QUANTUM COMMUNICATION ISSUES IN STORAGE AND PRIORITY

by

Çağlar Koca

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

Koç University

January, 2014

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Çağlar Koca

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Dr. Özgür Barış Akan(Advisor)

_____

Assoc. Prof. Dr. Alper Tunga Erdoğan

_____

Asst. Prof. Dr. Kaan Güven

Date: _____

Anneme ve Babama,

Suzan Koca, Ali Koca

ABSTRACT

Quantum Computation and Quantum Communication are emerging fields of science, in which principles of quantum mechanics are exploited to reach extraordinary results that may even seem counter-intuitive. However, the power of quantum computers are limited due to the fact that qubits cannot be stored or recalled as bits due to the no cloning theorem. In the first part of this thesis we focus on the problem of qubit storage. First, we propose the essential requirements for a good qubit storage system, which are access delay complexity, circuit complexity, maximum connectivity and use of ancillary qubits. Later, we introduce different possible qubit storage systems first as building blocks and then for large systems and comment on the feasibility of constructing these systems.

As another consequence of no cloning theorem, in quantum communications, a corrupt data cannot be retransmitted. Therefore, we may have to use corrupt data to its fullest. Moreover, since it is not yet possible to read and measure qubits as readily as bits, sometimes it may be necessary to evaluate which qubits should be handled first. In the second part of this thesis, in order to quantify the damage on a corrupt data and to prioritize our qubit reading, we propose a novel measure of bitwise information priority. We simulate our results in classical block codes in GF(2).

ÖZET

Kuantum Komputasyon ve Kuantum Haberleşme, Kuantum Mekaniği'nin prensiplerini kullanarak bazen akla yatkın görünmeyen sıradışı sonuçların ortaya çıktığı, bilimin gelişmekte olan dallarıdır. Buna karşılık, kuantum bilgisayarların gücü, klonlanamama teorisinin sonucu olarak kübitlerin bitler gibi depolanıp erişilememesi sebebiyle limitlidir. Bu tezin ilk ksmnda kübit depolama problemi üzerinde duruyoruz. İlk olarak iyi bir kübit depolama sisteminin sahip olması gereken özellikleri öneriyoruz. Bu özellikler, ulaşım hızı karmaşıklığı, devre karmaşıklığı, en fazla bağlantı sayısı ve tamamlayıcı kubit sayısıdır. Daha sonra, önce yapıtaşı olarak, sonra büyük sistemler halinde değişik kübit depolama sistemleri önerip, onları bu kriterlere göre karşılaştırıyoruz. Son olarak, bu sistemlerin yapılabilirliği üzerine yorum yapıyoruz.

Klonlanamama teorisinin başka bir sonucu olarak, kuantum haberleşmede, bozuk bilgi tekrar gönderilemez. Bu yüzden, bozuk bilgiden mümkün olduğunca yararlanılmalıdır. Ayrıca, kübitler bitler gibi hızlı okunamadığndan dolayı, hangi kübitlerin önce okunacağını hesaplamak faydalı olabilir. Bu tezin ikinci aşamasında, bozulan bilgiyi ölçmek ve okuyacağımız kübit önceliğini hesaplamak iin yeni bir bit önceliği ölçümü geliştiriyoruz. Sonuçlarımızı, GF(2)'deki klasik blok kodlarda test ediyoruz.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Dr. Özgür Barış Akan for his everlasting guidance and tolerance. Without his support, it would be impossible to complete this thesis.

I also would like to thank to Prof. Dr. Alper Tunga Erdoğan and Prof. Dr. Kaan Güven for their participation in my thesis jury and for their valuable feedback.

I am very grateful to all members of Next Generation Wireless Research Laboratory members. I specially want to thank Bige Deniz Ünlütürk, Mustafa Özger, Derya Malak and Ahmet Ozan Bicen for their valuable ideas which helped me construct this thesis.

I want to acknowledge the support of TÜBİTAK and Koç University, which made the completion of this thesis possible.

Finally, I would like to thank my parents, Suzan and Ali Koca, who motivated and supported me in all of my decisions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Quantum Computation is a recently introduced field of science, primarily by Richard Feynman. In his paper "Simulating Physics with Computers", Feynman shrewdly observed that if simulating a physical model requires exponential time, then constructing and observing the physical model might accomplish similar objectives [1]. Based on this principle, many quantum algorithms, some of which are counter-intuitive, have been suggested. These algorithms required a "quantum computer" which worked on "qubits" rather than classical bits. Just like the classical computers, quantum computers require central processing units, schedulers and memory. We introduce evaluation criteria for a good quantum memory management system. We, then develop some quantum memory architectures and use our criteria to evaluate their performance and feasibility of constructing such systems.

Due to the limitations in the quantum communications, i.e., impossibility or re-transmission and complications in locating and transferring qubits, we may need to make maximum use out of corrupt data. Even though this problem is more prominent in quantum communications, we may also face similar problems in classical communication. We propose a novel measure of bitwise information priority. Using this measure we determine the cost of losing a piece of information so that we can measure which parts of information is more prior, i.e., requires more protection. Further, we propose two novel measure to evaluate code strength: total protection and risk.

We investigate these concepts using block codes in GF(2). In this chapter, we briefly introduce the fundamental concepts and then, present the research objectives and solutions.

1.1 Quantum Computation

A quantum computer is a device that exploits the quantum mechanical phenomena to perform operations on data. Unlike the classical computers, quantum computers operate on quantum bits, i.e., qubits. However, they share many similarities with classical computers. Even though no fully functional quantum computer is constructed to the date, Oskin states that a quantum computer needs to consist of a quantum processor, a quantum scheduler and quantum memory [2]. A quantum scheduler is proposed to be a classical computer [2]. Research in both hardware and software of quantum processors and quantum memory systems is in progress.

1.1.1 Qubits and Bloch Sphere

Classical bits consist of zeros and ones. On the other hand, quantum bits, i.e., qubits, the quantum analogue of classical bits, are not restricted to zeros and ones. Qubits carry the information of a quantum vector, such as the polarization vector of a single photon or the spin vector of an electron. Therefore, qubits are vectors, rather than discrete identities like bits.

All possible qubits may be presented in the unit sphere, which is also called the Bloch Sphere. Figure 1.1 is a visualization of the Bloch Sphere. A qubit may be any point on the sphere of the Bloch Sphere. Note that there exist infinitely many possibilities for a qubit. Also note that memory and time requirements to simulate a qubit increase exponentially as the resolution increases.

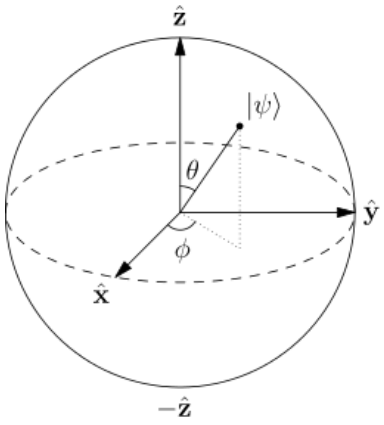

Figure 1.1: Bloch Sphere

1.1.2 Measurement Postulate

Measurement postulate is one of the primary revolutions brought by quantum mechanics. It states that a measurement performed on any state alters it instantly and permanently, as measured state *collapses* on one of the eigenstates of the measurement operator, meaning the previous state is lost forever.

Mathematically presenting, a measurement operator corresponding to the observable A can be represented as $A = \sum_i \lambda_i P_i$, where $P_i = |psi_m\rangle \langle psi_m|$. The wavefunction of the system can be represented in the basis of eigenvectors of A, i.e., $\Psi = \sum_i c_i \psi_m$. As stated earlier, a state collapses to one of the eigenstates of the measurement operator, and the probability of collapsing to the state ψ_i is given by $|c_i|^2$. The measurement is then described as $A|\psi_m\rangle = \lambda_m |\psi_m\rangle$. Note that λ_m is the measurement result.

Any qubit can be represented as $\sqrt{p_0}|0\rangle + \sqrt{p_1}|1\rangle$, where p_0 and p_1 are probabilities of measuring these results. After the measurement, the state of the qubit collapses to either $|0\rangle$ or $|1\rangle$. Therefore, the information prior to the measurement is lost. This is one of the primary issues concerning qubit transportation. It cannot be measured

elsewhere as the result cannot be transferred for further calculations.

1.1.3 Quantum Entanglement

Quantum Entanglement is a physical phenomenon in which two or more particles are intertwined so that neither of them can be described completely without the other. In other words, states of all parties in an entangled system is affected if one of them undergoes a change.

Entanglement is first described in a paper by Einstein, Podolsky and Rosen. They claimed that states of two particles described by the same wave function, even without interacting with each other, may be affected by a measurement on the other particle. They concluded that either there exists an unexplained communication between particles, i.e., spooky action at a distance, or wave function of the system lacks information on the system. This result causes either reality, i.e., there is no possible way to measure a value of a physical quantity with certainty; or locality, a set-up in a remote location can affect the results of of a measurement [3]. Later Bell showed that, indeed, either locality or reality is compromised. Counter intuitively, quantum particles do not have to obey local realism. [4]

EPR authors formulated entanglement as an impossible concept pertaining to the belief that local realism is a law of nature. Hence, they aimed to render quantum mechanics impossible. Entanglement may have been theorized to show the impossibility of quantum mechanics by impossibility of proofs [5], however entangled particles are later produced and they play large roles in quantum computation.

1.1.4 Quantum Circuits and Quantum Gates

In quantum algorithms, the input qubits are manipulated and some measurements are performed, usually at the latest stage of an algorithm. The operations that we

apply on qubits are called quantum gates, which are the building blocks of quantum circuits.

Quantum gates may represent a variety of operations, ranging from single qubit operations to multi qubit ones. Single qubit operations are the Hadamard Gate which introduce a rotation of π by $\hat{x} + \hat{z}$ axis and phase shift gates. Multi qubit operations are numerous. Some important ones are CNOT gate, which performs a controlled NOT operation, CCNOT which performs NOT operation with two controls, Swap gate, which swaps two qubits and Fredkin Gate or controlled Swap gate. Note that since qubits are not restricted to $|0\rangle$ or $|1\rangle$, hence controlled operations are not similar to digital logic operations with enable.

1.1.5 Quantum Teleportation

Quantum teleportation is one of the most counter intuitive concepts present in quantum information theory. It enables the communicating parties to send quantum information without the need of a quantum channel.

