GENERAL LOGICAL LINEAR NEAREST NEIGHBOR (LLNN) ARCHITECTURE FOR FAULT-TOLERANT QUANTUM COMPUTATION

A Thesis by

Pratima Kumari Mudbhari

Master of Science, Wichita State University, 2008

Bachelor of Engineering, Kathmandu University, Nepal, 2005

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

December 2008

GENERAL LOGICAL LINEAR NEAREST NEIGHBOR (LLNN) ARCHITECTURE FOR FAULT-TOLERANT QUANTUM COMPUTATION

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

_____
Preethika Kumar, Committee Chair

_____
Steven R. Skinner, Committee Member

_____
Stephen Brady, Committee Member

DEDICATION

To my God, my grandma, my parents, my sister, my brother, my teachers and my dear friends

Trust, believe and have faith.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my advisor, Dr. Preethika Kumar, for her full support and guidance given to me in accomplishing this thesis. Also, I would like to express my sincere appreciation to my committee member, Dr. Steven R. Skinner, for his precious time and valuable guidance. The effectiveness of their meetings and devotion to their precious time was the reason that enabled me to fulfill my obligations.

I would like to truly thank my committee member Dr. Stephen Brady from Mathematics Department for his valuable time and suggestions.

Special tribute is to be paid to my family members, my grandma Sabitri Singh, my father Pawan Mudbhari, my mother, Prem Lata Mudbhari, my sisters, Bandana Mudbhari and Jyoti Mudbhari and my brother, Bhaskar Mudbhari for their support, motivation and love.

I would also like to thank my friends Hassan Naseh, Ashok Pant, Dristi Shrestha, Huzefa Amreliwala, Anusha Dissayanayeke, Ruby Bohara, Kiran Kandimalla, Deepika Kumar and Ushan Goonawardane for being good friends during my happy and difficult times.

Most of all, I would like to thank God for everlasting love and bringing good people to me.

ABSTRACT

We present a general scheme for implementing fault-tolerant quantum computation. We discuss a general two-dimensional architecture of qubits involving only linear nearest neighbor interactions. Between the qubits by using ancillas, we show how to implement gate operations for encoding, error correction, fault-tolerant quantum computation and decoding procedures in our design. The architecture is designed with two different coupling parameters $\xi_1$and ξ_2 between the qubits. A universal set of gate operations (Controlled-NOT, Hadamard, Phase, T) are performed on the encoded logical qubits fault tolerantly, by pulsing the bias on the target qubit to a certain value for a chosen time duration.

Initially, we designed an architecture (discussed in Appendix A) for fault-tolerant computation that was capable of correcting errors, non fault-tolerantly. Since it is possible that the error correction circuit itself propagate errors, we designed an architecture that corrects errors, fault tolerantly. Finally, we compare our architecture to an existing architecture for fault-tolerant computation employing a linear one-dimensional nearest neighbor array of qubits and show how ours is more efficient.

Even though all through this work we use the specific instance of the 7-qubit Steane code in describing our gate implementations, our method can be extended to all systems employing transversal gates for fault-tolerant quantum computation.

PREFACE

A quantum computer is a device for computation that makes direct use of distinctively quantum mechanical phenomenon, such as superposition and entanglement, to perform operations on data. In a quantum computer, the fundamental unit of information is called a quantum bit or qubit. A qubit can exist not only in a state corresponding to the logical states 0 and 1 of a classical bit, but also in states corresponding to a superposition of these classical states. In other words, a qubit can exist as a $|0\rangle$, $|1\rangle$, or a linear combination of $|0\rangle$ and $|1\rangle$, i.e., the state of the qubit can be represented by the wave function, $|\psi\rangle = \alpha(t)|0\rangle + \beta(t)|1\rangle$ where $|\alpha(t)|^2 + |\beta(t)|^2 = 1$. The squares of the normalized complex coefficients, $|\alpha(t)|^2$ and $|\beta(t)|^2$, represent the probabilities of finding the qubit in the $|0\rangle$ and $|1\rangle$ states, respectively. Therefore, a qubit can be thought of as a unit vector in a two-dimensional complex vector space known as the Hilbert space and the joint state of N qubits in a quantum computer is represented by a 2^N dimensional vector.

Quantum computation is performing computations using the principles of quantum mechanics. In 1982, Feynman showed how a quantum system could be used to do computations. He also explained how such a machine would be able to act as a simulator for quantum physics. In other words, a physicist would have the ability to carry out experiments in quantum physics inside a quantum mechanical computer. Later, in 1985, Deutsch realized that Feynman's assertion could eventually lead to a general purpose quantum computer and published a crucial theoretical paper which showed that any physical process could be modeled perfectly by a quantum computer. Thus, a quantum computer would have capabilities far beyond those of any traditional classical computer. After Deutsch published this paper, the search began to find interesting applications for such a machine. In 1994, Shor came up with a quantum algorithm

named factorization. He showed how an ensemble of mathematical operations, designed specifically for a quantum computer, could be organized to enable such a machine to factor huge numbers extremely rapidly, much faster than is possible on conventional computers. These cryptosystems rely on the fact that extremely large numbers are "impossible" to factorize on present day classical computers. Although Shor's result was only theoretical, his algorithm sparked interest in quantum computation.

However, when it came to the physical realization of a quantum computer, physicists encountered fundamental practical problems. The quantum computers, which have been built so far, have been found to be extremely sensitive to external influence. Unlike classical computers, the errors made by a quantum computer are continuous, and so the accumulation of errors would most certainly destroy the coherence of the entire system. Therefore, an error correcting procedure is required to be found in order to do successful quantum computation. Such a type of error correction procedure was considered impossible, since classical error correcting is done mostly by duplication, whereas the no-cloning theorem prevented such duplications in quantum computing. In 1995, Peter Shor made the important discovery that a nine-qubit logical qubit can correct a single arbitrary error, despite the continuous nature of such errors. After Shor's error correcting code, many other codes were developed. Together Calderbank, Shor and Steane developed a general form of error correcting codes called the CSS codes. Daniel Gottesman developed yet another set of codes called stabilizer codes which encompassed CSS codes. Most Quantum Error Correcting codes (QEC) have been found to be stabilizer codes. Stabilizer codes constitute an important class of quantum error correction codes whose construction is analogous to classical linear codes. The purpose of good quantum error correcting is not just to protect transmitted information, but also to dynamically correct errors while undergoing computation.

The area of fault-tolerant computation uses the idea.

The basic idea of fault-tolerant quantum computation is to compute directly on encoded states such that decoding is never required before the computation. Each qubit in the original circuit can be replaced with an encoded block of qubits, using different error correction code such as Steane code, Shor's code and 5-qubit code. These error-correcting codes are able to correct a single error. Gates applied to encoded data are classified into two types: transversal and non-transversal. A transversal encoded gate is applied by performing the corresponding physical gate independently on each of the qubits comprising the encoded qubit. A non-transversal encoded gate is decomposed into a more complex set of physical operations, including multi-qubit physical operations between physical qubits within the same encoded qubit. The reason for choosing 7-qubit Steane code for fault-tolerant quantum computation is its transversal property, which involves gates that interacts the i^{th} qubit in a block with itself and the i^{th} qubits in other blocks. In this way, a single bad gate anywhere in the system can only spread to produce one error per block of the code, which prevents overwhelming the error tolerance of any single block. The Steane code allows us to perform fault-tolerant computation on a universal set of gates (CNOT, H, X, Z, S, T) where the gate operations are performed transversally.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CSS	Calderbank-Shor-Steane Codes
FT	Fault-Tolerant
FTEC	Fault-Tolerant Error Correction

CHAPTER 1

INTRODUCTION

Several schemes for encoding a quantum bit into a logical qubit using quantum error correcting codes exist in the literature [2, 7, 12]. Each qubit in the original circuit can be replaced with an encoded block of qubits, using different error correction codes such as 3-qubit code, 5-qubit code, 7-qubit Steane code and 9-qubit Shor's code. The whole purpose of encoding a physical qubit into a logical qubit is to introduce redundancy in order to combat errors due to decoherence. In quantum mechanics, quantum decoherence is the mechanism by which quantum systems interact with their environments to exhibit probabilistically additive behavior. Alternatively, the decay of quantum information caused by the coupling of the system to its environment (bath) is referred to as decoherence. Since, under decoherence, the system becomes entangled with its environment in some unknown way, description of the system by itself cannot be done without also referring to the environment (i.e. without also describing the state of the environment). Thus, the dynamics become non-unitary, i.e., irreversible, and this represents a major problem for the practical realization of quantum computers since they rely heavily on the undisturbed evolution of quantum systems.

Even though quantum error correcting codes are used to combat decoherence, an error correcting code, by itself, is useful, provided the quantum gate operations performed on the qubits during computation are effectively perfect. However, even a perfect two-qubit gate can propagate a pre-existing error from one of the qubits involved in the gate operation to the other one. Therefore, a single erroneous gate can indirectly cause errors in many qubits. For instance, under a CNOT gate operation, an X (bit-flip) error on the control qubit propagates to the target and a Z (phase-flip) error on the target propagates to the control [12]. A bit-flip error will change

the state of the qubit from one basis state to the other, i.e., if the state of the qubit initially is in the superposition state, $\alpha|0\rangle+\beta|1\rangle$, under the bit-flip error it changes to $\alpha|1\rangle+\beta|0\rangle$. A phase-flip error does not change the state of the qubit, but introduces a phase of "π" radians if the qubit is in the $|1\rangle$ state, i.e., if the state of the qubit initially is $\alpha|0\rangle+\beta|1\rangle$, under the phase-flip error it changes to $\alpha|0\rangle-\beta|1\rangle$. A Y error is a combination of a bit-flip and a phase-flip error. Therefore, in order to achieve arbitrarily good quantum computation, it becomes important not only to provide protection of stored or transmitted quantum information, but also to provide protection of the quantum information as it dynamically undergoes computation. This calls for fault-tolerant quantum computation.

The basic idea of fault-tolerant quantum computation is to compute directly on encoded states such that decoding is never required before the computation [12, 33]. Once the qubits are encoded, computations and error correction can be performed at any time. The qubits are not required to be decoded before the computation. This prevents propagation and accumulation of errors between the encoding and decoding procedures. Each qubit in the original circuit can be replaced with an encoded block of qubits, using different error correction codes such as 3-qubit code, 5-qubit code, 7-qubit Steane code and 9-qubit Shor's code which is discussed in [2, 7, 12]. These error-correcting codes are able to correct an arbitrary single-qubit error. Gates applied to an encoded qubit are classified into two types: transversal and non-transversal [2, 12]. Under a transversal gate operation, the i^{th} qubit in an encoded block interacts either with itself or the i^{th} qubits in other blocks. Thus, a single bad gate anywhere in the system can only spread to produce one error per block of the code which prevents overwhelming the error tolerance of any single block. A set of universal gate operations (CNOT, Hadamard, phase and T) are implemented transversally using 7-qubit Steane code which has been discussed in Appendix A.

On the other hand, a non-transversal encoded gate is decomposed into a more complex set of operations, including multi-qubit operations between physical qubits within the same encoded qubit.

In [8], an overview of the basic concepts behind quantum error correction and quantum fault tolerance is presented where stabilizer codes, CSS codes and transversal gates are discussed. A method for constructing optimal fault-tolerant approximations of a universal gate set using the specific case of the Steane code is shown in [2]. However, no architecture has been introduced to implement the gate operations. General circuits of non-fault-tolerant error correction using 7-qubit Steane code is presented in [33]. In [7], the disadvantages of non-fault-tolerant error correction is discussed where it is shown how an error can propagate from an ancilla to the encoded data when using non-fault-tolerant error correction. This means that even though fault-tolerant computation is employed, errors are introduced as a result of the non-fault-tolerant error correction circuit, which itself propagates errors that cannot be handled. To prevent this, the idea of fault-tolerant error correction was developed which is discussed in [7]. In [3], fault-tolerant error correction in a one-dimensional linear nearest neighbor architecture is implemented where the error correction code used is the 7-qubit Steane code. Since each logical qubit comprises seven physical qubits, 42 SWAP operations are required to get the individual logical qubits to the necessary locations to perform logical gate operations in the architecture shown in [3]. The architectures presented in this thesis are much more efficient than the one-dimensional approaches in [3], since SWAP operations are not required to perform computation between the logical qubits. This is because the encoded qubits are stacked in vertical arrays and transversal gate operations can be directly preformed between corresponding qubits of two encoded qubits.

