Quantum Circuit Synthesis using Group Decomposition and Hilbert Spaces

by

Michael S. Saraivanov

A thesis submitted in partial fulfillment of the
requirements for the degree of

Masters of Science
in
Electrical and Computer Engineering

Thesis Committee:
Marek Perkowski, Chair
Doug Hall
Xiaoyu Song

Portland State University
2013

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

The exponential nature of Moore's law has inadvertently created huge data storage complexes that are scattered around the world. Data elements are continuously being searched, processed, erased, combined and transferred to other storage units without much regard to power consumption. The need for faster searches and power efficient data processing is becoming a fundamental requirement. Quantum computing may offer an elegant solution to speed and power through the utilization of the natural laws of quantum mechanics. Therefore, minimal cost quantum circuit implementation methodologies are greatly desired.

This thesis explores the decomposition of group functions and the Walsh spectrum for implementing quantum canonical cascades with minimal cost. Three different methodologies, using group decomposition, are presented and generalized to take advantage of different quantum computing hardware, such as ion traps and quantum dots. Quantum square root of swap gates and fixed angle rotation gates comprise the first two methodologies. The third and final methodology provides further quantum cost reduction by more efficiently utilizing Hilbert spaces through variable angle rotation gates. The thesis then extends the methodology to realize a robust quantum circuit synthesis tool for single and multi-output quantum logic functions.

DEDICATION

To my family Carrie and Evan Saraivanov

ACKNOWLEDGMENTS

Professors:

Marek Perkowski, Doug Hall, Xiaoyu Song

Long time mentor, Zhenxin Jiang

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION TO QUANTUM TECHNOLOGY

In this chapter, based on literature, we will illustrate some of the powers of quantum computers in regards to speed, power savings, and computing parallelism. We will then introduce two quantum technologies. The first is the ion trap and the second is the quantum dot. Currently only the ion trap has been implemented as a quantum computer, however, quantum dots have advantages that may also make them a leading contender. DiVincenzo established a set of requirements that can be used to evaluate quantum computer technologies and determine their viability. The criteria are as follows:

- identification of well-defined qubits
- reliable state preparation
- long decoherence times
- accurate quantum gate operations and
- strong quantum measurements

Technologies such as ion traps and quantum dots are currently strong contenders, with ion traps showing the most promise. Other technologies,

such as super conductors, optical, and electrodynamics may surpass even the ion trap as rigorous research continues.

Integration of quantum computers into existing microprocessors may be an intermediate, and even a mandatory, scheme for utilizing quantum computer technologies. This type of hybrid integration can be thought of as having existing microprocessors interfacing with and utilizing quantum co-processors for performing powerful recursive computations and large search algorithms, in polynomial time (opposed to exponential time), with relatively low computational power. Speed and power savings may very well be the fundamental need for the utilization of quantum computers in the first place. Speed can be gained in two different methods in today's classical computers, either by increasing the switching speed, or currently and more feasibly, increasing the parallel processing paths. Basically, the computational power of a processor operating an optimized frequency can be doubled by simply instantiating a second processer path, operating at the same optimized frequency. A similar type of parallelism can be achieved by taking advantage of two quantum mechanical phenomena, known as quantum superposition and quantum entanglement.

1.1 Quantum Information Processing

Quantum superposition is a natural property of quantum particles, which implies that such a particle can exist partly in all of its possible states simultaneously, and any one state can be represented probabilistically. In other words, a single quantum particle has a non-zero probability of occupying any point in space. So in classical computers, the number of states is $N = 2^n$, where n is the number of bits, however the states are totally discrete. Whereas in quantum computers the number of states is the number of possible superpositions of 2^n states and n in this case is the number of quantum bits or qubits. Proving the existence of superposition and entanglement is not addressed in this thesis and goes beyond the scope of the thesis. However, it must be mentioned that in order to observe the position or state of a quantum particle, an exchange of energy must occur between the observational device and the quantum particle, which means that the state of the particle will be altered and will collapse to one of its pure states. For electrons, which are fermions, the pure states are spin up or spin down and can be thought of as a classical binary bit. So quantum operations should be performed between measurements in order to take advantage of superposition.

Quantum entanglement is another natural property of quantum particles, where the state of one particle affects the state of all other entangled particles, at an arbitrary distance. This means that individual qubit

operations can be performed and selectively alter the states of other entangled qubits, and form more complicated controlled operations or quantum gates. It should be pointed out that an entangled system of qubits is continuously subject to environmental background radiation, including noise from the equipment used to control the qubits, and therefore the entanglement period is relatively short (in some applications the decoherence time can be as short as micro seconds and up to several minutes). This means that the desirable qubit system is one in which the qubit operation times are orders of magnitude shorter than the decoherence times. Figure 1-1 shows the decoherence ratios of three different quantum qubit technology candidates.

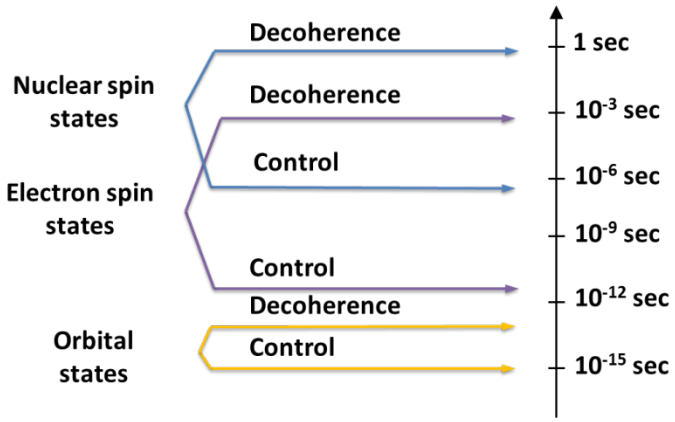


Figure 1-1. Decoherence times and the qubit operation control time for three different quantum computing technologies

Note that the longest decoherence times, with respect to the qubit operation, or spin control times, is for nuclear and electron quantum spin systems. The

ion trap is a technology that exploits nuclear spin states, and quantum dots exploit electron spin states. Both quantum systems will be discussed in greater detail in Sections 1.4 and 1.5, respectively, in this chapter.

Current classical computers utilize primitive logic gates, which output binary values based on the evaluation of a set of binary input values. Most commonly the gates have more input values than output values and form a surjective type of input-to-output mapping. This mapping is not reversible, meaning that the output cannot be resolved or traced to its discrete input values. This means that during the evaluation process in such primitive gates, energy is dissipated and discarded. In single electron transistors, the amount of energy needed to detect the presence of one electron has been quoted as ranging from several hundred meV to as few as 50meV. Rolf Landauer of IBM has established that the smallest amount of energy needed to erase 1 bit of information (or the merging of two bits to form one bit) corresponds to an increase in entropy by a minimum of $E = kTln2$, where k is Boltzmann's constant and T is the operating temperature in Kelvins. For a device operating at room temperature, that would equate to approximately 18meV. When classical CMOS devices process information at rates of millions per second and have done so continuously for many decades and simultaneously in hundreds of millions of devices, we can easily see that a substantial amount, multiple Hoover Dam sized power plants worth of power,

has been dissipated. Quantum computers on the other hand can be constructed entirely of quantum gates and theoretically have zero power dissipation as a result of information loss. Power is however, consumed as individual qubits are manipulated by electromagnetic pulses. It should be pointed out that reversible circuits can also be implemented in CMOS technologies but at the expense of increasing the device count by at least a factor 2x.

1.2 Structure of the Thesis

The contents of the rest of this thesis are organized as follows: In the remaining sections of this chapter the introduction to quantum gates as unitary matrices is outlined. In addition, a general overview of two popular quantum technologies, the ion trap and quantum dots, are introduced. For each technology, the basic process of generating the most primitive controlled universal quantum gates is surveyed. In Chapter 2, Tsutomu Sasao's group decomposition, using the Walsh spectrum, method for synthesizing classical reversible logic canonical cascades, is reviewed. In Chapter 3, the group decomposition method is modified so that it can be used for synthesizing quantum circuits. Three methodologies are derived and their utilization of the quantum computing hardware of Chapter 1 is evaluated. A leading contender, termed as EQB, quickly emerges. In Chapter 4, the EQB methodology is further extended to quantum circuit synthesis and lays down

the ground work for creating a software synthesis tool. The chapter continues by evaluating benchmark quantum circuits, and their associated Maslov cost against EQB cost, and provides a brief summary of the results. Finally, Chapter 5 concludes this thesis.

1.3 Quantum Gates

Quantum gates differ from classical logic gates, as quantum gates are realized by performing physical rotations on individual quantum states sequentially in time. Classical logic gates produce output values based on voltages on input wires and can perform such an evaluation simultaneously on many wires. This means that quantum computer operations are purely sequential and slower than that of classical operations. The speed of a quantum computer comes from the utilization of superposition as mentioned in Section 1.1, of this chapter. Individual qubit rotations can be sequentially performed to form two qubit conditional operations. One of the most basic universal gate operations in quantum computers is the Exclusive-Or operation. Universal gate refers to a gate that can be used to implement any type of function. This is in contrast to classical computers, with the exception of the NOT gate which is basic in both technologies, the universal XOR gate is one of the more complex primitive gates to realize. Individual rotations in quantum gates are analogous to individual CMOS devices in classical logic gates. The implementation of an XOR gate requires 5 rotations in a quantum

implementation, and between 10 and 12 CMOS devices in a classical reversible implementation.

1.3.1 Single Qubit Quantum Gates

Quantum computers map particle quantum states, such as electron orbitals and/or nuclear spins, in Hilbert spaces to other states according to the Schrödinger wave equation. Hilbert spaces extend lower dimensional space to a higher number of dimensions. This is needed since quantum particle states can be rotated in 3D complex space and exists in a linear superposition of all of its states. Since it is a linear superposition of multiple states, we can utilize linear unitary matrixes to perform qubit rotations. The 3D Bloch sphere is commonly used to visualize qubit rotations. Please note that the superposition of a quantum particle cannot be measured. Such a measurement will force a quantum particle to assume one of its basis states (for fermions that is spin up or spin down as stated previously). The Bloch sphere in Figure 1-2 shows the vectors that can be achieved by rotation in the X, Y, or Z axes:

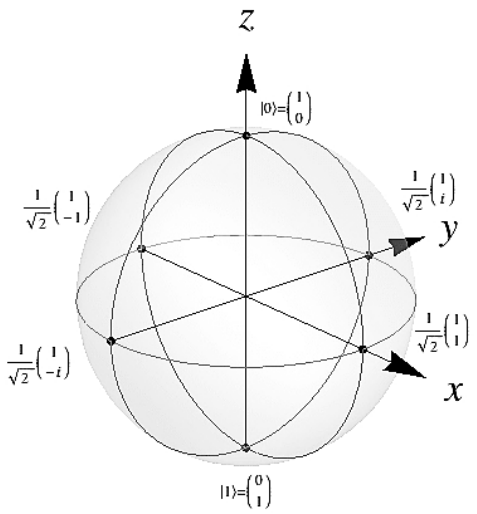


Figure 1-2. Bloch sphere with vector locations and values [1]

Superposition is represented by a point on the sphere that cannot be decomposed into two vectors and is not one of the $|0\rangle$, or $|1\rangle$ states.

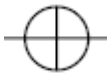
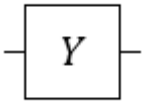
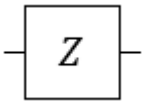
Assuming individual qubit rotations have been successfully realized with a high degree of fidelity or accuracy, the following are some of the most common rotation gates (note that the application a phase correction is required in order to account for imaginary vector space. This will be explained more in depth as its importance progresses in this thesis):

- The X gate is a π rotation about X axis, followed by a global phase correction of $\pi/2$, and is the quantum equivalent of a classical NOT gate since it can switch between the $|0\rangle$ and $|1\rangle$ states: $X_\pi = R_x(\pi) * R_{ph}(\pi/2)$

- The Y gate is a π rotation about Y axis. This gate also inverts, but does so in the imaginary plane. This gate also changes the probability of occurrence of the $|0\rangle$ and $|1\rangle$ states and has a global phase correction of $\pi/2$: $Y_\pi = R_y(\pi) * R_{ph}(\pi/2)$
- The Z gate is a π rotation about Z axis, along with a global pahse correction of $\pi/2$. This gate does no inversion and therefore does not affect the probabilities of the $|0\rangle$ and $|1\rangle$ states: $Z_\pi = R_z(\pi) * R_{ph}(\pi/2)$
- The Hadamard gate effectively describes superposition, or a rotation of π about the X and a $\pi/2$ rotation about the Y axis, and a $\pi/2$ global phase correction. The Hadamard gate rotates the qubit into superposition and completely randomizes the state (the probabilities of measuring the $|0\rangle$ and $|1\rangle$ states are equal at .5): $H = R_x(\pi) * R_y(\pi/2) * R_{ph}(\pi/2)$
- The second is the V gate, which is an X rotation by $\pi/2$ radians and a global phase correction of $\pi/4$. The V gate is also called the half of a NOT or Square-root-of-NOT, since it performs a partial inversion, and has its own Hermitian: $V = R_x(\pi/2) * R_{ph}(\pi/4)$
- The global phase gate rotates the reference of a qubit about the Z axis by ϕ, where $\phi =$ any fraction of π. It is very similar to a Z gate but rotates the reference frame of a qubit and the qubit itself. To avoid confusion, its angle is denoted by ϕ (note that the probabilities of measuring the $|0\rangle$

and $|1\rangle$ states are unchanged and is applied to most gates to account for the fact that a full rotation with imaginary space is 4π).

Table 1-1 summarizes the gate matrixes of some of the most common single qubit rotation gates, and their respective quantum gate symbols [2].

Gate Name	**Matrix**	**Symbol**
X Gate (global phase correction of $\pi/2$ appended)	$X_\pi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	
X Gate with arbitrary θ rotation (global phase correction omitted)	$R_x\theta = \begin{bmatrix} cos(\theta/2) & -isin(\theta/2) \\ -isin(\theta/2) & cos(\theta/2) \end{bmatrix}$	
Y Gate (global phase correction of $\pi/2$ appended)	$Y_\pi = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	
Y Gate with arbitrary θ rotation (global phase correction omitted)	$R_y\theta = \begin{bmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{bmatrix}$	
Z Gate ($\theta = \pi$) (global phase correction of $\pi/2$ appended)	$Z_\pi = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	

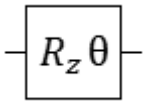
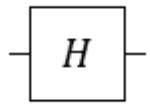
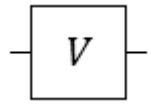
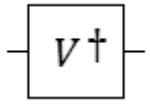
Z Gate with arbitrary θ rotation (global phase correction omitted)	$R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$	
Hadamard Gate (global phase correction of π/2 appended)	$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	
V Gate (global phase correction of π/4 appended)	$V = \frac{1}{2}\begin{bmatrix} (1+i) & (1-i) \\ (1-i) & (1+i) \end{bmatrix}$	
V† Gate (Hermition of V gate, phase correction of π/4 appended)	$V\,\dagger = \frac{1}{2}\begin{bmatrix} (1-i) & (1+i) \\ (1+i) & (1-i) \end{bmatrix}$	
Global Phase Gate (for any ϕ, this is always applied to Pauli gates for correcting phase)	$R_{ph}(\phi) = \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{i\phi} \end{bmatrix}$	

Table 1-1. Single Qubit Quantum Gates

The Bloch sphere in Figure 1-3 shows how the vector $\alpha|0\rangle + \beta|1\rangle = |\Psi\rangle$ is mapped.

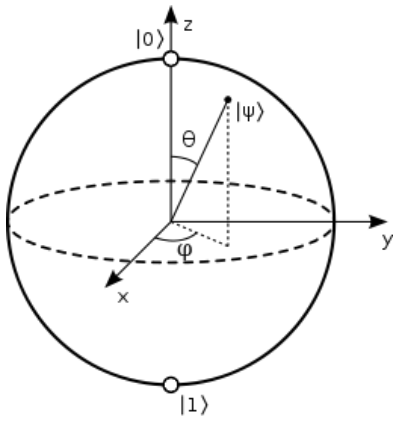


Figure 1-3. Bloch sphere showing vector components [3]

Note that there are multiple ways of generating equivalent gates and only well-known methods have been presented so far. Armed with the aforementioned permutative quantum gates or qubit rotations, we can perform combinational logic operations, which are represented by the multiplication of their respective matrixes. If a qubit is initialized to the $|1\rangle$ state and is then measured, the output value will also be $|1\rangle$ and vise-versa, if it had been initialized to the $|0\rangle$ state. The $|1\rangle$ qubit can be represented by the column vector of:

$$\alpha|0\rangle + \beta|1\rangle = |\Psi\rangle = |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

And if we wish to apply the X gate, the resultant vector is:

$$|\Psi_{result}\rangle = \begin{bmatrix}0\\1\end{bmatrix} * \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} = |0\rangle$$

The truth table for the X rotation is the following:

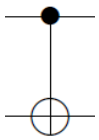
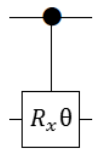
Input Qubit Sate	Output Qubit State
0	1
1	0

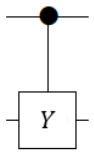
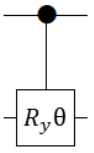
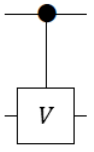
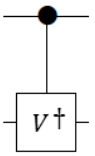
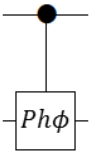
Table 1-2. X Gate Truth Table

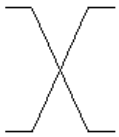
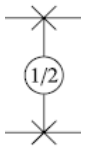
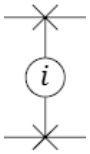
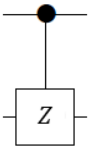
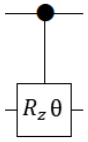
This means that a rotation of π radians about the X axis performs a quantum NOT operation or gate. Such gates can be sequentially applied (or multiplied) to move the qubit to any arbitrary position in space.

1.3.2 Multi Qubit Quantum Gates

Remember that these primitive single qubit gates cannot realize universal gates, gates with which any function can be implemented. Therefore we must utilize some kind of a two qubit interaction or exchange. Table 1-3 shows some of the more commonly used universal two qubit gates:

Controlled Gate Name	**Matrix**	**Symbol**
Controlled NOT Gate or Feynman gate (rotation about the X axis by π)	$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	
Controlled X gate (arbitrary rotation about the X axis)	$CR_x\theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & cos(\theta/2) & -isin(\theta/2) \\ 0 & 0 & -isin(\theta/2) & cos(\theta/2) \end{bmatrix}$	

Controlled NOT gate (rotation about the Y axis by π)	$CYNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix}$	
Controlled Y gate (arbitrary rotation about the Y axis)	$CR_y\theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & cos(\theta/2) & -sin(\theta/2) \\ 0 & 0 & sin(\theta/2) & cos(\theta/2) \end{bmatrix}$	
Controlled V Gate (controlled $NOT^{1/2}$ gate)	$CV = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (1+i)/2 & (1-i)/2 \\ 0 & 0 & (1-i)/2 & (1+i)/2 \end{bmatrix}$	
Controlled V† Gate (Hermition of V gate)	$CV\dagger = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (1-i)/2 & (1+i)/2 \\ 0 & 0 & (1+i)/2 & (1-i)/2 \end{bmatrix}$	
Interaction Gate (changes the phase of both qubits in equal directions, one qubit is chosen as a reference)	$Int = \begin{bmatrix} e^{-i\phi} & 0 & 0 & 0 \\ 0 & e^{-i\phi} & 0 & 0 \\ 0 & 0 & e^{i\phi} & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{bmatrix}$	
Controlled Phase Gate (same an interaction gate, but the phase of the first qubit is chosen as a reference)	$CPh_\phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{bmatrix}$	

$SWAP$ Gate (similar to a CNOT gate but for two qubits)	$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
$SWAP^{1/2}$ Gate (same as a $SWAP$ gate, but for half the rotation, or a V gate for two qubits. Qubit interaction changes spins in opposite directions)	$SWAP^{1/2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (1+i)/2 & (1-i)/2 & 0 \\ 0 & (1-i)/2 & (1+i)/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
$iSWAP$ Gate (same as a $SWAP$ gate, but for a full rotation, qubit interaction changes spins in opposite directions and a global phase of $\phi = \pi/2$)	$iSWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
Controlled Z Gate (a rotation of $\theta = \pi$ about the Z axis, with a global phase of $\phi = \pi/2$)	$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	
Controlled Zθ Gate (same as Z gate, but for any θ, and global phase omitted)	$CR_z\theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-i\theta/2} & 0 \\ 0 & 0 & 0 & e^{i\theta/2} \end{bmatrix}$	

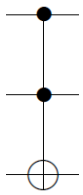
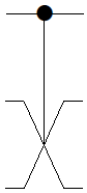
Toffoli Gate (controlled controlled NOT)	$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	
Fredkin Gate (controlled swap)	$CSWAP = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	

Table 1-3. Table of common Controlled Gates

There are many more gates to create and just as many more that need to be invented. In quantum circuits, qubit rotations are analogous to CMOS gates, and multiple qubits are analogous to parallel data paths in classical computers. This means that large quantum processors can be created from just a few qubits. Theoretically only three qubits are needed to realize an arbitrary reversible binary function, but will require more gates. In addition, since the quantum world is probabilistic, multiple state combinations can be evaluated simultaneously and expressed in terms of probability of occurrences.

1.4 Ion Trap Quantum Computer

Currently the most well understood quantum computer system exists in liquid state nuclear magnetic resonance devices. These type of systems use the bulk spin states of entire molecules as qubits. However early on in the development it was concluded that liquid quantum computers cannot be scaled and only primitive single and double qubit experiments were mostly conducted. In 2001, the largest NMR quantum computer implemented was seven qubits [4]. Though NMR shows a limited future in quantum computers, it has helped tremendously in understanding quantum behavior and quantum computing in general. NMR technology has helped set some standards of comparisons for other quantum computers. The ion trap quantum computer has been realized and has been generally recognized as a potentially scalable solution. In 2011, the biggest ion trap was created and contains fourteen entangled qubits at the University of Innsbruck (Austria) [5]. Such a quantum computer easily has enough computing power to perform complex factoring and search algorithms. The decoherence times of the qubits are in the seconds time frames, while individual rotations can be performed in less than a millisecond. This means that thousands of gate operations can be performed with high fidelity before the system loses coherence. The most common type of trap used to trap ion is the Paul trap. The Paul trap consists of four electromagnetic fields that trap ions in two spatial directions. Two

additional electrodes at each end provide lateral confinement and control. When the trap is placed in a vacuum and cooled, the ion string behaves as a single crystal. Laser pulses can then perform individual and multi ion rotations while a measurement can be made using a CCD camera. Figure 1-4 shows one proposal for implementation of an ion Paul trap. The ion trap is enclosed in a vacuum chamber in a controlled temperature. Optical windows are used to expose the ions to the CCD camera and the lasers.

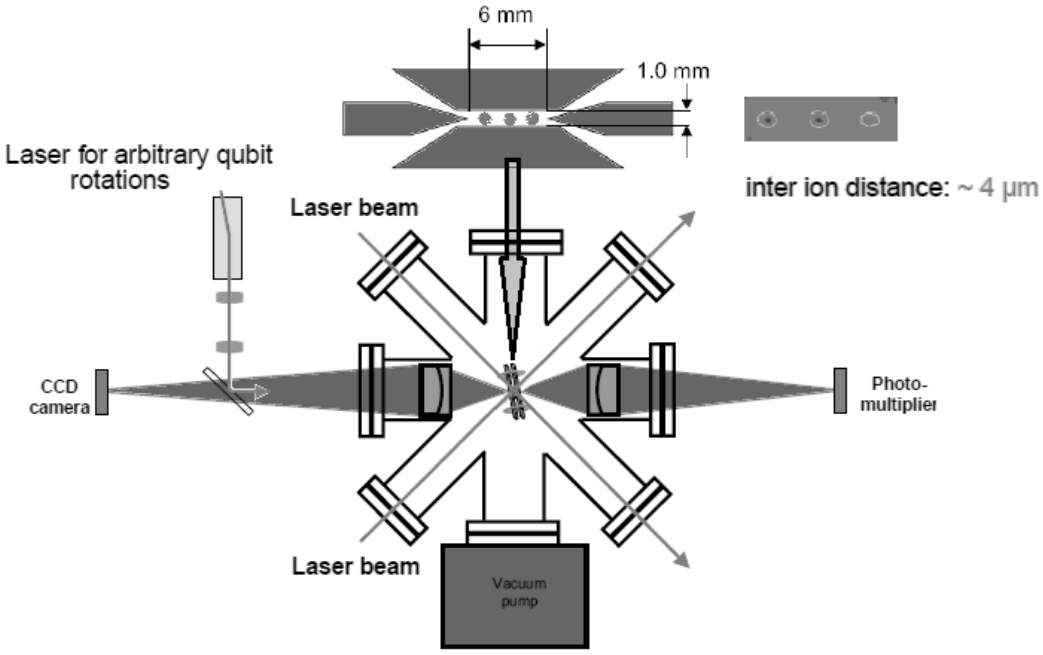


Figure 1-4. Schematic of ion trap and enclosure [6]

The actual ion trap is very small and has the dimensions of ~6mm from the tip of each end cap and only 1mm spacing between the magnetic elements. When the trap is activated, ions are loaded and squeezed together with a spacing of about ~4um (the more ions are loaded, the closer the spacing is).

The closer the inter ion spacing is, the more difficult it is to manipulate each ion independently. Figure 1-5 shows a linear ion trap alone.

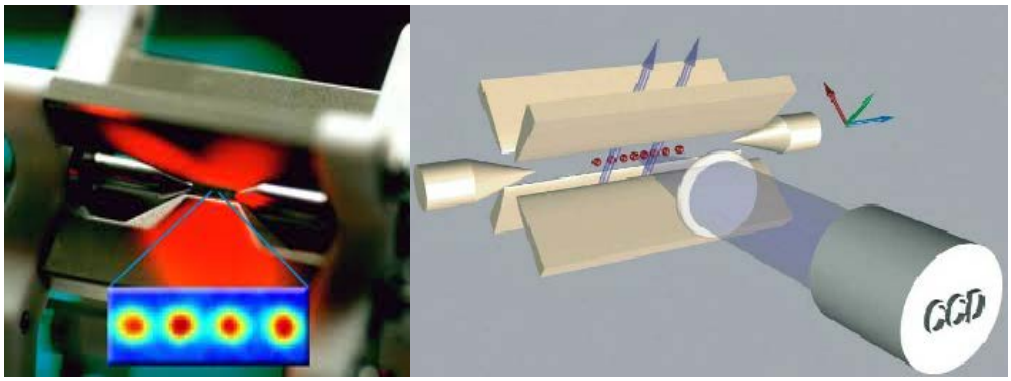


Figure 1-5. Ion trap from University of Innsbruck [6]

1.4.1 Ion Trap Qubit

The Ca+ 40 anion is commonly used since it offers many different electron orbital configurations, and has easily accessible energy levels via regular solid state lasers. Figure 1-6 shows the five lowest energy states of the Ca+ 40 anion.

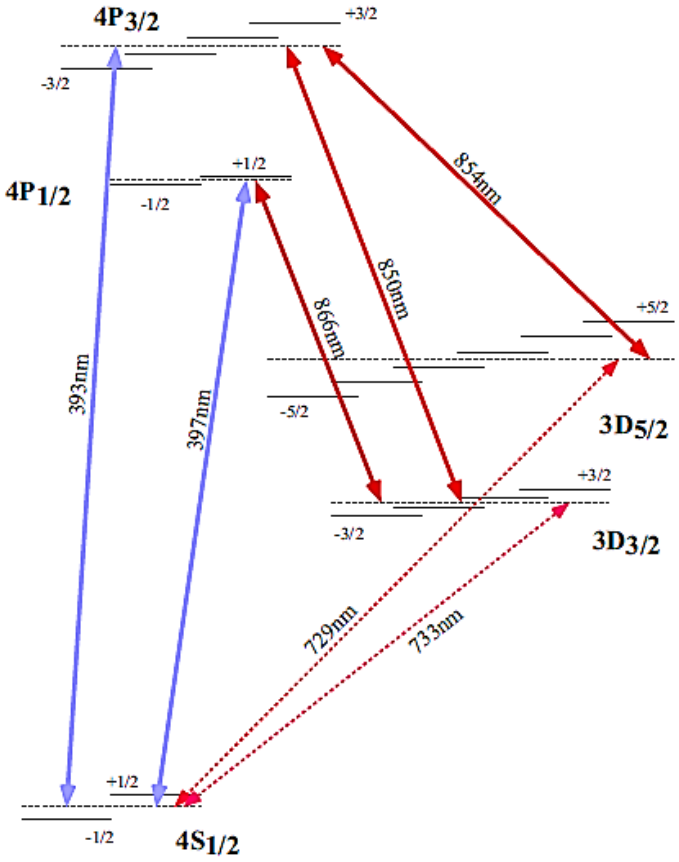


Figure 1-6. Energy diagram of Ca+ 40 ion and the laser connections [7]

A minimum of two energy levels are needed to implement a qubit, however higher logical levels can be acquired by using any of the additionally accessible energy states. In Figure 1-5, the distance between the energy levels corresponds to the amount of energy needed to change electron orbital configurations (larger distances require higher energy, or shorter wave length photons, and vice versa for shorter distances). For instance a blue laser, with a wave length of 397nm, can excite the ion so that electrons are sent from the $4S_{1/2}$ orbital to the $4P_{1/2}$ orbital. At this time the ion will spontaneously decay into a meta-stable state, such as the $3D_{5/2}$ state (the $4P_{1/2}$ orbital has a lifetime of 10 nanoseconds [8]). Note that each orbital has smaller

sub levels, called the Zeeman sub levels, and are made accessible by applying a constant magnetic field (the phase gate proposed by Cirac and Zoller will utilize the two lowest Zeeman levels from each orbital and an additional $3D_{3/2}$ auxiliary level). To read the quantum state of each ion, the ions must first be illuminated with a laser radiation of 397nm and if the ion is in the $3D_{5/2}$ state, it cannot be exited and will not emit photons. However, if the ion is in the $4S_{1/2}$, the laser radiation will excite the ion to the $4P_{1/2}$ and spontaneously decay, emitting photons, which are collected by the CCD camera.