Consider communicating parties, namely Alice and Bob; who contacted each other at least once via a quantum channel, and exchanged parts of two entangled qubits. The quantum channel is later lost, and now only a classical communication channel remains. Alice is in need of sending a qubit to Bob. As we have stated in chapter 1.1.1, it requires an infinite number of bits to describe a qubit accurately. Instead of going with the classical approach, Alice may teleport her qubit to Bob using the EPR pair that they hold.

Assume that Alice aims to teleport the qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where α and β are unknown coefficients. Alice and Bob also have an EPR pair and without loss of generality, we can assign $\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|11\rangle$ to be the EPR pair they are sharing. Hence

the input state the teleportation becomes $|\psi\rangle\,(\frac{1}{\sqrt{2}}\,|00\rangle)$, or

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}\Big[\alpha\,|0\rangle\,\big(\,|00\rangle + |11\rangle\,\big) + \beta\,|1\rangle\,\big(\,|00\rangle + |11\rangle\,\big)\Big]$$

Now Alice performs a CNOT operation on the target qubit and the EPR paired qubit she possesses, target qubit being the control qubit. The result becomes

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\Big[\alpha\,|0\rangle\,\big(\,|00\rangle + |11\rangle\,\big) + \beta\,|1\rangle\,\big(\,|10\rangle + |01\rangle\,\big)\Big]$$

The next operation is passing $|\psi\rangle$ through a Hadamard gate. The qubits then evolve to

$$|\psi_2\rangle = \frac{1}{2}\Big[\alpha\big(\,|0\rangle + |1\rangle\,\big)\big(\,|00\rangle + |11\rangle\,\big) + \beta\big(\,|0\rangle - |1\rangle\,\big)\big(\,|10\rangle + |01\rangle\,\big)\Big]$$

Now Alice measures the qubits she has, and send the results with the classical channel to Bob. We can rewrite $|\psi_2\rangle$ to see how Alice's measurements affect Bob's qubit.

$$|\psi_2\rangle = \frac{1}{2}\Big[\,|00\rangle\,\underbrace{(\alpha\,|0\rangle + \beta\,|1\rangle)} + |01\rangle\,\underbrace{(\alpha\,|1\rangle + \beta\,|0\rangle)} + |10\rangle\,\underbrace{(\alpha\,|0\rangle - \beta\,|1\rangle)} + |11\rangle\,\underbrace{(\alpha\,|1\rangle - \beta\,|0\rangle)}\Big]$$

Depending on results of Alice's measurements, Bob's qubit collapses to one of the four states underbraced above. Bob later can operate on his qubit to evolve it back to $|\psi\rangle$. Note that if the measurement result is $|00\rangle$, no additional operation is required.

Interestingly, the original qubit, $|\psi\rangle$ is lost during the action. Alice can only teleport his qubit. After the teleportation she loses her copy. Another important observation is that the EPR pair is destroyed, i.e., an EPR pair can be used to teleport only one qubit.

1.1.6 No Cloning Theorem

As we have seen above, quantum communications provide extraordinary possibilities, some of which can be considered counter intuitive. However quantum communications

also has an Achilles' heel. An unknown qubit cannot be cloned. It can be proved very simply.

Assume that there is a unitary time evolution operator which clones all possible states, namely $|\phi\rangle$. If $\mathbf{U}$ is a universal cloning operator, it must work as $\mathbf{U}|\phi\rangle|e\rangle = |\phi\rangle|\phi\rangle$, where $|e\rangle$ is any state on which we will copy of the state $|\phi\rangle$. Assume $|\psi\rangle$ is one of the states that $\mathbf{U}$ can operate on, so that the result is $|\psi\rangle|\psi\rangle$.

Time evolution operators in quantum mechanics are unitary, hence they preserves inner product. Therefore, the inner product of $|\psi\rangle$ with any $|\phi\rangle$ must be preserved. Therefore,

$$\begin{aligned}
\langle e|\langle\phi|\,|\psi\rangle|e\rangle &= \langle e|\langle\phi|\,\mathbf{U}^{\dagger}\mathbf{U}\,|\psi\rangle|e\rangle = \langle\phi|\langle\phi|\,|\psi\rangle|\psi\rangle \\
\langle e|\,|e\rangle\langle\phi|\,|\psi\rangle &= \langle\phi|\,|\psi\rangle\langle\phi|\,|\psi\rangle \\
\langle\phi|\,|\psi\rangle &= \big(\langle\phi|\,|\psi\rangle\big)^2
\end{aligned}$$

We assumed above that $\mathbf{U}$ was a universal cloner. However, the input range for the cloner is limited to $|\psi\rangle = |0\rangle$ or $|\psi\rangle = |\phi\rangle$, which contradicts with the universality of $\mathbf{U}$. Therefore, there is no universal cloning operator.

Note that since broadcasting is also an act of creating information on a different location, unknown qubits cannot be broadcast as well. This limits quantum communications solely to peer-to-peer communication. Also, when a qubit is transferred, the information is lost from the transmitter side of the channel forever.

1.2 Research Objectives and Solutions

The objectives of our research and the solution approaches are explained in this section.

1.2.1 Quantum Memory Management Systems

Classical memory systems employ high impedance gates. Normally many memory cells are connected to the same bus. In order to access a memory location, the location is activated with control signals. As all the non-active memory cells are in high impedance state, the content of the intended memory location is transferred to the bus and any hardware connected to the bus may receive them.

However, quantum circuits do not have high impedance gates. Due to the limitations in quantum communication, we need to design a totally different memory system. In this part of the thesis, we first state the criteria for a good quantum memory management system. We then suggest some possible one layered architectures for quantum memory management. These architectures are more like a building block of a larger memory organization, then the memory itself. Using these basic building blocks, we develop more feasible memory architectures and comment on their performance and feasibility using our criteria. Our approach in this section is to state some probable memory management systems and decide on the more feasible architectures, rather than suggesting an architecture superior on others.

1.2.2 Bitwise Memory Management Systems

Shannon Entropy is used to quantify the amount of information. However, it assumes all bits constructing a message are of the same importance. In fact, when examined by the receiver, bits composing a message may have varying importance, i.e., priority. Uncovering some bits may present more insight to the receiver. In this part of the thesis, we aim to device a measure for the priority of bits in a message. We use Block Codes to quantify our results. Later we extend our results to determine success of a code and suggest two novel code strength measures: Code Protection and Risk.

1.3 Thesis Outline

This thesis is organized as follows: In Chapter 2, we briefly mention quantum computing and quantum memory management. Next, we discuss the features of a successful quantum memory management systems. Later in the same chapter, we suggest and analyze different memory management systems and we compare the suggested architectures. Chapter 3 is on Bitwise Information Priority Measure. In this chapter, we first discuss why different bits in a message may carry different importance. Later we familiarize the readers with Information Value Theory, proposed by R. A. Howard. Later we provide information value calculations on block codes to maximize our gain from an arbitrary received message. Later in the same chapter we focus our attention to bitwise information value in block codes and compare priority of bits. Then, we suggest two novel code strength measure, code protection and risk. In the conclusion of the third chapter different block codes are discussed in terms of code protection and risk parameters. Chapter 4 is conclusion where we highlight important contributions to the field and suggest future directions.

Chapter 2

QUBIT STORAGE ARCHITECTURES

2.1 Introduction

Ever since Feynman hinted the quantum computing in 1982, a long way has been covered to turn this dream into reality. A number of quantum algorithms have revolutionized many fields of computing, especially in Searching [6] and Factoring [7]. Some quantum processor units [8, 9] and quantum arithmetic logic unit structures have been developed [10, 11] and some of them are engineered [11]. Some memory structures are also theorized [2, 12], however, to the best of our knowledge, no quantum memory management system feasible for large memory contents has been suggested.

Classical memory systems use finite number of buses to access a memory location. The desired memory location can be activated by various methods, usually with encoders or multiplexers, while the other memory cells are kept at high impedance state. The desired content may reach any place from the bus as it is not affected by other memory cells. Also, as the content of the cell is copied to the bus, the information stored in the cell is preserved.

Due to the lack of high impedance gates for quantum states, quantum memory cells cannot be accessed directly by their addresses. In fact, there are two mechanisms which may transport qubits from one location to another: Swapping and teleportation.

For a memory consisting of N qubits, complexity of reaching the desired content

is obviously $O(N)$. Therefore, brute force swapping without any organization is impractical for large memories. Teleportation itself cannot be a solution as well. We can only teleport a qubit to the location of an EPR singlet. After teleportation, the target qubit has to be transported from the location of the EPR singlet to the quantum processor, which introduces another transportion. Moreover, the EPR pairs are also qubits and they somehow have to be distributed to the teleportation zones. Note that sending EPR pairs are no different than retriving qubits. In both cases a qubit is transferred from a target to a destination.

In the second section of this chapter we will introduce the features of a qubit storage system. In Section III, we will explain and analyze possible one-staged storage systems using only swapping. In section IV, we will deepen our analysis to multi-layered storage architectures. In Section V, we will employ teleportation as well as swapping for storage. These given algorithms will be compared in section VI.

2.2 Properties of Qubit Storage Architectures

We suggest that a memory management system has four main properties:

1. Access Delay Complexity
2. Circuit Complexity
3. Maximum Connectivity
4. Ancillary Qubits

Access delay complexity, ADC, is how access time increases as the memory size increases. ADC affects the overall speed of the system. Circuit complexity, CC, is how the number of gates required to access a memory location increases as the

memory size increases. CC is related to the circuit size, cost of the system and feasibility. Maximum Connectivity, MC, is the most number of gates connected to a qubit. MC is a concept close to fan-out in classical circuits. Ancillary Qubits, AQ, are the number of qubits required to reach a memory location. These qubits are used in qubit teleportation, as we need an EPR pair to achieve teleportation. Note that MC and AQ give us direct values rather than complexities.

It is clear that all of these quantities should be small for a qubit storage architecture to be feasible. Below, we will explain and analyze the possible architectures with their access delay complexities, circuit complexities, maximum connectivity and ancillary qubit numbers.

2.3 One Layered Architectures with only Swapping

In this chapter we will discuss possible one layered qubit storage architectures. These architectures may not be feasible for large memories but they may constitute building blocks for larger memory structures. Note that we use controlled swap gates, i.e., Fredkin Gates to swap qubits.