Initially, an architecture was designed for performing fault-tolerant computation with non-fault-tolerant error correction, details of which have been presented in Appendices B and C. However, the error correction procedure being non fault-tolerant, the correction circuit itself propagates errors from the error syndrome qubits to the logical qubits. Moreover, performing error correction on several qubits only increases the number of errors introduced by the error correction procedures. For this reason, the architecture was modified to one that could correct errors fault tolerantly.

Here, we propose a two-dimensional architectural layout to implement *fault-tolerant quantum computation*. The architecture is able is correct errors fault tolerantly. The architecture only assumes nearest-neighbor interactions between qubits. The entire architecture is divided into two blocks - Block I and Block II. Each block is used for a different task. Block I is used for input/output, encoding and error correction, whereas Block II is used for computation. In Block I, an arbitrary qubit state is prepared and is encoded in to a logical qubit using the Steane code. If a single-qubit error occurs during encoding, the logical qubit is then corrected *fault tolerantly*, using an error correction circuit. The corrected logical qubits are next moved to Block II to perform the desired fault-tolerant quantum computations on it. We use the term "computer" for the part of the architecture where fault-tolerant quantum computations are performed on the encoded qubit. After computation, the logical qubit is moved to Block I for error correction. The logical qubit is decoded, if required, and the decoded state is available for read out. The encoding, error correction and data transfer operations within the architecture require gate operations. We show how to calculate parameters of the system to achieve the desired gate operations. (Appendices D and E discuss the evolution of a single-qubit and a two-qubit system, respectively. In Appendix F, we show a method to calculate the bias pulses that are required to

perform a CNOT gate operation in a linear array of qubits). Using terminology specific to Josephson junction qubits, in achieving these operations, the only control parameter is the bias acting on the individual qubits. The coupling between qubits and the tunneling parameters of individual qubits are treated as fixed parameters of the system.

CHAPTER 2

GENERAL LOGICAL LINEAR NEAREST NEIGHBOR (LLNN) ARCHITECTURE FOR FAULT TOLERANT COMPUTATION

P. K. Mudbhari, P. Kumar and S. R. Skinner

2.1 Abstract

We present a general scheme for implementing fault-tolerant quantum computation. We discuss a general two-dimensional architecture of qubits involving only linear nearest neighbor interactions. By using ancillas, we show how to implement gate operations for encoding, fault-tolerant error correction, fault-tolerant quantum computation and decoding procedures in our design. Even though we use the 7-qubit Steane code as a specific instance in describing our gate implementations, our method can be extended to all systems employing transversal gates for fault-tolerant quantum computation.

2.2 Index Terms

Coupling, decoding, encoding, error correction, fault tolerance, Josephson junction qubits, Steane code.

2.3 Introduction

Many proposals on architectures for quantum computing use Linear Nearest Neighbor (LNN) technology in a one dimensional array of qubits [1-6], which makes the computation viable. One problem that arises with this architecture is that when error detection and correction are incorporated into the computational scheme to make it fault-tolerant, the amount of computational overhead can greatly exceed the benefit. Fault-tolerant error correction (FTEC) [3, 7, 8, 9] is important in quantum computation, because it can correct errors due to noise and decoherence [10, 11]. This is accomplished by adding redundancy into the system through

encoding qubits into logical qubits, for which several schemes exist [12-27]. In the architecture example presented in this paper, we use the 7-qubit Steane code [2, 15, 28], where a logical qubit is comprised of 7 encoded qubits. This makes computation in a one-dimensional architecture difficult as logical operations require a large number of SWAP operations to get the individual qubits to the necessary locations [3, 4].

In this paper, we propose a two-dimensional architectural layout in which fault-tolerant quantum computation and fault-tolerant error correction using the 7-qubit Steane code are implemented. The architecture is much more efficient than one-dimensional approaches [3], since the encoded qubits are stacked in vertical arrays and SWAP operations are not required to perform computation between the logical qubits.

2.4 Overall Architecture

Our architectural design is shown in Fig 1. Qubits are represented by circles and are assumed to be identical in that they have the same tunneling parameter, which is again fixed during fabrication. Here, and throughout the paper, we will be using terminology specific to Josephson-junction qubits. The entire architecture is divided into 2 blocks: Block I, and Block II. Block I is used to input an arbitrary single qubit state, encode the qubit to corresponding logical qubit, correct the logical qubit whenever required and decode the logical qubit if desired. Block I is also used to reset the qubits at any instant of time by inputting $|0\rangle$'s. Block II is the computer where desired quantum computations are performed. We use the term "computer" for that part of the architecture where fault-tolerant quantum computations are performed on the encoded qubits. The size of Block II is determined by the number of qubits that are involved in the computation.

Initially, all the qubits are reset to the $|0\rangle$ state. Qubits within a block interact with each other through the couplings ξ_1 and ξ_2, represented as single and double lines, respectively, in the

figure. The couplings alternate along a linear array of qubits and are assumed to be fixed during fabrication.

To input any arbitrary state in Block I, input/output qubit I/O_1 or I/O_2 is prepared in the desired state. Suppose input/output qubit I/O_2 is prepared in some arbitrary single qubit state. The qubit is next encoded into a 7-qubit logical qubit using the Steane code. The encoding circuit is shown in Fig 2. The seven qubits comprising the logical qubit are F_1, F_2, F_3, F_4, F_5, F_6 and F_7 along line F (Fig 1). Observe that ancilla qubits are used to separate qubits F_1 through F_7 (ancillas are represented with a $|0\rangle$ within the circle). The purpose of ancillas between the encoded qubits is to facilitate gate operations on a logical qubit which are used to move the logical qubit from one block to another in the architecture using the bit-wise SWAP operations. The evolution of a qubit under a gate operation is affected by the states of the qubits coupled to it through the coupling parameters ξ_1 and ξ_2. Therefore, in performing two-qubit controlled operations, we cannot have two qubits in arbitrary states coupled to the target through identical couplings. This is because the target will not be able to distinguish the control qubit from the other [1, 29, 30]. Hence, ancillas are used to separate the qubits forming the encoded qubits along lines A and F. After encoding, if there are no errors in the logical qubit, it is moved to Block II, the computer, where the desired quantum computations are performed directly on it in a fault-tolerant way. The size of this block is determined by the number of qubits that are involved in the computation. If an arbitrary single-qubit error is introduced during computation on a logical qubit, it can be corrected by moving the logical qubit to Block I and performing error correction using the procedure shown in Fig 3. Note that an error introduced in the logical qubit during encoding can also be corrected using this circuit. After the computation, the logical qubits are decoded, if required, and the decoded state is available for read out as qubits I/O_1 or I/O_2.

At any instant of time, to reset the qubits in Block I, lines A and F are used. For this purpose, using either line, the qubit I/O_1 or I/O_2 is successively prepared in the $|0\rangle$ state which is move down along the line of qubits using SWAP operations until all the qubits along the line are in the $|0\rangle$ state.

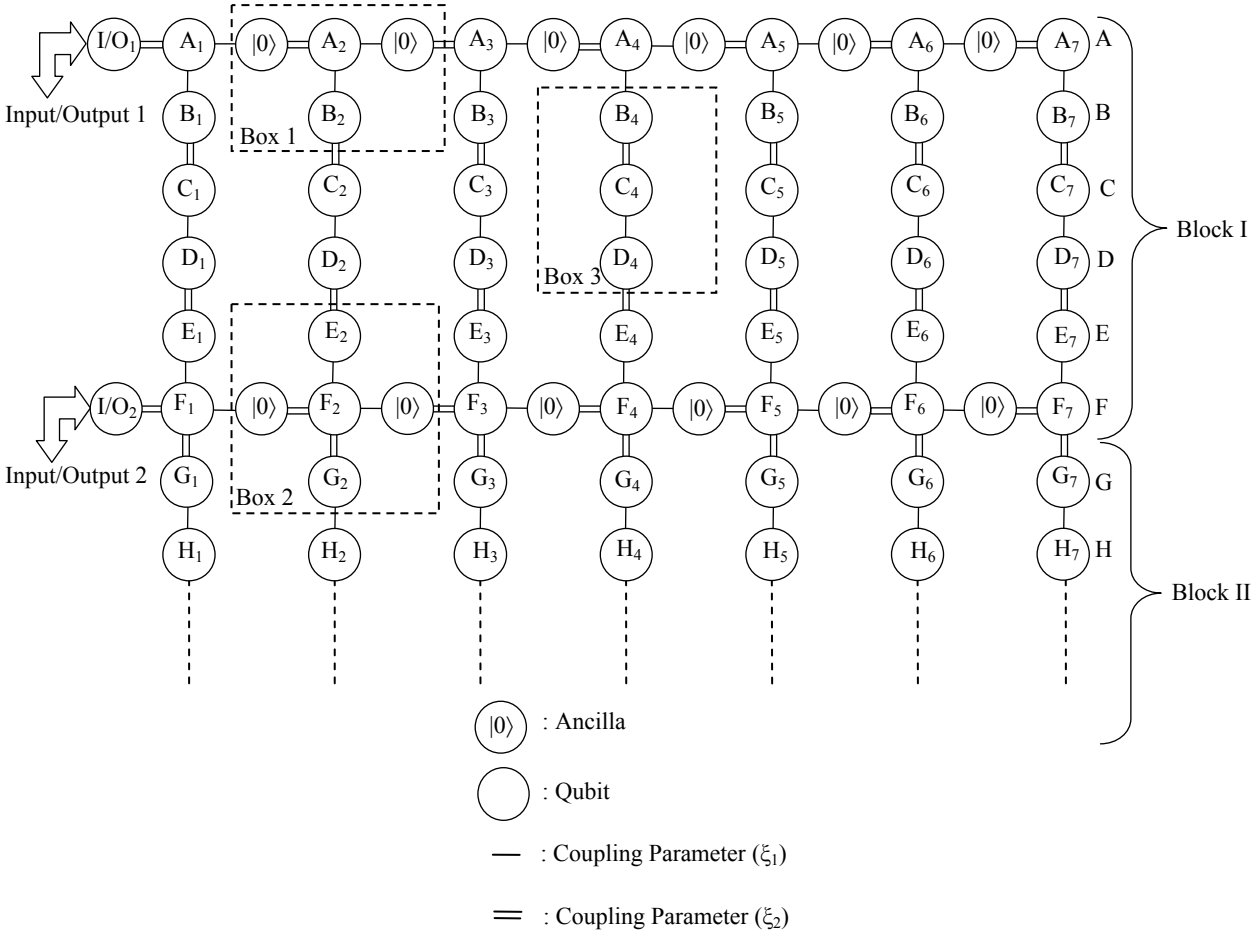

FIG 1: Two-dimensional architecture showing the layout of qubits for fault-tolerant computation. We assume only nearest neighbor interaction between the qubits. Each qubit is represented by a circle. Qubits within a block interact with each other through the couplings ξ_1 and ξ_2, represented as single and double lines, respectively. The couplings alternate along a linear array of qubits and are assumed to be fixed during fabrication. The size of Block II is determined by the number of qubits that are involved in the computation.

2.5 Encoding, Fault-Tolerant Quantum Computation and Decoding

The architecture uses the 7-qubit Steane code for encoding which can correct an arbitrary single-qubit error. The 7-qubit code encodes a single qubit to a logical qubit as follows [7]:

$$|0_L\rangle = \frac{1}{\sqrt{8}}\begin{pmatrix} |0000000\rangle + |1010101\rangle + |0110011\rangle \\ + |1100110\rangle + |0001111\rangle + |1011010\rangle \\ + |0111100\rangle + |1101001\rangle \end{pmatrix}$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}\begin{pmatrix} |1111111\rangle + |0101010\rangle + |1001100\rangle \\ + |0011001\rangle + |1110000\rangle + |0100101\rangle \\ + |1000011\rangle + |0010110\rangle \end{pmatrix} \quad (1)$$

Each encoded block of qubits comprises 7 qubits. The encoded zero state, $|0_L\rangle$, is the superposition of the even weight codewords from the classical 7-bit Hamming code. The encoded one state, $|1_L\rangle$, is the superposition of the odd weight codewords from the classical Hamming code (The weight of a codeword is the number of 1's in it).