To initialize the ions a 397nm cooling laser is used to move electrons to the outer orbital or to the $4S_{1/2}$ state. The $4S_{1/2}$ is the ground state, corresponding to the $|0\rangle$, or $|S\rangle$ state, and the $3D_{5/2}$ is second metastable state, corresponding to $|1\rangle$ or $|D\rangle$ state. When a Ca+40 atom is moving toward the laser source, it absorbs photons and slows down, effectively cooling the atom to near absolute zero temperature and enters its ground state. A second laser is detuned slightly below or slightly above the $3D_{5/2}$ transition state, forming two side bands of the desired wavelengths for the blue side band and the red side band laser. The side bands correspond to the energies required to move an ion in and out of the Zeeman energy levels and are in the order of a few MHz apart from the carrier laser wave length. Figure 1-7 shows the probability of exciting an anion into one of the Zeeman levels of the $|D\rangle$ state and the corresponding carrier laser and or its side band peaks:

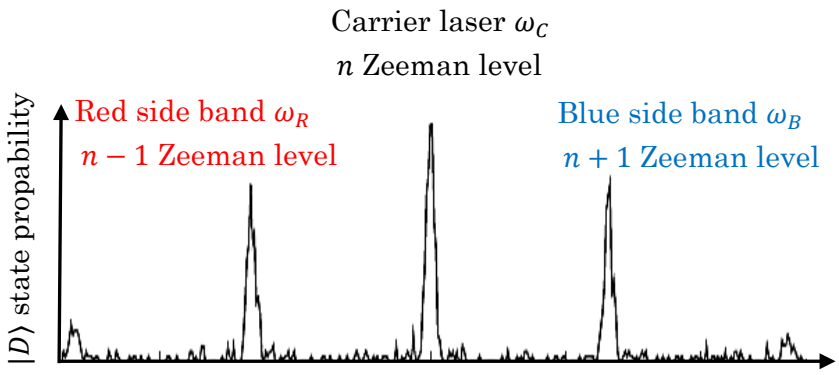


Figure 1-7. 729nm Laser Detuning (lower/upper sidebands) and state probability [6]

Changes in orbital configurations are accomplished by exchanging photon radiation with electrons. When an electron moves from a higher energy state to a lower energy state, it will release a photon, and when an electron is moved to a higher energy state, it will absorb a photon. By exposing the Zeeman levels, multiple ions can now exist in levels $|S,n\rangle$ and $|D,n\rangle$, also known as the phonon states, in addition to the electronic states of $|S\rangle$ and $|D\rangle$. Phonons are the energy quanta used to describe the collective motion of atoms, or molecules, that are condensed in matter or crystals. Transitions between levels are made with laser wave lengths of the red sideband, carrier, and blue sideband. Figure 1-8 shows how each level may be connected:

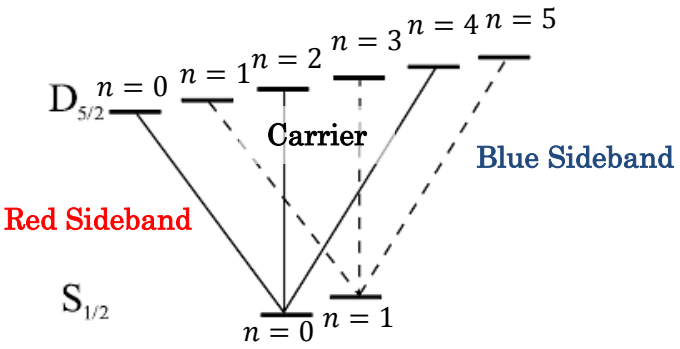


Figure 1-8. Ca+ 40 Vibrational/Phonon States and the laser side band detuning [6]

The Zeeman levels are a small quantum energy state in which ions, within a crystal, exhibit harmonic oscillations in addition to the electronic quantum states. So an ion at a given electronic quantum state can have an additional energy that sets the ion into a harmonic vibration. The harmonic oscillation pattern depends on the number of phonon quanta that are added or removed from the ion, as well as the atomic spin of the ion. The Zeeman levels for $n = 0$ (no phonons) and $n = 1$ (one phonon quanta), for the $3D_{5/2}$ and $4S_{1/2}$, are particularly important, as the $n = 0$ level has no oscillatory movement, while the $n = 1$ state exhibits a vibrational axial movement of approximately 6.3MHz, that affects adjacent ions, and therefore effectively adds phonon energy to all neighboring ions. This type of axial vibration literally couples an entire string of ions and is the precise mode of entanglement in which quantum phase data is sent to adjacent qubits. The vibrational states are Zeeman levels and are also called the center-of-mass (COM) modes and when

a qubit is moved to a COM mode, then all ions in the string or crystal are placed in the COM mode and begin to oscillate in a particular pattern. Cirac and Zoller utilize the lowest Zeeman level, COM mode (see Figure 1-8), to couple qubits to one another, for transferring phase information from qubit to qubit. The second Zeeman level is called a breathing mode and although not used by Cirac and Zoller gate, it can be used as a different ion coupling mode for other more complex gates. Figure 1-9 shows the Zeeman levels for phonons $n = 1$ and $n = 2$ and the motions induced in each ion string with respect to its total atomic spin (the ion crustal string exhibits internal waves of different types much like a block of gelatin):

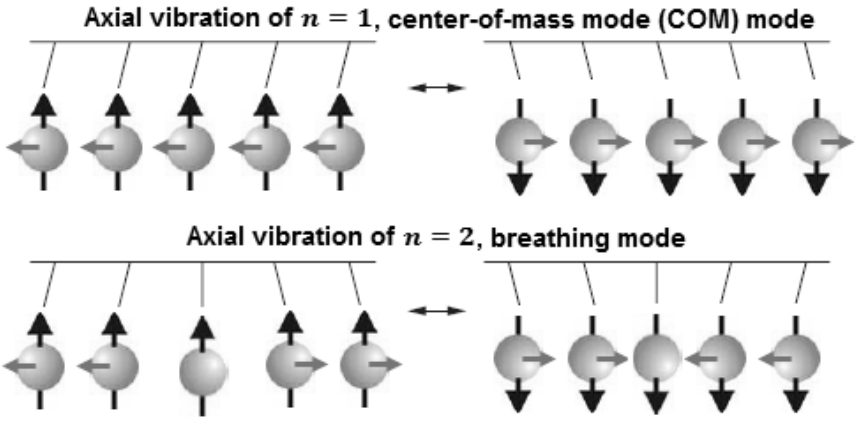


Figure 1-9. $n = 1$ and $n = 2$, ($n = 0$ is not shown as its movement is small and nearly stationary) [8], [6]

1.4.2 Ion Trap Logic Gates

Single qubit rotations are realized by exposing an ion to the carrier laser radiation for a specified amount of time, depending on the angle of rotation

desired. Electrons and laser photons exchange energies, electron orbitals are altered, and the ion's energy state is therefore reconfigured. The rotations can be represented by the following matrix:

$$R(\theta,\varphi)=\begin{bmatrix}\cos(\theta/2) & ie^{i\varphi}\sin(\theta/2)\\ ie^{-i\varphi}\sin(\theta/2) & \cos(\theta/2)\end{bmatrix} \qquad \text{Expression 1.1}$$

where the duration and intensity product of the laser pulse is θ and its phase difference to the ion is φ. A phase of $\varphi = 0, \pi, 2\pi, \ldots$ and $\varphi = \frac{\pi}{2}, \frac{3\pi}{2}, \ldots$ correspond to X and Y rotations respectively and as per Expression 1.1, the following matrices are derived:

$$R_x\theta = R(\theta,\pi)=\begin{bmatrix}\cos(\theta/2) & -i\sin(\theta/2)\\ -i\sin(\theta/2) & \cos(\theta/2)\end{bmatrix}$$

$$R_y\theta = R(\theta,\pi/2)=\begin{bmatrix}\cos(\theta/2) & -\sin(\theta/2)\\ \sin(\theta/2) & \cos(\theta/2)\end{bmatrix}$$

A phase rotation always accompanies an X or a Y rotation, whose angle is defined solely by φ. So the actual X and Y rotations are multiplied by the acquired phase rotation:

$$R_{ph}\varphi=\begin{bmatrix}e^{i\varphi} & 0\\ 0 & e^{i\varphi}\end{bmatrix} \text{ and } e^{\mp i\varphi} = cos\varphi \mp isin\varphi$$

Figure 1-10 shows a half of a π rotation about the X axis (X rotation), and creates a superposition of $|0\rangle$ and $|1\rangle$ states along with a phase of $-i$ (note

that the spheres with the arrows, represent electrons and their individual spins, and not the ion's atomic spin):

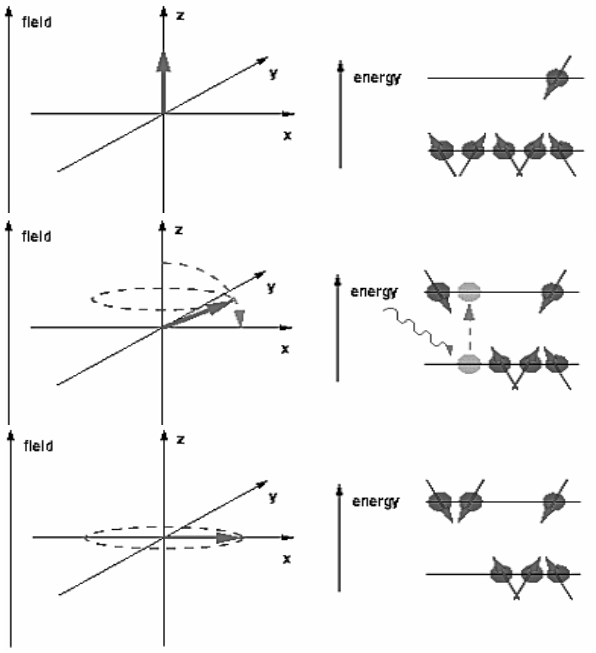


Figure 1-10. Single Qubit rotations by moving electron orbitals (the left side represents the vector rotation, the right side show how the electrons may be arranged on the two energy orbitals)

When the $Ca+\ 40$ ion is exposed to a coherent photon source, such as the carrier laser and or its side bands, the ion will absorb and re-emit photons in a cyclical fashion, known as the Rabi oscillations. In Figure 1-11A, a plot of one complete Rabi oscillation between the $|S\rangle$ and $|D\rangle$ states is shown (note that one complete cycle is 4π, due to the introduction of phase shift). Figure 1-11B. shows what Rabi oscillations look like in Hilbert space (a particular

time evolution of a Rabi oscillation can rotate the Eigen vector to any location in space):

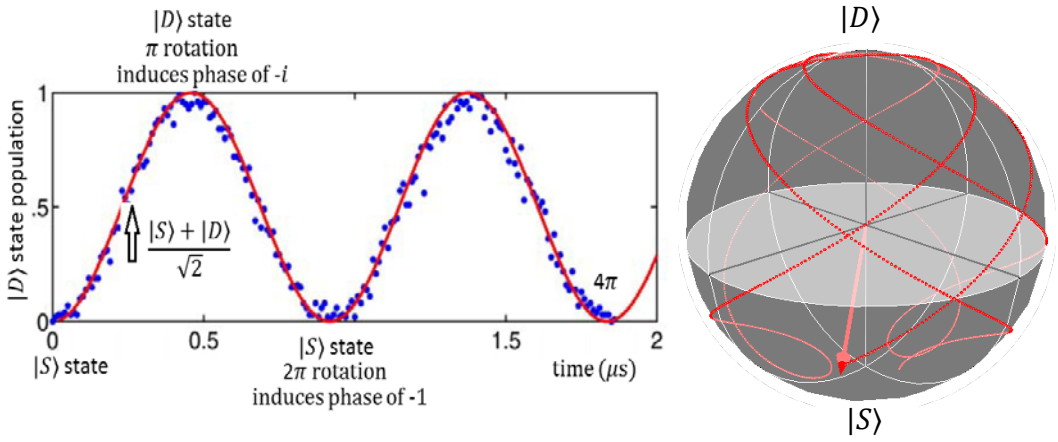


Figure 1-11. Rabi Oscillation between the $|S\rangle$ and $|D\rangle$ states (2-Dimesional A. and 3-Dimesional B.) [6]

There are several methods to realize a multi-qubit quantum gate. Cirac and Zoller were the first to propose the realization of the universal CNOT gate, by employing the two lowest Zeeman levels from the $4S_{1/2}$ and $3D_{5/2}$ orbitals and one additional auxiliary Zeeman level, from the $3D_{3/2}$ orbital [9]. The additional auxiliary state is denoted by $|D^*, 0\rangle$, and others are denoted by $|S, 0\rangle$, $|S, 1\rangle$, $|D, 0\rangle$ and $|D, 1\rangle$. Figure 1-12 shows how each energy state is connected to each other state:

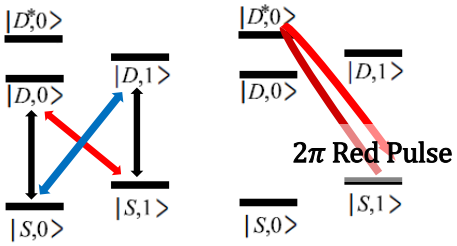
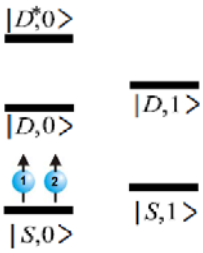
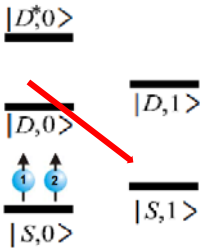
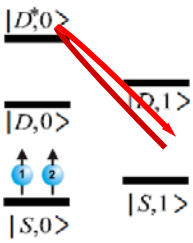
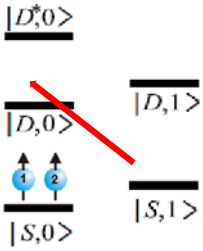
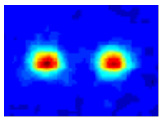


Figure 1-12. Ion Energy states transitions [6]

The lasers perform rotations on individual ions, however, when any ion is in the $n = 1$ vibrational mode, all of the ions are coupled to the COM mode. This means that phase differences can be induced by maneuvering ions into particular states and the COM mode. Table 1-4 demonstrates the process for implementing a Cirac-Zoller controlled phase gate (the rows represent the four possible states of the two qubit systems, and the columns represent the laser pulses that need to be applied):

Initial state, ions are cooled to $n = 0$	**Red π pulse on control ion (ion 1), $R_1^-(\pi, 0)$**	**Red 2π pulse on target ion (ion 2), $R_2^-(2\pi, 0)$**	**Red π pulse on control ion (ion 1), $R_1^-(\pi, \pi)$**	**CCD view of final state. Phase not observable**
 If ions are in state $\|SS\rangle$	 Pulse on ion 1 has no effect	 Pulse on ion 2 has no effect	 Pulse on ion 1 has no effect	 Ions remain in state $\|SS\rangle$

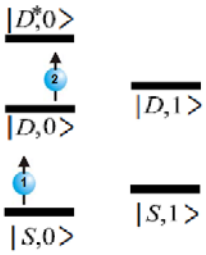
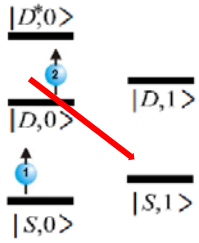
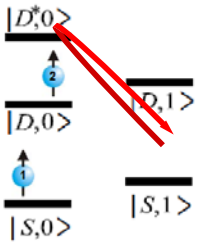
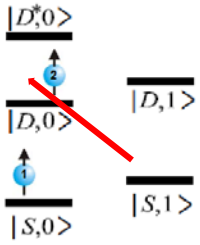
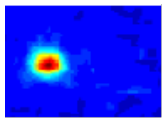
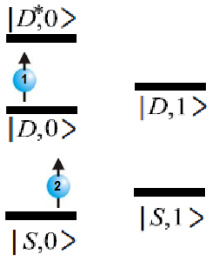
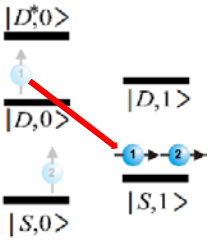
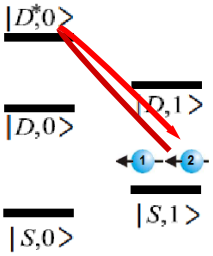
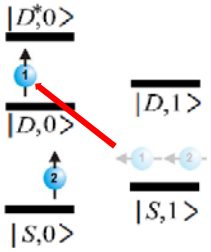
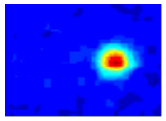
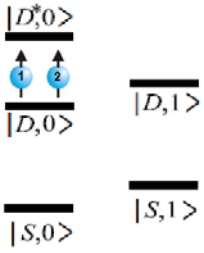
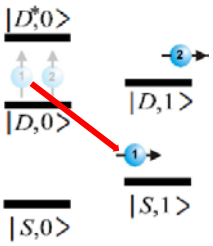
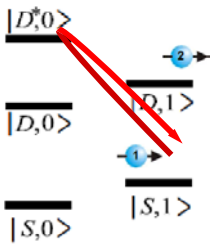
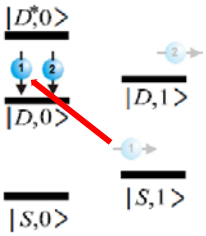
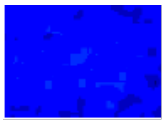
 If ions are in state $\|SD\rangle$	 Pulse on ion 1 has no effect	 Ions are unaffected by pulse	 Pulse on ion 1 has no effect	 Ions remain in state $\|SD\rangle$
 If ions are in state $\|DS\rangle$	 Ions enter COM and force i phase	 Ion 2 forces -1 phase to system	 Ions exit COM and force -i phase	 Ions are back in state $-\|DS\rangle$
 If ions are in state $\|DD\rangle$	 Ions enter COM and force i phase	 Pulse on ion 2 has no effect	 Ions exit COM and force -i rotation	 Ions are now in state $\|DD\rangle$

Table 1-4. Phase Gate Implementation

The combination of two $SWAP$ qubit operations, along with a composite 2π phase rotation is enough to create a controlled phase gate which is universal [10]. The expression below formulates the sequence mathematically:

$$U_{cp} = U_{SWAP(COM,Q_1)} e^{i\varphi(COM,Q_2)} U_{SWAP^{-1}(COM,Q_1)}$$

where φ, is a composite 2π phase rotation on the target qubit and typically requires three short laser pulses (the three pulse combination is not discussed in this thesis, as the exact order, duration, and intensities are still being researched and experimented with). Please note that since there are two working qubits, and a third internal vibrational qubit, all qubits are included in the matrices below. Since the aforementioned operations operate on two qubits and leave the third alone, each 4x4 matrix is converted to an 8x8 matrix, for all three qubits, by obtaining the tensor product (Kronecker product) of an identity 2x2 matrix and the 4x4 matrices.

The tensor product of a $SWAP$ gate and an unaltered qubit (note that a π rotation in an ion trap does not implement a standard $SWAP$ gate, however, placing a qubit from the $|D,0\rangle$ state to the $|S,1\rangle$ is a $SWAP$ instead of an X gate operation and since the duration of the pulse is π, a global phase is introduced to the qubit pair, so from expression 1.1, the result is $R(\pi,0) = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & i & 0 & 0 \\ 0 & 0 & i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

When one qubit, in the pair, is chosen as a reference phase of $\varphi = 2\pi$, the tensor product of an unaltered qubit and phase gate form a conditional phase gate. For this conditional COM phase gate a 2π redsideband pulse is applied as shown in Table 1-4 and any ion population only in the $|S,1\rangle$ state, which corresponds to $|0\rangle$ gains a phase of -1, $R(2\pi,0) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The last laser pulse operation performs another swap with a laser phase rotation and forms an inverse of $SWAP$, $R(\pi,\pi) = \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix}$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 \\ 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

As an example we can follow the multiplication of the matrices as shown below for U_{cp} implementation:

$$U_{SWAP(COM,Q_1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & i & 0 & 0 \\ 0 & 0 & i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$e^{i\varphi(COM,Q_2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$U_{SWAP^{-1}(COM,Q_1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 \\ 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$U_{cp} = U_{SWAP(COM,Q_1)} e^{i\varphi(COM,Q_2)} U_{SWAP^{-1}(COM,Q_1)} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The internal COM qubit is not used as a gate output and is only used internally to produce the controlled gates. Therefore the rows and columns that represent the internal COM qubit states can be removed from the above matrix, and the matrix is now reduced to the following:

$$U_{cp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 1-13 shows the equivalent quantum circuit:

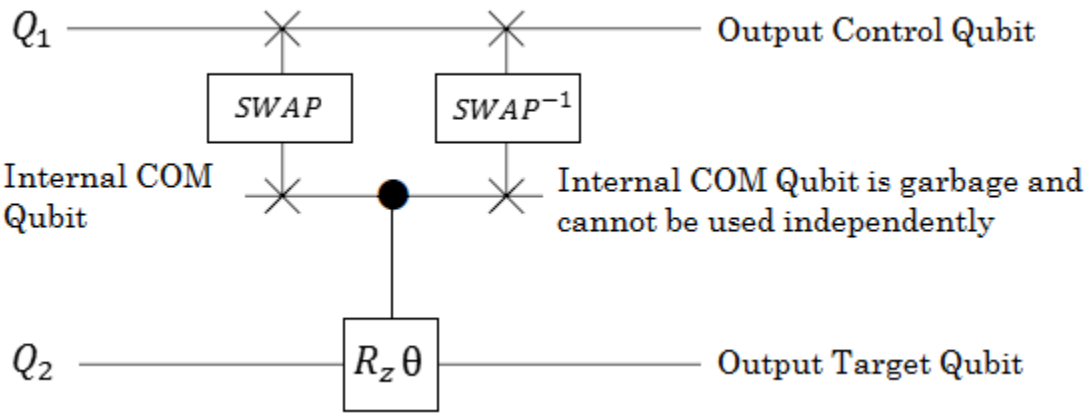


Figure 1-13. Controlled phase circuit (input is at left)

The truth table for U_{cp}, based on the circuit above, after an adjustment of a global phase of -1 is as follows:

Input Qubit states		Output Qubit states	
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	-1

Table 1-5. Controlled Z Gate Truth Table

Note the yellow output column indicates the qubit that realizes the function and the other output column is that of the control qubit, remaining unchanged in order to preserve reversibility.

It is important to note that the Cirac-Zoller gate is not the fastest implementation and significantly consumes the decoherence time of the system (each gate takes approximately a microsecond to implement, while the decoherence time of an ion trap is on the order of seconds). There are other variants that are much more efficient, however they are not included in this discussion because they are more complex and can be a thesis on their own. A controlled phase gate is totally universal along with individual qubit rotations. From this point forward, any complex gate can be realized. The ion trap is a very promising device for quantum computing and seems to adequately meet DiVincenzo's criteria, however it is far from a seamless transition for existing chip manufacturers, and therefore alternatives, such as quantum dots, are still vigorously being pursued. The next section will greatly elaborate on quantum dot systems.

1.5 Quantum Dot Quantum Computer

One of the main reasons that quantum dots make such an attractive device for quantum computers is the fact that quantum dot arrays can be

easily constructed on existing silicon wafer technologies. The semiconductor industry has invested billions of dollars in the design and millions of hours in the debugging of wafer processing machines. I imagine the industry will be very reluctant to abandon all of its manufacturing equipment and knowledge in order to switch into the manufacturing of quantum computers. Without a smooth, more continuous transition, into the existing manufacturing processes, quantum computers, I believe, will remain as computational tools for large research entities only (mainframe computers were once used in this type of capacity).

For the aforementioned reasons, quantum computers are desired, especially ones that can be constructed using SiGe or GaAs lateral quantum dots (quantum dots arranged in a single layer) on silicon wafers. For instance, to build SiGe quantum dots a slightly n-doped substrate can be used on to which a thick (1.5um) gradient of epitaxial SiGe is grown. The SiGe gradient starts at ~95% Si and 5% germanium and tops off at ~65% Si and 35% Ge. At this point the SiGe film is compressively strained and will transfer tensile strain to any film deposited above it. Another ~500nm of SiGe is grown on top to relax some of the strain so that a thin <10nm film of n+ doped Si can be grown with no strain defects (overly strained films can be stretched and distorted until cracks occur). This ~10nm film forms a sheet with high donor mobility, known as a two dimensional electron gas (2DEG)

and confines electron movement to only 2 spatial dimensions. The 2DEG film is then capped with an additional 30nm 70/30 SiGe film and a final SiO2 insulator. The 2DEG film has a low band gap energy and is now sandwiched between two layers of SiGe, which have higher band gap energy, to form the quantum well [11]. Figure 1-14 shows such a heterogeneous structure, without the thick SiGe gradient.

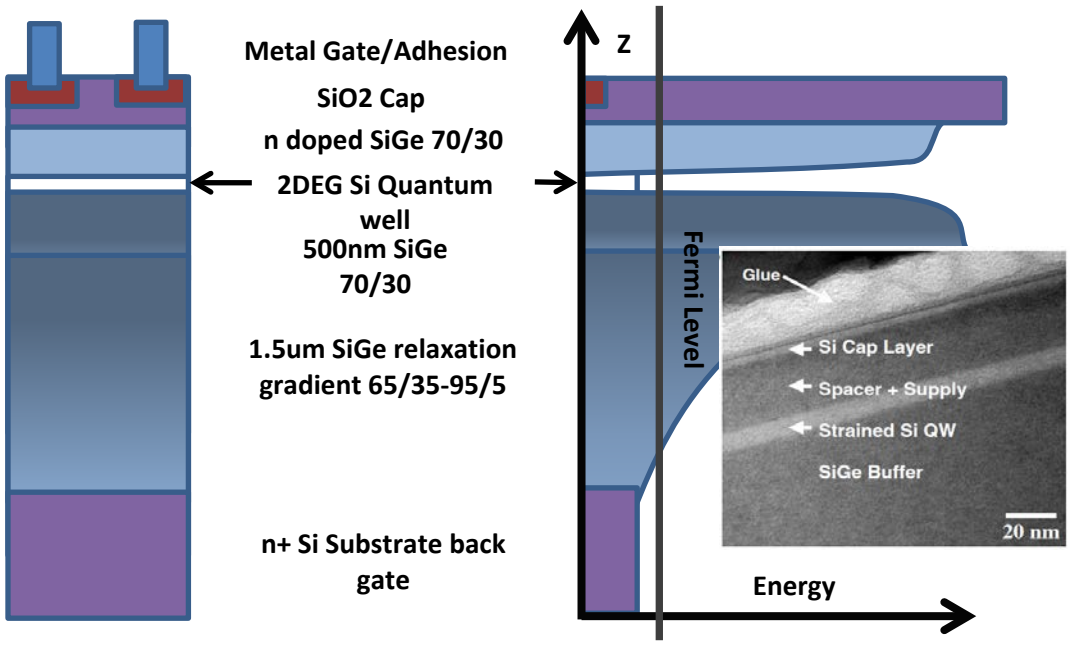


Figure 1-14. Left side shows the strained quantum dot films stack structure and strained quantum well formed in the 2DEG on the right side [11]

In the heterogeneous SiGe film stack (shown above), electron movement is only confined to two dimensions, but an electron confinement in all three spatial dimensions is required. To achieve the additional confinement in the X and Y directions in the 2DEG film, metal electrodes are placed on top of the

SiO2 film. The electrodes are then used to electrostatically confine the position of the electrons. Figure 1-15 shows the gate layout and the relative location of two quantum dots.

