF^n given access to unlimited ancilla [15] use $RTOF^4$ as a basic building block. Theorem 4 suggests that the construction given by Maslov is likely to be $CNOT$-optimal; or at least there's no obvious ways to improve with similar constructions.

Chapter 4

Upper bounds and Constructions

In this Chapter, we will begin by giving a construction of $RTOF^n$. When one clean ancilla is available, we will also give a construction of TOF^n with the aforementioned $RTOF^n$. These constructions, although asymptotically inefficient, give rise to a family of circuits implementing TOF^n more efficiently than current best known constructions for various small n. These improvements are shown in the table at the end.

Specifically, our construction of $RTOF^n$ in ROM without ancilla will have the following properties:

Theorem 5. *There is a construction of $RTOF^n$ in ROM without ancilla. Let $c(n)$ be the number of CNOTs used in this construction, $m = \lfloor \log_3 n - 1 \rfloor$, and $r = (n-1) - 3^m$. Then:*

$$c(n) = \begin{cases} (n-1)^{\log_3 6} & r = 0 \\ 6^m + (r)2^{m+1} & 0 < r \leq 3^m \\ (3r)2^m & r > 3^m. \end{cases}$$

Theorem 6. *In the same construction of $RTOF^n$, let $t(n)$ be the number of T gates used, $m = \lfloor \log_3 n - 1 \rfloor$, and $r = (n-1) - 3^m$. Then, $t(n) \leq \frac{8}{5}(6^m - 1) + r2^{m+2}$.*

Using this construction, when one ancilla is available, we also have the following two constructions of TOF^n:

Proposition 9. *When one clean ancilla is available, there exists a construction of TOF^n that uses at most $2(c(\lceil \frac{n}{3} \rceil + 1) + \frac{16}{3}n - 12$ $CNOT$ gates and at most $2(t(\lceil \frac{n}{3} \rceil + 1) + \frac{16}{3}n - 8$ T gates.*

Proposition 10. *When one dirty ancilla is available, there exists a construction of TOF^n that uses at most $2c(n-1) + 8$ $CNOT$ gates and at most $2t(n-1) + 6$ T gates.*

4.1 Constructing $RTOF^n$

Before beginning describing the construction, we will first show some helpful lemmas:

Lemma 6. *In Read-Only Memory Model, if there is a $RTOF^n$ with control qubits $q_1, ..., q_n$ and target qubit y, then one can obtain a $RTOF^{n+1}$ with control qubits $q_0, ..., q_n$ and target qubit y by replacing each $CX^{(q_1,y)}$ with $CCX^{(q_0,q_1,y)}$ or $CCiX^{(q_0,q_1,y)}$ (The circuit construction of $CCiX$ can be found in Figure 2.2)*

Proof. In the case we replace all the $CX^{(q_1;y)}$ with $CCX^{(q_0,q_1;y)}$, note that each occurrence of q_1 in the original circuit can be treated as q_0q_1 in the modified circuit. When any of the $CX^{(q_1,y)}$ is replaced with $CCiX^{(q_0,q_1;y)}$, each occurrence of q_0q_1 is either q_0q_1 or iq_0q_1, one can factor out the phase i to merge with the overall relative phase.[1] □

We divide the following lemma in three steps as a procedure to obtain $RTOF^5$, $RTOF^6$ and $RTOF^7$.

Lemma 7. *Consider a $RTOF^4$ in a 7-qubit network (pictured on the left side of Figure 4.1).*

Step 1: One can replace all the $CX^{(q_2;y)}$ with $Mar^{(q_1,q_2;y)}$ (Margolus Gate with control q_1, q_2 and target y) to obtain a $RTOF^5$ (as in Figure 4.1)

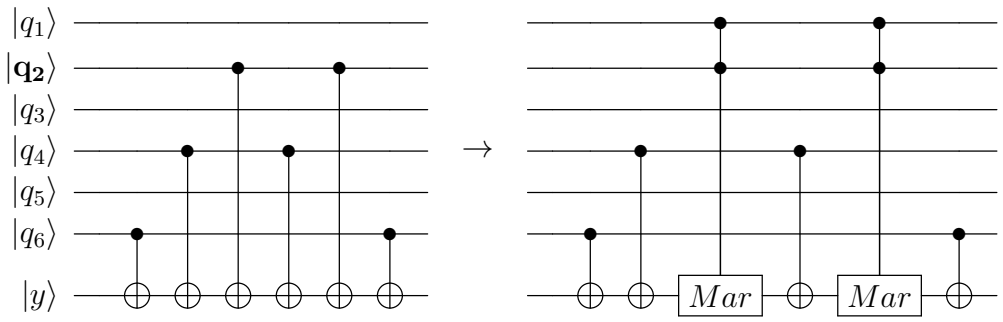


Figure 4.1: Extending $RTOF^4$ to $RTOF^5$ (Note that for simplicity, this and the following figures omit 1 qubit gates and only show the $CNOT$ structure

Step 2: One can furthermore replace all the $CX^{(q_4;y)}$ in the circuit for $RTOF^5$ with $Mar^{(q_3,q_4;y)}$ to obtain a $RTOF^6$ (as in Figure 4.2)

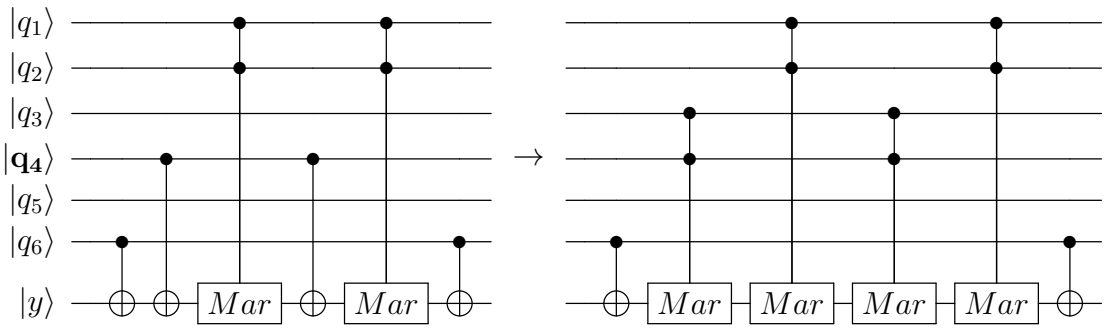


Figure 4.2: Extending $RTOF^5$ to $RTOF^6$

Step 3: One can finally replace all the $CX^{(q_6;y)}$ with $Mar^{(q_5,q_6;y)}$ to obtain a $RTOF^7$ (as in Figure 4.3)

[1]The original theorem and proof was first proposed by Maslov (2021)[2]. For completeness, we also present a proof here.

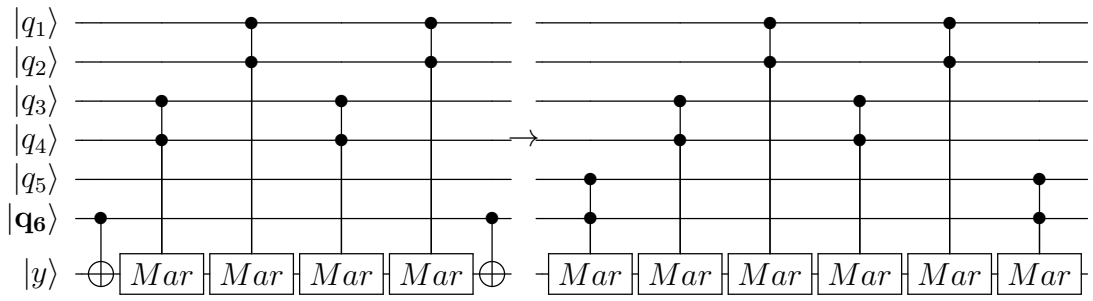


Figure 4.3: Extending $RTOF^6$ to $RTOF^7$

Proof. For simplicity, we denote q_i as i. Since we are working in ROM, all the single qubit unitaries in the circuit operate on the target qubit y. We denote U^y as U for single qubit unitaries in the circuit

We deal with Step 1 first. We would argue that the circuit implements an identity on $(q_4, q_6; y)$ up to a relative phase if and only if $|q_1q_2\rangle = |00\rangle, |01\rangle, |10\rangle$ and implements a $RTOF^3$ with control q_5, q_6 and target y if and only if $|q_1q_1\rangle = |11\rangle$

We denote the orginal circuit by $QCX^{(2;y)}SCX^{(2;y)}P$. Q, S, P are unitaries acting on qubits $4, 6$, and y. Then after the replacement of all the $CX^{(2;y)}$ with $Mar^{(1,2;y)}$, our circuit becomes $QMar^{(1,2;y)}SMar^{(1,2;y)}P$

Case 1: When $|q_1q_1\rangle = |00\rangle, |01\rangle$, we have $QMar^{(1,2;y)}SMar^{(1,2;y)}P$ functions as $QISIP$. Note that since $QCX^{(2;y)}SCX^{(2;y)}P$ is a circuit implementation of $RTOF^4$, we have $QISIP$ functions as a Relative Phase of identity.

Case 2: When $|q_1q_2\rangle = |10\rangle$, we have $Mar^{(1,2;y)}SMar^{(1,2;y)}$ functions as ZSZ. We expand S as the actual gates $TCX^{(4;y)}T^\dagger$. An easy calculation shows that $ZTCX^{(4;y)}T^\dagger Z = TCX^{(4;y)}T^\dagger$ when $q_4 = |0\rangle$ and $ZTCX^{(4;y)}T^\dagger Z = -TCX^{(4;y)}T^\dagger$ when $q_4 = |1\rangle$. Hence, we have $Mar^{(1,2;y)}S^{(4,5,6,7)}Mar^{(1,2;y)}$ functions as $IS^{(4,5,6,y)}I$ when $q_5 = |0\rangle$ and
$Mar^{(1,2;y)}S^{(4,5,6,y)}Mar^{(1,2;y)}$ functions as $-IS^{(4,5,6,y)}I$ when $q_4 = |1\rangle$. Note that since $QCX^{(2;y)}SCX^{(2;y)}P$ is a circuit implementation of $RTOF^4$, we have both $QISIP$ and $-QISIP$ function as a Relative Phase of identity.

