On Quantum Codes and Networks

Colin Michael Wilmott

Technical Report
RHUL–MA–2009–11
23 February 2009

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

ON QUANTUM CODES AND NETWORKS

Colin Michael Wilmott

Royal Holloway and Bedford New College,
University of London

Thesis submitted to
The University of London
for the degree of
Doctor of Philosophy
2008.

Declaration

The material contained herein is the result of my own studies. I declare that foregoing is true and correct.

Acknowledgements

It is a real pleasure to acknowledge the help that I have received in bringing this thesis to fruition. My supervisor Professor Peter Wild displayed patience, understanding and insight during our many hours of discussion. To my family and friends for their welcome distraction and source of excellent conversation. To these, I offer my sincere thanks for their sustained support at all stages. My graduate career was supported by an Engineering and Physical Sciences Research Council Fellowship.

Abstract

The concern of quantum computation is the computation of quantum phenomena as observed in Nature. A prerequisite for the attainment of such computation is a set of unitary transformations that describe the operational process within the quantum system. Since operational transformations inevitably interact with elements outside of the quantum system, we therefore have quantum gate evolutions determined with less than absolute precision. Consequently, the theory of quantum error-correction has developed to meet this difficulty. Further, it is increasingly evident that much effort is being made into finding efficient quantum circuits in the sense that for a library of realisable quantum gates there is no smaller circuit that achieves the same task with the same library of gates. A reason for this concerted effort is primarily due to the principle of decoherence. I give the construction for the set of unitary transformations that describe an error model that acts on a **d**-dimensional quantum system. I also give an overview of the theoretical framework associated with such unitary transformations and generalise results to cater for **d**-dimensional quantum states. I introduce two quantum gate constructions that generalises the qubit SWAP gate to higher dimensions. The first of these constructions is the WilNOT gate and the second is an efficient design also based on binomial summations. Both of these constructions yield a quantum qu**d**it SWAP gate determined only in the CNOT gate. Furthermore, the task of constructing generalised SWAP gates based on transpositions of qu**d**it states is argued in terms of the signature of a permutation. Based on this argument, we show that circuit architectures completely described by instances

of the CNOT gate can not implement a transposition of a pair of qu**d**its over dimensions $\mathbf{d} = 3 \bmod 4$. Consequently, our quantum circuits are of interest because it is not possible to implement a SWAP of qutrits by a sequence of transpositions of qutrits if only CNOT gates are used. I also give bounds on numbers of quantum codes predicated on **d**-dimensional quantum systems, and generalise the encoding and decoding architectures for qu**d**it codes.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The desire to comprehend philosophies at the edge of possibility continues to be a source of advancement today as it has been at any other time in history. If such a premise is taken to be a departure point in the challenge to extend the boundaries of knowledge then theoretical and technical innovation will abide. The Theory of Quantum Computation continues to advance our understanding of information as established in the seminal work of Shannon through an innovative analysis of the nature of noise. This development of a quantum mechanical computing framework has redefined quantum computation and inspires discoveries whose very nature lie at the frontier of reality.

Modern computing begins with the pioneering work of Charles Babbage and Alan Turing. An analytical machine put forward by Babbage conceived the principle on which modern computing rests. Over a century later, Turing improved upon the ideas of Babbage by devising a programmable means that would become a basis for the description of computing logic. This model of computation was then strengthened to illustrate the universal nature. This became known as the *Church-Turing thesis* [89] and acknowledges the equivalence between the principles of the Turing machine and the work of Alonzo Church in lambda calculus. However, it is the nature of innovation to challenge

common perception, and challenges from different forms of efficient computation coupled with the emergence of randomised algorithmic protocols have recast the Church-Turing thesis. These challenges may be seen to either strengthen the position of modern computing or illustrate the limitations in which modern computing is envisaged.

Quantum mechanics offers a new direction in the field of computation with an interpretation of system more profound than the classical interpretation of computation. Whereas the state of a classical computation can be described in terms of a set of observable elements, the state of a quantum computation can not be observed but rather is interpreted through a wave function associated with the system. The promise of a quantum computer rests with the complexity advantage it has over its classical counterpart. This was first mooted by Richard Feynman [29] in which he suggests that the intractable nature of using classical computers to describe quantum phenonmen might be overcome were one to design a computer that used quantum mechanical effects. In 1985 David Deutsch [24] proposed a model of a quantum computer that gave credence to Feynman's conjecture. The insights of Feynman and Deutsch into the theory of quantum compuation mirror the contributions made by Babbage and Turing to classical computation.

The task of constructing a quantum computer is predicated on firstly realising the inherent processing advantage of quantum compuation over its classical analogue and secondly, and more importantly, on controlling the sensitive quantum interference effects that explain the source of its computational power. However quantum computations also produce interactions between sensitive quantum information and noise in the system and this interaction results in decoherence, an outcome that destroys quantum information. Decoherence is an inevitable feature of quantum computation, and therefore, it is

of fundamental importance that any coupling between information and noise be controlled to within a suitable degree of precision. A number of proposals have emerged in recent years that put forth descriptions for a complete physical model for quantum computation. The first proposal is the Linear Ion Trap [17] which stores quantum information in a well-defined way. This technique prepares two distinct states in a trap of an electric field by targeting each state with a series of laser pulses. A second proposal uses Nuclear Magnetic Resonance [37], NMR, to measure the average nuclear spin state of atomic nuclei. Cavity Quantum Electrodynamics [87], QED, is another consideration whereby photons assume quantum state representations which together with cavity QED techniques permit the preparation of a quantum state.

Chapter 2 introduces a quantum channel that permits the transmission of a high dimensional quantum information state called a qu**d**it in the general dimension **d**. This is followed by a study on noise associated with such qu**d**it information states. I then discuss the theoretical nature of the corresponding error model in chapter 3 together with construction techniques. Chapter 4 describes the stabilizer formalism, a network to encode stabilizer codes and provides a result analogous to the number of distinct bases of a classical code. Chapter 5 gives an overview of quantum computation and introduces the question of constructing a generalised quantum SWAP gate. Chapter 6 introduces the first of our quantum gate constructions, the WilNOT gate, that describes the construction of a generalised quantum SWAP gate. This is followed by a second gate construction that describes an efficient generalised quantum SWAP gate. In this instance, efficiency is determined by the fewest use of the controlled-NOT gate that is required to compute the task. Finally, I discuss a number of qu**d**it codes in chapter 7 along with a result on the complexity associated with the encoding of a qu**d**it code.

1.1 Introduction to Quantum Mechanics

The consideration of information theory is the efficient transmission of information from sender to receiver. Classical information is recorded in a sequence of states over some finite alphabet Σ where it is most often maintained that $\Sigma = \{0, 1\}$. The classical unit of information is represented by a *bit* over the finite field $\mathbb{F}_2$. Quantum information theory is constructed under an analogous concept. The single state of a quantum system over $\mathbb{C}^\Sigma$ is called a *ket* and is represented by a *qubit* over $\mathbb{C}^2$ which is associated with the *computational basis states* $|0\rangle$ and $|1\rangle$ whereby

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1.1.1}$$

with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. If both α and β are non-zero then the state $|\psi\rangle$ is a *superposition* state with amplitudes α and β. The information source of a classical system is described by a set of probabilites and similarly, a quantum information source associates a set of probabilites, described by probability amplitudes $|\alpha_i|^2$, to basis states $|\psi_i\rangle$. Furthermore, while an n-dimensional state in $\mathbb{F}_2^n$ is required to represent n bits of classical information, a 2^n dimensional state in $\mathbb{C}^{2^n}$ is needed to describe n qubits. Therefore, the quantum system must remember 2^n complex numbers for an n qubit state, and it is this property of quantum system that is used in quantum information theory.

Let $\{|\psi_i\rangle\,, i \in \Sigma\}$ be a basis for $\mathbb{C}^\Sigma$ for which $|\psi\rangle = \sum_{i\in\Sigma} \alpha_i\,|\psi_i\rangle$. Then $\mathbb{C}^\Sigma$ explains span and independence, however we require the concepts of angle and length between vectors to endow the vector space $\mathbb{C}^\Sigma$ with geometry. A *linear functional* on a vector space is a scalar-valued function γ defined for

every vector $|\zeta\rangle$ and $|\eta\rangle$ and scalars α_1 and α_2 with the property that

$$\gamma(\alpha_1 |\zeta\rangle + \alpha_2 |\eta\rangle) \;\; = \;\; \alpha_1\gamma(|\zeta\rangle) + \alpha_2\gamma(|\eta\rangle). \tag{1.1.2}$$

To every vector space $\mathbb{C}^\Sigma$, we have a corresponding *dual space* $\left(\mathbb{C}^\Sigma\right)^\perp$ consisting of all linear functionals on $\mathbb{C}^\Sigma$. The single state of a quantum system over $\mathbb{C}^{\Sigma\perp}$ is called a *bra*. In particular, if $\{|\psi_i\rangle\,,\ i \in \Sigma\}$ is a basis in $\mathbb{C}^\Sigma$, then there is a uniquely determined basis $\{|\psi_j\rangle^\dagger\,,\ j \in \Sigma\}$ in $\left(\mathbb{C}^\Sigma\right)^\perp$ such that the linear functional $\psi_j(|\psi_i\rangle)$ is identically $|\psi_j\rangle^\dagger(|\psi_i\rangle) = \delta_{ij}$. In the language of Dirac, the action of the conjugate linear functional $\psi_j(|\psi_i\rangle)$ on $\mathbb{C}^\Sigma$ is written and defined to be $\langle\psi_j|\psi_i\rangle = \delta_{ij}$. To each $|\gamma\rangle = \sum_{j\in\Sigma}\beta_j\,|\psi_j\rangle$ and $|\psi\rangle = \sum_{i\in\Sigma}\alpha_i\,|\psi_i\rangle$ the value of $\gamma(|\psi\rangle)$ is determined by

$$\gamma(|\psi\rangle) = \langle\gamma|\psi\rangle = \sum_{i,j\in\Sigma}\beta_j^*\alpha_i\langle\psi_j|\psi_i\rangle = \sum_{i,j\in\Sigma}\beta_j^*\alpha_i\delta_{ij} = \sum_{i\in\Sigma}\beta_i^*\alpha_i. \tag{1.1.3}$$

where * denotes complex conjugation. Consequently, we are equipped to induce a geometry on $\mathbb{C}^\Sigma$ by defining *inner product* $(|\gamma\rangle\,,|\psi\rangle)$ on $|\gamma\rangle$ and $|\psi\rangle$ as $\gamma(|\psi\rangle)$. A function $(|\gamma\rangle\,,|\psi\rangle)$ from $\mathbb{C}^\Sigma \times \mathbb{C}^\Sigma$ to $\mathbb{C}$ is an inner product on the vector space $\mathbb{C}^\Sigma$ if the following conditions are satisfied

1. $(|\psi\rangle\,,|\psi\rangle) \geq 0$ with equality if and only if $|\psi\rangle = 0$
2. $\left(|\gamma\rangle\,,\sum_{i\in\Sigma}\alpha_i\,|\psi_i\rangle\right) = \sum_{i\in\Sigma}\alpha_i(|\gamma\rangle\,,|\psi_i\rangle)$
3. $(|\gamma\rangle\,,|\psi\rangle) = (|\psi\rangle\,,|\gamma\rangle)^*$

Hence, an *inner product space* is a vector space with an inner product. We say $|\psi\rangle$ and $|\gamma\rangle$ are *orthogonal* if the associated inner product vanishes. Furthermore, to each inner product space we can associate a canonical form by defining the *norm* of $|\psi\rangle$

$$\|\,|\psi\rangle\,\| \;\; = \;\; \sqrt{\langle\psi|\psi\rangle}. \tag{1.1.4}$$

A *unit ket* is a state with $\| \, |\psi\rangle \, \| = 1$. The basis set $\{|\psi_i\rangle \, , \; i \in \Sigma\}$ is called *orthonormal* if for all $|\psi_i\rangle$ and $|\psi_j\rangle$ it follows that $\langle\psi_i|\psi_j\rangle = \delta_{ij}$. A vector space $\mathbb{C}^{\Sigma}$ that is an inner product space and complete with respect to the norm is then called a *Hilbert space* over $\mathcal{H} = \mathbb{C}^{\Sigma}$. Furthermore, there is only one Hilbert space in distinct dimensions up to isomorphism.

A *linear operator* M on the Hilbert space $\mathbb{C}^{\Sigma}$ is a mapping that assigns to every state $|\zeta\rangle$ in $\mathbb{C}^{\Sigma}$ a state $M\,|\zeta\rangle$ in $\mathbb{C}^{\Sigma}$, in such a way that

$$M\left(|\zeta\rangle + |\eta\rangle\right) \;\; = \;\; M\,|\zeta\rangle + M\,|\eta\rangle \, . \tag{1.1.5}$$

The study of linear operators is known to elicit a matrix representation induced by way of linear functionals. Given a basis $\{|\psi_i\rangle \, , \; i \in \Sigma\}$ in $\mathbb{C}^{\Sigma}$ with a corresponding dual set $\{\langle\psi_j| \, , \; j \in \Sigma\}$ in $\mathbb{C}^{\Sigma\perp}$, we have it that $M\,|\zeta\rangle = \sum_{i\in\Sigma} \psi_i(|\zeta\rangle)\,|\psi_i\rangle = \sum_{i\in\Sigma} \alpha_i\,|\psi_i\rangle = |\psi\rangle$. Since every state is a linear combination in $|\psi_i\rangle$, then $M\,|\zeta\rangle = M(\sum_{j\in\Sigma} \alpha_j\,|\psi_j\rangle) = \sum_{i\in\Sigma} \alpha_{ij}\,|\psi_i\rangle \, ,$ for $j \in \Sigma$. Therefore, the set of linear functionals define M within a matrix formalism where such form is called an *outer product* representation. Suppose $|\zeta\rangle$ and $|\psi\rangle$ are states in the Hilbert space $\mathbb{C}^{\Sigma}$, we define $|\psi\rangle\,\langle\zeta|$ to be the outer product operator on $\mathbb{C}^{\Sigma}$ that maps $|\zeta\rangle$ to $|\psi\rangle$ and whose action is defined by

$$M(|\zeta\rangle) = (|\psi\rangle\,\langle\zeta|)\,|\zeta\rangle \equiv |\psi\rangle\,\langle\zeta|\zeta\rangle = \langle\zeta|\zeta\rangle\,|\psi\rangle \, . \tag{1.1.6}$$

An important consequence of the outer product formalism is the *completeness relation* for orthonormal vectors. For an orthonormal basis $\{|\psi_i\rangle \, , \; i \in \Sigma\}$ for $\mathbb{C}^{\Sigma}$, and arbitrary state $|\psi\rangle$ for which $\langle\psi_i|\psi\rangle = \alpha_i$, we have it that

$$\left(\sum_{i\in\Sigma} |\psi_i\rangle\,\langle\psi_i|\right)|\psi\rangle = \sum_{i\in\Sigma} |\psi_i\rangle\,\langle\psi_i|\psi\rangle = \sum_{i\in\Sigma} \alpha_i\,|\psi_i\rangle = |\psi\rangle \, . \tag{1.1.7}$$

The equation $\sum_{i\in\Sigma} |\psi_i\rangle\,\langle\psi_i| = I$ is known as the completeness relation.

A more general linear operator describes a number of important analogs in matrix algebra. Given the Hilbert space $\mathbb{C}^{\Sigma}$, let $\{\langle\zeta|\,,\ i \in \Sigma\}$ be a set of states in the dual space $\left(\mathbb{C}^{\Sigma}\right)^{\perp}$, then for any linear operator M we have a corresponding linear functional $\zeta(M\,|\psi\rangle)$ on $\mathbb{C}$. There exists a unique linear operator $M^{\dagger}$ on $\left(\mathbb{C}^{\Sigma}\right)^{\perp}$, with the property

$$(|\zeta\rangle\,, M\,|\psi\rangle) \;\; = \;\; (M^{\dagger}\,|\zeta\rangle\,, |\psi\rangle), \tag{1.1.8}$$

for all $|\zeta\rangle$ and $|\psi\rangle$ in $\mathbb{C}^{\Sigma}$. The linear operator $M^{\dagger}$ is called the *adjoint* of M. If $M = M^{\dagger}$, then M is a *Hermitian* operator. If M is Hermitian then $MM^{\dagger} = M^{\dagger}M$. An operator M is said to be *unitary* if $M^{\dagger}M = I$. Unitary operators are an important concept in the Hilbert space because they ensure that any action by such an operator conditioned on a ket preserves the unit condition in a Hilbert space. Furthermore, if both M and N are Hermitian operators then $(NM)^{\dagger} = M^{\dagger}N^{\dagger} = MN$. If NM is Hermitian, that is, $(NM)^{\dagger} = NM$, then we require $M^{\dagger}N^{\dagger} = NM$, whence, M and N commute. The commutation property holds special resonance in quantum mechanics where the *commutator* between operators M and N is defined as

$$[M, N] \;\; = \;\; MN - NM. \tag{1.1.9}$$

The operators M and N are then said to commute if $[M, N]$ vanishes.

Let $\{|\psi_i\rangle\,,\ i \in \Sigma\}$ be a basis for the Hilbert space $\mathbb{C}^{\Sigma}$ and write $\mathcal{A} = \sum_{i,j\in\Sigma} \alpha_{ij}\,|\psi_i\rangle\,\langle\psi_j|$. Let $\{|\zeta_k\rangle\,,\ k \in \Gamma\}$ be a basis for the Hilbert space $\mathbb{C}^{\Gamma}$ and write $\mathcal{B} = \sum_{k,l\in\Gamma} \beta_{kl}\,|\zeta_k\rangle\,\langle\zeta_l|$. The operators $\mathcal{A}$ and $\mathcal{B}$ define a *spectral representation* of the respective Hilbert spaces. The *tensor product* is an operation that creates a vector space of dimension $|\Sigma||\Gamma|$ from vector spaces with associated dimensions $|\Sigma|$ and $|\Gamma|$. A basis for the tensor product system $\mathcal{A} \otimes \mathcal{B}$ associated with the Hilbert space $\mathbb{C}^{\Sigma\Gamma}$ is given by $|\psi_i\rangle \otimes |\zeta_k\rangle\,,\ i \in \Sigma$ and $k \in \Gamma$

and the matrix representation $\mathcal{A} \otimes \mathcal{B}$ is

$$\begin{aligned} \mathcal{A} \otimes \mathcal{B} &= \left(\sum_{i,j \in \Sigma} \alpha_{ij} \left| \psi_i \right\rangle \left\langle \psi_j \right| \right) \otimes \left(\sum_{k,l \in \Gamma} \beta_{kl} \left| \zeta_k \right\rangle \left\langle \zeta_l \right| \right) \\ &= \sum_{i,j,k,l \in \Sigma\Gamma} \alpha_{ij} \beta_{kl} \left| \psi_i \right\rangle \left\langle \psi_j \right| \otimes \left| \zeta_k \right\rangle \left\langle \zeta_l \right| \\ &= \sum_{i,j,k,l \in \Sigma\Gamma} \alpha_{ij} \beta_{kl} (\left| \psi_i \right\rangle \otimes \left| \zeta_k \right\rangle)(\left\langle \psi_j \right| \otimes \left\langle \zeta_l \right|), \end{aligned} \tag{1.1.10}$$

where the $(i,k),(j,l)$ element of $\mathcal{A} \otimes \mathcal{B}$ is $\alpha_{ij}\beta_{kl}$. More generally, the Hilbert space associated with the tensor product space of n basis states over $\mathbb{C}^{\Sigma}$ is

$$\mathbb{C}^{\Sigma} \otimes \cdots \otimes \mathbb{C}^{\Sigma} \cong \mathbb{C}^{\Sigma^n}. \tag{1.1.11}$$

Quantum computation relies on the theory and practice of quantum measurements. As the description of a state of an n-qubit system grows exponentially in n, it becomes increasingly difficult to access the particular information held within the system. Such attempts to access the information provide a means for describing the effects of measurements on the system. The challenge of quantum computation is to sieve through the exponential amount of information in the state and perform a measurement to extract only vital pieces of information.

Quantum measurements are described by a collection $\{M_j\}$ of *measurement operators*. Given an orthornormal basis $\{|\psi_j\rangle\}$, a measurement on the quantum state $|\psi\rangle$ in the basis representation will yield the value ψ_j. This defines the measurement operator

$$M_j = \left| \psi_j \right\rangle \left\langle \psi_j \right| \tag{1.1.12}$$

that acts on the state $|\psi\rangle = \sum_{i \in \Sigma} \alpha_i |\psi_i\rangle$. Thus, the measurement operator

M_j extracts the component of a quantum state associated with $|\psi_j\rangle$,

$$\begin{aligned} M_j |\psi\rangle &= |\psi_j\rangle \langle\psi_j|\psi\rangle \\ &= \sum_{i\in\Sigma} \alpha_i \delta_{ij} |\psi_j\rangle \\ &= \alpha_j |\psi_j\rangle . \end{aligned} \tag{1.1.13}$$

The probability, p_j, that the result ψ_j occurs post measurement can be written as

$$\begin{aligned} p_j &= \langle\psi| M_j |\psi\rangle \\ &= \langle\psi|\psi_j\rangle\langle\psi_j|\psi\rangle \\ &= \sum_{i\in\Sigma} \alpha_i^* \alpha_i \delta_{ij} \\ &= \alpha_j^* \alpha_j \\ &= |\alpha_j|^2 \end{aligned} \tag{1.1.14}$$

An important consequence of the measurement process is that it alters the superposition state of a quantum state. If the result of the measurement of $|\psi\rangle$ is ψ_j then we can describe the state post measurement as

$$\begin{aligned} |\psi_j\rangle &= \frac{1}{\alpha_j} M_j |\psi\rangle \\ &= \frac{M_j |\psi\rangle}{\sqrt{\langle\psi| M_j |\psi\rangle}}. \end{aligned} \tag{1.1.15}$$

This result implies that additional information of the state $|\psi\rangle$ cannot be collected post measurement since it is the nature of the measurement M_j to distill a classical number ψ_j thereby collapsing $|\psi\rangle$ onto one of the basis eigenstates $|\psi_j\rangle$. Furthermore, we note

$$\begin{aligned} \langle\psi| M_j^\dagger M_j |\psi\rangle &= \langle\psi_j| \alpha_j^* \alpha_j |\psi_j\rangle \\ &= p_j \end{aligned} \tag{1.1.16}$$

with

$$\begin{aligned}\sum_{j\in\Sigma} \langle\psi| M_j^\dagger M_j |\psi\rangle &= \sum_{j\in\Sigma} \langle\psi_j| \alpha_j^* \alpha_j |\psi_j\rangle \\ &= \sum_{j\in\Sigma} p_j \\ &= 1. \end{aligned} \tag{1.1.17}$$

Thus, the set of measurement operators $\{M_j\}$ adhere to the completeness equation which is expressed by the fact that the probabilities sum to unity. Consequently, we have it that the measurement operator M_j admits the condition

$$M_j^\dagger = M_j = M_j M_j, \tag{1.1.18}$$

illustrating that measurement operators associated with the basis states are idempotent and Hermitian.

Hermitian operators play an important role in the theory of quantum computation and communication and have a representation as meaningful entities, or *observables*, of classical computation. This statement is qualified since Hermitian operators admit the result that measurement of a quantum state corresponds to classical outcome.

1.2 Introduction to Coding Theory

Coding theory is a branch of mathematics which seeks efficient solutions to the many problems concerning the safe and accurate transfer of information from one destination to another. Coding theory was initiated with a 1948 paper by Claude E. Shannon [74] on the mathematics of communication and with a 1950 paper by Richard Hamming [39] on the correction of errors on magnetic storage media which introduced the concept of error-correcting codes. As

our society becomes increasingly automated the applications of coding theory become more diverse. From mobile telephone communications to interstellar communications and compact disc recordings, coding theory has ensured its prominance in modern society.

The coding *channel* is the physical medium through which we transmit information. Within the spectrum of the channel there exist disturbances that interact in an unwanted manner with the information as it passes through the channel. More formally, these disturbances are referred to as *noise*, and result in an information output that differs from the original information input.

Coding theory concerns itself with the problem of constructing efficient error-correcting procedures to minimize the noise that act on the information within the channel. A certain class of classical codes called BCH codes elicit such an effective error-correcting procedure. The BCH codes, introduced independently by Bose and Ray-Chaudhuri [10] in 1960 and by Hocquenghem [43] in 1959, rank among the most widely studied and practiced of all error-correcting codes.

To speak of noise in terms of the errors it generates on the information set is to speak on the choice of channel over which we send such information. On the selection of an acceptable channel, we turn our considerations from the communication device to devising procedures to encode and decode the information for which the affects of noise are best minimized.

Information is transmitted over a channel. The channel takes the role of a communication device in which there is a *transmitter* and a *receiver*. Associated with the transmitter is an input alphabet Σ and corresponding to the receiver we have an output alphabet Γ. We assume both Σ and Γ to be of finite cardinality with $\Sigma \subseteq \Gamma$. The channel permits the transmission of information by a sequence of characters from the finite alphabet Σ. A *word* is

a sequence of such alphabet characters. We denote Σ^n to be the set of words of *length* n where Σ^n represents the Cartesian product of Σ with itself n times.

Furthermore, constraints are placed on Shannon's general channel model. The channel requires that the length of a word does not increase or decrease during transmission. Secondly, associated with the input alphabet Σ and output alphabet Γ, we have a graph with Σ on the left and Γ on the right where the edge (σ_i, γ_i) is assigned with a probability of the channel changing σ_i to γ_i. For each $\sigma_i \in \Sigma$, we have it that the sum of the probabilities on the edged incident with it equals 1. In the theory of error-correcting codes, it is often assumed that errors occur independently for each input character [52].

Shannon considered the difficulties in transmitting information over two types of channel. Firstly, Shannon described a perfect, or *noiseless*, channel whereby information is transmitted with perfect fidelity. However, the more interesting concept of channel transmission considered by Shannon was that of a *noisy* channel. The noisy channel is regarded as the cornerstone of Information Theory by providing a beautiful description between two of the more vital concerns of coding theory, namely, the maximal transmission of information against a maximal correctness of transmission of such information. The exact result allowed the understanding that should the flow of information over the channel be less than the maximum permitted by the channel then error-correction may take place. A commonly considered noisy channel is a Binary Symmetric Channel (BSC). The alphabets Σ and Γ have cardinality two and information is transmitted in a sequence of zeros and ones. The *reliability* of the binary symmetric channel is a real number p, $(1/2 < p < 1)$, where p represents the probability that the transmitted character is the same as the received character.

A *code* C is a collection of words from a subset of Σ^n. A *block* code

maintains that the code consists of words of the same length. Suppose that a block codeword is transmitted over a BSC channel and the received word differs from the original then we have it that noise of some nature interfered with the information set resulting in an error pattern. More formally, denote $\phi_p(\sigma, \gamma)$ to be the probabilty that a transmitted codeword σ differs in d positions from the received word γ. Then $\phi_p(\sigma, \gamma)$ is given by

$$\phi_p(\sigma, \gamma) = p^{n-d}(1-p)^d. \tag{1.2.1}$$

Since each codeword σ corresponds to a word γ, we have it that the most likely word received after transmission has the most likely probability

$$\phi_p(\sigma, \gamma) = max\{\phi_p(\sigma, \gamma)\}. \tag{1.2.2}$$

Consider addition and multiplication on the characters of $\Sigma = \{0, 1\}$ in the usual way, hence,

$$\begin{aligned} 0+0=0,\ 0+1=1+0=0,\ 1+1=0 \\ 0\cdot 0=0,\ 0\cdot 1=1\cdot 0=0,\ 1\cdot 1=1, \end{aligned} \tag{1.2.3}$$

and define componentwise addition and multiplication on Σ^n in a similar fashion as defined on Σ, then Σ^n constitutes a vector space. By the *weight* of a word $\sigma \in \Sigma^n$, $wt(\sigma)$, we mean the number of non-zero components of σ. This allows us to define the *Hamming distance*, or *distance*, as

$$d(\sigma, \gamma) = min\{wt(\sigma - \gamma)\}. \tag{1.2.4}$$

Equipped with these results, we are qualified to specify a mapping to describe broad classes of codes in a concise formalism. We now introduce a special class of code called *linear codes*.

We concern ourselves with the motivation of a certain class of code known to elicit good error-correcting properties. While we earlier described codes over Σ^n, we particularize this formalism of such a group code to a version endowed with certain algebraic structures. Let $\mathbb{F}_q$, where q is a prime power, be a *field*. A block code C over Σ^n is linear over $\mathbb{F}_q^n$ when we associate the alphabet Σ with the field $\mathbb{F}_q$ if $\mathcal{C}$ is a subspace of $\mathbb{F}_q^n$.

A linear code C over $\mathbb{F}_q^n$ is a subspace with dimension k if there exists a minimal set of k vectors that generated the code, and in such case, we say C is an $[n, k, d]_q$ code. Denote $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ to be the *basis* set of vectors that generate C. Then, we can represent the code C by a *generator matrix* $\mathbf{G}$ over $\mathbb{F}_q^{k\times n}$. The rows of $\mathbf{G}$ correspond to the k linearly independent set of vectors that generate C. In particular, $C = \{\alpha\mathbf{G} \mid \alpha \in \mathbb{F}_q^k\}$.

If $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ are vectors in $\mathbb{F}_q^n$, we define the *inner product* $< \mathbf{v}|\mathbf{w} >$ of $\mathbf{v}$ and $\mathbf{w}$ as

$$< \mathbf{v}|\mathbf{w} > = \sum_{i=0}^{n} v_i \cdot w_i. \tag{1.2.5}$$

Vectors $\mathbf{v}$ and $\mathbf{w}$ are *orthogonal* if $< \mathbf{v}|\mathbf{w} > = 0$. Given a generator matrix $\mathbf{G}$ for the code C, we say that $\mathbf{w}$ is orthogonal to the code C if $< \mathbf{v}_i|\mathbf{w} > = 0$ for all $\mathbf{v}_i, i \in |k|$. Therefore, the set of vectors orthogonal to C is called the *orthogonal complement*, or *dual code*, of C and is denoted $C^\perp$. Since C is a linear subspace of dimension k, we have it that the orthogonal complement $C^\perp$ is also a subspace of $\mathbb{F}_q^n$ with dimension $n - k$ such that for $\mathbf{v} \in \mathcal{C}$ and $\mathbf{w} \in C^\perp$, $< \mathbf{v}|\mathbf{w} > = 0$. Let $\mathbf{H}^T \in \mathbb{F}_q^{(n-k)\times n}$ be the generator matrix for the code $C^\perp$. Furthermore, a matrix $\mathbf{H} \in \mathbb{F}_q^{n\times(n-k)}$ is called a *parity-check matrix* for a linear code C if the columns of $\mathbf{H}$ generate the dual code $C^\perp$, and it is this matrix formalism that admits a useful approach to the task of error-correction.

Let C be a linear $[n, k, d]_q$ code and $\mathbf{u}$ is some word in $\mathbb{F}_q^n$, we define the *coset* of C determined by $\mathbf{u}$ to be the set of words $\mathbf{v} + \mathbf{u}$ for $\mathbf{v} \in C$. In particular, the coset $C + \mathbf{u}$ is

$$C + \mathbf{u} = \{\mathbf{v} + \mathbf{u} \mid \mathbf{v} \in C\}. \tag{1.2.6}$$

Suppose that the codeword $\mathbf{v} \in C$ is transmitted over a BSC and the word $\mathbf{u}$ is received. Suppose further that the received word differs from that which was transmitted resulting in an error pattern $e = \mathbf{v} - \mathbf{u}$. We seek an efficient decoding scheme by choosing a word e of least weight in the coset $C + \mathbf{u}$. The use of the parity-check matrix $\mathbf{H}$ associated with the code C enables an efficient procedure to determine a word e of least weight in the coset $C + \mathbf{u}$. To see this, we first note that for any received word $\mathbf{u} \in \mathbb{F}_q^n$, the *syndrome* of $\mathbf{u}$ is a word $\mathbf{uH} \in \mathbb{F}_q^{(n-k)}$. Should the syndrome $\mathbf{uH} = 0$ then the received word is a codeword in C. Alternatively, if $\mathbf{uH} \neq 0$ then we can correctly identify an error patterm e associated with the coset $C + \mathbf{u}$ that relates perfectly to $\mathbf{uH}$. We conclude that the codeword transmitted was most likely $\mathbf{v} = \mathbf{u} - e$.

Chapter 2

Quantum Error Correction

2.1 The Channel

We now consider the transmission of quantum information with respect to a quantum noisy channel where a full continuum of noise is maintained. While classical information can be transmitted and protected from the effects of noise by replication, quantum information cannot be copied with perfect fidelity. Introduced by Bennett *et al.* [5], *quantum teleportation* is an experimental demonstration of the means by which quantum communication is made possible and purports a fundamental distinction between quantum and classical information theory. Such distinction is maintained by the Bell-EPR correlations whereby an essential nonlocality principle, described by quantum entanglement, is revealed. This result was demonstrated experimentally by Aspect *et al.* in 1982 [1]. Quantum teleportation takes advantage of the non-local behaviour of quantum mechanics by treating quantum entanglement as an information resource. The first complete transmission of quantum information was performed by Nielsen *et al.* [66] in 1998.

The Church-Turing Principle [24] maintains that it is impossible to transmit quantum information by implementing a classical computation. Bennett

et al. [5] introduced quantum teleportation to overcome this limitation by developing a quantum algorithm that describes a complete communication transmission of quantum information.

Given an arbitrary finite alphabet Σ of cardinality $\mathbf{d}$, we process quantum information by specifying a state description of a finite dimension quantum space. In particular, the state description of the Hilbert space $\mathbb{C}^{\mathbf{d}}$. While the state of an Σ-dimensional Hilbert space can be more generally expressed as a linear combination of basis states $|\psi_i\rangle$, we write each orthonormal basis state of the $\mathbf{d}$-dimensional Hilbert space $\mathbb{C}^{\mathbf{d}}$ to correspond with an element of $\mathbb{Z}_{\mathbf{d}}$. In this context the basis $\{|0\rangle, |1\rangle, \ldots, |\mathbf{d}-1\rangle\}$ is referred to as the *computational basis*. Therefore, a state $|\psi\rangle$ of $\mathbb{C}^{\mathbf{d}}$ is given by

$$|\psi\rangle = \sum_{i=0}^{\mathbf{d}-1} \alpha_i |i\rangle, \tag{2.1.1}$$

where $\alpha_i \in \mathbb{C}$ and $\sum_{i=0}^{\mathbf{d}-1} |\alpha_i|^2 = 1$. A *qu***d***it* describes a state in the Hilbert space $\mathbb{C}^{\mathbf{d}}$. The state space of an n-qu**d**it state is the tensor product of the basis states of the single system $\mathbb{C}^{\mathbf{d}}$, written $\mathcal{H}^n = (\mathbb{C}^{\mathbf{d}})^{\otimes n}$, with corresponding orthonormal basis states given by

$$|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle = |i_1 i_2 \ldots i_n\rangle, \tag{2.1.2}$$

where $i_j \in \mathbb{Z}_{\mathbf{d}}$. The general state of a qu**d**it in the Hilbert space $\mathcal{H}^n$ is then written

$$|\psi\rangle = \sum_{(i_1 i_2 \ldots i_n) \in \mathbb{Z}_{\mathbf{d}}^n} \alpha_{(i_1 i_2 \ldots i_n)} |i_1 i_2 \ldots i_n\rangle, \tag{2.1.3}$$

where $\alpha_{(i_1 i_2 \ldots i_n)} \in \mathbb{C}$ and $\sum |\alpha_{(i_1 i_2 \ldots i_n)}|^2 = 1$.

Quantum teleportation describes how two parties, $\mathcal{A}$ and $\mathcal{B}$, process and communicate quantum information in a manner secure from the effects of error.

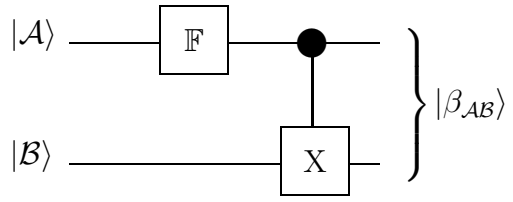


Figure 2.1: Generalised Bell State.

Suppose $\mathcal{A}$ wishes to communicate the state $|\psi\rangle$ then the goal of teleportation is to transmit that particular quantum information state to $\mathcal{B}$. Further suppose that $\mathcal{A}$ prepares the qu**d**it $|\mathcal{A}\rangle$ where $|\mathcal{A}\rangle \in \{|0\rangle, |1\rangle, \ldots, |\mathbf{d}-1\rangle\}$. Similarly, $\mathcal{B}$ prepares the qu**d**it $|\mathcal{B}\rangle$ where $|\mathcal{B}\rangle \in \{|0\rangle, |1\rangle, \ldots, |\mathbf{d}-1\rangle\}$. In order to achieve teleportation, $\mathcal{A}$ must interact the information state $|\psi\rangle$ with a two qu**d**it entangled state, $|\beta_{\mathcal{AB}}\rangle$. The entangled state $|\beta_{\mathcal{AB}}\rangle$ is called a *generalised Bell state* whereby $\mathcal{A}$ and $\mathcal{B}$ each possess one qu**d**it of this two qu**d**it state. To construct a generalised Bell state $|\beta_{\mathcal{AB}}\rangle$, we first apply the Fourier transform $\mathbb{F}\otimes I$ to the qu**d**it $|A\rangle$. This acts on basis states $|j\rangle|k\rangle$ as follows $(\mathbb{F}\otimes I)|j\rangle|k\rangle = \frac{1}{\sqrt{\mathbf{d}}}\sum_{i=0}^{\mathbf{d}} \omega^{ij}|i\rangle|k\rangle$ where ω is a primitive $\mathbf{d}^{\text{th}}$ root of unity in $\mathbb{C}$ such that $\omega^{\mathbf{d}} = 1$ and $\omega^t \neq 1$ for all $0 < t < \mathbf{d}$. Secondly, we follow the Fourier transform by the Controlled-NOT operation given by $|k\rangle|l\rangle \mapsto |k\rangle|l + k \pmod{\mathbf{d}}\rangle$ for all basis states $|k\rangle|l\rangle$ which maps the two qu**d**it state accordingly. Consequently, any pair of qu**d**its $|\mathcal{A}\rangle|\mathcal{B}\rangle$ from the $\mathbf{d}^2$ computational basis states of $\mathbb{C}^{\mathbf{d}} \otimes \mathbb{C}^{\mathbf{d}}$ generate a generalised Bell state. In particular, applying the Fourier transform

to the first half of the pair of qu**d**it states $|\mathcal{A}\rangle\,|\mathcal{B}\rangle$, we obtain,

$$\begin{aligned}
&\left(\frac{1}{\sqrt{\mathbf{d}}}\sum_{i=0}^{\mathbf{d}-1}\sum_{j=0}^{\mathbf{d}}\sum_{x=0}^{\mathbf{d}-1}\omega^{ix}\,|x\rangle\,|j\rangle\,\langle i|\,\langle j|\right)|\mathcal{A}\rangle\,|\mathcal{B}\rangle \\
&= \frac{1}{\sqrt{\mathbf{d}}}\sum_{i=0}^{\mathbf{d}-1}\sum_{j=0}^{\mathbf{d}-1}\sum_{x=0}^{\mathbf{d}-1}\omega^{ix}\,|x\rangle\,|j\rangle\,\langle i|\mathcal{A}\rangle\langle j|\mathcal{B}\rangle \\
&= \frac{1}{\sqrt{\mathbf{d}}}\sum_{x=0}^{\mathbf{d}-1}\omega^{\mathcal{A}x}\,|x\rangle\,|\mathcal{B}\rangle\,. \qquad (2.1.4)
\end{aligned}$$

The action of the Controlled-NOT operator on resulting state (2.1.4) completes the generalised Bell state construction

$$\begin{aligned}
&\left(\sum_{k=0}^{\mathbf{d}-1}\sum_{l=0}^{\mathbf{d}-1}|k\rangle\,|l+k\rangle\,\langle k|\,\langle l|\right)\frac{1}{\sqrt{\mathbf{d}}}\sum_{x=0}^{\mathbf{d}-1}\omega^{\mathcal{A}x}\,|x\rangle\,|\mathcal{B}\rangle \\
&= \frac{1}{\sqrt{\mathbf{d}}}\sum_{l=0}^{\mathbf{d}-1}\sum_{k=0}^{\mathbf{d}-1}\sum_{x=0}^{\mathbf{d}-1}\omega^{\mathcal{A}x}\,|k\rangle\,|l+k\rangle\,\langle k|x\rangle\langle l|\mathcal{B}\rangle \\
&= \frac{1}{\sqrt{\mathbf{d}}}\sum_{x=0}^{\mathbf{d}-1}\omega^{\mathcal{A}x}\,|x\rangle\,|\mathcal{B}+x\rangle \\
&= |\beta_{\mathcal{A}\mathcal{B}}\rangle\,. \qquad (2.1.5)
\end{aligned}$$

Since the Bell pair is an entangled state [60], any operator acting on the first qu**d**it held by $\mathcal{A}$ influences the state of the second qu**d**it held by $\mathcal{B}$. This condition permits the teleportation of the quantum information state $|\psi\rangle$ between parties $\mathcal{A}$ and $\mathcal{B}$ when $\mathcal{A}$ interacts $|\psi\rangle$ with the first half of the generalised Bell pair (2.1.5). To negate the effects of the Bell state transformations on $|\psi\rangle$, thereby allowing the teleportation of $|\psi\rangle$, $\mathcal{A}$ transforms $|\psi\rangle$ by applying the inverse of the generalised Controlled-NOT operator for qu**d**it states which is then followed by an application of the inverse Fourier transform. Now, the Fourier transform is unitary so its inverse is its adjoint, and the inverse of the generalised Controlled-NOT operation has its action defined as $|k\rangle\,|l\rangle \mapsto |k\rangle\,|l-k \pmod{\mathbf{d}}\rangle$.

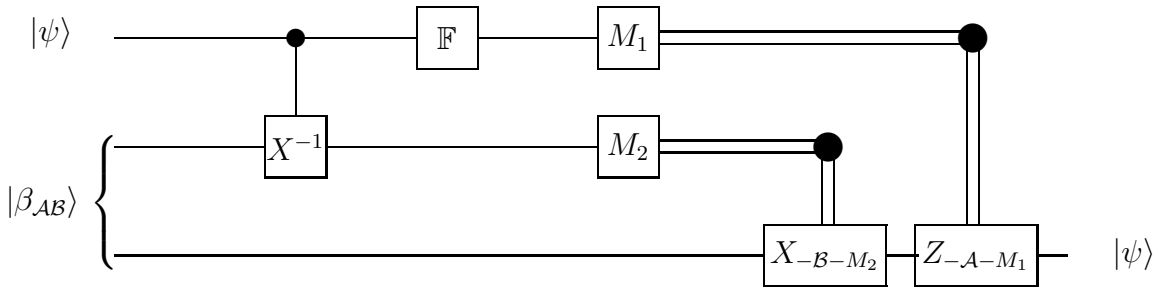


Figure 2.2: Quantum channel for teleporting a qu**d**it.