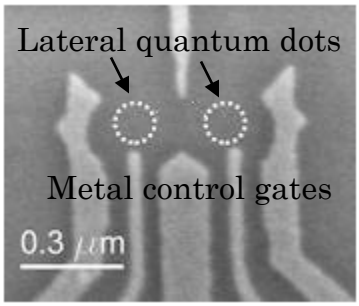


Figure 1-15. A two Quantum Dot device layout [12]

The electrodes basically create an additional energy depression within the 2DEG film The electrodes at the center of Figure 1-15 above are used to manipulate the distance between the quantum dots, which will allow for entanglement.

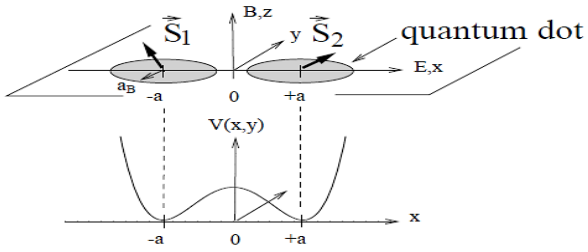


Figure 1-16. X and Y confinement diagram [13]

Only two qubits are shown in Figures 1-15 and 1-16, as most quantum dot systems are still in research and development, and functioning quantum dot arrays have not been constructed yet. However, once simple two qubit

operations have been mastered, a quantum dot array can be constructed as shown in Figure 1-17. Actually, entire 3D structures of quantum dots can be eventually constructed, but this type of system is probably decades away.

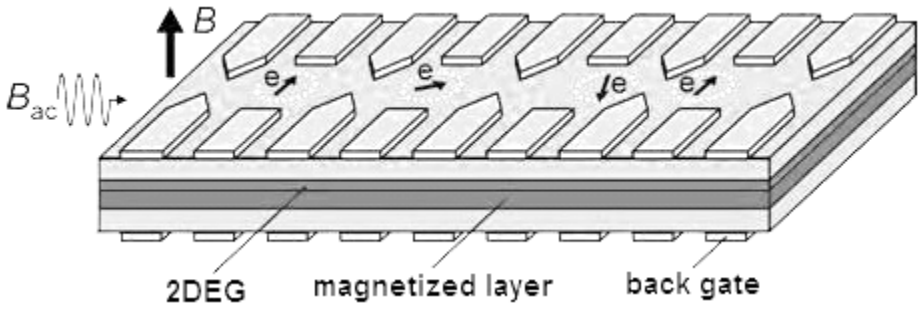


Figure 1-17. Two dimensiona lateral array of QD's [14]

The magnetic field shown in Figure 1-17 is used to initialize the quantum dots to a known state. Similar fields will have to be used on individual QD's to perform individual operations.

1.5.1 Quantum Dot Qubit

Large groups of trapped electrons are closely packed together and in unison begin to behave like a single electron wave function. Figure 1-18 shows an example.

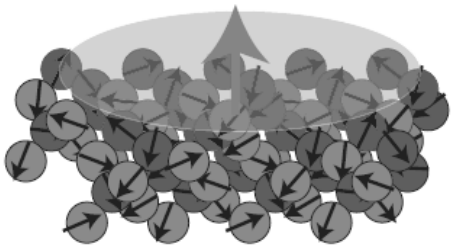


Figure 1-18. QD cluster formed out of many electrons or holes [15]

This type of grouping is, to some degree, process variation tolerant, especially when the location and the potential on the electrodes define the exact size and location of the quantum dot. Unfortunately this also means that each dot has to be individually tuned so that it is identical to other adjacent dots in the system. So the number of electrons trapped in a dot directly relates to the difficulty of utilizing the dot. Fewer electrons are desirable, but the electrode feature size and measurement equipment need to be precise enough in order to utilize smaller systems. Another very important feature of the quantum dot is the ease at which adjacent dots can be brought together for qubit-to-qubit interaction. By reducing the negative potential on the electrodes that separates two adjacent dots, and increasing the negative potential on each outside electrode, the dots can be pushed closer together (this is something that the ion trap cannot easily accomplish). As per Pauli's exclusion principal, no two identical fermions can occupy the same quantum state. This means that as each dot moves closer, each dot will assume opposing spins, or in other words, the total wave function of the two dot system will be anti-

symmetrical. Figure 1-19A and B show such an exchange occurring between two dots.

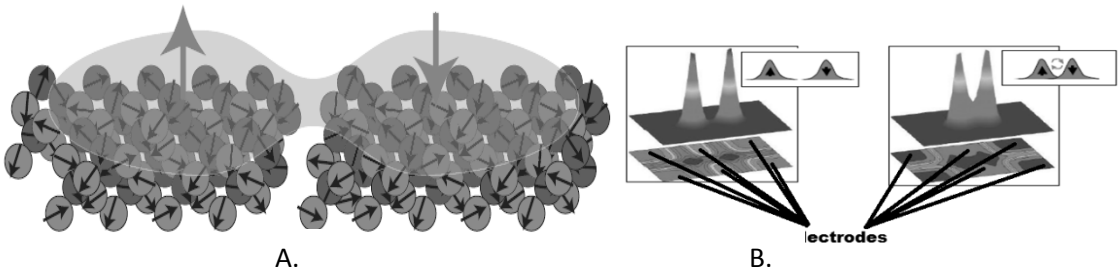


Figure 1-19. Two QD clusters pushed together for exchange, A. and the E-field mapping, B. [15]

This exchange is called the Heisenberg exchange and will be further explored in the next section, where multi-qubit gates are realized. Unfortunately the Heisenberg exchange is not enough to create universal gates. Individual electron manipulations are still required and remain a major challenge. Some proposals include the integration of a metal loop that surrounds a quantum dot, which can generate a sufficiently strong magnetic field without affecting adjacent dots. Figure 1-20 shows one implementation of a magnetic probe that can be potentially built and used to alter the spin of individual dots [12].

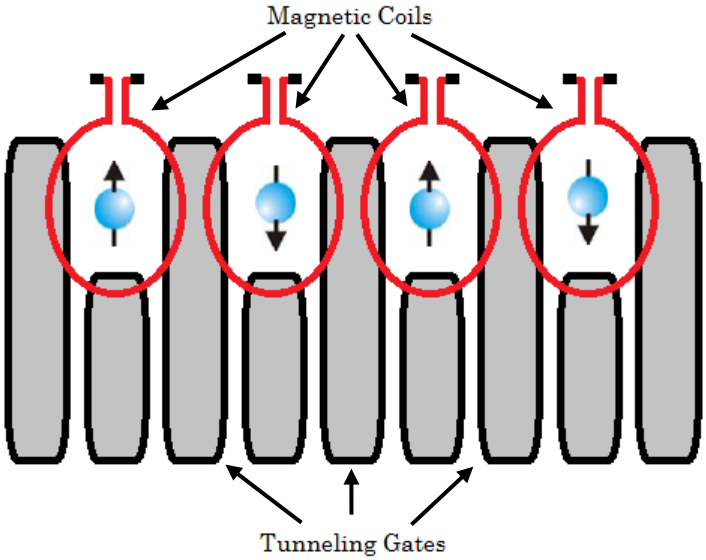


Figure 1-20. Four qubit system surrounded by electromagnetic elements that can alter individual electron spin direction

To my knowledge such a probe has not been realized yet. Another approach is to use optical pumping [16] and has shown more success, but again building such a structure on-chip will be a huge challenge. Nonetheless, theoretical and some experimental results have shown that single electron spins can be altered along any arbitrary axis, with a combination of an oscillating magnetic field, and optical pumping. Figure 1-21 shows a hypothetical Bloch sphere and the rotations that can potentially be achieved.

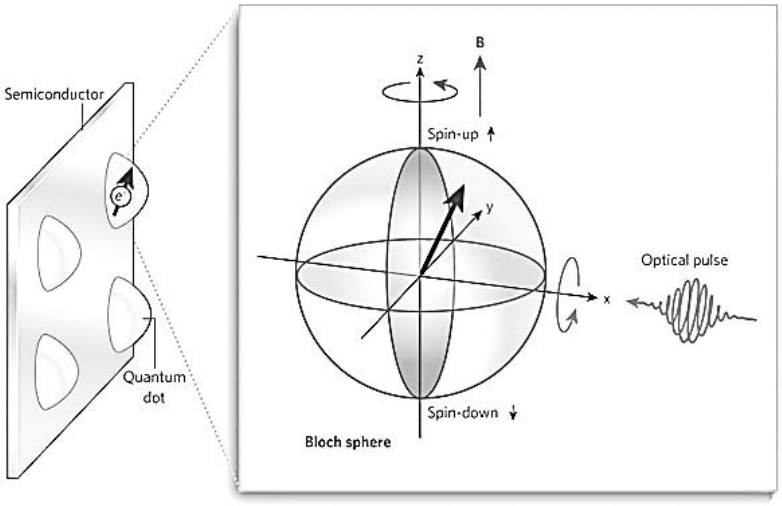


Figure 1-21. Visualization of a quantum and its corresponding Bloch sphere rotation representation [17]

Reading the spin information of quantum dots is also a challenge as on one hand quantum dots need to be isolated from the surrounding environment, to prevent decoherence, and on the other hand fast and accurate measurement devices need to have unobstructed access to the spin information. One popular approach is to use single electron transistors, SET's, as a means of extracting spin information for individual dots [18]. Since the quantum dot spin configurations are spatially distinct, a SET device built next to the quantum dots can detect the spin information. This is due to the fact that static gate potentials and the orientation of the electron spin changes the shape or contour of the electron probability density shape, as shown in the Figure 1-22 Different electron density shapes induce different charges on the SET island and the SET becomes an extremely sensitive electrometer.

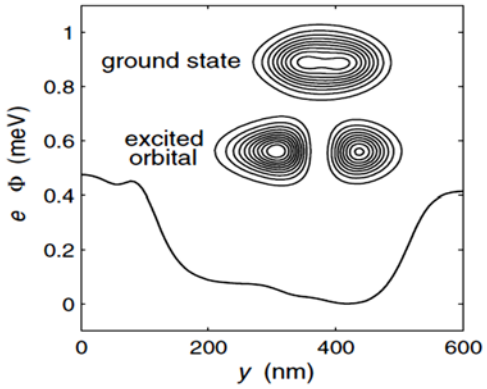


Figure 1-22. Contour plots of the electron probability densities, showing a detectable difference between the ground and excited states. The difference in the densities can be detected by SET transistors [18]

DiVincenzo et al have mathematically shown that rotations about the Z axis, along with the Heisenberg exchange are enough to realize any universal gate. And as long as the universal gate states have an accurate method for reading its quantum states, a quantum computer can be realized from quantum dots.

1.5.2 Quantum Dot Logic Gates

Heisenberg exchange can be utilized as suggested by DiVincenzo et al to realize universal quantum dot quantum gates. In order to keep the next derivation simple, it is assumed that we have a perfectly isolated quantum dot system that will not decohere due to coupling from the adjacent environment. Creating a model that includes environmental coupling, is a thesis on its own. Recall that the center electrode in a quantum dot system can be used to maneuver the quantum dots closer to one another. We can call this element the tunneling element as it effectively allows a quantum dot

containing higher energy to tunnel through the 2DEG film into the adjacent dot. So if tunneling is allowed to evolve for some small duration of time, the Hubbard Model specifies that a spin system will be subjected to a transient Heisenberg coupling [10]:

$$H_s(t) = J(t)\overrightarrow{S_1} * \overrightarrow{S_2}$$

Where $\overrightarrow{S_i}$ is the spin-1/2 operator for dot i, $J(t) = \frac{4{t_0}^2(t)}{u}$ is a time-dependent exchange constant, $t_0(t)$ is the on and off tunneling element function, and u is the charging energy of a single dot. If sufficient time is given to the system, the states of each dot will evolve into exactly the opposite direction. This is referred to as a total quantum swap. Unfortunately a total quantum swap is exactly what happens when such a system is measured and completely detangles the qubit system. However if the tunneling barrier is lowered for a duration of half of a full swap ($SWAP^{1/2}$), then some exchange between the dots has occurred and entanglement is maintained, so future operations can be performed. The effect of the pulsed Hamiltonian that applies a particular unitary time evolution operator to the initial state of the two spins is given by:

$$U_s(t) = Te^{\{-i\int_0^t H_s(t')dt'\}}$$

$$\Psi|(t)\rangle = U_s\Psi|(0)\rangle$$

A specific duration of the pulsed Heisenberg coupling leads to a special form of U_s. For $J(t) = \pi$, $U_s = U_{swap}$, which is a full swap and is not a universal operation. For $J(t) = \pi/2$, $U_s = U_{swap^{1/2}}$, which is a square root of a swap and is a universal operation. Figure 1-23A. shows a full swap evolution and Figure 1-23B. shows square root of a swap of an incoming state and the output state. The plots also show the time pulse on the tunneling gate as well as a calculation of the entangling of the system. The tunneling amplitude is denoted by $t_h(t)$, the square amplitudes of $|S_1\rangle$ and $|S_2\rangle$ are denoted by $|\varphi_1|^2$ and $|\varphi_2|^2$, the measure of entanglement is denoted by $\eta(t)$ [19].

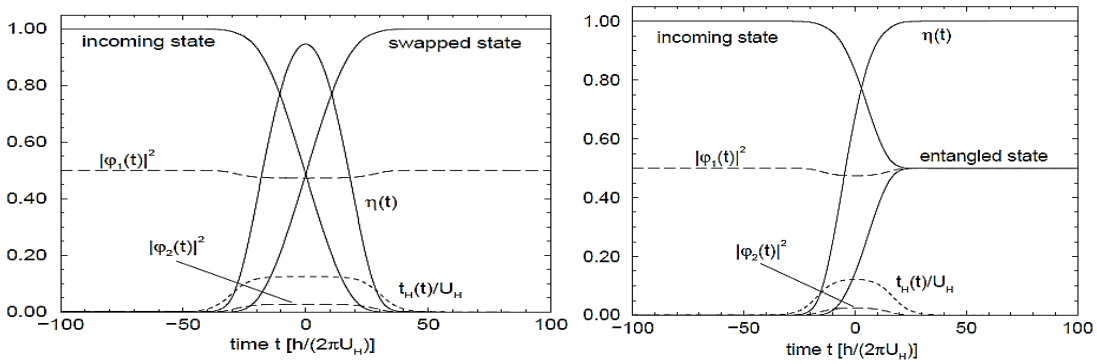


Figure 1-23. Full Swap and Square Root of Swap [19]

Exactly half of a swap (square root of swap) is ideal but not totally necessary and any deviation will be exposed as a reduction in the gate fidelity or probability of the resultant qubit states. The $U_{swap^{1/2}}$ is in fact is similar to the V gate mentioned earlier, except that it acts on two qubits. Finally, the combination of three single qubit rotations, along with two $SWAP^{1/2}$

operations is enough to create a controlled phase gate. Controlled phase gates along with Hadamard gates can inter create a CNOT gate, which is universal [10].

$$U_{cp} = e^{i\pi/2{S_1}^z} e^{-i\pi/2{S_2}^z} U_{swap^{1/2}} e^{i\pi{S_1}^z} U_{swap^{1/2}}$$

where $e^{i\pi/2{S_1}^z}$ is a π/2 Z rotation on qubit one, $e^{-i\pi/2{S_2}^z}$ is a -π/2 Z rotation on the second qubit, $U_{swap^{1/2}}$ is half of a swap, and $e^{i\pi{S_1}^z}$is π Z rotation. Please note that since there are two qubits, each 4x4 matrix is the Kronecker product of an identity 2x2 matrix that represents no change to one qubit and the 2x2 matrix of the other qubit that is manipulated.

Kronecker product of an unaltered qubit and a phase rotation:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\varphi} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{bmatrix}$$

Kronecker product of a phase rotation and an unaltered qubit:

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\varphi} & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{bmatrix}$$

As an example we can follow the multiplication of the matrices as shown below for U_{cp} implementation:

$$e^{i\pi/2{S_1}^z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

$$e^{-i\pi/2{S_2}^z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -i \end{bmatrix}$$

$$U_{swap^{1/2}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (1+i)/2 & (1-i)/2 & 0 \\ 0 & (1-i)/2 & (1+i)/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$e^{i\pi{S_1}^z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$U_{swap^{1/2}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (1+i)/2 & (1-i)/2 & 0 \\ 0 & (1-i)/2 & (1+i)/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$U_{cp} = e^{i\pi/2{S_1}^z} e^{-i\pi/2{S_2}^z} U_{swap^{1/2}} e^{i\pi{S_1}^z} U_{swap^{1/2}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Figure 1-24 shows the equivalent quantum circuit:

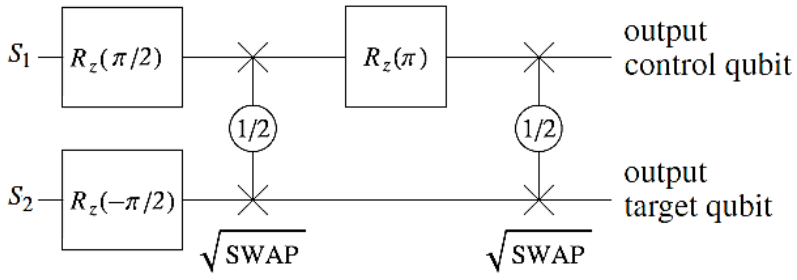


Figure 1-24. Controlled phase circuit (input is at left)

We can then use U_{cp}, sandwiched between two Hadamard gates to create a CNOT, or an XOR quantum gate, also known as the famous Feynman gate.

$$U_{xor} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{bmatrix} * \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{bmatrix}$$

$$= \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

Figure 1-25 shows the circuit realization. Note that the Z symbol represents a controlled phase rotation of the U_{cp} circuit:

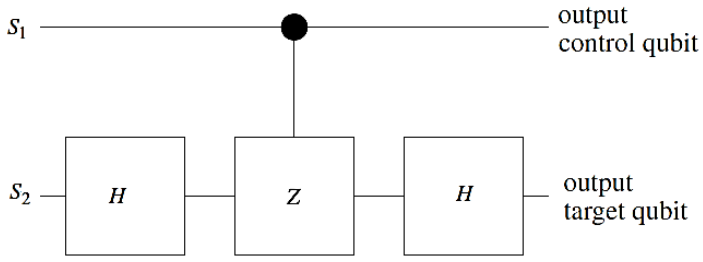


Figure 1-25. Representation of XOR circuit, using U_{cp}

The truth table for U_{xor}, based on the matrix above is as follows:

Input Qubit states		Output Qubit states	
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table 1-6. Controlled NOT/XOR Gate Truth Table

In this section we have seen the realization of quantum gates created from quantum dots and how one may use the gates to realize any arbitrary function. Please keep in mind that this text is a rather modest introduction to quantum computing. It should be pointed out here that the expected decoherence time for quantum dots is on the order of a millisecond while the qubit rotations have been simulated to be in the order of several hundred picoseconds. Though the ratio is large and sufficient to implement thousands of gates before decoherence, there is a finite time duration. Greater efforts are being made so that complicated functions can be implemented with as few gates as possible. Just as there are methods for optimizing classical circuits, similar methods need to be invented for quantum circuit optimizations.

1.6 Conclusions and Outstanding Research

As we have seen, quantum dots provide us with many potential applications. Before colloidal quantum dots can be utilized in the commercial sector, they must be fabricated totally cadmium free and devoid of any other toxic compounds. We most certainly do not want to compromise our health and environment for a few incremental improvements in our everyday electronic devices. Quantum dots in the biomedical field need to be also used with great caution, as we currently do not have a full understating of the potential dangers associated with the accumulation of nanoparticles in living organisms. Some experiments have found that quantum dots are not entirely innocuous and can interfere with cell DNA. As for quantum dot based quantum computers none have been realized to date, but there are many prestigious universities conducting relentless research in constructing solid state quantum computers. Currently ion trap devices are most popular and have been successfully used to perform powerful quantum algorithms, utilizing superposition. Despite some skepticism, the research community is in total agreement that quantum computing has been accomplished. I should mention that quantum computers are not a simple replacement for our classical computers, but can provide a tremendous computation boost when needed. I look forward to a future when quantum computers become a more popular buzz word.

As mentioned in Section 1.2 of this chapter, the thesis will shift topics and begin to explore methods for synthesizing quantum circuits using the aforementioned two technologies. The next chapter will review group decomposition and the Walsh spectrum for specifying classical reversible circuits, but in Chapter 3 and 4 all of the material present will coalesce together and form efficient synthesis methods for specifying quantum reversible circuits.

Chapter 2

MULTI-VALUED CANONICAL CASCADE REALIZATION WITH GROUP DECOMPOSITION

In this chapter I will introduce a method for realizing binary controlled ternary output functions by using a canonical cascade of cells derived from decomposing group functions and the Walsh spectrum. This methodology was first developed by Tsutomu Sasao and presented in [20] for classical reversible logic, but the method can be efficiently extended to quantum cascades, as will be shown in the next chapters.

2.1 Group Theory

A Group $\langle G, ., I \rangle$ is an algebraic structure consisting of a set of elements with an operation that combines any two elements to form a third element. In addition closure, associativity, identity and invertibility must be satisfied. Each is defined as follows:

a) Closure - For any $a, b \in G$, $a * b \in G$

b) Associativity - For any $(a * b) * c = a * (b * c)$

c) Identity - For any $a \in G$, a unique element I exists such that

$a * I = I * a = a$

d) Invertibility - For any $b \in G$, an element b exists such that $a * b = b * a = I$, where b is an inverse element, a^{-1}

When every element in a group is a power of a fixed element a in group G, then a cyclical group is formed. For example, if $G = \{a^0, a^1, a^2, a^3, a^4\}$ is a finite group and $a^0 = a^5$, then G is a cyclical group of order 5 or $modulo\ 5$. For completeness, the order of a group is equal to the number of unique elements (please note that an infinite cyclical group can also be created, when the number of elements extends to infinity). When a group consists of all of the permutations of its elements, then the cyclical group is symmetrical, otherwise the finite set is dihedral. In the aforementioned example, if each element is a single shift permutation (1⟶2, 2⟶3, …, 4⟶0) than the permutations are non-exhaustive (double shift 2⟶4 permutations are also possible) and the group formed is dihedral. If $G = \{a^0, a^1, a^2\}$, $a^0 = a^3$, and each element is a single shift, then G is a symmetrical cyclic group (double shifts are not possible as the double shift of 0⟶2 is the same as a single shift of 3⟶2, as long as $a^0 = a^3$,). In order to utilize group functions for realizing logical functions, we must avoid symmetries in which two different input combinations can produce the same logical output value. This means that we need to create cyclical groups (either symmetrical or dihedral) that have non-abelian properties (non-commutativity for any $a, b \in G$, $a * b \neq b * a$). To satisfy this requirement, we will utilize cyclical groups of any prime degree

(G_p, where p is any prime number) and combine them to generate non-abelian elements. The process is expanded in the next section.

2.2 Group Functions

In this section we will generate all of the elements required to produce a prime degree group, with non-abelian elements. We will use two groups, one that performs single shifts, and one that will perform a conditional shift direction change.

2.2.1 Cyclic Group Generation of Order Three

Generating a cyclic group of order three is ideal in realizing ternary logic, since each element can perform single shifts among three logical values. Combining two elements together is nothing more than a simple multiplication of the elements. The element shown in Figure 2-1 is the identity element, which leaves the input values unchanged and can be represented as simple wires:

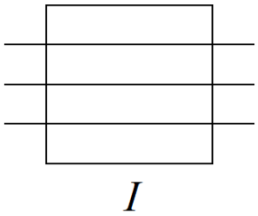


Figure 2-1. Identity Group Element

If a shift is desired between the input lines or values, than cell a can be applied to produce a (see Figure 2-2):

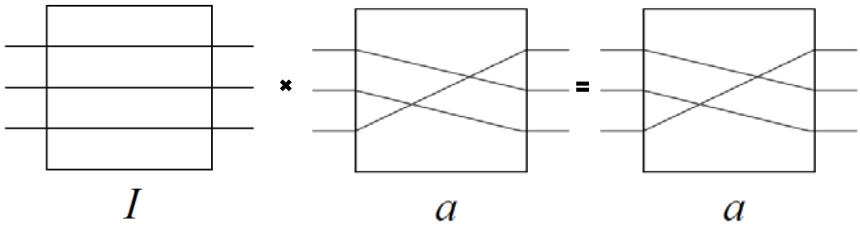


Figure 2-2. a Group Element

If an additional shift is desired, then cell a can be applied once again to a, to produce a^2 (see Figure 2-3):

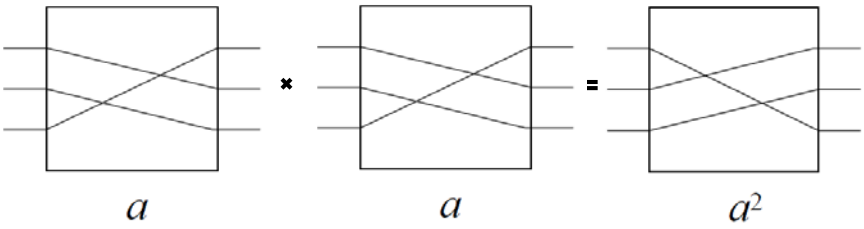


Figure 2-3. a^2 Group Element

If another a is applied to a^2 than a^3 is produced, which is the same as I (see Figure 2-4):

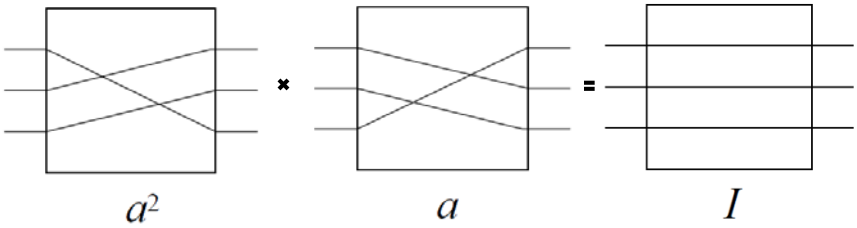


Figure 2-4. a^3 Group Element is also the Identity Element

Thus we have created all of the elements required to produce $C_3 = \{I, a, a^2\}$, a cyclic group of order three. Figure 2-5 shows the map that is derived (note that $a^0 = a^3 = I$ and $a^2 = a^{-1}$ in module 3):

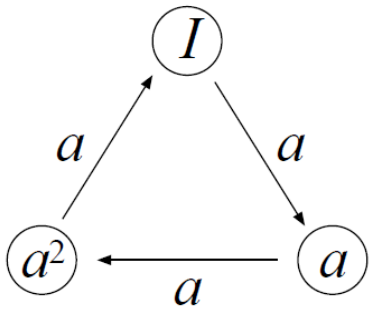


Figure 2-5. C_3 Group Map

2.2.2 Cyclic Group Generation of Order Two

Next we need to generate a group that be controlled with simple binary logic, which means that it needs to be a cyclic group of order two (zero state corresponds to no change and a logic one corresponds to the application of a group element). In addition it needs to generate non-abelian elements and ultimately allow us to traverse all output logical levels in any direction. The g element is introduced, which shifts the bottom two states and leaves the first untouched so that $I * g = g$ and $g^2 = I$ (see Figure 2-6):

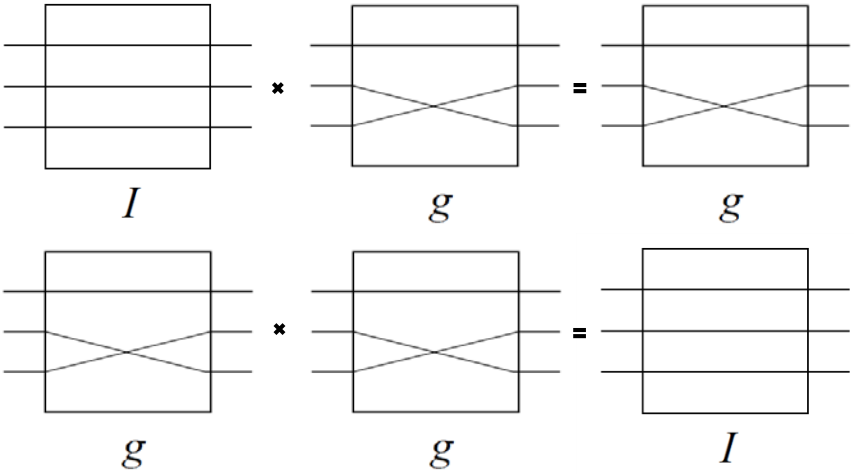


Figure 2-6. C_2 Group Elements (Identity, g and g^2)

Thus we have created $C_2 = \{I, g\}$, cyclic group of order 2. Figure 2-7 shows the map that is derived (note that $g^0 = g^2 = I$ and $g = g^{-1}$ in module 2):

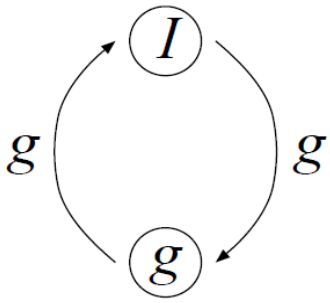


Figure 2-7. C_2 Group Map

2.2.3 Symmetrical Cyclic Group Generation of Degree Three, Order Six

A symmetrical group of degree three, order six is formed when C_2 and C_3, from the previous two sections are combined to form the following:

$S_3 = C_3 * C_2 = \{I, a, a^2\} * \{I, g\} = \{I, a, a^2, g, ag, ga\}$ (see Figure 2-8):

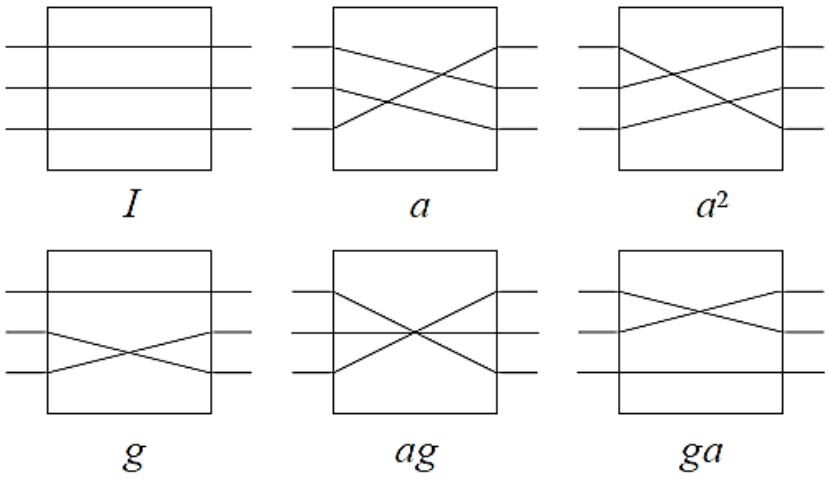


Figure 2-8. S_3 Group Elements ($I * g = g, a * g = ag, a^2 * g = ga$)

After multiplying every element from C_2 with every element from C_3 and eliminating redundant elements, such as $a^2 g = ga$, the group $S_3 = \{I, a, a^2, g, ag, ga\}$ is formed. Note that all of the elements are unique and that non-abelian elements, ag and ga are formed ($ag \neq ga$). Figure 2-9 is the map that is generated:

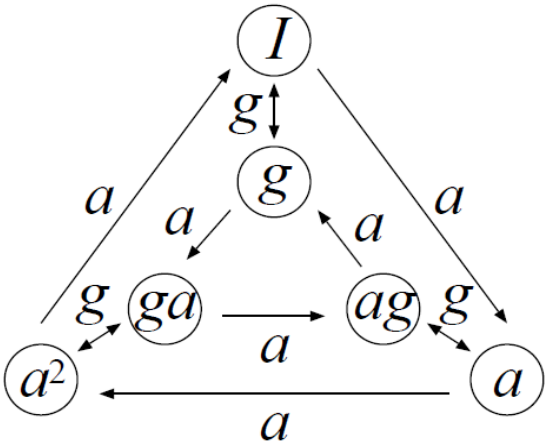


Figure 2-9. S_3 Group Map

Upon inspection of the map we can see that the combination of the two cyclic groups creates a counter clockwise path where g acts as a reflection. From the map we can also observe that $gag = a^{-1}$ and $ga^{-1}g = a$ and therefore form reflection about the vertex. The application of the reflection element effectively conjugates the a^n element. These observations will become more useful as reduction strategies will be introduced later in this chapter.