Case 3: When $|q_3q_4\rangle = |11\rangle$, we have $Mar^{(1,2;y)}SMar^{(1,2;y)}$ functions as XSX.
Again, since $QCX^{(2;y)}SCX^{(2;y)}P$ is a circuit implementation of $RTOF^4$, we have $QXSXP$ a circuit implementation of $RTOF^3$.

Hence, by replacing all the $CX^{(2;y)}$ with $Mar^{(1,2;y)}$, our circuit implements an identity up to relative phase when $|q_1q_2\rangle = |00\rangle, |01\rangle, |10\rangle$ and a Toffoli-3 up to a relative phase when $|q_1q_2\rangle = |11\rangle$.These implies that our circuit implements a $RTOF^5$ with control qubits $1, 2, 4, 6$ and target qubit y

Step 2 and Step 3 uses a similar argument, so we leave readers to check them. □

Corollary 3. *Lemma 7 gives an implementation of a $RTOF^5$ which uses 10 $CNOT$s, and a $RTOF^6$ which uses 14 $CNOT$s, and a $RTOF^7$ which uses 18 $CNOT$s*

Now we move on to prove theorem 5:

Theorem 5. *There is a construction of $RTOF^n$ in ROM without ancilla. Let $c(n)$ be the number of $CNOT$s used in this construction, $m = \lfloor \log_3 n - 1 \rfloor$, and $r = (n-1) - 3^m$. Then:*

$$c(n) = \begin{cases} (n-1)^{\log_3 6} & r = 0 \\ 6^m + (r)2^{m+1} & 0 < r \leq 3^m \\ (3r)2^m & r > 3^m. \end{cases}$$

Proof. In the circuit implementation of MGate, we insert markers $m_{1,0}, m_{1,1}, ..., m_{1,6}$ right before and after each $CNOT$ (as in Figure 4.4). We would use the circuit structures between each pair of markers as invariants for induction.

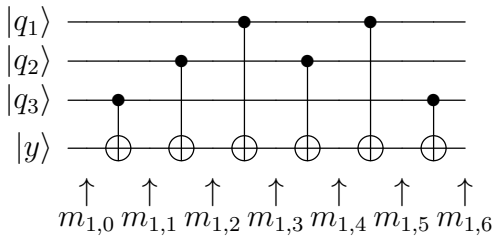


Figure 4.4: Mgate with markers between $CNOT$s

Step 1: We prove that the statement holds for $RTOF^n$ with $4 \leq n \leq 10$, and we furthermore show that in the circuit implementation of Relative Phase Toffoli-10, the gates between each adjacent pairs of markers implement an MGate exactly.

$RTOF^4$: Consider the MGate, it uses 6 $CNOT$s, and $c(4) = 6$. Note that each qubit has load 2.

$RTOF^5$: Apply Lemma 7 and Corollary 3, we have a circuit implementation of $RTOF^5$ which uses 10 $CNOT$s and $10 = c(5)$

$RTOF^6$: Apply Lemma 7 and Corollary 3, we have a circuit implementation of $RTOF^6$ which uses 14 $CNOT$s and $14 = c(6)$

$RTOF^7$: Apply Lemma 7 and Corollary 3, we have a circuit implementation of $RTOF^7$ which uses 18 $CNOT$s and $18 = c(7)$

$RTOF^8$: Apply Lemma 6 to replace the two CX with $CCiX$, we have a circuit implementation of $RTOF^7$ which uses 24 $CNOT$s and $24 = c(8)$

$RTOF^9$: Apply Lemma 6 to replace the two CX with $CCiX$, we have a circuit implementation of $RTOF^9$ which uses 30 $CNOT$s and $30 = c(9)$

$RTOF^{10}$: Apply Lemma 6 to replace the two CX with $CCiX$, we have a circuit implementation of $RTOF^{10}$ which uses 36 $CNOT$s and $36 = c(10)$

Notice that between each markers originally inserted, the gates implement a MGate exactly. Furthermore, each qubit has load 4.

Step k: We assume that we have a circuit implementation of $RTOF^{3^k+1}$. Each control qubit has load factor 2^k, and between each markers $m_{k-1,0}, ..., m_{k-1,6^{k-1}}$ inserted in Step $k-1$, there is a MGate exactly.

To simplify the indices, we consider this $RTOF^{3^k+1}$ in a $3^{k+1}+1$-network, and we treat qubits $q_3, q_6, q_9, ..., q_{3^{k+1}}$ as the control qubits of the $RTOF^{3^k}$, y as the target. Now we insert markers $m_{k,1}, ..., m_{k,6^k}$ to each $CNOT$s in this circuit.

Step k.r for r = 1
consider all the $CX^{3;y}$. Since each adjacent pairs of $CX^{3;y}$ can be found between markers $m_{k-1,2j-1}$ and $m_{k-1,2j}$, and we assume that the gates between $m_{k-1,2j-1}$ and $m_{k-1,2j}$ implement a $MGate^{3,6;y}$. By Lemma 7, if we replace all the $CX^{3;y}$ with $Mar^{2,3;y}$, the original occurrence of x_0x_1 becomes $\pm x_0x_1$. This increases the $CNOT$ costs by $2^k \times 2 = 2^{k+1}$. Hence, we have a valid implementation of $RTOF^{3^k+2}$ which uss $6^k + 2^{k+1}$ $CNOT$s, and note that $c(3^k + 1) = 6^k + 2^{k+1}$ exactly.

Step k.r for $2 \leq r \leq 3^k$
We repeat a similar procedure as Step k.1. Each time, we replace all the $CX^{(3r;y)}$ with $Mar^{(3r-1,3r;y)}$ as justified by Lemma 7, and this results in an increment of $2^k \times 2$ $CNOT$s per step. Hence, it is easy to see that when we finished with Step $3r$, we have an implementation of $RTOF^{3^k+r+1}$ that uses $6^k + r2^{k+1}$ $CNOT$s.

After Step $k.3^k$, we have an implementation of $RTOF^{2\times3^k+1}$ with control qubits $q_2, q_3, q_5, q_6, ..., q_{3^{k+1}-1}, q_{3^k}$ and target qubit y. Each control qubits $q_2, q_5, ..., q_{3^{k+1}-1}$ has load factor 2^k and each control qubits $q_3, q_6, ..., q_{3^k}$ has load factor 2^{k+1}, and the overall number of $CNOT$s used is $3^k \times 2^k + 3^k \times 2^{k+1} = 3 \times 6^k$. Note further that between each markers $m_{k,2j-1}$ and $m_{k,2j}$, we have a Margolus Gate

Step k.r for $r = 3^k + 1$
We replace all the $CX^{(2;y)}$ with $CCiX^{(1,2;y)}$ to obtain an implementation of a$RTOF^{2\times3^k+2}$, justified by Lemma 6. This results in an increment of 3×2^k $CNOT$s, so we have an implementation of a $RTOF^{2\times3^k+2}$ that uses $3 \times 6^k + 3 \times 2^k = 3(3^k + 1)2^k = (3r)2^k$ $CNOT$s

Step k.r for $3^k + 1 \leq r \leq 3^{k+1} - 3^k$
We replace all the $CX^{(3(r-3^k)-1;y)}$ with $CCiX^{(3(r-3^k)-2,3(r-3^k)-1;y)}$ to obtain an implementation of a $RTOF^{3^k+1+r}$, justified by Lemma 6. This results in an increment of 3×2^k $CNOT$s, so we have an implementation of a $RTOF^{(3^k+1+r)}$ that uses $(3(r-1))2^k+3\times2^k = (3r)2^k$ $CNOT$s.

After Step $k.3^{k+1}-3^k$, we have a $RTOF^{3^{k+1}+1}$ which uses 6^{k+1} $CNOT$s. Furthermore, each qubit has load factor 2^{k+1}, and the gates between markers $m_{k,2j-1}$ and $m_{k,2j}$ implement a MGate exactly. Hence, our induction invariant is preserved, and we are done.

□

Having proved our construction indeed implements $RTOF^n$ with $c(n)$ $CNOT$s, we move on to analyze the T-count of this circuit and prove theorem 6.

Theorem 6. *In the same construction of $RTOF^n$, let $t(n)$ be the number of T gates used, $s = \lfloor \log_3 n - 1 \rfloor$, and $r = (n-1) - 3^s$. Then,*[3] *$t(n) \leq \frac{8}{5}(6^s - 1) + r2^{s+2}$.*

Proof. Let $n = 3^s + r + 1$ where $s = \lfloor \log_3 n - 1 \rfloor$, and $r = (n-1) - 3^s$. We can see that each time we replace a $CNOT$ gate with either a Margolus gate or a CCiX gate, we will add 4 additional T gates to our circuit. Naively counting the number of T gates introduced at each step $k = 1, 2, ..., s-1$, we obtain a total number of

$$t(3^s + 1) < \sum_{k=0}^{s-1} 4 \times 3^k + 4 \times 3^k = \frac{8}{5}(6^s - 1)$$

[3]This bound is loose. In fact, the precise bound is $t(n) = \frac{38}{25}((n-1)^{\log_3 6} - 1) + \frac{2}{5}\log_3 n$ for when $n = 3^s + 1$ for some natural number s. The precise bound for general n can be characterized with an algorithm, which is not included in this report due to its length and ugliness.