We write the state of the quantum system held by $\mathcal{A}$ and $\mathcal{B}$, as

$$|\psi\rangle\,|\beta_{\mathcal{AB}}\rangle \;=\; \frac{1}{\sqrt{\mathbf{d}}}\sum_{a=0}^{\mathbf{d}-1}\alpha_a\,|a\rangle\left(\sum_{x=0}^{\mathbf{d}-1}\omega^{\mathcal{A}x}\,|x\rangle\,|\mathcal{B}+x\rangle\right). \tag{2.1.6}$$

$\mathcal{A}$ initiates teleportation of the quantum information state $|\psi\rangle$ by applying the inverse generalised Controlled-NOT operation between $|\psi\rangle$ and the qu**d**it of the generalised Bell state held by $\mathcal{A}$, thereby obtaining,

$$\begin{aligned}
&\left(\sum_{k=0}^{\mathbf{d}-1}\sum_{l=0}^{\mathbf{d}-1}\sum_{m=0}^{\mathbf{d}-1}|k\rangle\,|l-k\rangle\,|m\rangle\,\langle k|\,\langle l|\,\langle m|\right)\frac{1}{\sqrt{\mathbf{d}}}\sum_{a=0}^{\mathbf{d}-1}\alpha_a\,|a\rangle\sum_{x=0}^{\mathbf{d}-1}\omega^{\mathcal{A}x}\,|x\rangle\,|\mathcal{B}+x\rangle\\
&=\;\frac{1}{\sqrt{\mathbf{d}}}\sum_{k=0}^{\mathbf{d}-1}\sum_{l=0}^{\mathbf{d}-1}\sum_{m=0}^{\mathbf{d}-1}\sum_{a=0}^{\mathbf{d}-1}\sum_{x=0}^{\mathbf{d}-1}\alpha_a\omega^{\mathcal{A}x}\,|k\rangle\,|l-k\rangle\,|m\rangle\,\langle k|a\rangle\langle l|x\rangle\langle m|\mathcal{B}+x\rangle\\
&=\;\frac{1}{\sqrt{\mathbf{d}}}\sum_{a=0}^{\mathbf{d}-1}\sum_{x=0}^{\mathbf{d}-1}\alpha_a\omega^{\mathcal{A}x}\,|a\rangle\,|x-a\rangle\,|\mathcal{B}+x\rangle\,.
\end{aligned}\tag{2.1.7}$$

Following this result, $\mathcal{A}$ applies the discrete Fourier transformation on the first qu**d**it of the state (2.1.7). The outcome of this operation is to place the state (2.1.7) into the state given by

$$\left(\frac{1}{\sqrt{\mathbf{d}}}\sum_{i=0}^{\mathbf{d}-1}\sum_{y=0}^{\mathbf{d}-1}\sum_{j=0}^{\mathbf{d}-1}\sum_{n=0}^{\mathbf{d}-1}\omega^{iy}\,|y\rangle\,|j\rangle\,|n\rangle\,\langle i|\,\langle j|\,\langle n|\right)\frac{1}{\sqrt{\mathbf{d}}}\sum_{a=0}^{\mathbf{d}-1}\sum_{x=0}^{\mathbf{d}-1}\alpha_a\omega^{\mathcal{A}x}\,|a\rangle\,|x-a\rangle\,|\mathcal{B}+x\rangle$$

$$
\begin{aligned}
&= \frac{1}{\mathbf{d}} \sum_{i=0}^{\mathbf{d}-1} \sum_{y=0}^{\mathbf{d}-1} \sum_{j=0}^{\mathbf{d}-1} \sum_{n=0}^{\mathbf{d}-1} \sum_{a=0}^{\mathbf{d}-1} \sum_{x=0}^{\mathbf{d}-1} \alpha_a \omega^{iy} \omega^{\mathcal{A}x} \left|y\right\rangle \left|j\right\rangle \left|n\right\rangle \left\langle i|a\right\rangle \left\langle j|x-a\right\rangle \left\langle n|\mathcal{B}+x\right\rangle \\
&= \frac{1}{\mathbf{d}} \sum_{y=0}^{\mathbf{d}-1} \sum_{a=0}^{\mathbf{d}-1} \sum_{x=0}^{\mathbf{d}-1} \alpha_a \omega^{ay} \omega^{\mathcal{A}x} \left|y\right\rangle \left|x-a\right\rangle \left|\mathcal{B}+x\right\rangle \\
&= \frac{1}{\mathbf{d}} \sum_{y=0}^{\mathbf{d}-1} \sum_{a=0}^{\mathbf{d}-1} \sum_{x=0}^{\mathbf{d}-1} \sum_{z=0}^{\mathbf{d}-1} \alpha_a \omega^{ay} \omega^{\mathcal{A}x} \left|y\right\rangle \left|z\right\rangle \left\langle z|x-a\right\rangle \left|\mathcal{B}+x\right\rangle \\
&= \frac{1}{\mathbf{d}} \sum_{y=0}^{\mathbf{d}-1} \sum_{a=0}^{\mathbf{d}-1} \sum_{z=0}^{\mathbf{d}-1} \alpha_a \omega^{ay} \omega^{\mathcal{A}(z+a)} \left|y\right\rangle \left|z\right\rangle \left|\mathcal{B}+z+a\right\rangle \\
&= \frac{1}{\mathbf{d}} \sum_{y=0}^{\mathbf{d}-1} \sum_{z=0}^{\mathbf{d}-1} \omega^{\mathcal{A}z} \left|y\right\rangle \left|z\right\rangle \left(\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{a(y+\mathcal{A})} \left|\mathcal{B}+z+a\right\rangle \right). \qquad (2.1.8)
\end{aligned}
$$

The qu**d**it of the generalised Bell state held by $\mathcal{B}$ is transformed into the state $\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{a(y+\mathcal{A})} \left|\mathcal{B}+z+a\right\rangle$. Thus $\mathcal{A}$ has teleported a quantum information state $\left|\psi'\right\rangle$ to $\mathcal{B}$, however, it has been subjected to *error* over the channel and therefore $\mathcal{B}$ receives $\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{a(y+\mathcal{A})} \left|\mathcal{B}+z+a\right\rangle$ instead of $\sum_{a=0}^{\mathbf{d}-1} \alpha_a \left|a\right\rangle$. A measurement projection onto the computational basis state of $\mathbb{C}^{\mathbf{d}} \otimes \mathbb{C}^{\mathbf{d}}$ is performed by $\mathcal{A}$ on the first and second qu**d**it of the state of the quantum system (2.1.8) which yields two classical numbers. Simultaneously, the third qu**d**it of the state of the system (2.1.8) teleported to $\mathcal{B}$ collapses to a post-measurement state that is dependent upon the measurement outcome obtained by $\mathcal{A}$. Let M_1, M_2 be two classical numbers corresponding to the resulting states $\left|M_1\right\rangle \left|M_2\right\rangle$. Then the state of the qu**d**it held by $\mathcal{B}$ is given by $\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \left|\mathcal{B}+M_2+a\right\rangle$. The set M_1, M_2 is transferred by classical means to $\mathcal{B}$, where upon delivery $\mathcal{B}$ learns which of the generalised Pauli operators are required to correct the effect of the error. In particular, $\mathcal{B}$ applies the operators

$$
X_{-\mathcal{B}-M_2} = \sum_{x=0}^{\mathbf{d}-1} \left|x-\mathcal{B}-M_2\right\rangle \left\langle x\right|
$$

and

$$Z_{-\mathcal{A}-M_1} = \sum_{z=0}^{\mathbf{d}-1} \omega^{(-\mathcal{A}-M_1)z} \left|z\right\rangle \left\langle z\right|$$

in order to return the post-measurement state $\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \left|\mathcal{B}+M_2+a\right\rangle$ to the initial quantum information state $\left|\psi\right\rangle$. Hence, applying $X_{-\mathcal{B}-M_2}$ to the post-measurement state, $\mathcal{B}$ obtains

$$\begin{aligned}
&X_{-\mathcal{B}-M_2} \left(\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \left|\mathcal{B}+M_2+a\right\rangle \right) \\
&= \sum_{x=0}^{\mathbf{d}-1} \left|x-\mathcal{B}-M_2\right\rangle \left\langle x\right| \left(\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \left|\mathcal{B}+M_2+a\right\rangle \right) \\
&= \sum_{x=0}^{\mathbf{d}-1} \sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \left|x-\mathcal{B}-M_2\right\rangle \left\langle x|\mathcal{B}+M_2+a\right\rangle \\
&= \sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \left|a\right\rangle . \qquad (2.1.9)
\end{aligned}$$

The operator $Z_{-\mathcal{A}-M_1}$ is then applied by $\mathcal{B}$ on result (2.1.9) returning the post-measurement state to the quantum information state initially held by $\mathcal{A}$,

$$\begin{aligned}
&Z_{-\mathcal{A}-M_1} \left(\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \left|a\right\rangle \right) \\
&= \sum_{z=0}^{\mathbf{d}-1} \omega^{(-\mathcal{A}-M_1)z} \left|z\right\rangle \left\langle z\right| \left(\sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(M_1+\mathcal{A})a} \left|a\right\rangle \right) \\
&= \sum_{z=0}^{\mathbf{d}-1} \sum_{a=0}^{\mathbf{d}-1} \alpha_a \omega^{(\mathcal{A}+M_1)a} \omega^{(-\mathcal{A}-M_1)z} \left|z\right\rangle \left\langle z|a\right\rangle \\
&= \sum_{a=0}^{\mathbf{d}-1} \alpha_a \left|a\right\rangle . \qquad (2.1.10)
\end{aligned}$$

Thus $\mathcal{B}$ obtains the quantum information which $\mathcal{A}$ wished to transmit. The quantum information can only be obtained if it vanishes from $\mathcal{A}$ thereby upholding the no-cloning theorem. Furthermore, quantum teleportion provides a complete description of the transmitted state $\left|\psi\right\rangle$.

2.2 An Error Model

The challenge of quantum information processing is to elicit a reliable form of communication and to maintain such a form in the presence of quantum noise. Noise is a characteristic of the *environment* associated with an information state and is a property of an *open quantum system* that subjects an information state to unwanted interactions with the elements of the environment during teleportation. It is inevitable that the communication of an information state will cause interactions with the environment. However, prolonged contact between the information state and environment is soon to suffer in entanglement that degrades the information state. This process is called *decoherence*. Any strategy to stabilize quantum computations from the effects of noise will ultimately be required to deal with both the problems of decoherence and unitary imperfections of channel communication. Thus, to understand the fundamentals of noise propagation is to understand the formalism of a model that explains it.

Given a qu**d**it information state $|\psi\rangle = \sum_{i=0}^{\mathbf{d}-1} \alpha_i |i\rangle$ of the Hilbert space $\mathbb{C}^{\mathbf{d}}$, let us consider an adjoined environment space $|E\rangle$ endowed with an orthonormal basis of dimension $\mathbf{d}^2$. We suppose that both the state space of the qu**d**it and the corresponding environment space are initially independent systems. The joint state of the systems $|\psi\rangle$ and $|E\rangle$ is then $|\psi\rangle \otimes |E\rangle$ and its dynamics may be characterised when we further suppose that the joint system evolves according to some unitary operation. Given a unitary operation U, we write interaction of each basis qu**d**it with the environment under U as

$$U(|i\rangle \otimes |E\rangle) \;=\; \sum_{l=0}^{\mathbf{d}-1} \gamma_{-i+l,-i}(|i+l\rangle \otimes |e_{-i+l,-i}\rangle)$$

$$= \sum_{l=0}^{\mathbf{d}-1} |i+l\rangle \otimes \gamma_{-i+l,-i} |e_{-i+l,-i}\rangle , \quad (2.2.1)$$

for $i \in \{0, \ldots, \mathbf{d}-1\}$. By linearity of U, the dynamics of the joint system $|\psi\rangle \otimes |E\rangle$ is then

$$U(|\psi\rangle \otimes |E\rangle) = U\left(\left(\sum_{i=0}^{\mathbf{d}-1} \alpha_i |i\rangle\right) \otimes |E\rangle\right) = U\left(\sum_{i=0}^{\mathbf{d}-1} \alpha_i (|i\rangle \otimes |E\rangle)\right)$$
$$= \sum_{i=0}^{\mathbf{d}-1} \alpha_i U(|i\rangle \otimes |E\rangle) = \sum_{i=0}^{\mathbf{d}-1} \sum_{l=0}^{\mathbf{d}-1} \alpha_i |i+l\rangle \otimes \gamma_{-i+l,-i} |e_{-i+l,-i}\rangle . \quad (2.2.2)$$

Since $\frac{1}{\mathbf{d}} \sum_{z=0}^{\mathbf{d}-1} \omega^{zk} = 1$ if $z = 0$ and vanishes otherwise then equation (2.2.2) may be written as

$$\frac{1}{\mathbf{d}} \sum_{i=0}^{\mathbf{d}-1} \sum_{l=0}^{\mathbf{d}-1} \left(\alpha_i |i+l\rangle \otimes \left(\sum_{z=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} \omega^{zk} \gamma_{-i+l+z,-i+z} |e_{-i+l+z,-i+z}\rangle\right)\right)$$
$$= \frac{1}{\mathbf{d}} \sum_{i=0}^{\mathbf{d}-1} \sum_{l=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} \left(\alpha_i |i+l\rangle \otimes \left(\sum_{z=0}^{\mathbf{d}-1} \omega^{zk} \gamma_{-i+l+z,-i+z} |e_{-i+l+z,-i+z}\rangle\right)\right)$$
$$= \frac{1}{\mathbf{d}} \sum_{l=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} \left(\sum_{i=0}^{\mathbf{d}-1} \left(\alpha_i |i+l\rangle \otimes \left(\sum_{z=0}^{\mathbf{d}-1} \omega^{zk} \gamma_{-i+l+z,-i+z} |e_{-i+l+z,-i+z}\rangle\right)\right)\right)$$
$$= \frac{1}{\mathbf{d}} \sum_{l=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} \left(\sum_{i=0}^{\mathbf{d}-1} \left(\omega^{ik} \alpha_i |i+l\rangle \otimes \left(\sum_{z=0}^{\mathbf{d}-1} \omega^{-ik} \omega^{zk} \gamma_{-i+l+z,-i+z} |e_{-i+l+z,-i+z}\rangle\right)\right)\right)$$
$$= \frac{1}{\mathbf{d}} \sum_{l=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} \left(\sum_{i=0}^{\mathbf{d}-1} \left(\omega^{ik} \alpha_i |i+l\rangle \otimes \left(\sum_{z'=0}^{\mathbf{d}-1} \omega^{z'k} \gamma_{z'+l,z'} |e_{z'+l,z'}\rangle\right)\right)\right)$$
$$= \frac{1}{\mathbf{d}} \sum_{l=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} \left(\left(\sum_{i=0}^{\mathbf{d}-1} \omega^{ik} \alpha_i |i+l\rangle\right) \otimes \left(\sum_{z'=0}^{\mathbf{d}-1} \omega^{z'k} \gamma_{z'+l,z'} |e_{z'+l,z'}\rangle\right)\right) . \quad (2.2.3)$$

An outer product representation describes the set of errors that act on the joint quantum state under U. The operator $X_1 = \sum_{i=0}^{\mathbf{d}-1} |i+1\rangle \langle i|$ maps $\alpha_i |i\rangle$ to $\alpha_i |i+1\rangle$ for $i \in \{|0\rangle, \ldots, |\mathbf{d}-1\rangle\}$, and thus maps $\sum_{i=0}^{\mathbf{d}-1} \alpha_i |i\rangle$ to $\sum_{i=0}^{\mathbf{d}-1} \alpha_i |i+1\rangle$. Similarly, $Z_1 = \sum_{i=0}^{\mathbf{d}-1} \omega^i |i\rangle \langle i|$ maps $\alpha_i |i\rangle$ to $\omega^i \alpha_i |i\rangle$ and correspondingly maps $\sum_{i=0}^{\mathbf{d}-1} \alpha_i |i\rangle$ to $\sum_{i=0}^{\mathbf{d}-1} \omega^i \alpha_i |i\rangle$. Both X_1 and Z_1 are called

the *Weyl Pair* [96]. Consequently, the action of U on $|\psi\rangle \otimes |E\rangle$ is described by the set of operators $X_l Z_k = \sum_{i=0}^{\mathbf{d}-1} \omega^{ik} |i+l\rangle \langle i|$, $(l,k) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}$,

$$\sum_{l=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} \left(\left(\sum_{i=0}^{\mathbf{d}-1} \omega^{ik} \alpha_i |i+l\rangle \right) \otimes \frac{1}{\mathbf{d}} \left(\sum_{z'=0}^{\mathbf{d}-1} \omega^{z'k} \gamma_{z'+l,z'} |e_{z'+l,z'}\rangle \right) \right)$$

$$= \sum_{l=0}^{\mathbf{d}-1} \sum_{k=0}^{\mathbf{d}-1} X_l Z_k |\psi\rangle \otimes \gamma_{lk} |e_{lk}\rangle . \tag{2.2.4}$$

Thus, to correctly specify an error model that describes the action of a unitary operator U on the joint space $|\psi\rangle \otimes |E\rangle$, it is necessary that the environment $|E\rangle$, associated with an information state in $\mathbb{C}^{\mathbf{d}}$, be a Hilbert space of dimension $\mathbf{d}^2$. Following the action of U on the joint system, a measurement on the environment is performed with respect to the basis $|e_{mn}\rangle$, $(m,n) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}$ to diagnose the introduced error in result (2.2.4). Therefore, equation (2.2.4) provides the conceptual foundation of quantum error correction. Measurements taken in the environment basis initiate the correction step $(X_m Z_n)^{-1} = Z_{(-n \bmod \mathbf{d})} X_{(-m \bmod \mathbf{d})}$.

The *depolarization channel* is a well investigated error model, see [60], and references therein, and it is the error model with which we concern ourselves. We adapt its qubit form to cater for qu**d**its so that the effects of noise as those modelled by equation (2.2.4) can be explained. In particular, we have it that there is a $1 - \frac{(\mathbf{d}^2-1)p}{\mathbf{d}^2}$ probability that $|\psi\rangle$ will not suffer the effects of noise and a $\frac{p}{\mathbf{d}^2}$ probability that any given error will occur. The depolarizing channel for qu**d**its is therefore represented as

$$U(|\psi\rangle \otimes |E\rangle) \mapsto$$
$$\sqrt{1 - \frac{(\mathbf{d}^2-1)p}{\mathbf{d}^2}} |\psi\rangle \otimes |e_{X_0 Z_0}\rangle + \sqrt{\frac{p}{\mathbf{d}^2}} \left(\sum_{(k,l)\setminus(0,0) \in Z_{\mathbf{d}} \times Z_{\mathbf{d}}} X_l Z_k |\psi\rangle \otimes |e_{X_l Z_k}\rangle \right) . \tag{2.2.5}$$

In general, we consider the case where $1 - \frac{(\mathbf{d}^2-1)p}{\mathbf{d}^2} > \frac{p}{\mathbf{d}^2}$, and as such the probability of an error operator decreases exponentially with weight [2]. Consequently, good n-qu**d**it error-correcting codes are those for which all errors of weight $\leq \frac{p}{\mathbf{d}^2}n$ can be corrected. It is for this reason that the rationale for error analysis is guided by the necessity to detect and error correct error operators up to a given weight [2].

2.3 Definition and basic properties

The process of quantum error detection and correction raises difficulties not evident in the classical analogue. Firstly, the coding theorist must contend with an extra class of error, phase errors, and any subsequent linear combinations with qu**d**it flip errors that are introduced by the environment during information transmission. Secondly, information transmission causes communication signals to become attenuated which is primarily due to the physical capabilities of the hardware used. Consequently, the likelihood of errors occurring during transmission may increase. In particular, the 10^{-4} threshold level is widely assumed to be the fault-tolerant threshold for both environmentally induced and systematic errors. More rigorous bounds recently calculated suggest that this threshold could be closer to 10^{-5} [85], and references therein. For what follows, it is worth noting that a robust CNOT gate could operate at close to the 10^{-7} level, a level well within more rigorous threshold bounds. Furthermore, for systems whose Hamiltonian is unclear, the CNOT operating at the 10^{-7} level represents an important result as it suggests that operating the CNOT gate in this way can ensure that the error rate remains below the fault-tolerant error threshold [69, 85]. Finally, the no-cloning theorem maintains that it is impossible to generate copies of an unknown quantum state.

Classical error correction enables reliable communication by the process of data replication. However, measurements that allow classical information to be obtained cause the collapse of a quantum state thereby destroying the quantum information content. Quantum error detection and correction schemes do exist despite these difficulties.

A quantum code $\mathcal{C}$ consists of an encoding function $\mathbf{E}$ from the Hilbert space $\mathcal{H}_k = (\mathbb{C}^{\mathbf{d}})^{\otimes k} \equiv \mathbb{C}^{\mathbf{d}^k}$ to the Hilbert space $\mathcal{H}_n = (\mathbb{C}^{\mathbf{d}})^{\otimes n} \equiv \mathbb{C}^{\mathbf{d}^n}$, where k and n are integers and $k < n$. We define the codewords of a quantum code to be those states contained in the image of $\mathbf{E}$, $\mathrm{Im}(\mathbf{E})$. The *length* of the code is given by n, while k denotes the number of encoded message qu**d**its, or *logical* qu**d**its, of the code. The extra $n-k$ qu**d**its introduce additional information that allows the logical qu**d**its to be stored in a redundant manner where such redundancy can be used to detect transmission errors. A code $\mathcal{C}$ is a quantum $[\![n, k]\!]_{\mathbf{d}}$ code over $\mathbb{C}^{\mathbf{d}}$ if it is a subspace of dimension $\mathbf{d}^k$ in the Hilbert space $\mathcal{H}_n$.

An error operator E acting on a qu**d**it state can be written as a linear combination of $\mathcal{E} = \{X_l Z_k : (l,k) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}\}$. The *generalised Pauli group*, $\mathcal{G}_1$ is a group has order $\mathbf{d}^4$ generated by $\mathcal{E}$ and τI with center $\zeta(\mathcal{G}_1) = \langle \tau I \rangle$. For an n-qu**d**it system, any operator E of the group $\mathcal{G}_1^{\otimes n}$ can be written as

$$E = \tau^{\alpha}(X_{l_1} Z_{k_1}) \otimes (X_{l_2} Z_{k_2}) \otimes \cdots \otimes (X_{l_n} Z_{k_n}), \tag{2.3.1}$$

where $\alpha \in \{0, \ldots, \mathbf{d}-1\}$ and $((l_1,k_1),(l_2,k_2),\ldots,(l_n,k_n)) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n$. The *weight* of the error E, $wt(E)$, in $\mathcal{E}^{\otimes n}$ is the number of pairs (l_i, k_i) for which l_i and k_i are not both zero.

Let $|\psi\rangle \in \mathcal{C}$ be a codeword that is transmitted across a noisy quantum channel. Thus $|\psi\rangle$ has the expansion $\sum_{k=0}^{\mathbf{d}-1} \sum_{l=0}^{\mathbf{d}-1} X_l Z_k |\psi\rangle \otimes |e_{X_l Z_k}\rangle$. This result provides a conceptual starting point for which a quantum error-correcting

code can be explained. Let $\{\psi_1, \ldots, \psi_{n-k}\}$ be an orthogonal basis for $\mathcal{C}$ and write $\{\mathcal{E}\} = \{E_1, \ldots, E_{\mathbf{d}^2}\}$. To detect an error E in $\mathcal{E}$ in the computational basis of $\mathcal{C}$, we require that the state $E|\psi_i\rangle$ be orthogonal from all $|\psi_j\rangle$, hence, $\langle\psi_j|E|\psi_i\rangle = 0$. To correct an error, we require that the action of error E_l on $|\psi_i\rangle$ be distinct from the action of E_k, for $k \neq l$, on $|\psi_j\rangle$. Hence, a necessary condition for quantum error-correcting is given by $\langle\psi_j|E_k^\dagger E_l|\psi_i\rangle = 0$ for $i \neq j$ and for all $E_k, E_l \in \mathcal{E}$. If this condition were not satisfied for $i \neq j$ and $k \neq l$, we have $|\psi_z\rangle = E_l|\psi_i\rangle = E_k|\psi_j\rangle$. Suppose $|\psi_z\rangle$ is a received word then it would not be possible to determine the original codeword transmitted since the distinguishability of the orthogonal codewords is destroyed. To see that this necessary condition is also sufficient, let us assume that $E_k|\psi_j\rangle$ is distinct from $E_l|\psi_i\rangle$, for $i \neq j$ and $k \neq l$, and construct a decoding function $D(|\psi\rangle)$. Then there is a unique $|\psi_i\rangle \in \mathcal{C}$ and $E_{i(z)} \in \mathcal{E}$ for which $E_{i(z)=l}|\psi_i\rangle = |\psi_z\rangle$ is true. Thus we can decode $|\psi_z\rangle$ by defining $|\psi_i\rangle = D(|\psi_z\rangle)$.

We now give a formal treatment of the quantum error-correcting code conditions. The quantum error correcting conditions are a set of equations which can be checked to determine whether a quantum code protects against a particular type of noise from the environment. The set of error-correcting conditions for quantum error-correcting codes were introduced by Knill and Laflamme [52]. These conditions for qubit error-correcting codes were discussed further in Nielsen and Chuang [60]. We adapt the proof of Nielsen and Chuang [60] and consider the formulation of these conditions with respect to quantum basis states instead of density matrices. We also cater for quantum error-correcting codes over larger dimensions.

Theorem 1. [52] Let $\mathcal{C}$ be a subspace of the Hilbert space $\mathcal{H}$. Then $\mathcal{C}$ is a quantum error-correcting code for the error operators $\mathcal{E} = \{E_1, \ldots, E_{\mathbf{d}^2}\}$ if

and only if there exists $\alpha_{l,k} \in \mathbb{C}$ such that, for all $|\psi_i\rangle, |\psi_j\rangle \in \mathbb{C}$ and E_k, E_l in $\mathcal{E}$,

$$\langle\psi_j| E_k^\dagger E_l |\psi_i\rangle \quad = \quad \alpha_{l,k}\delta_{ij}. \tag{2.3.2}$$

Proof: It is necessary that

$$\langle\psi_j| E_k^\dagger E_l |\psi_i\rangle \quad = \quad 0, \tag{2.3.3}$$

for $i \neq j$, otherwise the action of an error operator would destroy the distinguishability of orthogonal codewords. The non-trivial content $\alpha_{l,k}$ of condition (2.3.2) presents a stronger argument than the necessary condition, in that the assumed Hermition matrix α with entries $\alpha_{l,k} = \langle\psi_i| E_k^\dagger E_l |\psi_i\rangle$ does not depend on i. This argument is qualified since the action of any error mapping must take orthogonal codewords to orthogonal states as any projective measurement on the error space would reveal information thereby disturbing the state of the system. Furthermore, suppose $\mathcal{F}$ is a quantum operation with operations $\{F_m\}$ which are linear combinations of the $\{E_k,\ k \in \{1, \ldots, \mathbf{d}^2\}$, that is, $F_m = \sum_k \beta_{m,k} E_k$ for some matrix $\beta_{m,k}$ of complex numbers. The action of the error operators F_m on a code basis state is given by

$$|\psi_i\rangle \otimes |E\rangle \mapsto \sum_m F_m |\psi_i\rangle \otimes |e_m\rangle \tag{2.3.4}$$

where the states $|e_m\rangle$ denote an orthonormal basis for the environment. Recovery is accomplished if there exist an operator $\mathcal{R}$ with operation $\{R_n\}$ such that

$$\sum_n R_n^\dagger R_n = I \tag{2.3.5}$$

and

$$\sum_{m,n} R_n F_m |\psi_i\rangle \otimes |e_{m,n}\rangle \quad \propto \quad |\psi_i\rangle \otimes |e_{m,n}\rangle . \tag{2.3.6}$$

Since $|e_{m,n}\rangle$ does not depend on i, it follows that

$$R_n F_m |\psi_i\rangle = \lambda_{m,n} |\psi_i\rangle , \tag{2.3.7}$$

$\lambda_{m,n} \in \mathbb{C}$. Therefore, $\{R_n F_m\}$ acts trivally on the code space $\sum_i |\psi_i\rangle$. With the assumption of completeness, that is condition (2.3.5), on $\{R_n\}$, we have it that

$$\begin{aligned} F_u^\dagger F_m |\psi_i\rangle &= F_u^\dagger \left(\sum_n R_n^\dagger R_n \right) F_m |\psi_i\rangle \\ &= \sum_n \lambda_{u,n}^* \lambda_{n,m} |\psi_i\rangle , \end{aligned} \tag{2.3.8}$$

thereby illustrating the trival action of $F_u^\dagger F_m$ on the basis states $\{|\psi_i\rangle\}$. Considering

$$\langle\psi_j| F_u^\dagger F_m |\psi_i\rangle = \alpha_{u,m} \delta_{ij}, \tag{2.3.9}$$

where $\alpha_{u,m} = \sum_n \lambda_{u,n}^* \lambda_{n,m}$, the error correcting condition follows.

To show that condition (2.3.2) is also sufficient, we need to illustrate that the recovery operator $\mathcal{R}$ can be explicitly constructed. By assumption, we have it that α is a Hermitian matrix and can therefore be diagonalised,

$$\langle\psi_j| F_u^\dagger F_m |\psi_i\rangle = \alpha_m \delta_{um} \delta_{ij}, \tag{2.3.10}$$

where $\sum_m \alpha_m = 1$ follows from the normalisation condition. For each n with $R_n \neq 0$ make the correspondence

$$R_n = \frac{1}{\sqrt{\alpha_n}} \sum |\psi_i\rangle \langle\psi_i| F_n^\dagger. \tag{2.3.11}$$

The set of recovery operators R_n satisify the normalisation condition since $\sum_n R_n^\dagger R_n = \sum_{n,i} \frac{1}{\alpha_n} F_n |\psi_i\rangle \langle\psi_i| F_n^\dagger$. The action of R_n on $F_m |\psi_i\rangle$ is given by

$\sqrt{\alpha_n}\delta_{mn}\,|\psi_i\rangle$. To show that this correction procedure is legitimate, we have it that

$$\sum_{n,m} R_n F_m\,|\psi_i\rangle \otimes |e_{m,n}\rangle = |\psi_i\rangle \otimes \left(\sum_n \sqrt{\alpha_n}\right)|e_{m,n}\rangle\,, \tag{2.3.12}$$

thus we can recover the code space from the effects of error.

The *distance* of the code $\mathcal{C}$ is the minimal weight of E in $\mathcal{E}_n$ such that condition (2.3.2) does not hold. An $[\![n,k]\!]_{\mathbf{d}}$ code $\mathcal{C}$ with distance d is called an $[\![n,k,d]\!]_{\mathbf{d}}$ quantum code.

Theorem 2. If a quantum code $\mathcal{C}$ corrects errors of weight $\leq \Re$ then $\mathcal{C}$ is $2\Re$ error-detecting. If $\mathcal{C}$ detects errors of weight $\leq \Im$ then $\mathcal{C}$ is $\lfloor \Im/2 \rfloor$ error-correcting.

Proof: For all m with $wt(E_m) \leq 2\Re$, write $E_m = E_k^\dagger E_l$ such that $wt(E_k^\dagger)$, $wt(E_l) \leq \Re$. Then $\langle\psi_j|E_m|\psi_i\rangle = \langle\psi_j|E_k^\dagger E_l|\psi_i\rangle = \alpha_{l,k}\delta_{ij}$. Thus, $\mathcal{C}$ is $2\Re$ error detecting where $\alpha_m = \alpha_{l,k}$.

For k,l with $wt(E_k^\dagger), wt(E_l) \leq \lfloor \Im/2 \rfloor$ then $E_k^\dagger E_l$ has weight $\leq \Im$. The product $E_k^\dagger E_l$ has support $\omega^{k,l}E_{k,l}$ and therefore $\langle\psi_j|E_k^\dagger E_l|\psi_i\rangle = \omega^{k,l}\langle\psi_j|E_{k,l}|\psi_i\rangle = \omega^{k,l}\alpha_{l,k}\delta_{ij}$ since $\mathcal{C}$ is $\Im$ error detecting. Writing $\alpha'_{l,k} = \omega^{k,l}\alpha_{l,k}$, then $\langle\psi_j|E_k^\dagger E_l|\psi_i\rangle = \alpha'_{l,k}\delta_{ij}$ and $\mathcal{C}$ is therefore $\lfloor \Im/2 \rfloor$ error-correcting.

Theorem 3. An $[\![n,k,d]\!]_{\mathbf{d}}$ quantum code detects errors of weight $\leq d-1$ and corrects errors of weight $\leq \lfloor (d-1)/2 \rfloor$.

Proof: Since $\mathcal{C}$ has distance d there exists an error E_m of weight $d-1$ such that $\langle\psi_j|E_m|\psi_i\rangle = \alpha_m\delta_{ij}$, thereby, illustrating $\mathcal{C}$ to be $d-1$ error-detecting. Writing E_m as the product of pairs $E_k^\dagger E_l$ with $wt(E_k^\dagger)$ and $wt(E_l) \leq \lfloor (d-1)/2 \rfloor$ then $\langle\psi_j|E_m|\psi_i\rangle = \langle\psi_j|E_k^\dagger E_l|\psi_i\rangle = \alpha_{l,k}\delta_{ij}$ and $\mathcal{C}$ is $\lfloor (d-1)/2 \rfloor$ error-correcting.

2.4 A Quantum Qudit Code

Quantum error correction was first demonstrated by Steane [83] and by Calderbank and Shor [14]. A group-theoretic formalism was introduced by Gottesman [31] and Calderbank *et al.* [12] and led to the discovery of many more quantum error correcting codes. Since then the theory of quantum error correction has matured remarkably quickly.

Let us consider information transmission over a channel that protects a single qu**d**it against error. Introduced by Shor [79], the quantum $[\![9, 1]\!]_2$ code illustrates that quantum error-correction is possible. The code proposed by Shor is a quantum bit generalisation of the classical 3-bit repetition code, and is characterised by specifying two basis states for the code subspace. These basis states are referred to as $|\bar{0}\rangle$, the *logical 0*, and $|\bar{1}\rangle$, the *logical 1*, and are written

$$|\bar{0}\rangle = [\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)]^{\otimes 3} \tag{2.4.1}$$

$$|\bar{1}\rangle = [\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)]^{\otimes 3} \tag{2.4.2}$$

Since the encoded qubit is written in the guise of entanglement, the information content is dispersed among the nine qubits where it is then said to be encoded nonlocally with protection from decoherence ensured because of this nonlocal property of encoded information.

We seek to extend error correction beyond the binary setting it was original envisaged and to which most results are derived. Suppose we wish to protect an unknown single quantum state of $\mathbb{C}^{\mathbf{d}}$ that has been prepared as

$$|\psi\rangle = \sum_{i=0}^{\mathbf{d}-1} \alpha_i |i\rangle \tag{2.4.3}$$

for $\alpha_i \in \mathbb{C}$ and $\sum_{i=0}^{\mathbf{d}-1} |\alpha_i|^2 = 1$.

Following Shor, we consider the $[\![9, 1]\!]_{\mathbf{d}}$ quantum code that encodes the qu**d**it basis states in nine qu**d**its,

$$|i\rangle \mapsto |\bar{i}\rangle = \frac{1}{\sqrt{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{ij} |jjj\rangle \otimes \frac{1}{\sqrt{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{j} |jjj\rangle \otimes \frac{1}{\sqrt{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{ij} |jjj\rangle \tag{2.4.4}$$

where $i \in \{0, \ldots, \mathbf{d}-1\}$ with ω is a $\mathbf{d}^{\text{th}}$ root of unity. The process of error-correction is made more transparent when the code (2.4.4) is viewed as a composition of two distinct mappings. Firstly, an inner mapping is given by

$$|i\rangle \mapsto \frac{1}{\sqrt{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{ij} |j\rangle \otimes \frac{1}{\sqrt{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{ij} |j\rangle \otimes \frac{1}{\sqrt{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{ij} |j\rangle \,. \tag{2.4.5}$$

This mapping protects the code from a phase error that, depending on the basis state, changes the phase of the the state to an altered phase. The phase error Z_k has the action

$$Z_k |j\rangle = \omega^{jk} |j\rangle \,. \tag{2.4.6}$$

The second mapping is a repetition code that encodes a single qu**d**it in three qu**d**its

$$|j\rangle \mapsto |jjj\rangle \,. \tag{2.4.7}$$

This code allows us to correct qu**d**it flip errors with action on $|j\rangle$ defined as

$$X_l |j\rangle = |j + l \pmod{\mathbf{d}}\rangle \,, \tag{2.4.8}$$

for $l \in \{0, \ldots, \mathbf{d}-1\}$. While a three qu**d**it code would suffice to protect against a single qu**d**it flip, we need to repeat the three qu**d**it cluster to protect against phase errors. Should these individual mappings be shown to correct a single qu**d**it flip and phase flip error, then the concatenated code (2.4.4) will correct an arbitrary error in one of the nine qu**d**its.

2.5 Qudit Error Correction

Error-correction has its foundations in classical theory, however, considering such thought within the quantum setting presents difficulties not evident in the classical setting. The classical majority voting scheme requires a measurement of bits to correct errors. However, a projective measurement taken on the information state will collapse the superposition leading to the loss of information. Furthermore, we note that where once we protected information by making copies of the information, the no-cloning theorem [60] maintains the quantum information cannot be copied. We take Shor's quantum $[\![9, 1]\!]_2$ binary repetition code [79] and generalise it to the qu**d**it setting as given in equation 2.4.4. We look at how error-correction on such a code may be addressed with respect to two procedures. The first procedure requires the use of non-demolition measurements [22] to construct the error syndrome. Recently, Lu and Marinescua [54] have provided a nondemolition algorithm argument that describes an error-correction procedure for use with the Steane code. The second procedure views error-correction and syndrome diagnosis with respect to a projective formalism.

A single qu**d**it may be encoded in three qu**d**its, and the corresponding superposition state is written

$$\sum_{i=0}^{\mathbf{d}-1} \alpha_i \left|i\right\rangle \mapsto \sum_{i=0}^{\mathbf{d}-1} \alpha_i \left|iii\right\rangle. \tag{2.5.1}$$

For a given three qu**d**it state $|a, b, c\rangle$, we have it that any measurement performed directly on the states leads to a collapse of the superposition state. To avoid such an outcome, a quantum nondemolition measurement [22] is instead performed on pairs of qu**d**its. Quantum nondemolition measurements were originally envisaged to be a measurement of a "total" photon number which

would determine whether quantum jumps due to loss had occurred while still preserving the superposition state [22]. Further, quantum error-correction requires nondemolition measurements of the error syndrome in order to preserve the quantum state [54]. Such nondemolition measurements allows us to construct the error syndrome and on examining the syndrome an error correcting procedure should be able to decide whether; an error has occurred for which no correcting action is take, one error has occurred for which we apply the corresponding Pauli transformation to the state in error, or lastly, if two or more errors have occurred for which a quantum error-correcting code capable of correcting a single error will fail. We now describe an alorithm based on the qu**d**it repetition code. Without loss of generality, let us consider a qu**d**it flip X_l acting on the first qu**d**it of the superposition,

$$\begin{aligned} \sum_{j=0}^{\mathbf{d}-1} \alpha_j \left|jjj\right\rangle &\mapsto (X_l \otimes I \otimes I) \sum_{j=0}^{\mathbf{d}-1} \left|jjj\right\rangle \\ &= \sum_{j=0}^{\mathbf{d}-1} \alpha_j \left|j+ljj\right\rangle. \end{aligned} \tag{2.5.2}$$

Figure 2.3 describes the circuit that returns one of the syndrome's values $(a+b)$. A similar argument returns the remaining syndrome values. The returned values $(b+c), (a+c)$ and $(a+b)$, with addition in $\mathbb{Z}_{\mathbf{d}}$, constitute a *syndrome* that reveals the error location. Figure 2.3 consists of a pair of CNOT gates. Both CNOT gates target the ancilla qudit prepared in the state $|0\rangle$ where it is necessary that measurement of the ancilla should not influence the superposition state of the system. A nondemolition measurement performed on the state $|j+ljj\rangle$ returns the syndrome $(b+c, a+c, a+b) = (2j, 2j+l, 2j+l)$. Since $(b+c) = 2j$ is the differing value of the syndrome, then the revealed error location is qudit position one. A unique solution for j can be found from $(b+c) = 2j$ when $\mathbf{d} > 2$ and $\mathbf{d} \neq 0$ (mod2) and l is given

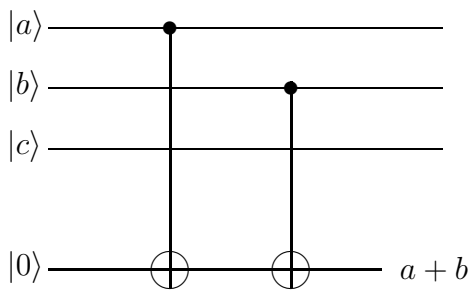


Figure 2.3: Non-demolition measurement circuit.

by $(a+c)-(b+c)$. Recovery of initial qu**d**it state is achieved by applying $(X_{-l}\otimes I\otimes I)$ to $\sum_{j=0}^{d-1}|j+ljj\rangle$.

As an alternative to error-corection via a nondemoltion preocedure, we may consider error-correction by describing a syndrome diagnosis under a projective formalism. Syndrome diagnosis via projective measurements on qubit repetition code is discussed in Nielsen and Chuang [60]. In respect of this, and to generalise, we have it that the projector $M_0=\sum_{j=0}^{\mathbf{d}-1}|jjj\rangle\langle jjj|$ measures no error having taken place on the logical qu**d**it state by returning a value of one. Similarly, the measurements $M_1=\sum_{i=0}^{\mathbf{d}-1}\sum_{j=0}^{\mathbf{d}-1}|j+ijj\rangle\langle j+ijj|$, $M_2=\sum_{i=0}^{\mathbf{d}-1}\sum_{j=0}^{\mathbf{d}-1}|jj+ij\rangle\langle jj+ij|$, and $M_3=\sum_{i=0}^{\mathbf{d}-1}\sum_{j=0}^{\mathbf{d}-1}|jjj+i\rangle\langle jjj+i|$ return a value of one, if the logical codeword has the qu**d**it flip error, $X_i,\ i\in\{1,\ldots,\mathbf{d}-1\}$, in position one, two, or three, respectively, of the logical qu**d**it. Suppose also that an error flips that state of the first qu**d**it so that the corrupted state is $|\psi\rangle=\sum_{j=0}^{d-1}\alpha_j|j+ljj\rangle$. Now note that $\langle\psi|M_1|\psi\rangle=1$ in this case. Furthermore, the projective measurement does not cause that state to change; it remains the same both before and after the measurement. Thus, the syndrome contains only information about what error has occurred and does not provide any information about the state itself. A repeat application of X_1 at the error location, with the identity elsewhere, on the logical state followed by the projector M_0 until the action of M_0 equals unity completes

the recovery.

Having protected the code against any possible qudit flip error, we follow in similar fashion to protect against phase errors. Therefore, we encode a single qudit using nine qudits according to

$$|i\rangle \mapsto \frac{1}{\sqrt[3]{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{kj} |jjj\rangle \otimes \sum_{j=0}^{\mathbf{d}-1} \omega^{kj} |jjj\rangle \otimes \sum_{j=0}^{d-1} \omega^{kj} |jjj\rangle \tag{2.5.3}$$

Suppose that a phase error occurs on the first qu**d**it, then the basis state are written as

$$\begin{aligned}(Z_{k'} \otimes I \otimes \cdots \otimes I) \frac{1}{\sqrt[3]{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{kj} |jjj\rangle \otimes \sum_{j=0}^{\mathbf{d}-1} \omega^{kj} |jjj\rangle \otimes \sum_{j=0}^{d-1} \omega^{kj} |jjj\rangle \\ = \frac{1}{\sqrt[3]{\mathbf{d}}} \sum_{j=0}^{\mathbf{d}-1} \omega^{(k+k')j} |jjj\rangle \otimes \sum_{j=0}^{\mathbf{d}-1} \omega^{kj} |jjj\rangle \otimes \sum_{j=0}^{d-1} \omega^{kj} |jjj\rangle \end{aligned} \tag{2.5.4}$$

for some $k' \in \{0, ..., \mathbf{d}-1\}$. The introduction of the error causes the relative sign of $|jjj\rangle$ in the first cluster of qu**d**its to change. As we do not measure the phase directly since such a measurement would destroy the superposition state then correction with respect to a non-demolition measurement is sought. The relative phase degree of pairs of the three qu**d**it clusters A, B, C motivate such a syndrome, in particular, $(B+C, A+C, A+B)$. As in the case of the qu**d**it-flip error, we determine the damaged cluster and error by comparing the results of the non-demolition measurement. The quantum circuit implemented in the qu**d**it flip case to generate a non-demolition syndrome may be used to generate a similar type of syndrome to detect a qu**d**it phase error. Alternatively, a circuit known as the Toffoli circuit can be implemented to give a similar outcome [86]. An n bit Toffoli gate, $\theta^{(n)}$ is defined as

$$(x_1, x_2, ..., x_{n-1}, y) \rightarrow (x_1, x_2, ..., x_{n-1}, y \oplus x_1 x_2 ... x_{n-1}) \tag{2.5.5}$$

The Toffoli gate allows us to take the product of the clusters' relative phase. The recorded ancilla, y, provides no individual information of a particular cluster's relative phase but rather it gives a combined measurement. The non-demolition syndrome associated with a Toffoli circuit is given by

$$(BC, AC, AB) \tag{2.5.6}$$

A particular value of the non-demolition syndrome, BC, is the result of implementing the mapping

$$(\omega^{kj} |j\rangle, \omega^{kj} |j\rangle, |0\rangle) \rightarrow (\omega^{kj} |j\rangle, \omega^{kj} |j\rangle, |0\rangle \oplus \omega^{kj} |j\rangle \omega^{kj} |j\rangle) \tag{2.5.7}$$

The returned ancilla $|0\rangle \oplus \omega^{(2k)j} |j\rangle$, along with similar values for the remaining elements of the syndrome permits error recovery.