2.3 Group Function Decomposition

Before we see the mechanical application of group decomposition to actual circuits, we need to establish a general expression for group decomposition that will be used throughout the rest of the chapter.

A group function $F(x) : B^n \rightarrow C_3$ decomposes as follows:

$$F(\hat{X}, x_n) = F_a(\hat{X}) g^{x_n} F_b(\hat{X}) g^{x_n}, \quad (2.1)$$

where x_n is a two-valued input variable and $F_a(\hat{X})$ and $F_b(\hat{X})$ are group functions $(\hat{X} = \left(x_{1,\ldots,} x_{m-1}\right) \in X^{m-1})$ that do not depend on any x_n input variable (proof for 2.1 can be found in [21]). The decomposition is synonymous to Shannon's expansion for a classical binary logic function and is essential in the design of canonical cascades. Figure 2-10 the canonical cascade function of 2.1:

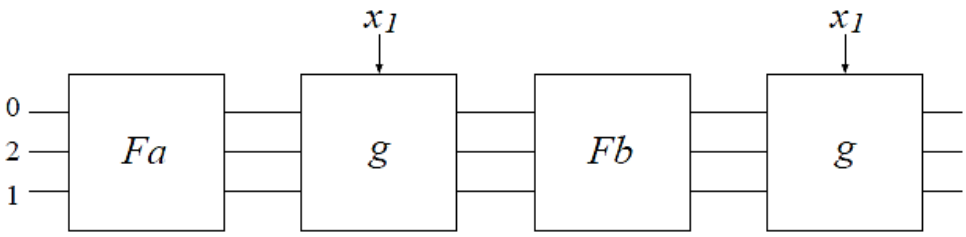


Figure 2-10. One Variable Canonical Cascade

2.3.1 Group decomposition applied to a single input variable canonical cascade

In this section the g element is modified so that a control line can be added. The section then proceeds to derive the Walsh matrix and the Walsh spectrum based on group decomposition from Section 2.3. The section then finishes with a simple one input variable example.

The decomposition indicates that element g needs to be modified so that it can be controlled with a binary input variable. When $x = 1$, g^x swaps the bottom two lines and when $x = 0$, g^x performs the identity permutation. Figure 2-11 shows a representation of g^x (note that the g^x element closely resembles a reversible controlled swap or Fredkin gate [22]):

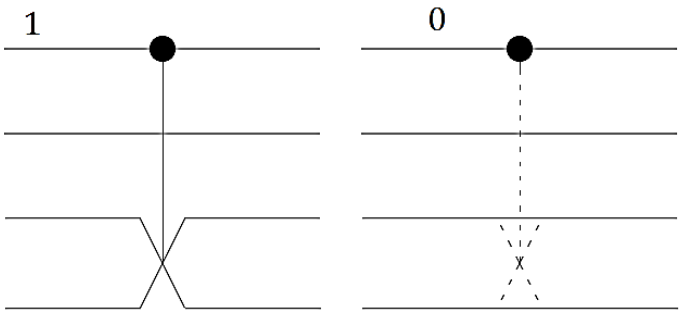


Figure 2-11. g Element modified to g^{x_n} so that it can be controlled

Since $g \in C_2$ and F is group function $B^n \rightarrow C_3$,

then $F(\hat{X}, x_n) = F_a(\hat{X}) g^{x_n} F_b(\hat{X}) g^{x_n}$, as stated before, and $F_a(\hat{X})$ and $F_b(\hat{X})$ are replaced by the simple shift elements from Figures 2-1 through 2-4, depending on their specified exponents:

$$F_a(\hat{X}) = a^{f_a(\hat{X})} \text{ and } F_b(\hat{X}) = a^{f_b(\hat{X})}$$

$$F(\hat{X}, x_n) = a^{f_a(\hat{X})} g^{x_n} a^{f_a(\hat{X})} g^{x_n}$$

Now $F(\hat{X}, x_n)$ is evaluated with input variable x_n set to 0 and 1:

$$F(\hat{X},0) = a^{f_a(\hat{X})}g^0a^{f_b\hat{X})}g^0 \text{ and } F(\hat{X},1) = a^{f_a(\hat{X})}g^1a^{f_b(\hat{X})}g^1$$

From Section 2.2 we know that $a^2 = a^{-1}$, $g^0 = I$, $gag = a^{-1}$ and $ga^{-1}g = a$, so $F(\hat{X},0)$ and $F(\hat{X},1)$ can be reduced to the following:

For low input $F(\hat{X},0) = a^{f_a(\hat{X})}a^{f_b(\hat{X})}$ and for high input $F(\hat{X},1) = a^{f_a(\hat{X})}a^{-f_b(\hat{X})}$

By adding the exponents we have the expressions below:

$$F(\hat{X},0) = a^{f_a(\hat{X})+f_b(\hat{X})}$$

$$F(\hat{X},1) = a^{f_a(\hat{X})-f_b(\hat{X})}$$

For convenience, we can express the above equations in terms of their powers only, in modulo 3, as follows:

$$f(\hat{X},0) = f_a(\hat{X}) + f_b(\hat{X})$$

$$f(\hat{X},1) = f_a(\hat{X}) - f_b(\hat{X})$$

Converting the expressions above in matrix form, we have the following:

$$\begin{bmatrix} f(\hat{X},0) \\ f(\hat{X},1) \end{bmatrix} = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} * \begin{bmatrix} f_a(\hat{X}) \\ f_b(\hat{X}) \end{bmatrix} \text{ or simply } \vec{F} = W_1 * \vec{w} \qquad (2.2)$$

Note that $\begin{bmatrix} f(\hat{X},0) \\ f(\hat{X},1) \end{bmatrix}$ is the result of the logical function and can consequently be identified as the truth vector, $\vec{F} = \begin{bmatrix} f(\hat{X},0) \\ f(\hat{X},1) \end{bmatrix}$. In addition note that matrix

$\begin{bmatrix}+1 & +1\\ +1 & -1\end{bmatrix}$ is a Walsh matrix for one variable, $W_1 = \begin{bmatrix}+1 & +1\\ +1 & -1\end{bmatrix}$. Lastly note that $f_a(\hat{X})$ and $f_b(\hat{X})$ represent the exponents of the a elements of the S_3 group and literally describe the canonical form of the circuit cascade, $\vec{w} = \begin{bmatrix}f_a(\hat{X})\\ f_b(\hat{X})\end{bmatrix} = \begin{bmatrix}w_a\\ w_b\end{bmatrix} = [w_a, w_b]^T$. So if the canonical cascade can be found by simply solving (2.2) for $\vec{w}$ then $\vec{w}$ is the Walsh spectrum of $\vec{F}$. If both sides of (2.2) are multiplied by the inverse of W_1 then the Wash spectrum of $\vec{F}$ becomes the following:

$$\vec{w} = ({W_1}^{-1})\vec{F}, \qquad (2.3)$$

The inverse of the Walsh matrix must be found without using its determinant, as the determinant of a Walsh matrix is always zero. However when the Walsh matrix is multiplied by its inverse, we will obtain the identity matrix. This notion will be used to find the inverse of the Walsh matrix:

$$W_1 * {W_1}^{-1} = I_1 = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

To find ${W_1}^{-1}$ for a single variable, using the elements of S_3, we can use simple math, but in modulo 3 terms. The steps are shown below:

$$\begin{bmatrix}+1 & +1\\ +1 & -1\end{bmatrix} * \begin{bmatrix}+1 & +1\\ +1 & -1\end{bmatrix} = \begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix} = 2\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = 2I_1 \equiv -I_1 \ (mod\ 3)$$

$$W_1 * W_1 \equiv -I_1 \ (mod\ 3)$$

$$W_1 * W_1 \equiv -\left(W_1 * {W_1}^{-1}\right) (mod\ 3)$$

$${W_1}^{-1} \equiv -W_1 \ (mod\ 3)$$

Before group decomposition is extended to multiple variables, the following simple one variable example will demonstrate the derivation of a circuit.

Example 2.1

Consider the function $f(x_1) = \overline{x_1}$, a simple NOT function. The truth vector for this function is $\vec{F} = \begin{bmatrix}1\\0\end{bmatrix}$, the inverse Walsh matrix for one variable is ${W_1}^{-1} = -\begin{bmatrix}+1 & +1\\+1 & -1\end{bmatrix}$, so we need to find $\vec{w}$. We can use equation (2.3):

$$\vec{w} = ({W_1}^{-1})\vec{F} = -\begin{bmatrix}+1 & +1\\+1 & -1\end{bmatrix} * \begin{bmatrix}1\\0\end{bmatrix} \equiv -\begin{bmatrix}1\\1\end{bmatrix} \ (mod\ 3)$$

$$\vec{w} = [w_a, w_b]^T \equiv [-1, -1]^T \ (mod\ 3)$$

The group decomposition expression (2.1) can be rewritten in canonical form as follows:

$$a^{f(x_1)} = a^{w_a} g^{x_1} a^{w_b} g^{x_1}$$

After replacing the exponents for a^{w_a} and a^{w_b} with the values obtained for $\vec{w}$, we have the following:

$$a^{f(x_1)} = a^{-1} g^{x_1} a^{-1} g^{x_1}$$

The corresponding circuit diagram is shown in Figure 2-12:

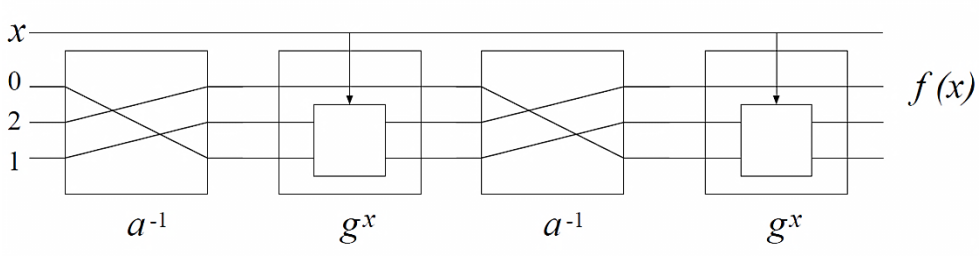


Figure 2-12. Canonical Cascade for $f(x_1) = \overline{x_1}$

(End of Example)

In Figure 2-12 x is extended to the output so that the circuit remains reversible, but the line does not carry the function value. This type of line is called a control line. The bottom lines are the target lines. Only the top target line is used for the function output. Note that the last g is not used, and can be removed, since the function output line will always be the top target line. The reversible circuit reduces to the circuit depicted in Figure 2-13:

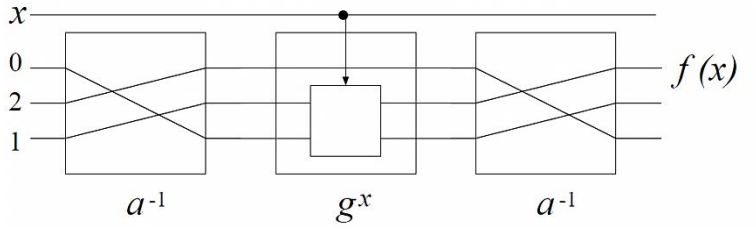


Figure 2-13. Reduced Canonical Cascade for $f(x_1) = \overline{x_1}$

2.3.2 Group decomposition applied to two input variable canonical cascade

In the previous section we saw an example that utilized the Walsh spectrum and group decomposition to realize a one input variable controlled NOT function. In this section the group decomposition and Walsh spectrum are extended to a two input variable function.

Expression (2.1), can be extended to two variables by:

$$F(x_1, x_2) = F_a(x_2)g^{x_1}F_b(x_2)g^{x_1}$$

but in this case F_a and F_b are group functions of one variable. Figure 2-14 shows the circuit diagram.

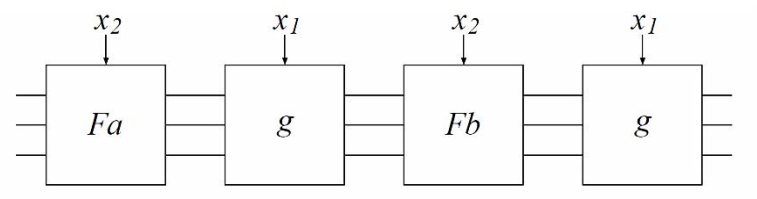


Figure 2-14. Cascade for all two variable functions

After decomposition, the canonical form for all functions with two input variables becomes:

$$a^{f(x_1,x_2)} = ((a^{w_a}g^{x_2}a^{w_b}g^{x_2})g^{x_1})((a^{w_c}g^{x_2}a^{w_d}g^{x_2})g^{x_1})$$

$$a^{f(x_1,x_2)} = a^{w_a}g^{x_2}a^{w_b}g^{x_1+x_2}a^{w_c}g^{x_2}a^{w_d}g^{x_1+x_2}$$

The circuit has 10 cells and is shown in Figure 2-15:

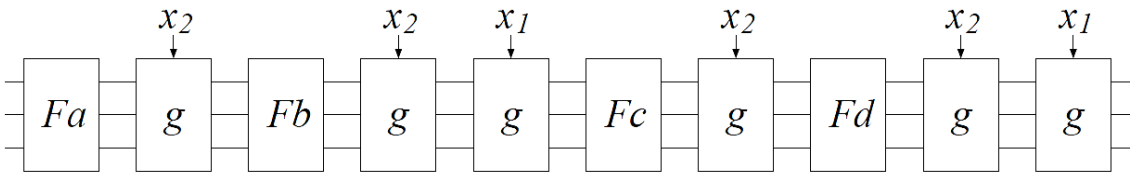


Figure 2-15. Canonical Cascade for all two variable functions after decomposition

Next we will expand the Walsh matrix to two variables by assigning 0 and 1 to x_1 and x_2 for all four possible combinations and apply vectors to the exponents as was done in section 2.3.1 for the single variable case. After applying the same property, of $gag = a^{-1}$, from Section 2.3.1, the following four equations are derived:

$$f(0,0) = w_a + w_b + w_c + w_d$$

$$f(0,1) = w_a - w_b + w_c - w_d$$

$$f(1,0) = w_a + w_b - w_c - w_d$$

$$f(1,1) = w_a - w_b - w_c + w_d$$

When above polarities are put in matrix form, we will notice that a Walsh ordered matrix is formed:

$$W_2 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

Lastly, we need to find the inverse of W_2, using the same process as described in the previous section and shown again below, but this time it is for two variables:

$$W_2 * {W_2}^{-1} = I_1 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

$$\begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix} * \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} = 4I_2$$

$$\equiv I_2 \ (mod\ 3)$$

$$W_2 * W_2 \equiv I_2 \ (mod\ 3)$$

$$W_2 * W_2 \equiv W_2 * {W_2}^{-1} \ (mod\ 3)$$

$$ {W_2}^{-1} \equiv W_2 \ (mod\ 3)$$

Note that W_2 is its own inverse. A generalization for finding the inverse Walsh matrix will be made at the end of the chapter.

Now that we have the two variable canonical cascade functions and the inverse two variable Walsh matrix, we can apply them to a two variable example.

Example 2.2

Consider the function $f(x_{1,}\, x_2) = x_1 \oplus x_2$, a simple XOR function. The truth vector for this function is $\vec{F} = [0, 1, 1, 0]^T$, the inverse Walsh matrix for two variables is ${W_2}^{-1} \equiv W_2\ (mod\ 3)$, so again we need to find $\vec{w}$. Once again we use equation (2.3):

$$\vec{w} = ({W_2}^{-1})\vec{F} = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ -2 \end{bmatrix} \equiv \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix} (mod\ 3)$$

$$\vec{w} = [w_a, w_b, w_c, w_d]^T \equiv [-1, 0, 0, 1]^T\ (mod\ 3)$$

The group decomposition expression for two variables in canonical form is:

$$a^{f(x_1,x_2)} = a^{w_a} g^{x_2} a^{w_b} g^{x_1+x_2} a^{w_c} g^{x_2} a^{w_d} g^{x_1+x_2}$$

After replacing the exponents for a^{w_a}, a^{w_b}, a^{w_c} and a^{w_d} with the values obtained for $\vec{w}$, and the same reductions as in Section 2.3.1 for the one variable case, we have the following canonical cascade function:

$$a^{f(x_1,x_2)} = a^{-1} g^{x_2} a^0 g^{x_1} g^{x_2} a^0 g^{x_2} a^1 g^{x_1+x_2} = a^{-1} g^{x_1+x_2} a^1 g^{x_1+x_2}$$

Figure 2-16 A. and B. show the resultant canonical cascade circuit diagram. Note that the last set of g elements are not used and can be removed (a reduction generalization will be derived at the end of the chapter):

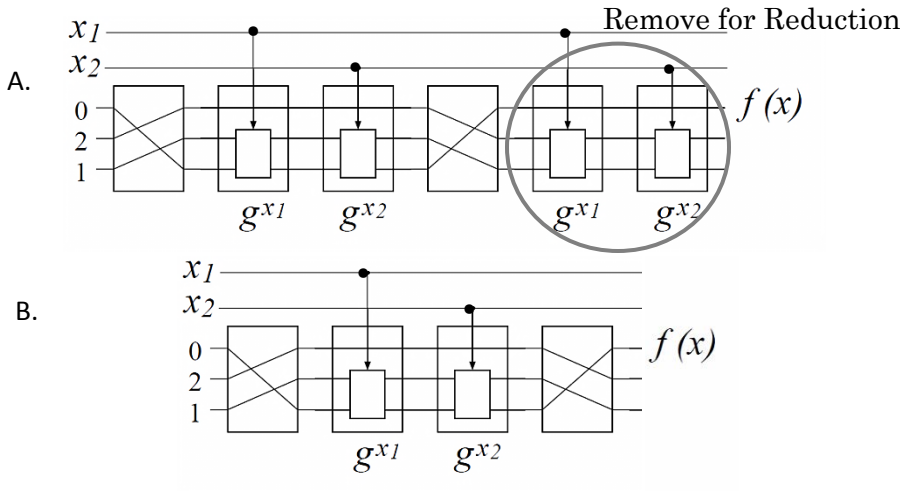


Figure 2-16. Cascade for $f(x_{1,}\, x_2) = x_1 \oplus x_2$, A. before and B. after reduction

(End of Example)

2.3.3 Group decomposition applied to three input variable canonical cascade

In the previous two sections we saw examples for one and two variable canonical cascades, which utilized only two of the three output levels available. To establish a solid pattern, and the utilization of all three output logic levels, an extension to three variables is in order. In this section the group decomposition and Walsh spectrum are extended to a three input variable function and we utilize the same process as in the previous two sections (some steps will be omitted).

Expression (2.1), can be extended to three variables, $F(x_1, x_{2,} x_3)$ and after decomposition, the following 22 cell canonical expression is derived:

$$a^{f(x_1,x_2,x_3)} = \{[((a^{w_a} g^{x_3} a^{w_b} g^{x_3}) g^{x_2})((a^{w_c} g^{x_3} a^{w_d} g^{x_3}) g^{x_2})] g^{x_1}\}$$

$$\{[((a^{w_e} g^{x_3} a^{w_f} g^{x_3}) g^{x_2})((a^{w_g} g^{x_3} a^{w_h} g^{x_3}) g^{x_2})] g^{x_1}\}$$

$$a^{f(x_1,x_2,x_3)}$$

$$= a^{w_a} g^{x_3} a^{w_b} g^{x_2+x_3} a^{w_c} g^{x_3} a^{w_d} g^{x_1+x_2+x_3} a^{w_e} g^{x_3} a^{w_f} g^{x_2+x_3} a^{w_g} g^{x_3} a^{w_h} g^{x_1+x_2+x_3}$$

Since there are three variables and eight combinations, eight equations are generated:

$$f(0,0,0) = w_a + w_b + w_c + w_d + w_e + w_f + w_g + w_h$$

$$f(0,0,1) = w_a - w_b + w_c - w_d + w_e - w_f + w_g - w_h$$

$$f(0,1,0) = w_a + w_b - w_c - w_d + w_e + w_f - w_g - w_h$$

$$f(0,1,1) = w_a - w_b - w_c + w_d + w_e - w_f - w_g + w_h$$

$$f(1,0,0) = w_a + w_b + w_c + w_d - w_e - w_f - w_g - w_h$$

$$f(1,0,1) = w_a - w_b + w_c - w_d - w_e + w_f - w_g + w_h$$

$$f(1,1,0) = w_a + w_b - w_c - w_d - w_e - w_f + w_g + w_h$$

$$f(1,1,1) = w_a - w_b - w_c + w_d - w_e + w_f + w_g - w_h$$

The matrix form is once again a Walsh matrix:

$$W_3 = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix}$$

The inverse Walsh matrix is the following:

$$W_3{}^2 = 8I_3 \equiv -I_3 \ (mod\ 3)$$

$$W_3{}^{-1} \equiv -W_3 \ (mod\ 3)$$

Example 2.3

Consider the function $f(x_1, x_2, x_3) = x_3 + x_2 + x_1$, a modulo 3 adder function. Truth Table 2-1 provides the truth vector:

x_3	x_2	x_1	$\vec{F}$ for $f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	1
1	0	1	2
1	1	0	2
1	1	1	1

Table 2-1. Modulo 3 Adder Truth Table

The truth vector for this function is $\vec{F} = [0,1,1,2,1,2,2,1]^T$, the inverse Walsh matrix is ${W_3}^{-1} \equiv -W_3\ (mod\ 3)$, so again we need to find $\vec{w}$, by using equation (2.3):

$$\vec{w} = -\begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \end{bmatrix} = -\begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{w} \equiv [0,1,1,0,1,0,0,0]^T\ (mod\ 3)$$

The resultant canonical cascade expression and diagram, after reduction and the removal of the last unused g element, are shown below and in Figure 2-17:

$$f(x_1, x_2, x_3) = g^{x_3} a^1 g^{x_2+x_3} a^1 g^{x_1+x_2} a^1$$

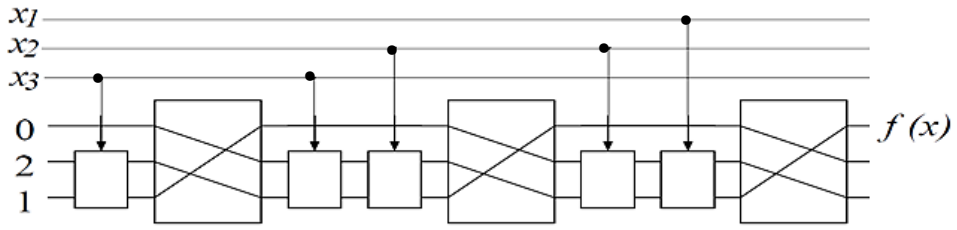


Figure 2-17. Canonical Cascade after reduction for the modulo three adder of binary arguments, $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$

(End of Example)

2.3.4 Group decomposition applied to p valued canonical cascades

So far we have seen canonical cascades of one, two, and three binary input variables that utilize a tri-logic output state system (three rail cascade circuits). The group decomposition method proposed in this chapter cannot use binary output logic or two rails, even though two is a prime number, because the number of group elements will introduce symmetry as mention at the beginning of this chapter. Two rail cascades are still possible as shown in [23], but require the Klein 4-group and alternating group of four elements. The element modifications are far less intuitive than the methods proposed in this thesis. Therefore we will limit the number of logic levels to prime numbers greater than two. Depending of the function desired, a significant reduction in the cascades can be achieved by simply using a higher order of prime logic levels. This is very similar to the bus width reduction that occurs when logic levels are increased (8-bit wide binary bus can be reduced to four lines wide, if each line had four states). For the aforementioned reason we will extend group decomposition to binary input, higher p valued canonical cascades.

To start with we modify the cyclical group to an order of p, so that $C_p = \{I, a, a^2, a^3 \dots, a^{p-1}\}$, and maps to a p-valued function $f: \{0,1\}^n \rightarrow \{0,1, \dots, p-1\}$, where the a element permutes the i rail into the $i+1$ rail. The second group remains with still only two elements, $C_2 = \{I, g\}$, however the g

element is modified to swap the i rail with the $p-1$ rail, when $i \neq 0$. The cell modifications for a and g are shown in Figure 2-18 for a 5-valued cascade:

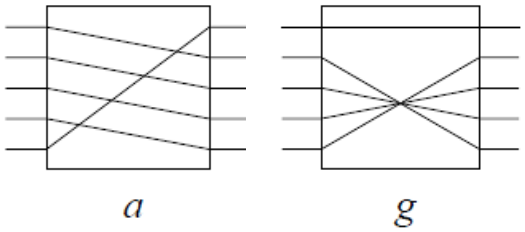


Figure 2-18. Modified a and g cells for 5-valued cascades

The two groups are then combined to form a dihedral group of order 10, degree 5 ($C_5 * C_2 = D_5$, with 10 total elements). The D_5 group map is shown in Figure 2-19 (note that $gag = a^{-1}$ is still true):

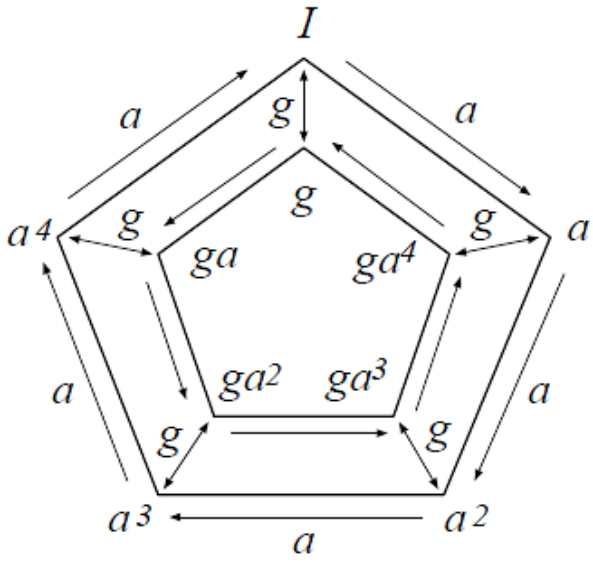


Figure 2-19. D_5 Group Map

From the previous three examples we showed that evaluating the group functions for all possible input vectors always forms a Walsh matrix. This means that the Walsh matrix is dependent only on the number of input variables and remains unaltered by the change of the logic levels. So for an extension to p-valued logic, no modification is required to the Walsh matrix.