T gates before we start step s. In the remaining steps $s.1, ..., s.r$, we notice that each time we "split" a qubit, we replace 2^s $CNOT$ gates with either a Margolus gate or a CCiX gate, which adds 4 T gate per replacement. Thus, we have the desired upper bound

$$t(n) < \frac{8}{5}(6^s - 1) + 4r \times 2^s = \frac{8}{5}(6^s - 1) + r \times 2^{s+2}.$$

We will sketch the proof for the precise bound. We can notice that some T gates do cancel out. Namely, we can notice that the second $CNOT$ in the MGate follows two single qubit gates $T^\dagger H$. Once we replace this $CNOT$ with either Margolus or CCiX, the left most two single qubit unitaries of the replacement is always HT, in which case a cancellation happens and our T-count is reduced by 2. We only add $CNOT$s of this form when replacing a $CNOT$ with CCiX, and once added cancellations will continue to happen for all subsequent replacements of this $CNOT$. With careful tracking of the amount of such $CNOT$s added each step and the cancellations in subsequent steps, we can derive the total number of cancellations to be

$$\sum_{k=0}^{s-1} 2 \times 6^k \times (s - k - 1) = \frac{2}{25}(6^s - 5n - 1).$$

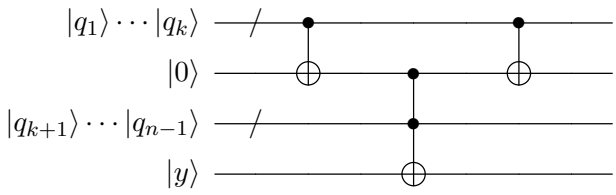
Subtracting these T gates from the total numbers of T gates added gives the desired bound. □

4.2 Constructing TOF^n with One Ancilla

We will now proceed to prove the claimed construction in proposition 9 and proposition 10.

Proposition 9. *When one clean ancilla is available, there exists a construction of TOF^n that uses at most $2(c(\lceil \frac{n}{3} \rceil + 1) + \frac{16}{3}n - 12$ $CNOT$ gates and at most $2(t(\lceil \frac{n}{3} \rceil + 1) + \frac{16}{3}n - 8$ T gates.*

Proof. First, we give a circuit that implements TOF^n with $\frac{32}{3}n + \mathcal{O}(1)$ $CNOT$ gates or T gates:



We notice that while implementing the two TOF^{k+1} gates, we can borrow the qubits $|q_{k+1}\rangle \cdots |q_{n-1}\rangle$ as dirty ancillary qubits. Since the computation done on the dirty ancillae is eventually uncomputed, this does not interfere with the functionality of the middle TOF^{n-k+1}. Similarly, we can use qubits $|q_1\rangle \cdots |q_k\rangle$ as dirty ancillae when computing TOF^{n-k+1}.

When there are $\lceil \frac{n-3}{2} \rceil$ dirty ancillae available, TOF^n gate can be implemented with $8n - 20$ $CNOT$s and $8n - 16$ T gates, as in proposition 11 [15]. It's easy to see that when choosing $k = \lceil \frac{n}{3} \rceil$, all Toffoli gates in the above construction have the required ancillae, thus the above construction uses at most $\frac{32}{3}n - 28$ $CNOT$ gates and at most $\frac{32}{3}n - 16$ T gates.

Furthermore, we can see that the first and third TOF^{k+1} can in fact be substituted with two $RTOF^{k+1}$ and $(RTOF^{k+1})^\dagger$,[15] the circuit complexity of which is bounded by

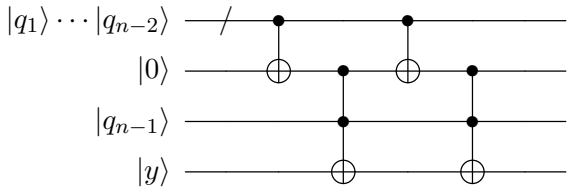
theorem 5 and 6. The relative phase introduced commutes through the middle Toffoli gate and eventually cancels each other out, keeping the controls unchanged. Then,

$$cost(TOF^n) = 2 \times cost(RTOF^{k+1}) + cost(TOF^{n-k+1}).$$

Choosing the same $k = \lceil \frac{n}{3} \rceil$, we have our claimed costs. □

Proposition 10. *When one dirty ancilla is available, there exists a construction of TOF^n that uses at most $2c(n-1)+8$ $CNOT$ gates and at most $2t(n-1)+6$ T gates.*

Proof. Clearly, the below construction implements TOF^n:



Now, replacing the top two TOF^{n-1} with our construction of $RTOF^{n-1}$, and the bottom two TOF^3 with the $CCiX$ gate gives the desired bound. □

We present a few current best known upper bounds to compare with our construction.

Proposition 11. *(Maslov, 2016) When $\lfloor \frac{n-3}{2} \rfloor$ dirty ancilla is available, TOF^n can be implemented with $8n-20$ $CNOT$ gates for $n \geq 5$.*

Proposition 12. *(Maslov, 2016) When one clean ancilla is available, TOF^n can be implemented with $12n-30$ $CNOT$ gates and $12n-24$ T-gates. When one dirty ancilla is available, TOF^n can be implemented with $16n-40$ $CNOT$ gates and $16n-32$ T-gates.*

Although asymptotically inefficient comparing proposition 10, the $CNOT$-cost and T-cost of our construction comes with a small leading coefficient, and thus gives practical improvements when n is small. To quantify these improvements, we record the $CNOT$-cost and T-cost of our TOF^n when different types of ancilla is available, and compare the cost to current best known constructions in the following tables. To avoid repetetion, the first line of each cell records cost of current best known constructions, whereas the second line of each cell records the cost our constructions. Furthermore, the lower cost is bolded.

	# $CNOT$	# Ancilla	Ancilla Type
	28	2	$\|xx\rangle$
TOF^6	28	**1**	$\|x\rangle$
	36	2	$\|xx\rangle$
TOF^7	36	**1**	$\|x\rangle$
	44	3	$\|xxx\rangle$
TOF^8	44	**1**	$\|x\rangle$

Table 4.1: When $n \leq 8$, our construction uses the same amount of $CNOT$ gates with less ancilla.

Ancilla	Clean Ancilla		Dirty Ancilla	
#Gates	$\#CNOT$	#T	$\#CNOT$	#T
	66	72	88	96
TOF^8	**40**	**48**	**44**	**66**
	78	84	104	112
TOF^9	**48**	**56**	**56**	**82**
	90	96	120	128
TOF^{10}	**56**	**68**	**68**	**98**
	102	108	136	144
TOF^{11}	**64**	**76**	**80**	**114**
	138	144	184	192
TOF^{14}	**88**	**108**	**128**	**186**

Table 4.2: When $8 \leq n \leq 14$, our construction uses less $CNOT$ gates and T gates when one ancilla of either type is available.

Ancilla	Clean Ancilla		Dirty Ancilla	
#Gates	$\#CNOT$	#T	$\#CNOT$	#T
	150	156	200	**208**
TOF^{15}	**96**	**116**	**144**	214
	210	216	280	288
TOF^{20}	**140**	**172**	**224**	374
	294	300	**392**	**400**
TOF^{27}	**204**	**244**	**392**	698
	330	336	**440**	**448**
TOF^{30}	**236**	**284**	472	710
	450	456	**600**	**608**
TOF^{40}	**348**	**440**	792	1206
	546	**552**	**728**	**736**
TOF^{48}	**428**	**552**	1048	1706
	570	**576**	**760**	**786**
TOF^{50}	**452**	592	1112	1834
	930	**936**	**1240**	**1248**
TOF^{80}	**844**	1072	2456	3754
	1158	**1164**	**1544**	**1552**
TOF^{99}	**1140**	1464	3624	5483
	1170	**1176**	**1560**	**1568**
TOF^{100}	1172	1512	3688	5578

Table 4.3: When one clean ancilla is available, our construction uses less $CNOT$ gates for $n < 100$ and less T gates for $n < 48$. When one dirty ancilla is available, our construction uses less $CNOT$ gates for $n < 27$ and less T gates for $n < 15$. Our construction is inefficient for $n \geq 100$.

Chapter 5

Conclusion and Future Works

This report lays out several foundations for the study of the costs of Relative Phase Toffoli Gates. We characterize costs in terms of the number of $CNOT$ gates used in the circuit and are able to derive lower bounds and upper bounds for the costs of Relative Phase Toffoli Gates. Specifically, we are able to prove a $\frac{3}{2}n - 1$ lower bound on $CNOT$ costs of $RTOF^n$ in R-W model and a $2n-2$ lower bound on $CNOT$ costs of $RTOF^n$ in ROM model. In the meantime, we are also able to prove the optimality of $RTOF^4$ in R-W Memory. This implies that the $\frac{3}{2}n-1$ lower bound could be further improved. We also present some partial results that could lead to $3n+$ Constant lower bound on the $CNOT$-cost of $RTOF^n$. Furthermore, one could also categorize the Relative Phase Toffoli Gates by the degree of freedoms they have in their relative phases. These partial results give a full proof of a $3n+$ constant lower bound for a class of Relative Phase Toffoli-n Gates where the relative phases have 2^{n-1} degrees of freedom. With these partial result, it is promising to extend this lower bound to general Relative Phase Toffoli-n Gates.

We are also able to derive an upper bound of $(n-1)^{\log_3 6}$ on $CNOT$ costs of $RTOF^n$ with explicit constructions. This construction of $RTOF^n$ could be directly used to implement TOF^n. Although asymptotically inefficient compared to known linear bounds, our constructions outperform current best-known implementations of TOF^n in terms of $CNOT$ cost, T cost, and ancilla count. With 1 clean ancilla, our construction offers improvements in terms of $CNOT$ Costs when $n < 100$, and in terms of T Costs when $n < 48$. These practical improvements are significant in the context of near-term quantum computer hardware, where high fidelity qubits and fault-tolerant $CNOT$ gates and T gates are difficult to realize. In the meantime, current best-known upper bounds on $CNOT$ costs for Relative Phase Toffoli Gates are asymptotically the same as those for Toffoli Gates. However, given the additional degrees of freedom, and the fact that one could construct a TOF^n using two $RTOF^n$ and 1 additional $CNOT$ with 1 ancilla, one would hope that there is a construction for $RTOF^n$ that has linearly less $CNOT$ costs than TOF^n.