2.3.1 Simple Swapping

Simple Swapping is basically to swap each qubit until the target qubit reaches its destination. A simple swapping circuit is depicted on Figure 2.1. It is obvious that the circuit complexity of Swapping is $O(N)$, as it requires N Fredkin Gates. The access delay complexity is also $O(N)$, due to the fact that on average we need to swap the memory content $\frac{N}{2}$ times. Its maximum connectivity is 2, as each qubit is connected to two gates. Due to the large amount of delay, we can say that simple swapping is not a feasible suggestion for fast access for large blocks. Note that output can be taken at any quantum wire.

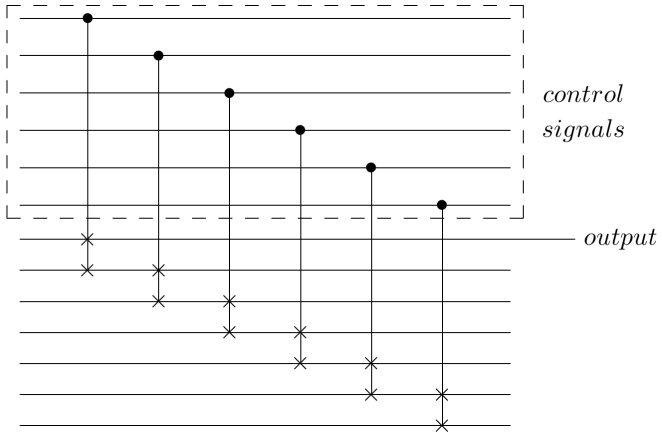

Figure 2.1: Simple Swapping Circuit

2.3.2 Connected Swapping

We suggest that in order to enhance the simple swapping, we can simply connect all the memory cells to the target location. Therefore, one swap will be enough to transfer the desired content to the target. The Connected Swapping is shown on Figure 2.2. Even though the circuit complexity remains as $O(N)$, the access delay complexity is now reduced to $O(1)$. However, the maximum connectivity becomes N. Note that, connecting many Fredkin Gates to a single memory cell may be impractical.

2.3.3 Logarithmic Swapping

We can further idealize swapping to make Connected Swapping more practical, by trading the maximum connectivity with access delay complexity. In Logarithmic Swapping, some memory cells have increased connectivity than others so that the desired content will hop to more connected memory locations to reach its target. Figure 3 displays our suggested Logarithmic Swapping scheme.

Note that in Logarithmic Swapping $\frac{N}{2}$ memory cells have one Fredkin Gate connected to them. $\frac{N}{4}$ of them have two, $\frac{N}{8}$ of them have three Fredkin Gates and so on.

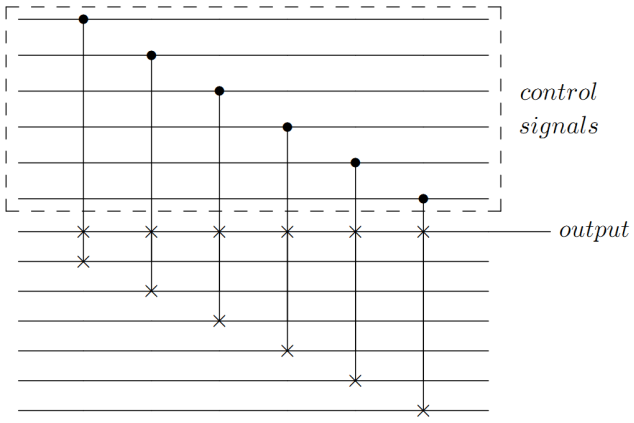

Figure 2.2: Connected Swapping Circuit

Due to this architecture we named it as logarithmic. The most connected memory cell will have $\log_2(N)$ Fredkin Gates. The total number Fredkin Gates needed to access all memory locations can be calculated as, $\sum_i^{\log_2(N)} i\frac{N}{2^i}$ which is also $O(N)$.

The architecture of the circuit is designed to make sure that no cell is connected to more than $\log_2(N)$ other cells. This is due to the fact that each cell hops to a twice as many connected cell to reach the output.

It is obvious that the worst case access delay time is proportional to $log_2(N)$. To calculate ADC, we assume that all memory locations are simultaneously accessed. $\frac{N}{2}$ least connected cells will jump to the second level with one move. After this jump, there will be $\frac{N}{2} + \frac{N}{4}$ cells in the second level. Summing them all gives us:

$$\sum_{i=1}^{log_2(N)} \frac{N}{2^i} = N\log(N) - N$$

We can find ADC for a single memory cell by dividing this result with N, hence ADC is $O(\log_2(N))$. Although the circuit complexity is not improved, the access time is much better compared to Simple Swapping and and much more practical compared to Connected Swapping.

Note that in the Logarithmic Swapping circuit there appears two cells are connected to three Fredkin Gates, instead of one cell. This is due to the fact that the size of the memory is only eight qubits. For larger circuits, $\frac{N}{8}$ of the cells having three connectivity holds.

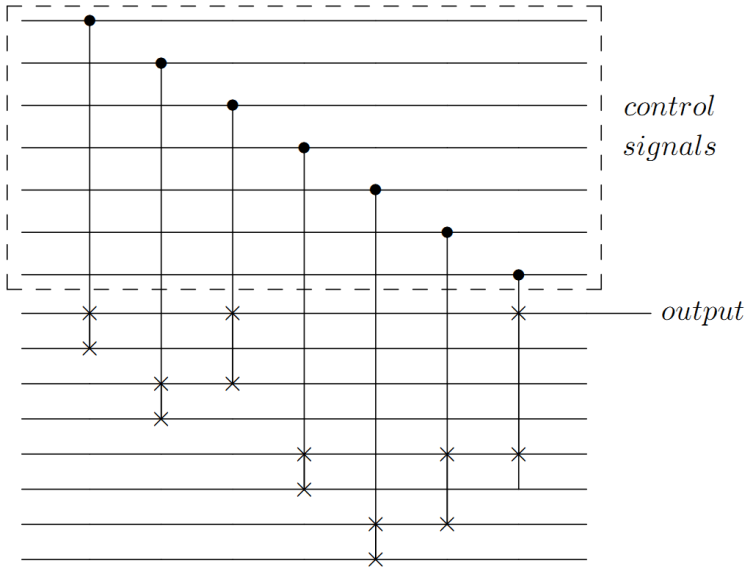

Figure 2.3: Logarithmic Swapping Circuit

2.4 Multi Layered Architectures with only Swapping

In the previous section, we discussed the one layered architectures using only swapping. In this section, we will use these architectures and organize multi layered qubit storage systems. Note that there exist an infinite number of possibilities, however we will focus only on three feasible circuitry.

2.4.1 Mixed Swapping

Using Simple Swapping and Logarithmic Swapping, it is possible to build more realistic storage architectures. By building fixed size memory blocks using Logarithmic Swapping, and connecting them at their ends using Simple Swapping, it is possible to reach any memory location relatively faster than Simple Swapping. It is also more realistic as the number of gates connected to a single memory cell is limited, unlike Connected Swapping and Logarithmic Swapping. A Mixed Swapping Circuit Scheme is provided on Figure 2.4.

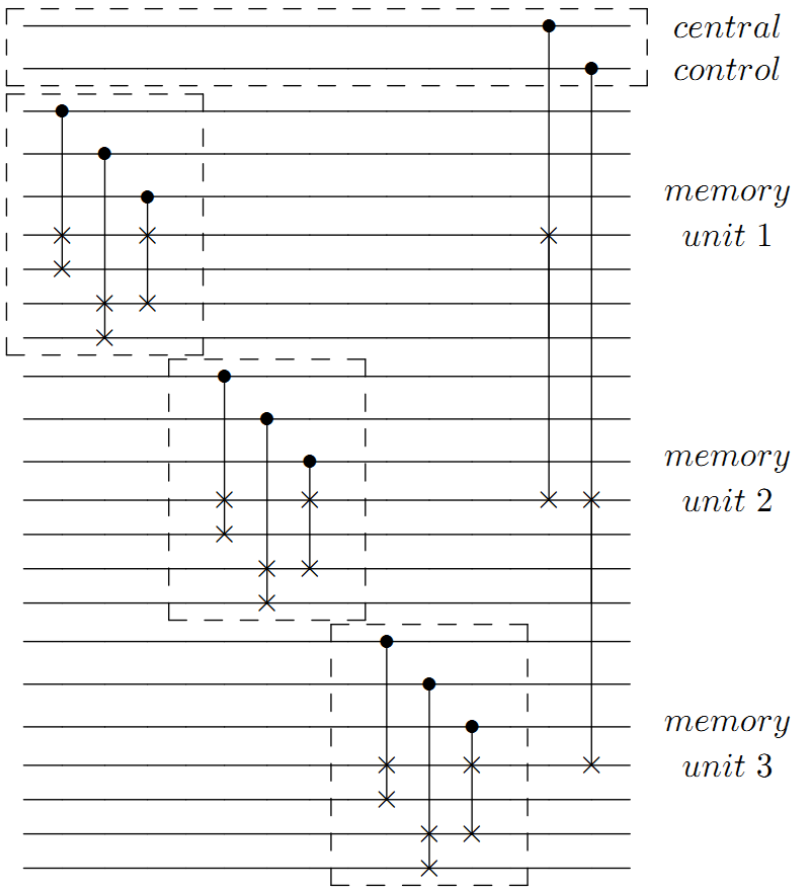

Figure 2.4: Mixed Swapping Circuit

We can calculate the circuit complexity as follows: If the memory block size is n, there will be $\frac{N}{n}$ memory blocks. Each memory block will have $O(n)$ Fredkin Gates and as a result, CC will be $O(N)$.

Access delay complexity depends both on the memory block size and number of total memory cells. Inside a memory block, ADC is $O(\log_2(n))$ swapping is required and there will be $\frac{N}{n}$ memory blocks. Therefore the ADC becomes $O(\frac{N}{n} log_2(n))$. For a fixed memory size, i.e., for constant N, minimizing this result gives us $n = \mathrm{e}$. Even though a memory block cannot be constructed with such small number of memory cells, to achieve lower ADC we need to keep the memory block sizes small.

The maximum connectivity for such a system depends only on the memory block size. Therefore, the maximum connectivity is fixed at $\log_2(n)$

2.4.2 Concatenated Connected Swapping

We can use memory blocks organized with Connected Swapping and connect them using Connected Swapping in the second layer as well. We can even increase number of layers in the circuitry. A Concatenated Connected Swapping Circuit is displayed in Figure 5.