The next example revisits example 2.1, where $f(x_1) = \overline{x_1}$. Recall again that the function will only utilize the lowest two logic levels.

Example 2.4

The canonical cascade for $f(x_1) = \overline{x_1}$ implementation, using 5-valued, D_5 group decomposition is demonstrated. The truth vector is $\vec{F} = [1, 0]^T$ and the inverse Walsh matrix in modulo 5 terms can be found as follows:

$$W_1 * {W_1}^{-1} = I_1$$

$$\begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}\begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = 2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \equiv 2I_1 \ (mod\ 5)$$

$$W_1 * W_1 \equiv 2W_1 * {W_1}^{-1} \ (mod\ 5)$$

$${W_1}^{-1} \equiv \frac{1}{2} W_1 \ (mod\ 5)$$

Equation (2.3) is used again to find $\vec{w}$:

$$\vec{w} = (W_1^{-1}) = \begin{bmatrix} w_a \\ w_b \end{bmatrix} = \begin{bmatrix} +1/2 & +1/2 \\ +1/2 & -1/2 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} (mod\ 5)$$

$$(a^{1/2})(a^{1/2}) = a,\ a^6 = (a^3)(a^3) \text{ so } a \equiv a^6 \therefore\ a^{1/2} \equiv a^3\ (mod\ 5)$$

$$f(x_1) = a^3 g^{x_1} a^3 g^{x_1}$$

Once again the last g is not used and is removed:

$$f(x_1) = a^3 g^{x_1} a^3$$

The resulting canonical cascade circuit diagram is shown in Figure 2-20:

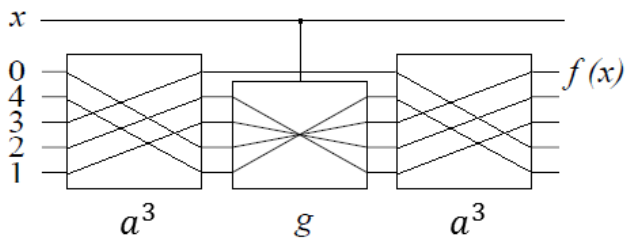


Figure 2-20. Reduced Canonical Cascade for $f(x_1) = \overline{x_1}$

(End of Example)

Example 2.5

The last example is a two-bit, four input variable, modulo 7 adder that shows how all logic values can be used in a seven-valued cascade using D_7 group decomposition. The adder adds two binary numbers, each two bit wide, to form a single modulo 7 output value (7-valued output), $f = 2(x_{22} + x_{21}) +$

($x_{12} + x_{11}$). The x_{22} and the x_{21} binary bits are multiplied by two to form the appropriate binary weight. Table 2-2 shows the truth table for the modulo 7 adder function.

x_{22}	x_{12}	x_{21}	x_{11}	$\vec{F}$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	1
0	1	0	1	2
0	1	1	0	3
0	1	1	1	4
1	0	0	0	2
1	0	0	1	3
1	0	1	0	4
1	0	1	1	5
1	1	0	0	3
1	1	0	1	4
1	1	1	0	5
1	1	1	1	6

Table 2-2. Modulo 7 Adder Truth Table

The truth vector is $\vec{F} = [0,1,2,3,1,2,3,4,2,3,4,5,3,4,5,6]^T$

To find the inverse Walsh matrix, we can square it and factor out the scalar value to produce an identity matrix: ${W_4}^2 = 2^4 I_4$ and the scalar is $2^4 \equiv 2\ (mod\ 7)$, therefore the inverse Walsh matrix is ${W_4}^{-1} \equiv \frac{1}{2} W_4\ (mod\ 7)$

After multiplying $\frac{1}{2} W_4$ with $\vec{F}$,

$$\vec{w} \equiv [3,3,-1,0,3,0,0,0,-1,0,0,0,0,0,0,0]^T\ (mod\ 7)$$

After simplification the canonical cascade is the following:

$$f = a^3 g^{x_{11}} a^3 g^{x_{11}} g^{x_{21}} a^{-1} g^{x_{21}} g^{x_{12}} a^3 g^{x_{21}} g^{x_{22}} a^{-1}$$

The canonical cascade circuit diagram is shown in Figure 2-21:

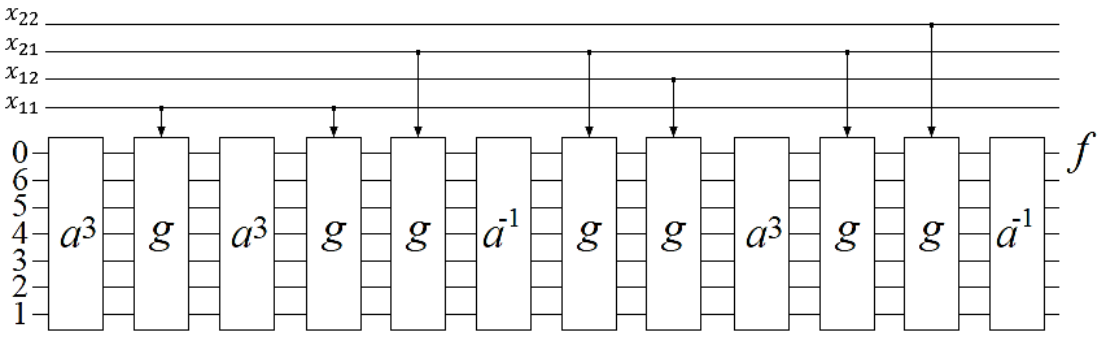


Figure 2-21. Reduced Canonical Cascade for two bit modulo 7 adder, $f = 2(x_{22} + x_{21}) + (x_{12} + x_{11})$

(End of Example)

2.4 Summary and Conclusions

In this chapter we introduced group theory and generated respective group elements. We modified the g element from the C_2 group to form a binary controlled element that swaps an even number of rails, and resembles the Fredkin reversible gate. We then combined C_2 and C_3 to form the S_3 group and introduced group decomposition to generate canonical cascades. The relevant examples extended the group decomposition method from one input variable function with three logic values, to a four input variable adder

with seven logic values. As the examples progressed the following theorems began to emerge:

1. A group function $F(x): B^n \rightarrow C_p$ decomposes to $(\hat{X}, x_n) = F_a(\hat{X}) g^{x_n} F_b(\hat{X}) g^{x_n}$, where x_n is a two-valued input variable, and $F_a(\hat{X})$ and $F_b(\hat{X})$ are group functions $(\hat{X} = \left(x_{1,\ldots,} x_{m-1}\right) \in X^{m-1})$ that do not depend on x_n.
2. The function truth vector $\vec{F}$ for $f: \{0,1\}^n \rightarrow \{0,1,\ldots,p-1\}$, can be realized as a cascade of elements of C_p and g. The coefficients from the canonical expression of the cascade are the same as in the Walsh spectrum, $\vec{w}$, of $\vec{F}$ and forms the expression $\vec{w} \equiv 2^{-n} W_n \vec{F}\ (mod\ p)$.
3. Group decomposition generates $N(n) = 3(2^n) - 2$ cells. All zero coefficients along with each adjacent g element, that is controlled by the same input can be removed (except in the first occurrence) via $g^n a^0 g^n = I$. In addition the g elements at the end of the canonical cascade are not used and can be removed (on average there are n such g elements).

The steps involved in generating a canonical case are algorithmic as follows:

1. Obtain the truth vector : $\vec{F}$
2. Multiply inverse Walsh matrix by the truth vector to obtain the Walsh spectrum, which is the canonical cascade: $\vec{w} = 2^{-n} W_n \vec{F}\ \ (mod\ p)$

3. Remove the 0 coefficients and the affected adjacent cells via relation $g^n a^0 g^n = I$ (except for the first 0 coefficient) and remove any unused g cells

As we can see, by the aforementioned steps, the process of generating canonical cascades is very simple. More importantly the canonical cascades have a one-to-one correspondence between the input and output maps and naturally form reversible logic. From Chapter 1, we saw that reversible logic can potentially save huge amounts of energy, and therefore efficient designs of cascades is very important [24]. In addition due to the sequential nature of quantum circuits, cascades become a natural method for describing and synthesizing quantum circuits as well. In the next two chapters we will explore the application group decomposition to quantum circuits with very different quantum mechanics.

Chapter 3

APPLICATION OF GROUP DECOMPOSITION TO QUANTUM TECHNOLOGIES

In this chapter, the physics of Chapter 1 and the mathematics of Chapter 2 will be combined, to form a single seamless methodology for synthesizing quantum circuits. The synthesis of quantum circuits has been long pursued in discovering fast and efficient methods for realizing quantum circuits that have minimum cost, in terms of quantum gates and in terms of minimal so called quantum cost as (Maslov quantum cost) [25]. There are always tradeoffs between speed accuracy and cost. In this chapter we will show how the Walsh spectrum and group decomposition can be utilized to produce low cost and fast quantum cascades. Methods for improving accuracy are also presented, however, the compromise between cost, speed, and accuracy is application specific.

3.1 Qubit Probabilistic Model

As pointed out in Chapter 1 rotations about the X or Y axis occur by changing electron orbitals from one energy state to another, as in the ion trap qubit manipulation, or by altering bulk spin electron orientation, as in the quantum dot qubit manipulation. In both scenarios the qubit manipulation

is quantized to the smallest energy quanta that can be detected by the measurement apparatus. In the ion trap, individual photons are counted [26], [27], while in quantum dots individual charge proximities are approximated. Discrete statistical behaviors are modeled using Poisson distribution models and can also be used to predict the states of qubits. Any superposition a qubit exists in can be statistically inferred by repeatedly executing the quantum operation and performing a measurement, or by parallel processing and measurement. When a qubit is in perfect superposition, the probability that a measurement will yield $|0\rangle$ or $|1\rangle$ is exactly 50% and the Eigen vector Ψ will be oriented along a plane midway between the $|0\rangle$ or $|1\rangle$ states. When the Eigen vector approaches the $|0\rangle$ or $|1\rangle$ locations, the probability of measuring each approaches, but does not reach 100%. Figure 3-1 shows various orientations of Eigen vectors and the corresponding distribution each will yield. It is interesting and very important to note that a rotation about the Z axis does not change the probability distributions, however, as pointed out in Chapter 1 such rotations are physically possible and do affect the location of the Eigen vector. Traversing this higher order space, provides an additional degree of freedom in Hilbert space, and thus allows for the ability to utilize elegant mathematics for describing quantum gates.

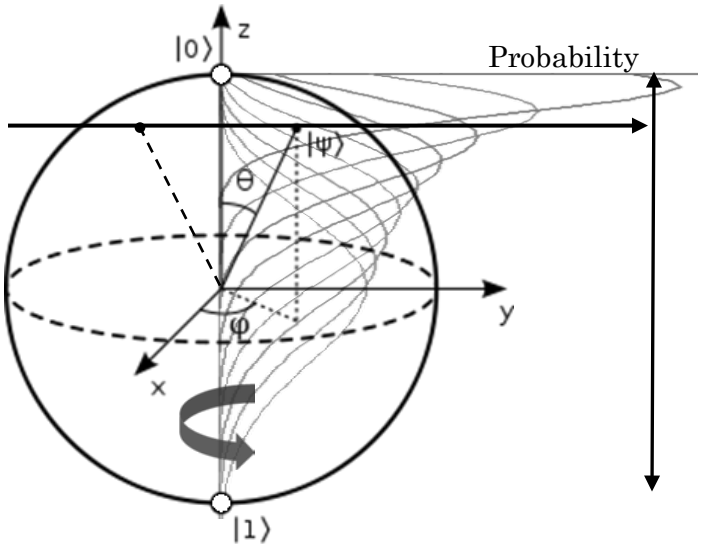


Figure 3-1. Poisson Distributions super imposed on a Bock Sphere

3.2 Qubit State Measurement

In Chapter 1, it was discussed that special hardware is needed in order to measure the state a qubit is occupying. In addition making a qubit measurement is destructive, meaning that executing a measurement will permanently alter the state of a qubit. Careful consideration must be used in determining when a measurement should be made, not only because a measurement changes the state of a qubit system, but also terminates the execution of a quantum gate and or algorithm. In order to mathematically show the probabilistic nature of quantum bit measurements, the following expressions are introduced:

$$P_1(m) = \langle \Psi | M_1 | \Psi \rangle \text{ and } P_0(m) = \langle \Psi | M_0 | \Psi \rangle$$

Where $P_n(m)$ is the probability of a qubit occupying a given observable state, m is the original state of the qubit, $|\Psi\rangle$ is the Eigen state vector $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ ($\alpha = cos(\theta/2)$ and $\beta = e^{i\phi}sin(\theta/2)$), and M_1 and M_0 are the measurement matrices for measuring states $|1\rangle$ or $|0\rangle$. It is important to note that the completeness probability formula must be satisfied:

$$1 = \Sigma_m P(m) = \Sigma_m \langle\Psi|M_n|\Psi\rangle$$

That is $M_1 + M_0$ must equal I and if $M_1 = |1\rangle\langle 1|$ and $M_0 = |0\rangle\langle 0|$, then $M_1 = \begin{bmatrix}0\\1\end{bmatrix} \otimes [0,1] = \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$ and $M_0 = \begin{bmatrix}1\\0\end{bmatrix} \otimes [1,0] = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$. Now note that $\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix} + \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix} = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} = I$.

The following simplifications can now be made as long as the Eigen state vector is in the form of $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ († denotes complex conjugate):

$$P_0(m) = \langle\Psi|M_0|\Psi\rangle = [\alpha,\beta]^\dagger \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}\alpha\\\beta\end{bmatrix} = |\alpha|^2$$

$$P_1(m) = \langle\Psi|M_1|\Psi\rangle = [\alpha,\beta]^\dagger \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}\alpha\\\beta\end{bmatrix} = |\beta|^2$$

For multiple vector states, the following shows the sum of all of the states and the probability of each:

$$|\Psi\rangle = \varphi_0|00\rangle + \varphi_1|01\rangle + \varphi_2|10\rangle + \varphi_3|11\rangle$$

Multiple qubits with n states and the respective measurement probability:

$$|\Psi\rangle = \varphi_0|m_0\rangle + \varphi_1|m_1\rangle + \cdots + \varphi_{n-1}|m_{n-1}\rangle$$

$$P_n(m) = \langle\Psi|M_n|\Psi\rangle = [\varphi_0, \varphi_1, \ldots \varphi_{n-1}]^\dagger = |\varphi_{n-1}|^2$$

Example 3.1

In this example a measurement for $|1\rangle$ and $|0\rangle$ will be performed immediately after the execution of a Y rotation on a qubit that is initialized to a state of $|1\rangle$ or $|0\rangle$ (the quantum circuits are shown in Figure 3-2):

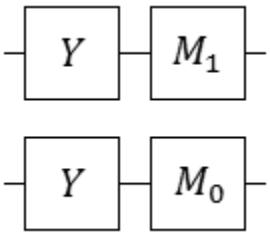


Figure 3-2. Measurement gates M_1 and M_0 after Y rotation gates

When a Y rotation is applied to an initial state of $|1\rangle$, the output state becomes $|-i\rangle$ and the phase information of $-i$ is preserved.

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -i \\ 0 \end{bmatrix}$$

When a Y rotation is applied to state $|0\rangle$, the output state becomes $|i\rangle$ and the phase information of i is preserved.

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix}$$

When the measurement operator M_1 is applied to a Y rotation matrix, the resultant matrix loses the phase rotation information of i or $-i$ respectively (see below). This confirms that the measurement operators are destructive and non-reversible.

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} * \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -i \\ 0 & 0 \end{bmatrix}$$ Measurement operator for measuring a $|1\rangle$, removes i

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ i & 0 \end{bmatrix}$$ Measurement operator for measuring a $|0\rangle$, removes $-i$

If the input state of the qubit is $|1\rangle$, after a Y rotation, the state of the qubit is then changed to $|\Psi\rangle = -i|0\rangle + 0|1\rangle$, $P_1(1) = |b|^2 = 0$ and $P_0(1) = |a|^2 = 1$. So the probability of measuring a $|1\rangle$ at the output is 0%, or 100% for measuring a $|0\rangle$.

$$P_1(1) = \langle\Psi|M_1|\Psi\rangle = [\alpha, \beta]^\dagger \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\beta|^2$$

$$[i, 0] * \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} -i \\ 0 \end{bmatrix} = 0$$

$$P_0(1) = \langle\Psi|M_0|\Psi\rangle = [\alpha, \beta]^\dagger \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\alpha|^2$$

$$[i, 0] * \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} -i \\ 0 \end{bmatrix} = 1$$

If the input state of the qubit is $|0\rangle$, after a Y rotation, the state of the qubit is changed to then $|\Psi\rangle = 0|0\rangle + i|1\rangle$, $P_1(0) = |\beta|^2 = 1$ and $P_0(0) = |\alpha|^2 = 0$. So the probability of measuring a $|1\rangle$ at the output is 100%, 0% for measuring a $|0\rangle$.

$$P_1(0) = \langle\Psi|M_1|\Psi\rangle = [\alpha, \beta]^\dagger \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\beta|^2$$

$$[0, -i] * \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 \\ i \end{bmatrix} = 1$$

$$P_0(0) = \langle\Psi|M_0|\Psi\rangle = [\alpha, \beta]^\dagger \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\alpha|^2$$

$$[0, -i] * \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 \\ i \end{bmatrix} = 0$$

(End of Example)

Example 3.2

In this example a measurement for $|1\rangle$ and $|0\rangle$ will be made immediately after the execution of a Hadamard gate on a qubit that is initialized to a state of $|1\rangle$ (the quantum circuits are shown Figure 3-3):

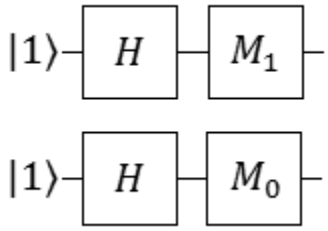


Figure 3-3. Measurement gates M_1 and M_0 after Hadamard gates

After the execution of the Hadamard gate on a qubit initialized to $|1\rangle$, the state becomes the following:

$$H|\Psi_1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} * \begin{bmatrix}0\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix}$$

After the measurement of M_1, the state becomes the following:

$$\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}0\\ -1\end{bmatrix}$$

The probability of measuring $|1\rangle$ becomes the following:

$$P_1(1) = \frac{1}{\sqrt{2}}[1,1] * \frac{1}{\sqrt{2}}\begin{bmatrix}0\\ -1\end{bmatrix} = {}^1\!/_2$$

Similarly, the process is the same for measuring $|0\rangle$:

$$\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 0\end{bmatrix}$$

$$P_0(1) = \frac{1}{\sqrt{2}}[1,-1] * \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 0\end{bmatrix} = {}^1\!/_2$$

(End of Example)

Note that the outcome is exactly the same for $P_1(1)$ and $P_0(1)$. And as per the definition of a Hadamard rotation, the outcomes for $P_1(0)$ and $P_0(0)$, would also be the same. The next example proves the aforementioned observation.

Example 3.3

In this example a measurement for $|1\rangle$ and $|0\rangle$ will be made immediately after the execution of a Hadamard gate on a qubit that is initialized to a state of $|0\rangle$ (the quantum circuits are shown in Figure 3-4):

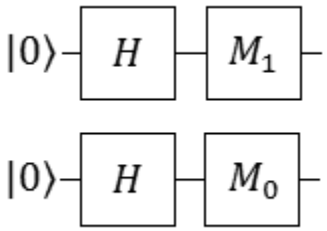


Figure 3-4. Measurement gates M_1 and M_0 after Hadamard gates

After the execution of the Hadamard gate on a qubit initialized to $|0\rangle$, the state becomes the following:

$$H|\Psi_0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} * \begin{bmatrix}1\\ 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}$$

After the measurement of M_1, the state becomes the following:

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} * \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The probability of measuring $|1\rangle$ becomes the following:

$$P_1(0) = \frac{1}{\sqrt{2}} [1, -1] * \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = {}^1/_2$$

Again, the process is the same for measuring $|0\rangle$:

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} * \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$P_0(0) = \frac{1}{\sqrt{2}} [1, -1] * \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = {}^1/_2$$

(End of Example)

Example 3.4

In this example a measurement for $|11\rangle$ of a joint state of two qubits is made immediately after the execution of a Hadamard rotation on each qubit, each initialized to a state of $|0\rangle$ (the quantum circuit is shown in Figure 3-5):

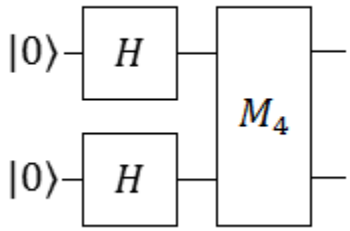


Figure 3-5. Double Measurement gate after Hadamard gates

The Kronecker product represents a Hadamard gate execution on both qubits:

$$H_{12}|\Psi_0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

$$\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix} * \begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}$$

So the probability of measuring a |11⟩ state at the output is .25 as shown below:

$$P_4(m) = \langle\Psi_0|M_{11}|\Psi_0\rangle =$$

$$\begin{bmatrix}M_{00} & 0 & 0 & 0\\ 0 & M_{01} & 0 & 0\\ 0 & 0 & M_{10} & 0\\ 0 & 0 & 0 & M_{11}\end{bmatrix} * \left[\tfrac{1}{2}(M_{00}), -\tfrac{1}{2}(M_{01}), -\tfrac{1}{2}(M_{10}), \tfrac{1}{2}(M_{11})\right]^{\dagger} =$$

$$= (1/2)^2 = \frac{1}{4}$$

(End of Example)

3.3 Multilevel Qubits

Multilevel qubits are called qudits and are existent as an arbitrary point in a Hilbert space with more than two dimensions. There exist as many qudits basis states as can be distinguished by the measurement equipment and the number of calculations made to statistically derive the location of an Eigen vector beyond as a statistical uncertainty. Depending on the quantum computer hardware, the number of discrete qudit states may be limited by the smallest detectable energy quanta. Multilevel qubits can be easily implemented to utilize the group decomposition methodology for describing quantum canonical cascades. No special modification is required to the group decomposition method, so long as multi-level quantum systems are readily available and logic level swap gates have been developed. From a mathematical perspective, multilevel systems and gates exist, however their physical implementation is at an even earlier infancy than the currently proposed binary quantum systems. Though multilevel systems may possess tremendous computing power and my offer simplifications in data structures, an in-depth discussion will be reserved for future work outside of this thesis. For now, in this thesis, multilevel logic will be collapsed and confided to two and three dimensional Hilbert spaces, where the probability of the location of an Eigen vector represents a logical state.

To show how vector space can be mathematically utilized, the bloch sphere is collapsed to a two dimensional space as shown in Figure 3-6, note that a full rotation about the center requires a rotation of 4π (each quadrant is a single π rotation). The 4π ration is accounted for in the rotations matrices presented in Table 1.3, in Chapter 1. A two dimensional Hilbert space has sufficient degrees of freedom to describe the majority of the unitary rotation gates described in Chapter 1. Complex numbers must be used to describe the vector directions within a two dimensional space:

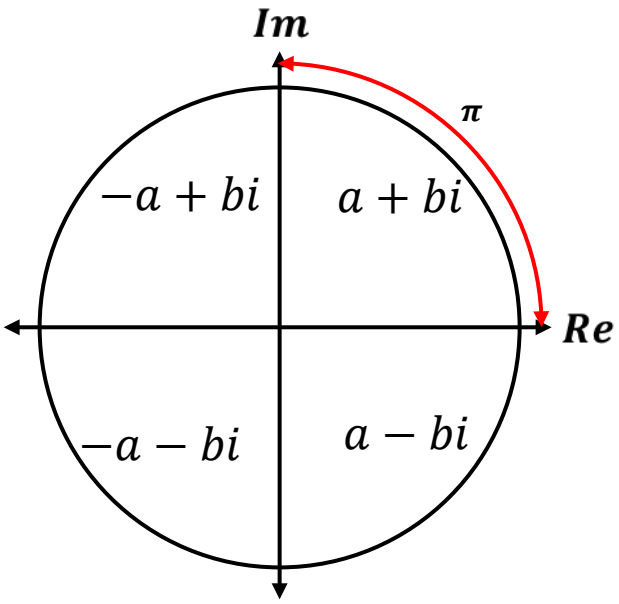


Figure 3-6. Two Dimensional Vector space

The points along the circumference are the number of possible locations of Eigen states. The fewer the number of states present, the more unique and distinguishable each state is and naturally the distance between each

increases. An increased distance allows for larger statistical error tolerance. Figure 3-7 shows the reduction of a continuous vector density to the three prime numbers used in Chapter 2 for group element generation (3, 5, and 7):

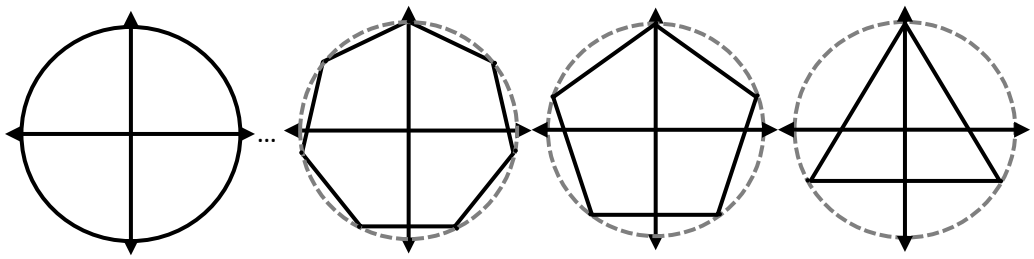


Figure 3-7. Reduction of Continuous Vector space to Discrete Vector locations (each vertex represents a vector location)

In Chapter 2 the methods used to describe multivalued canonical cascades required value systems of only prime numbers, so that only cyclical, non-decomposable groups are formed. That is some symmetries are eliminated so that only unique elements are formed, while other symmetries are preserved so that the unique elements form cyclical groups. Unfortunately when qubits are measured, only the observable is detectable, and therefore the states reduce to simple binary probabilities of $|0\rangle$ or $|1\rangle$, as discussed in Section 3.2 of this chapter. So if three vectors are chosen such that they are equidistant and form a ternary valued vector space, two of the three vectors are statistically indistinguishable, as shown in Figure 3-8:

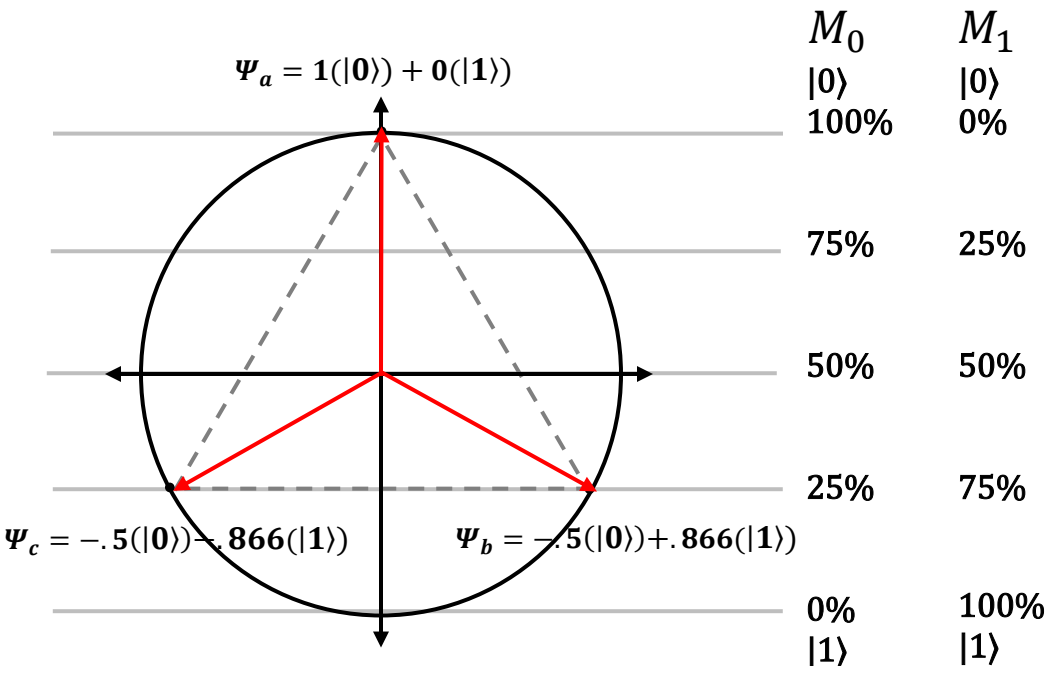


Figure 3-8. $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ Complex Conjugate Vector pairs have the same observable state

Both vectors, Ψ_b and Ψ_c, occupy different space, but both have the same probability of occurrence and therefore the same observable state. Cunning modifications are needed so that the group decomposition methodology can be utilized in the synthesis of quantum circuits. In Section 3.4 such modifications will be proposed, each with its own benefits and deficiencies.