Another implication of our result is the potential equivalence of ROM and R-W for $RTOF^n$, which is manifested when n is small. It was previously known that 3 $CNOT$s are required to implement a $RTOF^n$ in both R-W and ROM. Interestingly, the well-known optimal implementation of $RTOF^3$, the Margolus gate, is in ROM. Furthermore, current best known implementation of $RTOF^4$ is also in ROM and uses 6 $CNOT$s. We are able to prove that 6 $CNOT$s are required even if in R-W Memory. The optimality of $RTOF^4$ is an interesting result in its own right since $RTOF^4$ is used as a basic building block in current best known implementations of Toffoli gates; however, it also suggests that the ability to implement optimal $RTOF$ in ROM might not be a small circuit coincidence.

Chapter 6

Appendix

6.1 Lemmas for Lower Bounds

Lemma 2. *Let D be a diagonal matrix and U a unitary 2×2 matrix. Then if $X = UDU^\dagger$, where X is the Pauli-X gate, then $U = HZ$ up to a global phase.*

Proof. We shall first show this is true for special unitary matrices. We know that $D = \left(\begin{smallmatrix} \alpha & 0 \\ 0 & \beta \end{smallmatrix}\right)$ for $\alpha, \beta \in \mathbb{C}$. Since U is special we know there exists $x, y \in \mathbb{C}$ with $|x|^2 + |y|^2 = 1$ such that $U = \left(\begin{smallmatrix} x & y \\ -\overline{y} & \overline{x} \end{smallmatrix}\right)$. Putting it altogether we get the following equation,

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} x & y \\ -\overline{y} & \overline{x} \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} \overline{x} & -y \\ \overline{y} & x \end{pmatrix} = \begin{pmatrix} \alpha|x|^2 + \beta|y|^2 & -\alpha xy + \beta xy \\ -\alpha\overline{xy} + \beta\overline{xy} & \beta|x|^2 + \alpha|y|^2 \end{pmatrix}.$$

Which gives us the following system of equations,

1. $0 = \alpha|x|^2 + \beta|y|^2$
2. $1 = -\alpha xy + \beta xy$
3. $1 = -\alpha\overline{xy} + \beta\overline{xy}$
4. $0 = \beta|x|^2 + \alpha|y|^2$.

We know that neither α nor β are zero, as otherwise it wouldn't be invertible, so equations (1) and (4) tells us,

$$\frac{-\alpha}{\beta} = \frac{|y|^2}{|x|^2} \text{ and } \frac{-\beta}{\alpha} = \frac{|y|^2}{|x|^2} \implies -\alpha = \beta \text{ and } |x| = |y|.$$

From equations (2) and (3) we now have $2xy = \frac{1}{\beta}$ and $2\overline{xy} = \frac{1}{\beta}$, hence β and α are real. Since D must also be unitary we can assume $D = Z$. Then $xy = \frac{-1}{2}$. Since $|x|^2 + |y|^2 = 1$ and $|x| = |y|$, then $|x| = \frac{1}{\sqrt{2}}$ and thus $y = -\overline{x}$. So there exists $\theta \in [0, 2\pi)$ such that $x = \frac{1}{\sqrt{2}}e^{i\theta}$ and $y = -\frac{1}{\sqrt{2}}e^{-i\theta}$. Hence,

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} e^{i\theta} & -e^{-i\theta} \\ e^{i\theta} & e^{-i\theta} \end{pmatrix} \text{ and } U^\dagger = \frac{1}{\sqrt{2}} \begin{pmatrix} e^{-i\theta} & e^{-i\theta} \\ -e^{i\theta} & e^{i\theta} \end{pmatrix},$$

so

$$X = UZU^\dagger = \frac{1}{2} \begin{pmatrix} e^{i\theta} & -e^{-i\theta} \\ e^{i\theta} & e^{-i\theta} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} e^{-i\theta} & e^{-i\theta} \\ -e^{i\theta} & e^{i\theta} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 1 + e^{2i\theta} \\ 1 + e^{2i\theta} & 0 \end{pmatrix}$$

Hence, $1 + e^{2i\theta} = 2$ and thus $\theta = 0$ or π but we shall just choose $\theta = 0$ as they are the same up to a global phase. Plugging $\theta = 0$ into our original equations yields,

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = HZ.$$

With the special case proven, now suppose U is a general unitary. Then $U = \gamma V$, where V is a special unitary and $\gamma \in \mathbb{C}$ is a unit vector. Then

$$X = UDU^\dagger = (\gamma V)D(\gamma V)^\dagger = \gamma\overline{\gamma} VDV^\dagger = VDV^\dagger.$$

By what we have previously shown, $U = \gamma HZ$, which is merely a global phase of HZ. $\square$

Lemma 3. *Suppose $(*)$ is a n-qubit ROM circuit with m CNOTs on the target qubit and the only other gates are single qubit unitaries acting on the target qubit. We denote the unitary before the first CNOT as U_1 and the unitary after the first CNOT as U_2 and so on until U_{m+1} after the final CNOT. Two additional assumptions are required.*

1. *There exists a qubit q_k with only one related CNOT.*
2. *There exists a second qubit q_l such that either to the left or to the right of the q_k, q_l is the control bit of a CNOT gate acting on the target bit and is the only such CNOT gate on that wire on that side of the q_k CNOT.*

Then $(*)$ cannot implement $RTOF^n$.

Proof. Assume $(*)$ satisfies all of the hypothesis of the theorem and implements $RTOF^n$. Also assume that the $CNOT$ controlled by q_k is in between U_i and U. We will first fix the first $n-1$ wires with 4 different potential inputs and derive some relations amongst the matrices. Then if we run $|0\rangle_{n-1}$ into the first $n-1$ wires, then we know that the target wire would become a relative phase of the identity, i.e.

$$D_0 = U_{m+1} \cdots U_1, \text{ where } D_0 \text{ is a unitary diagonal.}$$

Now fix the first $n-1$ wires with $|k\rangle$ which is 0 on all qubits except for the k^{th} entry where it is 1. As last time, we know the target wire is still a relative phase of the identity so we get,

$$D_1 = U_{m+1} \cdots U_{i+1} X U_i \cdots U_1, \text{ where } D_1 \text{ is a unitary diagonal.}$$

Assume that the $CNOT$ corresponding to the qubit q_l is in between U_j and U_{j+1} with $j > i$ (the case for $j < i$ is nearly identical). Then fix the first $n-1$ wires with $|l\rangle$, defined similarly to $|k\rangle$, and we get by the same reasoning as above,

$$D_2 = U_{m+1} \cdots U_{j+1} X U_j \cdots U_{i+1} U_i U,$$

where D_2 is a unitary diagonal and U is just some unitary. Now fix $|j \oplus k\rangle$ and we get,

$$D_3 = U_{m+1} \cdots U_{j+1} X U_j \cdots U_{i+1} X U_i U,$$

where D_3 is a unitary diagonal and U is the same as above. Since the product of diagonals is again a diagonal and the transpose of a diagonal is again a diagonal, both $D_1^\dagger D_0$ and $D_3^\dagger D_2$ are both unitary diagonals. By explicit calculation, we have

$$D_4 = D_0 D_1^\dagger = (U_{m+1} \cdots U_1)(U_{m+1} \cdots U_{i+1} X U_i \cdots U_1)^\dagger = (U_{m+1} \dots U_{i+1}) X (U_{m+1} \dots U_{i+1})^\dagger,$$

and

$$\begin{aligned} D_5 = D_2 D_3^\dagger &= (U_{m+1} \cdots U_{j+1} X U_j \cdots U_{i+1} U_i U)(U_{m+1} \cdots U_{j+1} X U_j \cdots U_{i+1} X U_i U)^\dagger \\ &= (U_{m+1} \dots U_{j+1} X U_j \dots U_{i+1}) X (U_{m+1} \dots U_{j+1} X U_j \dots U_{i+1})^\dagger \end{aligned}$$

Letting $V_1 = U_{m+1} \dots U_{i+1}$ and $V_2 = U_{m+1} \dots U_{j+1} X U_j \dots U_{i+1}$, and observing that $X = V_1^* D_4 V_1$ and $X = V_2^\dagger D_5 V_1$, we can apply lemma 2.1 and see that $V_1 = V_2$, i.e.

$$U_{m+1} \dots U_{i+1} = U_{m+1} \dots U_{j+1} X U_j \dots U_{i+1} \implies X = I.$$

Some global phases are presumed throughout but since they are a congruence relation they are ignored. So we have derived our contradiction, and thus $(*)$ does not implement $ROTF^n$.