The advantage of such an architecture is obviously in the maximum connectivity. MC for each block is n and for a memory size of N, MC of the overall circuitry will be $max(n+1, \frac{N}{n})$. The best performance of this architecture is at $n = \sqrt{N}$, which is $\sqrt{N} + 1$. A better performance might be achieved using multiple layers. If we use m layers of Connected Swapping, MC will be $\sqrt[m]{N} + 1$.

Access delay complexity performance of this architecture is also favorable. At each memory block, ADC is simply $O(1)$ and for m layers it is $O(m)$.

Circuit complexity of Concatenated Connected Swapping can be calculated as follows. For each block CC is $O(n)$. In this configuration, there will be n^{m-1} blocks

in the first layer and n^{m-2} blocks in the second layer etc. Summing all the layers, we get $n\frac{1-n^{m-1}}{1-n}$ blocks, and for large n there approximately n^{m-1}. Multiplying this result with $O(n)$, the overall complexity becomes $O(n^m)$, which is equal to $O(N)$.

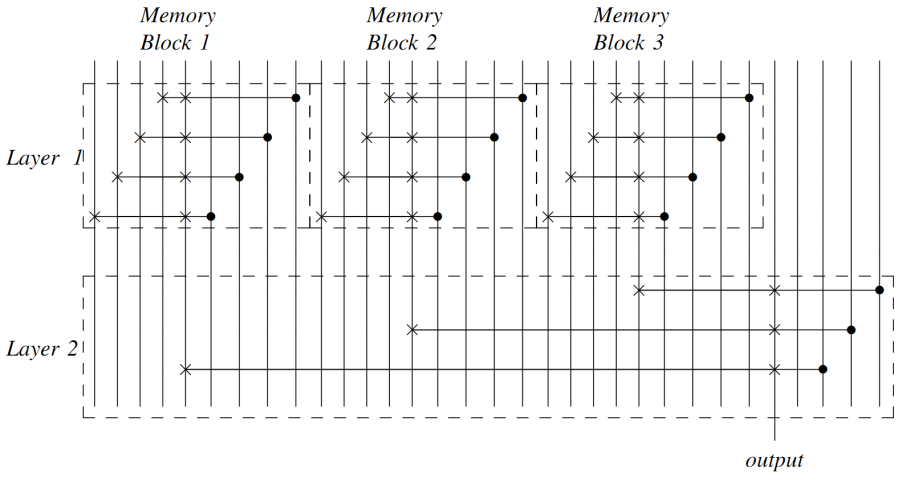

Figure 2.5: Concatenated Connected Swapping Circuit

2.4.3 Concatenated Logarithmic Swapping

We can organize memory blocks using Logarithmic Swapping and collect the outputs of these blocks in an intermediate location, where we can employ Logarithmic Swapping one more time. Furthermore, we can increase the number of layers in the memory architecture. A Concatenated Logarithmic Swapping Circuit is in Figure 2.6.

The circuit complexity for this architecture is also $O(N)$, as in the previous architectures. Note that if the block size is n, there will be $\frac{N}{n}$ for a memory size of N qubits. Each memory block has a CC of $O(n)$, hence the whole system has a CC of $O(N)$.

To calculate the access delay complexity, we need to consider the layered structure of the architecture. Inside a memory block the ADC is $O(\log_2(n))$, and in the second

layer, it is $O(\log_2(\frac{N}{n}))$. Summing these values give us $O(\log_2(N))$. Note that this result is independent of the block size. Note that addition of other layers do not change this result, as $O(\log_2(\frac{N}{n_1}) + \sum_i \log_2(\frac{n_i}{n_{i+1}}))$, which is again $O(\log_2(N))$.

The maximum connectivity is the connectivity in each block. In the memory blocks, MC is $\log_2(n)$ and in the intermediate location it is $\log_2(\frac{N}{n})$. If, the most connected qubits of the layers are connected to each other, MC becomes $\log_2(N)$, which gives us no advantage over the simple Logarithmic Swapping. Hence, we need to connect the least connected qubits in the first layer to the second layer. MC then becomes, $max(\log_2(\frac{N}{n}), \log_2(n))$. This result is definitely minimized for $n = \sqrt{N}$. For m layered architecture, MC becomes $log_2(\sqrt[m]{N})$. Note that this is the best result that we obtained among qubit storage architectures using swapping.

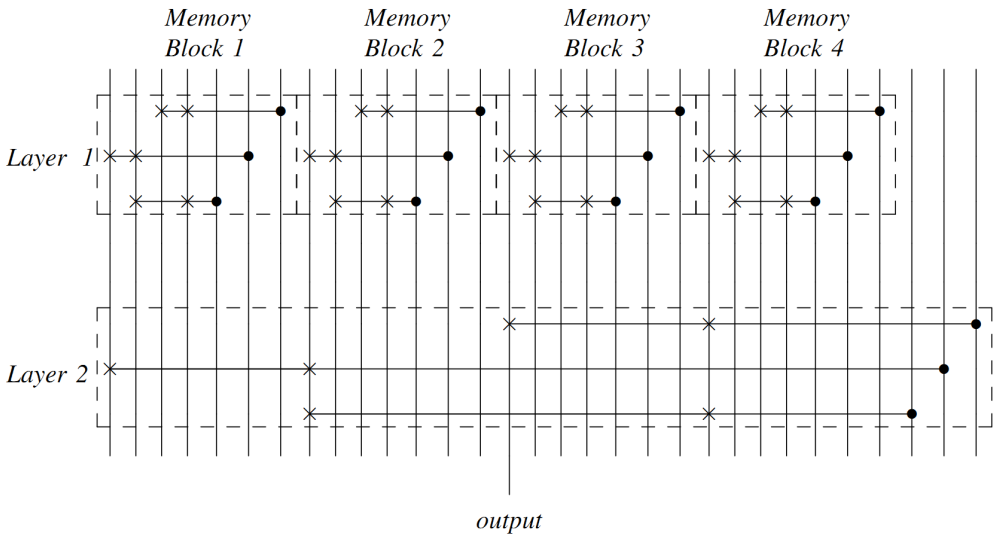

Figure 2.6: Concatenated Logarithmic Swapping Circuit

Note that no ancillary qubit is used in Swapping technique.

2.5 Architectures with Swapping and Teleportation

To reduce the access delay complexity due to swapping, Oskin et al, suggested employing teleportation as well [2]. Their suggestion is based on small memory units connected to qubit refresh units. Each memory unit is also located to a code teleporter. The contents of the memory cells are to reach the teleporter by Simple Swapping, and then they are to be teleported to their target location. Figure 2.7 is the memory structure suggested by Oskin et al.

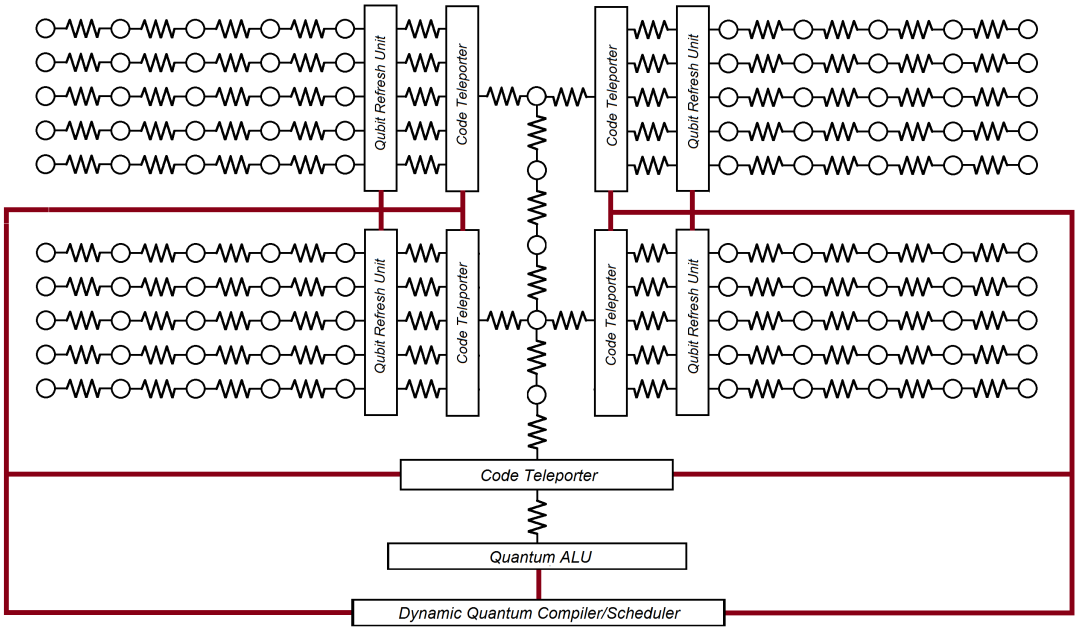

Figure 2.7: Quantum Memory Management with Swapping and Teleportation [2]

In this architecture, the main delay is due to swapping, and the number of swappings required is proportional to the size of quantum memory units. However, these memory units cannot be made arbitrarily small, as after teleportation the qubit does not directly reach its desired destination. Qubits may only be teleported to an in-

termediate location where one of the EPR pairs used for teleportation is stationed. Hence, the small sized memory units will require larger number of EPR pairs to be teleported and it increases the size of the intermediate location, where a second search needs to be done. In this section we will discuss the implementation of teleportation on different swapping techniques.

2.5.1 Access Delay Complexity

We can find an optimum limit for the number of teleporters for the fastest access. Assuming the time delay for teleportation is $O(1)$, the ADC is due to either swappings from the original memory location to the teleporter, or swappings from the intermediate location to the destination. If we have N memory cells and n teleporters, there will be $\frac{N}{n}$ memory cells per memory block with a teleporter. Teleporters transport the target memory cell content to an intermediate location, which can loosely be considered as RAM for a quantum computer. The size of the intermediate location is equal to the number of teleporters, as one of the EPR pairs employed by the teleporters will be residing here.

Simple Swapping

In case of employing Simple Swapping in both memory blocks and intermediate locations, ADC then becomes $O(\frac{N}{n} + n)$, which is basically $O(max(\frac{N}{n}, n))$. This result is clearly minimized for $n = \sqrt{N}$. If we construct three intermediate locations with sizes n_1 and n_2, the complexity becomes $O(max(\frac{N}{n_1}, \frac{n_1}{n_2}, n_2))$, and this result is minimized for $n_1 = N^{2/3}$ and $n_2 = N^{1/3}$. Generalizing this result for m intermediate locations, the complexity then becomes $O(\sqrt[m+1]{N})$.