Initially, all the qubits in the architecture are reset to the $|0\rangle$ state. Next, input/output qubit I/O_2 is prepared in any desired arbitrary state $\alpha|0\rangle+\beta|1\rangle$. The qubit is then encoded to the logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ along line F. The sequence of gate operations to encode the qubit is shown in Fig 2. Observe that ancilla qubits and the 7 qubits forming the logical qubit alternate along line F.

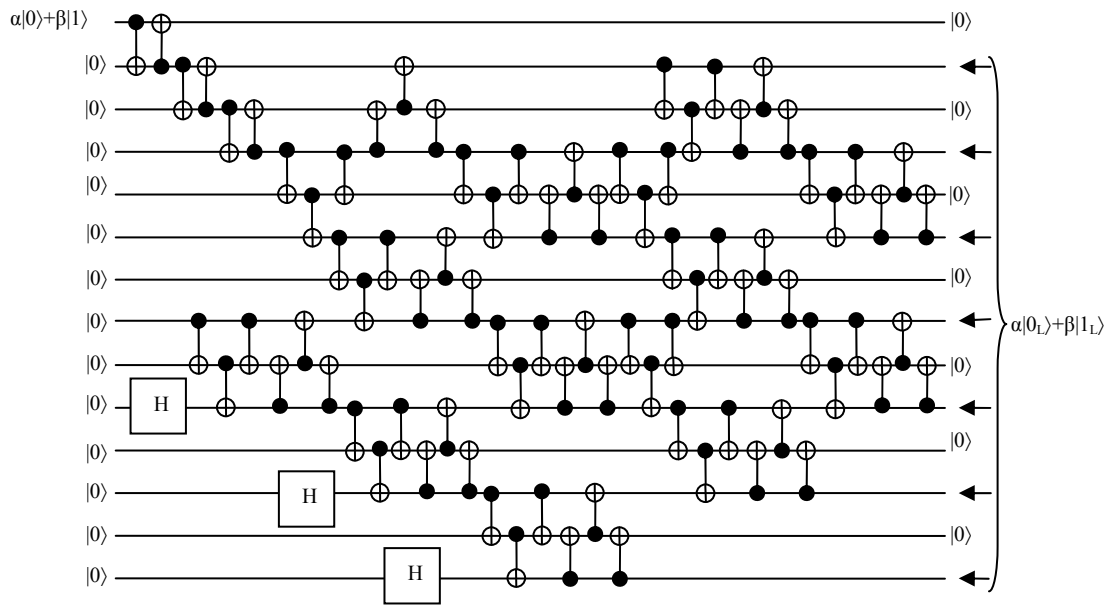

FIG 2: Non Fault-tolerant 7-qubit Steane code encoding circuit for our architecture. It takes an arbitrary state $\alpha|0\rangle+\beta|1\rangle$ and produces a logical state $\alpha|0_L\rangle+\beta|1_L\rangle$ in which each logical qubit is in between the ancillas.

The 7-qubit code is particularly convenient for fault-tolerant computations because a number of gates can be performed transversally, i.e., the i^{th} qubit in an encoded block interacts either with itself or the i^{th} qubit of another block. For universal quantum computing, it is sufficient to implement gates form a universal gate set. One such gate set comprises the CNOT, Hadamard, Phase and T gates [2]. In other words, any unitary operation within a quantum computer can be implemented using these gates. Therefore, to achieve fault-tolerant quantum computation using our architecture, it is sufficient to implement the logical CNOT, Hadamard, Phase and T gates.

The quantum gate operations performed in the architecture is accomplished by using the pulsed bias scheme [1, 29, 30], where the bias on the target qubit is pulsed low to a certain value for a chosen time duration. The only control parameter is the bias acting on the individual qubit. As previously mentioned, the coupling between qubits and the tunneling parameters of

individual qubits are treated as fixed parameters of the system. It is important to point that the same gate operation on a target requires pulses of different amplitudes depending on the way the qubit is interacting with its neighboring qubits.

Any computation will be one of three different cases in the architecture, each marked with a dashed box in Fig 1. The bias pulses for a particular gate operation for each of the three different cases are different. For instance, the pulses that are required to perform the CNOT and the Hadamard gate operations are given in Table 1 for the three different cases. Details for calculating these pulses have been presented elsewhere [1]. In the table, C and T stand for the control and the target qubits, respectively, Δ is the tunneling parameter, ε_T is the bias on the target, ξ_1 and ξ_2 are the couplings. The time duration for which these pulses are applied on the target qubit is $1/4\Delta$ [1].

Table 1: The table contains the pulses that we require to perform CNOT and Hadamard gate in the architecture. In the table, C and T stand for Control and Target qubit respectively. The term Δ is the tunneling parameter.

Box	**CNOT**			**Hadamard**	
	C	**T**	**Pulses**	**Qubit**	**Pulses**
1	B_2	A_2	$\varepsilon_T=-\xi_2$	A_2	$\varepsilon_T=\Delta+\xi_2$ $\varepsilon_T=\Delta+2\xi_1+\xi_2$
2	E_2	F_2	$\varepsilon_T=0$ $\varepsilon_T=-2\xi_2$	F_2	$\varepsilon_T=\Delta+2\xi_2$, $\varepsilon_T=\Delta$ $\varepsilon_T=\Delta+2\xi_1+2\xi_2$, $\varepsilon_T=\Delta+2\xi_1$
3	D_4	C_4	$\varepsilon_T=\xi_1+\xi_2$ $\varepsilon_T=\xi_1-\xi_2$	C_4	$\varepsilon_T=\Delta-\xi_1+\xi_2$, $\varepsilon_T=\Delta-\xi_1-\xi_2$ $\varepsilon_T=\Delta+\xi_1+\xi_2$, $\varepsilon_T=\Delta+\xi_1-\xi_2$

These pulses for different gate operation in the three different cases constitute the magnitude of the bias pulse on the target qubit performing the corresponding gate operations in the architecture.

Since Toffoli gate operation is only performed in case of Box 3 in our design, to perform this gate operation between qubits B_4 (control), C_4 (target) and D_4 (control), the target qubit C_4 is pulsed with the pulse $\varepsilon_T=\xi_1+\xi_2$ [1].

At any time during computation, if a single qubit arbitrary error occurs in an encoded qubit, the encoded logical qubit can be corrected by moving it to Block I and following the error correction circuit described in the next section.

After the computation, if no errors occur in a logical qubit, if the physical state is desired, the logical qubit is moved to line F of Block I, where decoding is performed. The decoding circuit is the encoding circuit run in reverse. It converts the logical qubit to the corresponding physical qubit. The input/output qubit I/O_2 acquires the decoded state which is available for read out.

2.6 Fault-Tolerant Error Correction

The 7-qubit code only protects against an arbitrary single-qubit error. A single-qubit arbitrary error can be introduced at any instant of time either during encoding or computation. In quantum computing, an arbitrary single-qubit error can be represented as a linear combination of the pauli matrices X, Y and Z along with the identity [31, 32]. Here, X represents a bit flip error, Z a phase flip error and Y a combination of a bit flip and a phase flip error.

Suppose we are required to perform error correction on the encoded qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ along line F. As a first step, the logical qubit is moved to line C through SWAP operations which are performed transversally, i.e., bit-wise. Next, using qubits I/O_1 and I/O_2, which are prepared in the $|0\rangle$ states, the qubits along lines A and F are each encoded to the logical zero state, $|0_L\rangle$ (equation (1)). This being done, the circuit [3, 7, 33] shown in Fig 3 is employed for performing fault-tolerant error correction on the encoded qubit, $\alpha|0_L\rangle+\beta|1_L\rangle$, along line C.

The error correction circuit has been divided into two parts, each marked by dashed lines in Fig 3. The first block corrects for a single-qubit bit-flip error while the second block corrects for a single-qubit phase-error on the logical qubit, respectively. Each gate operation on the

logical qubits in Fig 3 is implemented bitwise [2], i.e., for instance, the logical Hadamard gate is implemented by simply performing the Hadamard gate in parallel on each of the physical qubits comprising the encoded qubit. The controlled-controlled Z gate operation in the phase-error correction block in Fig 3 is also implemented transversally. Here, the gate operation involves three encoded qubits, two of which are controls. The gate can be achieved by performing Hadamard, Toffoli and Hadamard gate operation in that order on the target qubit. Methods for implementing these gates have been discussed in Section 2.5.

Figure 4 shows the internal circuitry for the error detection/correction blocks 1 and 4 in Fig 3. Observe that the qubits in the figure are numbered, i.e., the first qubit of line F is labeled F (1), the second qubit F (2) and so on. Qubits a_1, a_2, a_3 and s_1 are used to hold error syndromes, which are used to detect and correct a single-qubit error. Parts of the circuit that repeat themselves have been further grouped as blocks, M1 through M6. The gate operations comprising each of these blocks have been shown in Fig 4. The internal circuitry for the error detection/correction blocks 2 and 3 are the same as that for blocks 1 and 4 (Fig 4), except that the circuit is applied on lines A, B and C instead of lines D, E and F (see Fig 3).

After the encoded qubit passes through the bit error correction block, the logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ is available along line D (recall that it was initially along line C). Before performing phase error correction, the logical zero qubits, $|0_L\rangle$'s along lines C and E, and error syndromes, a_1, a_2, a_3 and s_1, that were used in the bit error correction block are reset to the $|0\rangle$ state. To reset these qubits, first of all, these qubits are moved to lines A and F as shown in Fig 3. Next, the input/output qubits I/O_1 and I/O_2 are successively prepared in the $|0\rangle$ state that are moved down along the lines, until all the qubits along lines A and F are in the $|0\rangle$ state. Once these lines are reset, the qubits long them are encoded to form logical zero qubits. This is because the phase

error correction block requires logical zero qubits and by preparing fresh encoded zero states, we are ensuring that corrupted logical zero qubits are not used in the phase error correction block.

The overall procedure is able to correct any arbitrary single-qubit error on the logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ fault-tolerantly. After the error correction procedure, the corrected logical qubit, $\alpha|0_L\rangle+\beta|1_L\rangle$, is along line C. It is important to point that, if the error correction circuit shown in Fig 3 was implemented in a one dimensional linear nearest neighbor architecture [3], several more gate operations would be required than that required in our architecture. This is because in a linear one-dimensional array, the encoded qubits are stacked successively along a line and therefore, to perform any two-qubit logical gate operation like the CNOT, 42 additional SWAP operations [3] are required in order to bring together the corresponding qubits of the two logical qubits. On the other hand, since the architecture shown in Fig 1 is a two-dimensional layout where the encoded qubits are stacked vertically, no additional SWAP operations are required to perform transversal two or three qubit gate operations between encoded qubits.

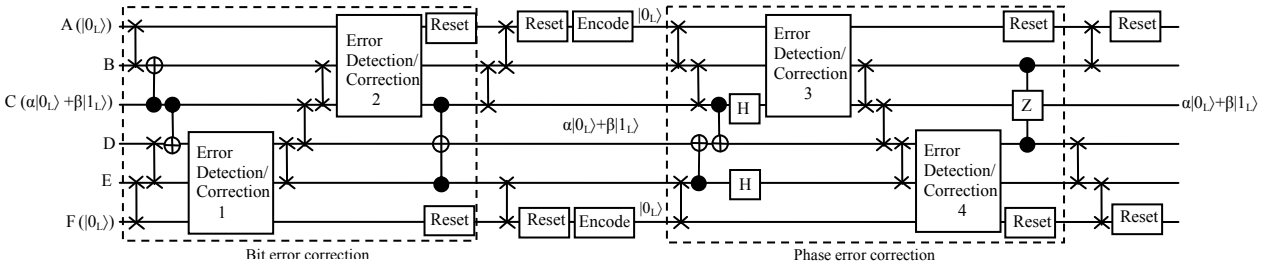

FIG 3: Fault-tolerant Error correction procedure used in the architecture. The figure describes the steps that are followed to correct any single arbitrary error fault-tolerantly.