□

Lemma 4. *Let D_1 and D_2 be special unitary diagonals and U a unitary matrix such that $D_1 = U D_2 U^\dagger$, all 2×2. Then one of the following must be true,*

1. *U is diagonal and $D_1 = D_2$,*
2. *U is anti-diagonal and $D_1 = D_2^\dagger$,*
3. *$D_1 = D_2 = \pm I$.*

Proof. We shall first assume U is a special unitary which permits us to assign $D_1 = [\alpha, 0]$, $D_2 = [\beta, 0]$, $U = [x, y]$. Then we have the following equation,

$$\begin{aligned} [\alpha, 0] &= [x, y][\beta, 0][\overline{x}, -y] \\ &= [\beta x, y\overline{\beta}][\overline{x}, -y] \\ &= [\beta|x|^2 + \overline{\beta}|y|^2, -\beta xy + \overline{\beta} xy]. \end{aligned} \tag{6.1}$$

Hence, $0 = -\beta xy + \overline{\beta} xy \implies \beta xy = \overline{\beta} xy$. From this we know that if both x and y are nonzero then $\beta = \pm 1$, so $D_1 = D_2 = \pm I$ and we attain option 3. Suppose $y = 0$, then $\alpha = \beta|x|^2 + \overline{\beta}|y|^2 \implies \alpha = \beta|x|^2 \implies \alpha = \beta$, hence $D_1 = D_2$. Suppose $x = 0$, then $\alpha = \overline{\beta}$ and $D_1 = D_2^\dagger$.
Now suppose U is not special. Then $U = \delta V$ where δ is a complex number of unit length and V is a special unitary. Then $D_1 = U D_2 U^\dagger = (\delta V) D_2 \overline{\delta} V^\dagger = V D_2 V^\dagger$, and we may apply our previous result since V is special. □

Lemma 5. *Let D be a special unitary diagonal, U a unitary, and A a special unitary anti-diagonal. If $A = UDU^\dagger$ then $D = \pm iZ$.*

Proof. Since trace is preserved under conjugation we know that $tr(A) = tr(D)$ so $tr(D) = 0$ since A is anti-diagonal. Letting $D = [x, 0]$, we can see that $x + \overline{x} = 0$ and hence x is purely imaginary and of norm one, hence $x = \pm i$ and $D = \pm iZ$. □

6.2 Optimality of MGate

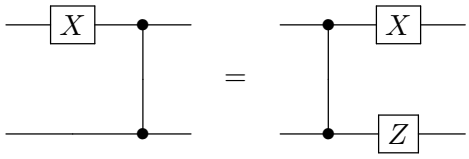
Theorem 4. *6 CNOTs are required to implement any Relative Phase Toffoli-4*

To aid us in our proof, we can make the following observations:

Observation 3. $CZ^{i;j} = CZ^{j;i}$

Hence, we do not distinguish the control and target and just write $CZ^{i,j}$ from now.

Observation 4.



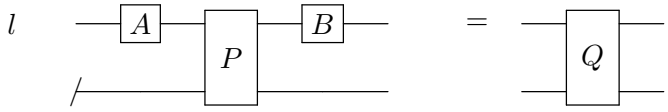
This is found in [18], this equation allows us to push a X gate through CZ with the cost of introducing another Z gate

Observation 5. *Let Q be a unitary matrix and let $|Q|_{CX}$ be the minimum CX gates required to implement Q and $|Q|_{CZ}$ be the minimum CZ gates required to implement Q, then $|Q|_{CX} = |Q|_{CZ}$.*

This observation was orginally proposed in [18], it follows from Observation 1

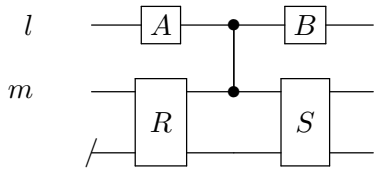
Theorem 7 (Markov& Shende, 2008)**.** *Suppose A, B are single qubit unitaries and P commutes with Z^l and the following circuit composed of A, B, P implements Q that commutes with Z^l, then at least one of the two conditions is true:*

1. *A, B are both diagonal or both anti-diagonal.*
2. *P takes the form $D \otimes P_0$ for some one qubit diagonal D*

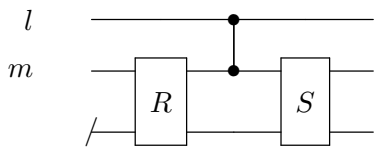


Corollary 4. *Let $\mathcal{L}$ be a circuit that implements a $RTOF^n$. If there is a control qubit l that has load factor 1, then we can assume that all the single qubit unitaries acting on l are diagonals.*

Proof. By assumption, circuit $\mathcal{L}$ takes the form



Take P as



Note that P commutes with Z^l and by our assumption $\mathcal{L}$ implements a $RTOF^n$ with l being one of the control qubit, if we denote this $RTOF^n$ by Q, then Q also commutes with Z^l. Hence, we are in the position to apply Theorem 7. All we need to show is that situation 2 can not happen in this case. Assume for contradiction that P takes the form $D \otimes P_0$ for some one qubit diagonal $\begin{pmatrix} d_0 & 0 \\ 0 & d_1 \end{pmatrix}$, if we move R, S to P's side, then the matrix $(I \otimes S^\dagger)(D \otimes P_0)(I \otimes R^\dagger) = D \otimes (S^\dagger P_0 R^\dagger)$ represents $CZ^{l,m}$, which is a contradiction. Hence, only situation 1 in Theorem 7 can happen, we can conclude that A, B are either both diagonals or anti-diagonals. Furthermore, by observation 4, one can push an X gate through the first qubit to force A, B to be diagonal matrices, so we are done. □

Theorem 8 (Markov& Shende, 2008)**.** *Suppose Q commutes with Z^l and let $\mathcal{L}$ be a CZ^l- circuit computing Q in which exactly two CZ's are incident on l, say $CZ^{l,m}$ and $CZ^{l,n}$. Then all non-diagonal one-qubit gates may be eliminated from qubit l at the cost of possibly (i) replacing $CZ(l,n)$ with $CZ(l,m)$ and (ii) adding one-qubit gates on qubits m, n*

Corollary 5. *Suppose the following circuit in Figure 6.1 implements a $RTOF^n$, then this circuit is either equal to (a) or (b) in Figure 6.2 where D is a single qubit diagonal gate.*

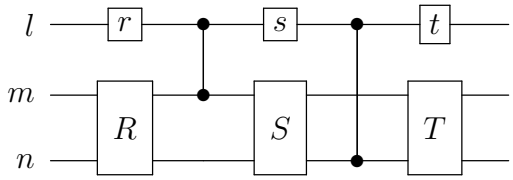
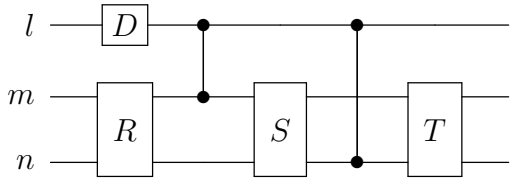
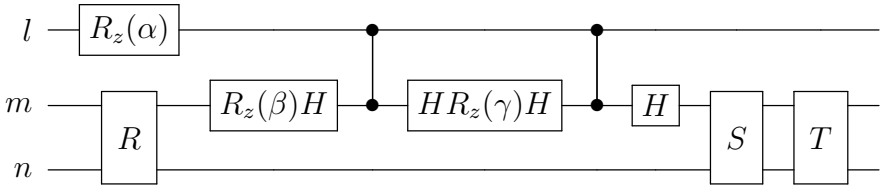


Figure 6.1: A circuit that implements a $RTOF^n$



(a)



(b)

Figure 6.2: Two possible circuit structures that Figure 6.1 could be reduced to

Proof. This is an easy corollary from Theorem 8 □

We assume that the circuits only contains single qubit unitaries and CZ gates. When we have a circuit diagram with only CZ showing up, we mean that we assume the single qubit unitaries between the CZ could be any untiary 2×2 matrices.

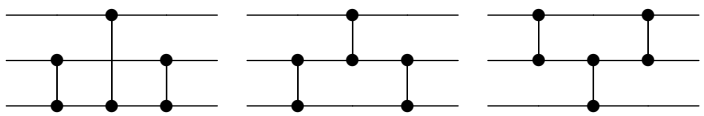
Definition 28. *Let P commutes with Z^l, then $P = |0\rangle\langle 0| \otimes P_0 + |1\rangle\langle 1| \otimes P_1$ if l is the most significant qubit. We define $E(P)$ to be the multi-set of eigenvalues of $P_1^\dagger P_0$.*

Definition 29. *Define $|P|_{CZ;l}$ to be the minimum number of CZ gates incident on l in any circuit for P in which the only entangling gates incident to P are CZ's.*

Theorem 9 (Markov &Shende, 2008)**.** *Let P commutes with Z^l, then*

- $|P|_{CZ;l} = 0$ *iff* $E(P) = \lambda\{1, 1, ...\}$ *for some* $\lambda \in \mathbb{C}$
- $|P|_{CZ;l} = 1$ *iff* $E(P) = \beta\{1, -1, 1, -1, ...\}$ *for some* $\beta \in \mathbb{C}$
- $|P|_{CZ;l} \leq 2$ *iff $E(P)$ is a set of unit norm complex numbers which come in conjugate pairs up to a global phase.*

Lemma 8. *The only circuits (up to permutation of the control qubits and taking inverses of circuits) that are not piece-wise separable with 3 CZ's in a 3-qubit network are the three circuits in Figure 6.3*



(a) (b) (c)

Figure 6.3: Possible CZ structures to implement $RTOF^3$

Proof. By applying Theorem 1 and Corollary 5, it is easy to check only the above three circuits (up to permutations and inverses) are not piece-wise separable. □

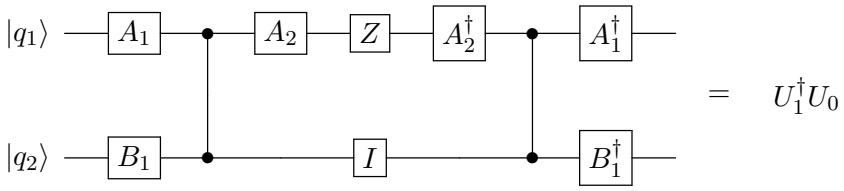
Lemma 9. *The CZ structures (b) and (c) in Figure 6.3 could not implement a $RTOF^3$.*

Proof. We will first prove that (b) in Figure 6.3 does not implement a Relative Phase Toffoli-3

Assume for contradiction that (b) could implement Relative Phase Toffoli-3. According to Corollary 4, we could assume that the single qubit unitaries acting on q_0 are diagonals. Note that if we assume the matrix that circuit (b) represents as U, U can be written as $\langle 0||0\rangle U_0 + \langle 1||1\rangle U_1$. We now denote the single qubit unitaries in the circuit by D_1, D_2 which are diagonals, and A_1, A_2, A_3, A_4, and B_1, B_2, B_3, as in figure 6.4