3.4 Cyclic Group modification for Quantum Circuitry

As mentioned earlier, three methods will be proposed that will utilize two level quantum systems for formal quantum circuit synthesis, using the group decomposition method from Chapter 2.

3.4.1 Method One: Swap and Controlled Swap Gates Only

The first method is to simply ignore all larger logical levels and replace them with binary quantum values. This is the equivalent of adding in Ancilla qubits that will not have a meaningful output, but will allow the decomposed group cascade to produce the desired output in terms of states of $|0\rangle$ and $|1\rangle$. Every a^n group element is mapped to a simple series of full swap gates and every g^{x_n} group element becomes a Fredkin gate whose control is tied to a specific input qubit. This method is the most direct and obvious way of modifying the group decomposition method to utilize quantum gates. At a first glance this method may appear to be the most expensive way of synthesizing quantum circuits, but as discussed in Chapter 1, Section 1.5, we realize that it really depends on the technology used. A full swap gate implemented with quantum dots can be generated within a single Hamiltonian time evolution, while the same implementation with an Ion trap can require as many as 11 separate Hamiltonian time evolutions. This means that any a^n element can be implemented with only two full swap gates. However, the cost a Fredkin gate remains expensive regardless of the quantum technology used from Chapter 1. Nonetheless, the synthesis method becomes extremely simple. The following example demonstrates the aforementioned method for the realization of a Toffoli like function, that is,

the output function vector is exactly the same, but the target qubit is not used as one of the input qubits.

Example 3.5

In this example, a Toffoli like function is implemented using the group decomposition methodology, from Chapter 2, with full swap and Fredkin gates. This is not a true Toffoli as there are three input qubits and three Ancilla qubits, which is different than a minimal implantation of a Toffoli gate with only three inputs and three outputs. The Ancilla qubits are initialized so that the zero logic level is replaced with quantum state $|0\rangle$ and all other remaining Ancilla qubits with quantum state $|1\rangle$. The reduced canonical cascaded yielded by group decomposition is the following:

$$f(x) = a^2 g^{x_3} a^2 g^{x_2} a^2 g^{x_2} g^{x_1} a^2 g^{x_2} a^1$$

After Ancilla initialization, the resultant quantum cascade is shown in Figure 3-9:

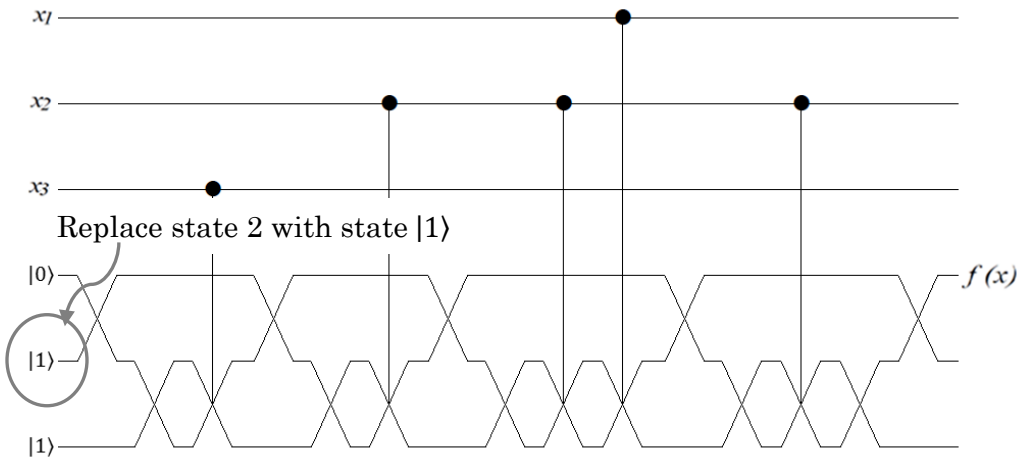


Figure 3-9. Group Decomposed Canonical Cascade with SWAP and Controlled SWAP gates

As can be seen from Figure 3-9 above, the implementation of the Toffoli function is expensive, costing a total of 15 gates (10 full swap gates and 5 Fredkin gates) and 3 Ancilla qubits, of which two are wasted. Nonetheless, the implementation synthesis is straight forward and minimal in terms of further reduction, using only full swap and Fredkin gates. Provided that future technologies permit efficient realizations of the SWAP and Fredkin gates, the methodology can become competitive and therefore is noteworthy.

(End of Example)

3.4.2 Method Two: Probabilistic State Model

The second method is to treat any observable and statistically distinguishable state as a discrete quantum logic level. This means that if the

decomposed groups require three logical states for the generation of classical reversible logic circuit, there will be only two observable quantum states, for the reasons described in Section 3.3 of this chapter. This also means that the number of quantum states p_q is equal to $2p-1$ classical logic states. So for a two level quantum systems, we must use the group decomposition method for three logical levels and if a three level quantum system is desired than a five level group decomposition needs to be utilized. In addition to approximately doubling of the logical states to $p_q = 2p-1$, a modification is needed for the a^n and g^{x_n} group elements.

The most direct method for converting the a^n elements into true quantum gates is to treat each as a primitive rotation about any axis (X or Y), that will directly affect the quantum observable state. The exponent n of the element a^n is converted to a fraction representing the partial rotation, about the bloch sphere that is needed to transform the qubit from one state to another (a full rotation of 360° returns a qubit to its initial state). All a^n elements directly translate to $a^{n/p}$. For instance, if the group decomposition uses three classical logical levels, then a^1 translates to $a^{1/3}$ or $a^{(1/3)*(2\pi)}$, or simply a $2\pi/3$ rotation. An a^2 element translates to $a^{2/3}$, or a $-4\pi/3$ rotation. The a^3 element remains as the identity element and represents a 2π rotation. As a consequence of the conversion of the a^n elements to simple rotation quantum gates, along with the treatment of observable and statistically

distinguishable states, as discrete quantum logical states, eliminates the need for multiple Ancilla qubits. The elimination of multiple Ancilla precludes the utilization of Fredkin gates, so the g^{x_n} element needs to be converted to a controlled quantum gate that operates on a single target qubit.

Recall from Section 2.2.3, from Chapter 2, that the when a cyclic group of order 3 is multiplied with a cyclic group of order 2, a symmetric group of order 6, degree 3 is formed. At each vertex a g^{x_n} element creates a reflection axis that effectively conjugates an a^n element to its group inverse ($gag = a^2$ and $ga^2g = a$). A similar type of conjugation occurs when a qubit is rotated about the Z axis by 180º, to form complex conjugate pairs about the Z axis (Z axis becomes a reflection axis). For instance a vector, along the X plane, at location 120º is $\Psi_x = -.5(|0\rangle) + .866(|1\rangle)$ and becomes $\Psi_x = -.5(|0\rangle) - .866(|1\rangle)$ after a Z rotation of 180º, that is $ZX^{2\pi/3}Z = X^{-2\pi/3}$ and $ZX^{-2\pi/3}Z = X^{2\pi/3}$ (note that Z rotation on a current state has no change in the quantum observable state, just as a g^{x_n} has no effect on the current classical logical state when applied). Figure 3-10 shows a three-state and five-state quantum systems traversing a block sphere:

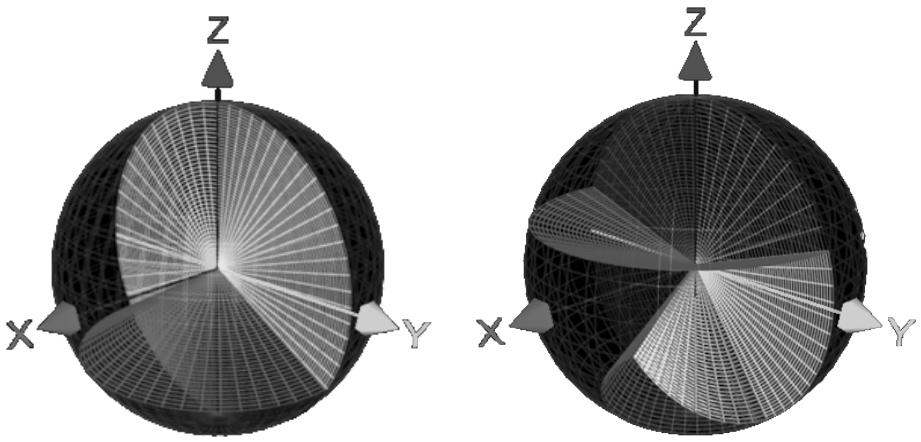


Figure 3-10. 3D Block Spheres showing CZ and X gate rotation paths (two quantum logic levels left, and three quantum logic levels right)

The next example demonstrates a group decomposed cascade, with three classical logic states and how it can be converted to a two-level quantum cascade.

Example 3.6

This example again generates a Toffoli function, using the same cascade from example 3.5. All of the a^n elements are converted to X rotations and every g^{x_n} is converted to a simple controlled Z quantum gate. Once again the reduced canonical cascade before quantum conversion is the following:

$$f(x) = a^2 g^{x_3} a^2 g^{x_2} a^2 g^{x_2} g^{x_1} a^2 g^{x_2} a^1$$

The conversion is accomplished by replacing the a^n element with R_x rotations and g^{x_n} elements with CZ_n gates. The equivalent number of quantum logical

levels is $p_q = (p+1)/2$ and after conversion the quantum equivalent two-level canonical cascade becomes:

$$f(x) = (R_x - 2\pi/3)(CZ_3)(R_x - 2\pi/3)(CZ_2)(R_x - 2\pi/3)(CZ_2)(CZ_1)(R_x - 2\pi/3)(CZ_2)(R_x 2\pi/3)$$

The quantum circuit is shown in Figure 3-11:

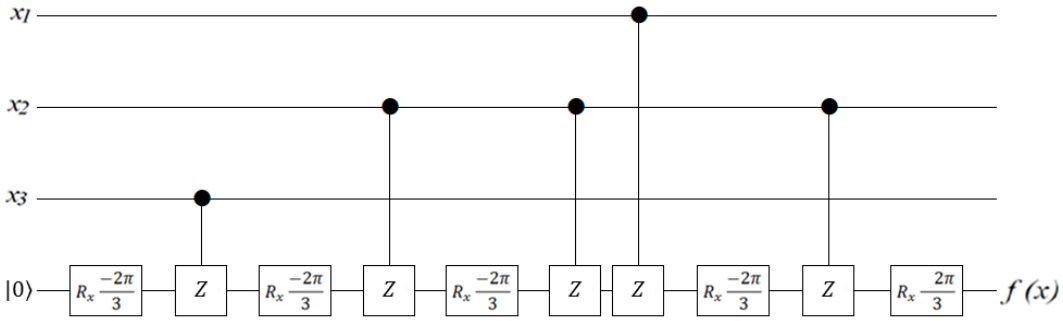


Figure 3-11. Probabilistic Multilevel Quantum Canonical Cascade for a Toffoli equivalent gate

As can be seen from Figure 3-11, the circuit is reduced by the elimination of two Ancilla qubits. The number of gates is also reduced from 15 (see example 3.5) to 10. Also note that the entire circuit is made up entirely of the most primitive quantum gates. The CZ gate is the cheapest controlled gate to implement in both of the technologies presented in Chapter 1. The X rotations are single time evolution Hamiltonians and are the simplest gates to implement in both technologies. The disadvantage of the circuit is that it is probabilistic in nature and its two states are $|0\rangle$ and $(.866)^2|1\rangle$ or 75% of $|1\rangle$.

This means the circuit will need to be executed multiple times before the $|1\rangle$ state can be statistically derived. There are other implications, as the number of observable quantum logic states increases, the statistical difference of adjacent individual states decreases, by $sin^2(\pi/p_q)$, as derived from Tyler's series, where p_q is the number of quantum logic levels. Lastly leaving a qubit outside of its basis states makes it extremely difficult to pass the quantum state from one cascade to another, making this method applicable to very narrow and specific types of quantum circuits. There is a way, however, to remedy this deficiency by extending p_q to large values and create more universal quantum cascades, as will be demonstrated in the next method.

(End of Example)

3.4.3 Method Three: Exact Quantum Binary (EQB)

The third and final method generates quantum canonical cascades with outputs that are purely in one of the basis states of $|0\rangle$ and $|1\rangle$. Recall from Chapter 2 that cyclical groups can have an infinite number of group elements. The repeated permutations of group elements need not reach identity again and if identity cannot be reached, an infinite dihedral group is formed. The advantage of using an infinite dihedral group is that it can still be decomposed and valid canonical cascades can still be derived. The quantum vector can also be rotated to any point on the 2 dimensional block

sphere, including quantum states $|0\rangle$ and $|1\rangle$, as the statistical distinguishability between adjacent states reduces to 0 as shown by $\lim_{p_q \to \infty} sin^2(\pi/p_q)$. This is an undesirable effect for the discrete logic in the previous example, but for exact quantum binary, this has the great advantage of continuity. All states that are between 0 and ∞ are ignored, while the 0 state is mapped to $|0\rangle$, and the ∞ state is mapped to $|1\rangle$. The exponents of the a elements simply take on the values of the inverse Walsh spectrum with no modulo applied, or $-2^{-n} W_n \vec{F}$, where n is the number of input variables, W_n is the Walsh matrix for n variables, and $\vec{F}$ is the function truth vector element. The exponent represents a fraction of a rotation between 0° and 180°, so the exponent can be directly multiplied by π to determine the exact quantum vector rotation. For instance $-2^{-n} W_n \vec{F}$ of a two input Feynman (XOR) function is $-2^{-2} W_2[0,1,1,0]$ and equals $1/4[2,0,0,-2]$, or $[1/2,0,0,-1/2]$, so the exponents for the a elements in cascade take on the actual calculated fractions (the XOR cascade becomes $a^{1/2} g^{x_1} g^{x_2} a^{-1/2}$). The g^{x_n} elements still remain as CZ gates, as explained in the previous example and still provide a reflection at the Z axis.

Example 3.7

This example once again generates a Toffoli function, using the same cascade from example 3.5. All of the a^n elements are converted to X rotations,

whose exponents are $-2^{-n}W_n\,\vec{F}$ with no modulo applied. Every g^{x_n} is converted to a simple controlled Z quantum gate, as before. And once again the reduced canonical cascade before quantum conversion is the following:

$$f(x) = a^{1/2}g^{x_3}a^{-1/4}g^{x_2}a^{-1/4}g^{x_2}g^{x_1}a^{-1/4}g^{x_2}a^{1/4}$$

After conversion the quantum equivalent 2 level canonical cascade becomes:

$$f(x) = (R_x\pi/2)(CZ_3)(R_x - \pi/4)(CZ_2)(R_x - \pi/4)(CZ_2)(CZ_1)(R_x - \pi/4)(CZ_2)(R_x\pi/4)$$

Since the target qubit operates at the same basis states, just as the control qubits, a significant simplification can be made at the input of x_3 and the target qubit. Since $f(x) = (R_x\pi/2)(CZ_3)(R_x\pi/2) = x_3$, it can be subtracted out and completely removed, thereby leaving a remainder X rotation of $\pi/4$. Figure 3-12 shows the quantum circuit before reduction:

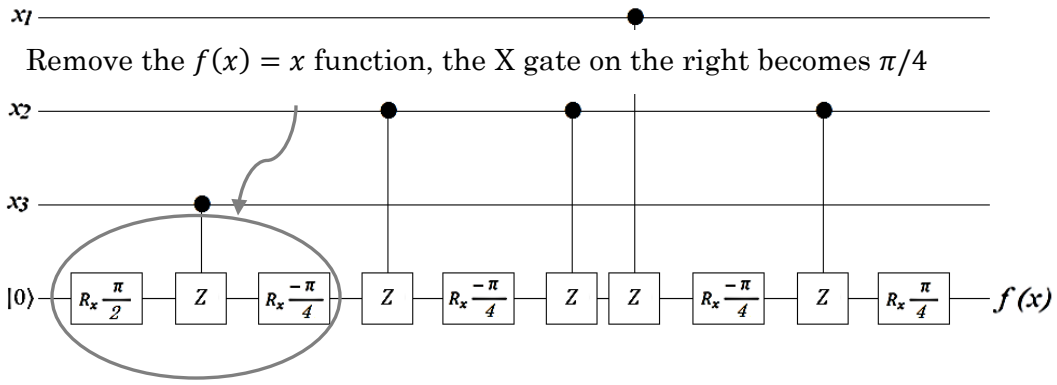


Figure 3-12. Exact Quantum Binary (EQB) Canonical Cascade before reduction of a Toffoli Gate

Since x_3 has only one control point to the target qubit, x_3 can safely assume the role of the target qubit. This is a significant reduction, since it removes all Ancilla qubits and reduces the primitive gate count to only eight. The reduced quantum cascade is shown in Figure 3-13:

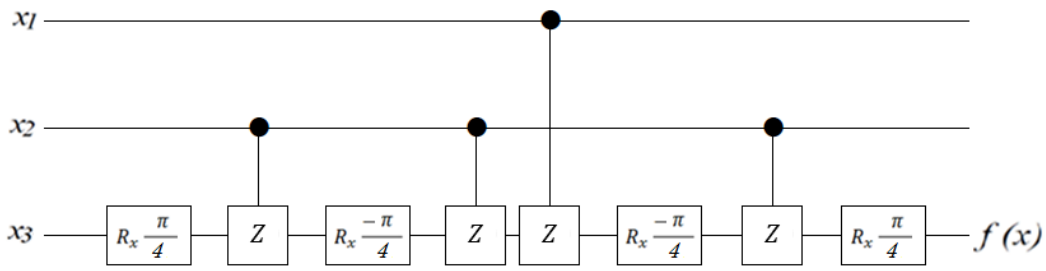


Figure 3-13. Exact Quantum Binary (EQB) Canonical Cascade after reduction of a Toffoli Gate

(End of Example)

The application of CZ gates literally spins a qubit about the Z axis and the number of rotations is determined by input qubit configuration. The X rotations cause the qubit to change spin orientation and is equivalent to a controlled precession. The qubit will settle at one of its observable states. Figure 3-14 shows how the precession of the qubit may appear during the execution of arbitrary gates using the group decomposition method with exact quantum binary output:

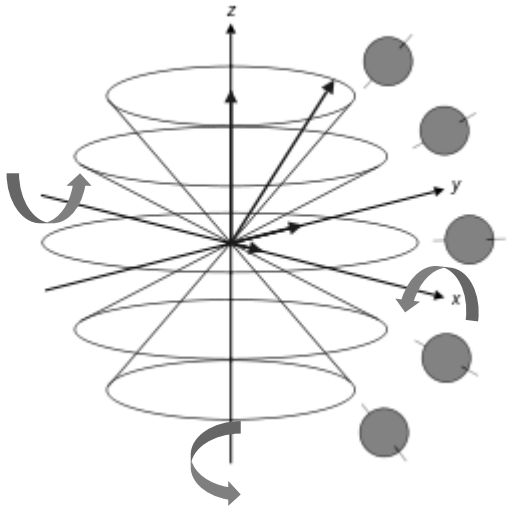


Figure 3-14. EQB Qubit Precession during EQB gate execution. The control Z gates rotate the qubit around the Z axis, while the X rotation gates changes the precession around the Z axis.

It should be clearly pointed out there is a strong dependency on the size of the angular rotation and the number of input variables via $A_{rot} = 2^{-n} W_n \vec{f}$. This dependacy suggests that the number of input variables is limited to the smallest accurate rotation the quantum computing hardware can handle. For instance, if the hardware is capable of 0.7° rotations, with an accuracy of < 0.35°, than the maximum number of inputs that can safely be utilized is approximately $n = 8$, depending on the function (this is worst case, where there are no cascade reductions). Currently ion trap quantum gate fidelities, with carful setup and execution, exceed 99.7% accuracy [28]. I will show in the next chapter that this is not a circuit size boundary, as one of the significant advantages of cascades is that they can be instantiated as black

boxes or single complex gates to create functions with any number of input variables.

3.5 Summary and Conclusions

At the beginning of Chapter 3, in Section 3.1, the probabilistic nature of qubits and quantum computing in general was discussed. It was shown that the orientation of a qubit can be directly inferred by its probability amplitude of its two basis states. While nature does not allow us to directly observe the quantum world, it has given us the ability to statistically derive it. More importantly, the probabilistic models give us insight and some understanding of the structure and complexity of higher dimensional space that qubits exist in. This multidimensional space in turn gives us shortcuts and pathways for solving certain problems more efficiently. Aside from massive quantum parallelism, as mentioned in Chapter 1, a superposition in a higher dimension can allow for an extra degree of freedom for quantum circuit synthesis. But before taking advantage of higher order space, its effects on the observables must be explored. Section 3.2 mathematically demonstrates the process of measuring a quantum qubit, after it has undergone some basic quantum rotations and the gross effects on the qubit after the measurement. Measurements also expose the fact that conjugate pairs, or any other phase rotations, cannot be statistically distinguished and can be safely treated as a single probabilistic state. As pointed out in Section 3.3 any statistically

distinguishable state can also be treated as a discrete quantum state. This allows us to overload quantum qubits and potentially treat each as a quantum qudit. Fortunately, the controlled phase rotations not only alter the qubit state in a non-observable alteration, but if implemented as a controlled phase rotation and when combined with X or Y rotations, form a universal quantum gate. Section 3.4 demonstrates three methods for modifying the group decomposition method into synthesizing quantum circuits; all the three methods harness the power of a controlled phase rotation.

1. **Method One: SWAP and controlled SWAP gates only** – This method utilized full swap gates as group rotation elements and directly reused the controlled swap gates, as Fredkin gates, to generate group decomposed canonical cascades. In addition, this method requires 3 Ancilla qubits, making its application expensive. Full swap gates are generally expensive, except when implemented with quantum dot technology. Controlled swap gates are built from controlled Z gates (regardless of technology controlled Z gates have to be built from other technology specific elementary gates, making the controlled swap gate expensive, regardless of technology).
2. **Method Two: Probabilistic state model** – This method takes advantage of the fact that qubits in certain superposition of its basis states can be treated as a discrete quantum state and when combined with

controlled Z gates, can generate group decomposed canonical cascades. Group rotation elements are converted to simple X or Y rotations and were combined with simple controlled Z gates. This allowed for the elimination of 2 Ancilla qubits, but since the cascades operated with non-basis states, the application of such cascades is once again very limited in scope.

3. **Method Three: Exact Quantum Binary (EQB)** – This final method takes advantage of the fact that the basis states can be reached with theoretically perfect accuracy by simply extending the number of discrete statistical states to infinity. This allows the cascade to operate in pure basis states allowing the combination of certain control qubits and the target qubit. Again group rotation elements are converted to X or Y rotation gates and when combined with controlled Z gates, decomposed canonical cascades are generated. The weakness of this methodology is that the rotation angles are highly correlated to the number of input variables, which in turn are controlled by the quantum computer hardware.

The methods proposed are not necessarily competing for lowest quantum cost, but most certainly confirm that the transition between reversible logic is smooth and continuous into quantum logic. The first and third methodologies have interesting characteristics for minimization and ease of synthesis and

each has certain attractiveness when considering the available technology. The second methodology has limited uses, mostly confined to binary controlled multi-valued quantum systems. A more careful and complete evaluation will be provided in the next chapters especially the exploration of more complex function and adapting the methods for multiple output synthesis.

Chapter 4

QUANTUM CIRCUIT SYNTHESIS USING EQB CANONICAL CASCADES

In this chapter I will explore more complex function synthesis using EQB and introduce a simple synthesis program created specifically for this thesis. The chapter starts off by examining how EQB can be implemented as a quantum black box and how phase errors can be corrected. The chapter then introduces the synthesis tool. Finally, various benchmark functions are compared against EQB synthesis.

So far this thesis has explored the generation of the most primitive quantum gates, using two promising technologies, and how Sasao's group decomposition can be modified to utilize these primitive quantum gates in synthesizing quantum circuits. EQB is a direct result of this relatively simple modification and elegant synthesis methodology. However, while the methodology proves totally successful in creating circuits with a relatively small number of inputs, it possess a major limitation when the angle of rotation for X and Y decreases as the number of inputs increases and exceeds the capabilities of the quantum computing hardware. For this reason a method needs to be explored and developed so that large input and multi-output functions can be safely realized without violating the capabilities of the quantum computer hardware. There is also a secondary flaw in EQB that

has been for the most part ignored, as it does not affect the measured output, but may affect the way EQB can be combined with other quantum circuits that operate in probabilistic space. Some quantum circuits need some sort of a quantum phase error correction circuitry, and EQB is no exception. As pointed out in Chapter 1, quantum computers will be subjected to environmental noise, which will cause affected bits to potentially flip their states. One way to detect and correct the bit flip errors is to estimate the phase acquired during the bit flip and use correction circuitry to counter rotate the affected bit into its correct basis state. Since EQB uses primitive X or Y rotations, with no global phase applied (recall the decomposition of X and Y gates from Chapter 1) in conjunction with Z gates, the output will, in certain input combinations, contain a phase of -1, $-i$, or i. Since phase is not observable, it does not have any impact of the output state. Nonetheless, a phase correction is in order, especially since it is already needed in any quantum circuitry due environmental decoherence. EQB phase induction can be corrected using standard phase error correction circuits along with phase adjustments made to the input qubits.

4.1 EQB in a Black Box

Before error correction circuitry can be added, a quantum function must be implemented as a unitary transformation. In addition the unitary transformation must also be implemented to have a control line, such that it

conditionally transforms an eigenvector bounded in a Hilbert space. That is, a quantum function Q is unitary if and only if $Q * Q^{\dagger} = I$ († denotes conjugate transpose), its eigenvectors are orthogonal ($Q^T = Q^{-1}$, its transpose is equivalent to its inverse), and Q can be implemented as CQ (controlled unitary transformation). To generate an EQB black box, large functions can be decomposed to smaller groups, with all Z gates encapsulated by rotation gates on either side. The remaining function fragment should contain sufficient rotation gates to rotate the target qubits to either of the basis states of $|0\rangle$ or $|1\rangle$. Figure 4-1 demonstrates one way to generate an EQB black box (note the sum of the angles of the X rotation gates are 0 or π):

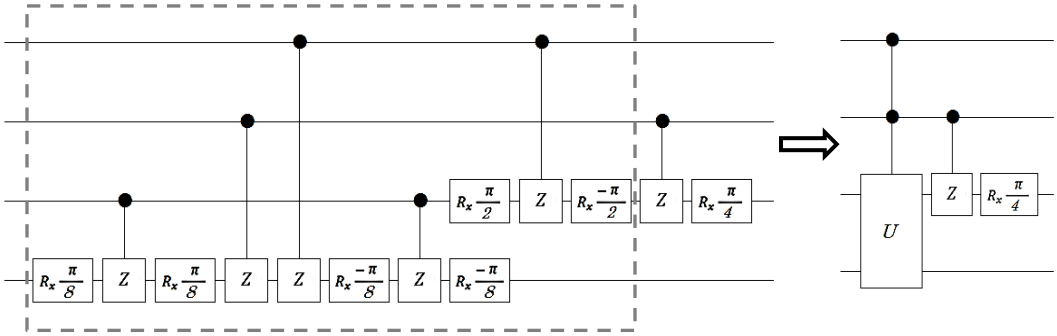


Figure 4-1. Unitary Gate Substitution

Since the black box, outlined by the dashed box in Figure 4-1 above, meets the unitary requirements, the entire circuit segment can be replaced by controlled U gate. Immediately after the U gate is also a convenient place to instantiate any error correction circuitry if needed. The next section outlines a possible way of generating error correction circuitry.