While a single error $X_l Z_k$ acting on any one of the nine qudits will cause no irrevocable damage, should more than one error occur then the encoded information will be damaged. For example, if the first two qu**d**its in a cluster flip then we misdiagnose the error and attempt recovery by flipping the third. Similarly, the encoded information will be damaged if phase errors occur in two different clusters. A phase error will be introduced in a misguided attempt at correction.

Chapter 3

Unitary Error Bases

The theory of quantum error correction is increasingly well understood. The recent results of Shor and Steane marked the initial development of quantum coding theory. Subsequently, Calderbank and Shor have shown that good quantum codes exist and formulated their notion of quantum control codes on classical binary group codes. Introduced by Gottesman [32] and Calderbank, Rains, Shor, and Sloane [13], the stabilizer code is a class of quantum code and further popularises the connection between the quantum and classical realm. A large body of theory and practice has been developed around the stabilizer formalism.

Teleportation was demonstrated to be an initial feature of information processing that makes use of a correlated system between the two parties to send quantum information in a secure manner. The means for why this is possible in the quantum setting and impossible in a classical analogy rests with a particular type of connection known as entanglement. Entanglement describes the process by which quantum states can be described with reference to each other.

The majority of publications relating to quantum coding theory are restricted to the binary setting. Popular models of quantum computing such

as the Ion Trap, Cavity QED, and NMR require the storage of qubits for an extended period of time and that such qubits be isolated from the environment to reduce decoherence. In addition, quantum hardware must measure qubits and perform controlled operations. Any successful implementation of these models will need to meet these requirements. A research group directed at the National Institute for Standards and Technology have made studies into how qubits are carried by a single ion. Monroe *et al.* [58] have illustrated that the quantum XOR gate can be implemented in an ion trap using a series of laser pulses. Another model of quantum hardware makes use of nuclear magnetic resonance (NMR) technology to prepare a maximally entangled state of three qubits.

The role of an error model in quantum information processing is crucial to the theory of quantum processing as it permits quantum error correction, teleportation, and physical demonstrations of quantum models. It is inevitable that a system will interact with the environment and to ensure that information is accurately processed we are required to understand the effects of environment on information. The set of Pauli matrices for a single qubit describes an error model for the binary case of quantum information theory. Indeed, a class of quantum code called stabilizer codes are defined as the common eigenspace of a subset of Pauli matrices thus illustrating an important role of an error model.

Before any thoughts of scaling quantum computing and hardware to a non-binary setting, we need to consider error operators that act on higher dimensional quantum states. Such error bases should provide a natural extension from binary to non-binary and lead to codes over the ring of integers modulo $\mathbf{d}$.

3.1 Higher Dimensional Unitary Error Bases

The bit and phase flip error basis is an important prime in the theory of quantum information. This basis has permitted the construction of a class of quantum code that identifies the common eigenspace of an Abelian subgroup of the well known Pauli group. The Pauli error group $\mathcal{G}_1 = \langle X, Z, iI \rangle$ contains the set $\mathcal{P} = \{I, X, Z, Y\}$ of Pauli matrices given by

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{3.1.1}$$

Any matrix A in $\mathbb{C}^2$ can be expressed as a linear combination of the Pauli matrices. In particular,

$$A = \frac{1}{2}(\mathrm{tr}(A)I + \mathrm{tr}(X^\dagger A)X + \mathrm{tr}(Y^\dagger A)Y + \mathrm{tr}(Z^\dagger A)Z). \tag{3.1.2}$$

We concern ourselves with the generalisation of unitary error basis associated with the Pauli group for higher dimensional quantum systems and confine ourselves to the set errors that form a basis of $\mathbb{C}^{\mathbf{d}^2}$ since it is a well known premise of quantum coding theory that if a code can correct a set $\mathcal{E}$ of errors then the code also corrects the linear span of $\mathcal{E}$ [32, 65]. To motivate a description of such a set $\mathcal{E}$ for higher dimensional systems, let us again consider the Pauli group $\mathcal{G}_1 = \langle X, Z, iI \rangle$ with center $\zeta(\mathcal{G}_1) = \langle iI \rangle$. The group $\mathcal{G}_1/\zeta(\mathcal{G}_1)$ is isomorphic to the Klein four-group, $V = \mathbb{Z}_2 \times \mathbb{Z}_2$ with correspondence given by $I = E_{(0,0)}, X = E_{(1,0)}, Z = E_{(0,1)}, Y = E_{(1,1)}$, and multiplication on $E_{(i,j)}$ and $E_{(k,l)}$ defined as $E_{i,j}E_{k,l} = E_{(i+k,j+l)}$. An error basis for a higher dimensional quantum system is identified with an orthonormal basis $\mathcal{E} = \{E_1, \ldots, E_{\mathbf{d}^2}\}$ over $\mathbb{C}^{\mathbf{d}}$ of $\mathbf{d} \times \mathbf{d}$ complex matrices with inner product $(A, B) = \frac{1}{\mathbf{d}}\mathrm{tr}(A^\dagger B)$.

Result (2.2.4) gave an explicit construction of the evolution of the joint state of a qu**d**it information vector with the environment state. The dynamics

of such evolution are characterised by an arbitrary unitary transformation upon which it is *a priori* clear that an error basis $\mathcal{E}$ of dimension $\mathbf{d}^2$ exists. Let us now describe some properties of such error bases.

Definition 1. The error bases $\mathcal{E}$ and $\mathcal{E}'$ are said to be equivalent if and only if there exist unitary matrices $U, U^\dagger \in \mathcal{H}$ and some $c \in \mathbb{C}$ such that

$$\mathcal{E}' = \{c\, UE_{i,j}U^\dagger \mid E_{i,j} \in \mathcal{E}\}. \tag{3.1.3}$$

Consequently, we show that any unitary error basis over the Hilbert space $\mathbb{C}^2$ is equivalent to the Pauli basis.

Theorem 4. [46] If $\mathcal{E} = \{E_1, \ldots, E_{\mathbf{d}^2}\}$ is an error basis and $\mathcal{E}'$ is related to $\mathcal{E}$ by (3.1.3) then $\mathcal{E}'$ is an error basis.

Definition 2. A $\mathbf{d}\times\mathbf{d}$ matrix A is said to be diagonalisable if it can be written in the form $A = UDU^\dagger$, where D is a diagonal $\mathbf{d} \times \mathbf{d}$ matrix with the eigenvalues of A as its diagonal entries and U is a unitary $\mathbf{d} \times \mathbf{d}$ matrix whose columns are the eigenvectors corresponding to the eigenvalues in D.

Theorem 5. [41] If A is an $\mathbf{d} \times \mathbf{d}$ unitary matrix then A is diagonalisable.

Proof If A is a unitary matrix then A has an inverse matrix and A is said to be non-singular. Therefore, A has $\mathbf{d}$ linearly independent eigenvectors, whence, by the diagonalisation theorem, A is diagonalisable.

Theorem 6. Every unitary error basis over $\mathcal{H} = \mathbb{C}^2$ is equivalent to the Pauli basis $\mathcal{P} = \{I, X, Z, Y\}$.

Proof Let $\mathcal{A} = \{A_1, A_2, A_3, A_4\}$ be an arbitrary unitary error basis in $\mathcal{H} = \mathbb{C}^2$. The diagonalisation theorem states that a $\mathbf{n} \times \mathbf{n}$ matrix A is diagonalisable if and only if A has $\mathbf{d}$ linearly independent eigenvectors. Without loss of generality, let us consider the unitary basis matrix A_1. Since

$\mathcal{H}$ is a unitary space then by the eigen decompostion theorem, A_1 has an eigen decomposition and can thus be diagonalised. Therefore $U_1 A_1 U_1^\dagger = I$ for some unitary matrix U_1. Put $B_i = U_1 A_i U_1^\dagger,\ i = 2, 3, 4..$ By definition 1, the set $\mathcal{B} = \{I, B_2, B_3, B_4\}$ is equivalent with the set $\mathcal{A}$. Next consider the basis matrix B_2 with $B_2 \neq U_1 A_1 U_1^\dagger$. The trace orthogonality condition requires that $\mathbf{tr}(B_2)$ vanishes. Since B_2 is a unitary matrix there exists a unitary matrix U_2 that diagonalizes it. Therefore the condition that $\mathbf{tr}(U_1 A_i U_1^\dagger B_2)$ vanishes implies that $(U_2 B_2 U_2^\dagger)_{11} + (U_2 B_2 U_2^\dagger)_{22} = 0$. In particular, $(U_2 B_2 U_2^\dagger)_{11} = -(U_2 B_2 U_2^\dagger)_{22}$. Hence, by the trace condition it is necessary that the diagonalised matrix $U_2 B_2 U_2^\dagger$ contain the $+1$ and -1 eigenvalues, whence, $U_2 B_2 U_2^\dagger = Z$. Put $C_j = U_2 B_j U_2^\dagger,\ j = 3, 4$. By defintion 1, the set $\mathcal{C} = \{I, Z, C_3, C_4\}$ is equivalent with the set $\mathcal{B}$. To determine the nature of the matrices C_3, C_4, we note that the diagonal elements of both matrices are necessarily zero to satisfy $\mathrm{tr}(C_j) = \mathrm{tr}(Z^\dagger C_j) = 0,\ j = 3, 4$. Since $C_3^\dagger C_3 = I$ there exists a unitary matrix U_3 such that $U_3 C_3 U_3^\dagger$ can be represented as an anti-diagonal matrix. So $U_3 C_3 U_3^\dagger = X$ for some unitary matrix U_3. Put $D_4 = U_3 C_4 U_3^\dagger$. Then the set $\mathcal{D} = \{I, Z, X, D_4\}$ is equivalent to the set $\mathcal{C}$. Finally, we consider the unitary matrix D_4. Since D_4 must square to unity, we have it that $D_{4_{12}} D_{4_{21}} = D_{4_{21}} D_{4_{12}} = 1$. Also required is $\mathrm{tr}(X D_4) = 0$. To satisfy this constraint it is necessary that $D_{4_{12}} = -D_{4_{21}}$. Substituting we have $-D_{4_{21}}^2 = 1$, hence, $D_{4_{21}} = -i$. Since $D_{4_{21}} D_{4_{12}} = 1$ with $D_{4_{21}} = -i$ then $D_{4_{12}} = i$. Therefore, $D_4 = Y$ and the result follows.

While any unitary error basis over $\mathbb{C}^2$ is therefore equivalent to the Pauli basis, there exist non-equivalent unitary error bases over $\mathbb{C}^{\mathbf{d}}$, $\mathbf{d} > 2$. There are two known constructions which yield non-equivalent error bases. Introduced by Werner, the shift-and-multiply error basis [95] is a combinatorial

construction while the nice error basis is attributed to Knill [50]. The question of whether every nice error basis is of shift and multiply type was shown by Klappenecker and Rötteler to be false [49]. We review the properties of these known error bases with particular emphasis focused on Knill's algebraic formalism and its relation to (2.2.4) that describes the action of a unitary operator U on a qu**d**it information state adjoined to the environment space.

3.2 Shift and Multiply Bases

A Latin square, L, of dimension $\mathbf{d}$ is a $\mathbf{d} \times \mathbf{d}$ matrix such that each element of the set $\mathbb{Z}_{\mathbf{d}}$ is contained exactly once in each row and in each column. A complex Hadamard matrix H of order $\mathbf{d}$ is a matrix in $GL(\mathbf{d}, \mathbb{C})$ such that $H_{i,k} \in \mathbb{C}$, $0 \leq i, k < \mathbf{d}$, and $(H)^\dagger H = \mathbf{d}I$. Let $H^{(j)}$, $0 \leq j < \mathbf{d}$, be a sequence of complex Hadamard matrices.

Definition 3. A shift and multiply basis of unitary matrices in $\mathcal{H} = \mathbb{C}^{\mathbf{d}}$ is a set of unitary operators E_{ij} satisfying the orthogonality relation $\frac{1}{\mathbf{d}}\mathrm{tr}(E_{ij}^\dagger E_{kl}) = \delta_{ik}\delta_{jl}$ where the unitary operators E_{ij} are defined by

$$E_{ij} = P_j \text{ diag } (H_{i,k}^j : 0 \leq k < \mathbf{d}), \tag{3.2.1}$$

for $(i, j) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}$. The permutation matrix P_j is defined as $P_j(L(j,k),k) = 1$, for $0 \leq k < \mathbf{d}$ and 0 otherwise.

A unitary matrix is of shift and multiply type if it is a composition of a permutation operator and a diagonal matrix.

Theorem 7. [49] A shift-and-mulitply basis is a unitary error basis.

Proof We need to show that $\mathrm{tr}(E_{ij}^\dagger E_{kl}) = 0$ when $(i, j) \neq (k, l)$. If $j \neq l$, then the resulting $P_j^\dagger P_l$ has a vanishing diagonal, whence $\mathrm{tr}(E_{ij}^\dagger E_{kl}) = 0$ for

any i and k. If $j = l$ and $i \neq k$, then $\mathrm{tr}(E_{ij}^{\dagger} E_{kl})$ is just the inner product of the i^{th} and k^{th} rows of the complex Hadamard matrix, H^j, hence $\mathrm{tr}(E_{ij}^{\dagger} E_{kl}) = 0$.

In order to construct a shift and multiply basis on unitary matrices in $\mathbb{C}^{\mathbf{d}}$, we require Latin squares and Hadamard matrices of appropiate dimension. While Result (2.2.4) illustrates that the interaction of a qu**d**it information state with the environment always produces an error basis in any dimension then same can not be said for Shift and Multiply bases. Latin squares are not completely classified, and real Hadamard matrices exist only in dimensions $\mathbf{d} = 1, 2$ or $\mathbf{d} \equiv 0 \bmod 4$ [15].

3.3 Nice Error Bases

The process of entanglement is crucial in the theory of quantum information as regards its role in effects such as quantum error correction, dense coding and teleportation. Understanding how each of these quantum processes work explains the reason for justifiable consideration of error bases as the pillar of quantum information processing. The Pauli group of matrices illustrate a wealth of structure seen through the clear and concise description of quantum stabilizer codes. Indeed, the basis associated with the Pauli group is a particular case of the error base formalism known as nice error bases.

Definition 4. A group is a set G with a binary operation $* : G \times G \mapsto G$ that satisfies the following properties

1. $g * h \in G$ for all $g, h \in G$.
2. $(g * h) * k = g * (h * k)$ for all $g, h, k, \in G$.
3. There exists a unique element $e \in G$ satisfying
   (a) $g * e = e * g = g$ for all $g \in G$.

(b) There exists a unique $h \in G$ such that $g * h = h * g = e$.

Definition 5. A group is called Abelian or commutative if it satifies $g*h = h*g$ for all $g, h \in G$.

Definition 6. Let G be a finite group of order $\mathbf{d}^2$. A nice error basis in the Hilbert space, $\mathcal{H}$, is defined as a set $\mathcal{E} = \{E_g \mid g \in G\}$ of unitary matrices such that

1. E_1 is the identity matrix.
2. $\mathrm{tr}(E_g) = \mathbf{d}\delta_{g,1}\ \forall g \in G$.
3. $E_g E_h = \omega(g,h)E_{gh}\ \forall\ g, h \in G$.

for some $\omega(g,h) \in \mathbb{C}^*$ where gh denotes the product of g and h in G.

Definition 7. Let G and $\mathcal{E}$ be groups. A *homomorphism* ϕ from G to $\mathcal{E}$ is a map $\phi : G \mapsto \mathcal{E}$ such that

1. $\phi(gh) = \phi(g)\phi(h)$ for all $g, h \in G$.
2. $\phi(e_G) = e_{\mathcal{E}}$.

Let us suppose that ϕ is a homomorphism then ϕ must preserve the inverse map so that $\phi(g)\phi(g^{-1}) = \phi(gg^{-1}) = \phi(e_G) = e_{\mathcal{E}}$. This is true when $E_g^\dagger = \omega(g, g^{-1})^{-1}E_{g^{-1}}$ and by property 3 of definition 6, we have $\omega(g, g^{-1})^{-1}E_g E_{g^{-1}} = \omega(g, g^{-1})\omega(g, g^{-1})^{-1}E_{gg^{-1}} = e_{\mathcal{E}}$. To see that ϕ preserves the Hermitian inner product on $\mathcal{E}$, choose distinct elements g, h of G such that $gh^{-1} \neq e_G$. Since $\phi(g) = E_g$ and $\phi(h^{-1}) = \omega(h, h^{-1})^{-1}E_{h^{-1}}$, then $\mathrm{tr}(E_{gh^{-1}}) = \mathrm{tr}(\omega(h, h^{-1})\phi(gh^{-1}))$ $= \mathrm{tr}(\omega(h, h^{-1})\phi(g)\phi(h^{-1})) = \mathrm{tr}(\omega(h, h^{-1})\omega(h, h^{-1})^{-1}E_g E_{h^{-1}}) = \mathrm{tr}(E_g)\mathrm{tr}(E_{h^{-1}})$. By property 2 of definition 6, we have it that $\mathrm{tr}(E_g)\mathrm{tr}(E_{h^{-1}})$ vanishes. Thus $\mathrm{tr}(E_{gh^{-1}}) = 0$ and the result follows.

Definition 8. A *unitary representation* is a group homomorphism $\phi : G \mapsto \mathcal{E}$ from a group G into the group of unitary transformations which preserves Hermitian inner product on $\mathcal{E}$.

The function ω from $G \times G$ into C^* that maps (g, h) to $\omega(g, h)$ is called the *factor system* of $\mathcal{E}$. Furthermore, if we assume that $\det(E_g) = 1, \forall g \in G$, which we may without loss of generality by multiplication of E_g by a suitable constant, then $\omega(g, h)$ is a $\mathbf{d}^{\text{th}}$ root of unity. A *very nice* error basis is one in which this condition holds. The set $\mathcal{E}_2^{'} = \{I, \iota X, \iota Z, \iota Y\}$ is a very nice error basis over dimension 2 and is a representation of the quaternions. In particular, we may illustrate by multiplication how the quaternion algebra is nothing more than a subset of the Pauli algebra of space;

$$\iota Y.\iota Z = \iota X, \quad \iota Z.\iota X = \iota Y, \quad \iota X.\iota Y = \iota Z. \tag{3.3.1}$$

A set $\mathcal{E}$ satisfying conditions 1 and 3 of definition 6 is called a *projective representation* of the index group G.

Definition 9. Let G and $\mathcal{E}$ be groups. An action of G on $\mathcal{E}$ is defined to be a homomorphism $\phi : G \mapsto \mathcal{E}$.

Thus ϕ is a group action of G into $\mathcal{E}$ since there exists a function that maps G to a matrix group in $\mathcal{H}$ which preserves group multiplication. In particular, for $g \in G$ we have an $E_g \in \mathcal{E}$ such that for $gh \in G$ then $E_{gh} \in \mathcal{E}$ follows. By condition 2 of definition 6, $\mathcal{E}$ is an orthogonal basis with respect to inner product. As such, we have it that by the Group Orthogonality theorem [38], ϕ is an irreducible representation and therefore $\mathcal{E}$ is said to act *irreducibly* on $\mathcal{H}$ as $\{0\}$ and $\mathcal{H}$ are the only $\mathcal{E}$-invariant subspaces of $\mathcal{H}$. For $g \in G, g \neq 1$, condition 2 of definition 6 implies that corresponding elements of $\mathcal{E}$ are not multiples of the identity matrix. Therefore, the representation ϕ is *faithful* since $\ker(\phi) = \{g \in G \mid E_g = I\} = \{1\}$.

Definition 10. [28] Let F be a field. A *Lie algebra* over F is an F-vector space L, together with a bilinear map, the *Lie bracket*

$$L \times L \to L, \qquad (x, y) \mapsto [x, y],$$

satisfying the following properties:

1. $[x, x] = 0$ for all $x \in L$, (L1)

2. $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$ for all $x, y, z \in L$. (L2)

The Lie bracket $[x, y]$ is often referred to as the *commutor* of x and y. Denote by $\mathrm{GL}(n, F)$ the vector space of all $n \times n$ matrices over F with the Lie bracket defined by

$$[x, y] := xy - yx$$

where xy is the usual product of the matrices x and y. Thus nice error bases of dimension $\mathbf{d}^2$ are described as faithful irreducible projective representations of a finite Lie group of order $\mathbf{d}^2$. We turn to provide some properties to strengthen the statements made regarding nice error bases.

Theorem 8. [51] If $\mathcal{E}$ is a nice error basis then gh is a group operation. If $\mathcal{E}$ is very nice then $\omega(g, h)$ is $\mathbf{d}^{\text{th}}$ root of unity.

Proof Denote by $\mathcal{G}$ the group generated by $\mathcal{E}$ with center $\zeta(\mathcal{G}) = \langle \omega I \rangle$. The subgroup $\zeta(\mathcal{G})$ consisting of scalar multiples of the identity is normal. Since all elements of $\mathcal{G}$ will appear in exactly one coset of the normal subgroup $\zeta(\mathcal{G})$, it follows that any two elements of the same coset of $\zeta(\mathcal{G})$ correspond up to a scalar. Associating each element of the quotient group $\mathcal{G}/\langle \omega I \rangle$ with the element g of the unique E_g contained in it establishes an isomorphism between $\mathcal{G}/\langle \omega I \rangle$ and the multiplicative structure of the elements. The orthogonality

conditions show that for an identity $E_g = \alpha E_h$, with α a suitable scalar, we have $g = h$. If $\mathcal{E}$ is a very nice error basis then we note that the determinant on both sides of condition 3 of defintion 6 equals unity and thus $\omega(g, h)$ is a $\mathbf{d}^{\text{th}}$ root of unity.

Theorem 9. [47] Let $\mathcal{E} = \{E_g \mid g \in G\}$ be a set of unitary matrices parametrized by the elements of a finite group G. The set $\mathcal{E}$ is a nice error basis with index group G if and only if ϕ is a unitary irreducible faithful projective representation of G of dimension $\mathbf{d} = |G|^{1/2}$.

Proof If $\mathcal{E}$ is a nice error basis, then by arguments earlier $\mathcal{E}$ is a unitary irreducible faithful projective representation of the group G of dimension $\mathbf{d} = |G|^{1/2}$. Conversely, suppose ϕ is a unitary faithful irreducible projective representation of a group G of dimension $\mathbf{d} = |G|^{1/2}$. Then $\mathcal{E}$ satisfies conditions 1 and 3 of definition 6 since it is a projective representation. The dimension result gives condition 2 for $g = 1$. Similarly, as ϕ is faithful and projective then for $g \neq 1$, hence, for all $g \in G - \ker \phi$, we have $\text{tr}E_{g1} = 0$. Thus, ϕ satisfies condition 2 of definition 6.

Theorem 10. [51] A nice error basis is a unitary error basis.

Proof Let $\mathcal{E}$ be a nice error basis. With respect to property 3 of a nice error basis $\mathcal{E}$, ϕ is a homomorphism from G into $\mathcal{E}$ when $E_g^\dagger = \omega(g^{-1}, g)E_{g^{-1}}$. Let $g, h \in G$ with $g^{-1}h \neq e_G$. Then $\text{tr}(E_{gh^{-1}}) = \text{tr}(\omega(h, h^{-1})\phi(gh^{-1})) = \text{tr}(\omega(h, h^{-1})\phi(g)\phi(h^{-1})) = \text{tr}(\omega(h, h^{-1})\omega(h, h^{-1})^{-1}E_g E_{h^{-1}}) = \text{tr}(E_g)\text{tr}(E_{h^{-1}})$. $\mathcal{E}$ is a unitary error basis if and only if the Hermitian inner product between distinct pairs of $\mathcal{E}$ vanishes. By property 2 of a nice error basis, we have it that both $\text{tr}(E_g)$ and $\text{tr}(E_{h^{-1}})$ vanish. Thus $\text{tr}(E_{gh^{-1}}) = 0$ and the result follows.

Example 1. Denote by ω the primitive $\mathbf{d}^{th}$ root of unity. Let us consider the set of operators X_i and Z_j for $i, j \in \mathbb{Z}_{\mathbf{d}}$ such that $X_i|k\rangle = |k + i \pmod{\mathbf{d}}\rangle$ and

$Z_j |k\rangle = \omega^{kj} |k\rangle$. Further consider the group $\mathcal{G} = \langle X_{\mathbf{d}}, Z_{\mathbf{d}}, \omega I\rangle$. Then $\mathcal{G}/\langle \omega I\rangle$ is isomorphic to $\mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}$ and $\mathcal{E} = \{X_i Z_j \mid (i,j) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}\}$ is a nice error basis.

We show that the set in Example 1 is a basis for the Hilbert space $\mathbb{C}^{\mathbf{d}}$ and has the properties of a nice error basis. We show that X_i and Z_j for $i, j \in \mathbb{Z}_{\mathbf{d}}$ forms a basis for $\mathbb{C}^{\mathbf{d}}$ and then we describe the solution given by Knill to show that the set X_i and Z_j for $i, j \in \mathbb{Z}_{\mathbf{d}}$ is a nice error basis.

Theorem 11. Denote by ω the primitive $\mathbf{d}^{th}$ root of unity Let us consider $X_i|k\rangle = |k + i \pmod{\mathbf{d}}\rangle$ and $Z_j |k\rangle = \omega^{kj} |k\rangle$. Then $\mathcal{E} = \{X_i Z_j \mid (i,j) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}\}$ is a basis over $\mathcal{H}$.

Proof To show that elements of $\mathcal{E}$ are linearly independent and span $\mathcal{H}$, it suffices to show that the basis $\{|a\rangle \langle b|\},\ a, b \in \mathbb{Z}_{\mathbf{d}}$, for $\mathcal{H}$ may be expanded as a linear combination of elements in $\mathcal{E}$ as both sets of operators are of size $\mathbf{d}^2$. Let us consider $\mathcal{E}$ in the $\{|a\rangle \langle b|\}$ basis as

$$E_{i,j} = \sum_{k=0}^{\mathbf{d}-1} \omega^{jk} |k + i\rangle \langle k| \tag{3.3.2}$$

then $E_{i,j} |l\rangle = X_i Z_j |l\rangle = X_i \omega^{jl} |l\rangle = \omega^{jl} |l + i\rangle$. Suppose we may express $|a\rangle \langle b|$ as the linear combination $|a\rangle \langle b| = \sum_{(i,j)\in\ \mathbf{d}\times\ \mathbf{d}} \xi_{i,j} E_{i,j}$. Then coefficient $\xi_{i,j}$ is given by

$$\begin{aligned}
\xi_{i,j} &= \frac{1}{\mathbf{d}} \mathrm{tr}\left(E_{i,j}^{\dagger} |a\rangle \langle b|\right) \\
&= \frac{1}{\mathbf{d}} \mathrm{tr}\left(\sum_{k=0}^{\mathbf{d}-1} \omega^{-jk} |k\rangle \langle k + i| a\rangle \langle b|\right) \\
&= \frac{1}{\mathbf{d}}\ \omega^{-bj}\ \langle b + i|a\rangle \\
&= \frac{1}{\mathbf{d}}\ \omega^{-bj}\ \delta_{b+i,a}.
\end{aligned} \tag{3.3.3}$$

We show that with ξ_{ij} defined as these values then $|a\rangle\langle b|$ is in the span of $\mathcal{E}$. Now,

$$\begin{aligned} E_{i,j}^{\dagger}|a\rangle\langle b| &= E_{i,j}^{\dagger}\sum \xi_{k,l}E_{k,l} \\ &= \xi_{i,j}I + \sum_{k,l\neq i,j}\xi_{k,l}E_{i,j}^{\dagger}E_{k,l} \end{aligned} \tag{3.3.4}$$

where $E_{i,j}^{\dagger}E_{k,l}$ has vanishing trace. Since

$$\sum_{(i,j)\in \mathbf{d}\times \mathbf{d}} \frac{1}{\mathbf{d}}\omega^{-bj}\delta_{b+i,a}E_{i,j} = \sum_{(i,j)\in \mathbf{d}\times \mathbf{d}} \frac{1}{\mathbf{d}}\omega^{-bj}\delta_{b+i,a}\left(\sum_{k=0}^{\mathbf{d}-1}\omega^{jk}|k+i\rangle\langle k|\right)$$

$$\begin{aligned} &= \sum_{k=0}^{\mathbf{d}-1}\sum_{(i,j)\in \mathbf{d}\times \mathbf{d}} \frac{1}{\mathbf{d}}\omega^{(k-b)j}\delta_{b+i,a}|k+i\rangle\langle k| \\ &= \sum_{k=0}^{\mathbf{d}-1}\sum_{j=0}^{\mathbf{d}-1}\frac{1}{\mathbf{d}}\omega^{(k-b)j}|k+a-b\rangle\langle k| \\ &= |a\rangle\langle b| \end{aligned} \tag{3.3.5}$$

as $\sum_j \frac{1}{\mathbf{d}}\omega^{(k-b)j} = \delta_{k,b}$. Then $\langle b|\sum_{(i,j)\in \mathbf{d}\times \mathbf{d}}\xi_{i,j}E_{i,j}|a\rangle = \delta_{a,b}$. Hence, $|a\rangle\langle b| = \sum_{(i,j)\in \mathbf{d}\times \mathbf{d}}\xi_{i,j}E_{i,j}$ and the result follows.

Let X_i and Z_j be the linear operators acting on the space $\mathbb{C}^{\mathbf{d}}$ that are defined by the matrices with entries $\mathrm{X}_{m,n} = \delta_{m,n-i(\mathrm{mod}\ \mathbf{d})}$, $\mathrm{Z}_{m,n} = \omega^{mj}\delta_{m,n}$ respectively.

Theorem 12. [2] The operators X_i and Z_j form a nice error basis.

Proof Since $X_1Z_1 = \omega Z_1X_1$ then

$$\begin{aligned} \mathrm{X}_i\mathrm{Z}_j &= \omega^{ij}\mathrm{Z}_j\mathrm{X}_i \\ (\mathrm{X}_i\mathrm{Z}_j)(\mathrm{X}_k\mathrm{Z}_l) &= \omega^{il-jk}(\mathrm{X}_k\mathrm{Z}_l)(\mathrm{X}_i\mathrm{Z}_j) \\ (\mathrm{X}_i\mathrm{Z}_j)(\mathrm{X}_k\mathrm{Z}_l) &= \omega^{-jk}\mathrm{X}_{i+k}\mathrm{Z}_{j+l}. \end{aligned} \tag{3.3.6}$$

The Hermitian transposes of X_i and Z_j are obtained by raising to the power $\mathbf{d}-1$, whence,

$$(\mathrm{X}_i)^\dagger = (\mathrm{X}_i)^{\mathbf{d}-1}, \qquad (\mathrm{Z}_j)^\dagger = (\mathrm{Z}_j)^{\mathbf{d}-1} \tag{3.3.7}$$

and

$$\mathrm{X}^{\mathbf{d}} = \mathrm{Z}^{\mathbf{d}} = I. \tag{3.3.8}$$

Using the above expressions we obtain

$$\mathrm{tr}\left((\mathrm{X}_i\mathrm{Z}_j)^\dagger(\mathrm{X}_k\mathrm{Z}_l)\right) = \mathrm{tr}\left(\omega^{-(k-i)(l-j)}\mathrm{X}_{(k-i)}\mathrm{Z}_{(l-j)}\right). \tag{3.3.9}$$

Noting that $\mathrm{tr}(\mathrm{X}_i\mathrm{Z}_j) = \mathbf{d}\ \delta_{i,0}\ \delta_{j,0}$, the result follows.

Similarly, we may consider the case of an n-qu**d**it error basis. With reference to the tensor product construction (1.1.10) of quantum systems, a $\mathbf{d}^n$-dimensional Hilbert space $\mathcal{H}^n$ is identified with the n-fold tensor product of $\mathbf{d}$-dimensional Hilbert spaces $\mathcal{H}$. Therefore, an element $E_{i,j}$ of the group $\mathcal{G}^{\otimes n}$ may be written as $E_{i,j} = \omega^\alpha(X_{i_1}Z_{j_1}) \otimes (X_{i_2}Z_{j_2}) \otimes \cdots \otimes (X_{i_n}Z_{j_n})$ where $\alpha \in \{0,\ldots,\mathbf{d}-1\}$ and $((i_1,j_1),(i_2,j_2),\ldots,(i_n,j_n)) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n$. The group $\mathcal{G}^{\otimes n}/\langle\omega I\rangle$ is isomorphic to $\mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n$ and thus $\mathcal{E}^{\otimes n} = \{E_{i,j} \mid (i,j) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n\}$ is a nice error basis.

The nice error basis $\mathcal{E}$ in Example 1 is also a shift-and-multiply basis for the case $\mathbf{d}=3$. Consider the Latin square $L = (j-i \bmod 3)$ for $(i,j) \in \mathbb{Z}_3 \times \mathbb{Z}_3$, and the complex Hadamard matrix $H = (\omega^{kl})$ for $(k,l) \in \mathbb{Z}_3$ with $\omega = e^{2\pi i/3}$. Then

$$L = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix} \qquad \text{and} \qquad H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}. \tag{3.3.10}$$

By equation 3.2.1, we have it that the basis matrices E_{01} and E_{12} are given by

$$E_{01} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad \text{and} \qquad E_{12} = \begin{pmatrix} 0 & 0 & \omega^2 \\ 1 & 0 & 0 \\ 0 & \omega & 0 \end{pmatrix}. \tag{3.3.11}$$

The question that arises is to decide whether every nice error basis is of shift and multiply type. This problem was posed by Schlingemann and Werner [73]. Klappenecker and Rötteler [49] solved this problem by illustrating that there exist shift and multiply error bases that are not nice error bases. A *wicked error basis* is an error basis that not equivalent to a nice error basis.

Example 2. Let $\mathcal{E}_\alpha$ be the shift-and-multiply basis associated with L, H_α where

$$L = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \\ 1 & 2 & 3 & 0 \end{pmatrix} \qquad H_\alpha = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & e^{i\alpha} & -e^{i\alpha} \\ 1 & -1 & -e^{i\alpha} & e^{i\alpha} \end{pmatrix} \tag{3.3.12}$$

Let $Q^\times$ denote the multiplicative group of rationals. If $\alpha \in Q^\times$, then $\mathcal{E}_\alpha$ is not equivalent to a nice error basis. To see this, suppose there exist $U, U^\dagger \in U(4)$, and a suitable scalar $a_{i,j}$ such that the set $\{a_{i,j} U E_{i,j} U^\dagger | i, j = 0 \ldots 3\}$ is a nice error basis. We may assume that the group G generated by the matrices $a_{i,j} U E_{i,j} U^\dagger$ is finite. However, notice that $\mathcal{E}_\alpha$ contains the matrices $E_{0,0} = I$ and $E_{2,0} = \mathrm{diag}(1, -1, e^{i\alpha}, -e^{i\alpha})$. Consequently, we have it that $(a_{0,0} U E_{0,0} U^\dagger)(a_{2,0} U E_{2,0} U^\dagger)^{-1} = a_{0,0} {a_{2,0}}^{-1} U E_{0,0} E_{2,0}^\dagger U^\dagger = a_{0,0} {a_{2,0}}^{-1} U E_{2,0} U^\dagger$ is an element of the group G. Since G is finite, it follows that $a_{0,0} {a_{2,0}}^{-1} U E_{2,0} U^\dagger$ and hence $a_{0,0} {a_{2,0}}^{-1} E_{2,0}$ is of finite order. Then $a_{0,0} {a_{2,0}}^{-1}$ and $a_{0,0} {a_{2,0}}^{-1} e^{i\alpha}$ are necessarily roots of unity. This implies that $e^{i\alpha}$ would also have to be a root of unity, thus contradicting the assumption that $\alpha \in Q^\times$.

Chapter 4

The Stabilizer Formalism

Stabilizer codes were introduced independently by Gottesman [32] and Calderbank *et.al.* [13] and rank among the most widely studied and practised of all quantum error-correcting codes. The premise of the stabilizer formalism is that a quantum code $\mathcal{Q}$ can be efficiently described by a subgroup of the error group $\mathcal{G}^n$. The subgroup that describes the quantum code is called the stabilizer of $\mathcal{Q}$ and is denoted by $\mathcal{S}$. The stabilizer of $\mathcal{Q}$ is defined to be the set of operators $\mathcal{M}$ of $\mathcal{G}^n$ for which the condition $\mathcal{M}\left|\psi\right\rangle = \left|\psi\right\rangle$ is satisfied for all codewords $\left|\psi\right\rangle$. Thus $\mathcal{S}$ is a particular subgroup of the error group that maintains the common $+1$-eigenspace of $\mathcal{Q}$. Given the dimension in which the stabilizer formalism was originally considered, we have it that the error operators in $\mathcal{G}^n$ either commute or anti-commute. This property of the error group elicits a relatively simple procedure to detect errors that occur within the code. The set of all operators that commute with the stabilizer is called the centralizer. Any error operator that lies outside of the centralizer anti-commutes with the code stabilizer and can therefore be detected. Since the stabilizer is necessarily Abelian then the centralizer is the normalizer of the error group.

The stabilizer formalism permits a correspondence with classical linear

codes over $\mathbb{F}_2$. Such correspondence allows us to relate the normalizer of the error group to the dual space of a linear code and through which we can use well-established techniques to determine the error operators that go undetected by the quantum code.

4.1 Basic Definitions

Consider the Galois field $\mathbb{F}_{p^m}$ with $\mathbf{d} = p^m$ elements for some prime p and positive integer m. Let $\mathbb{F}_{\mathbf{d}}^{2n}$ denote the set of all vectors of length $2n$ over $\mathbb{F}_{\mathbf{d}}$. Then $\mathbb{F}_{\mathbf{d}}^{2n}$ is a vector space of dimension $2n$ over $\mathbb{F}_{\mathbf{d}}$.

Definition 11. A subspace C of $\mathbb{F}_{\mathbf{d}}^{2n}$ is called a linear code of length $2n$ over $\mathbb{F}_{\mathbf{d}}$.

Definition 12. Define a bilinear form on $\mathbb{F}_{\mathbf{d}}^{2n}$ to be the inner product (1.2.5). The code C is orthogonal if for all codewords i, j contained in C the inner product vanishes. The dual code $C^{\perp}$ is the set of elements $k \in \mathbb{F}_{\mathbf{d}}^{2n}$ that are orthogonal to C. In particular, $C^{\perp} = \{k \in \mathbb{F}_{\mathbf{d}}^{2n} \mid \langle k|i\rangle = 0 \ \forall \ i \in C\}$.

A quantum error E of a $\mathbf{d}$-dimensional system is a linear operator acting on the Hilbert space $\mathbb{C}^{\mathbf{d}}$. A basis for a set of errors acting on $\mathbb{C}^{\mathbf{d}}$ is given by construction (2.2.4) and is shown to be $\mathcal{E} = \{X_i Z_j \mid (i,j) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}\}$. In particular, $\mathcal{E}$ is a nice error basis on $\mathbb{C}^{\mathbf{d}}$. Similarly, the set $\mathcal{E}^n = \{E_{i,j} \mid (i,j) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n\}$ is a nice error basis on the Hilbert space $\mathbb{C}^{\mathbf{d}^n}$. Denote by $\mathcal{G}^n$ the group generated by the elements of $\mathcal{E}^n$, hence, $\mathcal{G}^n = \{\omega^{\alpha} E_{i,j} \mid (i,j) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n, \ \alpha \in \mathbb{Z}_{\mathbf{d}}\}$. The group $\mathcal{G}^n$ has a number of interesting properties used in the construction of good quantum codes. In particular,

1. Each operator $E_{i,j} \in \mathcal{G}^n$ is a unitary.

2. $(E_{i,j})(E_{k,l}) = \omega^{jk-il}(E_{k,l})(E_{i,j})$.

3. $(E_{i,j})(E_{k,l}) = \omega^{-jk} E_{i+k,j+l}$.

Let us look first at the qubit case. The group $\mathcal{G}^n/\langle iI \rangle$ has order 2^{2n} and is isomorphic to $\mathbb{F}_{\mathbf{2}}^{2n}$. We write a correspondence between the group $\mathcal{G}^n/\langle iI \rangle$ and the vector space $\mathbb{F}_{\mathbf{2}}^{2n}$ as

$$E_{i,j} = \otimes_{z=1}^{n} X_{i_z} \cdot \otimes_{z=1}^{n} Z_{j_z} \equiv (i_1, i_2, \ldots, i_n | j_1, j_2, \cdots, j_n) = (i|j). \quad (4.1.1)$$

The properties of elements in $\mathcal{G}^n$ have a natural correspondence in $\mathbb{F}_{\mathbf{2}}^{2n}$. Given relation (4.1.1), multiplication and commutativity in $\mathcal{G}^n$ is thus

1. $(i|j)(k|l) = (-1)^{-jk}(i+k|j+l)$
2. $(i|j)(k|l) = (-1)^{jk-il}(k|l)(i|j)$

in $\mathbb{F}_{\mathbf{2}}^{2n}$, respectively.

Definition 13. The weight of $(i|j) \in \mathbb{F}_{\mathbf{2}}^{2n}$ is $\mathrm{wt}(i|j) = |\{z|\ i_z \neq 0 \text{ or } j_z \neq 0\}|$.

Definition 14. The minimum distance d of a linear code C equals the minimum weight of non-zero codewords.

4.2 Stabilizer Codes

Let $\mathcal{Q}$ be a quantum error correcting code. The stabilizer of $\mathcal{Q}$ is a formalism that describes a quantum code in terms of error operators of $\mathcal{G}^n$ which maintain a common eigenspace. We review the binary stabilizer formalism in this section and in the following section we generalise aspects of the binary stabilizer formalism.

Definition 15. A *stabilizer code* $\mathcal{Q}$ is a subspace of $\mathbb{C}^{2^n}$ that satifies the relation

$$\mathcal{Q} = \bigcap_{\mathcal{M} \in \mathcal{S}} \{|\psi\rangle \in \mathbb{C}^{2^n} \mid \mathcal{M}|\psi\rangle = |\psi\rangle\} \quad (4.2.1)$$

for some subgroup $\mathcal{S}$ of $\mathcal{G}^n$.

Quantum stabilizer codes are thus defined as the +1-eigenspace of the operators of a Abelian subgroup $\mathcal{S}$ of $\mathcal{G}^n$. Since $\mathcal{S}$ describes the set of operators that leave each state in the quantum code invariant therefore $\mathcal{S}$ is said to stabilize the code. By relation (4.1.1) and its commutative property, elements $\mathcal{M}_{i,j}$ and $\mathcal{M}_{k,l}$ of $\mathcal{S}$ commute if and only if $\sum_{z=0}^{n-1} j_z k_z - i_z l_z = 0$. Therefore, the task of finding an Abelian subgroup $\mathcal{S}$ for a K-dimensional, $K = \mathbf{d}^k$, quantum code $\mathcal{Q}$ is related to finding a $n-k$-dimensional subspace C of $\mathbb{F}_{\mathbf{d}}^{2n}$. Since the requirement $\sum_{z=0}^{n-1} j_z k_z - i_z l_z$ must vanish, we have it that $C \subseteq C^{\perp}$.