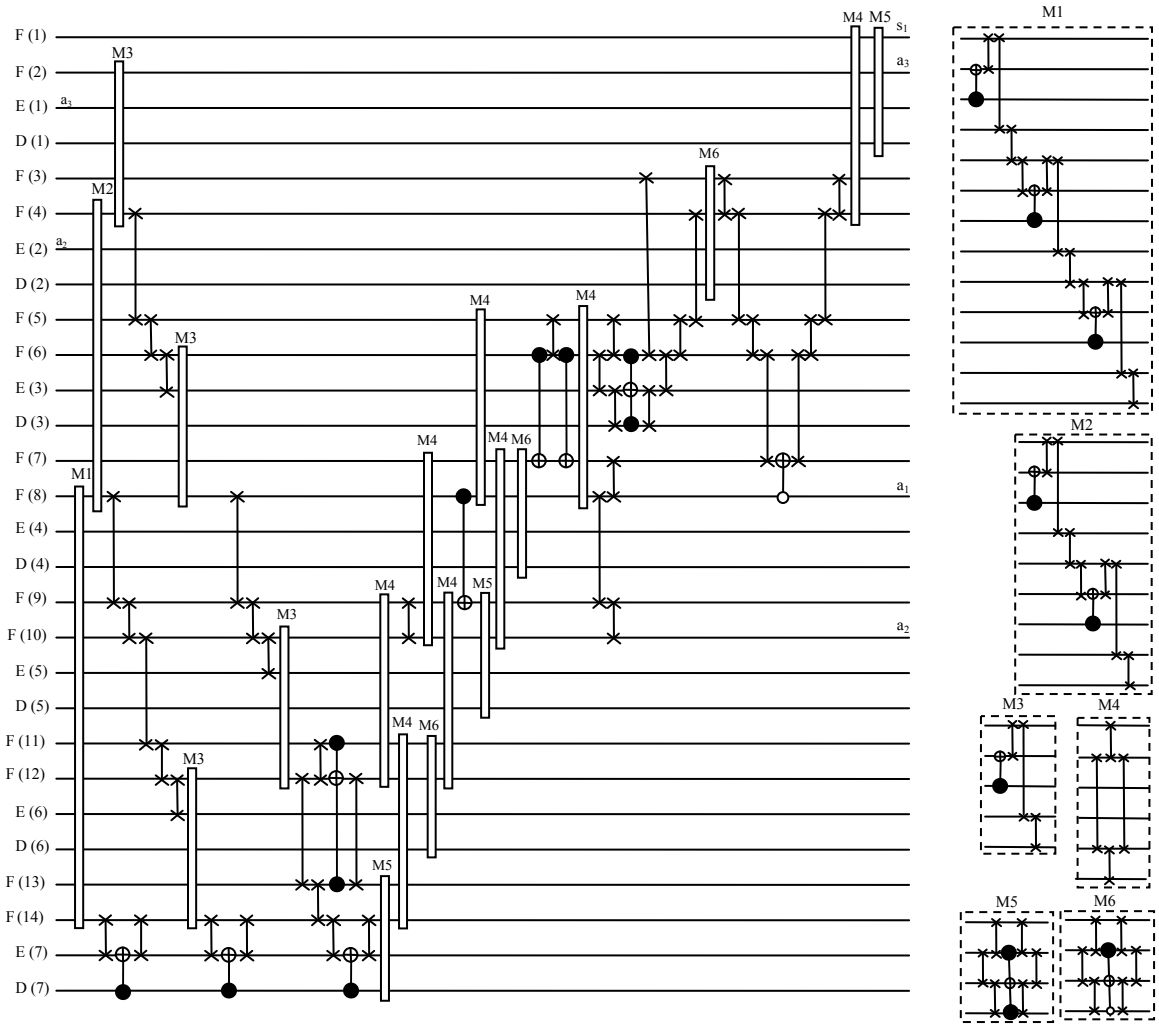

FIG 4: The equivalent error detection/correction block of Fig 3 that is used in our architecture. Observe that the qubits in the figure are numbered, i.e., the first qubit of line F is labeled F (1), the second qubit F (2) and so on. Parts of the circuit that repeat themselves have been further grouped as blocks, M1 through M6.

2.7 Conclusion

In this paper, we have shown a general approach for implementing fault-tolerant quantum computation in a linear nearest neighbor architecture. We present a two-dimensional architecture involving ancillas and show how the computation in performed fault tolerantly. In our discussion, we show the computations for the specific case of the 7-qubit Steane code. However, our method can be extended to any code where fault-tolerant gates can be implemented transversally. The gate operations are achieved by varying the biases on the individual qubits

forming a logical qubit. We also compare our architecture with a one-dimensional linear nearest neighbor architecture and show how our architecture is more efficient.

2.8 References

[1] P. Kumar and S. R. Skinner," Controlled-NOT Gate Operations in a Linear Nearest-Neighbor Arrays", the ninth international conference on Quantum Communication, Measurement and Computing (QCMC-2008) Calgary, Canada.

[2] A. G. Fowler," Constructing arbitrary single-qubit fault-tolerant", quant-ph/0411206v1 (2004).

[3] Thomas Szkopek, P.Oscar Boykin, Vwani P. Roychowdhury, Eli Tablonovitch, *Fellow, IEEE*, Geoffery Simms, Mark Gyure, and Bryan Fong," Threshold Error Penalty for Fault-Tolerant Quantum Computation with nearest Neighbor Communication", IEEE Transactions on Nanotechnology, Vol. 5, No. 1, January 2006.

[4] A. G. Fowler, C. D. Hill and L. C. L. Hollenberg," Quantum Error Correction on Linear Nearest Neighbor Qubit Arrays", quant-ph/0311116v1 (2003).

[5] A. G. Fowler, S. J. Devitt and L. C. L. Hollenberg," Implementation of Shor's Algorithm on a Linear Nearest Neighbor Qubit Array", quant-ph/0402196v1 (2004).

[6] S. A. Kutin," Shor's Algorithm on a Nearest-Neighbbor Machine", quant-ph/0609001v1 (2006).

[7] John Preskill," Fault-Tolerant Quantum Computation", quant-ph/9712048 (1997).

[8] Daniel Gottesman," Quantum Error Correction and Fault-Tolerance", quant-ph/0507174 (2005).

[9] A. M. Steane," Active stabilization, quantum computation and quantum state synthesis", quant-ph/9611027v1 (1996).

[10] W. G. Unruh, Phys. Rev. A **51**, 992 (1995).

[11] G. M. Palma, K.-A. Suominen, and A. K. Ekert, Proc. R. Soc. London A **452**, 567 (1996).

[12] M. A. Nielson and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, Cambridge, England, 2001).

[13] G. Chen, D. A. Church, B. G. Englert, C. Henkel, B. Rohwedder, M. O. Scully, M. S. Zubairy, Quantum computing Devices (2007).

[14] D. Gottesman, "An Introduction to Quantum Error Correction," in *Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millenium*, ed. S. J. Lomonaco, Jr., pp. 221-235 (American Mathematical Society, Providence, Phode Island, 2002), quant-ph/004072.

[15] D. Gottesman, quant-ph/0507174 (2005).

[16] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).

[17] J. Preskill, Proc. R. Soc.London A **454**, 385 (1998).

[18] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).

[19] E. M. Rains *et al.*, Phys. Rev. Lett **79**, 953 (1997).

[20] A. R. Calderblank *et al.*, IEEE Trans. Inform. Theory **44**, 1369 (1998).

[21] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).

[22] A. R. Calderblank *et al.*, Phys. Rev. Lett **78**, 405 (1997).

[23] A. R. Calderblank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).

[24] E. Knill and R. Laflamme, Phys. Rev. A **55**, 900 (1997).

[25] A. M. Steane, Proc. R. Soc.London A **452**, 2551 (1996).

[26] A. M. Steane, Nature (London) **399**, 124 (1999).

[27] R. Laflamme *et al.*, Phys. Rev. Lett. **77**, 198 (1996).

[28] J. Niwa, K. Matsumoto, H. Imai," Simulating the Effects of Quantum Error-Correction Schemes", quant-ph/0211071v1 (2002).

[29] P. K. Gagnebin *et al.*, Phys. Rev. A **72**, 042311 (2005).

[30] P. Kumar and S. R. Skinner, "Simplified Approach to Implementing Controlled Unitary Operations in a Two-Qubit System", submitted to Phys. Rev. A.

[31] Ashley M.Stephens, Austin G. Fowler and Lloyd C.L.Hollenberg," Universal fault tolerant quantum computation in bilinear nearest neighbor arrays", quant-ph/0702201(2008).

[32] David P. DiVincenzo and Peter W. Shor," Fault-Tolerant Error Correction with efficient Quantum codes", quant-ph/9605031(1996).

[33] P. J. Salas and A. L. Sanz," Effect of ancilla's structure on quantum error correction using the 7-qubit Calderbank-Shor-Steane code", quant-ph/0405012 (2004).

CHAPTER 3

CONCLUSION

We have shown a general approach for implementing fault-tolerant quantum computation using a two-dimensional. The idea of fault-tolerant quantum computation is to perform the gate operations directly on the encoded qubits in a transversal way so that decoding is never required. Thus, a single bad gate anywhere in the system can only spread to produce one error per block of the code which prevents overwhelming the error tolerance of any single block.

Initially, we designed an architecture for performing fault-tolerant computation with non-fault-tolerant error correction, which have been presented in Appendix A. However, due to non fault-tolerant error correction approaches the correction circuit itself propagates error from the error syndromes (i.e. ancillas) to the logical qubits. Therefore, the architecture was modified to one that could correct error fault tolerantly.

We showed the quantum gate operations performed in our architecture are achieved by using the pulsed bias scheme where the bias of the target qubit is pulsed low to a certain value for a chosen time duration. Throughout the work, we showed the computation for the specific case of the 7-qubit Steane code. However, our method can be extended to any code where fault-tolerant gates can be implemented transversally. We also compared our architecture to an existing architecture for fault-tolerant computation employing a linear one-dimensional nearest neighbor array of qubits and show how ours is more efficient.

REFERENCES

LIST OF REFERENCES

[34] N.-J. Wu *et al.*, quant-ph/9912036 (1999).

[35] R. Vrijen *et al.*, Phys. Rev. A **62**, 012306 (2000).

[36] B. Golding and M. I. Dykman, cond-mat/0309147 (2003).

[37] E. Novais and A. H. C. Neto, cond-mat/0308475 (2003).

[38] L. C. L. Hollenberg *et al.*, cond-mat/0306235 (2003).

[39] L. Tian and P. Zoller, quant-ph/0306085 (2003).

[40] K. Yang *et al.*, Chinese Phys. Lett **20**, 991 (2003).

[41] M. Feng *et al.*, quant-ph/0304169 (2003).

[42] J. K. Pachos and P. L. Knight, Phys. Rev. Lett. **91**, 107902 (2003).

[43] M. Friesen *et al.*, Phys. Rev. B **67,** 121301(R) (2003).

[44] P. Solinas *et al.*, Phys. Rev. B **67**, 121307(R) (2003).

[45] J. H. Jefferson *et al.*, Phys. Rev. A **66**, 042328 (2002).

[46] T. D. Ladd *et al.*, Phys. Rev. Lett. **89**, 017901 (2002).

[47] V. N. Golovach and D. Loss, Semicond. Sci. Tech. **17**, 355 (2002).

APPENDICES

Appendix A

Fault-Tolerant Gates using 7-Qubit Steane Code

CNOT Gate

The logical CNOT can be implemented transversally by performing CNOT gate operations bitwise between each of the qubits in the control encoded block and the corresponding qubits in the target encoded block. Figure 5 shows the circuit for realizing a fault-tolerant CNOT gate.