We consider what the circuit in Figure 6.4 reduces to if we remove $q_0 = |0\rangle$ and $q_0 = |1\rangle$. These are illustrated in Figure 6.5

Hence,



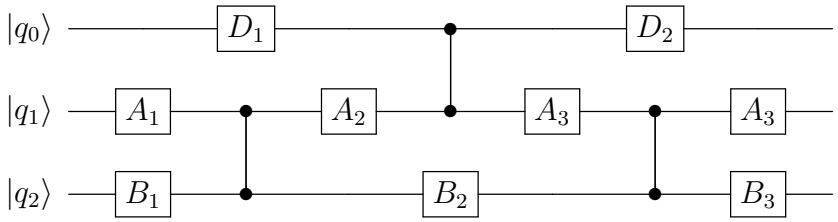
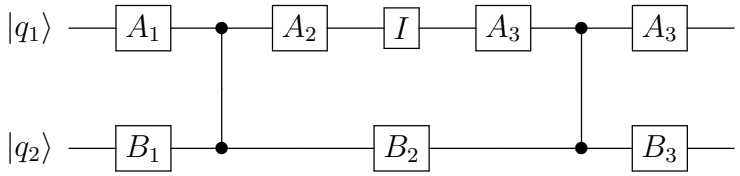
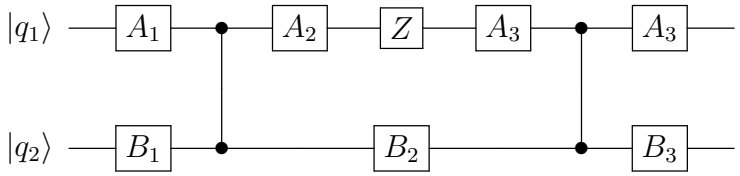


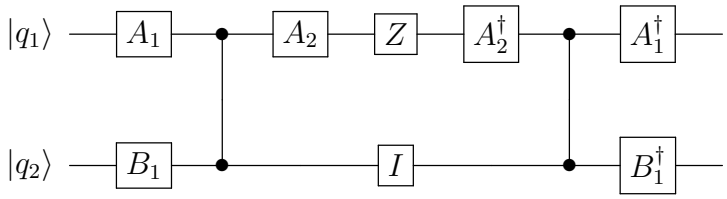
Figure 6.4: The CZ structure in Figure 6.3 with the presence of single qubit unitaries



(a) circuit $\mathcal{C}_0$ up to a global phase obtained from removing $|q_0\rangle = |0\rangle$



(b) circuit $\mathcal{C}_1$ up to a global phase obtained from removing $|q_1\rangle = |1\rangle$



(c) Circuit $\mathcal{C}_0\mathcal{C}_1^{-1}$ up to a global phase

Figure 6.5

with a global phase

By Theorem 9, the eigenvalues of $U_1^\dagger U_0$ are $\{1, -1, 1, -1\}$ up to a constant factor with norm 1

This implies that $U_1^\dagger U_0$ has the following representation

$$e^{i\theta}\begin{pmatrix} 1 & & & \\ & -1 & & \\ & & 0 & b \\ & & c & 0 \end{pmatrix}$$

where $|e^{i\theta}| = 1$ and $bc = 1$, we denote $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $X_b = \begin{pmatrix} 0 & b \\ c & 0 \end{pmatrix}$, and assume without loss of generality that $A_1,\ A_2, B_1 \in SU(2)$, so denote $A_1 = \begin{pmatrix} \alpha_1 & -\overline{\beta_1} \\ \beta_1 & \overline{\alpha_1} \end{pmatrix}$ and

$A_2 = \begin{pmatrix} \alpha_2 & -\overline{\beta_2} \\ \beta_2 & \overline{\alpha_2} \end{pmatrix}$
then one can check that we have

$$\begin{pmatrix} (|\alpha_2|^2 - |\beta_2|^2)I & (-2\overline{\alpha}\overline{\beta_2})I \\ (-2\alpha_2\beta_2)I & (|\beta_2|^2 - |\alpha_2|^2)I \end{pmatrix}$$
$$= e^{i\theta} \begin{pmatrix} B_1(|\alpha_1|^2 Z + |\beta_1|^2 X_b)B_1^\dagger & \alpha_1\overline{\beta_1}B_1(Z - X_b)B_1^\dagger Z \\ \overline{\alpha_1}\beta_1 Z B_1(Z - X_b)B_1^\dagger & Z B_1(|\beta_1|^2 Z + |\alpha_1|^2 X_b)B_1^\dagger Z \end{pmatrix}$$

so $(|\alpha_2|^2 - |\beta_2|^2)I = e^{i\theta}B_1(|\alpha_1|^2 Z + |\beta_1|^2 X_b)B_1^\dagger$, which is a contradiction.

We now go on to disprove the other structure, structure (c) in Figure 6.3. Assume for contradiction, that (c) would implement a Relative Phase Toffoli-3, then by Corollary 5, we have the circuit either equal to (a) or (b) in Figure 6.6 (the single qubit unitaries on $|q_0\rangle$ in both circuits (a) and (b) in Figure 6.6 are diagonal gates.)

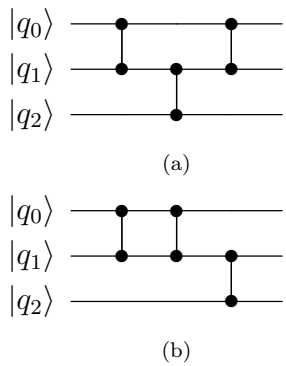


Figure 6.6: Two possible CZ structures (c) in 6.3 reduces to

Obviously, (b) is piece-wise separable, which is a contradiction, we are left with (a). Similarly, we assume that the circuit in (a) of Figure 6.6 computes U, then U can be written as $|0\rangle\langle 0|U_0 + |1\rangle\langle 1|U_1$. We now denote the single qubit unitaries in the circuit in (a) of Figure 6.6 by D(a diagonal matrix), and A_1, A_2, A_3, A_4, and B_1, B_2, B_3, as in figure XXX

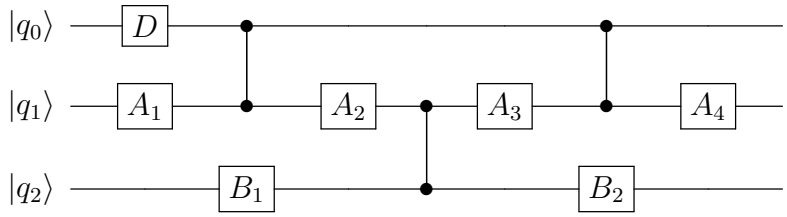


Figure 6.7: The CZ structure in (a) of Figure 6.6 with the presence of single qubit unitaries

By the result of Theorem 9, $U_1^\dagger U_0$ is of the form the eigenvalues of $U_1^\dagger U_0$ are $\{a, \overline{a}, b, \overline{b}\}$ up to a constant factor with norm 1
This implies that $U_1^\dagger U_0$ can be written as

$$e^{i\theta} \begin{pmatrix} a & & & \\ & \overline{a} & & \\ & & 0 & c \\ & & d & 0 \end{pmatrix}$$

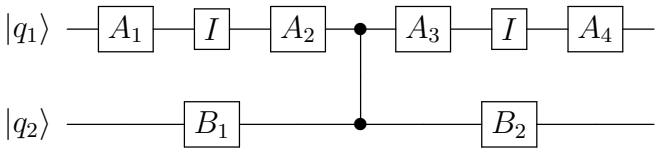
where $|e^{i\theta}| = 1$ and $cd = -1$, we denote $I_a = \begin{pmatrix} a & 0 \\ 0 & \overline{a} \end{pmatrix}$ and $X_c = \begin{pmatrix} 0 & c \\ d & 0 \end{pmatrix}$, and assume without loss of generality that $A_1,\ A_2, B_1 \in SU(2)$, so denote $A_1 = \begin{pmatrix} \alpha_1 & -\overline{\beta_1} \\ \beta_1 & \overline{\alpha_1} \end{pmatrix}$ and $A_2 = \begin{pmatrix} \alpha_2 & -\overline{\beta_2} \\ \beta_2 & \overline{\alpha_2} \end{pmatrix}$. Furthermore, note that $A_3^\dagger Z A_3 = (-i)A_3^\dagger (iZ) A_3$. Note that $A_3^\dagger (iZ) A_3 \in SU(2)$, so we denote $A_3^\dagger (iZ) A_3 = \begin{pmatrix} \alpha_3 & -\overline{\beta_3} \\ \beta_3 & \overline{\alpha_3} \end{pmatrix}$

Note that $A_2 Z A_1 = \begin{pmatrix} \alpha_1\alpha_2 + \beta_1\overline{\beta 2} & \overline{\alpha_1}\overline{\beta_2} - \alpha_2\overline{\beta_1} \\ \alpha_1\beta_2 - \overline{\alpha_2}\beta_1 & -\overline{\alpha_1\alpha_2} - \overline{\beta_1\beta_2} \end{pmatrix}$

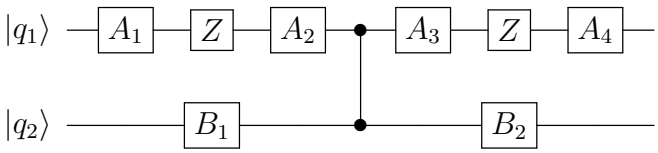
if we let $x_0 = \alpha_1\alpha_2 + \beta_1\overline{\beta 2}$ and $y_0 = \alpha_1\beta_2 - \overline{\alpha_2}\beta_1$, then we can write $A_2 Z A_1$ as $\begin{pmatrix} x_0 & \overline{y_0} \\ y_0 & -\overline{x_0} \end{pmatrix}$

we denote $A_1^\dagger A_2^\dagger$ as $\begin{pmatrix} x_1 & -\overline{y_1} \\ y_1 & \overline{x_1} \end{pmatrix}$