To determine the dimension of $\mathcal{Q}$ consider the generators of the code stabilizer $\mathcal{S}$. If $\mathcal{S}$ has independent generators $\mathcal{M}_i,\ i \in \{1, \ldots, n-k\}$ then the dimension of $\mathcal{Q}$ is 2^k, and therefore there are k encoded qubits. To illustrate this claim, we have it that $\mathcal{M}_i \in \mathcal{S}$ must satisfy $\mathcal{M}_i^2 = I$. Should $\mathcal{M}_i^2 = -I$ then $\mathcal{M}_i$ cannot have eigenvalue +1. For each $\mathcal{M}_i \neq \pm I$ the number of eigenvectors with eigenvalue +1 equals the number of eigenvectors with eigenvalues -1. Therefore, for such an $\mathcal{M}_i$ there exists an $\mathcal{N} \in \mathcal{G}^n$ that anti-commutes with $\mathcal{M}_i$,

$$\mathcal{M}_i \mathcal{N} \ = \ -\mathcal{N} \mathcal{M}_i. \tag{4.2.2}$$

In particular, $\mathcal{M}_i |\psi\rangle = |\psi\rangle$ if and only if $\mathcal{M}_i \mathcal{N} |\psi\rangle = -\mathcal{N} |\psi\rangle$. The action of $\mathcal{N}$ establishes an isomorphism between the +1 eigenstates of $\mathcal{M}$ and the -1 eigenstates of $\mathcal{M}$. Hence, there are $\frac{1}{2}(2^n) = 2^{n-1}$ mutually orthogonal states that satisfy

$$\mathcal{M}_1 |\psi\rangle = |\psi\rangle , \tag{4.2.3}$$

where $\mathcal{M}_1 \in \mathcal{S}$. Similarly, consider an $\mathcal{M}_2 \in \mathcal{G}^n$ that commutes with $\mathcal{M}_1$ with $\mathcal{M}_2 \neq \pm I, \pm \mathcal{M}_1$. We can choose an $\mathcal{N} \in \mathcal{G}^n$ that commutes with $\mathcal{M}_1$ but anti-commutes with $\mathcal{M}_2$. Such a choice of $\mathcal{N}$ preserves the +1 eigenstates

of $\mathcal{M}_1$ however within this space, it interchanges the +1 and -1 eigenstates of $\mathcal{M}_2$. It follows that the space satisfying

$$\mathcal{M}_1 |\psi\rangle = \mathcal{M}_2 |\psi\rangle = |\psi\rangle \tag{4.2.4}$$

has dimension 2^{n-2}. Following in a corresponding manner, if $\mathcal{M}_j$ is choosen to be independent of the set of stabilizers $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{j-1}\}$ then there exists an $\mathcal{N} \in \mathcal{G}_n$ that commutes with $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{j-1}$, but anti-commutes with some $\mathcal{M}_j$. Therefore, restricted to the space with $\mathcal{M}_1 = \mathcal{M}_2 = \cdots = \mathcal{M}_{j-1} = +1$, we note that $\mathcal{M}_j$ has again equal degeneracy, that is, the number of eigenvectors with eigenvalue +1 equals the number of eigenvectors with eigenvalues -1. The inclusion of $\mathcal{M}_j$ to the set of code stabilizers $\mathcal{S}$ will have the effect of reducing the dimension of $\mathcal{Q}$ by a factor of two. With $n-k$ generators in the stabilizer $\mathcal{S}$, the dimension of the remaining space, $\mathcal{Q}$, is $2^n(\frac{1}{2})^{n-k} = 2^k$. Given a quantum stabilizer code $\mathcal{Q}$, we write

$$P_{\mathcal{Q}} = \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M} \in \mathcal{S}} \mathcal{M} \tag{4.2.5}$$

to be the orthogonal projection operator onto $\mathcal{Q}$. We have it from result (1.1.18) that $P_{\mathcal{Q}}$ is idempotent and $P_{\mathcal{Q}} = P_{\mathcal{Q}}^{\dagger}$.

Theorem 13. The dimension of $\mathcal{Q}$ is 2^k.

Proof. For $E_{i,j} \in \mathcal{G}^n / \langle \omega I \rangle$ we have it that $\mathrm{tr}(E_{i,j}) = 0$ by condition 2 of Definition 6. Then the dimension of the code $\mathcal{Q}$ is

$$\begin{aligned} |\mathcal{Q}| &= \mathrm{tr}(P) \\ &= \mathrm{tr}\left(\frac{1}{|\mathcal{S}|} \sum_{\mathcal{M} \in \mathcal{S}} \mathcal{M} \right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2^{n-k}} \sum_{\alpha \in \ \mathbf{d}} \mathrm{tr}(I) \\
&= \frac{2^n}{2^{n-k}} \\
&= 2^k. \qquad (4.2.6)
\end{aligned}
$$

We also note that the $n-k$ stabilizer generators $\mathcal{M}_i$ act as check operators for the code $\mathcal{Q}$. The error operator E_l either commutes or anti-commutes with a stabilizer generator $\mathcal{M}_i$. Should E_l and $\mathcal{M}_i$ commute then for $|\psi\rangle \in \mathcal{Q}$, we have it that $\mathcal{M}_i E_l |\psi\rangle = E_l \mathcal{M}_i |\psi\rangle = E_l |\psi\rangle$. Therefore, the error E_l preserves the $+1$ eigenvalue value of $\mathcal{M}_i$. Correspondingly, should E_l and $\mathcal{M}_i$ anticommute then $\mathcal{M}_i E_l |\psi\rangle = -E_l \mathcal{M}_i |\psi\rangle = -E_l |\psi\rangle$ and the eigenvalue -1 is recorded by $\mathcal{M}_i$. Consider the stabilizer set $\{\mathcal{M}_i\}$, and error set $\{E_l\}$, and write $\mathcal{M}_i E_l = (-1)^{s_{i,l}} E_l \mathcal{M}_i$. The set of values $\{s_{i,l}\}$ constitute the *syndrome* for error E_l acting on $\mathcal{Q}$. The code $\mathcal{Q}$ is *nondegenerate* if the syndrome is distinct for all $E_l \in \mathcal{G}^n$. Elements of the code C are linearly independent if they are linearly independent over the vector space $\mathbb{F}_{\mathbf{d}}^{2n}$. The stabilizer formalism is a powerful expression in that a strength lies in the simplicity with which it detects and corrects errors. The set of error operators for which the code $\mathcal{Q}$ detects is precise and further distinguishes the role of error bases within quantum information theory.

4.3 On Stabilizer Equivalence

Coding theory has many established formalisms that elicit concise techniques for error correction and one such description is that given by class of codes called linear codes. Recall that a classical code $\mathcal{C}$ is a linear code if and only if $\mathcal{C}$ is a subspace of the vector space $\mathbb{F}_q^n$. Furthermore, a linear subspace $\mathcal{C}$ of dimension k over $\mathbb{F}_q^n$ can be specified if there exists a basis consisting of a

minimal set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ such that $\mathcal{C} = \{\sum_{i=1}^{k} \alpha_i \mathbf{v}_i \mid \alpha_i \in \mathbb{F}_q\}$. A basis set for the subspace $\mathbb{C}$ provides a succinct description of a linear code. We now consider the question [44] relating to the number of bases associated with a linear code.

Theorem 14. A binary linear code of dimension k has precisely

$$\frac{1}{k!} \prod_{i=0}^{k-1} (2^k - 2^i)$$

different bases.

Proof. The set of code generators $\{\mathbf{v}\}$ can be chosen from among the 2^k codewords. The first generator $\mathbf{v}_1$ can be chosen in $2^k - 1$ ways. To maintain linear independence, the second generator $\mathbf{v}_2$ can be chosen in $2^k - 2$ ways. The third generator $\mathbf{v}_3$, independent of the first and second choices, can be chosen in $2^k - 4$ ways. We continue this process until a set of generators $\{\mathbf{v}_i\},\ i = 1, \ldots, k$, is chosen. By the product rule, the number of ways a set of k such generators can be chosen is

$$\prod_{i=0}^{k-1} (2^k - 2^i),$$

of which there are $k!$ arrangements and the result follows.

As a classical $(n, k, d)_q$ linear code $\mathcal{C}$ over $\mathbb{F}_q^n$ can be described efficiently by a k-dimensional basis set, a quantum stabilizer $[\![n, k, d]\!]_{\mathbf{d}}$ code $\mathcal{Q}$ over the Hilbert space $\mathbb{C}^{\mathbf{d}^n}$ can be described by a $\mathbf{d}^k$-dimensional subspace of $\mathbb{C}^{\mathbf{d}^n}$. A basis for $\mathbb{C}^{\mathbf{d}^n}$ is given by construction (2.2.4) and by result (4.1.1) extends to the set $\mathcal{E}^n = \{E_{i,j} \mid (i, j) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n\}$. Recall $\mathcal{G}^n$ to be the group generated by the elements of $\mathcal{E}^n$ then a stabilizer of $\mathcal{Q}$ describes a quantum code in terms of operators of $\mathcal{G}^n$ that maintain a common eigenspace. In particular, a stabilizer code $\mathcal{Q}$ is a subspace of $\mathbb{C}^{\mathbf{d}^n}$ that satifies $\mathcal{Q} = \bigcap_{\mathcal{M} \in \mathcal{S}} \{|\psi\rangle \in \mathbb{C}^{\mathbf{d}^n} \mid \mathcal{M}\,|\psi\rangle = |\psi\rangle\}$ for some stabilizer subgroup $\mathcal{S}$ of $\mathcal{G}^n$.

Theorem 15. A stabilizer code $\mathcal{Q}$ of dimension $\mathbf{d}^k$ has precisely

$$\mathbf{d}^{\frac{(n-k-1)(n-k)}{2}} \prod_{i=0}^{n-k-1} (\mathbf{d}^{(n-i)} - 1) \tag{4.3.1}$$

different stabilizer sets.

Proof. The codewords associated with a qu**d**it stabilizer code are given by

$$|\overline{c_1 \dots c_k}\rangle = \overline{X}_1^{c_1} \dots \overline{X}_k^{c_k} \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M} \in \mathcal{S}} \mathcal{M} \, |00 \dots 0\rangle. \tag{4.3.2}$$

The stabilizer code $\mathcal{Q}$ is defined as the common eigenspace of $\mathcal{M} \in \mathcal{S}$. There are $\mathbf{d}^{n-k}$ stabilizer elements that by definition maintain the codes $+1$-eigenspace in a nontrivial manner. Furthermore, conditioning the $\mathbf{d}^{n-k}$ stabilizer elements with $\mathbf{d}^k$ elements of the encoded $\bar{X}$, we find that there are a total of $\mathbf{d}^n$ elements that maintain the codes common eigenspace. Therefore, to count the numbers of ways that such elements can be chosen for a particular stabilizer code, we note there are $\mathbf{d}^n - 1$ choices for $\mathcal{M}_1$. $\mathcal{M}_2$ can then chosen independently in $\mathbf{d}^n - \mathbf{d}$ ways, followed by $\mathbf{d}^n - \mathbf{d}^2$ choices for M_3. In particular, $\mathcal{M}_l$ can be chosen to preserve the common eigenspace of $\mathcal{Q}$ in $\mathbf{d}^n - \mathbf{d}^{l-1}$ ways. In total there are

$$\begin{aligned} \prod_{i=0}^{n-k-1} \left(\mathbf{d}^n - \mathbf{d}^i\right) &= \mathbf{d}^{\sum_{i=0}^{n-k-1} i} \prod_{i=0}^{n-k-1} \left(\mathbf{d}^{(n-i)} - 1\right) \\ &= \mathbf{d}^{\frac{(n-k-1)(n-k)}{2}} \prod_{i=0}^{n-k-1} \left(\mathbf{d}^{(n-i)} - 1\right) \end{aligned} \tag{4.3.3}$$

choices for elements $\mathcal{M}_i$ that maintain the common eigenspace of a qu**d**it code $\mathcal{Q}$, and the result follows.

Theorem 16. The number of stabilizer sets in the Hilbert space $\mathbb{C}^{\mathbf{d}^n}$ that maintain a common eigenspace is given by

$$\mathbf{d}^{\frac{n-k-1(n-k)}{2}} \prod_{i=0}^{n-k-1} \left(\mathbf{d}^{2(n-i)} - 1\right).$$

Proof. The error group associated with an n-qu**d**it system is the n-fold product $\mathcal{G}^n = \{\omega^\alpha E_{ij} \mid (i,j) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n,\ \alpha \in \mathbb{Z}_{\mathbf{d}}\}$. To count the number of elements within $\mathcal{G}^n$ that maintain a $+1$-eigenspace we note that the first such element $\mathcal{M}'_1$ can be chosen in $\mathbf{d}^{2n} - 1$ ways. Let $\mathcal{M}'_2$ be another element of $\mathcal{G}^n$ that commutes with $\mathcal{M}'_1$ but is distinct from $\mathcal{M}'_1$. It can be shown that there then exists an $N'_2 \in \mathcal{G}^n$ that commutes with $\mathcal{M}'_1$ but fails to commute with $\mathcal{M}'_2$. Hence, M'_2 maintains the $+1$ eigenspace of $\mathcal{M}'_1$. It follows that the number of choices for $\mathcal{M}'_2$ that maintain a common eigenspace with $\mathcal{M}'_1$ is $\frac{\mathbf{d}^{2n}}{\mathbf{d}} - \mathbf{d}$. Following in a similar fashion, we note that for $\mathcal{M}'_l \neq \{\mathcal{M}'_1, \mathcal{M}'_2, \ldots, \mathcal{M}'_{l-1}\}$, then suitable N'_l can be found that commutes with $\mathcal{M}'_1, \mathcal{M}'_2, \ldots, \mathcal{M}'_{l-1}$ but does not commute with $\mathcal{M}'_l$. Whence, the number of choices for $\mathcal{M}'_l$ that preserve the commmon eigen space of the set $\{\mathcal{M}'_i\}$, $i = 1, \ldots, l-1$, is $\frac{\mathbf{d}^{2n}}{\mathbf{d}^{l-1}} - \mathbf{d}^{l-1}$. Therefore, the number of elements $\mathcal{M}'_i$ that maintain a common eigenspace of a code $\mathcal{Q}$ is given by

$$\begin{aligned}\prod_{i=0}^{n-k-1}\left(\frac{\mathbf{d}^{2n}}{\mathbf{d}^i} - \mathbf{d}^i\right) &= \mathbf{d}^{\sum_{i=0}^{n-k-1} i}\prod_{i=0}^{n-k-1}\left(\mathbf{d}^{2(n-i)} - 1\right) \\ &= \mathbf{d}^{\frac{n-k-1(n-k)}{2}}\prod_{i=0}^{n-k-1}\left(\mathbf{d}^{2(n-i)} - 1\right). \qquad (4.3.4)\end{aligned}$$

Theorem 17. The number of distinct stabilizer sets associated with $[\![n,k,d]\!]_{\mathbf{d}}$ qu**d**it codes over $\mathbb{C}^{\mathbf{d}^n}$ is

$$\prod_{i=0}^{n-k-1}(\mathbf{d}^{(n-i)} + 1).$$

Proof. The quotient obtained from equations (4.3.4) and (4.3.3) is given by

$$\prod_{i=0}^{n-k-1}\left(\frac{\mathbf{d}^{\frac{n-k-1(n-k)}{2}}\mathbf{d}^{2(n-i)} - 1}{\mathbf{d}^{\frac{n-k-1(n-k)}{2}}\mathbf{d}^{(n-i)} - 1}\right) = \prod_{i=0}^{n-k-1}\left(\mathbf{d}^{(n-i)} + 1\right) \qquad (4.3.5)$$

and the result follows.

4.4 Error Correcting Capabilities

When contructing a quantum code, we concern ourselves with two groups. First is the stabilizer subgroup $\mathcal{S}$ that characterises the code, and second, is the group called the normalizer of $\mathcal{S}$, denoted $\mathcal{N}(\mathcal{S})$, which contains elements that commute with $\mathcal{S}$ but which lie outside of $\mathcal{S}$.

Definition 16. Denote by $\mathcal{N}(\mathcal{S})$ the group of elements E of $\mathcal{G}^n$ such that $E\mathcal{M}E^\dagger \in \mathcal{S}$ for all $\mathcal{M} \in \mathcal{S}$. We call $\mathcal{N}(\mathcal{S})$ the normalizer of $\mathcal{S}$.

While elements of the normalizer commute with the stabilizer but lie outside of it they go undetected by the code since they maintain the codes common eigenspace. To determine those errors that go undetected by the code, we relate the requirement of commutativity among elements of the stabilizer to a classical analogy.

The requirement of commutativity of $\mathcal{S}$ degenerates to the adherence of the condition $\sum_{z=0}^{n-1} j_z k_z - i_z l_z = 0$ among elements of the stabilizer and thus fixes a correspondence between the normalizer $\mathcal{N}(\mathcal{S})$ of $\mathcal{G}^n$ and the linear subspace $C^\perp$ of $\mathbb{F}_{\mathbf{d}}^{2n}$. In particular, the set of all elements $E_{i,j}$ of $\mathcal{G}^n$ that commute with $\mathcal{S}$ corresponds to the set of vectors $(i|j)$ contained in $C^\perp$.

Theorem 18. [32, 60] Let $\mathcal{S}$ be the stabilizer group for the quantum code $\mathcal{Q}$. Suppose that $\{E_{i,j}\}$ is a set of elements in $\mathcal{G}^n$ such that $E_{i,j}^\dagger E_{k,l} \notin \mathcal{N}(\mathcal{S}) - \mathcal{S}$ for all (i,j) and $(k,l) \notin C^\perp \backslash C$. Then $\{E_{i,j}\}$ is a correctable set of errors for the code $\mathcal{Q}$.

Proof. Let $\mathcal{P}$ be the projection operator on $\mathcal{Q}$. We consider three cases.

1. Consider $E_i^\dagger E_j \in \mathcal{S}$. Then

$$\begin{aligned} E_i^\dagger E_j \mathcal{P} &= \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M}\in\mathcal{S}} E_i^\dagger E_j \mathcal{M} \\ &= \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M}\in\mathcal{S}} \mathcal{M} E_i^\dagger E_j \\ &= \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M}'\in\mathcal{S}} \mathcal{M}' \\ &= \mathcal{P}. \end{aligned} \tag{4.4.1}$$

   $\mathcal{P}$ is invariant under multiplication by $\mathcal{S}$ and hence $E_i^\dagger E_j$ is correctable.

2. Consider $E_i^\dagger E_j \in \mathcal{G}^n - \mathcal{N}(\mathcal{S})$. Then

$$\begin{aligned} E_i^\dagger E_j \mathcal{P} &= \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M}\in\mathcal{S}} E_i^\dagger E_j \mathcal{M} \\ &= -\mathcal{M}_1 E_i^\dagger E_j \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M}'\in\mathcal{S}\setminus\mathcal{M}_1} \mathcal{M}'. \end{aligned}$$

   Next

$$\begin{aligned} \mathcal{P} E_i^\dagger E_j \mathcal{P} &= -\mathcal{P}\mathcal{M}_1 E_i^\dagger E_j \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M}'\in\mathcal{S}\setminus\mathcal{M}_1} \mathcal{M}' \\ &= -\mathcal{P} E_i^\dagger E_j \frac{1}{|\mathcal{S}|} \sum_{\mathcal{M}\in\mathcal{S}} \mathcal{M} \end{aligned} \tag{4.4.2}$$

   when $E_i^\dagger E_j \in \mathcal{G}^n - \mathcal{N}(\mathcal{S})$. Therefore $\mathcal{P} E_i^\dagger E_j \mathcal{P} = 0$ and $E_i^\dagger E_j$ takes $\mathcal{Q}$ to an orthogonal subspace.

3. Consider $E_i^\dagger E_j \in \mathcal{N}(\mathcal{S}) - \mathcal{S}$. This is the set of operators that commute with all the elements of $\mathcal{S}$ but do not lie in $\mathcal{S}$. By Theorem 13, it follows that Q has dimension $\mathbf{d}^k$. Form the projector $\mathcal{P}'$ by adjoining the elements of $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ to $\mathcal{S}$. Then the image of $\mathcal{P}'$ is the quantum code $\mathcal{Q}'$ whose dimension is less than $\mathbf{d}^k$. Hence $E_i^\dagger E_j$ can not be corrected.

Finally, the minimum distance of a quantum stabilizer code $\mathcal{Q}$ relates to the classical minimum distance of $C^{\perp}\backslash C$. The minimum distance d of a stabilizer code $\mathcal{Q}$ is given by the minimum weight $\min\{\mathrm{wt}(i|j)\}$ of $(i|j) \in C^{\perp}\backslash C$. This result follows from Theorem 18, whence, the set of error operators $E_{i,j}$ corresponding to $C^{\perp}\backslash C$ are not detected by $\mathcal{Q}$. In summary, let $(i|j) \in \mathbb{F}_{\mathbf{d}}^{2n}$ be a basis for a code C over $\mathbb{F}_{\mathbf{d}}$. Then the set of $n-k$ stabilizers $\mathcal{M}_{i,j}$ define a quantum $[\![n, K, d]\!]_{\mathbf{d}}$ stabilizer code $\mathcal{Q}$ of length n, dimension $K = \mathbf{d}^k$ and a minimum distance d over $\mathbb{F}_{\mathbf{d}}$ which can correct a set of errors $E_{i,j} \in \mathcal{G}^n$ precisely when $(i,j)^{-1}(k,l) \notin C^{\perp}\backslash C$.

4.5 Encoding the Stabilizer

It is well known [12, 31] that a particular class of quantum code can be described according to the stabilizer formalism. By specifying an abelian subgroup of an error group associated with the Pauli basis, the stabilizer formalism elicits a concise description for many quantum codes. Under the action of each element of the abelian subgroup, the code remains invariant. Since the specified group is an abelian subgroup, its elements can be simultaneously diagonalized. We then have it that the stabilizer code is the common eigenspace having eigenvalue +1 of all elements of the abelian subgroup.

Many aspects of quantum error-correcting codes can be described under the guise of its classical counterpart. A quantum code is described efficiently by the structure of the stabilizer formalism, and it is through this description that the process of encoding binary quantum codes is revealed [19, 21]. We give the encoding process for the binary form of quantum codes. We generalise the encoding process to qu**d**it codes in a later section.

The connection to a classical perspective is achieved through the language

of binary vector spaces. Associated with a quantum stabilizer code is a set of generating elements $\mathcal{M}_\alpha,\ \alpha = 1, \ldots, n-k$, belonging to $\mathcal{G}^n/\langle \iota I \rangle$. In a manner similar to 4.1.1, make the correspondence between $\mathcal{G}^n/\langle \iota I \rangle$ and $\mathbb{F}_2^{2n}$ as $\mathcal{M} = (\otimes_{z=1}^n X_{i_z} \cdot \otimes_{z=1}^n Z_{i_z}) \equiv (i_1, i_2, \ldots, i_n | j_1, j_2, \cdots, j_n) = (i|j)$ where i_z is given by

$$i_z = \begin{cases} 1 & \text{if} \quad X_z = X \text{ or } Y \\ 0 & \text{if} \quad X_z = I \text{ or } Z \end{cases} \tag{4.5.1}$$

and j_z by

$$j_z = \begin{cases} 1 & \text{if} \quad Z_z = Z \text{ or } Y \\ 0 & \text{if} \quad Z_z = I \text{ or } X. \end{cases} \tag{4.5.2}$$

The product of the stabilizers corresponding to $(i|j)$ and $(k|l)$ is the stabilizer corresponding to $(-1)^{\sum_{z=0}^{n} j_z k_z}(i+k|j+l)$. The conjugate of the stabilizer corresponding to $(k|l)$ by the stabilizer corresponding to $(i|j)$ is the stabilizer corresponding to $(-1)^{\sum_{z=0}^{n} i_z l_z + j_z k_z}(k|l)$. To encode the stabilizer code we form an X-matrix of stabilizers as the $n \times (n-k)$ matrix X^0 with elements given by $X_{z\alpha} = i_z$ where $\mathcal{M}_\alpha = (i_1, i_2, \ldots, i_n | j_1, j_2, \ldots, j_n)_\alpha$ for all $\alpha = 1, \ldots, n-k$ and $z = 1, \ldots, n$. In a similar fashion we construct the Z-matrix, Z^0 whose columns represent the phase space of the generators. This pair of $n \times (n-k)$ matrices completely determine the quantum stabilizer code and initiates the encoding process. Now sometimes a matrix G can, after elementary row operations, be put in the form $G = (I_m|J)$, where I_m is the $m \times m$ identity matrix and J is an $m \times (n-m)$ matrix. If this can be done, then we say that the generating matrix can be put in standard form. The encoding of the stabilizer code is revealed by transforming the stabilizer to standard form. Under vector space properties, a stabilizer code with a corresponding vector space representation remains invariant to a corresponding stabilizer code with a vector space in standard

form. Let us denote $X_{z\alpha}$ and $Z_{z\alpha}$ to be the initial matrices that represent the stabilizer code. Implementing Gaussian operations on the augmented matrix $(X_{z\alpha}|Z_{z\alpha})$, we obtain $(X^1|Z^1)$ of the following form

$$X^1 = \begin{array}{c} \\ n-r\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{n-k-r} \quad \overbrace{}^{r} \\ \begin{pmatrix} 0 & A \\ 0 & I \end{pmatrix} \end{array} \quad Z^1 = \begin{array}{c} \\ n-r\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{n-k-r} \quad \overbrace{}^{r} \\ \begin{pmatrix} B & C \\ D & E \end{pmatrix} \end{array}$$

where r is the rank of $X_{z\alpha}$. Those columns of the X^1 matrix that form a linearly independent set are described as primary stabilizers while the remaining null vectors are called secondary stabilizers [19]. Z^1 has no particular form, however, we can perform Gaussian elimination on the first $n-r$ rows and the $n-k-r$ columns to transform B, the $(n-r)\times(n-k-r)$ submatrix of Z^1, into B' of type

$$B' = \begin{array}{c} \\ k\{ \\ s\{ \\ n-k-r-s\{ \end{array} \begin{array}{c} \overbrace{}^{s} \quad \overbrace{}^{n-k-r-s} \\ \begin{pmatrix} 0 & B_1 \\ 0 & B_2 \\ 0 & I \end{pmatrix} \end{array}$$

The resulting forms of the resulting augmented matrix $(X^2|Z^2)$ matrices is given by

$$X^2 = \begin{array}{c} \\ k\{ \\ s\{ \\ n-k-r-s\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{s} \quad \overbrace{}^{n-k-r-s} \quad \overbrace{}^{r} \\ \begin{pmatrix} 0 & 0 & A_1 \\ 0 & 0 & A_2 \\ 0 & 0 & A_3 \\ 0 & 0 & I \end{pmatrix} \end{array}$$

and

$$Z^2 = \begin{array}{c} \\ k\{ \\ s\{ \\ n-k-r-s\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{s} \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \overbrace{}^{n-k-r-s} \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \overbrace{}^{r} \\ \\ \\ \\ \\ \end{array} \begin{pmatrix} 0 & B_1 & C_1 \\ 0 & B_2 & C_2 \\ 0 & I & C_3 \\ D_1 & D_2 & E \end{pmatrix}$$

The last r columns of both matrices correspond to the linearly independent set of primary vectors with the preceeding columns representing the secondary vectors. For a positive valued r, commutativity of each pair of stabilizers, one corresponding to one of the first s columns and the other corresponding to one of the last r columns is ensured if D_1 vanishes but may not necessarily follow otherwise. To specify the codewords, we construct a basis for the code by amalgamating the standard form of the stabilizer with k pairs of seed vectors $\bar{X}$ and $\bar{Z}$. The stabilizers corresponding to seed vectors commute with all elements of the stabilizer and with each other except for the case when $\bar{X}_i = \bar{Z}_j$ in which case they anticommute. $\bar{X}$ and $\bar{Z}$ will satisfy this requirement if we define the matrices of the seed vectors as

$$\bar{X} = \begin{array}{c} \\ k\{ \\ s\{ \\ n-r-k-s\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{k} \\ \\ \\ \\ \\ \end{array} \begin{pmatrix} I \\ 0 \\ B_1^\top \\ 0 \end{pmatrix} \quad \bar{Z} = \begin{array}{c} \\ k\{ \\ s\{ \\ n-k-r-s\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{k} \\ \\ \\ \\ \\ \end{array} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The set of columns of $\bar{X}$ are independent of the primary vectors. The stabilizers corresponding to the seed vectors commute with the stabilizers corresponding to the secondary vectors. This is because the relation $\sum_{z=0}^{n-1} i_z l_z + j_z k_z = 0$ for $(i|j)$ corresponding to a seed vector and $(k|l)$ corresponding to a secondary vectors. In particular, for $j = 0$ we have it that the $\sum_{z=0}^{n-1} i_z l_z$ vanishes since

matrix product $\sum (I.B_1 + B_1.I) = 0$. The basis set for a quantum stabilizer code is then written as

$$X^* = \begin{array}{c} \\ k\{ \\ s\{ \\ n-k-r-s\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{k} \quad \overbrace{}^{s} \quad \overbrace{}^{n-k-r-s} \quad \overbrace{}^{r} \\ \begin{pmatrix} I & 0 & 0 & A_1 \\ 0 & 0 & 0 & A_2 \\ B_1^\top & 0 & 0 & A_3 \\ 0 & 0 & 0 & I \end{pmatrix} \end{array}$$

and

$$Z^* = \begin{array}{c} \\ k\{ \\ s\{ \\ n-k-r-s\{ \\ r\{ \end{array} \begin{array}{c} \overbrace{}^{k} \quad \overbrace{}^{s} \quad \overbrace{}^{n-k-r-s} \quad \overbrace{}^{r} \\ \begin{pmatrix} 0 & 0 & B_1 & C_1 \\ 0 & 0 & B_2 & C_2 \\ 0 & 0 & I & C_3 \\ 0 & 0 & D_2 & E \end{pmatrix} \end{array}$$

We utilise these representations in the next section to construct a network encoder for stabilizer codes.

4.6 Network for Encoding

The construction of an encoding network to transform quantum basis states into corresponding codewords is revealed on the evaluation of the stabilizer standard form coupled with a set of constructed seed operators. The encoding of each k logical qubit state is written

$$|\overline{c_1 c_2 \dots c_k}\rangle \;\; \mapsto \;\; \overline{X}_1^{c_1} \dots \overline{X}_k^{c_k} \frac{1}{|\mathcal{S}|} \left(\sum_{\mathcal{M} \in \mathcal{S}} \mathcal{M} \right) |\overbrace{0 \dots 0}^{n}\rangle. \qquad (4.6.1)$$

Let us consider the $[\![8, 3, 5]\!]$ qubit code [12, 31] with a stabilizer description given by

$$\begin{aligned} M_1 &= X \otimes X \otimes X \otimes X \otimes X \otimes X \otimes X \otimes X \\ M_2 &= Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z \\ M_3 &= X \otimes I \otimes X \otimes I \otimes Z \otimes Y \otimes Y \otimes Y \\ M_4 &= X \otimes I \otimes Y \otimes Z \otimes X \otimes I \otimes Y \otimes Z \\ M_5 &= X \otimes Z \otimes I \otimes Y \otimes I \otimes Y \otimes X \otimes Z. \end{aligned} \tag{4.6.2}$$

The standard form associated with this set of operators and the seed operators is derived as

$$X_M^* = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad Z_M^* = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

To encode this formalism, let us first consider a simplified version of the standard form in which we set Z_M^* to vanish, and thus ignoring all phase shifts on the code. Consequently, the matrix array X_M^* is then given by the network in Figure 4.1. While the encoding network implements bit-flip operations that correspond to non-zero entries of X_M^*, it also illustrates the importance of how the standard form contributes to a clear exposition and an efficient computation. Let us now consider the action of phase shifts as seen by the matrix array Z_M^* on the encoding process. The action of a phase flip on a quantum states serves only to alter the phase of that particular state. Since the phase flip does not flip the individual state of the system, it follows that we can insert the set of phase flips which corresponding to those non-zero elements of Z_M^* into our

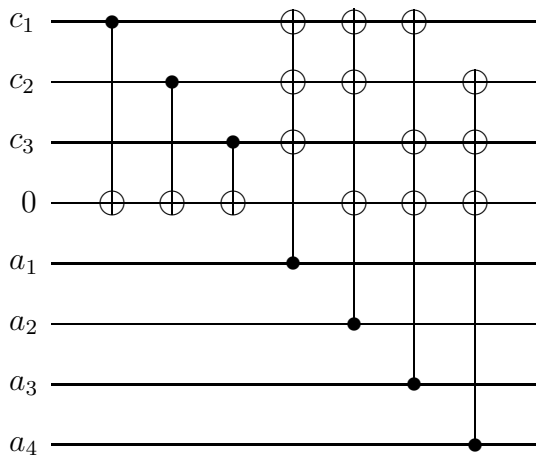


Figure 4.1: Encoding network for X_M^*.

encoding network where $Y = X \cdot Z$. Finally, a complete network encoding is achieved by implementing a Hadamard transform, H, on the last $n-k$ qubits with $R = H \cdot Z$. A Hadamard transform of the last $n-k$ qubits ensures that the input states on the first k qudits, $|c_1c_2c_3\rangle$, only determined the set of codewords. Were this not the case then the last $n-k$ inputs to the encoder would influence the construction of the codewords since each of the last $n-k$ qubits in the state $|0\rangle$ would be flipped to the state $|1\rangle$. This would mean that those qubits now flipped to $|1\rangle$ would help describe the codewords. Figure 4.2 illustrates the encoding circuit up to $X_M^*Z_M^*$. The complete encoding circuit is given in Figure 4.3.

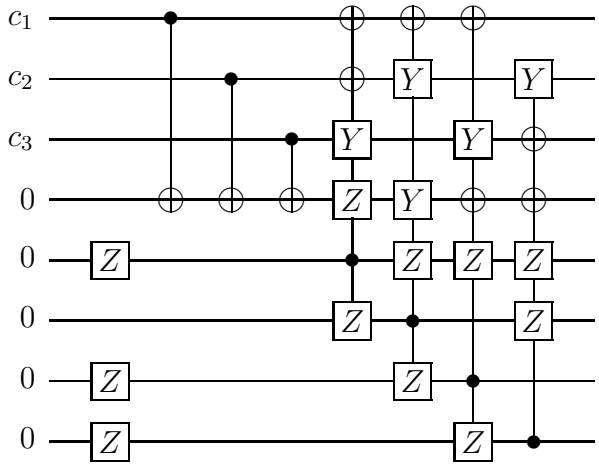


Figure 4.2: Encoding network for $X_M^* Z_M^*$.

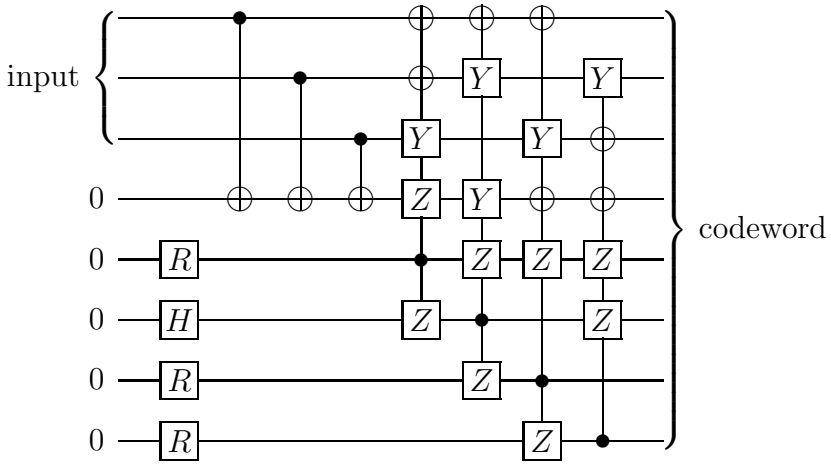


Figure 4.3: An Encoding Network for the $[[8, 3, 5]]$ qubit code.

Chapter 5

Quantum Computation

5.1 Introduction

Introduced in his seminal paper, David Deutsch [24] proposed a new theory of computation; the quantum theory of computation. Deutsch's proposal copper-fastened arguments given by Richard Feynman [29] that promised algorithms to render many intractable classical problems feasible. By defining a computation predicated on exploiting inherently quantum phenomena, Deutsch offered insights into the nature of quantum mechanics and provided a basis for the realisation of a quantum computer. To realise a quantum computer one must be concerned with the search for and implementation of quantum algorithms. A quantum algorithm actuates quantum effects to perform computational tasks with exponential speedup over its respective classical counterparts. Increased theoretical and experimental research in the area of quantum computing is providing such a basis for the discovery of further insights into the nature of quantum algorithms. However, incidents of truely remarkable quantum computing insights such as Shor's factorisation algorithm are few in number. Shor's quantum algorithm was the first example of the exponential speedup in resources over classical counterparts and that more examples of such insights

largely remain outside the confines of present knowledge stands in testament to the understanding required to develop and exploit the theory of quantum computing.

A quantum computer is in essence the realisation of quantum algorithms which in many aspects is analogous to a classical computer with qu**d**its replacing the classical bits as the conduit of information and unitary quantum logic gates in respect of Boolean logic gates acting as operational elements. A quantum logic gate is a transformation operator which is applied to a qu**d**it state in an equivalent fashion as a classical gate would be applied to a classical bit. Furthermore, and in contrast to the classical case, quantum logic gates must ensure that the superposition state of a qu**d**it be maintained. This feature of quantum logic gates implies that the normalisation condition for qu**d**it states be preserved after the application of the operator.

The requirement of normalisation of transformed qu**d**it states necessitates the further precondition of unitarity on quantum logic gates. Unitarity is key to setting the quantum and classical realms apart since transformed quantum states must maintain *coherence*, or isolation from the environment. By the linearity of quantum mechanics, it was shown (2.2.2) that unitarity of quantum operators can only be preserved within a quantum system if the state space of a qu**d**it is isolated from the environment. Moreover, for the reason that quantum logic gates are necessarily unitary transformations, quantum gates are therefore said to exhibit a *reversible* nature. Herein lies a distinguishing feature of quantum computation since the reversible logic elements exhibited in models of quantum computing do not lead to the irretrievable loss of information associated with classical irreversible logic.

Since the theory of quantum computation is the theory of computation then all classical irreversible operations must be resolved within the quantum

mechanical setting. To motivate an explanation for the nature of the exponential speedup in algorithm operation offered by quantum algorithms, we first ought to consider at the picture of classical irreversibility. A reversible computation evaluates invertible functions $f : \{0,1\}^n \to \{0,1\}^n$. Quantum circuitry is fundamentally predisposed to this type of reversible computation since any such computation can be reversed ensuring the existence of a suitable unitary matrix for which f represents a legitimate quantum gate. In contrast, should one consider the irreversible function $f : \{0,1\}^n \to \{0,1\}^m$, $n > m$. Associated with f is a gate with an m-bit range and n-bit domain. Each of the m outputs can be characterised by one of the 2^{2^n} descriptions for a binary function $f : \{0,1\}^n \mapsto \{0,1\}$. Therefore, there are $(2^{2^n})^m$ $n \times m$ gates that describe f of which only $2^n!$ are reversible. The function f typifies many classical irreversible gates such as AND, XOR, and NAND. Bennett [6] has shown that any classical computation can be made reversible and therefore does not lead to loss of entropy. Consider the function $\tilde{f}$ given by $\tilde{f} : \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ where $\tilde{f}(x; 0^{(m)}) = (x; f(x))$ and $0^{(m)}$ denotes the setting of m-bits to zero. Since $\tilde{f}$ takes the inputs $(x; 0^{(m)})$ to a distinct output then $\tilde{f}$ can be extended to an invertible function on $n + m$ bits. Thus, for any irreversible function $f : \{0,1\}^n \to \{0,1\}^m$ there is an invertible $\tilde{f}$ which evaluates f. Understanding how classical irreversibility can be made reversible in a quantum setting may explain the delicate balance between loss of coherence, *decoherence*, in the system and controlled computation of quantum algorithms.

1The set of functions f that describe arbitrary classical operations is associated with a corresponding set of logic gates. Furthermore, a set of gates is called *universal* if instances of it are the only components required to replicate the actions of all logic gates [26]. The quantum mechanical analogue of the classical universal set of logic gates is a set of unitary operators that

generate the action of all unitary transformations in the Hilbert space. It was shown that the 2-qubit controlled-NOT gate and the single rotational qubit gate describes a universal set for quantum computation [3, 26]. Unitary transformations act on quantum states and effect particular changes over a period of time. A quantum state is therefore said to *evolve* under such transformations and correspondingly a *time evolution* operator acting on a quantum state can always given by a unitary transformation. Suppose some quantum state is given by $|\psi\rangle$ and further suppose there exists some unitary transformation U acting on the state. If the state of the system is $|\psi(t_0)\rangle$ at some initial time t_0 then its relation to the evolved state $|\psi(t_1)\rangle$ at time t_1 is given by $|\psi(t_1)\rangle = U(t_1)\,|\psi(t_0)\rangle$. Since U is an arbitrary operator, it is dependent on the system and not on the initial state $|\psi(t_0)\rangle$. The dynamics of U are described by the action of some Hamiltonian $\mathbf{H}$ in Schrödinger's equation [64]. To determine the dynamics of U, let us first consider a small increment in time on $|\psi(t)\rangle$ under U. Hence, we have $|\psi(t+\triangle t)\rangle = U(t+\triangle t)\,|\psi(t)\rangle$. Taking a Taylor series expansion of $U(t+\triangle t)$ and coupled with the assumption of $\mathbf{H}$, it follows that $U(t+\triangle t) = 1 - \iota/\hbar\mathbf{H}\triangle t$ where $\iota^2 = -1$. The action of $U(t+\triangle t)$ on the state $|\psi(t)\rangle$ is written $|\psi(t+\triangle t)\rangle = (1 - \iota/\hbar\mathbf{H}\triangle t)\,|\psi(t)\rangle$. Hence,

$$\frac{|\psi(t+\triangle t)\rangle - |\psi(t)\rangle}{\triangle t} \;=\; -\frac{\iota}{\hbar}\mathbf{H}\,|\psi(t)\rangle\,, \tag{5.1.1}$$

whence,

$$\iota\hbar\frac{\partial\,|\psi(t)\rangle}{\partial t} \;=\; \mathbf{H}\,|\psi(t)\rangle\,. \tag{5.1.2}$$

To determine the nature of the unitary transformation describing the time evolved state $|\psi(t_1)\rangle$, we consider the solution proposed by the Schrödinger equation (5.1.2). Since $\mathbf{H}$ is a fixed Hermitian operator, we can deduce its

spectral decomposition. In particular, $\mathbf{H}$ has an orthonormal set of eigenvectors $|i\rangle$ corresponding with real eigenvalues λ_i such that $\mathbf{H}\,|i\rangle = \lambda_i\,|i\rangle$. For an arbitrary state $|\psi(t)\rangle = \sum_{i=0}^{\mathbf{d}-1} \alpha_i(t)\,|i\rangle$, the time evolution of this arbitrary state is described by the Schrödinger equation

$$\begin{aligned} \iota\hbar\frac{\partial(\sum_{i=0}^{\mathbf{d}-1} \alpha_i(t)\,|i\rangle)}{\partial t} &= \mathbf{H}\sum_{i=0}^{\mathbf{d}-1} \alpha_i(t)\,|i\rangle \\ &= \sum_{i=0}^{\mathbf{d}-1} \alpha_i(t)\lambda_i\,|i\rangle \end{aligned} \tag{5.1.3}$$

and has solution

$$\sum_{i=0}^{\mathbf{d}-1} \alpha_i(t)\,|i\rangle = \sum_{i=0}^{\mathbf{d}-1} e^{-\iota/\ \lambda_i t_1}\alpha_i(t_0)\,|i\rangle \tag{5.1.4}$$

Hence, for a time evolved state, $t = t_1$, we have

$$\begin{aligned} |\psi(t_1)\rangle &= \sum_{i=0}^{\mathbf{d}-1} e^{-\iota/\ \lambda_i t_1}\alpha_i(t_0)\,|i\rangle \\ &= \begin{pmatrix} e^{-i/h\lambda_0 t_1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & & 0 \\ 0 & \dots & 0 & e^{-\iota/\ \lambda_{d-1} t_1} \end{pmatrix} \sum_{i=0}^{\mathbf{d}-1} \alpha_i(t_0)\,|i\rangle \\ &= U(t_1)\,|\psi(t_0)\rangle\,. \end{aligned} \tag{5.1.5}$$

The matrix $U(t_1) = e^{-\iota/\ \mathbf{H}t_1}$ maintains that the Hamiltonian $\mathbf{H}$ is a generator of the unitary transformation $U(t_1)$. This is because the quantum mechanical Hamiltonian operater $\mathbf{H}$ in the description of $U(t_1)$ corresponds to the total energy of the system. From classical mechanics it follows that the energy of the Hamiltonian operator generates the time evolution. Thus, any time evolution on a quantum state in the Hilbert space is the result of applying the Hamiltonian of the unitary transformation.