Connected Swapping

If we employ Connected Swapping, ADC is definitely $O(m)$, as in each layer, the ADC is O(1), hence the number of layers provides us ADC.

Logarithmic Swapping

Assuming we employ Logarithmic Swapping for both the memory blocks and the intermediate locations, the ADC is $O(\log_2(\frac{N}{n})) + \log_2(n))$, as both of the operations had to be performed in order. Hence, ADC becomes $O(log_2(N))$ which is the same result obtained for the storage systems with only Logarithmic Swapping. Furthermore, using more than one layer of intermediate locations will not change the result as the complexity will then become $O(\log_2(\frac{N}{n_1}) + \log_2(\frac{n_1}{n_2}) + \log_2(n_2))) = O(log_2(N))$, where n_1 and n_2 are the sizes of the intermediate locations.

2.5.2 Circuit Complexity

In Swapping and Teleportation storage system, the circuit complexity is mainly the number of Fredkin Gates and teleporters in the main memory. In the previous section, we calculated the circuit complexity of any of the swapping techniques as $O(N)$. The circuit complexity of any memory block will be $O(n)$, and since there will be $\frac{N}{n}$ such units, the complexity will be $O(N)$. For multiple intermediate layer structures, this result does not change as the higher layers will serve less memory blocks and will have a smaller degree of complexity.

Regardless of the swapping method and number of intermediate locations, the circuit complexity for N memory cells is always $O(N)$.

2.5.3 Maximum Connectivity

Maximum Connectivity of Swapping and Teleportation architecture depends only on the swapping mechanism as teleportation introduces no further connectivity to the system.

Simple Swapping

In case of Simple Swapping, MC is simply 2, as each memory cell is connected to two other cells.

Connected Swapping

If we use Connected Swapping for the whole architecture, MC is determined by the largest memory block size. Since MC is n for a block of size n, MC for the architecture is $max(\frac{N}{n}, n)$. As in the use of Logarithmic Swapping this result is optimized for $n = \sqrt{N}$ and for an m intermediate locationed structure it becomes $\sqrt[m+1]{N}$.

Logarithmic Swapping

If we use Logarithmic Swapping, the memory block size, i.e., memory cells per teleporter determines the connectivity of the architecture. If there are n teleporters, then MC becomes $max(\log_2(\frac{N}{n}), \log_2(n))$, which is then minimized for $n = \sqrt{N}$. For multi layered architectures, the connectivity is minimized for a block size of $\sqrt[m+1]{N}$ for m intermediate locations. Therefore, MC becomes $\log_2(\sqrt[m+1]{N})$.

2.5.4 Ancillary Qubits

In order to teleport one qubit, an EPR pair is required, hence for every teleportation two ancillary qubits are used. If there are m intermediate locations, obviously the number of ancillary qubits required will be $2m$. As a result, although the number

Table 2.1: Comparison of Qubit Storage Architectures

Architecture	ADC	CC	MC	AQ
Simple Swapping	$O(N)$	$O(N)$	2	0
Connected Swapping	$O(1)$	$O(N)$	N	0
Logarithmic Swapping	$O(log_2(N))$	$O(N)$	$log_2(N)$	0
Mixed Swapping	$O(\frac{N}{n}log_2(n))$	$O(N)$	$log_2(n)$	0
Concatenated Connected Swapping	$O(m)$	$O(N)$	$\sqrt[m]{N}+1$	0
Concatenated Logarithmic Swapping	$O(log_2(N))$	$O(N)$	$log_2(\sqrt[m]{N})$	0
Simple Swapping and Teleportation	$O(\sqrt[m+1]{N})$	$O(N)$	2	2m
Connected Swapping and Teleportation	$O(m)$	$O(N)$	$\sqrt[m+1]{N}$	2m
Logarithmic Swapping and Teleportation	$O(\log_2(N))$	$O(N)$	$\log_2(\sqrt[m+1]{N})$	2m

of intermediate locations decrease the access delay complexity, it also increases the amount of ancillary qubits required. Therefore, there is a trade off between ancillary qubit number and access delay complexity.

2.6 Conclusion

In this work, we have suggested some novel qubit storage structures and analyzed them in the light of four important factors, namely, Access Delay Complexity, Circuit Complexity, Maximum Connectivity and Ancillay Qubits required. A summary of this paper is presented on Table 2.1.

First of all, we can deduce from Table 2.1 that the circuit complexity values for all these circuits are the same. Although we can decrease the circuit size in some structures, the growth will still be linear.

We can see that access delay complexity can be traded off with maximum connectivity. Systems with larger maximum connectivity values offer less access delay complexity. However, these systems are harder to build for large memories. Nevertheless, they can be considered for small fast access memories, analogous to the cache memory of a classical microprocessor.

We also realize that, apart from techniques using Connected Swapping, the best ADC value achieved is $O(log_2(N))$. Investigating the MC values, the best complexity is $O(log_2(n))$ for Mixed Swapping.

Another important observation from Table I is that techniques with teleportation do not provide any additional advantage over Logarithmic, Mixed or Concatenated Logarithmic Swapping techniques. In addition, the ancillary qubits are required only in teleportation. The required AQ number makes multi-layered Swapping and Teleportation system unfeasible, as that many ancillary qubits will be required to write it in the memory as well as reading it. In these architecture schemes, EPR pairs for teleportation has to be constantly created and distributed to memory locations. This fact reduces the feasibility of these techniques even further.

We conclude that Logarithmic, Mixed and Concatenated Logarithmic Swapping techniques are all possible, as these architectures have both smaller ADC and MC

Although these architectures are well suited to transfer qubits from one location to another, a total quantum memory management system would require classical computers integrated to the quantum memory management systems as well. During accessing of any qubit, many will change their location hence a track of all these operations should be kept in classical memory cells. In a future work, these architectures should be refined and the classical computation complexities included as well.

Chapter 3

CLASSICAL AND QUANTUM INFORMATION PRIORITY

3.1 Introduction

Even sixty years after its publication, principals suggested in Shannon's colossal work, "A Mathematical Theory of Communication" still govern digital communication world. Using those principles, channel capacities for multitudinous channels have been calculated [13, 14]; even capacities for quantum channels are established [15]; and compression systems for various sources have been designed [16]. The primary principle is that information can be measured by the uncertainty of the source. In other words, the amount of information of a source is measured by the inability of the receiver to predict coming messages. Every bit, every symbol or every letter is considered the same, regardless of the content of the message. [17]

However, this is not the case in our daily lives. Even though they might consist of similar numbers of bits, almost all mail servers prioritize some of our mails and filter out spam. While watching soccer match from cable TV, in spite of using similar number of bits to encode every cm^2 of the screen, audience tend to focus around the ball more than other regions. A person planning to take a bus at 18:00 first investigates availability of buses closer to 18:00. As we can see, content, spatial or temporal orientation of a message play more prominent role in the value of information, compared to the number of bits in the encoded message.

Realizing the importance of the content, R. A. Howard developed Information

Value Theory [17]. Howard suggested that receiving a critical tip in the stock market might have the same probability with choosing a certain dish for dinner. Shannon entropy of both of the sources may be the same, however their importance are certainly different. Even though Howard focused on the economical aspects of information, i.e., the expected profit; his work is used in a variety of areas from prioritizing health research [18] to investments in petroleum industry [19]. Information Value Theory is used in communications theory as well, as in quantification of relevance in sensor networks [20].

Concerning more about the physical aspects of communication, value of each bit in a message is not the same. For example, a 30 kB password protected document is useless without its 8 byte password. If you have the document but not the password, than the 8 bytes password is worth 30 kB amount of information. We can even dramatize our example by given out 7 bytes of the password. Assuming no trial-and-error is permitted, the remaining bit is worth 15 kB. Similarly, the password without the document is as much useless as the document without password.

We can find some examples from real communication architectures as well. In a half duplex channel, corruption of an acknowledgement signal, ACK, has the same importance as the corruption of a whole frame: both results in the frame to be re-transmitted. However, corruption of the ACK usually requires less bit errors, making the ACK bits more valuable.

The same approach is valid for bad intentions as well. If Eve wants to corrupt a document by erasing a limited number of bits, her first target should be the password bits. Similarly, if Eve wants to decrease the throughput of a channel by the least amount of intervention, she should target the ACK. Therefore, the information value, or priority, of a bit can be measured by the amount of bits it undercovers.

In some cases, information pieces may not be prior, but their loss may increase the risk associated with losing a much larger information vector. Consider a scenario

in which password to a 30 kB document is transmitted with 12 bits, 4 of which are parity bits. Depending on the error probability of the channel, loss of the parity bits may not be important in the decoding process of the password. However, losing these bits puts all 30 kB document at risk, as the only protection the whole document receives is the four parity bits. Hence, we need to compute not only the information priority, but also the risk associated with any information vector to determine their importance.

Information priority is useful is there exists some constraints on bits to be transmitted or read. If there is an energy constraint limiting the number of bits that may be transmitted, prior bits should be transmitted first. Or, if reading a data requires a long time, i.e., measuring a qubit in quantum communications, qubits may be prioritized to receive a fast result with minimum error probability.

In the second section of this chapter we will briefly revisit the Information Value Theory. In section III, using some principles in the Information Value Theory, we will investigate information value in coding theory in $GF(2)$. In section IV and V we will calculate bit values and priorities for Hamming Codes. We then will extend our investigation to the code protection offered by the parity bits, and the concept of risk in section VI. Secion VII is the conclusion.

3.2 Information Value Theory

Information Value Theory is proposed by R. A. Howard to determine value of a certain information. Assume that our expected gain with our current knowledge is $E(C)$, and some information regarding a certain random variable is available at a certain cost. The value of information is calculated by the increase in our gain, i.e., $V(X) = E(C) - E(C|X)$. In the publication a bidding problem is used as a toy problem to demonstrate the value of different clairvoyance services used in bidding.