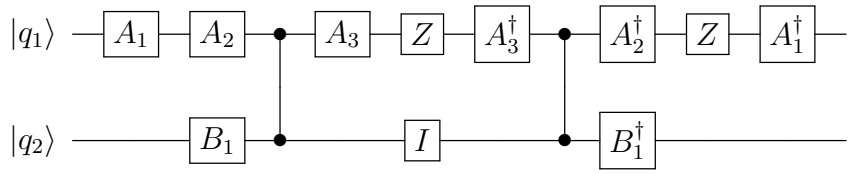
then one can see that



(a) circuit $\mathcal{C}_0$ up to a global phase obtained from removing $|q_0\rangle = |0\rangle$



(b) circuit $\mathcal{C}_1$ up to a global phase obtained from removing $|q_1\rangle = |1\rangle$



(c) Circuit $\mathcal{C}_0\mathcal{C}_1^{-1}$ up to a global phase

Figure 6.8

$$
\begin{aligned}
&(A_2ZA_1 \otimes B_1)(U_1^\dagger U_0)(A_1^\dagger A_2^\dagger \otimes B_1^\dagger) \\
&= \begin{pmatrix} x_0B_1 & \overline{y_0}B_1 \\ y_0B_1 & -\overline{x_0}B_1 \end{pmatrix} \begin{pmatrix} I_a & \\ & X_c \end{pmatrix} \begin{pmatrix} x_1B_1^\dagger & -\overline{y_1}B_1^\dagger \\ y_1B_1^\dagger & \overline{x_1}B_1^\dagger \end{pmatrix} \\
&= \begin{pmatrix} x_0B_1I_a & \overline{y_0}B_1X_c \\ y_0B_1I_a & -\overline{x_0}B_1X_c \end{pmatrix} \begin{pmatrix} x_1B_1^\dagger & -\overline{y_1}B_1^\dagger \\ y_1B_1^\dagger & \overline{x_1}B_1^\dagger \end{pmatrix} \\
&= \begin{pmatrix} B_1(x_0x_1I_a + \overline{y_0}y_1X_c)B_1^\dagger & B_1(-x_0\overline{y_1}I_a + \overline{x_1y_0}X_c)B_1^\dagger \\ B_1(y_0x_1I_a - \overline{x_0}y_1X_c)B_1^\dagger & B_1(-y_0\overline{y_1}I_a - \overline{x_0x_1}X_c)B_1^\dagger \end{pmatrix}
\end{aligned}
$$

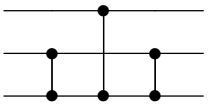
Hence, according to circuit diagram (c) in 6.8, we have that

$$
\begin{pmatrix} \alpha_3 I & -\overline{\beta_3}I \\ \beta_3 I & \overline{\alpha_3}I \end{pmatrix} = \begin{pmatrix} B_1(x_0x_1I_a + \overline{y_0}y_1X_c)B_1^\dagger & ZB_1(-x_0\overline{y_1}I_a + \overline{x_1y_0}X_c)B_1^\dagger \\ B_1(y_0x_1I_a - \overline{x_0}y_1X_c)B_1^\dagger Z & ZB_1(-y_0\overline{y_1}I_a - \overline{x_0x_1}X_c)B_1^\dagger Z \end{pmatrix}
$$

up to a global phase.

we can see that if $\alpha_3 \neq 0$, then both x_0x_1 and $\overline{y_0}y_1$ have to be 0, which is a contradiction. Hence, $\alpha_3 = 0$ and $x_0x_1 = \overline{y_0}y_1 = 0$, this leads to a contradiction in these two equations $-\overline{\beta_3}I = B_1(y_0x_1I_a - \overline{x_0}y_1X_c)B_1^\dagger Z$ and $\beta_3 I = B_1(y_0x_1I_a - \overline{x_0}y_1X_c)B_1^\dagger Z$ □

Theorem 10. *The following circuit is the only valid implementation of a Relative Phase Toffoli-3 with* 3 *CZ's.*



Proof. This follows from Lemma 8 and Lemma 9. □

Observation 6. *Suppose $\mathcal{L}$ is a CZ^l-circuit computing a Relative Phase Toffoli-n gate in which exactly two CZ's are incident on l, say $CZ^{l,m}$ and $CZ^{l,n}$. Assume the unitaries acting on the remaining circuits are R, S, T. Then either RZ^mSZ^nT or RZ^mSZ^nT implements a Relative Phase Toffoli-$n-1$*

Proof. By Corollary5, there are two circuits that $\mathcal{L}$ is equal to. In both cases, we are able to fix $l = |1\rangle$ and look at the remaining circuit as in both cases, there is only a diagonal gate operating on l. Hence, it is easy to check that either RZ^mSZ^nT or RZ^mSZ^mT implements a Relative Phase Toffoli $-n-1$. □

Observation 7. *When there are $5CZ$'s in a $4-$ qubit circuit, there has to be a qubit of load factor 2.*

so to prove Theorem 4, we split into two cases: the case when there is a control qubit of load factor 2, and the case when the only qubit of load factor 2 is the target qubit.

Lemma 10. *When there are less than $5CZ$'s in a 4 qubit circuit and there is a control qubit of load factor 2, the circuit could not implement a Relative Phase Toffoli-4*

Proof. Consider an arbitrary 4-qubit circuit with exactly 5 CZ gates and at least one control qubit of load factor 2 (Without loss of generality, call it q_0). Assume for contradiction that it implements a Relative Phase Toffoli-4. By Corollary 5, we may assume that there are two

equivalent circuits such that the single qubit unitaries on q_0 is a diagonal matrix. Then one could either argue that the circuit is piece-wise separable or remove q_0 to derive an invalid implementation of Relative Phase Toffoli-3, which are contradictions.
Here is one example of how this argument works. Consider the following circuit, we assume for contradiction that it could implement a Relative Phase Toffoli-4. Notice that q_0 has load factor 2.

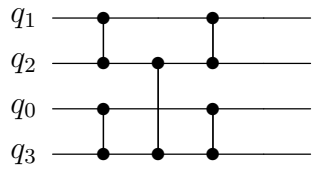
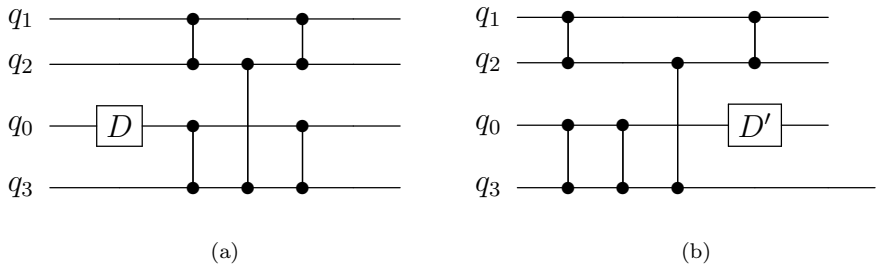


Figure 6.9: One example of 5 CZ's in a 4-qubit network with a control qubit of load factor 2

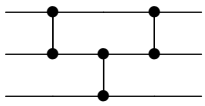
By Corollary 5, the circuit is either equal to either (a) in or [b]



(a) (b)

Figure 6.10: Two CZ structures that the circuit in Figure 6.9 is equivalent to (D and D' are diagonal matrices and they are the only single qubit unitaries acting on q_0 in (a) and (b) respectively)

Hence, we can remove $q_0 = |1\rangle$ to derive an implementation of a Relative Phase Toffoli-3, but in both cases, the CZ structure after removing q_0 is



which is not a valid CZ structure for Relative Phase Toffoli-3. □

Theorem 11. *The circuit in Figure 6.11 can not implement U where $U = e^{i\theta}\begin{pmatrix} Z & & & \\ & Z & & \\ & & Z & \\ & & & X_b \end{pmatrix}$ ($Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $X_b = \begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix}$ with $ab = 1$, $e^{i\theta}$ is an arbitrary complex number of magnitude 1)*

Proof. By Corollary 5, we may assume the circuit in Figure 6.11 can be reduced to either (a) or (b) in Figure 6.12

Note that (a) in Figure 6.12 can be reduced to only one diagonal gate D operating on the first qubit, which obviously can not implement a Relative Phase Toffoli-3. Hence, we may assume that the circuit can be reduced to (b) in Figure 6.12.