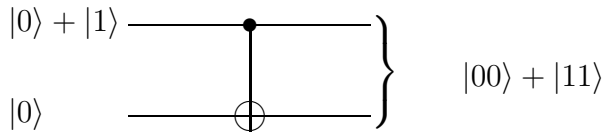


Figure 5.1: Controlled-NOT produces entangled states.

5.2 Multiple Quantum Gates

A universal quantum computer is a well-defined set of unitary gates that can be designed to order the computation of quantum algorithms. Of crucial importance for the successful performance of computations is this well-defined set of gates. Thus far, we have considered the action of single quantum gates on states. This description alone represents an incomplete model of computation. The crux of successful quantum computation is the implementation of multiple quantum gates. The most elementary of multiple quantum gates is to consider some unitary operator U within a controlled-U two qubit operation. The corresponding transformation given by transformation is written as $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$ where the I operation represents the identity transformation. This controlled two qubit operator is so called since the application of U on the second qubit is decided by the state of the first qubit. The classic controlled-U gate is the controlled-NOT gate. The action of a controlled-NOT gate with respect to the computational basis is given as $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus x\rangle$ where $\oplus$ represents addition modulo 2.

The controlled-NOT gate plays an important role in quantum computation [27]. It is the quantum mechanical analogue of the classical connective XOR gate and is a principle component for universal computations. It can be used to produce maximally entangled states similar to the set of EPR pairs [60].

Furthermore, the controlled-NOT gate acts as a measurement gate [25] and provides a basis for a so-called *nondemolition* measurement [22] that permits the constuction of a syndrome table as used in error detection and correction.

5.3 Quantum Circuits

The quantum network approach to computation resembles the classical procedure to computing [92] where quantum circuits are formed from a composition of quantum states, quantum gates and quantum wires [61]. Computations are described within the Hilbert space $\mathcal{H} = (\mathbb{C}^{\mathbf{d}})^{\otimes n}$ of n qu**d**its where each horizontal quantum wire corresponds to the individual $\mathbb{C}^{\mathbf{d}}$ subspaces. Fundamental to computation is the ability to store and control quantum information and thus necessitates a suitable state space description to represent the information vector. Computations are then a finite sequence of time-evolution quantum gates set along the quantum wires to effect suitable transformations. Vertical wires in a quantum circuit represent the *coupling* of arbitrary pairs of quantum gates in a manner similar to a controlled-U gate. The *depth* of a circuit refers to the maximum number of time evolution operators required to effect necessary state changes. The *width* of a circuit is the maximum number of gates in operation in any one time frame, and the *complexity* associated with implementing a quantum circuit relates to the number of the gates needed to solve the task. The task of constructing quantum circuitry architectures is largely the problem of developing algorithms to describe various quantum processes. Many theoretical approaches to quantum code design can now be illustrated within a physical network. Computation of some inherently quantum phenomenon by a quantum computer rests with the ability to find the Hamiltonian for the Schrödinger equation which describes the dynamics of the

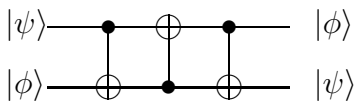


Figure 5.2: Quantum circuit swapping two qubits.

phenomenon. This challenge degenerates to providing solutions to an exponential number of differential equations that describe the Hamiltonian. The growth of computations required for modest increases in the number of qu**d**its describing some quantum operation makes the classical computation of the quantum system infeasible.

The crux of design and construction of a quantum computer operating in a $\mathbf{d}^n$-dimensional Hilbert space lies with the suitable specification of quantum gates that encode the support of a quantum code or implement quantum algorithms. To realise such applications on a quantum computer requires corresponding gate circuitries that manipulate quantum states on both individual and on controlled qu**d**its. The controlled-NOT gate on a pair of qubits has been experimentally realised by a string of ions in a linear Pauli trap [11]. The SWAP gate is an integral feature of the circuitry design of the quantum Fourier transform. Figure 5.2 illustrates a quantum circuit swapping two qubit by the repeated use of quantum controlled-NOT operators. Thus, the question of designing a generalised quantum SWAP gate to implement a permutation of quantum qu**d**it states that utilize the generalised controlled-NOT gate is a concern of universality in quantum computation in higher dimensional quantum systems. How such designs can be extended within a mathematical framework beyond the binary setting is of interest.

The concern of computation is to process information through the use a well defined model that can be both understood algorithmically and realised

by some network topology. This is a concern encountered in both the classical and quantum forms of computation where it is desirable that classes of computation be implemented efficiently thereby requiring polynomial resources to complete a task. While Shannon [74] demonstrated that most Boolean functions require network topologies of exponential resources to compute, it was Shor's factorisation algorithm [79] that introduced the first model of quantum computation to successfully exploit the laws of quantum mechanics and process information efficiently. Such a model of computation now offered credence to years of speculative potential in the area by explicitly illustrating a form that provided an exponential speedup in resources over the best known classical form. It also signalled a marked increase in research into the means of encoding classical information within quantum mechanical systems thereby revealing aspects of what we now call a quantum computer.

Most often it is assumed that a quantum computer is predicated on a collection of two-level quantum mechanical systems called qubits. However, there has been the view to generalise to **d**-level, or qu**d**it, quantum mechanical systems. Implementing a quantum computer within a qubit or qu**d**it framework requires the ability to prepare a quantum state in a chosen basis, most often maintained to be the computational basis, to perform a desired operation on some input state, and finally, to read the outcome of an operation on an input state. While our knowledge of classical computation has influenced the approach we take to implementing quantum computations, the task of successfully implementing a quantum operation is made extremely difficult by the phenomena of quantum parallelism and quantum entanglement. Fortunately, the simultaneous interactions of quantum states expressed during computation can be controlled to a degree by considering quantum operations as a composition of simpler operations on fewer quantum states. These

constituent operations are called quantum gates and surprisingly the only restriction quantum mechanics places on these gates is that they be unitary.

The influence classical computation renders on quantum computation is again evident by mode of algorithm execution. The quantum circuit model [24, 25] reflects the classical circuit and executes quantum algorithms by applying a sequence of quantum gates incrementally to act on one or more quantum states. Those gates that have been experimentally demonstrated are said to be elements of the quantum gate library. Unfortunately, there are only a handful of quantum gates that can be experimentally realised within the coherence time of their systems [90]. However, Barenco *et al.* [3] showed that any quantum operation on a set of n-qubits can be restricted to a composition of Controlled-NOT, CNOT, and single qubit gates. For this reason, we say that the qubit gate library consisting of single qubit gates and CNOT is universal. Furthermore, it has become standard in quantum information to express any n-qubit quantum operation as a composition of single qubit gates and CNOT gates. Consequently, the CNOT gate has acquired special status as the hallmark of multiqubit control [91].

It is becoming increasingly evident that much effort continues to be made into finding efficient quantum circuits in the sense that for the given gate library there is no smaller circuit that achieves the same task. A reason for this concerted effort is primarily due to the principle of decoherence. While a quantum computer is predicated on the undisturbed evolution of quantum coherences, decoherence represents a major but unavoidable problem for the practical realization of quantum computers as it describes the error state of a quantum computer which is introduced through the interaction of system and environment. As with classical computation, the properties of experimentally realisable quantum gates influence the execution time of quantum algorithms.

Since decoherence is unavoidable, it is therefore advantageous to ensure a minimal use of computational resources to limit excution time and exposure to the quantum environment. Thus, a minimal use of computational resources serves to cap the total decohering time delivered by the execution of quantum gates.

While universality is a key concern, the minimisation of gate counts is a recent concern. To this end, research groups now focus on searching for universal n-qubit gates that contain fewest uses of CNOT gates [7, 8, 76, 77]. This is because the cost of experimentally realising a CNOT gate exceeds the cost associated by single qubit gates [8]. Consequently, construction of quantum circuits that minimise the use of CNOT are important from the point of view of execution time of the corresponding quantum algorithm which is positively correlated with the decohering time of the circuit. Indeed, the experimental realisation of a CNOT gate is a covented goal among quantum information groups [91]. Experimental difficulties in realising certain quantum gates has meant that universality of general n-qubit operations is a key motivation of many quantum information groups. This is because both CNOT and single qubit gates operations have now been experimentally realised. Universal quantum circuits for an arbitrary two-qubit operator with six, four and three CNOT gates have been found [78, 90, 91]. In fact, characterising the exact CNOT complexity of an arbitrary n-qubit operation and constructing the corresponding efficient quantum circuit is seen as an ambitious task, even by numerial analysis [91]. Research into universal circuit constructions have done considerable work optimising their constructions [59]. In particular, Vatan and Williams construct a quantum circuit for a general two-qubit operation that requires at most three CNOT gates and fifteen one-qubit gates and show that their construction is optimal [90]. However, crucial to this result is the demand that

quantum circuit for the two-qubit SWAP gate require at least three CNOT gates [90]. To show the correctness of this claim, Vatan and Williams utilize the notion of entangling power as introduced by Zanardi et. al [97]. More recently, a scheme to realise the quantum SWAP gate between flying and stationary qubits has been presented by Liang and Li [53]. It is maintained [53] that experimentally realising the quantum SWAP gate is a necessary condition for the networkability of quantum computation. In fact, the SWAP gate can be used to store quantum information, to teleport atomic or ionic states [53] and is a fundamental element in the circuit implementation of Shor's algorithm [30].

Although universal two-qubit circuits with fewest uses of CNOT gates are known, the three-qubit Toffoli gate is known to require at most six CNOT gates [63], however, only five CNOT gates have been shown to be necessary [91]; see also the references therein. More generally, devising quantum gate constructions that extend to n-qubits remains an open problem. It is believed that the study of quantum circuit minimisation and generalisation of qubit circuit architectures to qu**d**it circuit architectures will require different construction techniques than those presently known. A geometric approach to quantum circuit minimalisation has been put forward by Nielsen [59] that seeks to find the length of a minimal geodesic with respect to a suitable Finsler metric. A striking feature of this approach is that once an initial position and velocity of the geodesic are determined the remainder of the geodesic can be completely evaluated by a second order differential equation. This is in contrast with the usual case of classical and quantum circuitry design where part of a circuit does not aid the complete design. However, it is acknowledged the two-qubit circuit synthesis can be used in a peephole optimization of larger circuits [78].

The problem of constructing an efficient quantum circuit for a given n-qubit operation is seen as an important task for the complete networkability of quantum computation. Unfortunately, the efficiency of universal designs that accomplish this task is often known for the worst case of n-qubit operators. For example, the asymptotic number of CNOT gates used by Barenco's decomposition to implement any n-qubit operation is $\mathrm{O}(n^3 4^n)$ [76]. On the other hand, it is believed that there are interesting operations which might require a polynomial number of CNOT gates and that such operations might also have an efficient construction within higher dimensional quantum systems. Indeed, considering new and efficient circuit designs within the qu**d**it setting taken with new approaches to circuitry design ought to be merit in itself. However, in the context of information processing, it is maintained that there are advantages in moving from the qubit paradigm to the qu**d**it paradigm. For instance, since the entropy of a message depends on the alphabet used, it ought to be that and increasing the alphabet size should allow for the construction of better error-correcting codes [36]. It has also been pointed out that a quantum system composed of a pair of three dimensional subsystem shows new features when compared to a two qubit system [36].

For efficient quantum computations, it is important to determine exactly how many uses of the CNOT gate are required to effectuate a particular operation. We have seen that the optimal universal two-qubit circuit proposed by Vatan and Williams [90] necessarily requires that the two-qubit SWAP gate demands three CNOT gates. The SWAP gate has been illustrated to be a cornerstone in the networkability of quantum computation. Therefore, a question remains, can a generalised SWAP gate for higher dimensional quantum system be efficiently constructed from generalised CNOT gates? Should this be the case then the generalised SWAP gate for higher dimensional quantum systems

might be a cornerstone in the networkability of quantum computer based on qu**d**its. To this end, we note the following results.

5.4 On Permutations

Consider the set $\mathrm{N} = \{1, 2, \ldots, n\}$ and let $\sigma : \mathrm{N} \mapsto \mathrm{N}$ be a bijection. We say $\sigma = \begin{bmatrix} 1 & 2 & \ldots & n \\ i_1 & i_2 & \ldots & i_n \end{bmatrix}$, where $i_k \in \mathrm{N}$ is the image of $k \in \mathrm{N}$ under σ, is a *permutation* of the set N.

Let σ and τ be two permutations of N. We define the product $\sigma \cdot \tau$ by $(\sigma \cdot \tau)(i) = \sigma(\tau(i))$, for $i \in \mathrm{N}$, to be the composition of the mapping τ followed by σ. These permutations taken with $(\cdot)$ form a group denoted S_n which is called the *symmetric group* of degree n.

Given the permutation σ and for each $i \in \mathrm{N}$, let us consider the sequence $i, \sigma(i), \sigma^2(i), \ldots$. Since σ is a bijection and N is finite there exist a smallest positive integer $\ell = \ell(i)$ depending on i such that $\sigma^\ell(i) = i$. The *orbit* of i under σ then consists of the elements $i, \sigma(i), \ldots, \sigma^{\ell-1}(i)$. By a *cycle* of σ, we mean the ordered set $(i, \sigma(i), \ldots, \sigma^{\ell-1}(i))$ which sends i into $\sigma(i)$, $\sigma(i)$ into $\sigma^2(i)$,$\ldots$, $\sigma^{\ell-2}(i)$ into $\sigma^{\ell-1}(i)$, and $\sigma^{\ell-1}(i)$ into i and leaves all other elements of N fixed. Such a cycle is called an (ℓ)-cycle. We refer to 2-cycles as *transpositions*. A pair of elements $\{\sigma(i), \sigma(j)\}$ is an *inversion* in a permutation σ if $i < j$ and $\sigma(i) > \sigma(j)$. Any permutation can be written as a product of transpositions. The number of transpositions in any such product is even if and only if the number of inversions is even, and consequently, we say the permutation is even. Similarly, a permutation is odd if it can be written as a product of an odd number of transposition and hence has an odd number of inversions.

Lemma 1. Every permutation can be uniquely expressed as a product of disjoint cycles.

Proof: Let σ be a permutation. Then the cycles of the permutation are of the form $i, \sigma(i), \ldots, \sigma^{\ell-1}(i)$. Since the cycles are disjoint and by the multiplication of cycles, we have it that the image of $i \in N$ under σ is the same as the image under the product, ς, of all the disjoint cycles of σ. Then, σ and ς have the same effect on every element in N, hence, $\sigma = \varsigma$.

Every permutation in S_n has then a *cycle decomposition* that is unique up to ordering of the cycles and up to a cyclic permutation of the elements within each cycle. Further, if $\sigma \in S_n$ and σ is written as the product of disjoint cycles of length $n_1, \ldots, n_k$, with $n_i \leq n_{i+1}$, we say $(n_1, \ldots, n_k)$ is the *cycle type* of σ.

As a result of Lemma 1, every permutation can be written as a product of transpositions. Since the number of transpositions needed to represent a given permutation is either even or odd, we define the *signature* of a permutation as

$$\operatorname{sgn}(\sigma) = \begin{cases} +1 & \text{if } \sigma \text{ is even} \\ -1 & \text{if } \sigma \text{ is odd} \end{cases} \tag{5.4.1}$$

To each permutation, let us associate a permutation matrix A_σ whereby

$$A_\sigma(j, i) = \begin{cases} 1 & \text{if } \sigma(i) = j \\ 0 & \text{otherwise} \end{cases} \tag{5.4.2}$$

The mapping $f : S_n \mapsto \det(A_\sigma)$ is a group homomorphism, where

$$\det(A_\sigma) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{\sigma(i),i} \tag{5.4.3}$$

The kernel of this homomorphism, $\ker f$, is the set of even permutations. Consequently, we have it that σ is even if and only if $\det(A_\sigma)$ equals $+1$. The kernel of the homomorphism signature defines the alternating group. Note that the set of odd permutation can not form a subgroup but they form a coset of the alternating group.

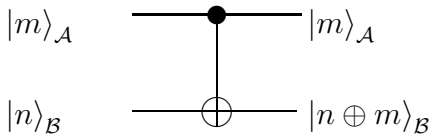
Let us consider the following problem. Given a pair of **d**-dimensional quantum systems, system $\mathcal{A}$ in the state $|\psi\rangle$ and system $\mathcal{B}$ in the state $|\phi\rangle$, determine if it is possible swap the states of the corresponding systems so that system $\mathcal{A}$ is in the state $|\phi\rangle$ and that system $\mathcal{B}$ is in the state $|\psi\rangle$.

5.4.1 On the swap of a pair of Qutrits

Let $\mathcal{H}_{\mathcal{A}}$ and $\mathcal{H}_{\mathcal{B}}$ be two **d**-dimensional Hilbert spaces with bases $|i\rangle_{\mathcal{A}}$ and $|i\rangle_{\mathcal{B}}, I \in \mathbb{Z}_{\mathbf{d}}$ respectively. Let $|\psi\rangle_{A}$ denote a pure state of the quantum system $\mathcal{H}_{\mathcal{A}}$. Similarly, let $|\phi\rangle_{\mathcal{B}}$ denote a pure state of the quantum system $\mathcal{H}_{\mathcal{B}}$ and consider an arbitrary unitary transformation $U \in \mathrm{U}(\mathbf{d}^2)$ acting on $\mathcal{H}_{\mathcal{A}} \otimes \mathcal{H}_{\mathcal{B}}$. Let U_{CNOT1} [90] denote a CNOT gate that has qu**d**it $|\psi\rangle_{\mathcal{A}}$ as the control qu**d**it and $|\phi\rangle_{\mathcal{B}}$ as the target qu**d**it;

$$U_{\mathrm{CNOT1}} |m\rangle_{\mathcal{A}} \otimes |n\rangle_{\mathcal{B}} = |m\rangle_{\mathcal{A}} \otimes |n \oplus m\rangle_{\mathcal{B}}, \qquad m, n \in \mathbb{Z}_{\mathbf{d}} \tag{5.4.4}$$

where $i \oplus j$ denote modulo **d** addition. In gate circuitry notation, the CNOT1 gate is given by



(5.4.5)

Similarly, let U_{CNOT2} [90] denote a CNOT gate that has qu**d**it $|\psi\rangle_{\mathcal{A}}$ as the target qu**d**it and $|\phi\rangle_{\mathcal{B}}$ as the control qu**d**it;

$$U_{\mathrm{CNOT2}} |m\rangle_{\mathcal{A}} \otimes |n\rangle_{\mathcal{B}} = |m \oplus n\rangle_{\mathcal{A}} \otimes |n\rangle_{\mathcal{B}}, \qquad m, n \in \mathbb{Z}_{\mathbf{d}} \tag{5.4.6}$$

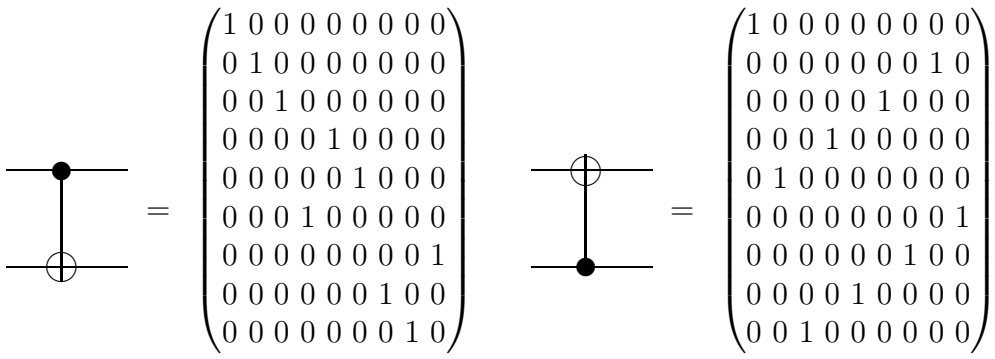
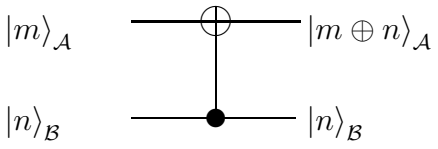


Figure 5.3: Matrix representations of CNOT types.

In gate circuitry notation, the CNOT2 gate is given by



(5.4.7)

Both circuits are read from left to right. Each horizontal line in both circuits is called a wire and represents a finite dimensional quantum system. The evolution of a quantum state over time is described by a set of unitary transformations acting on the state description of the quantum system. Such state evolutions are represented at various locations along the wire.

We now show that a swap of two qutrits is not possible using a composition of CNOT gates alone. The point of this argument is to illustrate that a quantum gate construction which permutes the states of three qutrit systems can not be described by a set of qutrit transpositions induced by the CNOT gate alone. Were this not the case then we would have a much simpler, with repect to CNOT complexity, solution to the problem of construction a generalised SWAP of **d** qu**d**it system.

To show this, we first note that any sequence of CNOT gates acting on the qutrit states $|\psi\rangle_{\mathcal{A}}$ and $|\phi\rangle_{\mathcal{B}}$ can be written as a composition of the gates

CNOT1 and CNOT2. The CNOT1 and CNOT2 gates can be described in the following way; the permutation matrix corresponding to the CNOT1 gate takes the value 1 in row $3m + n$ and column $3m + (m \oplus n)$, $m, n = 0, 1, 2$. Similarly, the matrix corresponding to the CNOT2 gate takes the value 1 in row $3m + n$ column $3(m \ominus n) + n$. These unitary matrix representations for a CNOT gate are given in Fig. 5.3. Furthermore, both the CNOT1 matrix and CNOT2 matrix have determinant +1 since the permutation corresponding to each of the respective matrices is even.

Let us now assume that there exists a gate that swaps a pair of qutrit states and that such a gate is composed using only the CNOT gate. Such a swap gate will then be a composition of the gates CNOT1 and CNOT2. Since each CNOT circuit acting on a pair of qutrits is a composition of CNOT1 and CNOT2, it follows that any such composition will be equivalent to some product of their respective unitary matrices. Such a product matrix product will necessarily have determinant +1 as its constituent elements have determinant +1. However, the matrix transformation representation required to effectuate the swap of a pair of qutrits is given by

$$\begin{pmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \end{pmatrix}. \tag{5.4.8}$$

This matrix takes the value 1 in row $3m+n$ column $3n+m$ and has determinant -1. Thus, no composition of the former can yield the latter and the result follows.

5.5 On the swap of a pair of qudits

Barenco *et al.* [3] showed that any unitary transformation on a set of qubits can be decomposed into a sequence of CNOT and single-qubit gates [91]. We now consider the problem of swapping a pair of **d**-dimensional quantum states using only CNOT gates such that the system $\mathcal{H}_{\mathcal{A}}$ begins in the state $|\psi\rangle_{\mathcal{A}}$ and ends in the state $|\phi\rangle_{\mathcal{A}}$ while correspondingly the system $\mathcal{H}_{\mathcal{B}}$ begins in the state $|\phi\rangle_{\mathcal{B}}$ and ends in the state $|\psi\rangle_{\mathcal{B}}$. Our argument will be that a transposition qu**d**it states induces some unitary matrix $U(\mathbf{d}^2)$ over $\mathcal{H}_{\mathcal{A}} \otimes \mathcal{H}_{\mathcal{B}}$ whose circuit architecture can not be completely determined by using only CNOT gates.

Recall the particular problem concerning the swap of a pair of qutrit systems. We have shown how the unitary matrices U_{CNOT1} and U_{CNOT2} both have determinant +1. We also showed that this is in contrast to matrix U_{SWAP} which describes the swapping of states of a pair of quantum systems where such a matrix has determinant -1. Consequently, no compostion of CNOT gates alone can induce the matrix that determines the action of the SWAP gate. Another way to look at this is the following. The permutations

$$\begin{aligned} \sigma_{\text{CNOT1}} &= \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ 0\ 1\ 2\ 4\ 5\ 3\ 8\ 6\ 7 \end{pmatrix} \\ \sigma_{\text{SWAP}} &= \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ 0\ 5\ 6\ 7\ 4\ 1\ 2\ 3\ 8 \end{pmatrix} \end{aligned} \tag{5.5.1}$$

that correspond to the unitary matrices U_{CNOT1} and U_{SWAP} have corresponding cycle types $(1, 1, 1, 3, 3)$ and $(1, 1, 1, 2, 2, 2)$. Hence, a CNOT gate fixes three basis states and permutes the remaining states in two cycles of length 3. Each such cycle may be written as a product of two transpositions. Whence, the signature of the CNOT permutation is +1. On the other hand, a SWAP gate that swaps the states of a pairs of qutrits contains three fixed elements and a set of three transpositions and therefore the signature of the SWAP

permutation is -1 and it follows that no composition of CNOT gates can lead to an execution of a swap of a pair of qutrit systems.

More generally, a CNOT gate acting on a pair of $\mathbf{d}$-dimensional quantum systems corresponds to a permutation of the $\mathbf{d}^2$ basis states. We consider the case when $\mathbf{d} = p$ is a prime. For prime dimensions $\mathbf{d} = p$ the case of CNOT1, we have it that the basis states $|m\rangle_{\mathcal{A}} \otimes |n\rangle_{\mathcal{B}}$ of the system $\mathcal{H}_{\mathcal{AB}}$ are mapped mapped to $|m\rangle_{\mathcal{A}} \otimes |n \oplus m\rangle_{\mathcal{B}}$. The permutation associated with the CNOT1 mapping fixes $\mathbf{d}$ basis states and has $(\mathbf{d}-1)$ cycles of length $\mathbf{d}$, each of which may be written as a product of $\mathbf{d}$-1 transpositions. CNOT1 yields a permutation that can then be composed of $(\mathbf{d}-1)^2$ transpositions of qu**d**it basis states. Similarly, the CNOT2 gate acting on a pair of qu**d**it basis states maps $|m\rangle_{\mathcal{A}} \otimes |n\rangle_{\mathcal{B}}$ of $\mathcal{H}_{\mathcal{AB}}$ to $|m \oplus n\rangle_{\mathcal{A}} \otimes |n\rangle_{\mathcal{B}}$. There are $\mathbf{d}$ fixed basis elements under the CNOT2 mapping and ($\mathbf{d}$-1) cycles, each a product of $\mathbf{d}$-1 transpositions. Therefore, the signature of the CNOT permutation is -1 for dimension $\mathbf{d} = 2$ and $+1$ for odd prime dimensions. Now suppose a CNOT gate acting on a pair of qu**d**its within system $\mathcal{H}_{\mathbf{d}^{\mathbf{d}}}$. Further suppose that such an action is described by $U_{\text{CNOT}} \otimes I_{\mathbf{d}^{\mathbf{d}-2}}$. This matrix representation induces a permutation of $\mathbf{d}^{(\mathbf{d}-2)}$ copies of the $\mathbf{d}^2$ basis elements targeted by the CNOT gate and it follows that the signature of corresponding permutation is -1 only for dimensions $\mathbf{d} = 2$.

Let us consider a SWAP gate that swaps that states of a pair of qu**d**its. Such a gate corresponds to a permutation of the $\mathbf{d}^2$ basis states of system $\mathcal{H}_{\mathbf{d}^2}$ which maps basis states $|m\rangle_{\mathcal{A}} \otimes |n\rangle_{\mathcal{B}}$ to basis states $|n\rangle_{\mathcal{A}} \otimes |m\rangle_{\mathcal{B}}$. Under this mapping there are $\mathbf{d}$ fixed basis elements and $\mathbf{d}(\mathbf{d}-1)/2$ transpositions which describe the interchanging of all remaining basis states. Thus, the signature of the permutation corresponding to the SWAP gate of a pair of qu**d**its is -1 for dimensions $\mathbf{d} = 2$ or 3 (mod 4) and $+1$ for dimensions $\mathbf{d} = 0$ or 1 (mod

4). Thus when $\mathbf{d} = 3 \pmod 4$ the SWAP cannot be realised the CNOT gates alone.

Further consider a cycle of $\mathbf{d}$ quantum states that maps basis states $|u\rangle_{\mathcal{I}} \otimes |v\rangle_{\mathcal{J}} \otimes |w\rangle_{\mathcal{K}} \cdots \otimes |z\rangle_{\mathcal{M}}$ to the basis states $|z\rangle_{\mathcal{I}} \otimes |u\rangle_{\mathcal{J}} \otimes |v\rangle_{\mathcal{K}} \cdots \otimes |y\rangle_{\mathcal{M}}$. As above the cycle structure of this permutation depends on the factorisation of the dimension of the quantum system. Thus, for prime dimensions, the permutation corresponding to a cycle of $\mathbf{d}$ qu**d**it states contains $\mathbf{d}$ fixed states and $(\mathbf{d}^{\mathbf{d}} - \mathbf{d})/\mathbf{d}$ cycles of length $\mathbf{d}$. Consequently, there are $(\mathbf{d}^{(\mathbf{d}-1)} - 1)(\mathbf{d} - 1)$ transpositions association with the cycle of $\mathbf{d}$ qu**d**it systems. Over even dimension $\mathbf{d}$, the permutation signature of such is -1 and $+1$ for odd dimension $\mathbf{d}$.

The task swapping a pair of qu**d**it states has been argued in terms of the signature of a permutation. Based on this argument, we have shown that a CNOT gate acting on a pair of qu**d**its corresponds to a permutation whose signature is $+1$, for odd prime dimensions. A SWAP of pairs of qu**d**it systems yields a permutation whose signature is -1 for dimensions $\mathbf{d} = 2$ or $3 \pmod 4$ and $+1$ for dimensions $\mathbf{d} = 0$ or $1 \pmod 4$. By this argument alone, circuit architectures completely described by instances of the CNOT gate can not be used to implement a SWAP of a pair of qu**d**its for dimensions $\mathbf{d} = 3 \pmod 4$.

Chapter 6

On the WilNOT Gate

Of central importance to the theory of quantum computation is the role assumed by single or multiple qu**d**it gates that establish a basis for quantum circuitry designs. A quantum circuit is an assembly of discrete sets of components which describe computational procedures [63]. A physical implementation of such designs describes the process of computation whereby the evolution of a qu**d**it state and its influence on other states can be modelled. Quantum computation is therefore a process that identifies the changes imposed on a quantum state during the implementation of quantum gates in a manner analogous to classical implementation of logic gates. Our ability to preserve quantum coherence rests with our ability to implement efficient quantum computations.

Fundamental to successful quantum computations is the application of single and multiple qu**d**it gates. The most important multiple gate in the realm of binary quantum computation is the two qubit controlled NOT gate [27] as illustrated in Figure 5.1. The controlled NOT gate can realise quantum computation networks such as entanglement, teleportation and error correction and is the principle multiple qubit gate in the universal set of gates that describes arbitrary quantum computations [3, 63]. The dynamics of the controlled NOT gate as described by its transformation on pairs of qubits can

be considered as an extension of the single gate NOT operation. In analogy with the binary case, the generalised controlled NOT gate acts on two qu**d**it states. Correspondingly, the generalised NOT gate is crucial to the working of the generalised controlled NOT gate and serves as the crux for popular computations of error model and coding constructions in higher dimensions.

The qu**d**it representation of a quantum state provides a natural mechanism by which quantum computations can be implemented. That such a computation is made possible initially lies with the notion of state signature. In particular, the correspondence of quantum information α_k with a computational qu**d**it basis element $|k\rangle$ and the subsequent genesis of the quantum state $\sum_{k=0}^{\mathbf{d}-1} \alpha_k |k\rangle$ in the Hilbert space $\mathbb{C}^{\mathbf{d}}$. Such a correspondence between information and a Hilbert space representation is prerequisite to quantum computation since the successful transmission of any information state is predicated on encoding the basis states associated with the quantum information elements rather than the information itself which thereby prevents the collapse of the state superposition and loss of information. This is in contrast to the classical setting where no such correspondence is proffered nor required.

6.1 The WilNOT Gate

The WilNOT gate is a generalised quantum SWAP operator, see Figure 6.1. By this I mean, suppose that the first quantum system $\mathcal{A}_0$ prepared in the state $|e_0\rangle_0$, the second system $\mathcal{A}_1$ prepared in the state $|e_1\rangle_1$ and so forth, with the final system $\mathcal{A}_{\mathbf{d}-1}$ prepared in the state $|e_{\mathbf{d}-1}\rangle_{\mathbf{d}-1}$. Construction of the WilNOT gate over prime dimension yields a generalised SWAP gate so that the system $\mathcal{A}_0$ is in the state $|e_1\rangle_0$, the system $\mathcal{A}_1$ is in the state $|e_2\rangle_1$ and

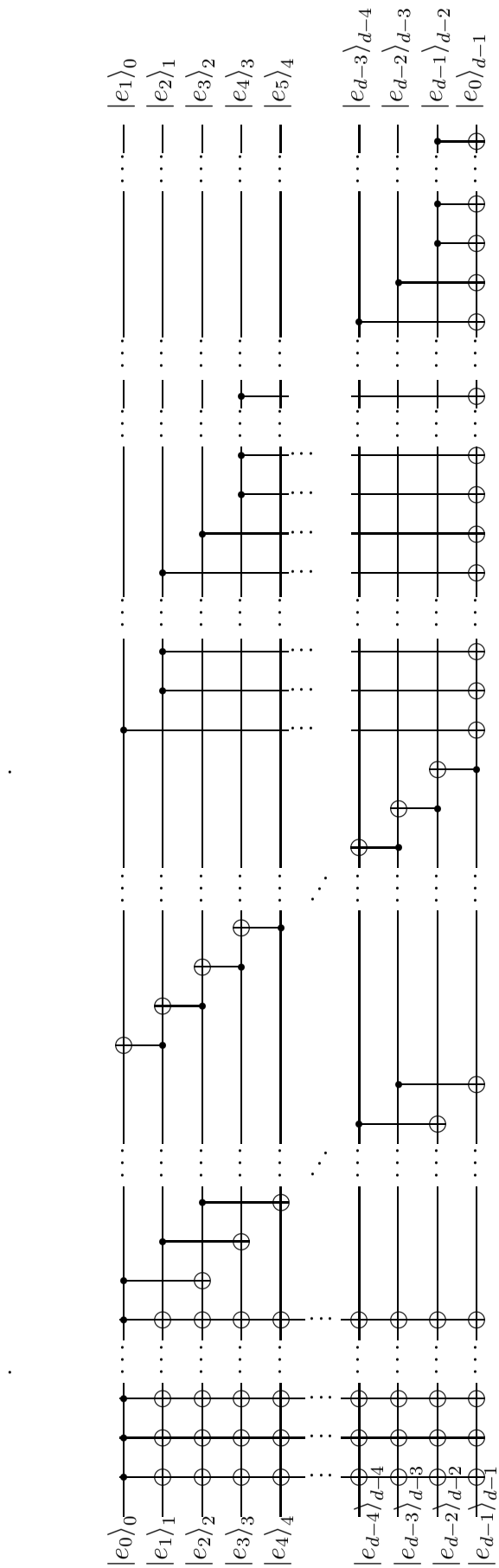


Figure 6.1: The WilNOT Gate; A Generalised SWAP Gate.

so forth, until the system $\mathcal{A}_{\mathbf{d}-1}$ is in the state $|e_0\rangle_{\mathbf{d}-1}$. Central to this implementation is the use of the generalised quantum controlled-NOT operator, $|x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus x \bmod \mathbf{d}\rangle$. It has been shown that the generalised quantum NOT operator is an important primitive in higher dimensional quantum systems. For instance, the generalised quantum NOT operator is analogous the qu**d**it flip, or generalised Pauli X matrix, operator that serves to provide a basis for **d**-dimensional Hilbert spaces. Contrary to the claim that considerations of higher dimensional quantum system would provide negligible innovative theoretical insights [63], one argument in support of such considerations can be seen in the context of information processing [36]. Since the entropy of a message is dependent upon alphabet size, it may be that by increasing the alphabet size allows for better quantum code constructions. Therefore, in a similar manner to the qubit flip operator providing a description of binary quantum codes, the generalised NOT operator may aid a concise representation of qu**d**it codes that engender good coding parameters.

Lemma 2. [72] $\sum_{n=0}^{k} \binom{l+n}{n} = \binom{l+k+1}{k}$.

Theorem 19. Let $\mathbf{d} = p$ be a prime. The WilNOT operator algorithm provides a constuction for generalised qauantum SWAP operator through uses of the generalised quantum controlled-NOT operator. The quantum SWAP operator has

Input: $|e_k\rangle_k ; k = 0, \ldots, \mathbf{d}-1$.

Output: $|e_{k+1}\rangle_k ; k = 0, \ldots, \mathbf{d}-2, |e_0\rangle_{\mathbf{d}-1}$.

The WilNOT operator algorithm is described as follows,

Input: $e_k := i_k^0; k = 0, \ldots, \mathbf{d}-1$

Output: $i_k^{\mathbf{d}+2} = e_{k+1}; k = 0, \ldots, \mathbf{d}-1, i_{\mathbf{d}-1}^{\mathbf{d}+2} = e_0$.

Stage 1. Initialisation: $j = 0$.

$$e_k \;:=\; i_k^0$$

for $k = 0, \ldots, \mathbf{d} - 1$.

The WilNOT gate is initiated by Stage 1 and step $j = 0$ by making the correspondence between a representative input element i_k^0 of the WilNOT gate algorithm each standard basis state e_k.

Stage 2. For $j = 1, \ldots, \mathbf{d} - 1$.

$$\begin{aligned} i_0^j &= i_0^{j-1} \\ i_k^j &= i_{k-1}^j + i_k^{j-1}; k = 1, \ldots \mathbf{d} - 1. \end{aligned}$$

Stage 2 consists of $\mathbf{d} - 1$ steps which repeat the sequence of gates of step $j = 1$. The sequence of gates at step $j = 1$, see Fig. 6.2, is targeted on the systems $\mathcal{A}_1, \ldots, \mathcal{A}_{\mathbf{d}-1}$. Each step of Fig. 6.2 is a composition of CNOT gates acting on consecutive pairs of systems and is written as a shorthand form to represent the sequence of CNOT gates given in Fig. 6.3. The algorithm process of step $j = 1$ transforms the input sequence $i_0^0, i_1^0, i_2^0, \ldots, i_{\mathbf{d}-1}^0$ to the resulting state given by $i_0^0, \sum_{k=0}^1 i_k^0, \sum_{k=0}^2 i_k^0, \ldots, \sum_{k=0}^{\mathbf{d}-1} i_k^0$. In a similar manner, the WilNOT gate at step $j = 2$ takes the output from step $j = 1$ as input and repeats the sequence of gates. The resulting state of the circuit at step $j = 2$ is given by i_0^0, $i_0^0 + \sum_{k=0}^1 i_k^0$, $i_0^0 + \sum_{k=0}^1 i_k^0 + \sum_{k=0}^2 i_k^0$, $\ldots$, $i_0^0 + \sum_{k=0}^1 i_k^0 + \sum_{k=0}^2 i_k^0 + \cdots + \sum_{k=0}^{\mathbf{d}-1} i_k^0$. This process continues to step $j = \mathbf{d} - 1$. Figure 6.4 illustrates initialisation on the circuit and the subsequent $\mathbf{d} - 1$ steps of Stage 2.

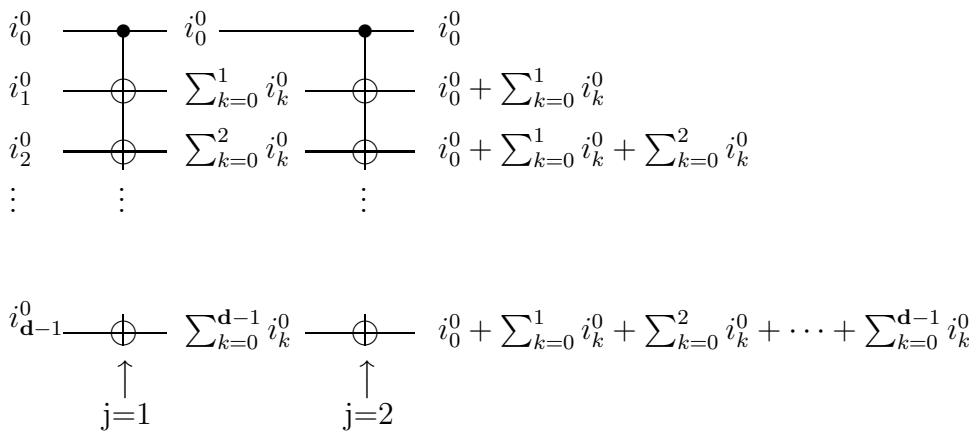


Figure 6.2: WilNOT gate; Stage 2, steps j = 1, 2.

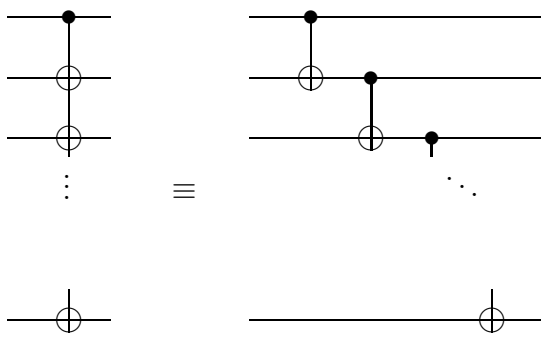


Figure 6.3: WilNOT gate; Stage 2. Algorithm step operates on successive pairs.

Stage 3. $j = \mathbf{d}$.

$$\begin{aligned} i_0^{\mathbf{d}} &= i_0^{\mathbf{d}-1} \\ i_1^{\mathbf{d}} &= i_1^{\mathbf{d}-1} \\ i_k^{\mathbf{d}} &= i_{k-2}^{\mathbf{d}} + i_k^{\mathbf{d}-1}; k = 2, \ldots, \mathbf{d}-1. \end{aligned}$$

The sequence of values $i_0^{\mathbf{d}-1}, i_1^{\mathbf{d}-1}, \ldots, i_{\mathbf{d}-1}^{\mathbf{d}-1}$ corresponding to the final step of Stage 2 are carried foward as an input sequence for Stage 3 and step $j = \mathbf{d}$. The algorithm step keeps the values $i_0^{\mathbf{d}-1}, i_1^{\mathbf{d}-1}$ and returns them as outcomes $i_0^{\mathbf{d}}, i_1^{\mathbf{d}}$ for step $j = \mathbf{d}$. The remaining systems are then

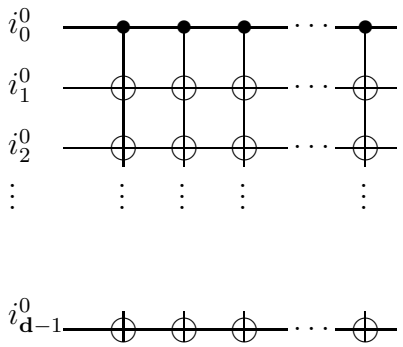


Figure 6.4: WilNOT gate; Stage 2, steps j = 1,. . . , **d**-1.

targeted in an iterative process. For instance, the outcome $i_2^{\mathbf{d}}$ for step $j = \mathbf{d}$ is given by $i_0^{\mathbf{d}} + i_2^{\mathbf{d}-1}$. This value is then stored as the result $i_2^{\mathbf{d}}$ for e_2 at Stage 3. The outcome state for $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ at Stage 3 have thus been determined. To evaluate the result value for $\mathcal{A}_3$, the algorithm computes $i_1^{\mathbf{d}} + i_3^{\mathbf{d}-1}$ and stores this value as the outcome $i_3^{\mathbf{d}}$ for Stage 3. Figure 6.5 illustrates the process that determines the current state of the algorithm following stage 3 and step $j = \mathbf{d}$ in diagrammatic shorthand form for the sequence of CNOTs.