It is stated that expectation value of a random variable given a certain state S is $E(u|b,S) = \int E(u|b,v,S) f_v(v|S) \mathrm{d}v$, where $f_v(v)$ is the probability density function of another random variable v, in which u is dependent on and b is a variable which we can choose freely. Clearly, the maximum value of $E(u|b,S)$ is

$$E(u_{max}|b,S) = \underset{b}{Max}\ E(u|b,S)$$

If u is a function of N random variables then the expectation value of u becomes

$$E(u|b,S) = \int \cdots \int E(u|v_i,b,S) f_{v_i}(v_i|S) \mathrm{d}v_i, \quad i = 1 \ldots N$$

The expectation value of maximum u changes if we obtain some information regarding any of the random variables v_i. The information might be perfect so that we know the exact value of v_k. In this case, the exact value of v_k is used and v_k is omitted in the joint probability density $f_{v_i}(v_i|S)$. Or, the information might be imperfect so that the joint probability density function is altered using the newly gained knowledge. In any case, the value of information regarding v_k is given by

$$V(v_k) = \underset{b}{Max}\ E(u|v_k,b,S) - \underset{b}{Max}\ E(u|b,S)$$

Since we can alter b freely, we can use a b value according to our knowledge on v_k. Although it appears as if $V(v_k)$ can be negative, this is not the case. Even though it was not stated, the negative value of $v(v_k)$ implies that we had very limited knowledge about v_k prior to obtaining information, hence, our first calculation of $E(u|S)$ was vastly erroneous. Therefore, using $\left|V(v_k)\right|$ is more appropriate.

We see that using information value theory, we can calculate how much we learn of a system by unraveling a certain piece of information. We now can quantify the information lost by erasure in the codeword using concepts presented above.

3.3 Information Value in Block Codes

Investigating the block codes is a good starting point to examine the value of bits in all codes. They have separated and usually unrelated blocks that can be treated independently. Moreover, their error detection and correction mechanisms are simpler compared to other codes. Hence, we will use block codes as our starting point to quantify the information value associated with unraveling a certain bit in a received vector.

A block code has n information and k parity bits. Clearly, some of the errors that might occur in the channel can be corrected or detected using the parity bits. This correction is based on the idea that a block code is able to correct r errors if there is no overlap among the spheres centered at code vectors with a radius r in the $n + k$ dimensional space. If a vector V in the receiver end of the channel (received vector) is in a sphere, the information can be uniquely decoded. Otherwise, we can guess which code vectors might be implied by the sender using the available information. Note that, even if we uniquely decode a vector some uncertainty remains unless the channel is perfect.

Now, we introduce erasures and errors to the channel, where the received vectors $\{V_1...V_i\}$ may have some omitted and/or erroneous bits. Let us assume an extra ordinary scenario where a malicious identity, Eve, tampers the set of data by erasing only a limited number of bits. How can Eve achieve the most damage without changing the number of erased bits?

Here, we suggest that a good received vector has less entropy, so that there remains little uncertainty of the codeword sent. Even if the transmission seems exact, entropy is never zero, unless the channel is perfect. For example, consider the standard

generating matrix for [7,4] Hamming Code:

$$G = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,0 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix} \tag{3.1}$$

Obviously, $V_0 = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,0 \end{pmatrix}$ is a code word. Assume a very erroneous binary symmetric channel with error probability $e = 0.1$. Then, even if the received vector is exactly V_0, the probability that V_0 being the intended message by the sender is $f_0 = 0.989$. Similarly probability of intended message being $V_1 = \begin{pmatrix} 0\,0\,0\,1\,1\,1\,1 \end{pmatrix}$ becomes $f_1 = 0.000150$ and probability of being $V_2 = \begin{pmatrix} 0\,0\,1\,0\,0\,1\,1 \end{pmatrix}$ is $f_2 = 0.00136$ and so on.

As stated in the Information Value Theory chapter, we aim to maximize our gain with our current knowledge. Assume that for the generator matrix stated above, there are 16 different decision corresponding with each of the codewords. Further, assume that the receiver chooses one of the decisions according to the probability of each matching codeword. The expected gain can be defined as $T = \sum_i f_i S_i$, where S_i is the consequence of the decision corresponding to the codeword C_i and f is the probability mass function (pmf) of the codewords, i.e., f_i is the probability of the received vector V being sent as C_i. f can be calculated for a received vector V for a codebook C as

$$f_i = P(C_i|V) = \frac{P(C_i \cap V)}{P(V)} = \frac{P(V|C_i)P(C_i)}{\sum_i P(V|C_i)P(C_i)} \tag{3.2}$$

Note that S_i could be anything depending on the context and different values of the function S_i do not have to have the same unit(s).

Now Eve wants to tamper the data so that the pmf of the codewords f alters.

Using Information Value Theory, we can calculate the damage of the tampering as:

$$\begin{aligned} D &= \Delta T \\ &= E(S|V) - E(S|V') \\ &= \sum_i f_i S_i - \sum_i f_i' S_i \end{aligned}$$

where V' is the received vector and f_i' is the pmf of the codewords after tampering. Now we can concentrate on the effect of the erasing a single bit. Assume that f^q is the pmf and V_q is the remaining vector after the erasure of the q^{th} bit. Hence the value of the q^{th} bit is:

$$D_q = E(S|V) - E(S|V_q) \tag{3.3}$$

$$= \sum_i f_i S_i - \sum_i f_i^q S_i \tag{3.4}$$

To find the value of a bit from the damage, we need to consider that unraveling and erasing bits are similar actions.

Proposition 1. *Unraveling a bit is the exact opposite of erasing a bit, and "information value" of a bit is negative of the damage caused by erasure of the bit.*

Hence $-D_q$ gives us the priority of the q^{th} bit of the received vector.

3.4 Bitwise Information Value in Block Codes

In the previous section, we quantified the information value of a bit in regards of the difference in the weighted sum of the consequences between the tampered and original received vectors. In this section, we will develop an information theoretic quantification for priority. Such a measure must answer the question of how many bits of information are required to fix the tampered data.

3.4.1 Tampering with Erasures

For the sake of argument, let us assume that Eve can tamper the data only by erasures. Further, even after the tampering, we assume that the codeword sent by the receiver is one of the highest probability codewords in the pmf. In this case, an information value quantification measure must have the following properties:

- Its unit must be bits.
- If tampered bit(s) do not change the pmf of the codewords, the information value of the tampering is zero.
- If the pmf of the codewords becomes a constant function, the information value of tampering is at most equal to k for an (n, k) block code with k information vectors and equality is achieved for perfect channels.
- Information values corresponding to alterations in different received vectors must be additive.

As we can see, as the uncertainty increases, the damage by the tampering increases. This is due to the fact that if uncertainty is large, we cannot be sure if the decoded message is the same as the message encoded by the sender.

Entropy, i.e., uncertainty associated with f is equal to $-\sum_i f_i log f_i$. Eve must make sure that the uncertainty in $\{V_i\}$ after tampering must be maximum. Thus, we propose that we can measure the damage, or the information value, using the difference between the entropy values of the codewords before and after the tampering. Therefore,

$$D = \sum_i f(i) log f(i) - \sum_i f'(i) log f'(i)$$

$$D = H(C|V') - H(C|V) \tag{3.5}$$

where f is the original pmf of encoded vectors, f' is pmf of encoded vectors after tampering and C is the set of codewords.

Note that such a measure satisfies all the properties listed above. Further note that this measurement is similar to the measurement suggested in the previous section, only with consequences S_i are changed to $\log f(i)$ so that the measurement result is in bits.

3.4.2 Tampering with Altering

Now, let us remove the restriction on Eve so that she can change bits as well as erase them. Assume that the codeword C_i is sent by the transmitter and V is read by the receiver. Eve tampers V to V'. In this case, measuring the difference between entropy values before and after tampering does not provide us accurate results, as Eve can reduce the entropy by making V' closer to another codeword than C_i. It is obvious that the damage in this case is more than the damage performed by erasing the whole message, which is bound by k in an (n, k) block code. Therefore it exceeds the upper bound suggested for the tampering with erasures.

We proposed above that unraveling and erasing bits are similar actions, i.e., the damage due to erasure is equal to the negative of the information gained by unraveling. We can use proposition 1 and consider tampering of V to V_i in two steps. Assume first that Eve erased all the bits in V and then changed it back to V, i.e.,

$$V \longrightarrow_1 0 \longrightarrow_2 V'$$

where 0 is the null vector. It is obvious that the damage by the first operation is $D_1 = k - \sum_i f_i \log f_i$.

We can calculate the change due to the second step using proposition one. Assume that the received vector is null and some of the bits are unraveled so that it becomes V'. The information value of such a change can be calculated as $D_2 = k - \sum_i f'_i \log f'_i$.

However, since in this case V' is closer to some other codeword than the intended codeword C_i, this change has to be considered negative, i.e., there is a further loss of information in the second step.

As we listed in the properties required at the beginning of the section, information values due to alterations on different received vectors must be additive. Since we separated the tampering from V to V' into two different alterations on two different vectors, we can add the information value of the results together to obtain the damage due to the tampering.

$$D = D_1 + D_2 = 2k - \sum_i f_i \log f_i - \sum_i f'_i \log f'_i$$

$$D = 2k - H(C|V) - H(C|V') \tag{3.6}$$

Note that, damage due to tampering can be negative, if the tampering causes the pmf of the codewords further peak at the intended codeword by the transmitter. Hence the bitwise information value of damage on a received vector is in the interval $[-2k, 2k]$.

3.5 Bit Priority in Block Codes

In the previous section, we established how we can measure the damage done by tampering of block codes in bits. Now we will move to investigate effects of tampering a single bit in block codes. We will first examine effects of erasure in a single bit and then move continue with alterations, as we have done in the previous section.

3.5.1 Tampering with Erasures

Assume that the q^{th} bit of the received vector V is erased. Let us call the remaining vector V_q. The damage can then be expressed as the difference in the entropy values

after erasure of bit q, i.e.,

$$D_q = \sum_i f_i' \log f_i' - \sum_i f_i \log f_i$$

$$D_q = H(C_i|V) - H(C_i|V_q) \tag{3.7}$$

Now, Eve plans to erase only one bit to tamper the data. Erasing which bit will increase the entropy most, i.e., the decodability of the received vector? The answer to this question lies in the bitwise damage values of erasing different bits in a codeword. The generator matrices are chosen to offer similar protection for each bit, however erasures and errors in the message may increase the priority of one bit over the other.