4.2 Phase Kick-Back

Recall that controlled phase gates operate on two qubits and induce a phase change in both qubits. If a specific qubit is chosen as a reference qubit, then it assumes the role of a control qubit, otherwise there is no explicit control qubit. Figure 4-2 shows the equivalence of two controlled phase gates, with different control and target qubits (the inversion is simulated with two swap gates):

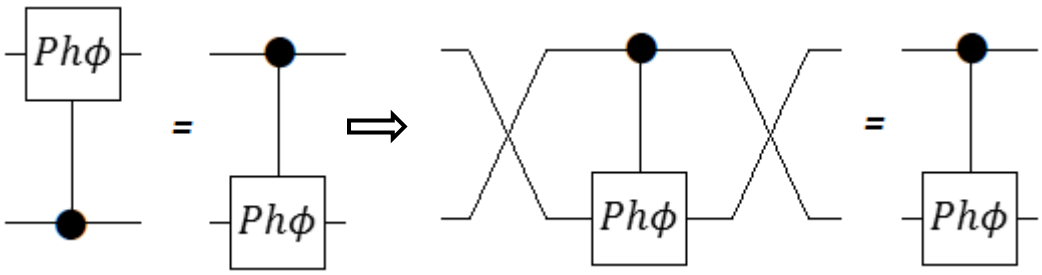


Figure 4-2. Controlled Phase Gate Inversion and Equivalence

The equivalent result is shown with matrix multiplication below:

$$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix} * \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&e^{i\phi}\end{bmatrix} * \begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix} = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&e^{i\phi}\end{bmatrix}$$

The Figure 4-2 and the aforementioned matrix multiplication shows that phase is relative and affects two qubits. The phase of one qubit is transferred through the control line to alter the phase of the other qubit. This convenient behavior is referred to as phase kick-back. Error correction circuitry can also

take advantage of this phenomenon by converting the phase of the control qubit into probabilistic amplitude that can be either measured and/or used to control an error correction gate. This is true for any controlled gate that introduces a phase rotation into its target qubit. The Figure 4-3 shows how the control qubit of a controlled arbitrary Z rotation gate can be converted into a controlled arbitrary X rotation gate:

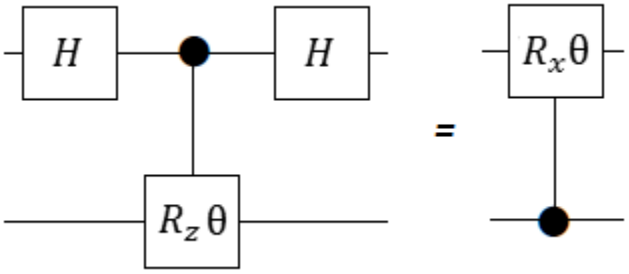


Figure 4-3. Converting Phase-Kick-Back into Probability Amplitude

The result is shown with matrix multiplication (for simplicity $\theta = \pi$ and a phase of $\pi/2$ has been applied, which essentially creates a controlled NOT gate):

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{bmatrix} * \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}$$

The angle of the X rotation is equal to the angle of the Z rotation and does not need to be used to create an upside down controlled X gate. Instead, the controlled X rotation can be used as the control line to a gate which can

correct or alter the phase and/or the probability amplitude of any other output target line. Figure 4-4 shows how such a circuit can be used to correct the phase and amplitude of output qubits:

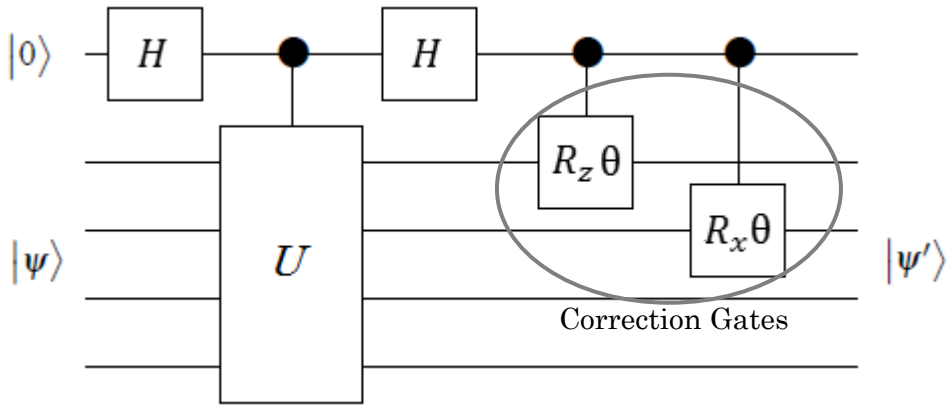


Figure 4-4. Application of Error Correction Gates

For as long as EQB circuit segments are used with other EQB circuit segments, or any other segments that are phase tolerant, than EQB can safely be used for circuit synthesis. If EQB is combined with circuit segments that require specific phase, than phase estimation and correction circuitry must be applied at the EQB output prior to concatenation. It should also be pointed out the phase estimation circuitry has other important uses, such as factoring algorithms and Fourier transforms [29], however they are beyond the scope of this thesis.

4.3 EQB Synthesis Tool

The EQB synthesis tool is a simple program, currently implemented in Visual Basic for Applications (VBA) that allows for simple quantum circuit design and result verification. The decision to use VBA was made solely based on availability and interfacing quickly with excel and user input forms or windows. The EQB synthesis tool can generate a canonical cascade that is modified for quantum application, exactly as outlined in Chapter 3, and also provides a solution for classical reversible cascades as outlined by Sasao in Chapter 2. Currently the EQB synthesis tool is limited to 10 input variables, as it has to generate a string with 2^{2n} elements that represents a Walsh matrix, which significantly increases the processing time needed to calculate a solution. The user must provide the number of input variables, the number of logic levels and a completely specified function truth vector. If the EQB synthesis is desired, the user must type in EQB in the logic level field. The program also provides the user with a visual representation of the quantum rotations that are being executed on the target qubit, as the cascade is either executed one gate at a time or the entire cascade executed at once. Figure 4-5 shows the user interface:

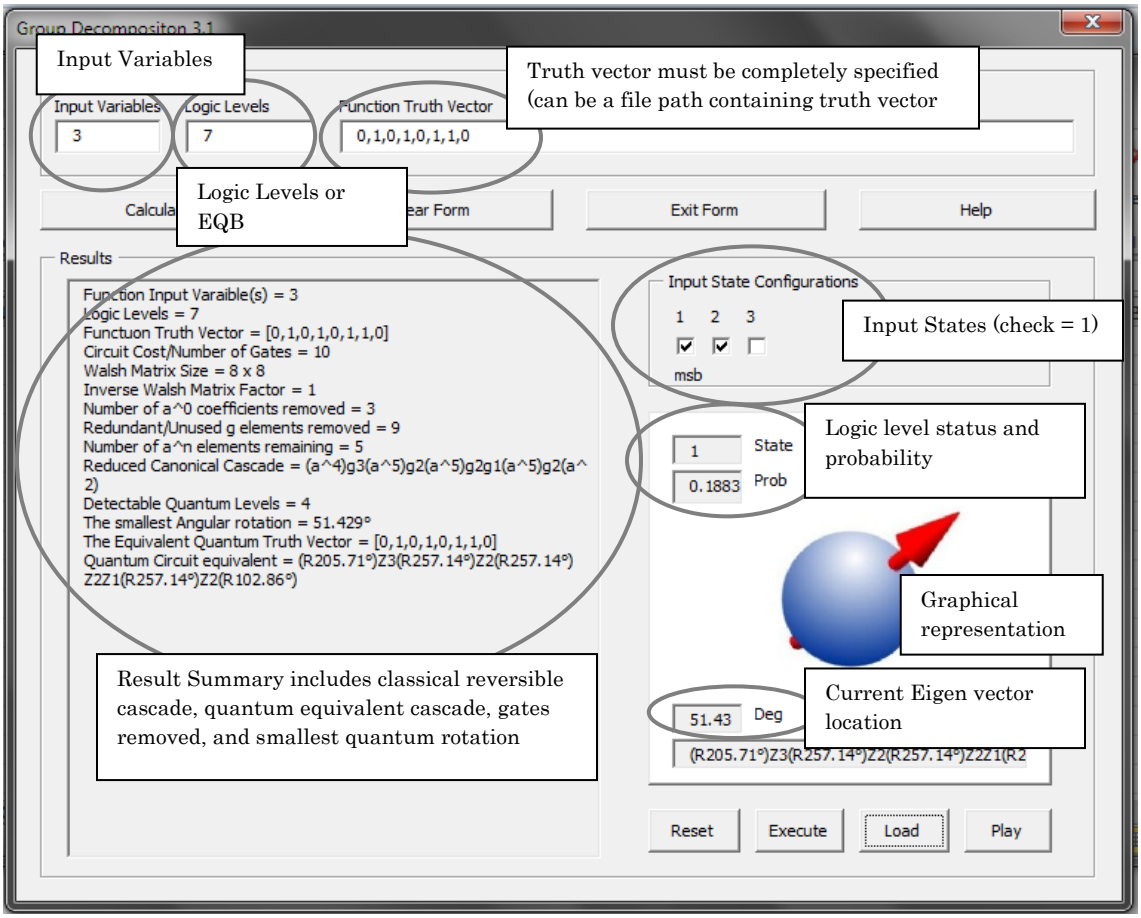


Figure 4-5. Quantum Canonical Cascade Synthesis Application

The EQB synthesis tool starts off by generating a string which represents a Walsh ordered matrix of size 2^{2n}, multiplies it by the truth vector, and deletes all string elements containing a zero coefficient, along with affected adjacent elements $(g^n a^0 g^n = I)$. The synthesis then generates an inverse Walsh factor based on the fraction 2^{-n}, with a modulo of $2p$, where p is the logic levels entered by the user. If EQB is desired, than no modulo is applied. The inverse Walsh factor is multiplied by the remaining string elements to

generate the Walsh spectrum as $\vec{w} = 2^{-n} W_n \vec{F}$. If EQB is requested, the program checks the truth vector for symmetry and if symmetry exists, than the target qubit becomes an input for the least significant input variable. This symmetry allows for the removal of at least one Z gate, one rotation gate and removes the Ancilla qubit. It should be pointed out that searching for symmetries and responding to certain symmetries almost always seems to result in a reduction in the cascade. Currently the EQB synthesis does not perform an exhaustive search for symmetries nor symmetries resulting from any don't care states or any other ESOP minimizations. It is very possible that additional reductions will be discovered if such searches are exhaustively performed on functions with a large number of input variables, but such analysis will be reserved for future work.

4.4 EQB Bench Mark Comparison

In this section a methodology is developed for evaluating the quantum cost of EQB compared to other methodologies, such as ESOP synthesis in XORCISM software, on a limited set of benchmark reversible quantum circuits. It should be pointed out that EQB may not be the most minimal implementation and performing a more aggressive truth vector search for symmetries, may provide additional reductions. Also note that quantum cost is very much technology specific. As was shown in Chapter 3, Section 3.4.1, a circuit implementation with the SWAP is cheaper with quantum dot than it

is with ion trap technologies. Lastly, if multiple technologies survive in the long run, each technology will have its own decoherence time, and each gate will consume a certain amount of execution time. Thus gate cost should be evaluated by the amount of decoherence time a circuit consumes. However since current technologies are still in rapid development, their decoherence times are not well documented, and gate decoherence time consumption evaluation will be saved for future work. For now we will use Maslov quantum cost as a robust referance [25].

Recall from Chapter 3 that the most universal gate that can be implemented with ion traps and quantum dots is the controlled Z gate. Both technologies interestingly required even more elementary operations, such as qubit swaps and individual qubit rotations. Both technologies can implement a controlled Z gate with a total of five elementary operations (either laser pulses or other electromagnetic manipulations). Unanimously it appears that these elementary qubit operations can be combined into a single composite operations that implement controlled gates with a cost of one. In addition it takes at least two additional qubit manipulations, encapsulating the controlled Z gate, to implement a controlled NOT operation with a quantum cost of one [25] (a Z gate sandwiched between two Hadamard gates implements a controlled NOT gate). Similar recombination is appropriate with EQB, where each single Z gate with an adjacent individual rotation gate

can be safely considered to have a quantum cost of one. The combination of the X rotation gates with the controlled Z gates is the equivalent of creating a controlled V or a Feynman gate. If there are controlled Z gates without an adjacent individual rotation gate, than the controlled Z gate has a cost of one. Literally, the quantum cost is entirely dependent on the number of controlled Z gates in the cascade and whether or not the target qubit is an Ancilla qubit or not. Lastly, since EQB function induces a potential phase error, the error correction circuitry has an associated quantum cost, however this cost will be counted separately, as its instantiation is application specific. The error correction circuitry has an average quantum cost of two for every input qubit, not used as a target (each control line has two Hadamard gates), and every target qubit will have one correction gate. EQB error correction circuitry can be further optimized on a case-by-case basis, as some of the target qubits are controlled by other target qubits and may not need any phase correction.

Multi-output functions are currently not addressed well in EQB, but as mentioned before with unitary function segments, multi-output function can be quickly assembled and reduced. Figure 4-6 demonstartes an example of a Fredkin gate assembled by two CNOT gates and a Toffoli (phase correction excluded).

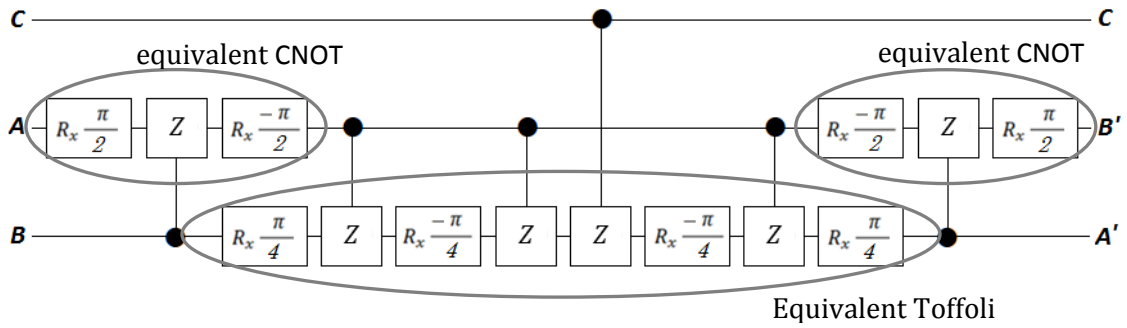


Figure 4-6. Equivalent Fredkin Gate

Another important multi-output function is the 1-bit adder. The Carry Out function is implemented first in EQB and then the three input exclusive OR function is implemented for the sum output. Figure 4-7 shows the equivalent 1-bit adder (phase correction excluded).

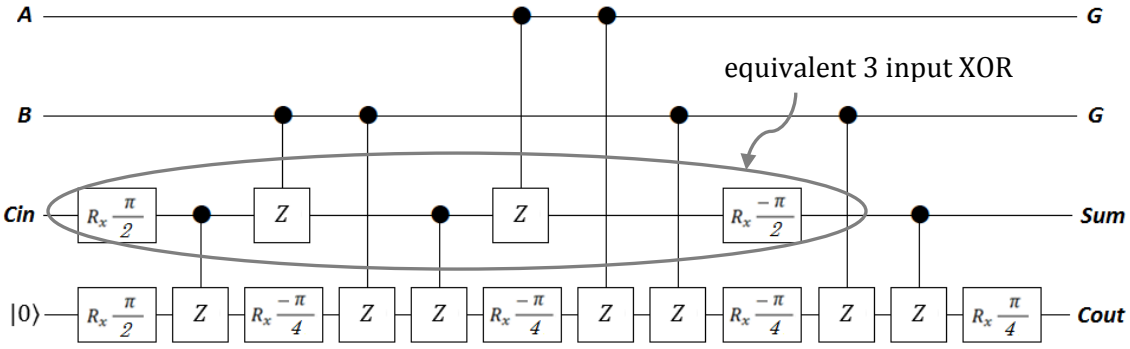


Figure 4-7. Equivalent 1-Bit adder

Table 4-1 summarizes some benchmark circuits, generated by EQB, against quantum costs reported by RevLib [30] and Dimitri Maslov [25]. Each benchmark function was chosen because its outputs are limited to one or two

(EQB is not currently optimized for multiple outputs). Table 4-1 is organized as follows:

- EQB Gate Count – Records the primitive gate count reported by EQB and includes all controlled Z gates and all X rotation gates.
- EQB Quantum Cost – Records the number of controlled gates reported by EQB after absorption of all X rotation gates into neighboring controlled Z gates (Maslov quantum cost).
- Correction Quantum Cost – Records the number of gates needed to detect a phase shift through the control lines and number of correction gates needed to correct the phase at each output (on average the cost is $n(2+m)$, where n is the number of inputs and m is the number of outputs).
- Ancilla Qubits – Records whether an Ancilla qubit is used as the output line.
- Best Reported – Records the lowest quantum cost reported by the aforementioned references [25], [30].
- Worst Reported – Records the highest quantum cost reported by the aforementioned references [25], [30].
- Percent Potential Improvement – Records the percentage of reduction in Maslov cost. If there is no reduction, than no change (n.c.) is reported and if EQB increases the cost, then worse is reported.

Benchmark Function	EQB Primitive Gate Count	EQB Quantum Cost (Maslov cost)	Correction Quantum Cost	Ancilla Qubits	Best Reported from [25], [30]	Worst Reported from [25], [30]	Percent Potential Improve-ment
Fredkin (controlled swap)	14	6	4	0	7	15	14
Feynman (CNOT)	3	1	3	0	1	n/a	n.c.
Toffoli	8	4	6	0	5	n/a	20
Toffoli4, t4	19	11	9	0	13	n/a	15
Toffoli5, t5	42	26	12	0	26	29	n.c.
Toffoli6, t6	89	57	15	0	38	61	worse
1-bit Adder / rd32	16	9	6	1	9	29	n.c.
4 greater than 10 (4gt10)	42	26	9	1	34	53	14
4 greater than 11 (4gt11)	8	4	9	1	7	16	43
4 greater than 12 (4gt12)	42	26	9	1	41	58	37
4 greater than 13	19	11	9	1	15	34	27

(4gt13)							
4 greater than 4 (4gt4)	42	26	9	1	54	89	52
4 greater than 5 (4gt5)	19	11	9	1	21	29	48

Table 4-1. Quantum Cost Comparison Table

4.5 Summary and Conclusions

In this chapter, it is recognized that the synthesis method in EQB induces a potentially undesired phase shift at the output of each qubit. Normally this does not pose any measurement issue or issues with instantiating EQB with other non-EQB quantum cascade, but should be eliminated if other downstream gates require a specific phase. For this reason phase kickback was explored as a potential remedy and alludes to the fact that phase correction circuitry can be combined with other error correction circuitry aimed at eliminating environmental decoherence. The EQB simulation program was introduced to demonstrate the ease with which group decomposition can be employed to synthesize quantum circuits and other smaller segments. These unitary segments are low cost and can be used to generate larger functions with multiple outputs. When combined with phase correction, a direct comparison can be made to other benchmark functions. However since phase correction is only required for specific implementations

and therefore its instantiation is needed only once at the output, its cost is reported separately. Since large functions can be implemented in segments, other well known reduction algorithms can be utilized to further reduce the quantum cost, especially if the function is not fully specified.

Chapter 5

CONCLUSION AND FUTURE WORK

Since the prevailing quantum computing hardware may not be easily predicted, and its associated implications cannot be fully anticipated, a universal low cost quantum synthesis methodology is needed. Such a methodology is presented in this thesis, which has its own benefits and implications. In these final summarizing remarks, a brief list of outstanding issues is presented. Further analysis of the outstanding issues is appropriate, as it may provide additional discoveries to more efficiently synthesize and implement quantum circuitry.

5.1 Summary

As presented in Chapter 1 of this thesis, there are a wide range of quantum computer technologies being aggressively researched. Each technology has its own methodology for implementing efficient quantum gates for single and multi-qubit operations. It is very clear however, that unitary orthonormal operators are a convenient way to describe the complex behaviors of the quantum world within the computing context. Two drastically different quantum technologies were presented in Chapter 1 and the reader was presented with a high level overview of the process of

implementing quantum computing hardware and the derivation of single and multi-qubit operations, which can then be universally utilized for logic function generation. The selection for presenting the two technologies, the ion trap and quantum dots, was made because both are relatively easy to comprehend and implement quantum computers. The two technologies however produce two fundamentally different types of quantum gates. In the ion trap, the most natural and universal multi qubit gate that can be implemented is a controlled phase gate, while a swap gate is best in the quantum dot implementation. While both types of gates are universal, meaning that they can be used to implement any function, their cost on implementation is far more expensive when the implementation is reversed (a square-root-of-swap gate in an ion trap implementation forms the most expensive two qubit gate, and the same is true for the phase gate in a quantum dot implementation). It is then interesting to note that efficient circuit synthesis is highly technology dependent.

In Chapter 2 an interesting paper written by Tsutomu Sasao demonstrates an inventive and convenient way to synthesize binary controlled multi-valued classically revisable functions, using group function decomposition. Though the Sasao's method was proposed only for classical reversible functions, with minor adjustments, the methodology can be extended to quantum circuits. Of course the extension is ultimately made

possible by the fact that quantum circuits are reversible in nature. The significance of the group decomposition method for quantum circuit synthesis is that it can utilize swap gates and controlled rotation gates, making the synthesis methodology highly generalized and technology independent.

In Chapter 3 the group decomposed elements of Chapter 2 were mapped to quantum gates, and therefore truly adapting the methodology to quantum circuit synthesis. In an attempt to demonstrate that group decomposition is technology independent, the group elements were both converted to purely swap gates, as well as purely rotational gates. An extension to multi-valued logic was also made, using probabilistic methods and therefore preserving the unitization of the quantum computing hardware presented in Chapter 3. It is also important to note that the group decomposition method can be extended to other multi-valued quantum computing hardware (hardware such as ions with hyperfine levels and double quantum dot systems), but such technologies are not well defined so far and therefore excluded from this thesis. A final modification for quantum circuit synthesis, using group decomposition, was made possible for purely binary quantum systems, while utilizing the two dimesional Hilbert space that qubits inherently occupy. This method was termed exact quantum binary, or EQB, and is the product of the desire to generalize and prove that the group decomposition methodology can utilize any quantum technology to efficiently implement

quantum circuits. EQB works for all binary controlled, binary output functions, by increasing the number of logic levels to infinity and producing a smooth continuity between the basis quantum states. This is important, as it allows for the harnessing of the most important quantum computing characteristics and massive computing parallelism through superposition. As most systems in nature, there are trade-offs, and EQB is no exception. Since EQB is confined to two basis states, of control and outut variables, the qubit rotation angles become highly correlated to the number of input combinations. This means that the angle of rotation decreases by a factor determined by the number of inputs and essentially limits the EQB application to a finite number of inputs. In turn the number of inputs is determined by the smallest high fidelity qubit rotation achievable by the quantum computing hardware.

In Chapter 4 another fundamental EQB inconsistency was exposed. EQB does a great job at implementing functions with relatively low quantum cost, but it does so by inadvertently inducing a quantum phase change at the output. The phase contamination does not pose any risk of compromising a measurement at the output of a function, and normally does not pose a risk of inducing logic errors when EQB functions are concatenated to other non EQB functions. However, phase correction circuitry is in order, especially when the phase correction can be implemented in conjunction with any error correction

circuitry needed to preserve computational basis from environmental decoherence. The chapter continues with the introduction of a synthesis program that generates quantum canonical cascades based on completely specified truth vectors. The program provides a quick and easy method to test and evaluate synthesized circuits by quickly determining the Walsh spectrum of a function, employing simple simplification methods, and producing a canonical cascade made up of either swap gates or rotation gates. The chapter finally provides the evaluation of the performance of EQB against some benchmark functions, that were earlier implemented with other synthesis methods. The analysis shows that EQB is cost competitive, even with the addition of phase correction circuitry, provided that the number of inputs is small enough so that the quantum computing hardware produces gate rotations with high fidelity.

5.2 Accomplishments

The major accomplishment in this thesis is the adaptation of Sasao's group decomposition and Walsh spectrum method for describing classical reversible canonical cascades, into describing quantum canonical cascades. Though the transition from classical reversible logic to quantum reversible logic is somewhat smooth, the quantum computing hardware is a major consideration to account for. What may appear to be an efficient method for circuit synthesis in one technology may be very expensive and infeasible in

another. The group decomposition method had to be further generalized so that it can provide efficient synthesis in multiple types of quantum computing hardware. This adaptation was accomplished by converting the elements in the cyclical groups into primitive quantum rotation gates with variable angles. The mapping of the elements is fundamental to EQB is the primary contribution of this thesis.

5.3 Future Works

EQB is a direct result of this thesis and though it succeeds in its mission of establishing a simple, efficient method for synthesizing quantum circuits with group decomposition, it is not a complete work. At minimum, sufficient time needs to be dedicated to eliminate, or perhaps reduce, the relationship between the angle of rotation and the number of input variables. I strongly believe that there exists a way of confining the reduction of the angle in the rotation gates, such that it is sufficient in producing just enough degrees of freedom and allow the implementation of unrestrictedly large functions. In addition, the phase contamination should be eliminated via the rotation gates just as is done in the elementary Pauli rotation gates and other controlled quantum gates, the difficulty of course being that the input variable combinations or states determine the phase contamination. Other areas of improvement include the extension of EQB to multiple outputs and the reuse of garbage outputs. Reduction strategies are another unlimited area that can

be explored to take advantage of partially specified functions, ESOP minimizations, and some function properties and function symmetries. Lastly, as technologies become better defined, EQB can be modified to use gate implementation that can be a more natural implementation in the quantum computing hardware. One example is using square-root-of-swap gates to not only implement functions, but also comply with linear nearest neighbor requirements of quantum computing, as well as other hardware restrictions.

REFERENCES

[1] Rubin, Brad, "Brad Rubin's Computatorium™, The Bloch Sphere Part 1," Brad Rubin & Associates, Inc, 29 12 2008. [Online]. Available: http://bradrubin.com. [Accessed 11 5 2013].

[2] Williams, Collin P., "Quantum Gates," in *Explorations in Quantum computing, Texts in Computer Science*, London, Springer-Verlag Limited, 2011, p. Chapter 2.

[3] Smite-Meister, "Wikimedia Commons, the free media repository," Wikimedia Commons, 30 1 2009. [Online]. Available: http://commons. wikimedia.org. [Accessed 11 5 2013].

[4] Lieven M. K.Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S.Yannoni, Mark H.Sherwood, Isaac L.Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature,* vol. 414, no. 6866, pp. 883-887, 2001.

[5] Thomas Monz, Philipp Schindler, Julio T. Barreiro, Michael Chwalla, Daniel Nigg, William A. Coish, Maximilian Harlander, Wolfgang Hänsel, Markus Hennrich, and Rainer Blatt, "14-Qubit Entanglement: Creation and Coherence," *Physical Review Letters,* vol. 106, no. 13, p. 130506 (4 pages), 2011.

[6] C. F. Roos, Mark Riebe, H. Häffner, W. Hänsel, J. Benhelm, G. P. T. Lancaster, C. Becher, F. Schmidt-Kaler and R. Blatt, "Quantum Optics and Spectroscopy, Innsbruck University, Austria," 28 5 2004. [Online]. Available: http://heart-c704.uibk.ac.at/recent/controlandmeasuremente. html. [Accessed 11 5 2013].

[7] Stacey, John-Patrick, *Stabilization and Control In a Linear Ion Trap,* Oxford: Wadham College, University of Oxford, 2003.

[8] Holzscheiter, Michael H, "Ion-Trap Quantum Computation," *Los Alamos*

Science, vol. 27, pp. 264-283, 2002.

[9] Lee, PJ and Brickman, KA and Deslauriers, L and Haljan, PC and Duan, LM and Monroe, C, "Phase control of trapped ion quantum gates," *Journal of Optics B: Quantum and Semiclassical Optics,* vol. 7, no. 10, p. S371, 2005.

[10] Burkard, Guido and Loss, Daniel and DiVincenzo, David P, "Coupled quantum dots as quantum gates," *Physical Review B,* vol. 59, no. 3, p. 2070, 1999.

[11] Eriksson, Mark A and Friesen, Mark and Coppersmith, Susan N and Joynt, Robert and Klein, Levente J and Slinker, Keith and Tahan, Charles and Mooney, PM and Chu, JO and Koester, SJ, "Spin-based quantum dot quantum computing in silicon," *Quantum Information Processing,* vol. 3, no. 1-5, pp. 133-146, 2004.

[12] Koppens, FHL and Buizert, Christo and Tielrooij, Klaas-Jan and Vink, IT and Nowack, KC and Meunier, Tristan and Kouwenhoven, LP and Vandersypen, LMK, "Driven coherent oscillations of a single electron spin in a quantum dot," *Nature,* vol. 442, no. 7104, pp. 766-771, 2006.

[13] Loss, Daniel and DiVincenzo, David P, "Quantum computation with quantum dots," *Physical Review A,* vol. 57, no. 1, p. 120, 1998.

[14] Golovach, Vitaly N and Loss, Daniel, "Electron spins in artificial atoms and molecules for quantum computing," *Semiconductor science and technology,* vol. 17, no. 4, p. 355, 2002.

[15] Jacob Mason Taylor, *Hyperfine interactions and quantum information processing in quantum dots,* Cambridge, Massachusetts: Harvard University, 2006.

[16] Greilich, A and Economou, Sophia E and Spatzek, S and Yakovlev, DR and Reuter, D and Wieck, AD and Reinecke, TL and Bayer, M, "Ultrafast optical rotations of electron spins in quantum dots," *Nature Physics,* vol. 5, no. 4, p. 2009, 262-266.

[17] Edamatsu, Keiichi, "Quantum physics: Swift control of a single spin," *Nature,* vol. 456, no. 7219, pp. 182-183, 2008.

[18] Friesen, Mark and Tahan, Charles and Joynt, Robert and Eriksson, MA, "Spin Readout and Initialization in a Semiconductor Quantum Dot," *Physical review letters,* vol. 92, no. 3, p. 037901, 2004.