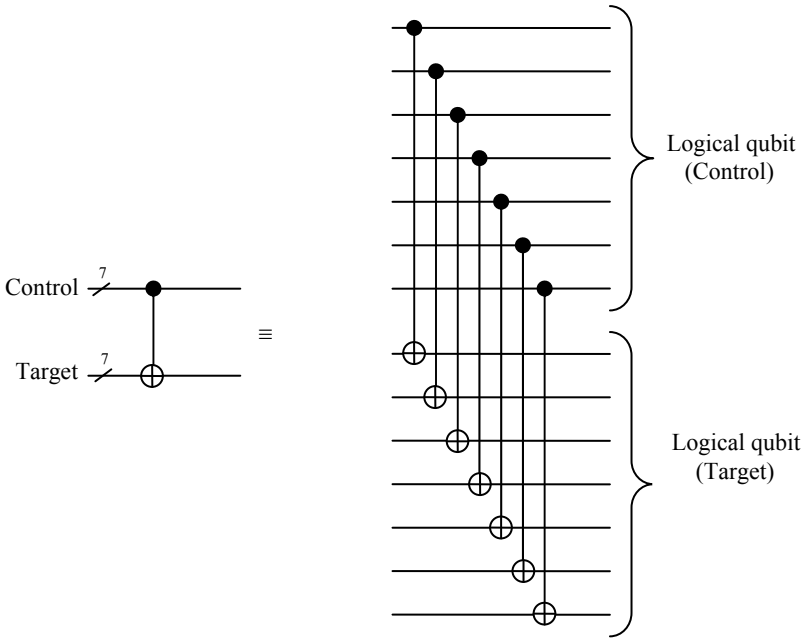

FIG 5: Implementation of fault-tolerant CNOT gate operation between qubit blocks using the Steane code. The gate operation is realized by performing CNOT gate operations bitwise between corresponding qubits in the two different blocks.

Hadamard and Phase Gate

The logical Hadamard transform can be implemented by simply performing the Hadamard transform in parallel on each of the physical qubits. The circuits for realizing fault-tolerant Hadamard and Phase gates for the Steane code are shown in Figures 6 and 7, respectively. This kind of an implementation is called bitwise since we can simply apply the desired operation individually on every qubit of the code block.

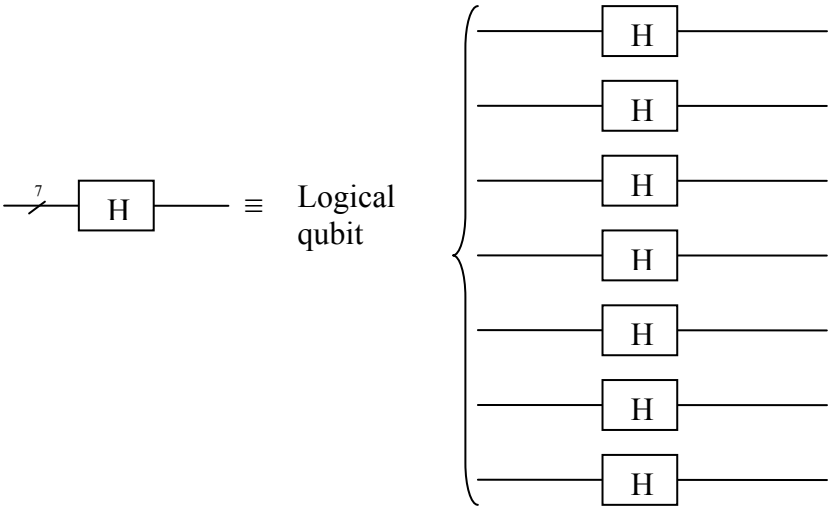

FIG 6: Circuits for implementing fault-tolerant Hadamard gates for Steane code. This kind of an implementation is called bitwise since we can simply apply the desired operation on every qubit of a code block.

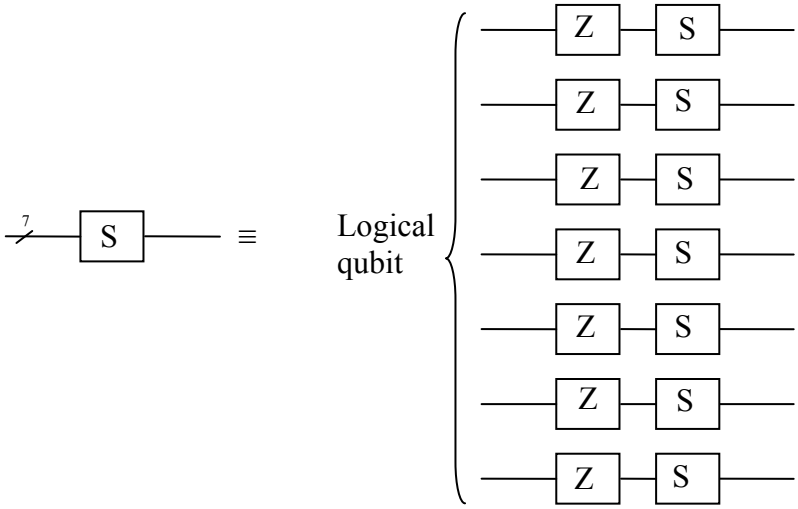

FIG 7: Circuit for implementing fault-tolerant phase gate for Steane code.

T (or π/8) Gate

To form a complete set of universal gates, we need a non -transversal gate like the π/8 or T gate which is much more complex to implement fault-tolerantly. Figure 8 shows the circuit for implementing a fault-tolerant π/8 gate. From the figure we can see that we require two blocks of qubits – one comprises a block of ancilla qubits and the other an encoded block of data qubits. To implement the fault-tolerant π/8 gate, we first perform H and T gates on the qubits forming the ancilla block. Next a fault-tolerant CNOT gate operation is performed with the ancilla block as the control logical qubits. Next, the state of the originally encoded qubits is measured (the rectangle with the arrow in the circuit denotes a measurement). Recall that the double line denotes a classical bit. If we get a 0, no operation is performed on the ancilla block. However, if we get a 1, we apply the S and X gates to the original ancilla qubits which brings it to the state T $(\alpha|0_L\rangle + \beta|1_L\rangle)$. (Note that the X gate is the Pauli spin operator $\boldsymbol{\sigma_X}$, which corresponds to a NOT gate operation, which can be performed transversally).

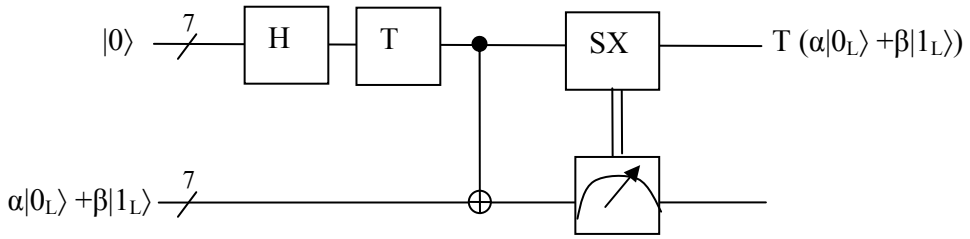

FIG 8: Circuit for implementing a fault-tolerant $\pi/8$ gate. There are two blocks of qubits each comprising of seven physical qubits – one block comprises a block of ancilla qubits and the other a block of encoded data. The rectangle block with the arrow denotes a measurement while the double line denotes a classical bit.

Appendix B

General Logical Linear Nearest Neighbor (LLNN) Architecture for Fault-Tolerant Computation with non Fault-Tolerant Error Correction

Figure 9 shows an architecture for fault-tolerant quantum computation that is able to correct any single-qubit arbitrary error non fault tolerantly. The architectural layout only assumes nearest-neighbor interactions between qubits. Each qubit is represented by a circle. It is also assumed that the qubits in the design are identical in that they have the same tunneling parameter which is again fixed during fabrication. Initially, all the qubits are reset to the $|0\rangle$ state. Qubits within a block interact with each other through the couplings ξ_1 and ξ_2, represented as single and double lines, respectively, in the figure. The couplings alternate along a linear array of qubits and are assumed to be fixed during fabrication.

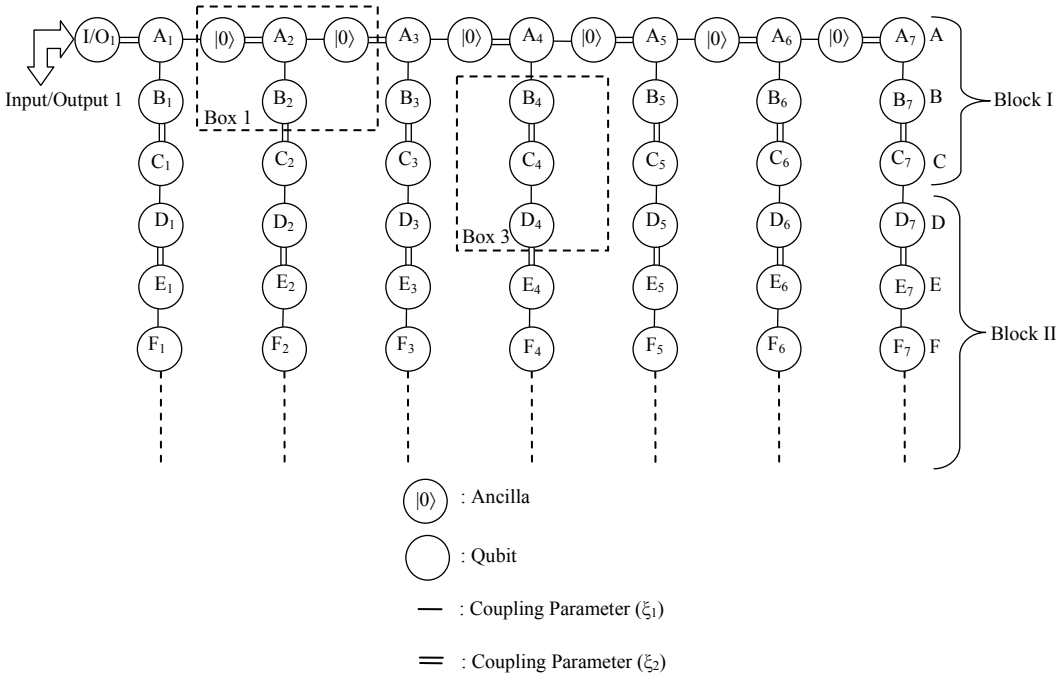

FIG 9: Two-dimensional architecture showing the layout of qubits for fault-tolerant computation with non-fault-tolerant error correction. Qubits within a block interact with each other through the couplings ξ_1 and ξ_2, represented as single and double lines, respectively, in the Figure. The size of Block B is determined by the number of qubits that are involved in the computation.

The entire architecture is divided into 2 Blocks: Block I and Block II. Block I is used to input arbitrary single qubit states, encode the qubits to corresponding logical qubit, correct the logical qubits *non fault tolerantly* whenever required and decode the logical qubit if desired. To input any arbitrary state in block I, input/output qubit I/O_1 is prepared in the desired state and the qubit is encoded to the corresponding logical qubit along line A. In Fig 9, the 7-qubits comprising the logical qubit alternate with ancilla qubits (ancillas are represented with a $|0\rangle$ within a circle). Any single-qubit arbitrary error that is either introduced by encoding or during quantum computation is corrected by a non fault-tolerant error correction procedure. At any instant of time, to reset the qubits in Block I, line A is used. For this purpose, input/output qubit I/O_1 is successively prepared in the $|0\rangle$ state, which is then moved down along the line of qubits using SWAP operations, until all the qubits along the line are in the $|0\rangle$ state. The encoded logical qubit is then moved to Block II which is the computer. Any required quantum computations are performed directly on the logical qubit, fault-tolerantly, in this block. During computation, the logical qubit can be moved to Block I for error correction, if a single-qubit error occurs on any one of the 7 qubits forming the encoded qubit.

The pulses that are required to perform gate operations in the architecture are given in Table 1 (Chapter 2, Section 2.5, page 12). The table shows the pulses required to perform CNOT and Hadamad gates for Box 1 and Box 3 in Fig 9.