Consider a received vector $V_0 = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,0 \end{pmatrix}$ prepared with the standard generating matrix for [7,4] Hamming Code, given in equation 1. Even though it is one of the codewords in the codebook, there still remains some uncertainty as the channel is not perfect. Assume a very erroneous channel with $e = 0.1$ to dramatize the effects of bit erasure. In this case, even if no bit is erased, entropy remained after receiving V_0 is $H_0 = 0.119$. When any of the bits are erased, the entropy increases to $H_{0i} = 0.397$, meaning the information value of erasure of any of the bits of V_0 is $\Delta H_i = H_{0i} - H_0 = 0.278$ for $i = 1 \ldots 7$.

Assume if we have a few vectors, all of which are perfectly transmitted except erasures. For the sake of argument we can choose them all the null vector, i.e., V_0 as the complete null vector, $V_1 = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,\_ \end{pmatrix}$, $V_2 = \begin{pmatrix} 0\,0\,0\,0\,0\,\_\,\_ \end{pmatrix}$, $V_3 = \begin{pmatrix} 0\,0\,0\,0\,\_\,\_\,\_ \end{pmatrix}$ and so on. Attacking which of these vectors will deal the most damage on the entire message?

We calculated the entropy increase in V_i for $i = 1 \ldots 6$ due to deletion of one more bit using $\Delta H_i = H_i^{final} - H_i^{initial}$ for a channel with $e = 0.1$, and presented the results in figure 1.

The most interesting result on figure 1 is that entropy increase is not a monotonous increasing function of the number or previously erased bits. There might be some cases

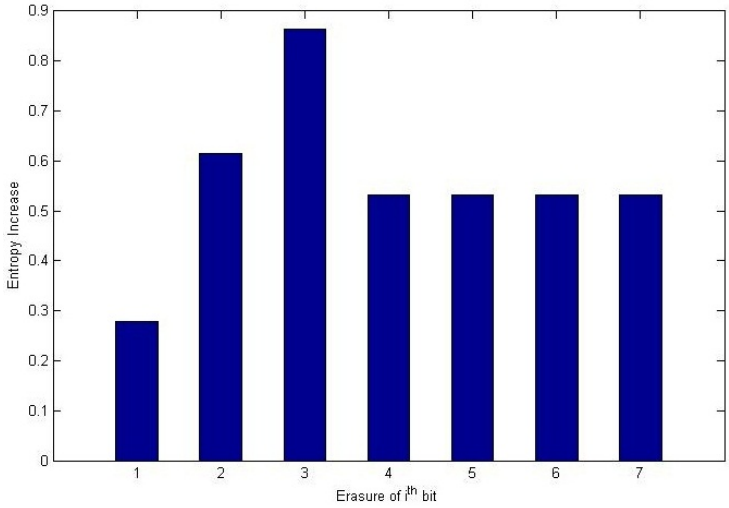

Figure 3.1: Entropy increase due to removing one more bit from V after $i-1$ bits are removed

that erasing one more bit of a more tampered received vector may not cause as much damage as erasing a bit from a less tampered vector.

Investigating the Hamming Code in figure 1, removal of the third bit after the first two bits effects the system more than that of the fourth bit after the first three bits; i.e., information associated with the first parity bit is more than one of the information bits. Therefore, if Eve wants to tamper two vectors belonging to standard [7,4] Hamming Code codebook with only three erasures, to obtain maximum damage, she should erase all of them from the same codeword and do no tampering on the other received vector. But in case of six erasures she must distribute the six erasures to two codewords equally to attain the maximum damage, rather than erasing them all from a single received vector.

Another interesting result is that the bitwise priorities values of the last four bits are all equal to the channel capacity $C = 1 - H(e) = 0.531$, for a binary symmetric channel of error probability $e = 0.1$. This is not a coincidence and we shall prove it

Table 3.1: Entropy of removing one more bit after n bits are removed for error-free vectors in [7,4] Hamming Code

bits removed	$e = 0.1$	$e = 0.01$	$e = 0.001$
0	0.278	0.00452	6.43×10^{-5}
1	0.614	0.0851	0.0115
2	0.865	0.233	0.341
3	0.531	0.919	0.989

below.

Lemma 1. *After parity bits are removed, bitwise priorities of all remaining bits are equal to the capacity of the channel.*

Proof. It directly follows the definiton of channel capacity for the i^{th} bit as $I_i = \Delta H = H(C|V^i) - H(C|V)$, where I_i is the capacity for the i^{th} bit and V^i is the received vector V with i^{th} bit erased. □

Note that there is no restriction for the channels which have different error rates for different bits. Only the error rate for the examined bit is important, i.e., if each bit has different channel characteristics their bitwise priorities will still be equal to the capacity of the channel they are transmitted in.

In figure 1, we have seen that although it is not a monotonous increasing function, the uncertainty of received vectors tend to be greater if they already have erasures. Therefore, in most cases, Eve must first attack received vectors already with erasures. In Table 3.1 and 3.2, the entropy values associated with removing one more bit after n bits have been removed in received vectors is presented with respect to different error probabilities for a [7,4] and [7,3] Hamming Code respectively.

Table 3.2: Entropy of removing one more bit after n bits are removed for error-free vectors in [7,3] Block Code

bits removed	$e = 0.1$	$e = 0.01$	$e = 0.001$
0	8.68×10^{-5}	1.26×10^{-7}	1.65×10^{-10}
1	3.00×10^{-3}	4.28×10^{-5}	5.60×10^{-7}
2	8.22×10^{-2}	1.14×10^{-2}	1.47×10^{-3}
3	0.157	2.27×10^{-2}	2.95×10^{-3}

Table 3.1 and 3.2 offer interesting results. First of all, bitwise entropy values for any bits are positive, tentatively meaning that a correct information on the nature of a bit usually improves the quality and decreases the entropy. We present the formal proof of this statement below.

Theorem 1. *Entropy value of any bit is non-negative, if there exists only erasures on the received vector.*

Proof. $V = (v_1 \dots v_q \dots v_n)$ and $V_q = (v_1 \dots \_ \dots v_n)$, where some of the v_is may be blank in both of them but only v_q exists in V and blank in V_q. It is obvious that $H(C_i|V_q) = H(C_i|(v_1 \dots v_q \dots v_n)) + H(C_i|(v_1 \dots v_q{}' \dots v_n))$.

Using Equation 3;

$$\begin{aligned} D_q &= H(C_i|V) - H(C_i|V_q) \\ &= H(C_i|V) - \{H(C_i|V) + H(C_i|(v_1 \dots v_q' \dots v_n))\} \\ &= -H(C_i|(v_1 \dots v_q' \dots v_n). \end{aligned}$$

We know that $H \geq 0$ for any distribution. Hence $D_q \leq 0$. Since D_q is the damage done by removal of q^{th} bit, entropy value of q^{th} bit is non-negative. □

Secondly, sum of entropies of all the bits added with the entropy remained in case of no erasure gives us 4, i.e., k, the total number of information bits in [n,k] Hamming Code. A more general statement for this observation is stated below.

Theorem 2. *Sum of entropy value of all bits and the entropy of the codeword is equal to the entropy of the codebook.*

Proof. The generator, transmitter, channel and receiver constitutes an isolated system. Entropy in an isolated system is conserved. Therefore total entropy is equal to the entropy in the generator, i.e., entropy of the codebook. □

3.5.2 Tampering with Altering

Now consider the vector $V_0 = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,1 \end{pmatrix}$. Obviously it is not in the codebook. Even though there is a high chance that it is V_0, the entropy is $H = 1.502$ for the channel with $e = 0.1$. Calculating the entropy increases by deletion of any bits we obtain:

$$H_i = \begin{cases} 0.531 & \text{if } i \neq 7 \\ -1.105 & \text{if } i = 7 \end{cases}$$

The negative entropy increase implies that by erasing that bit, which was already erroneous actually helps decoding the received vector.

Now we can calculate the entropy increase due to altering a bit. As we stated in the derivation of equation 6, if the most probable codeword changes due to the alterations, we need to assume all the information in the received vector is lost and some misguiding (negative) information is added.

Consider the received vector $V = \begin{pmatrix} 0\,0\,0\,0\,0\,1\,0 \end{pmatrix}$ transmitted in a channel with $e = 0.1$ as in our previous examples. The entropy of the codeword distribution is 1.502 and the most probable codeword is the null vector with $p = 0.720$. Eve alters this codeword to $V' = \begin{pmatrix} 0\,0\,0\,0\,1\,1\,0 \end{pmatrix}$. The most probable codeword becomes

$C = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,0 \end{pmatrix}$ again with $p = 0.720$ and the entropy of the codeword distribution remains unchanged. The priority of changing fifth bit from 0 to 1 is then calculated by summing the priority of erasing all data from the vector V and the priority of adding misguiding information, which is equal to 3.004.

3.5.3 Tampering a Vector with Non-Standard Generating Matrix

We have seen that the priority of deleting any bit is the same for the standard generating matrix for [7,4] and [7,3] Hamming Codes. However we do not have to use a standard generating matrix. If we use a custom generating matrix, bits might not have the same importance, i.e., the same priority. Consider a custom generator matrix, G' given below.

$$G' = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix}$$

Bitwise entropies of the information bits are presented in Figure 2. Note that the fourth and fifth bits have less entropy, i.e., less priority than other bits, due to being repeated by parity bits. Similarly, the first bit has the maximum priority as it is not included in any of the other bits; hence receives no protection from parity bits.

We can see that removal of some bits has less entropy associated with them, hence, their removal is not urgent to corrupt a communication system. Therefore, Eve must concentrate on bits with higher priority i.e., the first and second bits.

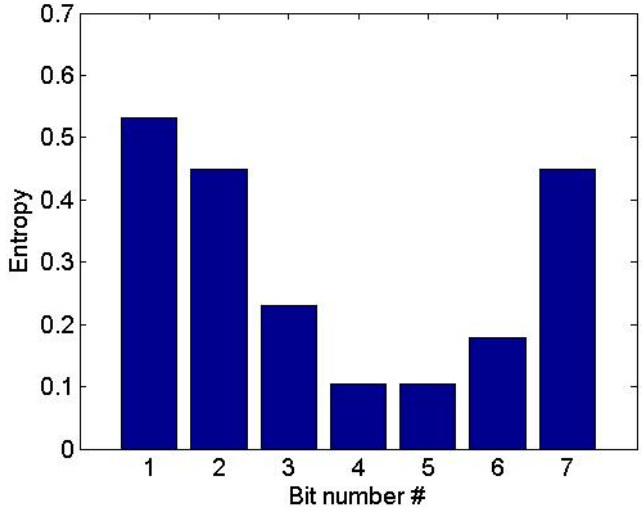

Figure 3.2: Entropy gained due to removal of n^{th} bit for a block code with generator matrix G'

3.6 Code Protection and Risk

We have stated that regardless of the error correction mechanism, there remains some uncertainty, i.e., entropy, of the received message due to the probable errors in the channel. In case of block codes, the parity bits reduces the entropy on the received message. We claim that the total amount of entropy reduced by the parity bits is the protection offered to the message. In other words, for an [n,k] Hamming Code, the protection offered by the parity bits is:

$$P = H(V_k) - H(V_n) = \sum_{i=k+1}^{n} D_i$$

where V_n is any codevector, V_k is the information vector D_i is the priority of the i^{th} bit in the code. Protection offered by [7,4] and [15,11] Hamming Codes for different channels is presented in Table III. Note that protection depends on the error probability of the channel as well as coding scheme.