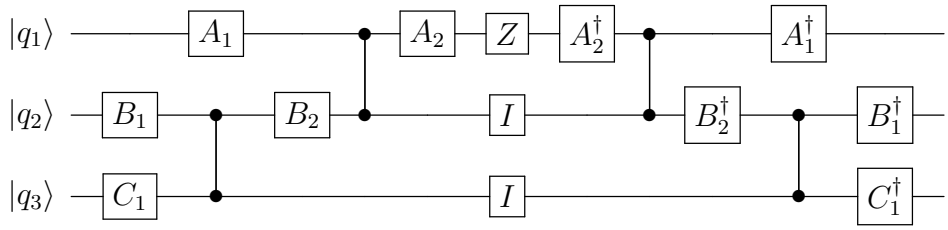
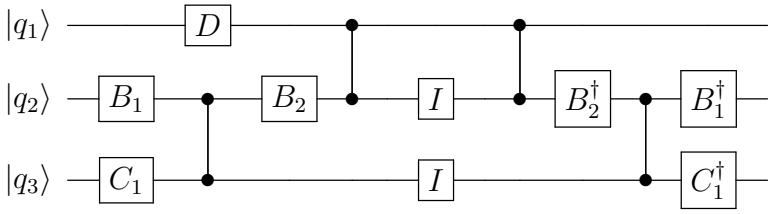
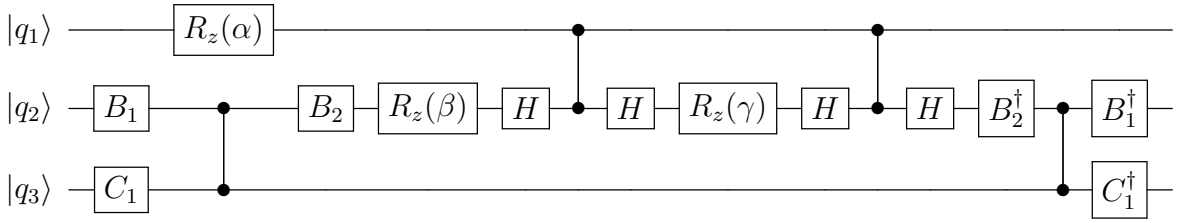


Figure 6.11: Circuit that can not implement a special type $RTOF^3$ (A_1, A_2, B_1, B_2, C_1 are arbitrary single qubit unitaries)



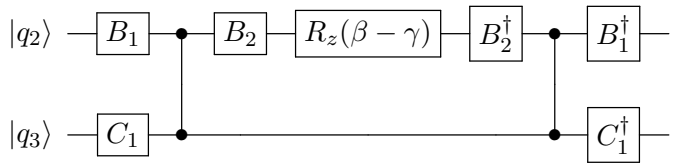
(a) (D is a diagonal gate)



(b)

Figure 6.12: The circuits in Figure 6.11 are reduced to

In the circuit in (b) of Figure 6.12, we remove $q_1 = |0\rangle$ to obtain circuit $\mathcal{C}_0$ that computes U_0, which is equal to $e^{i\theta} I \otimes Z$



$= e^{i(\theta-\alpha/2)} I \otimes Z$

Hence, $(B_2^\dagger R_z(\beta-\gamma) B_2) \otimes I = e^{i(\theta-\alpha/2)} \begin{pmatrix} C_1 Z C_1^\dagger & 0 \\ 0 & Z C_1 Z C_1^\dagger Z \end{pmatrix}$
which is a contradiction since this implies $I = Z$ up to a global phase.

□

Lemma 11. *When there are less than* $5CZ$*'s in a* 4 *qubit circuit and the only qubit of load factor* 2 *is the target qubit, the circuit could not implement a Relative Phase Toffoli-4*

Proof. Note that when the only qubit of load factor 2 is the target qubit, then there must be at least one qubit of load factor 1. A quick inspection using piece-wise separabiltiy of circuits show that a circuit with two control qubits of load factor 1 could not implement

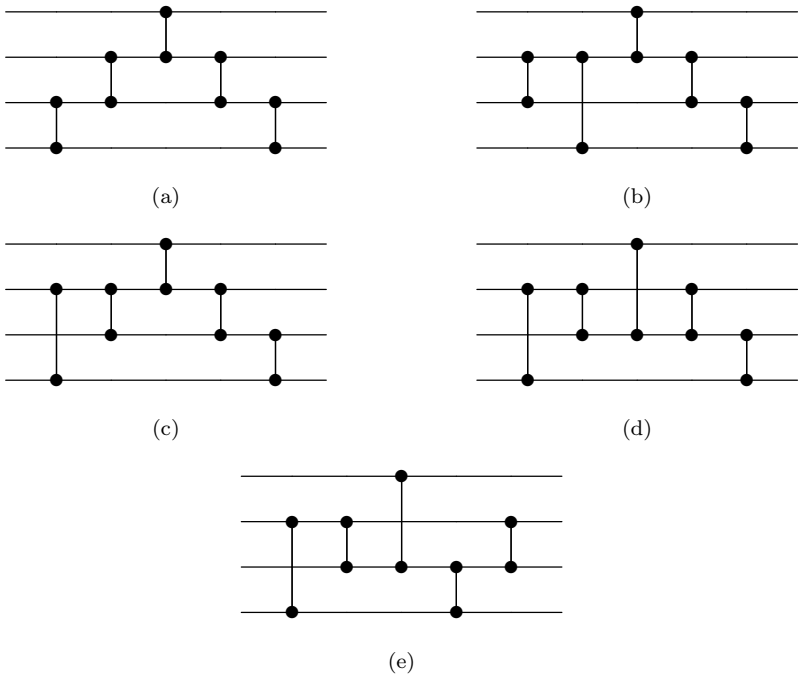


Figure 6.13: Possible CZ structures to implement $RTOF^4$ with 5 CZ's and only one qubit of load factor 2, which is the target qubit

Relative Phase Toffoli gates. Hence, we have exactly one qubit of load factor 1 in the circuit. This leaves us with the CZ structures as represented in Figure **??**

For each of the above structures, assume for contradictions that it could implement a Relative Phase Toffoli-4. Remove $q_0 = |0\rangle$, call the resulting circuit C_0 and the unitary it implements U_0. Then remove $q_0 = |1\rangle$, call the resulting circuit C_1 and the unitary it implements U_1. Consider circuit $C_0C_1^{-1}$ and circuit $C_1^{-1}C_0$. Note that the unitary $C_0C_1^{-1}$ implements is $U_1^\dagger U_0$. By Theorem9, since q_0 has load factor 1, we may assume that $U_1^\dagger U_0 = e^{i\theta}\begin{pmatrix} Z & & & \\ & Z & & \\ & & Z & \\ & & & X_b \end{pmatrix}$. Similarly, since $C_1^{-1}C_0$ implements $U_0U_1^\dagger$ and q_0 has load factor 1, we may also assume that $U_0U_1^\dagger = e^{i\theta'}\begin{pmatrix} Z & & & \\ & Z & & \\ & & Z & \\ & & & X_{b'} \end{pmatrix}$. Notice that if we exchange the first two most significant qubit in the matrix $e^{i\theta}\begin{pmatrix} Z & & & \\ & Z & & \\ & & Z & \\ & & & X_b \end{pmatrix}$, we get the matrix $e^{i\theta}\begin{pmatrix} Z & & & \\ & Z & & \\ & & Z & \\ & & & X_{1/b} \end{pmatrix}$

Hence, we may conclude that both $C_0C_1^{-1}$ and $C_1^{-1}C_0$ and the transposition of the first two

control qubits implement a unitary $e^{it}\begin{pmatrix} Z & & & \\ & Z & & \\ & & Z & \\ & & & X_a \end{pmatrix}$. However, notice that at least one of $C_0C_1^{-1}$, $C_1^{-1}C_0$ and the transposition of the first two control qubits has the same structure as in Figure 6.11, which is a contradiction by Theorem 11.

□

References

[1] F. Abalyev, A. Gainutdinova, M. Karpinski, C. Moore, and C. Pollett, *On the computational power of probabilistic and quantum branching program*, Information and Computation, 203 (2005), pp. 145–162.

[2] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, *Elementary gates for quantum computation*, Physical Review A, AC5710 (1995).

[3] D. Barrington, *Bounded-with polynomial-size branching programs recognize exactly those languages in* $nc^1, Journal of Computer and System Sciences, 38(1989), pp.$ $150 - -164$.

[4] P. Benioff, *The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines*, Journal of Statistical Physics, 22 (1980), p. 563–591.

[5] E. Bernstein and U. Vazirani, *Quantum complexity theory*, SIAM Journal on computing, 26 (1997), pp. 1411–1473.

[6] D. Deutsch, *Quantum theory, the church–turing principle and the universal quantum computer*, Proceedings of the Royal Society A, 400 (1985).

[7] D. Deutsch and R. Jozsa, *Rapid solution of problems by quantum computation*, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, 439 (1992), pp. 553–558.

[8] R. Feynman, *Simulating physics with computers*, International Journal of Theoretical Physics, 21 (1982), p. 467–488.

[9] C. Gidney, *Using quantum gates instead of ancilla bits*, Jun 2015.

[10] L. K. Grover, *A fast quantum mechanical algorithm for database search*, in Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 1996, pp. 212–219.

[11] J. Haah, *Product decomposition of periodic functions in quantum signal processing*, Quantum, 3 (2019), p. 190.

[12] Y. Hamoudi and F. Magniez, *Quantum time-space tradeoff for finding multiple collision pairs*, TQC, (2020).

[13] A. S. Holevo, *Bounds for the quantity of information transmitted by a quantum communication channel*, Probl. Peredachi Inf., 9 (1973), pp. 3–11.

[14] H. Klauck, R. Spalek, and R. Wolf, *Quantum and classical strong direct product theorems and optimal time-space tradeoffs*, SIAM Journal on Computing, 36 (2007).

[15] D. Maslov, *On the advantages of using relative phase toffolis with an application to multiple control toffoli optimization*, Physical Review A 93, 77 (2016), pp. 1130–1131.

[16] D. Maslov, J.-S. Kim, S. Bravyi, T. J. Yoder, and S. Sheldon, *Quantum advantage for computations with limited space*, Nature Physics, (2021).

[17] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, (2000).

[18] V. Shende and I. Markov, *On the cnot-cost of toffoli gates*, Quantum Inf. Comput., 9 (2009), pp. 461–486.

[19] V. V. Shende, S. S. Bullock, and I. L. Markov, *Synthesis of quantum-logic circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25 (2006), pp. 1000–1010.

[20] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, in Proceedings 35th annual symposium on foundations of computer science, Ieee, 1994, pp. 124–134.

[21] G. Song and A. Klappenecker, *The simplified toffoli gate implementation by margolus is optimal*, arXiv: Quantum Physics, (2003).

[22] J. von Neumann, *Mathematical Foundations of Quantum Mechanics*, 1932.

[23] I. Wegener, *The complexity of Boolean functions*, Wiley-Teubner series in computer science, 1987.

[24] W. K. Wootters and W. H. Zurek, *Bounds for the quantity of information transmitted by a quantum communication channel*, Nature, 299 (1982), pp. 802–803.