Encoding, Fault-Tolerant Quantum Computation and Decoding

Originally all the qubits in the architecture are assumed to be in the |0⟩ state. Next, input/output qubit I_1 is prepared in any desired arbitrary state α|0⟩+β|1⟩. The qubit is then encoded to a logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ along line A. The sequence of gate operations is shown in Fig 2 (Chapter 2, Section 2.5, page 11). The qubits A_1, A_2, A_3, A_4, A_5, A_6 and A_7 comprises the logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$.

Fault-tolerant quantum computation is performed in Block II in a similar way as it is performed in the architecture shown in Fig 1. At any instant of time during computation, the logical qubit can be moved to Block I for error correction.

If the physical qubit state is desired, the logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ is brought to line A of Block I. The decoding operation is carried out as shown in Fig 10. The input/output qubit I/O_1 acquires the decoded state that is available for read out.

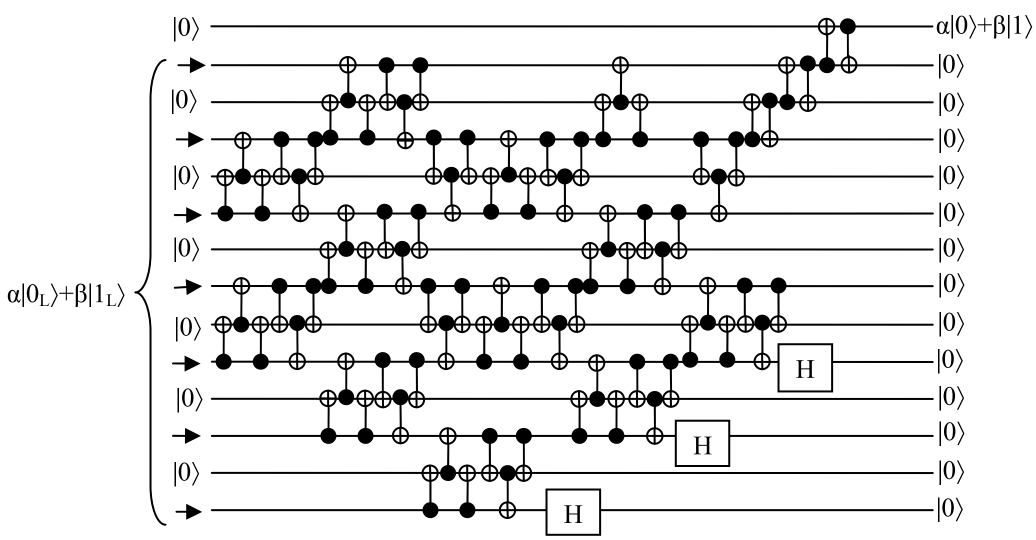

FIG 10: The decoding circuit that is used in our architecture to decode the 7-qubit Steane code to the respective physical state. Input/output qubit I_1 acquires the decoded state that is read out.

Non Fault-Tolerant Error Correction

To perform error correction in the architecture (Figure 5), as a first step, the encoded logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ from line A is moved to line C via logical SWAP operations. Each logical SWAP operation between two encoded qubits is performed transversally, i.e., bit-wise. Next, error correction is performed following the gate operations shown in Fig 4 (Chapter 2, Section 2.6, page 16) between the qubits that are along lines A, B and C instead of lines D, E and F. The circuit is able to correct a single qubit error occuring in the logical qubit. After the bit-flip error correction, the error syndromes a_1, a_2, a_3 and s_1 that are along line A are reset. Next, a logical Hadamard gate operation is performed on the logical qubit $\alpha|0_L\rangle+\beta|1_L\rangle$ that is along line C to convert the phase flip error on the logical qubit to the corresponding bit flip error. Next, the circuit shown in Fig 4 is performed between the qubits that are along lines A, B and C. The logical Hadamard gate operations are again carried out on the logical qubit to bring back their states. The error syndromes a_1, a_2, a_3 and s_1 that are along line A are again reset.

Drawback:

A non-fault-tolerant error correction circuit itself can easily propagate error from error syndromes (ancillas) to the logical qubit. The problem is that a single ancilla, say a_1, is used as a target for four successive CNOT gates. If a single phase error occurs in the ancilla qubit at some stage that error can feed back to two or more of the qubits in the data block. This results in propagation and accumulation of errors.

Appendix C

7-Qubit Steane Code Error Detection and Correction

The general circuit for error detection/correction is shown in Fig 11.

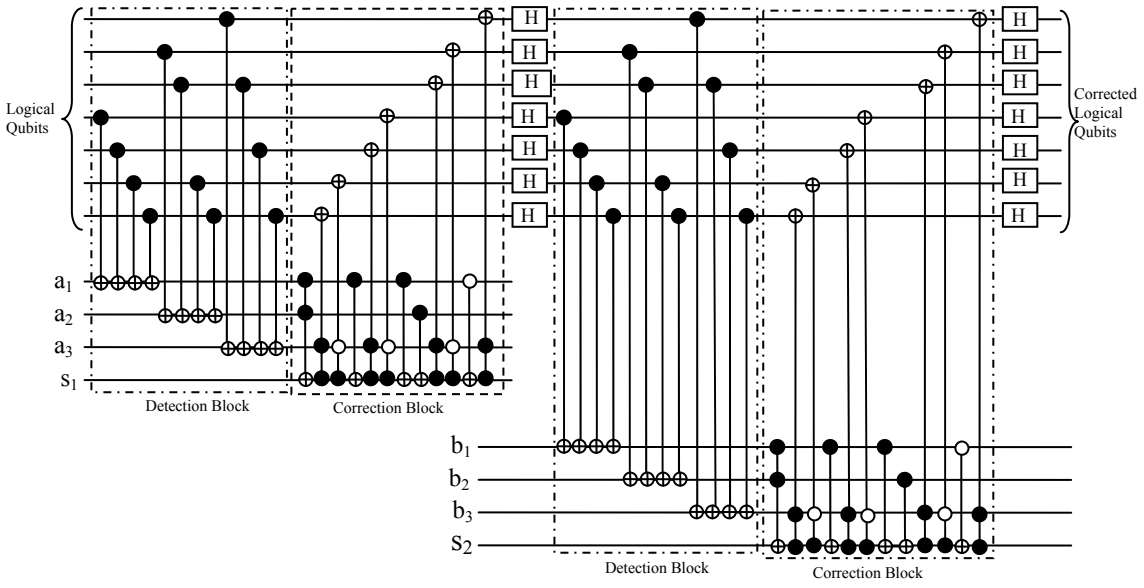

FIG 11: The Error-Detection/Correction circuit that detects and corrects a single bit error occurred in encoded qubits. The additional ancillas (or error syndromes) a_1, a_2 and a_3 are used to detect and an s_1 is to correct any single arbitrary bit flip error. Under the logical Hadamard gate operation, the phase flip error is transformed to a bit flip error. Another set of error syndromes b_1, b_2, b_3 and s_1 are used to detect and correct the bit flip error. Again the logical Hadamard gate is applied to get back the logical sates. The overall procedure is able to correct any single arbitrary error occurred in the logical qubits. The dark dot and vacate dot represent the control one and the control zero, respectively.

The additional ancillas a_1, a_2, a_3 are used to detect and an ancilla s_1 is used to correct a single bit flip error occurring on the logical qubit without measuring the ancillas. The Hadamard gates are applied on each logical qubit to convert a phase flip error to a bit flip error. Another set of error syndromes, b_1, b_2, b_3 and s_2, are used to detect and correct the corresponding single bit flip error. The Hadamard gates are again applied on the logical qubit to bring back their states. The whole procedure corrects any arbitrary single-qubit error occurring on a logical qubit.

Appendix D

Single Qubit System

The time evolution of a closed quantum system is described by the Schrödinger equation,

$$i\hbar \frac{\partial |\Psi(t)\rangle}{\partial t} = \mathbf{H}|\Psi(t)\rangle \ , \tag{D-1}$$

where $\hbar$ is the Planck's constant, h, divided by 2π and **H** is a Hermitian operator known as Hamiltonian of the system. It represents the total energy function of the system. Since the qubit is represented by a column vector, the Hamiltonian can be represented by a 2×2 Hermitian (self-adjoint) matrix.

Equation (D-1) can be written as,

$$\frac{\partial |\Psi(t)\rangle}{|\Psi(t)\rangle} = \frac{\mathbf{H}}{i\hbar}\partial t \tag{D-2}$$

Further equation (D-2) can be written as,

$$\ln|\Psi(t)\rangle = \frac{\mathbf{H}}{i\hbar}t \tag{D-3}$$

This can be expressed as the formal solution to the Schrödinger wave equation that gives us the state of the system as a function of time as follows:

$$|\psi(t)\rangle = \exp\left(\frac{-i\mathbf{H}t}{\hbar}\right)|\psi(0)\rangle = U|\psi(0)\rangle \tag{D-4}$$

The most general form of the Hamiltonian of a single qubit system is given as

$$\mathbf{H} = \Delta\sigma_X + k\sigma_Y + \varepsilon\sigma_Z \tag{D-5}$$

Here, Δ, ε, k are parameters of the system under consideration. For superconducting qubits, Δ is the tunneling parameter and ε is the bias acting on the qubit, k=0 and σ_X, σ_Y and σ_Z are the Pauli matrices given by

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \text{ and } \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{D-6}$$

Therefore, equation (D-5) can be expanded as,

$$\mathbf{H} = \begin{pmatrix} \varepsilon & \Delta \\ \Delta & -\varepsilon \end{pmatrix} \tag{D-7}$$

From the equation (D-4), the unitary matrix will be,

$$U = \exp\left(-\frac{i\mathbf{H}t}{\hbar}\right) \tag{D-8}$$

Since, the exponential of any matrix A can be solved by using inverse of Laplace-Transforms, we have

$$\exp(At) = L^{-1}(sI - A)^{-1} \tag{D-9}$$

where, A is any matrix and I is an identity matrix.

Also,

$$\left.\begin{aligned} L^{-1}\left(\frac{s}{s^2 + w^2}\right) &= \cos(wt) \\ L^{-1}\left(\frac{w}{S^2 + w^2}\right) &= \sin(wt) \end{aligned}\right\} \tag{D-10}$$

Therefore, if

$$A = -\frac{i\mathbf{H}}{\hbar} = \begin{pmatrix} -\frac{i\varepsilon}{\hbar} & -\frac{i\Delta}{\hbar} \\ -\frac{i\Delta}{\hbar} & \frac{i\varepsilon}{\hbar} \end{pmatrix}, \tag{D-11}$$

we have,

$$U = \exp(At) = L^{-1}\left[(sI - A)^{-1}\right]$$

$$= L^{-1}\begin{pmatrix} \dfrac{s}{s^2 + \left(\dfrac{\sqrt{\Delta^2 + \varepsilon^2}}{\hbar}\right)^2} - \dfrac{\dfrac{i\varepsilon}{\hbar}}{s^2 + \left(\dfrac{\sqrt{\Delta^2 + \varepsilon^2}}{\hbar}\right)^2} & \dfrac{-\dfrac{i\Delta}{\hbar}}{s^2 + \left(\dfrac{\sqrt{\Delta^2 + \varepsilon^2}}{\hbar}\right)^2} \\ \dfrac{-\dfrac{i\Delta}{\hbar}}{s^2 + \left(\dfrac{\sqrt{\Delta^2 + \varepsilon^2}}{\hbar}\right)^2} & \dfrac{s}{s^2 + \left(\dfrac{\sqrt{\Delta^2 + \varepsilon^2}}{\hbar}\right)^2} + \dfrac{\dfrac{i\varepsilon}{\hbar}}{s^2 + \left(\dfrac{\sqrt{\Delta^2 + \varepsilon^2}}{\hbar}\right)^2} \end{pmatrix}$$

$$= L^{-1}\begin{pmatrix} \dfrac{s}{s^2 + w^2} - \left(\dfrac{i\varepsilon}{\hbar w}\right)\dfrac{w}{s^2 + w^2} & \left(-\dfrac{i\Delta}{\hbar w}\right)\dfrac{w}{s^2 + w^2} \\ \left(-\dfrac{i\Delta}{\hbar w}\right)\dfrac{w}{s^2 + w^2} & \dfrac{s}{s^2 + w^2} + \left(\dfrac{i\varepsilon}{\hbar w}\right)\dfrac{w}{s^2 + w^2} \end{pmatrix}$$

$$= \begin{pmatrix} \cos(wt) - \dfrac{i\varepsilon}{\hbar w}\sin(wt) & -\dfrac{i\Delta}{\hbar w}\cos(wt) \\ -\dfrac{i\Delta}{\hbar w}\cos(wt) & \cos(wt) + \dfrac{i\varepsilon}{\hbar w}\sin(wt) \end{pmatrix} \tag{D-12}$$

where, $w = \dfrac{\sqrt{\Delta^2 + \varepsilon^2}}{\hbar}$.