Stage 4. $j = \mathbf{d}$+1.

$$\begin{aligned} i_k^{\mathbf{d}+1} &= i_k^{\mathbf{d}} + i_{k+1}^{\mathbf{d}} \\ i_{\mathbf{d}-1}^{\mathbf{d}+1} &= i_{\mathbf{d}-1}^{\mathbf{d}}; k = 0, \ldots, \mathbf{d}-2. \end{aligned}$$

Stage 4 consists of a single step, $j = \mathbf{d}$+1, whose primary algorithm operation acts as a CNOT on the **d**-1 consecutive pairs of systems $(\mathcal{A}_k, \mathcal{A}_{k+1})$ for $k = 0, \ldots, \mathbf{d}-2$, computing $(i_k^{\mathbf{d}} + i_{k+1}^{\mathbf{d}})$ and storing these values as the outcome $i_k^{\mathbf{d}+1}$. The value $i_{\mathbf{d}-1}^{\mathbf{d}}$ is returned as the outcome $i_{\mathbf{d}-1}^{\mathbf{d}+1}$.

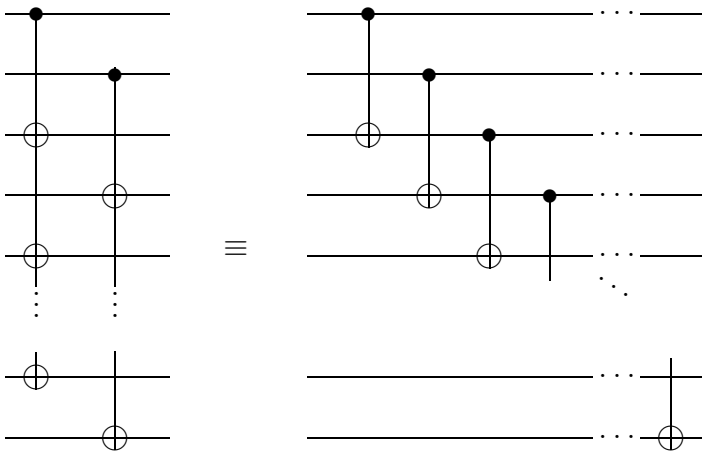


Figure 6.5: WilNOT gate; Stage 3, step $j = \mathbf{d}$.

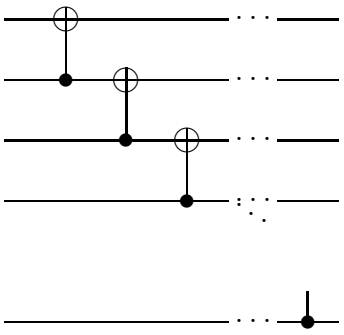


Figure 6.6: WilNOT gate; Stage 4, step $j = \mathbf{d} + 1$.

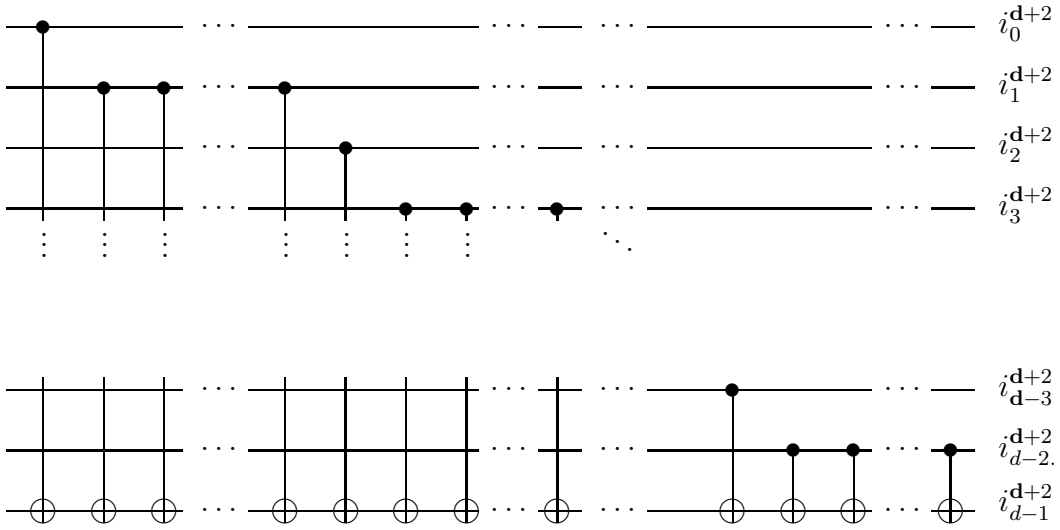


Figure 6.7: WilNOT gate; Stage 5, step $j = \mathbf{d} + 2$.

Stage 5. $j = \mathbf{d}$+2.

$$\begin{aligned} i_k^{\mathbf{d}+2} &= i_k^{\mathbf{d}+1} \\ i_{\mathbf{d}-1}^{\mathbf{d}+2} &= i_{\mathbf{d}-1}^{\mathbf{d}+1} + \sum_{k=0}^{\mathbf{d}-2} \eta_k i_k^{\mathbf{d}+2};\ k = 0, \ldots, \mathbf{d}-2 \end{aligned}$$

with

$$\sum_{k=0}^{\mathbf{d}-2} \eta_k i_k^{\mathbf{d}+2} := \sum_{t=0}^{\lfloor \frac{\mathbf{d}-2}{2} \rfloor} (\mathbf{d}-1) i_{2t+1}^{\mathbf{d}+2} + \sum_{t=0}^{\frac{\mathbf{d}-3}{2}} i_{2t}^{\mathbf{d}+2} \qquad (6.1.1)$$

Stage 5 concludes the WilNOT gate transformation with a set of gates targeted on system $\mathcal{A}_{\mathbf{d}-1}$ whose current state is represented by $i_{\mathbf{d}-1}^{\mathbf{d}+1}$. The values $i_0^{\mathbf{d}+2}, i_1^{\mathbf{d}+2}, \ldots, i_{\mathbf{d}-2}^{\mathbf{d}+2}$ for the respective systems $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{\mathbf{d}-2}$ are unchanged from their representative values $i_0^{\mathbf{d}+1}, i_1^{\mathbf{d}+1}, \ldots, i_{\mathbf{d}-2}^{\mathbf{d}+1}$ at step $j = \mathbf{d} + 1$ and are returned as outcomes in the final state for step $j =$

$\mathbf{d}$+2. The final state of $\mathcal{A}_{\mathbf{d}-1}$ is given by $i_{\mathbf{d}-1}^{\mathbf{d+2}} = i_{\mathbf{d}-1}^{\mathbf{d+1}} + \sum_{k=0}^{\mathbf{d}-2} \eta_k i_k^{\mathbf{d}+2}$ $= i_{\mathbf{d}-1}^{\mathbf{d+1}} + i_0^{\mathbf{d+1}} + (\mathbf{d}-1) i_1^{\mathbf{d+1}} + i_2^{\mathbf{d+1}} + (\mathbf{d}-1) i_3^{\mathbf{d+1}} + \cdots + i_{\mathbf{d}-3}^{\mathbf{d+1}} + (\mathbf{d}-1) i_{\mathbf{d}-2}^{\mathbf{d+1}}$. Thus, for odd valued k there is a gate with $i_k^{\mathbf{d}+2}$ as control and for even valued k there are $\mathbf{d}$-1 gates with $i_k^{\mathbf{d}+2}$ as control. This is represented in Figure 6.7.

Proof: We show that the algorithm outputs $i_k^{\mathbf{d}+2} = e_{k+1}$ for $k = 0, \ldots, \mathbf{d}+2$ and $i_{\mathbf{d}-1}^{\mathbf{d}+2} = e_0$. At step $j = 0$, we have it that,

$$e_k := i_k^0;\ k = 0, ..., \mathbf{d} - 1.$$

At Stage 2, step $j = 1$ the algorithm sets $i_0^1 = i_0^0$ and computes i_1^1 as $i_1^1 = i_0^1 + i_1^0$. Similarly, $i_2^1 = i_1^1 + i_2^0 = i_0^0 + i_1^0 + i_2^0 = \sum_{m=0}^{2} i_m^0$. Therefore, for $k = 1, \ldots, \mathbf{d} - 1$, we have,

$$\begin{aligned} i_k^1 &= i_{k-1}^1 + i_k^0 \\ &= i_{k-2}^1 + i_{k-1}^0 + i_k^0 \\ &= i_{k-3}^1 + i_{k-2}^0 + i_{k-1}^0 + i_k^0 \\ &= \ldots \\ &= i_0^1 + i_1^0 + i_2^0 + \cdots + i_{k-3}^0 + i_{k-2}^0 + i_{k-1}^0 + i_k^0 \\ &= \sum_{m=0}^{k} i_m^0 \end{aligned} \tag{6.1.2}$$

The next step, Stage 2 step $j = 2$, implements a repeat set of gates of step 1. By definition $i_0^2 = i_0^1 = i_0^0$. The case for $k = 1, \ldots, \mathbf{d} - 1$ follows from the algorithm step,

$$\begin{aligned} i_k^2 &= i_{k-1}^2 + i_k^1 \\ &= i_{k-2}^2 + i_{k-1}^1 + i_k^1 \\ &= \ldots \end{aligned}$$

$$
\begin{aligned}
&= i_0^2 + i_1^1 + i_2^1 + \cdots + i_{k-2}^1 + i_{k-1}^1 + i_k^1 \\
&= i_0^0 + \sum_{m=0}^{1} i_m^0 + \sum_{m=0}^{2} i_m^0 + \cdots + \sum_{m=0}^{k} i_m^0 \\
&= \sum_{l=0}^{k} \sum_{m=0}^{l} i_m^0 \\
&= \sum_{m=0}^{k} \sum_{l=m}^{k} i_m^0 \\
&= \sum_{m=0}^{k} \sum_{l=0}^{k-m} i_m^0 \\
&= \sum_{m=0}^{k} \binom{k-m+1}{1} i_m^0 \qquad (6.1.3)
\end{aligned}
$$

We show by induction that, for $j = 1, \ldots, \mathbf{d}-1$, $i_k^j = \sum_{m=0}^{k} \binom{k-m+j-1}{j-1} i_m^0$, $k = 0, \ldots, \mathbf{d}-1$. We have shown that this is true for $j = 1$. Let $1 \leq j < \mathbf{d}-1$ and suppose that

$$
i_k^j = \sum_{m=0}^{k} \binom{k-m+j-1}{j-1} i_m^0, \qquad (6.1.4)
$$

$k = 0, \ldots, \mathbf{d}-1$ Now, $i_0^{j+1} = i_0^j = i_0^0$. For $1 \leq k \leq \mathbf{d}-1$, we have

$$
\begin{aligned}
i_k^{j+1} &= i_k^j + i_{k-1}^{j+1} \\
&= i_k^j + i_{k-1}^j + i_{k-2}^{j+1} \\
&= \ldots \\
&= i_k^j + i_{k-1}^j + \cdots + i_2^j + i_1^j + i_0^{j+1} \\
&= i_k^j + i_{k-1}^j + \cdots + i_2^j + i_1^j + i_0^j. \qquad (6.1.5)
\end{aligned}
$$

Since $i_0^{j+1} = i_0^j$ follows from the algorithm step, we have it that $i_k^{j+1} = \sum_{m=0}^{k} i_m^j$. Hence, by the induction process,

$$
\begin{aligned}
i_k^{j+1} = \sum_{m=0}^{k} i_m^j &= i_0^0 + \sum_{l=0}^{1} \binom{1-l+j-1}{j-1} i_l^0 + \sum_{l=0}^{2} \binom{2-l+j-1}{j-1} i_l^0 + \dots \\
&\quad + \sum_{l=0}^{k} \binom{k-l+j-1}{j-1} i_l^0 \\
&= \sum_{m=0}^{k} \sum_{l=0}^{m} \binom{m-l+j-1}{j-1} i_l^0 \\
&= \sum_{m=0}^{k} \sum_{l=0}^{k} \binom{m-l+j-1}{j-1} i_l^0 \\
&= \sum_{l=0}^{k} \sum_{m=l}^{k} \binom{m-l+j-1}{j-1} i_l^0 \\
&= \sum_{l=0}^{k} \binom{k-l+j}{j} i_l^0. \qquad (6.1.6)
\end{aligned}
$$

Therefore, the result is true for $j+1$

$$
i_k^{j+1} = \sum_{m=0}^{k} \binom{k-m+j}{j} i_m^0 \qquad (6.1.7)
$$

and the result follows by induction.

The algorithm at Stage 3, step $j = \mathbf{d}$

$$
\begin{aligned}
i_0^{\mathbf{d}} &= i_0^{\mathbf{d}-1} = i_0^0 \\
i_1^{\mathbf{d}} &= i_1^{\mathbf{d}-1} = \sum_{m=0}^{1} \binom{1-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 = (\mathbf{d}-1) i_0^0 + i_1^0. \qquad (6.1.8)
\end{aligned}
$$

Implementing the algorithm step $i_k^{\mathbf{d}} = i_{k-2}^{\mathbf{d}} + i_k^{\mathbf{d}-1}$ for $k = 2, \dots \mathbf{d}-1$, we have it that

$$
i_2^{\mathbf{d}} = i_0^{\mathbf{d}} + i_2^{\mathbf{d}-1} = i_0^0 + \sum_{m=0}^{2} \binom{2-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0
$$

$$
\begin{aligned}
i_3^{\mathbf{d}} &= i_1^{\mathbf{d}} + i_2^{\mathbf{d}-1} = \sum_{m=0}^{1} \binom{1-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + \sum_{m=0}^{3} \binom{3-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \\
i_4^{\mathbf{d}} &= i_2^{\mathbf{d}} + i_4^{\mathbf{d}-1} = i_0^0 + \sum_{m=0}^{2} \binom{2-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + \sum_{m=0}^{4} \binom{4-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \\
&\dots
\end{aligned}
$$

Therefore for odd valued k,

$$
\begin{aligned}
i_k^{\mathbf{d}} &= \sum_{t=0}^{\frac{k-1}{2}} \sum_{m=0}^{2t+1} \binom{2t+1-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \\
&= \sum_{t=0}^{\frac{k-1}{2}} \sum_{m=2t}^{2t+1} \binom{2t+1-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \quad \text{(as } \mathbf{d} \text{ is prime)} \\
&= \sum_{t=0}^{\frac{k-1}{2}} (\mathbf{d}-1) i_{2t}^0 + i_{2t+1}^0 \qquad (6.1.9)
\end{aligned}
$$

and similarly for even valued k,

$$
\begin{aligned}
i_k^{\mathbf{d}} &= \sum_{t=0}^{\frac{k}{2}} \sum_{m=0}^{2t} \binom{2t-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \\
&= \sum_{t=0}^{\frac{k}{2}} \sum_{m=2t}^{2t} \binom{2t-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \quad \text{(as } \mathbf{d} \text{ is prime)} \\
&= \sum_{t=0}^{\frac{k}{2}} (\mathbf{d}-1) i_{2t-1}^0 + i_{2t}^0. \qquad (6.1.10)
\end{aligned}
$$

Stage 4, step $j = \mathbf{d}+1$ of the algorithm is given by $i_k^{\mathbf{d}+1} = i_k^{\mathbf{d}} + i_{k+1}^{\mathbf{d}}$ for $k = 0, \dots, \mathbf{d}-2$. Let us consider $i_k^{\mathbf{d+1}}$. There are two cases; for even valued k we note that

$$
i_k^{\mathbf{d}} = i_{k-2}^{\mathbf{d}} + i_k^{\mathbf{d}-1}
$$

$$
\begin{aligned}
&= i^{\mathbf{d}}_{k-4} + i^{\mathbf{d}-1}_{k-2} + i^{\mathbf{d}-1}_{k} \\
&= \dots \\
&= \sum_{t=0}^{\frac{k}{2}} i^{\mathbf{d}-1}_{2t} \qquad (6.1.11)
\end{aligned}
$$

while

$$
\begin{aligned}
i^{\mathbf{d}}_{k+1} &= i^{\mathbf{d}}_{k-1} + i^{\mathbf{d}-1}_{k+1} \\
&= i^{\mathbf{d}}_{k-3} + i^{\mathbf{d}-1}_{k-1} + i^{\mathbf{d}-1}_{k+1} \\
&= \dots \\
&= \sum_{t=0}^{\lfloor \frac{k+1}{2} \rfloor} i^{\mathbf{d}-1}_{2t+1}. \qquad (6.1.12)
\end{aligned}
$$

Therefore, $i^{\mathbf{d}+1}_{k} = \sum_{t=0}^{k+1} i^{\mathbf{d}-1}_{t}$. Alternatively, for odd valued k then $i^{\mathbf{d}}_{k} = \sum_{t=0}^{\frac{k-1}{2}} i^{\mathbf{d}-1}_{2t+1}$ while $i^{\mathbf{d}}_{k+1} = \sum_{t=0}^{\frac{k+1}{2}} i^{\mathbf{d}-1}_{2t}$ and $i^{\mathbf{d}+1}_{k} = \sum_{t=0}^{k+1} i^{\mathbf{d}-1}_{t}$. Hence,

$$
\begin{aligned}
i^{\mathbf{d}+1}_{k} &= \sum_{t=0}^{k+1} i^{\mathbf{d}-1}_{l} \\
&= \sum_{l=0}^{k+1} \sum_{m=0}^{t} \binom{t-m+\mathbf{d}-2}{\mathbf{d}-2} i^{0}_{m} \\
&= \sum_{m=0}^{k+1} \sum_{l=m}^{k+1} \binom{l-m+\mathbf{d}-2}{\mathbf{d}-2} i^{0}_{m} \\
&= \sum_{m=0}^{k+1} \binom{k-m+\mathbf{d}}{\mathbf{d}-1} i^{0}_{m} \\
&= i^{0}_{k+1} \bmod \mathbf{d}. \qquad (6.1.13)
\end{aligned}
$$

Recall that since the dimesion, $\mathbf{d} = p$, considered is prime, under arithmetic modulo $\mathbf{d}$, $\binom{k-m+\mathbf{d}}{\mathbf{d}-1}$ vanishes for $m \neq k+1$ and therefore we deduce that $i^{\mathbf{d}+1}_{k} = i^{0}_{k+1}$ for $k = 0, \dots, \mathbf{d}-2$. When k = $\mathbf{d}$-1 we have

$$
i^{\mathbf{d}+1}_{\mathbf{d}-1} = i^{\mathbf{d}}_{\mathbf{d}-1} = \sum_{t=0}^{\lfloor \frac{\mathbf{d}-1}{2} \rfloor} \sum_{m=0}^{2t} \binom{2t-m+\mathbf{d}-2}{\mathbf{d}-2} i^{0}_{m}. \qquad (6.1.14)
$$

The WilNOT gate concludes at Stage 5 step $j = \mathbf{d}+2$ with the implementation of a sequence of gates targeted on $i_{\mathbf{d}-1}^{\mathbf{d}+1}$. For $k = 0, \ldots, \mathbf{d}-2$, we have the result

$$i_k^{\mathbf{d}+2} \;=\; i_k^{\mathbf{d}+1} \;=\; i_{k+1}^0 \bmod \mathbf{d}. \tag{6.1.15}$$

For $k = \mathbf{d}-1$, the value of $i_{\mathbf{d}-1}^{\mathbf{d}+2}$ is given by

$$\begin{aligned} i_{\mathbf{d}-1}^{\mathbf{d}+2} &= i_{\mathbf{d}-1}^{\mathbf{d}+1} + \sum_{k=0}^{\mathbf{d}-2} \eta_k i_k^{\mathbf{d}+2} \\ &= \sum_{t=0}^{\lfloor \frac{\mathbf{d}-1}{2} \rfloor} \sum_{m=0}^{2t} \binom{2t-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + \sum_{k=0}^{\mathbf{d}-2} \eta_k i_k^{\mathbf{d}+2}. \end{aligned}$$

To show that this returns the desired result, we consider the value $i_{\mathbf{d}-1}^{\mathbf{d}+1} \bmod \mathbf{d}$.

Lemma 3. $i_{\mathbf{d}-1}^{\mathbf{d}+1} \bmod \mathbf{d} = \sum_{t=0}^{\lfloor \frac{\mathbf{d}-1}{2} \rfloor} i_{2t}^0 + \sum_{t=0}^{\lfloor \frac{\mathbf{d}-1}{2} \rfloor - 1} (\mathbf{d}-1) i_{2t+1}^0$.

Proof.

$$\begin{aligned} i_{\mathbf{d}-1}^{\mathbf{d}+1} &= \sum_{t=0}^{\frac{\mathbf{d}-1}{2}} \sum_{m=0}^{2t} \binom{2t-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \\ &= \sum_{m=0}^{\mathbf{d}-1} \sum_{t=\lceil \frac{m}{2} \rceil}^{\frac{\mathbf{d}-1}{2}} \binom{2t-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0. \end{aligned} \tag{6.1.16}$$

Since $\binom{2t-m+\mathbf{d}-2}{\mathbf{d}-2} = 0 \bmod \mathbf{d}$ for $t > \lceil \frac{m}{2} \rceil$ then

$$\begin{aligned} i_{\mathbf{d}-1}^{\mathbf{d}+1} \bmod \mathbf{d} &= \sum_{m=0}^{\mathbf{d}-1} \binom{2\lceil \frac{m}{2} \rceil - m + \mathbf{d} - 2}{\mathbf{d}-2} i_m^0 \\ &= \sum_{l=0}^{\frac{\mathbf{d}-1}{2}} i_{2l}^0 + \sum_{l=0}^{\lfloor \frac{\mathbf{d}-2}{2} \rfloor} (\mathbf{d}-1) i_{2l+1}^0. \end{aligned} \tag{6.1.17}$$

Thus, $i_{\mathbf{d}-1}^{\mathbf{d}+1} \bmod \mathbf{d} = \sum_{t=0}^{\frac{\mathbf{d}-1}{2}} i_{2t}^0 + \sum_{t=0}^{\lfloor \frac{\mathbf{d}-2}{2} \rfloor} (\mathbf{d}-1) i_{2t+1}^0$. By definition of Stage 5, we have it that $\sum_{k=0}^{\mathbf{d}-2} \eta_k i_k^{\mathbf{d}+2} = \sum_{t=0}^{\lfloor \frac{\mathbf{d}-2}{2} \rfloor} (\mathbf{d}-1) i_{2t+1}^{\mathbf{d}+2} + \sum_{t=0}^{\frac{\mathbf{d}-3}{2}} i_{2t}^{\mathbf{d}+2}$. The value of

$i_{\mathbf{d}-1}^{\mathbf{d}+2}$ is then given by

$$\begin{aligned} i_{\mathbf{d}-1}^{\mathbf{d}+2} &= i_{\mathbf{d}-1}^{\mathbf{d}+1} + \sum_{k=0}^{\mathbf{d}-2} \eta_k i_k^{\mathbf{d}+2} \\ &= \sum_{t=0}^{\lfloor \frac{\mathbf{d}-1}{2} \rfloor} i_{2t}^0 + \sum_{t=0}^{\lfloor \frac{\mathbf{d}-2}{2} \rfloor} (\mathbf{d}-1) i_{2t+1}^0 + \sum_{t=0}^{\lfloor \frac{\mathbf{d}-2}{2} \rfloor} i_{2t+1}^0 + \sum_{t=1}^{\lfloor \frac{\mathbf{d}-1}{2} \rfloor} (\mathbf{d}-1) i_{2t}^0. \quad (6.1.18) \end{aligned}$$

Consequently, $i_{\mathbf{d}-1}^{\mathbf{d}+2} \bmod \mathbf{d} = i_0^{\mathbf{d}+2} = i_0^0$. Stage 5 of the algorithm ensures that the WilNOT gate effectuates the transformation of an input sequence given by $i_k^0 = e_k$ for $k = 0, \ldots, \mathbf{d}-1$ to the sequence $i_k^{\mathbf{d}+2} = e_{k+1}$ for $k = 0, \ldots, \mathbf{d}-2$ and $i_{\mathbf{d}-1}^{\mathbf{d}+1} = e_0$, thereby finalising the construction process for a generalised quantum SWAP gate. We show that the network SWAPS all $\mathbf{d}^{\mathbf{d}}$ sequences of input states.

Theorem 20. Let $\mathcal{A}_0, \ldots, \mathcal{A}_{\mathbf{d}-1}$ be $\mathbf{d}$-dimensional systems with bases $|e_0\rangle_j, |e_1\rangle_j, \ldots, |e_{\mathbf{d}-1}\rangle_j$, $j = 0, \ldots, \mathbf{d}-1$, where $e_0, \ldots, e_{\mathbf{d}-1} \in \mathbb{Z}_{\mathbf{d}}$. Let $\mathcal{A} = \mathcal{A}_0 \otimes \cdots \otimes \mathcal{A}_{\mathbf{d}-1}$. If a network implements a SWAP on each basis state $|a_0 a_1 \ldots a_{\mathbf{d}-1}\rangle = |a_0\rangle_0 \otimes |a_1\rangle_1 \otimes \cdots \otimes |a_{\mathbf{d}-1}\rangle_{\mathbf{d}-1}$ of $\mathcal{A}$ where $a_0, \ldots, a_{\mathbf{d}-1} \in \mathbb{Z}_{\mathbf{d}}$ then the network implements a SWAP on any input state $|\psi\rangle = |\psi_0\rangle_0 \otimes |\psi_1\rangle_1 \otimes |\psi_{\mathbf{d}-1}\rangle_{\mathbf{d}-1}$.

Proof: Let $|\psi_j\rangle_j = \sum_{k_j=0}^{\mathbf{d}-1} \alpha_{jk_j} \left|e_{k_j}\right\rangle_j$, $j = 0, \ldots, \mathbf{d}-1$. Then

$$|\psi\rangle = \sum_{k_0=0}^{\mathbf{d}-1} \cdots \sum_{k_{\mathbf{d}-1}}^{\mathbf{d}-1} \alpha_{0k_0} \ldots \alpha_{(\mathbf{d}-1)k_{\mathbf{d}-1}} |k_0 \ldots k_{\mathbf{d}-1}\rangle . \quad (6.1.19)$$

Now,

$$\begin{aligned} \mathrm{SWAP}(|\psi\rangle) &= \sum_{k_0=0}^{d-1} \cdots \sum_{k_{d-1}} \alpha_{0k_0} \ldots \alpha_{(\mathbf{d}-1)k_{\mathbf{d}-1}} \mathrm{SWAP}(|k_0 \ldots k_{\mathbf{d}-1}\rangle) \\ &= \sum_{k_0=0}^{\mathbf{d}-1} \cdots \sum_{k_{\mathbf{d}-1}}^{\mathbf{d}-1} \alpha_{0k_0} \ldots \alpha_{(\mathbf{d}-1)k_{\mathbf{d}-1}} |k_1 \ldots k_{\mathbf{d}-1} k_0\rangle \\ &= \sum_{k_1=0}^{\mathbf{d}-1} \cdots \sum_{k_{\mathbf{d}-1}}^{\mathbf{d}-1} \sum_{k_0=0}^{\mathbf{d}-1} \alpha_{1k_1} \ldots \alpha_{(\mathbf{d}-1)k_{\mathbf{d}-1}} \alpha_{0k_0} |k_1 \ldots k_{\mathbf{d}-1} k_0\rangle \\ &= |\psi_1\rangle_0 \otimes \cdots \otimes |\psi_{\mathbf{d}-1}\rangle_{\mathbf{d}-2} \otimes |\psi_0\rangle_{\mathbf{d}-1} \quad (6.1.20) \end{aligned}$$

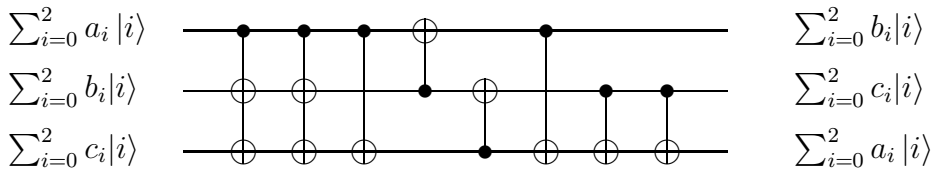


Figure 6.8: Qutrit WilNOT SWAP Network.

as required.

In particular, for an input quantum state of a **d**-fold quantum system whose first system $\mathcal{A}_0$ is prepared in the state $|e_0\rangle_0$, whose second system $\mathcal{A}_1$ is prepared in the state $|e_1\rangle_1$ and so forth, and whose final system $\mathcal{A}_{\mathbf{d}-1}$ of the input state is prepared in the state $|e_{\mathbf{d}-1}\rangle_{\mathbf{d}-1}$, an application of the WilNOT gate over prime dimensions yields a generalised SWAP gate so that the system $\mathcal{A}_0$ is in the state $|e_1\rangle_0$, the system $\mathcal{A}_1$ is in the state $|e_2\rangle_1$ and so forth, until the system $\mathcal{A}_{\mathbf{d}-1}$ is in the state $|e_0\rangle_{\mathbf{d}-1}$. Furthermore, a WilNOT$^{(l)}$, $l < \mathbf{d}$, operator composed of l repeating WilNOT gates can be constructed to effectuate a cyclic shift of quantum states through l quantum systems of a **d**-fold qu**d**it system.

6.2 WilNOT Example: The Qutrit Case

The qubit network that swaps two arbitrary qubit states is well known [60]. When restricted to the qubit setting, the WilNOT operator yields the unitary transformation matrices that swap the states of a pair of arbitrary qubits. We give an example of how WilNOT is used to swap the information content of three arbitrary qutrit states by defining the required unitary transformation

matrices, see Figure 6.8. For the case **d**=3 the WilNOT operator produces the following sequence of states of systems $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ on input $|i\rangle_0, |j\rangle_1, |k\rangle_2$.

Stage 1. $|i\rangle_0 |j\rangle_1 |k\rangle_2$

Stage 2, step 1. $|i\rangle_0 |i+j\rangle_1 |i+j+k\rangle_2$

Stage 2, step 2. $|i\rangle_0 |2i+j\rangle_1 (|3i+2j+k\rangle_2 = |2j+k\rangle_2)$

Stage 3. $|i\rangle_0 |2i+j\rangle_1 |i+2j+k\rangle_2$

Stage 4. $(|3i+j\rangle_0 = |j\rangle_0)(|3i+3j+k\rangle_1 = |k\rangle_1 |i+2j+k\rangle_2$

Stage 5. $|j\rangle_0 |k\rangle_1 (|i+3j+3k\rangle_2 = |i\rangle_2)$

The unitary transformation matrices associated with the WilNOT operator over $\mathbb{C}^{27} \equiv \mathbb{C}^{3^3}$ are as follows; let U_1 be the unitary transformation corresponding to Stage 1, step 1. Then $U_1(|i\rangle_0 |j\rangle_1 |k\rangle_2) = |i\rangle_0 |i+j\rangle_1 |i+j+k\rangle_2$. Let $|a\rangle_0 = \sum_{i=0}^{2} a_i |i\rangle, |b\rangle_1 = \sum_{i=0}^{2} b_i |i\rangle, |c\rangle_2 = \sum_{i=0}^{2} c_i |i\rangle$. Then we may write $|a\rangle_0 \otimes |b\rangle_1 \otimes |c\rangle_2$ as $\sum_{i_1=0}^{2} \sum_{i_2=0}^{2} \sum_{i_3=0}^{2} a_{i_1} b_{i_2} c_{i_3} |i_1 i_2 i_3\rangle$. Thus

$$
\begin{aligned}
&U_1(|a\rangle_0 \otimes |b\rangle_1 \otimes |c\rangle_2) \\
&= U_1\left((a_0|0\rangle + a_1|1\rangle + a_2|2\rangle) \otimes (b_0|0\rangle + b_1|1\rangle + b_2|2\rangle) \otimes (c_0|0\rangle + c_1|1\rangle + c_2|2\rangle)\right) \\
&= U_1(a_0b_0c_0|000\rangle + a_0b_0c_1|001\rangle + a_0b_0c_2|002\rangle + a_0b_1c_0|010\rangle + a_0b_1c_1|011\rangle \\
&+a_0b_1c_2|012\rangle + a_0b_2c_0|020\rangle + a_0b_2c_1|021\rangle + a_0b_2c_2|022\rangle + a_1b_0c_0|100\rangle \\
&+a_1b_0c_1|101\rangle + a_1b_0c_2|102\rangle + a_1b_1c_0|110\rangle + a_1b_1c_1|111\rangle + a_1b_1c_2|112\rangle \\
&+a_1b_2c_0|120\rangle + a_1b_2c_1|121\rangle + a_1b_2c_2|122\rangle + a_2b_0c_0|200\rangle + a_2b_0c_1|201\rangle \\
&+a_2b_0c_2|202\rangle + a_2b_1c_0|210\rangle + a_2b_1c_1|211\rangle + a_2b_1c_2|212\rangle + a_2b_2c_0|220\rangle \\
&+a_2b_2c_1|221\rangle + a_2b_2c_2|222\rangle) \\
&= a_0b_0c_0|000\rangle + a_0b_0c_1|001\rangle + a_0b_0c_2|002\rangle + a_0b_1c_0|011\rangle + a_0b_1c_1|012\rangle \\
&+a_0b_1c_2|010\rangle + a_0b_2c_0|022\rangle + a_0b_2c_1|020\rangle + a_0b_2c_2|021\rangle + a_1b_0c_0|111\rangle \\
&+a_1b_0c_1|112\rangle + a_1b_0c_2|110\rangle + a_1b_1c_0|122\rangle + a_1b_1c_1|120\rangle + a_1b_1c_2|121\rangle \\
&+a_1b_2c_0|100\rangle + a_1b_2c_1|101\rangle + a_1b_2c_2|102\rangle + a_2b_0c_0|222\rangle + a_2b_0c_1|220\rangle
\end{aligned}
$$

$$+a_2b_0c_2\,|221\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|211\rangle$$
$$+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|210\rangle\,. \tag{6.2.1}$$

Thus for Stage 2, step 1 the matrix corresponding to U_1 is (for the usual lexicographic ordering for rows and coloumns)

$$U_1 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0
\end{pmatrix}. \tag{6.2.2}$$

(6.2.3)

The state returned after step 1, is now an entangled state. This is beacause there is a positive entangling power measure associated with the CNOT gate

[90]. As we seek a generalised quantum SWAP gate, we have by extension of the qubit SWAP gate, to construct a quantum gate whose entangling power measure is zero [90]. The state of the WilNOT gate is remains entangled until all the unitary transformations described by WilNOT are applied. The set of unitary transformations are given by definition of each step in WilNOT. We now give the remaining set of unitarity transformations and the action of each unitary on the corresponding input states.

The unitary matrix corresponding to Stage 2, step 2 is the same as U_1. Let the transformation corresponding to Stage 2 of WilNOT be U_2. So $U_2 = U_1^2$. We may write the action of U_2 on the state of the system prior to application of step 2 of WilNOT as

$$
\begin{aligned}
&U_2(|a\rangle \otimes |b\rangle \otimes |c\rangle) \\
&= U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|011\rangle + a_0b_1c_1\,|012\rangle \\
&+a_0b_1c_2\,|010\rangle + a_0b_2c_0\,|022\rangle + a_0b_2c_1\,|020\rangle + a_0b_2c_2\,|021\rangle + a_1b_0c_0\,|111\rangle \\
&+a_1b_0c_1\,|112\rangle + a_1b_0c_2\,|110\rangle + a_1b_1c_0\,|122\rangle + a_1b_1c_1\,|120\rangle + a_1b_1c_2\,|121\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|222\rangle + a_2b_0c_1\,|220\rangle \\
&+a_2b_0c_2\,|221\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|211\rangle \\
&+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|210\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|012\rangle + a_0b_1c_1\,|010\rangle \\
&+a_0b_1c_2\,|011\rangle + a_0b_2c_0\,|021\rangle + a_0b_2c_1\,|022\rangle + a_0b_2c_2\,|020\rangle + a_1b_0c_0\,|120\rangle \\
&+a_1b_0c_1\,|121\rangle + a_1b_0c_2\,|122\rangle + a_1b_1c_0\,|102\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|101\rangle \\
&+a_1b_2c_0\,|111\rangle + a_1b_2c_1\,|112\rangle + a_1b_2c_2\,|110\rangle + a_2b_0c_0\,|210\rangle + a_2b_0c_1\,|211\rangle \\
&+a_2b_0c_2\,|212\rangle + a_2b_1c_0\,|222\rangle + a_2b_1c_1\,|220\rangle + a_2b_1c_2\,|221\rangle + a_2b_2c_0\,|201\rangle \\
&+a_2b_2c_1\,|202\rangle + a_2b_2c_2\,|200\rangle\,. \qquad (6.2.4)
\end{aligned}
$$

The matrix corresponding to U_2 is

$$U_2 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0
\end{pmatrix}. \tag{6.2.5}$$

Let U_3 denote the unitary transformation corresponding to Stage 3 of WilNOT. Then $U_3(|ijk\rangle) = |ij(i+k)\rangle$. The result of applying U_3 is

$$U_3(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|012\rangle + a_0b_1c_1\,|010\rangle$$

$$+a_0b_1c_2\,|011\rangle + a_0b_2c_0\,|021\rangle + a_0b_2c_1\,|022\rangle + a_0b_2c_2\,|020\rangle + a_1b_0c_0\,|120\rangle$$

$$+a_1b_0c_1\,|121\rangle + a_1b_0c_2\,|122\rangle + a_1b_1c_0\,|102\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|101\rangle$$

$$U_3 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0
\end{pmatrix} \tag{6.2.6}$$

$$+a_1b_2c_0\,|111\rangle + a_1b_2c_1\,|112\rangle + a_1b_2c_2\,|110\rangle + a_2b_0c_0\,|210\rangle + a_2b_0c_1\,|211\rangle$$

$$+a_2b_0c_2\,|212\rangle + a_2b_1c_0\,|222\rangle + a_2b_1c_1\,|220\rangle + a_2b_1c_2\,|221\rangle + a_2b_2c_0\,|201\rangle$$

$$+a_2b_2c_1\,|202\rangle + a_2b_2c_2\,|200\rangle)$$

$$= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|012\rangle + a_0b_1c_1\,|010\rangle$$

$$
\begin{aligned}
&+a_0b_1c_2\,|011\rangle + a_0b_2c_0\,|021\rangle + a_0b_2c_1\,|022\rangle + a_0b_2c_2\,|020\rangle + a_1b_0c_0\,|121\rangle \\
&+a_1b_0c_1\,|122\rangle + a_1b_0c_2\,|120\rangle + a_1b_1c_0\,|100\rangle + a_1b_1c_1\,|101\rangle + a_1b_1c_2\,|102\rangle \\
&+a_1b_2c_0\,|112\rangle + a_1b_2c_1\,|110\rangle + a_1b_2c_2\,|111\rangle + a_2b_0c_0\,|212\rangle + a_2b_0c_1\,|210\rangle \\
&+a_2b_0c_2\,|211\rangle + a_2b_1c_0\,|221\rangle + a_2b_1c_1\,|222\rangle + a_2b_1c_2\,|220\rangle + a_2b_2c_0\,|200\rangle \\
&+a_2b_2c_1\,|201\rangle + a_2b_2c_2\,|202\rangle\,. \qquad (6.2.7)
\end{aligned}
$$

Let U_4 and U_5 be the unitary transformations given by $U_4(|ijk\rangle) = |(i+j)jk\rangle$ and $U_5(|ijk\rangle) = |i(j+k)k\rangle$. Then Stage 4 of WilNOT is done by applying U_4 and then U_5. The results of applying U_4 and U_5 are

$$
\begin{aligned}
&U_4(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|012\rangle + a_0b_1c_1\,|010\rangle \\
&+a_0b_1c_2\,|011\rangle + a_0b_2c_0\,|021\rangle + a_0b_2c_1\,|022\rangle + a_0b_2c_2\,|020\rangle + a_1b_0c_0\,|121\rangle \\
&+a_1b_0c_1\,|122\rangle + a_1b_0c_2\,|120\rangle + a_1b_1c_0\,|100\rangle + a_1b_1c_1\,|101\rangle + a_1b_1c_2\,|102\rangle \\
&+a_1b_2c_0\,|112\rangle + a_1b_2c_1\,|110\rangle + a_1b_2c_2\,|111\rangle + a_2b_0c_0\,|212\rangle + a_2b_0c_1\,|210\rangle \\
&+a_2b_0c_2\,|211\rangle + a_2b_1c_0\,|221\rangle + a_2b_1c_1\,|222\rangle + a_2b_1c_2\,|220\rangle + a_2b_2c_0\,|200\rangle \\
&+a_2b_2c_1\,|201\rangle + a_2b_2c_2\,|202\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|112\rangle + a_0b_1c_1\,|110\rangle \\
&+a_0b_1c_2\,|111\rangle + a_0b_2c_0\,|221\rangle + a_0b_2c_1\,|222\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|021\rangle \\
&+a_1b_0c_1\,|022\rangle + a_1b_0c_2\,|020\rangle + a_1b_1c_0\,|100\rangle + a_1b_1c_1\,|101\rangle + a_1b_1c_2\,|102\rangle \\
&+a_1b_2c_0\,|212\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|211\rangle + a_2b_0c_0\,|012\rangle + a_2b_0c_1\,|010\rangle \\
&+a_2b_0c_2\,|011\rangle + a_2b_1c_0\,|121\rangle + a_2b_1c_1\,|122\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|200\rangle \\
&+a_2b_2c_1\,|201\rangle + a_2b_2c_2\,|202\rangle \qquad (6.2.8)
\end{aligned}
$$

and

$$
U_5(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|112\rangle + a_0b_1c_1\,|110\rangle
$$

$$U_4 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0
\end{pmatrix} \tag{6.2.9}$$

$$+a_0b_1c_2\,|111\rangle + a_0b_2c_0\,|221\rangle + a_0b_2c_1\,|222\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|021\rangle$$

$$+a_1b_0c_1\,|022\rangle + a_1b_0c_2\,|020\rangle + a_1b_1c_0\,|100\rangle + a_1b_1c_1\,|101\rangle + a_1b_1c_2\,|102\rangle$$

$$+a_1b_2c_0\,|212\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|211\rangle + a_2b_0c_0\,|012\rangle + a_2b_0c_1\,|010\rangle$$

$$+a_2b_0c_2\,|011\rangle + a_2b_1c_0\,|121\rangle + a_2b_1c_1\,|122\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|200\rangle$$

$$U_5 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{pmatrix}. \quad (6.2.10)$$

$$+a_2b_2c_1\,|201\rangle + a_2b_2c_2\,|202\rangle)$$

$$= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|011\rangle + a_0b_0c_2\,|022\rangle + a_0b_1c_0\,|102\rangle + a_0b_1c_1\,|110\rangle$$

$$+a_0b_1c_2\,|121\rangle + a_0b_2c_0\,|201\rangle + a_0b_2c_1\,|212\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|001\rangle$$

$$+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|020\rangle + a_1b_1c_0\,|100\rangle + a_1b_1c_1\,|111\rangle + a_1b_1c_2\,|122\rangle$$

$$
\begin{aligned}
&+a_1b_2c_0\,|202\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|221\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|010\rangle \\
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|101\rangle + a_2b_1c_1\,|112\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|200\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|222\rangle\,. \qquad (6.2.11)
\end{aligned}
$$

Let U_6 and U_7 be the unitary transformations given by $U_6(|ijk\rangle) = |ij(i+k)\rangle$ and $U_7(|ijk\rangle) = |ij(j+k)\rangle$. Then Stage 5 of WilNOT is done by applying U_6 and then applying U_7 twice. The result of applying U_6 is

$$
\begin{aligned}
&U_6(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|011\rangle + a_0b_0c_2\,|022\rangle + a_0b_1c_0\,|102\rangle + a_0b_1c_1\,|110\rangle \\
&+a_0b_1c_2\,|121\rangle + a_0b_2c_0\,|201\rangle + a_0b_2c_1\,|212\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|020\rangle + a_1b_1c_0\,|100\rangle + a_1b_1c_1\,|111\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|202\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|221\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|010\rangle \\
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|101\rangle + a_2b_1c_1\,|112\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|200\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|222\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|011\rangle + a_0b_0c_2\,|022\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|111\rangle \\
&+a_0b_1c_2\,|122\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|211\rangle + a_0b_2c_2\,|222\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|020\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|112\rangle + a_1b_1c_2\,|120\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|212\rangle + a_1b_2c_2\,|220\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|010\rangle \\
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|110\rangle + a_2b_1c_2\,|121\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|210\rangle + a_2b_2c_2\,|221\rangle\,. \qquad (6.2.12)
\end{aligned}
$$

Similarly, the action of U_7 on the state 6.2.12 may be given as

$$
\begin{aligned}
&U_7(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|011\rangle + a_0b_0c_2\,|022\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|111\rangle \\
&+a_0b_1c_2\,|122\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|211\rangle + a_0b_2c_2\,|222\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|020\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|112\rangle + a_1b_1c_2\,|120\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|212\rangle + a_1b_2c_2\,|220\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|010\rangle
\end{aligned}
$$

$$
\begin{aligned}
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|110\rangle + a_2b_1c_2\,|121\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|210\rangle + a_2b_2c_2\,|221\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|012\rangle + a_0b_0c_2\,|021\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|112\rangle \\
&+a_0b_1c_2\,|121\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|212\rangle + a_0b_2c_2\,|221\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|022\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|110\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|222\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|011\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|111\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|220\rangle\,. \qquad (6.2.13)
\end{aligned}
$$

The second application of U_7 gives

$$
\begin{aligned}
&U_7(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|012\rangle + a_0b_0c_2\,|021\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|112\rangle \\
&+a_0b_1c_2\,|121\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|212\rangle + a_0b_2c_2\,|221\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|022\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|110\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|222\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|011\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|111\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|220\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|010\rangle + a_0b_0c_2\,|020\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|110\rangle \\
&+a_0b_1c_2\,|120\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|210\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|011\rangle + a_1b_0c_2\,|021\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|111\rangle + a_1b_1c_2\,|121\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|211\rangle + a_1b_2c_2\,|221\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|012\rangle \\
&+a_2b_0c_2\,|022\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|112\rangle + a_2b_1c_2\,|122\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|222\rangle\,. \qquad (6.2.14)
\end{aligned}
$$

The unitary matrices U_6, and U_7 have the following forms

$$U_6 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0
\end{pmatrix} \tag{6.2.15}$$

and

$$U_7 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0
\end{pmatrix}. \quad (6.2.16)$$

Note that state 6.2.14, the state of the system following the application of the last unitary transformation as defined by WilNOT, can be written as

$$a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|010\rangle + a_0b_0c_2\,|020\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|110\rangle$$
$$+a_0b_1c_2\,|120\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|210\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|001\rangle$$
$$+a_1b_0c_1\,|011\rangle + a_1b_0c_2\,|021\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|111\rangle + a_1b_1c_2\,|121\rangle$$

$$+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|211\rangle + a_1b_2c_2\,|221\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|012\rangle$$
$$+a_2b_0c_2\,|022\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|112\rangle + a_2b_1c_2\,|122\rangle + a_2b_2c_0\,|202\rangle$$
$$+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|222\rangle$$

$$\text{SWAP} = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{pmatrix} \tag{6.2.17}$$

$$= b_0c_0a_0\,|000\rangle + b_0c_0a_1\,|001\rangle + b_0c_0a_2\,|002\rangle + b_0c_1a_0\,|010\rangle + b_0c_1a_1\,|011\rangle$$

$$
\begin{aligned}
&+b_0c_1a_2\,|012\rangle + b_0c_2a_0\,|020\rangle + b_0c_2a_1\,|021\rangle + b_0c_2a_2\,|022\rangle + b_1c_0a_0\,|100\rangle \\
&+b_1c_0a_1\,|101\rangle + b_1c_0a_2\,|102\rangle + b_1c_1a_0\,|110\rangle + b_1c_1a_1\,|111\rangle + b_1c_1a_2\,|112\rangle \\
&+b_1c_2a_0\,|121\rangle + b_1c_2a_1\,|121\rangle + b_1c_2a_2\,|122\rangle + b_2c_0a_0\,|200\rangle + b_2c_0a_1\,|201\rangle \\
&+b_2c_0a_2\,|202\rangle + b_2c_1a_0\,|210\rangle + b_2c_1a_1\,|211\rangle + b_2c_1a_2\,|212\rangle + b_2c_2a_0\,|220\rangle \\
&+b_2c_2a_1\,|221\rangle + b_2c_2a_2\,|222\rangle\,. \qquad (6.2.18)
\end{aligned}
$$

The state (6.2.18) is separable and has the form

$$
\begin{aligned}
&(b_0\,|0\rangle + b_1\,|1\rangle + b_2\,|2\rangle) \otimes (c_0\,|0\rangle + c_1\,|1\rangle + c_2\,|2\rangle) \otimes (a_0\,|0\rangle + a_1\,|1\rangle + a_2\,|2\rangle) \\
&= |b\rangle \otimes |c\rangle \otimes |a\rangle\,, \qquad (6.2.19)
\end{aligned}
$$

thereby illustrating WilNOT to be a quantum SWAP gate. Figure 6.8 gives the WilNOT network for a qutrit SWAP of arbitrary quantum states. The qutrit SWAP matrix for swapping the information content of three arbitrary qutrit state is the product of the above transformations and is given by SWAP, equation 6.2.17.