Table 3.3: Protection Offered by different Block Codes

Block Code	$e = 0.01$	$e = 0.001$	$e = 0.0001$
$[7, 3]$	0.242	3.42×10^{-2}	4.42×10^{-3}
$[7, 4]$	0.323	4.563×10^{-2}	5.892×10^{-3}
$[7, 4]'$	0.006	8.566×10^{-5}	1.121×10^{-6}
$[15, 9]$	0.727	0.102	1.32×10^{-2}
$[15, 10]$	0.807	0.114	1.47×10^{-2}
$[15, 11]$	0.888	0.125	1.62×10^{-2}

In some coding structures, information bits and parity bits may not be separated. Some block code with non-standard generating matrices can be such structures. To calculate the protection in such codes, we need to consider the most prior k bits as information bits for an [n,k] block code. Note that in Table III $[7, 4]'$ is formed with the non-standard generating matrix discussed in the previous section.

Even employing the parity bits do not reduce the uncertainty in the received message to zero. There still remains some risk for the code to be misinterpreted by the receiver. We call the remaining uncertainty as error entropy, H_e, of the code.

We also note that if a codeword with a larger information vector is corrupted, more information is lost compared to a codeword with smaller information length. Hence we can claim that the risk associated with codewords having longer information blocks in case of corruption is higher, due to the fact that a larger message will be lost. To measure the risk of information loss, R, of the code, we simply multiply the error entropy with the information vector length; k value of an [n,k] code.

$$R = kH_e = kH(V_n)$$

Table 3.4: Risk of Different Block Codes

Block Code	$e = 0.01$	$e = 0.001$	$e = 0.0001$
$[7, 3]$	6.112×10^{-6}	8.708×10^{-10}	1.150×10^{-13}
$[7, 4]$	6.237×10^{-4}	8.812×10^{-7}	1.157×10^{-9}
$[7, 4]'$	0.3294	0.0457	0.00590
$[15, 9]$	1.031×10^{-3}	1.420×10^{-6}	1.860×10^{-9}
$[15, 10]$	3.957×10^{-4}	4.354×10^{-8}	5.751×10^{-12}
$[15, 11]$	8.806×10^{-3}	1.215×10^{-5}	1.591×10^{-8}

Such a measure must satisfy an important boundary condition in case of perfect transmission. If the channel is error free, risk of the code must be zero. It is obvious that, for channels with $e = 0$, the risk becomes zero.

The risk of different coding schemes are presented in table IV. We can see that there is a trade-off between the risk and the rate, as in the protection. It is obvious from the results that for error prone channels, shorter codewords must be used.

3.7 Conclusion

In this paper we proposed different measures for code protection in block codes using the bitwise entropy values. Even though the use of block codes is limited, any measure proposed in this work can be used for any different coding scheme, and can be used to compare them.

Receiving or destroying whole messages is much easier, rather than corrupting specific bits. In fact, classically, receiving messages does not require any computational power or cause time delay. However, if the computational power required to read the bits would be higher than the power to calculate entropies, or if there was a

time delay involvement with receiving whole messages, such knowledge would be very helpful. In quantum communication architectures, measuring the qubits can introduce time delays, and classical probability calculations may be useful to undercover more context with less protection in a limited amount of time.

A similar case is when the power constraints of the transmitter is extremely tight. In such a situation, transmitter mush choose the more prior bits to send and expect the receiver to have the enough resources and background knowledge to successfully decode the received message.

We also make use of risk and protection measurements to offer similar protection for the content we transmit. For example, if we are sending a document and its password through the same channel, we can make sure that password is protected strong enough.

In this work, we concentrated solely on block codes as block codes are much easier to analyze than other coding schemes. Due to computational limitations, only [7,4] and [15,11] codes are investigated. For a future study, code protection for concatenated codes and convolutional codes may be studied. Also, as a later work, burst error protection and burst error risks should be analyzed as well.

Chapter 4

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this section, we sum up the contributions of each chapter and underline the important results.

4.1 Contributions

4.1.1 Quantum Memory Management Systems

Quantum Computation, even though first suggested thirty years ago, is still at its infancy. It is shown that a fully fledged quantum computer may change the course of computation altogether, mainly in cryptography. In spite of its immense computational capacities, quantum computers lack adequate memory management systems yet. For this reason, we build the first half of this thesis on quantum memory management systems. We first suggested important parameters to analyze how a good quantum memory management system must work. These parameters were all designed to measure how the memory system would behave at increasing memory size. Later we suggested several quantum memory management systems. According to the criteria we established, some of them rendered impracticable to build, while some others were more practical.

In brief, our analysis showed that quantum memory management systems with teleportation are not more efficient then systems without teleportation. We also concluded that Logarithmic, Mixed and Concatenated Logarithmic Swapping techniques

are all more feasible than others.

4.1.2 Bitwise Information Priority Measure

In this study, we proposed a novel measure, to distinguish the importance of bits in a communication system. We suggested that some bits may be more important than others and receiver side must put priority on retrieving these. This is especially important on quantum communications, where classical error correction systems do not work and measuring the qubits may take more considerably more time than making probabilistic calculations. We also extended our work to two novel code performance measures, code protection and risk.

4.2 Future Research Directions

The specific hardware mechanics of a quantum computer is still unknown, as quantum computers are only operated in laboratories and very limited operations are ever performed on them. Later developments may introduce new parameters for a quantum memory management system. Therefore, this study may be revisited in the future on the lights of a new physical developments in quantum computation.

Our analysis in Bitwise Information Priority Measure was limited only to block codes with small codewords due to computational capacity. Moreover, stronger computers are unlikely change this fact as required computer capacity to analyze a code increases exponentially for larger codes. An approximate calculations approach should be devised and employed for larger codes including convolutional codes.

REFERENCES

[1] R. Feynman "Simulating Physics with Computers," *Internat. J. Theoret. Phys.,* 21, pp. 467 - 488. 1982.

[2] M. Oskin, F. T. Chong, I. L. Chuang. "A Practical Architecture for Reliable Quantum Computers," *IEEE Computer* vol 35, no 1, pp.79-87 January 2002.

[3] A. Einstein, N. Rosen and B. Podolsky, "Can quantum mechanical description of physical reality be considered complete?", *Phys. Rev*, vol 44, p. 777, 1935.

[4] J. S. Bell., "On Einstein Podolsky Rosen Paradox", *Phys.*, vol 1, no 3, p. 195-200, 1964.

[5] J. S. Bell., "On the impossible pilot wave," *Foundations of Physics*, vol. 12, pp. 989-999, 1982.

[6] L. Grover., "A fast quantum mechanical algorithm for database search," *Proc. 28th Ann. ACM Symp. Theory of Computation, ACM Press*, pp. 212-219, 1996.

[7] P. Shor., "Algorithms for quantum computation: Discrete logarithms and factoring," *Proc 35th Ann. Symp. Foundations of Computer Science. IEEE CS Press*, p. 125, 1994.

[8] P. Rabl, D. DeMille and J. M. Doyle at al., "Hybrid quantum processors: Molecular ensembles as quantum memory for solid state circuits", *Phys. Rev. Lett*, vol. 97 pp. 1869-76, July 2006.

[9] L. DiCarlo, J. M. Chow, J. M. Gambetta et al, "Demonstration of two qubit algorithms with a superconducting quantum processor," *Nature*, vol. 260, pp. 240-244, 2009.

[10] M. K. Thomsen, R. Glck, H. B. Axelsen, "Reversible arithmetic logic unit for quantum arithmetic", *Jour. Phys. A*, vol. 42, no. 38, p. 2002, 2010.

[11] A. Vetteh, K. Walus, G. A. Jullien, et al. "Quantum dot cellular automata carry-look-ahead adder and barrel shifter", *IEEE Emerging Telecommunications Technologies Conference*, 2002.

[12] H. Haffner, C. F. Roos, R. Blatt, "Quantum computing with trapped ions", 2008. arXiv:quant.ph/0809.4368v1

[13] B. Atakan and O. B. Akan, "Deterministic capacity information ow in molecular nanonetworks", *Nano Communication Networks*, vol. 1, pp. 31-42, 2010.

[14] E. E. Narimanov and P. Mitra, "The channel capacity of a Fiber Op- tics Communication System: Perturbation Theory", *Journal of Lightwave Technology*, vol. 20, no 3, pp.530- 2002.

[15] P. Hausladen, R. Jozsa, B. Schumacher et al., "Classical Information Capacity of a Quantum Channel", *Phys. Rev. A* 54 pp. 1869-76, 1996.

[16] D. S. Ornstein and B. Weiss. "Entropy and Data Compression Schemes", *IEEE Trans. Inform. Theory*, vol. 39, pp. 7883, Jan. 1993.

[17] R. A. Howard, "Information Value Theory", *IEEE Trans. System Science and Cybernetics*, vol, SSC 2, no. 1, pp. 22-26, 1966.

[18] K. P. Claxton and J. Mark., "Using Value of Information Analysis to Prioritise Health Research: Some Lessons from Recent UK Experience", *Pharmacoeconomics.*, vol 24, no 11, pp. 1055-1068, 2006.

[19] R. B. Bratvold, J. E. Bickel and H. P. Lohne. "Value of Information in the Oil and Gas Industry: Past, Present, and Future," *SPE Reservoir Evauluation and Engineering*, vol 12, no 4, pp. 630-638, 2009.

[20] D. Polani, T. Martinez, J. Kim, "An Information Theoretic Approach for Quantication of Relevance", *Proc. 6th European Conference on Articial Life*, 2001.