$$\left.\begin{aligned}&\text{since, } \left|\Psi(t)\right\rangle = \alpha(t) + \beta(t) = \begin{pmatrix}\alpha(t)\\ \beta(t)\end{pmatrix}\\ &\text{where } \left|\Psi(0)\right\rangle = \alpha(0) + \beta(0) = \begin{pmatrix}\alpha(0)\\ \beta(0)\end{pmatrix} = \begin{pmatrix}a+ib\\ c+id\end{pmatrix}\end{aligned}\right\} \tag{B-13}$$

where, a, b, c and d are real numbers and $a^2+b^2+c^2+d^2=1$, using equations (D-12) and (D-13) in equation (D-4):

$$\begin{aligned}\begin{pmatrix}\alpha(t)\\ \beta(t)\end{pmatrix} &= \begin{pmatrix}\cos(wt) - i\varepsilon\dfrac{\sin(wt)}{\hbar w} & -i\Delta\dfrac{\sin(wt)}{\hbar w}\\ -i\Delta\dfrac{\sin(wt)}{\hbar w} & \cos(wt) + i\varepsilon\dfrac{\sin(wt)}{\hbar w}\end{pmatrix}\begin{pmatrix}a+ib\\ c+id\end{pmatrix}\\ &= \begin{pmatrix}\left(\cos(wt) - i\varepsilon\dfrac{\sin(wt)}{\hbar w}\right)(a+ib) + \left(-i\Delta\dfrac{\sin(wt)}{\hbar w}\right)(c+id)\\ \left(-i\Delta\dfrac{\sin(wt)}{\hbar w}\right)(a+ib) + \left(\cos(wt) + i\varepsilon\dfrac{\sin(wt)}{\hbar w}\right)(c+id)\end{pmatrix}\end{aligned} \tag{D-14}$$

Therefore,

$$\begin{aligned}\alpha(t) = &\left(a\cos\frac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}t + (b\varepsilon + d\Delta)\frac{\sin\left(\dfrac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}\right)t}{\sqrt{\Delta^2+\varepsilon^2}}\right)\\ &+ i\left(b\cos\frac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}t + (-a\varepsilon - c\Delta)\frac{\sin\left(\dfrac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}\right)t}{\sqrt{\Delta^2+\varepsilon^2}}\right)\end{aligned} \tag{D-15}$$

$$\beta(t) = \left(c\cos\frac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}t + \left(-d\varepsilon + b\Delta\right)\frac{\sin\left(\frac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}\right)t}{\sqrt{\Delta^2+\varepsilon^2}} \right) + i\left(d\cos\frac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}t + \left(-a\Delta + c\varepsilon\right)\frac{\sin\left(\frac{\sqrt{\Delta^2+\varepsilon^2}}{\hbar}\right)t}{\sqrt{\Delta^2+\varepsilon^2}} \right) \quad \text{(D-16)}$$

Suppose the qubit starts initially in the $|1\rangle$ state, i.e., a=b=d=0, c=1.

$$\begin{pmatrix} a+ib \\ c+id \end{pmatrix} = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{(D-17)}$$

Therefore, equation (D-16) becomes,

$$\beta(t) = \cos(2\pi\sqrt{\Delta^2+\varepsilon^2})t + i\varepsilon\frac{\sin(2\pi\sqrt{\Delta^2+\varepsilon^2})t}{\sqrt{\Delta^2+\varepsilon^2}} \quad \text{(D-18)}$$

where we have normalized h to 1.Then an expression for the probability in the $|1\rangle$ state, $P_{|1\rangle}$, can be written as

$$P_{|1\rangle} = |\beta|^2 = \beta\beta^* = \frac{1}{2} + \frac{\varepsilon^2}{2\left(\Delta^2+\varepsilon^2\right)} + \frac{\Delta^2}{2\left(\Delta^2+\varepsilon^2\right)}\cos\left(2\pi.2\sqrt{\Delta^2+\varepsilon^2}\right)t \quad \text{(D-19)}$$

which is an oscillatory function as follows:

$$P_{|1\rangle}(t) = X + Y\cos(2\pi\pi ft - \theta) \quad \text{(D-20)}$$

where the offset X, the amplitude Y, the angular frequency f and the phase θ of probability oscillation are given as,

$$Offset,\ X = \frac{1}{2} \pm \frac{\varepsilon^2}{2\left(\Delta^2 + \varepsilon^2\right)} \tag{D-21}$$

$$Amplitude,\ Y = \pm \frac{\Delta^2}{2\left(\Delta^2 + \varepsilon^2\right)} \tag{D-22}$$

$$Frequency,\ f = 2\sqrt{\left(\Delta^2 + \varepsilon^2\right)} \tag{D-23}$$

$$Phase,\ \theta = 0 \tag{D-24}$$

Here, the "+" and "-" correspond to the cases when the qubit starts off initially in the $|0\rangle$ and $|1\rangle$ states, respectively.

Case I: Suppose the value of ε is increased such that it is much greater than the parameter Δ, i.e., $\varepsilon >> \Delta$. The offset, X, as given by equation (D-21), can be approximated as:

$$\text{Offset},\ X = \frac{1}{2} \pm \frac{\varepsilon^2}{2\left(\Delta^2 + \varepsilon^2\right)} \approx \frac{1}{2} \pm \frac{1}{2} \tag{D-25}$$

This is either close to 1 when the system starts out in the $|1\rangle$ state, or close to 0 when the system starts out in the $|0\rangle$ state. Next, under the effect of the large bias, the amplitude of oscillations will be:

$$\text{Amplitude},\ Y = \pm \frac{\Delta^2}{2\left(\Delta^2 + \varepsilon^2\right)} \approx 0 \tag{D-26}$$

Since the tunneling parameter, Δ, is very small compared to the bias parameter, ε, the amplitude of oscillations are of very small magnitude, i.e., negligible. Therefore, the expression for the probability function, $P_{|1\rangle}$ becomes:

$$P_{|1\rangle}(t) = \frac{1}{2} \pm \frac{1}{2} \pm \frac{\Delta^2}{2\left(\Delta^2 + \varepsilon^2\right)} \cos\left(4\pi\sqrt{\left(\Delta^2 + \varepsilon^2\right)}\ t\right) \approx \frac{1}{2} \pm \frac{1}{2} \tag{D-27}$$

This gives the value of 1 or 0 depending on whether the qubits starts off initially in the $|1\rangle$ or $|0\rangle$ states, respectively. This shows that by making ε large, the effect of the Δ parameter on the evolution of the qubit can be neglected, i.e., the qubit is forced to remain in its initialized state.

Case II: Suppose the bias parameter is made 0. For this case, the offset and the amplitude terms will be,

$$\text{Offset, } X = \frac{1}{2} \pm \frac{\varepsilon^2}{2\left(\Delta^2 + \varepsilon^2\right)} = \frac{1}{2} \tag{D-28}$$

$$\text{Amplitude, } Y = \pm \frac{\Delta^2 + k^2}{2\left(\Delta^2 + \varepsilon^2\right)} = \frac{1}{2} \tag{D-29}$$

Therefore, the probability function $P_{|1\rangle}$ is

$$P_{|1\rangle}(t) = \frac{1}{2} \pm \frac{1}{2} \cos\left(2\pi(2\Delta)\ t\right) \tag{D-30}$$

This is an oscillatory function with a frequency of oscillation 2Δ. Since the cosine function takes values between -1 and +1, the maximum and minimum values of $P_{|1\rangle}$ vary between 0 and 1. This means that the state of the qubit is switching between the two basis states at a frequency of oscillation 2Δ.

Appendix E

Coupled Two-Qubit System

The two-qubit coupled system consists of two qubits which interact with each other through an interaction Hamiltonian. The type of interaction depends upon the physical system under consideration. Since N qubits in a quantum computer represent 2^N dimensions in Hilbert space, therefore, the state space of the two qubit system is a four-dimensional Hilbert space. The basis that spans this space is obtained by taking the tensor product between the basis states of the two qubits which are:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \qquad \text{(E-1)}$$

An arbitrary state of the two-qubit coupled system will be

$$|\psi(t)\rangle = \alpha(t)|00\rangle + \beta(t)|01\rangle + \gamma(t)|10\rangle + \delta(t)|11\rangle \tag{E-2}$$

where, $|\alpha(t)|^2 + |\beta(t)|^2 + |\gamma(t)|^2 + |\delta(t)|^2 = 1.$

Here $\alpha(t)$, $\beta(t)$, $\gamma(t)$ and $\delta(t)$ are the probability amplitudes of the system in each of the four states, $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$, respectively.

As with a single qubit system, the time evolution of a two-qubit coupled system is governed by the Schrödinger wave equation and the state of the system at any time is:

$$|\psi(t)\rangle = \exp\left(\frac{-i\mathbf{H}t}{\hbar}\right)|\psi(0)\rangle = \mathbf{U}|\psi(0)\rangle, \tag{E-3}$$

Since $|\psi(0)\rangle$ and $|\psi(t)\rangle$ are 4×1 column vectors, $\mathbf{H}$, and therefore $\mathbf{U}$, are 4×4 matrices. The general form of the Hamiltonian of the coupled system of qubits A and B will be:

$$\mathbf{H} = \mathbf{H}_A + \mathbf{H}_B + \mathbf{H}_I, \tag{E-4}$$

where,

$$\mathbf{H}_A = \Delta_A\sigma_{XA} + k_A\sigma_{YA} + \varepsilon_A\sigma_{ZA}$$

$$\mathbf{H}_B = \Delta_B\sigma_{XB} + k_B\sigma_{YB} + \varepsilon_B\sigma_{ZB}$$

$$\mathbf{H}_I = J_X\sigma_{XA}\sigma_{XB} + J_Y\sigma_{YA}\sigma_{YB} + J_Z\sigma_{ZA}\sigma_{ZB}$$

Here, A and B represent the two qubits, $\mathbf{H}_A$ and $\mathbf{H}_B$ are the uncoupled Hamiltonians for qubits A and B, respectively, with parameters ε_A, Δ_A, k_A, ε_B, Δ_B and k_B. $\mathbf{H}_I$ is the interaction energy between the two qubits with coupling constants J_X, J_Y and J_Z. The matrices σ_{XA}, σ_{ZA}, σ_{XB} and σ_{ZB} are Kronecker products of the Pauli matrices with the identity matrix, which are calculated as follows:

$$\sigma_{XA} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

$$\sigma_{YA} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & -i \\ i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix},$$

$$\sigma_{ZA} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix},$$

$$\sigma_{XB} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\sigma_{YB} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{pmatrix},$$

$$\sigma_{ZB} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

(E-5)

The Hamiltonian **H** given by equation (E-4) can be written as

$$\mathbf{H} = \begin{pmatrix} \varepsilon_A + \varepsilon_B + J_Z & \Delta_B - ik_B & \Delta_A - ik_A & J_X - J_Y \\ \Delta_B + ik_B & \varepsilon_A - \varepsilon_B - J_Z & J_X + J_Y & \Delta_A - ik_A \\ \Delta_A + ik_A & J_X + J_Y & -\varepsilon_A + \varepsilon_B - J_Z & \Delta_B - ik_B \\ J_X - J_Y & \Delta_A + ik_A & \Delta_B + ik_B & -\varepsilon_A - \varepsilon_B + J_Z \end{pmatrix} \tag{E-6}$$

Depending upon the values of the coupling constants, there are different types of interactions between the two qubits. When $J_X = J_Y$, the interaction is called the XXZ interaction. When $J_X = J_Y = 0$, the interaction is called the Ising interaction which is commonly used to model the interaction between superconducting Josephson junction qubits. When $J_Z = 0$, the interaction is called the XY interaction. When $J_X = J_Y = J_Z$, the interaction is called the Heisenberg interaction.