[19] Schliemann, John and Loss, Daniel and MacDonald, AH, "Double-occupancy errors, adiabaticity, and entanglement of spin qubits in quantum dots," *Physical Review B,* vol. 63, no. 8, p. 085311, 2001.

[20] Sasao, Tsutomu, "Cascade realizations of two-valued input multiple-valued output functions using decomposition of group functions," in *Multiple-Valued Logic, 2003. Proceedings. 33rd International Symposium on*, IEEE, 2003, pp. 125-132.

[21] Yoeli, Michael and Turner, James, "Decompositions of group functions with applications to two-rail cascades," *Information and Control,* vol. 10, no. 6, pp. 565-571, 1967.

[22] Fredkin, Edward and Toffoli, Tommaso, "Conservative Logic," *International Journal of Theoretical Physics,* vol. 21, pp. 219-253, 1982.

[23] Short, Robert A, "Two-rail cellular cascades," in *Proceedings of the November 30--December 1, 1965, fall joint computer conference, part I*, ACM, 1965, pp. 355-369.

[24] Lukac, Martin and Pivtoraiko, Mihail and Mishchenko, Alan and Perkowski, Marek, "Automated synthesis of generalized reversible cascades using genetic algorithms," in *5th International Workshop on Boolean Problems*, 2002, pp. 33-45.

[25] Dmitri Maslov, "Reversible Logic Synthesis Benchmarks Page," 2 9 2009. [Online]. Available: http://webhome.cs.uvic.ca/~dmaslov/. [Accessed 11 5 2013].

[26] Lanyon, Benjamin P and Barbieri, Marco and Almeida, Marcelo P and Jennewein, Thomas and Ralph, Timothy C and Resch, Kevin J and Pryde, Geoff J and O'Brien, Jeremy L and Gilchrist, Alexei and White,

Andrew G, "Simplifying Quantum Logic Using Higher-dimensional Hilbert Spaces," *Nature Physics,* vol. 5, no. 2, pp. 134-140, 2008.

[27] James, Daniel FV and Kwiat, Paul G and Munro, William J and White, Andrew G, "Measurement of qubits," *Physical Review A,* vol. 64, no. 5, p. 052312, 2001.

[28] Babikov, Dmitri, "Accuracy of gates in a quantum computer based on vibrational eigenstates," *The Journal of chemical physics,* vol. 121, p. 7577, 2004.

[29] Shor, Peter W, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM journal on computing,* vol. 26, no. 5, pp. 1484-1509, 1997.

[30] Robert Wille, Daniel Große, Lisa Teuber, Gerhard W. Dueck and Rolf Drechsler, "RevLib: An Online Resource for Reversible Functions and Reversible Circuits," in *Int'l Symp. on Multi-Valued Logic*, RevLib, 2008, pp. 220-225.

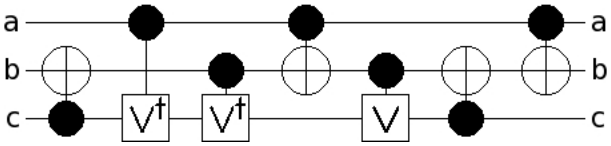
APPENDIX A

Bench Mark Circuit Schematics used for EQB comparison in Chapter 4, Table 4-1. The schematics are downloaded from http://www.revlib.org [30]. Not all circuits in Table 4-1 had available schematics and are not included in this appendix.

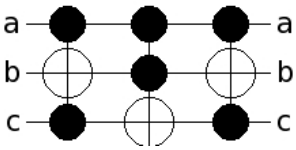
Fredkin Gate Truth Table:

abc	abc
000	000
001	001
010	010
011	011
100	100
101	110
110	101
111	111

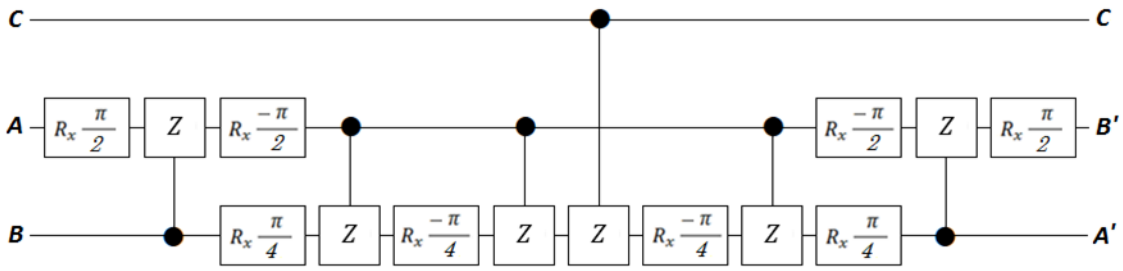
Fredkin Gate with cost of 7:



Fredkin Gate with cost of 15:



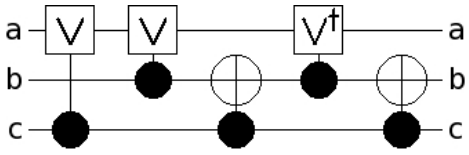
Fredkin Gate Generated by EQB with cost of 6:



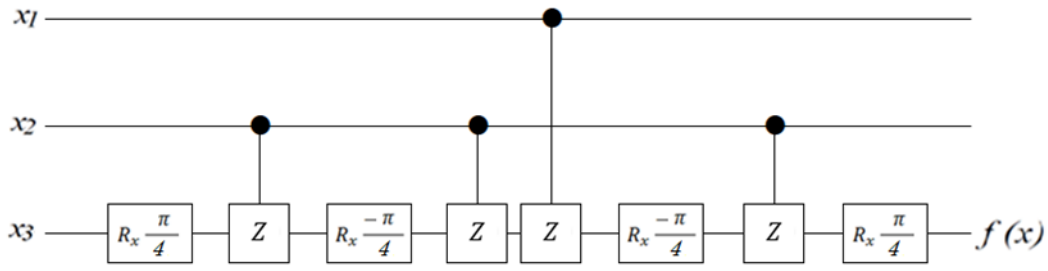
Toffoli Gate Truth Table:

abc	abc
000	000
001	001
010	010
011	111
100	100
101	101
110	110
111	011

Toffoli Gate with cost of 5:



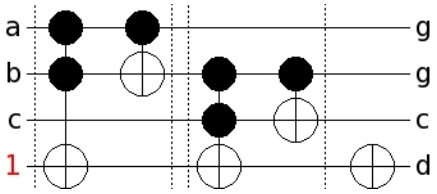
Toffoli Gate Generated by EQB with cost of 4:



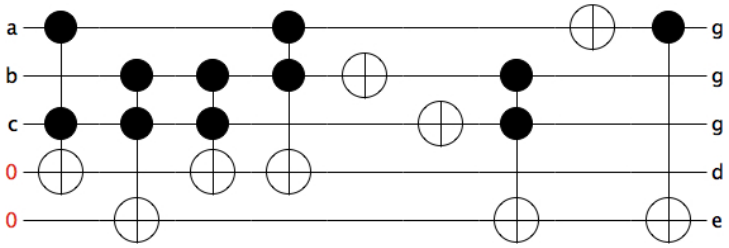
1-bit Adder, rd32, Truth Table:

adc	cd
000	00
001	01
010	01
011	10
100	01
101	10
110	10
111	11

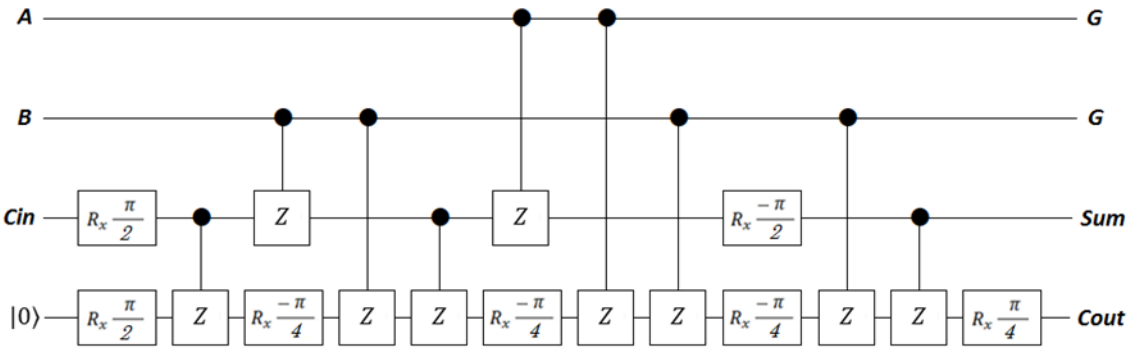
1-bit Adder, rd32 with cost of 9:



1-bit Adder, rd32 with cost of 29:



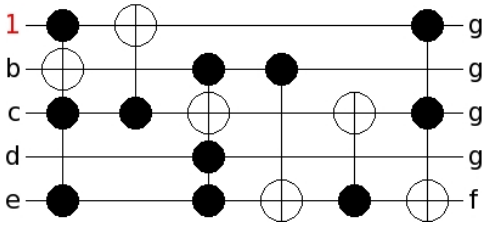
1-bit Adder, rd32, Generated by EQB with cost of 9:



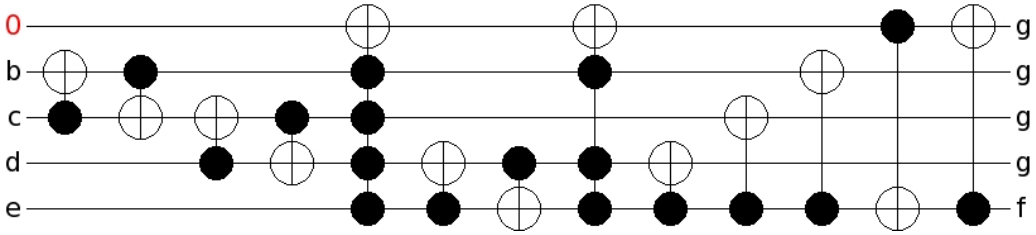
4 greater than 10 (4gt10) Truth Table:

bcde	f
1011	1
1100	1
1101	1
1110	1
1111	1

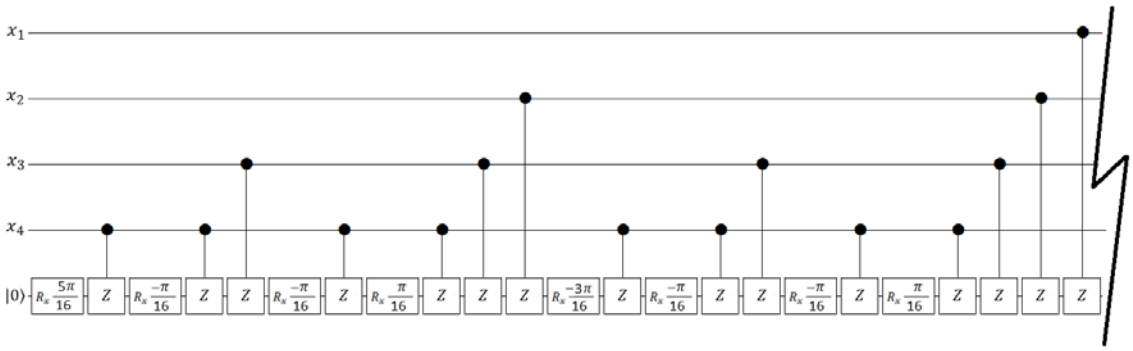
4 greater than 10 (4gt10) with cost of 34:

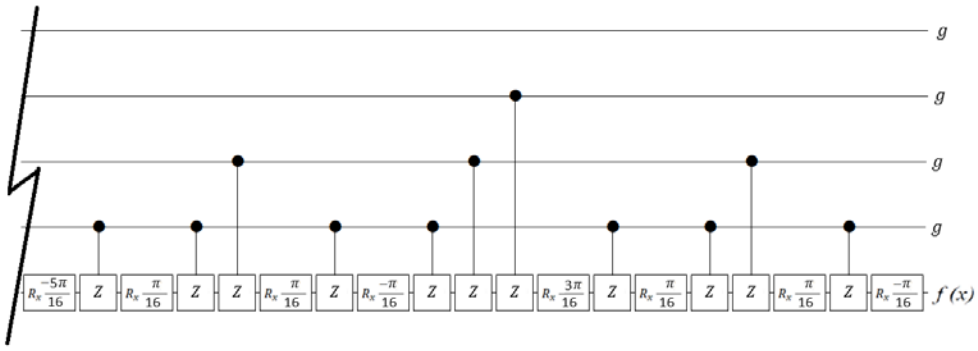


4 greater than 10 (4gt10) with cost of 53:



4 greater than 10 (4gt10) Generated by EQB with cost of 26 (x_1 is msb):

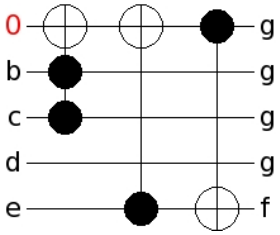




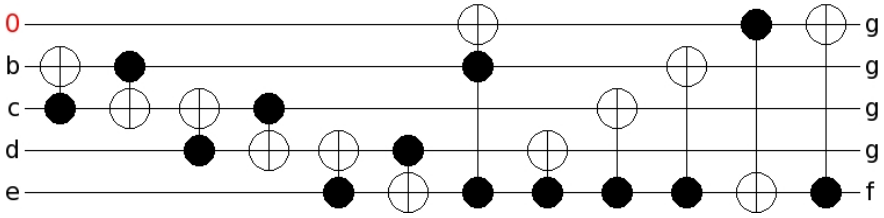
4 greater than 11 (4gt11) Truth Table:

bcde	f
1100	1
1101	1
1110	1
1111	1

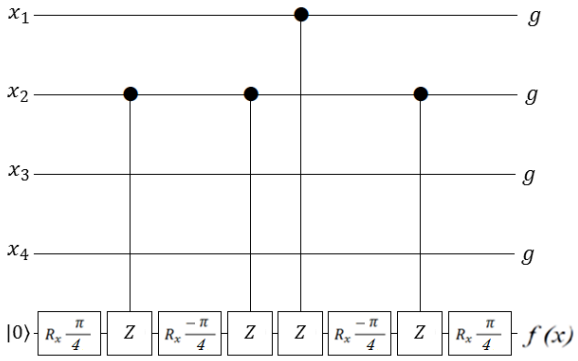
4 greater than 11 (4gt11) with cost of 7:



4 greater than 11 (4gt11) with cost of 16:



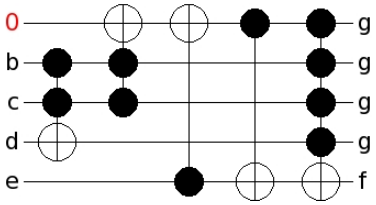
4 greater than 11 (4gt11) Generated by EQB with cost of 4 (x_1 is msb):



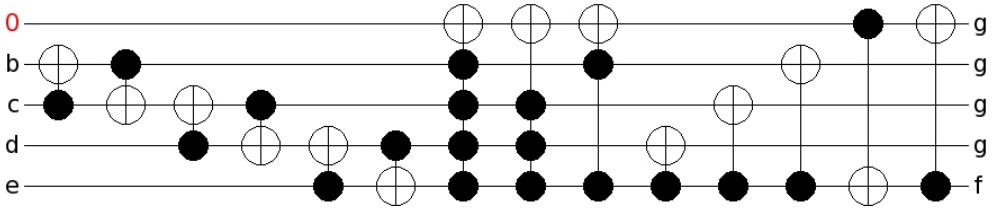
4 greater than 12 (4gt12) Truth Table:

bcde	f
1101	1
1110	1
1111	1

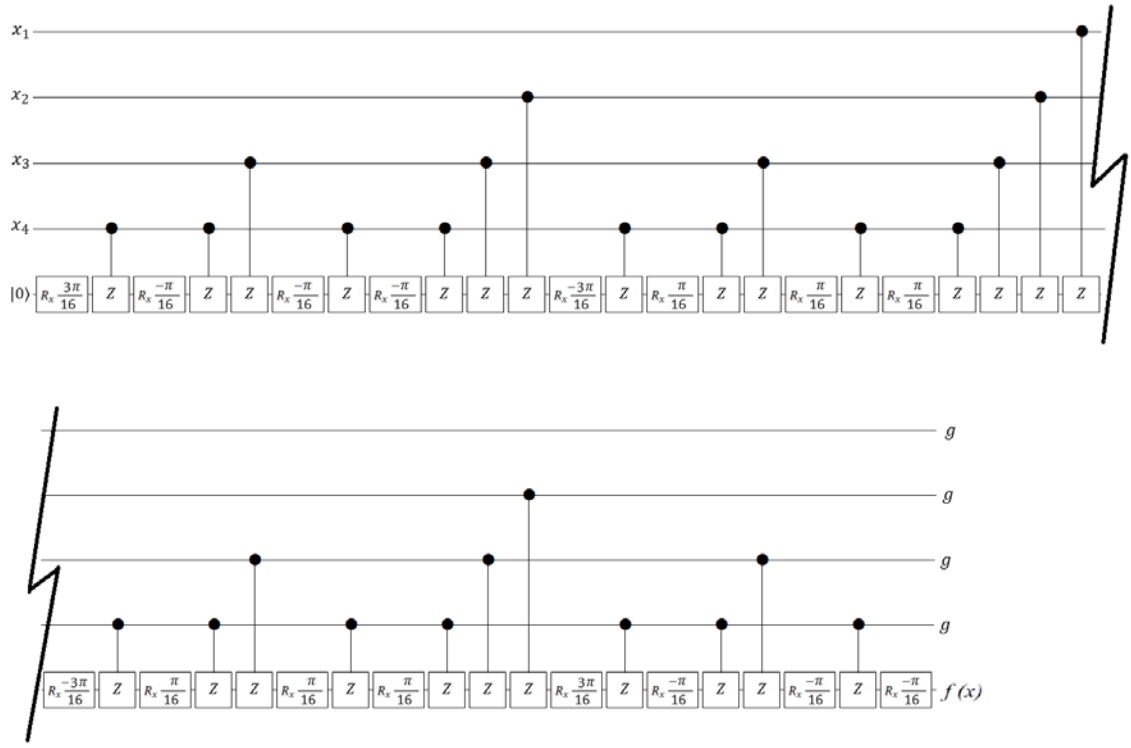
4 greater than 12 (4gt12) with cost of 41:



4 greater than 12 (4gt12) with cost of 58:



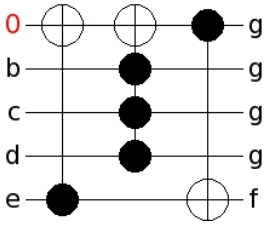
4 greater than 12 (4gt12) Generated by EQB with cost of 26 (x_1 is msb):



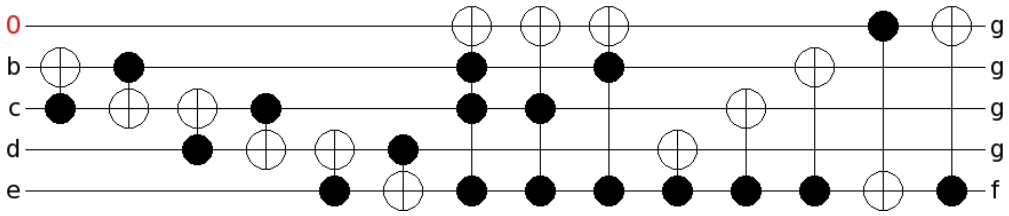
4 greater than 13 (4gt13) Truth Table:

bcde	f
1110	1
1111	1

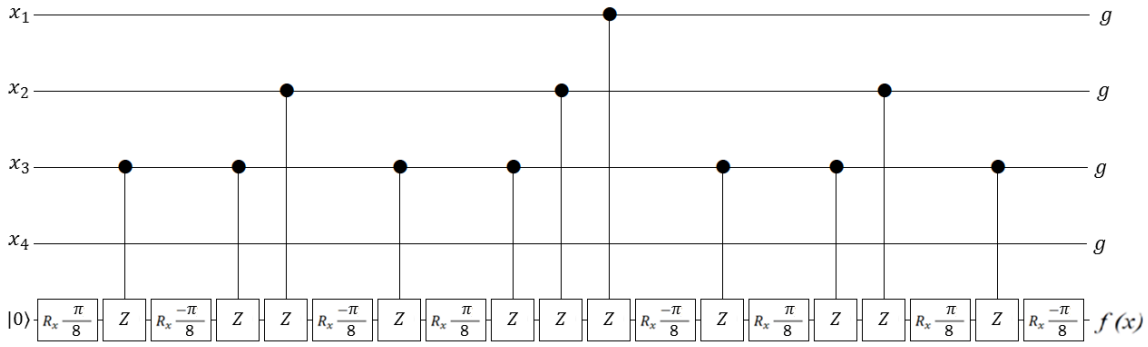
4 greater than 13 (4gt13) with cost of 15:



4 greater than 13 (4gt13) with cost of 34:



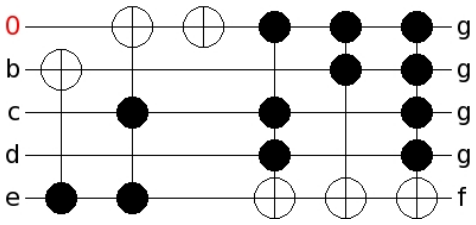
4 greater than 13 (4gt13) Generated by EQB with cost of 11 (x_1 is msb):



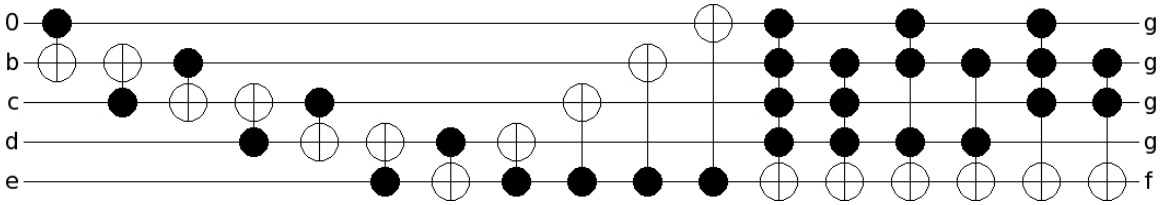
4 greater than 4 (4gt4) Truth Table:

bcde	f
0101	1
0110	1
0111	1
1000	1
1001	1
1010	1
1011	1
1100	1
1101	1
1110	1
1111	1

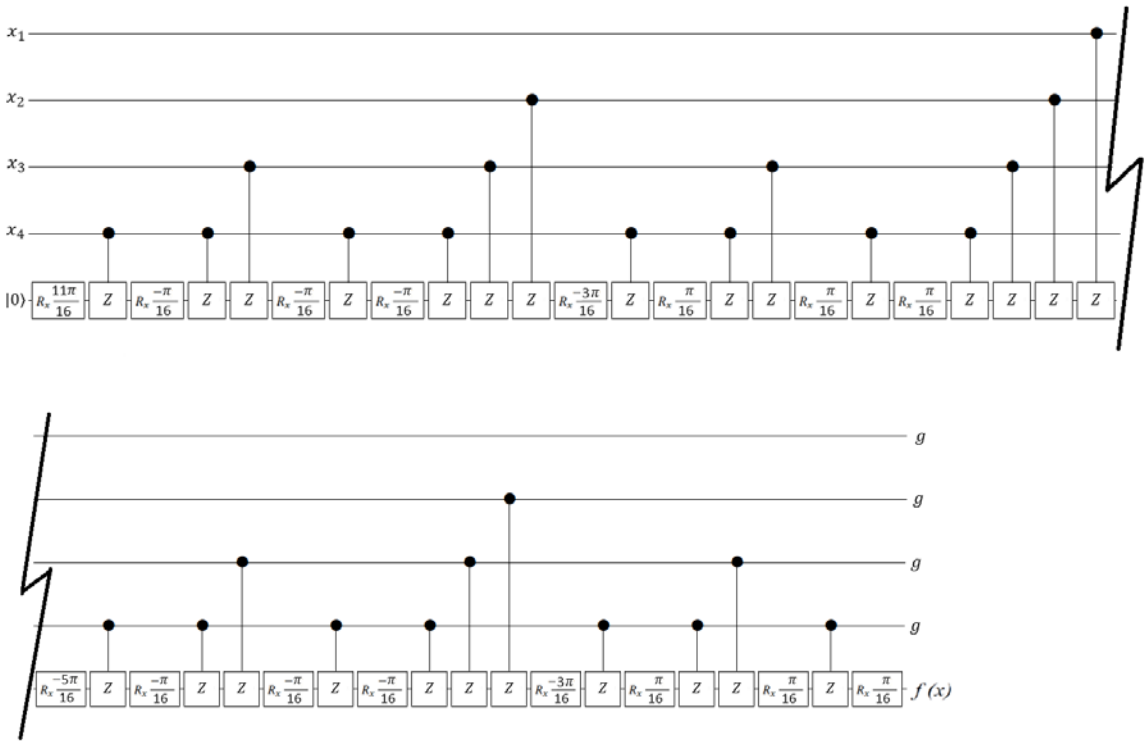
4 greater than 4 (4gt4) with cost of 54:



4 greater than 4 (4gt4) with cost of 89:



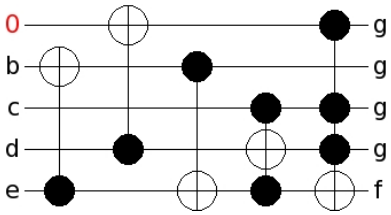
4 greater than 4 (4gt4) Generated by EQB with cost of 26 (x_1 is msb):



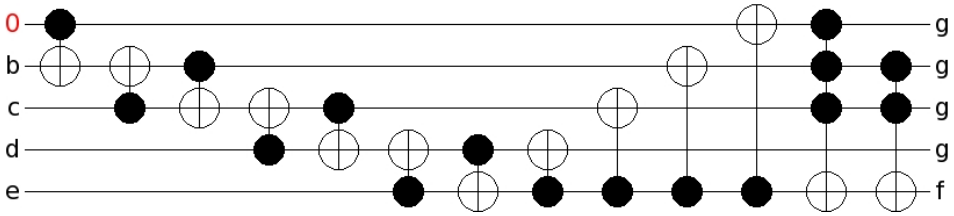
4 greater than 5 (4gt5) Truth Table:

bcde	f
0110	1
0111	1
1000	1
1001	1
1010	1
1011	1
1100	1
1101	1
1110	1
1111	1

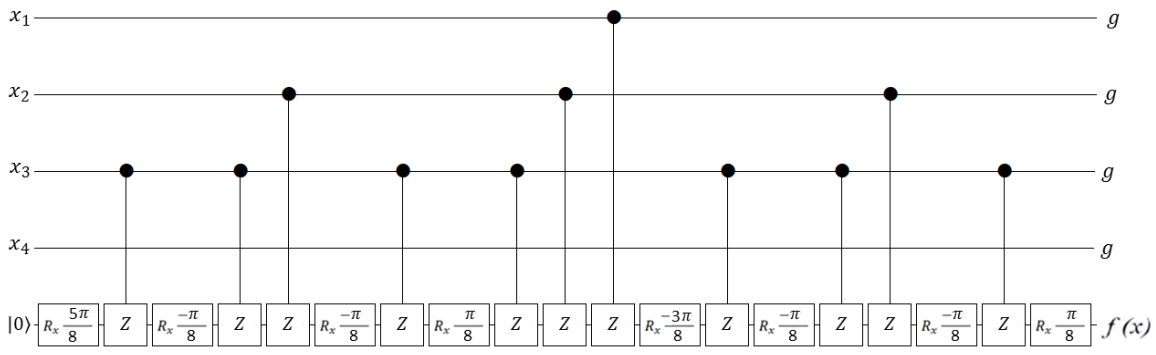
4 greater than 5 (4gt5) with cost of 21:



4 greater than 5 (4gt5) with cost of 29:



4 greater than 5 (4gt5) Generated by EQB with cost of 11 (x_1 is msb):



APPENDIX B

This thesis is accompanied by the Group Decomposition with EQB circuit synthesis Macro tool. The macro was created in VBA and is imbedded in the provided Quantum Circuit Synthesis using Group Decomposition with EQB macro.xls file.

File name:	Quantum Circuit Synthesis using Group Decomposition with EQB macro.xls
File Type:	Microsoft Excel 97-2003 Worksheet (xls)
File size:	4MB
Required Software:	Microsoft Excel 97-2003 or later
System Requirements:	Pentium processor with a clock speed of at least 233MHz and 128MB of RAM
Operating System:	Windows 2000 or later
For Mac:	To run on Mac systems, must use MS office 2004 or later versions (functionality has not been tested on Mac systems)

Instructions:

1. Download the "Quantum Circuit Synthesis using Group Decomposition with EQB macro.xls" file to any desired destination
2. Open the "Quantum Circuit Synthesis using Group Decomposition with EQB macro.xls" file
3. If security settings show warnings, ensure the file is opened for editing and macro's are enabled
4. Click the button "Group Decomposition with EQB 3.1 Macro" to start the macro. To exit the macro click on the "Exit Form" button (clicking on the X will reload the form)