6.3 On the WilNOT Gate over Even Dimensions Greater than Two

In this section, we consider the question of whether or not the WilNOT operator can be altered so that a generalised SWAP gate can be constructed over even dimensions. This question is motivated by the case $\mathbf{d} = 4$ in which we considered if it was possible to swap the states of four 4-dimensional system whereby first system $\mathcal{A}_0$ prepared in the state $|e_0\rangle_0$ is left in the state $|e_1\rangle_0$, the second system $\mathcal{A}_1$ is prepared in the state $|e_1\rangle_1$ is left in the state $|e_2\rangle_1$, the third system $\mathcal{A}_2$ prepared in the state $|e_2\rangle_2$ is left in the state $|e_3\rangle_2$ and finally the system $\mathcal{A}_3$ prepared in the state $|e_3\rangle_3$ is left in the state $|e_0\rangle_3$. To this end, let us consider an operator with Stage 1 and Stage 2 identical to those of the WilNOT gate given in section 6.1. By (6.1.4) (with $j = \mathbf{d} - 1$), at Stage 2 and step $j = \mathbf{d} - 1$, the state of the algorithm is given by $i_0^{\mathbf{d}-1} = i_0^0$, and $i_k^{\mathbf{d}-1} = \sum_{m=0}^{k} \binom{k-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0$ for $k = 1, \ldots, \mathbf{d} - 1$. To effectuate the algorithm state (see (6.1.9) and (6.1.10))

$$\begin{pmatrix} i_0^0 \\ (\mathbf{d}-1)i_0^0 + i_1^0 \\ i_0^0 + (\mathbf{d}-1)i_1^0 + i_2^0 \\ \vdots \\ (\mathbf{d}-1)i_0^0 + i_1^0 + (\mathbf{d}-1)i_2^0 + \cdots + i_{\mathbf{d}-1}^0 \end{pmatrix} \tag{6.3.1}$$

on systems $\mathcal{A}_0, \ldots, \mathcal{A}_{\mathbf{d}-1}$, the WilNOT algorithm process at Stage 3 for prime $\mathbf{d}$ given in Section 6.1 requires revision when we consider dimensions $\mathbf{d} = 0 \bmod 2$. Instead we take $i_k^{\mathbf{d}-1}$ with the following linear combination

$$\sum_{s=0}^{k-2} a_s i_{k-2-s}^{\mathbf{d}-1} = \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \right), \tag{6.3.2}$$

where

$$a_s = \mathbf{d} - \left[\binom{s+2+\mathbf{d}-2}{\mathbf{d}-2} + \sum_{t=0}^{s-1} a_t \binom{s-t+\mathbf{d}-2}{\mathbf{d}-2} \right] + (-1)^s,$$

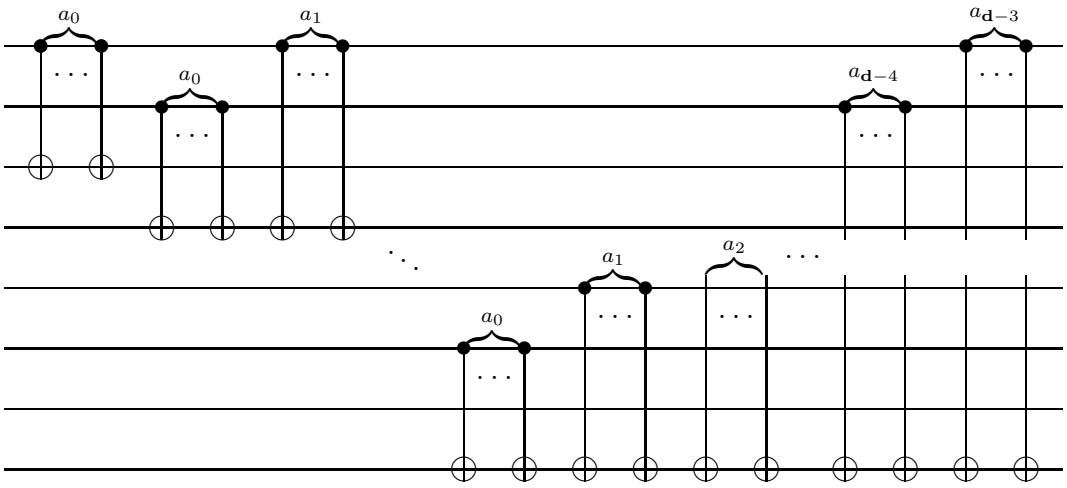


Figure 6.9: WilNOT gate over dimensions $\mathbf{d} = 0 \bmod 2$; Stage 3.

and the result modulo $\mathbf{d}$ given by (6.3.1) can be obtained for Stage 3, step $j = \mathbf{d}$.

Theorem 21. For $\mathbf{d} = 0 \bmod 2$, the algorithm process at Stage 3, step $j = \mathbf{d}$ given by

$$\begin{aligned} i_k^{\mathbf{d}} &= i_k^{\mathbf{d}-1} + \sum_{s=0}^{k-2} a_s i_{k-2-s}^{\mathbf{d}-1} \\ &= \sum_{m=0}^{k} \binom{k-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \right), \end{aligned}$$

for $k = 0, \ldots, \mathbf{d}-1$, returns outcome (6.3.1).

Proof. For $k = 0, 1$, we have it that $i_0^{\mathbf{d}} = i_0^{\mathbf{d}-1}$ and $i_1^{\mathbf{d}} = i_1^{\mathbf{d}-1}$. Thus, the states e_0 and e_1 are given as i_0^0 and $(\mathbf{d}-1)i_0^0 + i_1^0$ respectively. The state e_2 is written as

$$i_2^{\mathbf{d}} = \sum_{m=0}^{2} \binom{2-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + a_0 i_0^0$$

$$
\begin{aligned}
&= \left(\binom{\mathbf{d}}{\mathbf{d}-2} + (\mathbf{d} - \binom{\mathbf{d}}{\mathbf{d}-2} + 1)\right) i_0^0 + (\mathbf{d}-1)i_1^0 + i_2^0 \\
&= (i_0^0 + (\mathbf{d}-1)i_1^0 + i_2^0) \bmod \mathbf{d}. \qquad (6.3.3)
\end{aligned}
$$

We show by induction that, for $k = 0 \bmod 2$, $i_k^{\mathbf{d}} = \sum_{m=0}^{k} \binom{k-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \right) = i_0^0 + (\mathbf{d}-1)i_1^0 + i_2^0 + \cdots + (\mathbf{d}-1)i_{k-1}^0 + i_k^0$ modulo $\mathbf{d}$ and for $k \neq 0 \bmod 2$, $i_k^{\mathbf{d}} = (\mathbf{d}-1)i_0^0 + i_1^0 + (\mathbf{d}-1)i_2^0 + \cdots + (\mathbf{d}-1)i_{k-1}^0 + i_k^0$ modulo $\mathbf{d}$. We have shown that this is true for $k = 0, 1, 2$. Suppose $0 \leq k \leq \mathbf{d}-2$ and further suppose that

$$
\begin{aligned}
i_k^{\mathbf{d}} &= \sum_{m=0}^{k} \binom{k-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \right) \\
&= \sum_{m=0}^{k} (-1)^{k-m} i_m^0 \qquad (6.3.4) \\
&= (i_0^0 + (\mathbf{d}-1)i_1^0 + i_2^0 + \cdots + (\mathbf{d}-1)i_{k-1}^0 + i_k^0) \bmod \mathbf{d}
\end{aligned}
$$

for $k = 0 \bmod 2$, and

$$
= ((\mathbf{d}-1)i_0^0 + i_1^0 + (\mathbf{d}-1)i_2^0 + \cdots + (\mathbf{d}-1)i_{k-1}^0 + i_k^0) \bmod \mathbf{d}
$$

for $k \neq 0 \bmod 2$. Therefore, for $j = \mathbf{d}$, we have,

$$
\begin{aligned}
i_{k+1}^{\mathbf{d}} &= \sum_{m=0}^{k+1} \binom{k+1-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 + \sum_{s=0}^{k-1} \left(a_s \sum_{m=0}^{(k-1)-s} \binom{(k-1)-s-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0 \right) \\
&= \sum_{m=0}^{k} \left(\binom{k-m+\mathbf{d}-2}{\mathbf{d}-2} i_{m+1}^0 + \binom{k+1+\mathbf{d}-2}{\mathbf{d}-2} i_0^0 \right) \\
&+ \sum_{s=0}^{k-1} a_s \left(\left(\sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+\mathbf{d}-2}{\mathbf{d}-2} i_{m+1}^0 \right) + \binom{(k-1)-s+\mathbf{d}-2}{\mathbf{d}-2} i_0^0 \right) \\
&= \sum_{m=0}^{k} (-1)^{k-m} i_{m+1}^0 + \left(\binom{k+1+\mathbf{d}-2}{\mathbf{d}-2} + \sum_{s=0}^{k-1} a_s \binom{k-1-s+\mathbf{d}-2}{\mathbf{d}-2} \right) i_0^0. \qquad (6.3.5)
\end{aligned}
$$

Recall that the binomial coefficients of $i_k^{\mathbf{d}-1} = \sum_{m=0}^{k} \binom{k-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0$ are precisely those coefficients of $i_{k+1}^{\mathbf{d}-1} = \sum_{m=0}^{k+1} \binom{k+1-m+\mathbf{d}-2}{\mathbf{d}-2} i_m^0$ for $m = 1, \ldots, k+1$. Hence,

the particular combination of systems $\mathcal{A}_{(k-2)-s}$ that return the state $i_k^{\mathbf{d}} = (i_0^0 + (\mathbf{d}-1)i_1^0 + i_2^0 + \cdots + (\mathbf{d}-1)i_{k-1}^0 + i_k^0) \bmod \mathbf{d}$ is the combination that effectuates a similar sequence on $i_{k+1}^{\mathbf{d}}$ for $m = 1, \ldots, k+1$. Since $k = 0 \bmod 2$, then for $i_{k+1}^{\mathbf{d}}$ we require that the scalar value for i_0^0 degenerates to $(\mathbf{d}-1) \bmod \mathbf{d}$. Thus, for $m = 0$ and by definition of a_s, we have

$$\begin{aligned}
&\left(\binom{(k+1)+\mathbf{d}-2}{\mathbf{d}-2} + \sum_{s=0}^{k-1} a_s \binom{(k-1)-s+\mathbf{d}-2}{\mathbf{d}-2}\right) i_0^0 \\
&= \left(\binom{(k+1)+\mathbf{d}-2}{\mathbf{d}-2} + \sum_{s=0}^{k-2} \left(a_s \binom{(k-1)-s+\mathbf{d}-2}{\mathbf{d}-2}\right) + a_{k-1}\right) i_0^0 \\
&= \left(\binom{(k+1)+\mathbf{d}-2}{\mathbf{d}-2} + \sum_{s=0}^{k-2} a_s \binom{(k-1)-s+\mathbf{d}-2}{\mathbf{d}-2}\right. \\
&\left. + \left(\mathbf{d} - \left[\binom{(k+1)+\mathbf{d}-2}{\mathbf{d}-2} + \sum_{s=0}^{k-2} a_s \binom{(k-1)-s+\mathbf{d}-2}{\mathbf{d}-2}\right] + (-1)^{k+1}\right)\right) i_0^0 \\
&= (-1)^{k+1} i_0^0. \qquad (6.3.6)
\end{aligned}$$

Hence, $i_{k+1}^{\mathbf{d}} = \sum_{m=0}^{k+1} (-1)^{k+1-m} i_m^0$, and the result follows. We now implement Stage 4 of the WilNOT gate, which is written as

$$i_k^{\mathbf{d}+1} \;=\; i_k^{\mathbf{d}} + i_{k+1}^{\mathbf{d}} \qquad (6.3.7)$$

for $k = 0, \ldots, \mathbf{d}-2$ and

$$i_{\mathbf{d}-1}^{\mathbf{d}+1} \;=\; i_{\mathbf{d}-1}^{\mathbf{d}} + \sum_{m=0}^{\mathbf{d}-2} (-1)^{\mathbf{d}-1-s} i_m^0, \qquad (6.3.8)$$

and a revised Stage 5 given by $\sum_{k=1}^{\mathbf{d}-1} \eta_k^* i_k^{\mathbf{d}+2} = \sum_{t=0}^{\lfloor \frac{\mathbf{d}-1}{2} \rfloor} (\mathbf{d}-1) i_{2t+1}^0 + \sum_{t=0}^{\frac{\mathbf{d}-2}{2}} i_{2t}^0$ which then returns

$$\begin{pmatrix} i_1^0 \\ i_2^0 \\ i_3^0 \\ \vdots \\ i_{\mathbf{d}-1}^0 \\ (\mathbf{d}-1) i_0^0 \end{pmatrix}. \qquad (6.3.9)$$

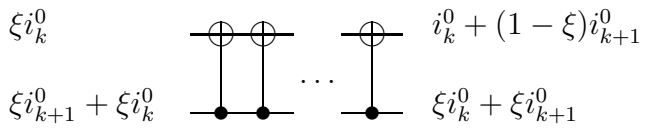


Figure 6.10: $P_\xi - 1$ gates on pairs (i_k, i_{k+1}).

So, although we have not achieved our aim, our modification of the WilNOT gate for $\mathbf{d}$ even has produced a similar state, namely a SWAP but with a sign change for the system $\mathcal{A}_{\mathbf{d}-1}$. We have not been able to modify WilNOT to produce the SWAP gate. We give the following argument to show that a different approach would be required. Result (6.3.9) is a particular outcome for an even valued $\mathbf{d}$ and once entered into this sequence of CNOTs seems not to return an output with scalars on $e_0, \ldots, e_{\mathbf{d}-1}$ all equal to unity. To show this claim, let us consider the more general case given by

$$\begin{pmatrix} \xi i_1^0 \\ \xi i_2^0 \\ \xi i_3^0 \\ \vdots \\ \xi i_{\mathbf{d}-1}^0 \\ (\mathbf{d}-\xi) i_0^0 \end{pmatrix}. \tag{6.3.10}$$

Consider the pairs $(\xi i_k^0, \xi i_{k+1}^0)$, for $k \in \{1, \ldots, \mathbf{d}-3\}$, and further consider the pair $(\xi i_{\mathbf{d}-1}^0, \xi i_0^0)$.

Given the paired input sequence $(\xi i_k^0, \xi i_{k+1}^0)$, for $k \in \{1, \ldots, \mathbf{d}-3\}$, and a mapping that targets i_{k+1}, we have it that $(\xi i_k^0, \xi i_{k+1}^0) \mapsto (\xi i_k^0, \xi i_k^0 + \xi i_{k+1}^0)$. Denote by P_ξ the inverse mod $\mathbf{d}$ of ξ, whence, $P_\xi \xi = 1 \bmod \mathbf{d}$. Applying $P_\xi - 1$ gates, see Figure 6.10, from the control with value $\xi i_k^0 + \xi i_k^0$ to the

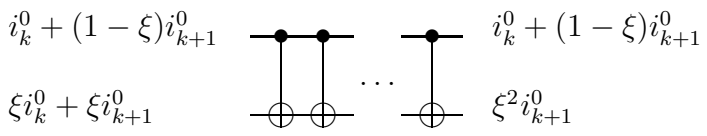


Figure 6.11: $\mathbf{d} - \xi$ gates on $(i_{\mathbf{d}-2}, i_{\mathbf{d}-1})$.

target corresponding to the output ξi_k^0 results in

$$\begin{aligned}(\xi i_k^0, \xi i_k^0 + \xi i_{k+1}^0) &\mapsto (P_\xi \xi i_k^0 + (P_\xi - 1)\xi i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0) \\ &= (i_k^0 + (1-\xi)i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0). \qquad (6.3.11)\end{aligned}$$

To eliminate the value ξi_k^0 from result (6.3.11), $\mathbf{d} - \xi$ gates are implemented on the target $\xi i_k^0 + \xi i_{k+1}^0$ thereby illustrating the mapping

$$\begin{aligned}(i_k^0 + (1-\xi)i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0) &\mapsto (i_k^0 + (1-\xi)i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0 \\ &\quad + (\mathbf{d} - \xi)(i_k^0 + (1-\xi)i_{k+1}^0)) \\ &= (i_k^0 + (1-\xi)i_{k+1}^0, \xi i_{k+1}^0 \\ &\quad + (\mathbf{d} - \xi)(1-\xi)i_{k+1}^0)) \\ &= (i_k^0 + (1-\xi)i_{k+1}^0, \xi i_{k+1}^0 \\ &\quad + (-\xi + \xi^2)i_{k+1}^0) \\ &= (i_k^0 + (1-\xi)i_{k+1}^0, \xi^2 i_{k+1}^0). \qquad (6.3.12)\end{aligned}$$

In a similar fashion, let us consider the final pair $(\xi i_{\mathbf{d}-1}, (\mathbf{d} - \xi)i_0)$. Applying those gates that correspond to result (6.3.11) and result (6.3.12) on the pair $\xi i_{\mathbf{d}-1}, (\mathbf{d} - \xi)i_0$ returns the outcome $(i_{\mathbf{d}-1}^0 + (\xi - 1)i_0^0, -\xi^2 i_0^0)$. Thus, we have

the mapping given by

$$\begin{pmatrix} \xi i_1^0 \\ \xi i_2^0 \\ \vdots \\ \xi i_{\mathbf{d}-3}^0 \\ \xi i_{\mathbf{d}-2}^0 \\ \xi i_{\mathbf{d}-1}^0 \\ (\mathbf{d}-\xi) i_0^0 \end{pmatrix} \quad \mapsto \quad \begin{pmatrix} i_1^0 + (1-\xi) i_2^0 \\ \xi^2 i_2^0 \\ \vdots \\ i_{\mathbf{d}-3}^0 + (1-\xi) i_{\mathbf{d}-2}^0 \\ \xi^2 i_{\mathbf{d}-2}^0 \\ i_{\mathbf{d}-1}^0 + (\xi-1) i_0^0 \\ -\xi^2 i_0^0 \end{pmatrix}. \tag{6.3.13}$$

Since the scalar values ξ^2 and $-\xi^2$ can not simultaneously be unity, it seems that any mapping on the state (6.3.10) will fail to return a state whose scalars values all equal unity. That such outcome in result (6.3.13) is best possible suggests that the WilNOT algorithm fails to extend over dimensions $\mathbf{d} = 0 \bmod 2$. Therefore, it seems that WilNOT cannot be modified for the case $\mathbf{d}$ even to permit a cycle of states such that first system $\mathcal{A}_0$ prepared in the state $|e_0\rangle_0$ is left in the state $|e_1\rangle_0$, the second system $\mathcal{A}_1$ is prepared in the state $|e_1\rangle_1$ is left in the state $|e_2\rangle_1$, the third system $\mathcal{A}_2$ prepared in the state $|e_2\rangle_2$ is left in the state $|e_3\rangle_2$ and finally the system $\mathcal{A}_3$ prepared in the state $|e_3\rangle_3$ is left in the state $|e_0\rangle_3$. Interestingly, WilNOT can demonstrate the case where first system $\mathcal{A}_0$ prepared in the state $|e_0\rangle_0$ is left in the state $|e_2\rangle_0$, the second system $\mathcal{A}_1$ is prepared in the state $|e_1\rangle_1$ is left in the state $|e_3\rangle_1$, the third system $\mathcal{A}_2$ prepared in the state $|e_2\rangle_2$ is left in the state $|e_0\rangle_2$ and finally the system $\mathcal{A}_3$ prepared in the state $|e_3\rangle_3$ is left in the state $|e_1\rangle_3$.

Chapter 7

On Binomial Summations and an Efficient Generalised Quantum SWAP Gate

Let us consider a set of $\mathbf{d}$ qu**d**it quantum systems, the first system $\mathcal{A}_0$ prepared in the state $|e_0\rangle_0$, the second system $\mathcal{A}_1$ prepared in the state $|e_1\rangle_1$ and so forth, with the final system $\mathcal{A}_{\mathbf{d}-1}$ prepared in the state $|e_{\mathbf{d}-1}\rangle_{\mathbf{d}-1}$. We ask whether or not it is possible to construct a network to implement an efficient generalised SWAP gate so that in the output of the network the system $\mathcal{A}_0$ is in the state $|e_1\rangle_0$, the system $\mathcal{A}_1$ is in the state $|e_2\rangle_1$ and so forth, until the system $\mathcal{A}_{\mathbf{d}-1}$ is in the state $|e_0\rangle_{\mathbf{d}-1}$ where $e_0, \ldots, e_{\mathbf{d}-1} \in \{0, \ldots, \mathbf{d}-1\}$.

7.1 The Construction

Let k and l be positive integers. For non-negative integers j, the function $f(j) = \binom{j}{k} \bmod l$ is a cyclic function of j [Lu, Tsai] [55]. We use the periodic property of this function to study the computational problem associated with the construction of an efficient generalised SWAP gate for quantum systems over dimensions $\mathbf{d}$. In particular, the binomial summation $a_j = \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$ will be related to a quantum network for an efficient quantum gate,

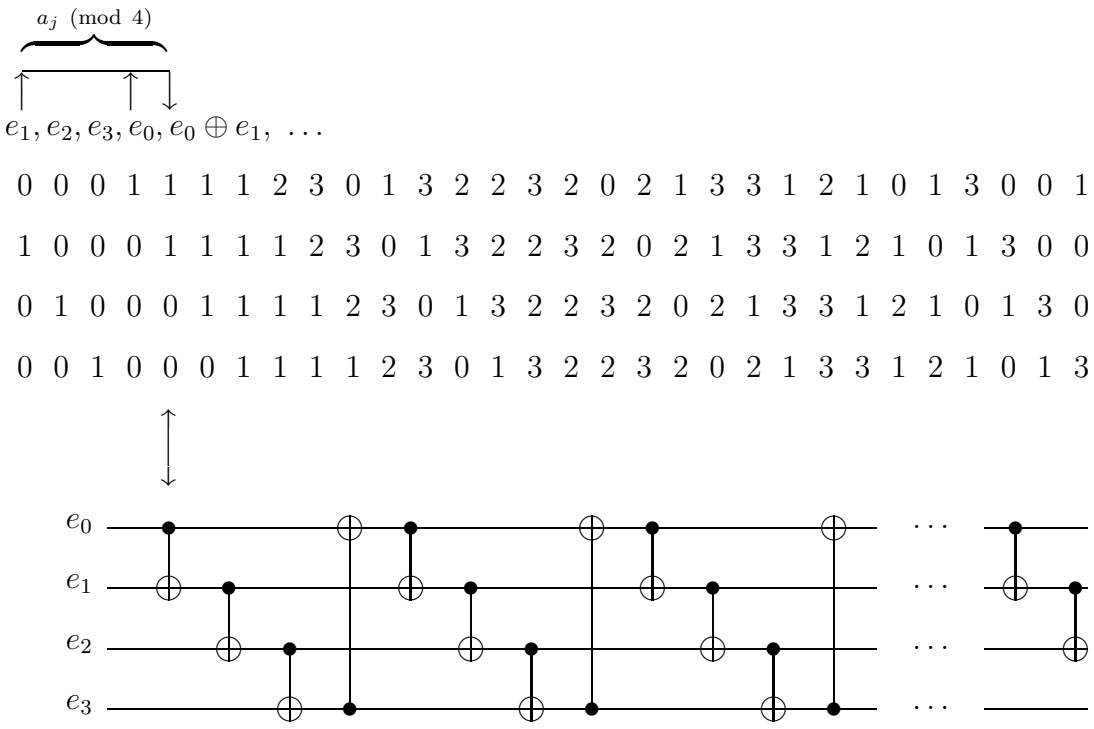


Figure 7.1: Binomial summation Quantum SWAP network construction over dimension 4.

where such a network is constructed from by periodic application of CNOT gates connecting adjacent systems of the network.

Figure 7.1 illustrates such a quantum netwrok for 4-dimensional quantum states. The diagram shows the sequence of CNOT gates of the network. The columns of the array on integers (modulo 4) describe the states of the target systems after the corresponding CNOT gates have been applied, so the arrow indicates the correspondence between the system $\mathcal{A}_1$ being in the state $|e_0 + e_1\rangle_1$ (where $e_0 + e_1$ is calculated modulo 4) and the application of the

first CNOT gate. The sum $e_0 + e_1$ may be represented as the dot product of the row vector e_0, e_1, e_2, e_3 and the corresponding column $(1, 1, 0, 0)^T$. Thus the array of integers (modulo 4) has rows indexed by $\{0, 1, 2, 3\}$ and columns indexed by (time $t =$) $-3, -2, -1, 0, 1, 2, 3, \ldots$ such that column $t = 4s + j$ with $j \in \{0, 1, 2, 3\}$ corresponds to system $\mathcal{A}_j$.

Denote the entries in the array by b_{it}, where $i \in \{0, 1, 2, 3\}$, and $t = -3, -2, -1, 0, 1, 2, 3, \ldots$. Put $a_t = \sum_{i=0}^{3} e_1 b_{it}$, $t = -3, -2, -1, 0, 1, 2, 3, \ldots$. Then, for $t = 4s + j$ with $j \in \{0, 1, 2, 3\}$, the state of the system $\mathcal{A}_j$ is $|a_{4s+j}\rangle_j$ ($(\mathcal{A}_j)$ has been the target of $s + (1 - \delta_{0j})$ CNOT gates).

In general for a network of $\mathbf{d}$ systems (of dimension $\mathbf{d}$), the periodic arrangement of CNOT gates with control system $\mathcal{A}_j$ and target system $\mathcal{A}_{j+1}$ for $j = 0, \ldots, \mathbf{d} - 1$ (where $j + 1 = 0$ for $j = \mathbf{d} - 1$) means that the sequence (a_t), where $|a_{\mathbf{d}s+j}\rangle_j$ is the state of system $\mathcal{A}$ at time $t = \mathbf{d}s + j$ with $j \in \{0, \ldots, \mathbf{d}-1\}$ and $\mathcal{A}_j$ has been the target of $s + (1 - \delta_{0j})$ CNOT gates, satisfies the recurrence $a_{t+\mathbf{d}} = a_{t+\mathbf{d}-1} + a_t$ for all $t \geq -\mathbf{d} + 1$, since the CNOT gate replaces the state $|a_{\mathbf{d}s+j}\rangle_j$ of $\mathcal{A}_j$ with $\left|a_{\mathbf{d}(s+1)+j}\right\rangle_j = \left|a_{\mathbf{d}s+j} + a_{\mathbf{d}(s+1)+j-1}\right\rangle_j$. With initial states $|a_{-\mathbf{d}+1}\rangle_1 = |e_1\rangle_1, \ldots, |a_{-1}\rangle_{\mathbf{d}-1} = |e_{\mathbf{d}-1}\rangle_{\mathbf{d}-1}$, $|a_0\rangle_0 = |e_0\rangle_0$ the terms of the sequence (a_t) may be written as $a_t = \sum_{i=0}^{\mathbf{d}-1} e_i b_{it}$ for sequences (b_{it}) which satisfy the recurrences $b_{i(t+\mathbf{d})} = b_{i(t+\mathbf{d}-1)} + b_{it}$, $i = 0, \ldots, \mathbf{d} - 1$. Indeed

$$\begin{aligned} a_{t+\mathbf{d}-1} + a_t &= \sum_{i=0}^{\mathbf{d}-1} e_i b_{i(t+\mathbf{d}-1)} + \sum_{i=0}^{\mathbf{d}-1} e_i b_{i(t)} \\ &= \sum_{i=0}^{\mathbf{d}-1} e_i (b_{i(t+\mathbf{d}-1)} + b_{it}) \end{aligned} \tag{7.1.1}$$

and

$$a_{t+\mathbf{d}} = \sum_{i=0}^{\mathbf{d}-1} e_i b_{i(t+\mathbf{d})}. \tag{7.1.2}$$

We also note that, for $i = 0, \ldots, \mathbf{d} - 2$ the sequence (b_{it}) is a translate of the

sequence $(b_{(i+1)t})$ by 1 place. (Indeed these sequences are all translates of one another.)

We begin by considering the solution of the recurrence relation $a_{(t+\mathbf{d})} = a_{t+\mathbf{d}-1} + a_t$. This turns out to be $a_t = \sum_{i=0}^{t/\mathbf{d}} \binom{j-(\mathbf{d}-1)}{i}$.

Lemma 4. [72] $\binom{x}{i} + \binom{x}{i+1} = \binom{x+1}{i+1}$.

Lemma 5. Let $\mathbf{d}$ be a positive integer. The sequence $\langle a_n \rangle$ defined by $a_n = \sum_{i=0}^{n/\mathbf{d}} \binom{n-(\mathbf{d}-1)i}{i}$ satisfies the recurrence relation $a_{n+\mathbf{d}} = a_{n+\mathbf{d}-1} + a_n$ with initial conditions $a_0 = \cdots = a_{\mathbf{d}-1} = 1$.

Proof: Clearly the sequence $\langle a_n \rangle$ as defined satisfies $a_0 = \cdots = a_{\mathbf{d}-1} = 1$. Let l be a non-negative integer and let $m \in \{0, \ldots, \mathbf{d}-1\}$. Then $a_{l\mathbf{d}+m} = \sum_{i=0}^{l} \binom{(l-i)\mathbf{d}+m+i}{i}$ and $a_{l\mathbf{d}+m+\mathbf{d}-1} = a_{(l+1)\mathbf{d}+m-1} = \sum_{i=0}^{l+1} \binom{(l+1-i)\mathbf{d}+m+i-1}{i}$ $= 1 + \sum_{i=1}^{l+1} \binom{(l+1-i)\mathbf{d}+m+i-1}{i} = 1 + \sum_{i=0}^{l} \binom{(l-i)\mathbf{d}+m+i}{i+1}$. Hence, by lemma 4, $a_{l\mathbf{d}+m} + a_{l\mathbf{d}+m+\mathbf{d}-1} = 1 + \sum_{i=0}^{l} \binom{(l-i)\mathbf{d}+m+i+1}{i+1} = 1 + \sum_{i=1}^{l+1} \binom{(l+1-i)\mathbf{d}+m+i}{i} = \sum_{i=0}^{l+1} \binom{(l+1-i)\mathbf{d}+m+i}{i} = a_{(l+1)\mathbf{d}+m}$ as required.

The solution sequence $\langle a_j \rangle$ is periodic since the recurrence relation is reversible and the sequence must repeat as soon as $\mathbf{d}$ consecutive terms (of which there are only finitely many possibilities) are repeated. To determine the period we need some combinatorial results.

Lemma 6. [72] $\sum_{i=0}^{k} \binom{j+i}{i} = \binom{j+k+1}{k}$ for all $j \geq 0$.

Lemma 7. Let p be prime and let $j \geq -1$. Then for $j = p-1 \pmod p$, we have $\binom{p+j}{p-1} = 1 \pmod p$ and for $j = 0, \ldots, p-2 \pmod p$, we have $\binom{p+j}{p-1} = 0 \pmod p$.

Proof: Let us write $\binom{p+j}{p-1}$ as a quotient of factorials, and consequently, we have it that $\binom{p+j}{p-1} = \frac{(p+j)(p+j-1)\ldots(p)}{(j+1)!}$. There is a multiple of p in the numerator not cancelled by a factor in the denominator except when $j = p-1 \pmod p$, and the result follows.

Theorem 22. [55] Let p be a prime number and let a, and k be any positive integers. The integer function $\binom{j}{k}$ modulo p^a for $j \geq k$ has has the cycle length p^{a+e} where $e = \lfloor \log_p k \rfloor$.

We now determine the period of the solution sequence $\langle a_j \rangle$ when $\mathbf{d}$ is prime.

Theorem 23. Let $\mathbf{d}$ be a prime and consider the sequence $\langle a_j \rangle$ defined by $a_j = \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \pmod{\mathbf{d}}$. Then $\langle a_j \rangle$ has the cycle length $\mathbf{d}^2 - 1$.

Proof: The sequence $\langle a_j \rangle$ satisfies the recurrence $a_{j+\mathbf{d}} = a_{j+\mathbf{d}-1} + a_j$ for all j. Since $a_0 = a_1 = \cdots = a_{\mathbf{d}-1} = 1$, it is sufficient to show that $a_{\mathbf{d}^2-1} = 1, a_{\mathbf{d}^2-2} = 0, \ldots, a_{\mathbf{d}^2-\mathbf{d}} = 0$. This implies that $a_{j+\mathbf{d}^2-1} = 1$ for $j = 0, \ldots, \mathbf{d}-1$, and so $a_{j+\mathbf{d}^2-1} = a_j$ for all $j \geq 0$. Now, for $j = 0, \ldots, \mathbf{d}-2$,

$$\begin{aligned} a_{j+\mathbf{d}^2-\mathbf{d}} &= \sum_{i=0}^{\frac{j+\mathbf{d}^2-\mathbf{d}}{\mathbf{d}}} \binom{j+\mathbf{d}^2-\mathbf{d}-(\mathbf{d}-1)i}{i} \\ &= \sum_{i=0}^{\mathbf{d}-1} \binom{j+i}{i}. \end{aligned} \tag{7.1.3}$$

By lemma 6,

$$\begin{aligned} \sum_{i=0}^{\mathbf{d}-1} \binom{j+i}{i} &= \binom{j+\mathbf{d}}{\mathbf{d}-1} \pmod{\mathbf{d}} \\ &= \begin{cases} 0 \bmod \mathbf{d} & \text{for } 0 \leq j < \mathbf{d}-1 \\ 1 \bmod \mathbf{d} & \text{for } j = \mathbf{d}-1. \end{cases} \end{aligned} \tag{7.1.4}$$

Thus the period divides $\mathbf{d}^2 - 1$. Next, we show that $P_a = \mathbf{d}^2 - 1$ is the smallest cycle length. For $i = 0$, $\binom{j-(\mathbf{d}-1)i}{i} = 1$ for all $j \geq 0$. Let i be such that $1 \leq I < \frac{\mathbf{d}^2-1}{\mathbf{d}}$. Then the sequence (c_j) with $c_j = \binom{j-(\mathbf{d}-1)i}{i}$ satisfies $c_j = 0$ for $j = 0, \ldots, \mathbf{d}i - 1$ and $c_j = 1$ for $j = \mathbf{d}i$ and is periodic with period $\mathbf{d}$ for $j \geq \mathbf{d}i$ by Theorem 22. Thus, $a_{l\mathbf{d}} = l + 1$ for $l = 0, \ldots, \mathbf{d}-1$. Now any $\mathbf{d}$ consecutive terms of $a_0, \ldots, a_{\mathbf{d}^2-1}$ includes a term $a_{l\mathbf{d}}$ for some l and so cannot

be $\mathbf{d}-1$ consecutive zeroes and one 1 until the terms from $\mathbf{d}^2-\mathbf{d}$ to $\mathbf{d}^2-1$. This establishes the result.

Since the cycle length $\mathbf{d}^2-1$ is coprime to $\mathbf{d}$ at the completion of a cycle the states of the system will be cycled round. Infact since $\mathbf{d}^2-1=\mathbf{d}-1=-1(\text{mod }\mathbf{d})$ they are shift by one position and the network implements the SWAP gate – system $\mathcal{A}_0$ will be in the state $|e_1\rangle_0$, the initial state of $\mathcal{A}_1$, $\mathcal{A}_1$ will be in the state $|e_2\rangle_1$, ..., $\mathcal{A}_{\mathbf{d}-1}$ will be in the state $|e_0\rangle_{\mathbf{d}-1}$. This can be seen by the following argument. Since the sequences $\langle b_{it}\rangle$, $i=0,\ldots,\mathbf{d}-1$, have cycle length $\mathbf{d}^2-1$ we have $(b_{0t},\ldots,b_{(\mathbf{d}-1)t})$ equal to: $(0,1,0,\ldots,0)$ for $t=\mathbf{d}^2-\mathbf{d}$ corresponding to system $\mathcal{A}_0$; $(0,0,1,0,\ldots,0)$ for $t=\mathbf{d}^2-\mathbf{d}+1$ corresponding to system $\mathcal{A}_1$; ...; $(0,\ldots,0,1)$ for $t=\mathbf{d}^2-2$ corresponding to system $\mathcal{A}_{\mathbf{d}-2}$; $(1,0,\ldots,0)$ for $t=\mathbf{d}^2-1$ corresponding to system $\mathcal{A}_{\mathbf{d}-1}$.

7.1.1 The case $\mathbf{d}=p^m$

Now let us consider the prime power dimension $\mathbf{d}=p^m$ with p prime and let us further consider the family of binomial summations $\sum_{i=0}^{j/p^m}\binom{j-(p^m-1)i}{i}$. We conjecture that the cycle length of $a_j=\sum_{i=0}^{j/p^m}\binom{j-(p^m-1)i}{i} \mod p^m$ is $p^{m-1}(p^{2m}-1)$. We do not have a proof of this but we have written a computer programme to calculate this value and have verified the conjecture for $\mathbf{d}=4,8$, and 9; see Table 7.1. Note that if this conjecture is true then, since $\gcd(p^m,p^{m-1}(p^{2m}-1))=p^{m-1}\neq 1$, the network does not produce a SWAP gate when $m>1$. Although at the end of a cycle the systems are shifted round, they are shifted p^{m-1} places and the systems return to their original states after p applications of the network. Thus it provides a cyclic SWAP on each of p^{m-1} groups of p systems, the systems of a group have indices congruent modulo p^{m-1}. What distinguishes this network from p^{m-1} copies of the SWAP network for $\mathbf{d}=p$ is that in this larger network the systems that

are swapped do not actually directly interact through a CNOT gate.

We now consider the problem of determining the cycle length of the sequence $\langle a_j \rangle$ for $\mathbf{d} = p^m$ $(m > 1)$. We have it that over such dimensions the sequence $\langle a_j \rangle$ satifies the recurrence $a_{j+p^m} = a_{j+p^m-1} + a_j$ for all j. Since $a_0 = a_1 = \cdots = a_{p^m-1} = 1$ by definition, it is sufficient to show that $a_{p^{m-1}(p^{2m}-1)} = 1$ and $a_{p^{m-1}(p^{2m}-1)-j}$ vanishes for $j = 1, \ldots, p^m - 1$. It follows that $a_{p^{m-1}(p^{2m}-1)+j} = 1$ for $j = 0, \ldots, p^m - 1$. Following the approach for $\mathbf{d} = p$ we find that

$$\begin{aligned}
a_{p^{m-1}(p^{2m}-1)-j} &= \sum_{i=0}^{p^{2m-1}-1} \binom{p^{m-1}(p^{2m}-1) - j - (p^m-1)i}{i} \bmod p^m \\
&= \sum_{i=0}^{p^{2m-1}-1} \binom{(p^m-1)(p^{2m-1}+p^{m-1}-i) - j}{i} \bmod p^m \\
&= \sum_{i=0}^{p^{2m-1}-1} \binom{(p^m-1)(p^{m-1}-i) - j}{i} \bmod p^m \\
&= \sum_{i=0}^{p^{2m-1}-1} \binom{(1-p^m)i - p^{m-1} - j}{i} \bmod p^m. \qquad (7.1.5)
\end{aligned}$$

We ask the question if there exists a combinatorial approach which illustrates that the sum $\sum_{i=0}^{p^{2m-1}-1} \binom{(1-p^m)i-p^{m-1}-j}{i} \bmod p^m$ equals 1 for $j = 0$ and vanishes for $j = 1, \ldots, p^m - 1$. We have not been able to find an answer but outline the approaches we have taken.