Under the Ising interaction, $J_Z = \xi$ and $k = 0$, equation (E-6) becomes

$$\mathbf{H} = \begin{pmatrix} \varepsilon_A + \varepsilon_B + \xi & \Delta_B & \Delta_A & 0 \\ \Delta_B & \varepsilon_A - \varepsilon_B - \xi & 0 & \Delta_A \\ \Delta_A & 0 & -\varepsilon_A + \varepsilon_B - \xi & \Delta_B \\ 0 & \Delta_A & \Delta_B & -\varepsilon_A - \varepsilon_B + \xi \end{pmatrix} \tag{E-7}$$

Suppose it is required to perform a unitary operation on qubit B only, without performing any operation on qubit A. Since qubit A is coupled to qubit B, the evolution of qubit B will depend on the state of qubit A. To fix the state of qubit A, the bias on it is made much greater than the tunneling parameter, i.e., $\varepsilon_A >> \Delta_A$. The resulting Hamiltonion is

$$\mathbf{H} = \begin{pmatrix} \varepsilon_A + \varepsilon_B + \xi & \Delta_B & 0 & 0 \\ \Delta_B & \varepsilon_A - \varepsilon_B - \xi & 0 & 0 \\ 0 & 0 & -\varepsilon_A + \varepsilon_B - \xi & \Delta_B \\ 0 & 0 & \Delta_B & -\varepsilon_A - \varepsilon_B + \xi \end{pmatrix} \tag{E-8}$$

which is a block diagonal matrix. Each block is a 2 × 2 matrix which represent the two subspaces in the four-dimensional Hilbert space. The reduced Hamiltonian governing the evolution of the target qubit, B, in subspaces where the control qubit, A, is $|0\rangle$ and $|1\rangle$ are

$$\mathbf{H}_{B|0\rangle} = \begin{pmatrix} \varepsilon_A + \varepsilon_B + \xi & \Delta_B \\ \Delta_B & \varepsilon_A - \varepsilon_B - \xi \end{pmatrix} \tag{E-9}$$

$$\mathbf{H}_{B|1\rangle} = \begin{pmatrix} -\varepsilon_A + \varepsilon_B - \xi & \Delta_B \\ \Delta_B & -\varepsilon_A - \varepsilon_B + \xi \end{pmatrix}, \tag{E-10}$$

respectively.

Appendix F

CNOT gate operation in a Linear Nearest Neighbor array of Qubits

Figure 12 shows a linear nearest neighbor array of qubits where each qubit is represented by a circle. There are two coupling parameters ξ_1 and ξ_2 which alternate along the line. These coupling and tunneling parameters are assumed to be fixed during fabrication.

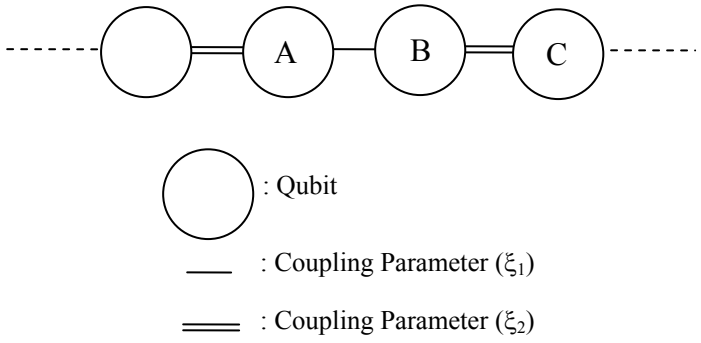

FIG 12: Linear nearest neighbor array of qubits where each qubit is coupled only to the qubits adjacent to it. Each qubit is represented by a circle and the coupling parameters ξ_1 and ξ_2 are represented by single and double solid lines respectively.

Suppose a CNOT gate operation is performed between A (Control) and B (Target). Here, the evolution of qubit B is determined by the state of qubits A and C, where both of them can be in any arbitrary state. Under the CNOT operation between qubits A and B, it is required that qubit B flip its state only when qubit A is in the $|1\rangle$ state. The state of qubit C should not interfere with the gate operation, i.e., it behaves as a "dummy" qubit.

The Hamiltonian of the three-qubit system is:

$$H = \Delta_A \sigma_{XA} + \Delta_B \sigma_{XB} + \Delta_C \sigma_{XC} + \varepsilon_A \sigma_{ZA} + \varepsilon_B \sigma_{ZB} + \varepsilon_C \sigma_{ZC} + \xi_1 \sigma_{ZA} \sigma_{ZB} + \xi_2 \sigma_{ZB} \sigma_{ZC} \qquad \text{(F-1)}$$

Here, the system parameter k = 0 for SQUIDs. From Appendix E, the reduced Hamiltonian for the target qubit B can be written as,

$$H = \Delta\sigma_X + (\varepsilon_T \pm \xi_1 \pm \xi_2)\sigma_z \tag{F-2}$$

Here, the coupling terms ξ_1 and ξ_2 either add or subtract from the bias on the target (ε_T) depending on whether qubits A and C are in the $|0\rangle$ or $|1\rangle$ states respectively. Table 2 shows a state table under the CNOT operation between qubits A and B with C acting as a "dummy" qubit. The corresponding frequencies of oscillation of the probability function have also been listed [1].

Table 2: Table shows the states of qubits before and after application of CNOT gate operation and corresponding frequencies.

The states of Qubits before CNOT operation			The states of Qubits after CNOT operation			Frequency
A	B	C	A	B	C	
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T + \xi_1 + \xi_2)^2}$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T + \xi_1 - \xi_2)^2}$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T + \xi_1 + \xi_2)^2}$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T + \xi_1 - \xi_2)^2}$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T - \xi_1 + \xi_2)^2}$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T - \xi_1 - \xi_2)^2}$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T - \xi_1 + \xi_2)^2}$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$2\sqrt{\Delta^2 + (\varepsilon_T - \xi_1 - \xi_2)^2}$

For the first four states given in the table, the target qubit B does not change its state when the control qubit A is in the $|0\rangle$ state, irrespective of the state of the qubit C. This implies that qubit B undergoes an integer number of complete oscillation cycles.

Moreover, since the target qubit B returns to its initial state, the attenuation to the amplitude is of no concern.

Next, when qubit A is in the $|1\rangle$ state, qubit B changes its state (last 4 states in Table 2). This means the target qubit B undergoes an odd integer number of half cycles in its transitional state and flips its state. In this case, there should be no attenuation to the amplitude of oscillations. With the effective bias (ε_T - $\xi_1 \pm \xi_2$) acting on the target qubit B when the control qubit A is in the $|1\rangle$ state, it is necessary that ε_T, ξ_1 and ξ_2 cancel each other in order to realize a switched logic state with no attenuation to the amplitude at a half cycle. To cancel ε_T, ξ_1 and ξ_2, two different pulses are required: $\varepsilon_T = \xi_1 + \xi_2$ and $\varepsilon_T = \xi_1 - \xi_2$. In both cases the bias on the target qubit is pulsed. The table below shows the frequencies under the different pulses:

Table 3: Calculation of pulses that is required to pulse target qubit for CNOT gate operation

Without any pulse	1st Pulse: $\varepsilon_T=\xi_1+\xi_2$	2nd Pulse: $\varepsilon_T=\xi_1-\xi_2$
$2\sqrt{\Delta^2+(\varepsilon_T+\xi_1+\xi_2)^2}$	$2\sqrt{\Delta^2+4\varepsilon_T{}^2}=\frac{P}{T}$	$2\sqrt{\Delta^2+4\xi_1{}^2}=\frac{Q}{T}$
$2\sqrt{\Delta^2+(\varepsilon_T+\xi_1-\xi_2)^2}$	$2\sqrt{\Delta^2+4\xi_1{}^2}=\frac{Q}{T}$	$2\sqrt{\Delta^2+4\varepsilon_T{}^2}=\frac{P}{T}$
$2\sqrt{\Delta^2+(\varepsilon_T+\xi_1+\xi_2)^2}$	$2\sqrt{\Delta^2+4\varepsilon_T{}^2}=\frac{P}{T}$	$2\sqrt{\Delta^2+4\xi_1{}^2}=\frac{Q}{T}$
$2\sqrt{\Delta^2+(\varepsilon_T+\xi_1-\xi_2)^2}$	$2\sqrt{\Delta^2+4\xi_1{}^2}=\frac{Q}{T}$	$2\sqrt{\Delta^2+4\varepsilon_T{}^2}=\frac{P}{T}$
$2\sqrt{\Delta^2+(\varepsilon_T-\xi_1+\xi_2)^2}$	$2\sqrt{\Delta^2+4\xi_2{}^2}=\frac{R}{T}$	$2\Delta=\frac{M+1/2}{T}$
$2\sqrt{\Delta^2+(\varepsilon_T-\xi_1-\xi_2)^2}$	$2\Delta=\frac{M+1/2}{T}$	$2\sqrt{\Delta^2+4\xi_2{}^2}=\frac{R}{T}$
$2\sqrt{\Delta^2+(\varepsilon_T-\xi_1+\xi_2)^2}$	$2\sqrt{\Delta^2+4\xi_2{}^2}=\frac{R}{T}$	$2\Delta=\frac{M+1/2}{T}$
$2\sqrt{\Delta^2+(\varepsilon_T-\xi_1-\xi_2)^2}$	$2\Delta=\frac{M+1/2}{T}$	$2\sqrt{\Delta^2+4\xi_2{}^2}=\frac{R}{T}$

Here, P, Q, R and M are integers, where $M = 0$. Let $\xi_1 = m\varepsilon_T$ and $\xi_2 = (1\text{-}m)\varepsilon_T$, where m is some number. Then,

$$2\Delta=\frac{1}{2T}\Rightarrow\Delta^2T^2=\frac{1}{16} \tag{F-3}$$

Also we have,

$$2\sqrt{\Delta^2+4\varepsilon_T{}^2}=\frac{P}{T}$$

$$\Rightarrow\frac{\varepsilon_T{}^2}{\Delta^2}=\frac{4P^2-1}{4} \tag{F-4}$$

and,

$$2\sqrt{\Delta^2 + 4\xi_1{}^2} = \frac{Q}{T}$$

$$\Rightarrow \Delta^2 + 4m^2\varepsilon^2 = \frac{Q^2}{4T^2} \tag{F-5}$$

and,

$$2\sqrt{\Delta^2 + 4\xi_2{}^2} = \frac{R}{T}$$

$$\Rightarrow \Delta^2 + 4(1-m)^2\varepsilon^2 = \frac{R^2}{4T^2} \tag{F-6}$$

Subtracting equation (F-5) from equation (F-6), we get

$$\frac{\varepsilon_T{}^2}{\Delta^2} = \frac{R^2 - Q^2}{1-2m} \tag{F-7}$$

From equations (F-4) and (F-7), we get

$$R^2 - Q^2 = \frac{(4P^2 - 1)}{4}(1-2m) \tag{F-8}$$

Since R and Q are integers, therefore R^2 - Q^2 should be integer that makes (1-4m) = 4*K (where, K is an integer). Therefore,

$$m = \frac{1-4K}{2} \tag{F-9}$$

For different value of m, we need to find a set of values for P, Q and R so that all three will be integers. For P = 20, Q = 30 and R = 60, a set of values for the parameters are Δ=0.025GHz, ξ_1=0.75GHz and ξ_2= 0.5GHz for a time step of 10ns. The two pulses that we need to pulse the bias in order to perform a CNOT gate operation will be, ε_T=1.25GHz and ε_I=0.25GHz.