We now consider an alternative approach in evaluating the binomial summation $a_j = \sum_{i=0}^{j/p^m} \binom{j-(p^m-1)i}{i} \bmod p^m$. Let $A(z)$ be the power series $\sum_{j\geq 0} a_j z^j$ and denote by $[z^j]A(z)$ the coefficient of z^j in $A(z)$; thus $[z^j]A(z) = a_j$. We now determine the *generating function* for $A(z)$. A generating function is a clothesline on which we hang up a sequence of numbers for display [94]. In particular, the jth term of the sequence $\langle a_j \rangle$ is the coefficient of z^j in the expansion of its generating function as a power series. To find the generating function associated the sequence $\langle a_j \rangle$, we first note that the sequence $\langle a_j \rangle$

satisfies the recurrence relation

$$a_j = \begin{cases} 0 & \text{if} \quad j < 0 \\ 1 & \text{if} \quad j = 0, \ldots, p^m - 1 \\ a_{j-1} + a_{j-p^m} & \text{otherwise.} \end{cases} \tag{7.1.6}$$

This recurrence relation can be expressed as the single equation, we have

$$a_j = a_{j-1} + a_{j-p^m} + [j = 0] \tag{7.1.7}$$

where $[j = 0]$ adds 1 when $j = 0$. To demonstrate the generating function for $A(z)$, we multiply both sides of (7.1.7) by z^j and sum over j. Thus, we get

$$\begin{aligned} \sum_{j=0}^{\infty} a_j z^j &= \sum_{j=0}^{\infty} a_{j-1} z^j + \sum_{j=0}^{\infty} a_{j-p^m} z^j + \sum_{j=0}^{\infty} [j = 0] z^j \\ &= \sum_{j=0}^{\infty} a_j z^{j+1} + \sum_{j=0}^{\infty} a_j z^{j+p^m} + 1 \\ &= z \sum_{j=0}^{\infty} a_j z^j + z^{p^m} \sum_{j=0}^{\infty} a_j z^j + 1. \end{aligned} \tag{7.1.8}$$

Consequently, the generating function for $A(z)$ is given by $1/(1 - z - z^{p^m})$. Now, we can restate our problem to find a closed form for $A(z)$, and thus evaluate $[z^j]A(z)$. We use the following result from Graham *et al.* [35].

Lemma 8. [35] $\frac{1}{(1-\alpha z)^{i+1}} = \sum_{n=0}^{\infty} \binom{i+n}{i} \alpha^n z^n$.

We seek $[z^j]A(z) = [z^j]1/(1 - z - z^{p^m})$. Consider the series,

$$1/(1 - \alpha z)^{i+1} = \sum_{j=0}^{\infty} \binom{i+j}{i} \alpha^j z^j. \tag{7.1.9}$$

Consider further a finite sum of such series;

$$S(z) = \frac{\beta_1}{(1 - \alpha_1 z)^{i_1+1}} + \cdots + \frac{\beta_N}{(1 - \alpha_N z)^{i_N+1}}. \tag{7.1.10}$$

Then $[z^j]S(z)$ is the finite sum of coefficients given by

$$[z^j]S(z) = \beta_1 \binom{i_1 + j}{i_1} \alpha_1^j + \cdots + \beta_N \binom{i_N + j}{i_N} \alpha_N^j. \tag{7.1.11}$$

Now, let $A(z) = \frac{1}{B(z)}$ where $B(z) = 1 - z - z^{p^m}$. We show that $B(z)$ has distinct roots. Let us suppose that $B(z)$ has a set of repeated roots. Then, we have it that $B(z)$ and $B'(z)$ share a set of common roots [68]. Since $B'(z) = -1 - p^m(z)^{p^m-1}$, it follows that a repeated root α satisfies $B'(\alpha) = -1 - p^m(\alpha)^{p^m-1} = 0$, and consequently, $\alpha^{p^m-1} = -1/p^m$. Whence, $\alpha^{p^m} = -\alpha/p^m$. Furthermore, as $B(\alpha)$ vanishes, we have it that $1 - \alpha + \alpha/p^m = 0$, or equivalently, $1 - \alpha(1 - 1/p^m) = 0$ from which we deduce $\alpha = p^m/(p^m - 1)$ to be the only candidate for a repeating root. Therefore, $\alpha = p^m/(p^m - 1)$ should satisfy the equation $\alpha^{p^m-1} = -1/p^m$, as a consequence of being a root of $B'(z)$. That is $\frac{(p^m)^{p^m-1}}{(p^m-1)^{p^m-1}} = \frac{-1}{p^m}$. Now $(p^m)^{p^m} \equiv 0 \pmod{p}$ while $-(p^m - 1)^{p^m-1} \equiv -1 \pmod{p}$ implies that $\alpha = p^m/(p^m - 1)$ is not a root of $B'(z)$. Therefore, $B(z)$ has distinct roots.

Writing $B(z)$ in the form $(z - b_1)\dots(z - b_{p^m})$, and taking reciprocals α_k of b_k, $k = 1, \dots, p^m$, we establish a correspondence with the polynomial $(1 - \alpha_1 z)\dots(1 - \alpha_{p^m} z)$. Thus, $A(z) = 1/((1 - \alpha_1 z)\dots(1 - \alpha_{p^m} z))$ may take the form $\beta_1/(1-\alpha_1 z) + \dots + \beta_{p^m}/(1 - \alpha_{p^m} z)$, for some β_l, $l = 1, \dots, p^m$. Note $1/(1 - \alpha z)$ is a special case of Lemma 8 with $i = 0$.

Theorem 24. We claim that

$$[z^j]A(z) \;=\; \sum_{l=1}^{p^m} \beta_l \alpha_l^j, \tag{7.1.12}$$

where $\beta_l = -\alpha_l/B'(1/\alpha_l)$.

Proof:

$$\lim_{z \to 1/\alpha_l} (z - 1/\alpha_l)A(z) = \lim_{z \to 1/\alpha_l} (z - 1/\alpha_l)\mathrm{S}(z), \tag{7.1.13}$$

where $\mathrm{S}(z)$ is the special case of equation 7.1.11 with $i_l = 0$. Now, it follows

that

$$\begin{aligned}\lim_{z\to 1/\alpha_l}(z-1/\alpha_l)A(z) &= \lim_{z\to 1/\alpha_l}(z-1/\alpha_l)\frac{1}{B(z)}\\ &= \lim_{z\to 1/\alpha_l}\frac{z-1/\alpha_l}{B(z)-B(1/\alpha_l)}\\ &= \frac{1}{B'(1/\alpha_l)} \qquad (7.1.14)\end{aligned}$$

and,

$$\begin{aligned}\lim_{z\to 1/\alpha_l}(z-1/\alpha_l)\mathrm{S}(z) &= \lim_{z\to 1/\alpha_l}(z-1/\alpha_l)\sum_{k=1}^{p^m}\frac{\beta_k}{(1-\alpha_k z)}\\ &= \lim_{z\to 1/\alpha_l}\frac{\beta_l(z-1/\alpha_l)}{-\alpha_l(z-1/\alpha_l)}\\ &= \frac{\beta_l}{-\alpha_l}. \qquad (7.1.15)\end{aligned}$$

Consequently, we deduce $\beta_l = \frac{-\alpha_l}{B'(1/\alpha_l)}$, for $l = 1, \ldots, p^m$, since

$$\lim_{z\to 1/\alpha_l}(z-1/\alpha_l)\frac{\beta_k}{(1-\alpha_k z)} \qquad (7.1.16)$$

vanishes for $k \neq l$ and the result follows.

7.1.2 Examples

We now test our closed form, equation 7.1.12, against the sequence of integers arising for instances of the binomial summation $a_j = \sum_{i=0}^{j/p^m}\binom{j-(p^m-1)i}{i}$ for cases $p^m = 4$ and $p^m = 8$. We wrote a Maple program [56] to calculate the roots α_j and the coefficients β_j and the sequence $\langle a_j\rangle$. See Appendix. Two examples of the output of our programme are given below.

Example 3. Case $p^m = 4$. The reciprocals of the roots of $1 - z - z^4$ are given

as

$$\begin{aligned}
\alpha_1 &= -.8191725134, \\
\alpha_2 &= .219447421 - .9144736630\iota, \\
\alpha_3 &= .219447421 + .9144736630\iota, \\
\alpha_4 &= 1.380277569.
\end{aligned}$$

Since $\beta_l = \frac{-\alpha_l}{B'(1/\alpha_l)}$, we have

$$\begin{aligned}
\beta_1 &= .1305102698, \\
\beta_2 &= .1610008758 + .1534011260\iota, \\
\beta_3 &= .1610008758 - .1534011260\iota, \\
\beta_4 &= .5474879784.
\end{aligned}$$

Following result (7.1.12), a closed form for the binomial coefficients a_j is then given by

$$\begin{aligned}
a_j &= (.1305102698)(-.8191725134)^j \\
&\quad + (.1610 + .1534\iota)(.2194 - .9144\iota)^j \\
&\quad + (.1610 - .1534\iota)(.2194 + .9144\iota)^j \\
&\quad + (.5474879784)(1.380277569)^j. \qquad (7.1.17)
\end{aligned}$$

Maple Input: for j from 0 to 25 do; a_j; end;
Maple Output: 1,1,1,1,2,3,4,5,7,10,14,19,26,36,50,69,95,
131,181,250,345,476,657,907,1252,1728.

Example 4. Case $p^m = 8$. The reciprocals of the roots of $1 - z - z^8$ are given

as

$$\begin{aligned}
\alpha_1 &= -.9115923535, \\
\alpha_2 &= -.6157823065 - .6871957511\iota, \\
\alpha_3 &= -.6157823065 + .6871957511\iota, \\
\alpha_4 &= .1033089835 - .9564836042\iota \\
\alpha_5 &= .1033089835 + .9564836042\iota, \\
\alpha_6 &= .8522421840 - .6352622030\iota, \\
\alpha_7 &= .8522421840 + .6352622030\iota, \\
\alpha_8 &= 1.232054631. \qquad (7.1.18)
\end{aligned}$$

The set $\beta_l,\ l = 1, \ldots, 8$, is

$$\begin{aligned}
\beta_1 &= .06378010282, \\
\beta_2 &= .06449005934 + .02789285455\iota, \\
\beta_3 &= .06449005934 - .02789285455\iota, \\
\beta_4 &= .06911712233 + .06926484155\iota, \\
\beta_5 &= .06911712233 - .06926484155\iota, \\
\beta_6 &= .1188399306 + .1719523210\iota, \\
\beta_7 &= .1188399306 - .1719523210\iota, \\
\beta_8 &= .4313256714,
\end{aligned}$$

Again a closed form for the binomial coefficients a_j is given by

$$\begin{aligned} a_j &= (.06378010282)(-.9115923535)^j \\ &\quad + (.0644 + .0278\iota)(-.61578230 - .68719575\iota)^j \\ &\quad + (.0644 - .0278\iota)(-.61578230 + .68719575\iota)^j \\ &\quad + (.0691 + .0692\iota)(.10330898 - .95648360\iota)^j \\ &\quad + (.0691 - .0692\iota)(.10330898 + .95648360\iota)^j \\ &\quad + (.1188 + .1719\iota)(.85224218 - .63526220\iota)^j \\ &\quad + (.1188 - .1719\iota)(.85224218 + .63526220\iota)^j \\ &\quad + (.4313256714)(1.232054631)^j . \end{aligned} \tag{7.1.19}$$

Maple Input: for j from 0 to 25 do; a_j; end;
Maple Output:1,1,1,1,1,1,1,1,2,3,4,5,6,7,8,9,11,14,18,23,29,
36,44,53,64,78.

7.2 On the cycle length of $\langle a_j \bmod p^m \rangle$

Having found a closed form for $A(z) = \sum_{j\geq 0} a_j z^j$, we now outline the classical means by which the cycle length of $\langle a_j \rangle$ mod p^m may be determined.

The classical method: We fix a particular value of $m > 1$ and having obtained the closed form for the coefficients of the corresponding polynomial $A(z)$, we evaluate the closed form of a_j for $j = 0, \ldots, p^{m-1}(p^{2m} - 2)$. To show that $p^{m-1}(p^{2m} - 1)$ is the cycle length of $\langle a_j \rangle$ mod p^m, we show that firstly $a_{p^{m-1}(p^{2m}-1)} = 1$ and $a_{p^{m-1}(p^{2m}-1)-j}$ vanishes for $j = 1, \ldots, p^m - 1$, and lastly, we show the subsequence of p^m opening 1s does not appear until j is $p^{m-1}(p^{2m} - 1)$. While this proof follows the corresponding proof for the case $\mathbf{d} = p$, it differs in that Maple is required. Table 7.1 gives the cycle length of

the sequence $\langle a_j \rangle$ for small values of p^m. We have calculated the cycle length for all prime powers up to 3125 and they all agree with the conjecture.

We now consider the sequence $\langle a_j \rangle$ where $a_j = \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$ for composite $\mathbf{d}$ with prime factorization $\mathbf{d} = p_1^{m_1} \dots p_r^{m_r}$. We may calculate the cycle length of $\langle a_j \rangle$ modulo $\mathbf{d}$ by calculating its cycle length modulo $p_t^{m_t}$ for each $t = 1, \dots, r$. The cycle length of $\sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod p_t^{m_t}$, for $t = 1, \dots, r$, is given by the j for which $\sum_{i=1}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod p_t^{m_t}$ equals 1 and which has a preceding sequence $(a_{j-\mathbf{d}+1}, \dots, a_{j-1}) = (0, \dots, 0) \bmod p_t^{m_t}$. Given that the mapping $\lambda_{\mathbf{d},(p_1^{m_1} \dots p_r^{m_r})} : \mathbb{Z}_{\mathbf{d}} \mapsto \mathbb{Z}_{p_1^{m_1}} \times \dots \times \mathbb{Z}_{p_r^{m_r}}$ is well-defined then the cycle length $\left| \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d} \right|$ of the recurrence relation of $\langle a_j \rangle$ is given by the LCM $\left\{ \left| \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod p_t^{m_t} \right| \right\}_{t=1}^{r}$. Table 7.1 gives some initial values for the cycle length of $\sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$. We state this result as a theorem.

Theorem 25. Let $\sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$ be an integer sequence and consider the decomposition $\left\{ \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod p_t^{m_t} \right\}_{t=1}^{r}$ of $\sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$ into a direct product of disjoint cycles. Let $\left| \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d} \right|$ be the cycle length of $\sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$ be and let $\left\{ \left| \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod p_t^{m_t} \right| \right\}_{t=1}^{r}$ be the cycle lengths of $\left\{ \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod p_t^{m_t} \right\}_{t=1}^{r}$, respectively. Then, the cycle length of $\sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$ is LCM $\left\{ \left| \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod p_t^{m_t} \right| \right\}_{t=1}^{r}$.

The results provided in Table 7.1 are the cycle lengths of a_j for some small values of $\mathbf{d}$. The results are two-fold; firstly, the results give the minimum number of CNOT gates required to effectuate a quantum gate according the periodicity of the binomial summation construction. Secondly, use of the binomial summation construction illustates for which dimensions the efficient generalised SWAP gate can be realised. In particular, we have an efficient SWAP gate in those dimensions $\mathbf{d}$ for which the cycle length of the function a_j is equivalent to $-1 \bmod \mathbf{d}$. For dimensions $\mathbf{d}$ where our efficient SWAP

$\mathbf{d}$	$\left\lvert \sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d} \right\rvert$
2	$3 = 2^2 - 1$
3	$8 = 3^2 - 1$
4	$30 = 2(2^4 - 1)$
5	$24 = 5^2 - 1$
6	$\mathrm{LCM}(63, 728) = 6552$
7	$48 = 7^2 - 1$
8	$252 = 4(2^6 - 1)$
9	$240 = 3(3^4 - 1)$

Table 7.1: Cycle length of $\sum_{i=0}^{j/\mathbf{d}} \binom{j-(\mathbf{d}-1)i}{i} \bmod \mathbf{d}$.

gate is not possible then the cycle length of a_j describes a permutation, other than that permutation sought, of the $\mathbf{d}$ input qu$\mathbf{d}$it states. For example, in dimension 4 the cycle length of a_j is equivalent to 2 mod 4. Therefore, we have it that the quantum gate associated with this particular instance of the binonial summation construction evolves four 4-dimensional quantum states such that the state $|e_0\rangle_0$ of the first quantum system $\mathcal{A}_0$ is transposed with the state $|e_2\rangle_2$ of the third quantum system $\mathcal{A}_2$ so that the quantum system $\mathcal{A}_0$ is in the state $|e_2\rangle_0$ and the quantum system $\mathcal{A}_2$ is in the state $|e_0\rangle_2$. Correspondingly, the state $|e_1\rangle_1$ of the second quantum system $\mathcal{A}_1$ is transposed with the state $|e_3\rangle_3$ of the fourth quantum system $\mathcal{A}_3$ so that the quantum system $\mathcal{A}_1$ is in the state $|e_3\rangle_1$ and the quantum system $\mathcal{A}_3$ is in the state $|e_1\rangle_3$. Moreover, the gate network for an implementation of an efficient generalised SWAP$^{(l)}$ gate, for $l = 1, \ldots, \mathbf{d}-1$, is an instance of l applications of this design. Such a network cycles the initial state description through l systems much like the l-fold WilNOT gate.

Furthermore, the cycle length of the function $a_j = \sum_{i=0}^{j/6} \binom{j-5.i}{i} \bmod 6$ over dimension 6 is the least common multiple of the cycle lengths associated with the functions $\sum_{i=0}^{j/6} \binom{j-5.i}{i} \bmod 2$ and $\sum_{i=0}^{j/6} \binom{j-5.i}{i} \bmod 3$. The cycle lengths of $\sum_{i=0}^{j/6} \binom{j-5.i}{i} \bmod 2$ and $\sum_{i=0}^{j/6} \binom{j-5.i}{i} \bmod 3$ are 63 and 728, respectively. By Theorem 25, the cycle length of $\langle a_j \rangle$ mod 6 is 6552; the least common multiply of 63 and 728. However, since 6552 is equivalent to 0 mod 6, we have it that the CNOT architecture associated with the binomial summation function $\sum_{i=0}^{j/6} \binom{j-5.i}{i} \bmod 6$ acts trivally on the input state of six 6-dimensional quantum states.

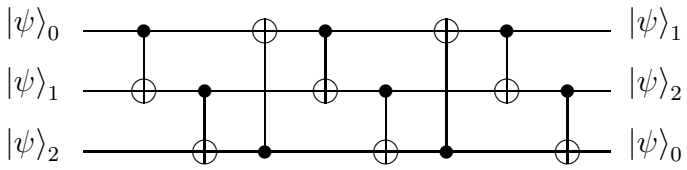


Figure 7.2: Quantum SWAP gate for qutrit states.

7.3 Efficient Qutrit SWAP

Earlier we gave an example of how the WilNOT network swapped the states of three qutrits, see section 6.2. This quantum network based on the WilNOT construction was composed of ten CNOT gates where addition was implemented modulo 3. We improve on the WilNOT construction by considering a construction based on binomial summations that yields a quantum gate which is also composed entirely from uses of the CNOT gate. This new construction is efficient in that it provides a construction of a generalised quantum SWAP gate determined by fewest uses of the CNOT gate.

Figure 7.2 gives an efficient generalised quantum SWAP gate for qutrit states which is based upon our binomial summation construction. As an instance of Theorem 20, let us suppose that the first quantum system $\mathcal{A}_0$ prepared in the state $|e_0\rangle_0$, the second system $\mathcal{A}_1$ prepared in the state $|e_1\rangle_1$ and the third system $\mathcal{A}_2$ prepared in the state $|e_2\rangle_2$. Implementing our efficient quantum SWAP gate yields the system $\mathcal{A}_0$ in the state $|e_1\rangle_0$, the system $\mathcal{A}_1$ is in the state $|e_2\rangle_1$ and the system $\mathcal{A}_2$ is in the state $|e_0\rangle_2$ and this is achieved with eight CNOT gates as aposed to the ten CNOT gates of the WilNOT construction. Note further, that our efficient construction is based on a set of binomial coefficients which satisfy the linear recurrence relation whose generating function is given by $1/(1-z-z^{p^m})$. Generating functions provide elegant

means for the storing of information about the nature of coefficients in their expansion [9]. For our purpose, the nature of the recurrence relation is seen through periodicity of the coefficients modulo 3. We relate this periodic nature to instances of the CNOT gate acting on basis states in determination of an efficient quantum swap gate. Furthermore, since our quantum gate is based on a linear recurrence relation, we illustrate that knowing part of quantum gate design will completely aid in determining the remainder of the quantum gate just as knowing the initial conditions of a recurrence relation determines all other coefficients in the recurrence. Such type of construction is unusual in that knowing part of the gate does not necessarily yield the remainder. However, Nielsen [59] has proposed a similar idea to our binomial construction. In this construction, Nielsen maintains that knowing the initial position and velocity of a geodesic, one can completely determine the remainder of the geodesic by knowing its second order differential equation. Nielsen then maintains that such an equation describes the shortest path between the identity operator and some quantum gate U. In this respect, our construction ideas are similar; quantum gates can be constructed from knowing just particular features of the gate. Our method is based on a combinatorial approach while Nielsen considers efficient method through use of geodesics.

We now illustrate the action of each use of the generalised CNOT gate arising in our efficient construction on the input quantum states and subsequent evolved states of the system. The first unitary gate U_1 acts on the input state of three arbitrary qutrits where the first quantum system $\mathcal{A}_0$ on the input state is prepared in the state $|e_0\rangle_0$, the second system $\mathcal{A}_1$ on the input state is prepared in the state $|e_1\rangle_1$ and the third system $\mathcal{A}_2$ on the input state is

prepared in the state $|e_2\rangle_2$. The action of U_1 in the input state is given by

$$
\begin{aligned}
&U_1(|a\rangle \otimes |b\rangle \otimes |c\rangle) \\
&= U_1\left((a_0|0\rangle + a_1|1\rangle + a_2|2\rangle) \otimes (b_0|0\rangle + b_1|1\rangle + b_2|2\rangle) \otimes (c_0|0\rangle + c_1|1\rangle + c_2|2\rangle)\right) \\
&= U_1(a_0b_0c_0|000\rangle + a_0b_0c_1|001\rangle + a_0b_0c_2|002\rangle + a_0b_1c_0|010\rangle + a_0b_1c_1|011\rangle \\
&+a_0b_1c_2|012\rangle + a_0b_2c_0|020\rangle + a_0b_2c_1|021\rangle + a_0b_2c_2|022\rangle + a_1b_0c_0|100\rangle
\end{aligned}
$$

$$
U_1 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}. \tag{7.3.1}
$$

$$
\begin{aligned}
&+a_1b_0c_1\,|101\rangle + a_1b_0c_2\,|102\rangle + a_1b_1c_0\,|110\rangle + a_1b_1c_1\,|111\rangle + a_1b_1c_2\,|112\rangle \\
&+a_1b_2c_0\,|120\rangle + a_1b_2c_1\,|121\rangle + a_1b_2c_2\,|122\rangle + a_2b_0c_0\,|200\rangle + a_2b_0c_1\,|201\rangle \\
&+a_2b_0c_2\,|202\rangle + a_2b_1c_0\,|210\rangle + a_2b_1c_1\,|211\rangle + a_2b_1c_2\,|212\rangle + a_2b_2c_0\,|220\rangle \\
&+a_2b_2c_1\,|221\rangle + a_2b_2c_2\,|222\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle + a_0b_1c_1\,|011\rangle \\
&+a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle + a_1b_0c_0\,|110\rangle \\
&+a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle + a_2b_0c_1\,|221\rangle \\
&+a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|210\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle\,. \qquad (7.3.2)
\end{aligned}
$$

The second unitary matrix U_2 corresponds to the second CNOT gate given in Figure 7.2 and acts on the state of the system after application of the first unitary transformation. The action of the unitary matrix U_2 on the state of the system

$$
\begin{aligned}
&U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle + a_0b_1c_1\,|011\rangle \\
&+a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle + a_1b_0c_0\,|110\rangle \\
&+a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle + a_2b_0c_1\,|221\rangle \\
&+a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|210\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle) \qquad (7.3.3)
\end{aligned}
$$

is

$$
\begin{aligned}
&U_2(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle + a_0b_1c_1\,|011\rangle \\
&+a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle + a_1b_0c_0\,|110\rangle
\end{aligned}
$$

$$U_2 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0
\end{pmatrix} \tag{7.3.4}$$

$$+a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle + a_1b_1c_2\,|122\rangle$$

$$+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle + a_2b_0c_1\,|221\rangle$$

$$+a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|210\rangle$$

$$+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle)$$

$$= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|011\rangle + a_0b_1c_1\,|012\rangle$$

$$
\begin{aligned}
&+a_0b_1c_2\,|010\rangle + a_0b_2c_0\,|022\rangle + a_0b_2c_1\,|020\rangle + a_0b_2c_2\,|021\rangle + a_1b_0c_0\,|111\rangle \\
&+a_1b_0c_1\,|112\rangle + a_1b_0c_2\,|110\rangle + a_1b_1c_0\,|122\rangle + a_1b_1c_1\,|120\rangle + a_1b_1c_2\,|121\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|222\rangle + a_2b_0c_1\,|220\rangle \\
&+a_2b_0c_2\,|221\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|211\rangle \\
&+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|210\rangle\,. \qquad (7.3.5)
\end{aligned}
$$

The unitary matrix U_3 is given as

$$
U_3 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix} \qquad (7.3.6)
$$

The third unitary matrix U_3 corresponds to the third CNOT gate given in Figure 7.2 and acts on the state of the system after application of the second

unitary and its action on the state

$$
\begin{aligned}
&U_2U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle + a_0b_1c_1\,|011\rangle \\
&+a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle + a_1b_0c_0\,|110\rangle \\
&+a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle + a_2b_0c_1\,|221\rangle \\
&+a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|210\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle)
\end{aligned}
\tag{7.3.7}
$$

is

$$
\begin{aligned}
&U_3(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|011\rangle + a_0b_1c_1\,|012\rangle \\
&+a_0b_1c_2\,|010\rangle + a_0b_2c_0\,|022\rangle + a_0b_2c_1\,|020\rangle + a_0b_2c_2\,|021\rangle + a_1b_0c_0\,|111\rangle \\
&+a_1b_0c_1\,|112\rangle + a_1b_0c_2\,|110\rangle + a_1b_1c_0\,|122\rangle + a_1b_1c_1\,|120\rangle + a_1b_1c_2\,|121\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|222\rangle + a_2b_0c_1\,|220\rangle \\
&+a_2b_0c_2\,|221\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|211\rangle \\
&+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|210\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|101\rangle + a_0b_0c_2\,|202\rangle + a_0b_1c_0\,|111\rangle + a_0b_1c_1\,|212\rangle \\
&+a_0b_1c_2\,|010\rangle + a_0b_2c_0\,|222\rangle + a_0b_2c_1\,|020\rangle + a_0b_2c_2\,|121\rangle + a_1b_0c_0\,|211\rangle \\
&+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|110\rangle + a_1b_1c_0\,|022\rangle + a_1b_1c_1\,|120\rangle + a_1b_1c_2\,|221\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|201\rangle + a_1b_2c_2\,|002\rangle + a_2b_0c_0\,|122\rangle + a_2b_0c_1\,|220\rangle \\
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|001\rangle + a_2b_1c_2\,|102\rangle + a_2b_2c_0\,|011\rangle \\
&+a_2b_2c_1\,|112\rangle + a_2b_2c_2\,|210\rangle\,.
\end{aligned}
\tag{7.3.8}
$$

As outlined, this construction is based on a linear recurrence relation and therefore there is a cyclic nature to our gate construction. This is evident when we consider our gate within the qutrit setting. As such, our construction

is based upon repeat application of the unitrary matrices U_1, U_2, and U_3. As evident in Figure 7.2 the forth gate in the construction corresponds to the unitary matrix U_1. In particular, we have it that

$$
\begin{aligned}
&U_4(U_3U_2U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle + a_0b_1c_1\,|011\rangle \\
&+a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle + a_1b_0c_0\,|110\rangle \\
&+a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle + a_2b_0c_1\,|221\rangle \\
&+a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle + a_2b_2c_0\,|210\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle)) \\
&= U_4(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|101\rangle + a_0b_0c_2\,|202\rangle + a_0b_1c_0\,|111\rangle + a_0b_1c_1\,|212\rangle \\
&+a_0b_1c_2\,|010\rangle + a_0b_2c_0\,|222\rangle + a_0b_2c_1\,|020\rangle + a_0b_2c_2\,|121\rangle + a_1b_0c_0\,|211\rangle \\
&+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|110\rangle + a_1b_1c_0\,|022\rangle + a_1b_1c_1\,|120\rangle + a_1b_1c_2\,|221\rangle \\
&+a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|201\rangle + a_1b_2c_2\,|002\rangle + a_2b_0c_0\,|122\rangle + a_2b_0c_1\,|220\rangle \\
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|001\rangle + a_2b_1c_2\,|102\rangle + a_2b_2c_0\,|011\rangle \\
&+a_2b_2c_1\,|112\rangle + a_2b_2c_2\,|210\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|111\rangle + a_0b_0c_2\,|222\rangle + a_0b_1c_0\,|121\rangle + a_0b_1c_1\,|202\rangle \\
&+a_0b_1c_2\,|010\rangle + a_0b_2c_0\,|212\rangle + a_0b_2c_1\,|020\rangle + a_0b_2c_2\,|101\rangle + a_1b_0c_0\,|201\rangle \\
&+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|120\rangle + a_1b_1c_0\,|022\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|211\rangle \\
&+a_1b_2c_0\,|110\rangle + a_1b_2c_1\,|221\rangle + a_1b_2c_2\,|002\rangle + a_2b_0c_0\,|102\rangle + a_2b_0c_1\,|210\rangle \\
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|220\rangle + a_2b_1c_1\,|001\rangle + a_2b_1c_2\,|112\rangle + a_2b_2c_0\,|011\rangle \\
&+a_2b_2c_1\,|122\rangle + a_2b_2c_2\,|200\rangle\,. \qquad (7.3.9)
\end{aligned}
$$

In a similar fashion, we have it that

$$U_5(U_4U_3U_2U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle$$

$$
\begin{aligned}
&+a_0b_1c_1\,|011\rangle + a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle \\
&+a_1b_0c_0\,|110\rangle + a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle \\
&+a_1b_1c_2\,|122\rangle + a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle \\
&+a_2b_0c_1\,|221\rangle + a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle \\
&+a_2b_2c_0\,|210\rangle + a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle))) \\
&= U_5(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|111\rangle + a_0b_0c_2\,|222\rangle + a_0b_1c_0\,|121\rangle + a_0b_1c_1\,|202\rangle \\
&+a_0b_1c_2\,|010\rangle + a_0b_2c_0\,|212\rangle + a_0b_2c_1\,|020\rangle + a_0b_2c_2\,|101\rangle + a_1b_0c_0\,|201\rangle \\
&+a_1b_0c_1\,|012\rangle + a_1b_0c_2\,|120\rangle + a_1b_1c_0\,|022\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|211\rangle \\
&+a_1b_2c_0\,|110\rangle + a_1b_2c_1\,|221\rangle + a_1b_2c_2\,|002\rangle + a_2b_0c_0\,|102\rangle + a_2b_0c_1\,|210\rangle \\
&+a_2b_0c_2\,|021\rangle + a_2b_1c_0\,|220\rangle + a_2b_1c_1\,|001\rangle + a_2b_1c_2\,|112\rangle + a_2b_2c_0\,|011\rangle \\
&+a_2b_2c_1\,|122\rangle + a_2b_2c_2\,|200\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|112\rangle + a_0b_0c_2\,|221\rangle + a_0b_1c_0\,|120\rangle + a_0b_1c_1\,|202\rangle \\
&+a_0b_1c_2\,|011\rangle + a_0b_2c_0\,|210\rangle + a_0b_2c_1\,|022\rangle + a_0b_2c_2\,|101\rangle + a_1b_0c_0\,|201\rangle \\
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|122\rangle + a_1b_1c_0\,|021\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|212\rangle \\
&+a_1b_2c_0\,|111\rangle + a_1b_2c_1\,|220\rangle + a_1b_2c_2\,|002\rangle + a_2b_0c_0\,|102\rangle + a_2b_0c_1\,|211\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|222\rangle + a_2b_1c_1\,|001\rangle + a_2b_1c_2\,|110\rangle + a_2b_2c_0\,|012\rangle \\
&+a_2b_2c_1\,|121\rangle + a_2b_2c_2\,|200\rangle \qquad (7.3.10)
\end{aligned}
$$

where the unitary matrix U_5 is given by the matrix U_2. The next unitary matrix U_6, and as illustrated in Figure 7.2, given by this construction is the matrix which corresponds to the matrix U_3. We then have it that

$$
\begin{aligned}
&U_6(U_5U_4U_3U_2U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle \\
&+a_0b_1c_1\,|011\rangle + a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle
\end{aligned}
$$

$$
\begin{aligned}
&+a_1b_0c_0\,|110\rangle + a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle \\
&+a_1b_1c_2\,|122\rangle + a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle \\
&+a_2b_0c_1\,|221\rangle + a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle \\
&+a_2b_2c_0\,|210\rangle + a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle)) \\
&= U_6(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|112\rangle + a_0b_0c_2\,|221\rangle + a_0b_1c_0\,|120\rangle + a_0b_1c_1\,|202\rangle \\
&+a_0b_1c_2\,|011\rangle + a_0b_2c_0\,|210\rangle + a_0b_2c_1\,|022\rangle + a_0b_2c_2\,|101\rangle + a_1b_0c_0\,|201\rangle \\
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|122\rangle + a_1b_1c_0\,|021\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|212\rangle \\
&+a_1b_2c_0\,|111\rangle + a_1b_2c_1\,|220\rangle + a_1b_2c_2\,|002\rangle + a_2b_0c_0\,|102\rangle + a_2b_0c_1\,|211\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|222\rangle + a_2b_1c_1\,|001\rangle + a_2b_1c_2\,|110\rangle + a_2b_2c_0\,|012\rangle \\
&+a_2b_2c_1\,|121\rangle + a_2b_2c_2\,|200\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|012\rangle + a_0b_0c_2\,|021\rangle + a_0b_1c_0\,|120\rangle + a_0b_1c_1\,|102\rangle \\
&+a_0b_1c_2\,|111\rangle + a_0b_2c_0\,|210\rangle + a_0b_2c_1\,|222\rangle + a_0b_2c_2\,|201\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|022\rangle + a_1b_1c_0\,|121\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|112\rangle \\
&+a_1b_2c_0\,|211\rangle + a_1b_2c_1\,|220\rangle + a_1b_2c_2\,|202\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|011\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|122\rangle + a_2b_1c_1\,|101\rangle + a_2b_1c_2\,|110\rangle + a_2b_2c_0\,|212\rangle \\
&+a_2b_2c_1\,|221\rangle + a_2b_2c_2\,|200\rangle\,. \qquad (7.3.11)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&U_7(U_6U_5U_4U_3U_2U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle \\
&+a_0b_1c_1\,|011\rangle + a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle \\
&+a_1b_0c_0\,|110\rangle + a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle \\
&+a_1b_1c_2\,|122\rangle + a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle \\
&+a_2b_0c_1\,|221\rangle + a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle \\
&+a_2b_2c_0\,|210\rangle + a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle)))
\end{aligned}
$$

$$
\begin{aligned}
&= U_7(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|012\rangle + a_0b_0c_2\,|021\rangle + a_0b_1c_0\,|120\rangle + a_0b_1c_1\,|102\rangle \\
&+a_0b_1c_2\,|111\rangle + a_0b_2c_0\,|210\rangle + a_0b_2c_1\,|222\rangle + a_0b_2c_2\,|201\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|022\rangle + a_1b_1c_0\,|121\rangle + a_1b_1c_1\,|100\rangle + a_1b_1c_2\,|112\rangle \\
&+a_1b_2c_0\,|211\rangle + a_1b_2c_1\,|220\rangle + a_1b_2c_2\,|202\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|011\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|122\rangle + a_2b_1c_1\,|101\rangle + a_2b_1c_2\,|110\rangle + a_2b_2c_0\,|212\rangle \\
&+a_2b_2c_1\,|221\rangle + a_2b_2c_2\,|200\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|012\rangle + a_0b_0c_2\,|021\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|112\rangle \\
&+a_0b_1c_2\,|121\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|212\rangle + a_0b_2c_2\,|221\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|022\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|110\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|222\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|011\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|111\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|220\rangle \qquad (7.3.12)
\end{aligned}
$$

where the unitary matrix U_7 is the matrix given by U_1. Our final gate as outlined in this contruction and evident in Figure 7.2 is the unitary matrix U_8. The unitary matrix U_8 corresponds to the unitary matrix U_2, and we have

$$
\begin{aligned}
&U_8(U_7U_6U_5U_4U_3U_2U_1(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|001\rangle + a_0b_0c_2\,|002\rangle + a_0b_1c_0\,|010\rangle \\
&+a_0b_1c_1\,|011\rangle + a_0b_1c_2\,|012\rangle + a_0b_2c_0\,|020\rangle + a_0b_2c_1\,|021\rangle + a_0b_2c_2\,|022\rangle \\
&+a_1b_0c_0\,|110\rangle + a_1b_0c_1\,|111\rangle + a_1b_0c_2\,|112\rangle + a_1b_1c_0\,|120\rangle + a_1b_1c_1\,|121\rangle \\
&+a_1b_1c_2\,|122\rangle + a_1b_2c_0\,|100\rangle + a_1b_2c_1\,|101\rangle + a_1b_2c_2\,|102\rangle + a_2b_0c_0\,|220\rangle \\
&+a_2b_0c_1\,|221\rangle + a_2b_0c_2\,|222\rangle + a_2b_1c_0\,|200\rangle + a_2b_1c_1\,|201\rangle + a_2b_1c_2\,|202\rangle \\
&+a_2b_2c_0\,|210\rangle + a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|212\rangle)) \\
&= U_8(a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|012\rangle + a_0b_0c_2\,|021\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|112\rangle \\
&+a_0b_1c_2\,|121\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|212\rangle + a_0b_2c_2\,|221\rangle + a_1b_0c_0\,|001\rangle
\end{aligned}
$$

$$
\begin{aligned}
&+a_1b_0c_1\,|010\rangle + a_1b_0c_2\,|022\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|110\rangle + a_1b_1c_2\,|122\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|210\rangle + a_1b_2c_2\,|222\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|011\rangle \\
&+a_2b_0c_2\,|020\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|111\rangle + a_2b_1c_2\,|120\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|211\rangle + a_2b_2c_2\,|220\rangle) \\
&= a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|010\rangle + a_0b_0c_2\,|020\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|110\rangle \\
&+a_0b_1c_2\,|120\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|210\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|011\rangle + a_1b_0c_2\,|021\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|111\rangle + a_1b_1c_2\,|121\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|211\rangle + a_1b_2c_2\,|221\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|012\rangle \\
&+a_2b_0c_2\,|022\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|112\rangle + a_2b_1c_2\,|122\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|222\rangle\,. \qquad (7.3.13)
\end{aligned}
$$

Now, the state of the system after application of the final unitary matrix U_8 may be written

$$
\begin{aligned}
&a_0b_0c_0\,|000\rangle + a_0b_0c_1\,|010\rangle + a_0b_0c_2\,|020\rangle + a_0b_1c_0\,|100\rangle + a_0b_1c_1\,|110\rangle \\
&+a_0b_1c_2\,|120\rangle + a_0b_2c_0\,|200\rangle + a_0b_2c_1\,|210\rangle + a_0b_2c_2\,|220\rangle + a_1b_0c_0\,|001\rangle \\
&+a_1b_0c_1\,|011\rangle + a_1b_0c_2\,|021\rangle + a_1b_1c_0\,|101\rangle + a_1b_1c_1\,|111\rangle + a_1b_1c_2\,|121\rangle \\
&+a_1b_2c_0\,|201\rangle + a_1b_2c_1\,|211\rangle + a_1b_2c_2\,|221\rangle + a_2b_0c_0\,|002\rangle + a_2b_0c_1\,|012\rangle \\
&+a_2b_0c_2\,|022\rangle + a_2b_1c_0\,|102\rangle + a_2b_1c_1\,|112\rangle + a_2b_1c_2\,|122\rangle + a_2b_2c_0\,|202\rangle \\
&+a_2b_2c_1\,|212\rangle + a_2b_2c_2\,|222\rangle \\
&= b_0c_0a_0\,|000\rangle + b_0c_0a_1\,|001\rangle + b_0c_0a_2\,|002\rangle + b_0c_1a_0\,|010\rangle + b_0c_1a_1\,|011\rangle \\
&+b_0c_1a_2\,|012\rangle + b_0c_2a_0\,|020\rangle + b_0c_2a_1\,|021\rangle + b_0c_2a_2\,|022\rangle + b_1c_0a_0\,|100\rangle \\
&+b_1c_0a_1\,|101\rangle + b_1c_0a_2\,|102\rangle + b_1c_1a_0\,|110\rangle + b_1c_1a_1\,|111\rangle + b_1c_1a_2\,|112\rangle \\
&+b_1c_2a_0\,|121\rangle + b_1c_2a_1\,|121\rangle + b_1c_2a_2\,|122\rangle + b_2c_0a_0\,|200\rangle + b_2c_0a_1\,|201\rangle
\end{aligned}
$$

$$+b_2c_0a_2\,|202\rangle + b_2c_1a_0\,|210\rangle + b_2c_1a_1\,|211\rangle + b_2c_1a_2\,|212\rangle + b_2c_2a_0\,|220\rangle$$
$$+b_2c_2a_1\,|221\rangle + b_2c_2a_2\,|222\rangle\,. \tag{7.3.14}$$

In particular, we note that state 7.3.14 has the form

$$(b_0\,|0\rangle + b_1\,|1\rangle + b_2\,|2\rangle) \otimes (c_0\,|0\rangle + c_1\,|1\rangle + c_2\,|2\rangle) \otimes (a_0\,|0\rangle + a_1\,|1\rangle + a_2\,|2\rangle)$$
$$= |b\rangle \otimes |c\rangle \otimes |a\rangle\,, \tag{7.3.15}$$

$$\text{SWAP} = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{pmatrix} \tag{7.3.16}$$

thereby illustrating our binomial summation construction to be a quantum SWAP gate.

Chapter 8

Examples of Qudit Codes

8.1 The Qudit Shor Code

The Shor code for a qu**d**it state is a generalisation of Shor's original quantum code which protects codewords from the effects of generalised qu**d**it and phase flips as described in the error model (2.2.4) for **d**-dimensional quantum states. We encode the superposition state of a qu**d**it according to the error mapping given by

$$\sum_{i=0}^{\mathbf{d}-1} \alpha_i \left|i\right\rangle \mapsto \frac{1}{\mathbf{d}^{3/2}} \sum_{i=0}^{\mathbf{d}-1} \alpha_i \left(\left(\sum_{j=0}^{\mathbf{d}-1} \omega^{ij} \left|jjj\right\rangle \right)^{\otimes 3} \right). \tag{8.1.1}$$

The encoding transformation (8.1.1) is a composition of two distinct transformations which encode the qu**d**it state against phase errors and qu**d**it flip errors. The quantum circuit associated with the encoding transformation is represented in the initial section of Figure 8.1. Recovery of the information state from the effects of the environment is achieved by processing the received state through the decoding circuit as illustrated in the latter part of Figure 8.1. The decoding scheme is outlined in two stages; firstly, the decoding circuit reverses the order the the gate operations as they appear in the encoding circuit and introduces a set of inverse Toffoli gates. The Toffoli gate [62, 86]

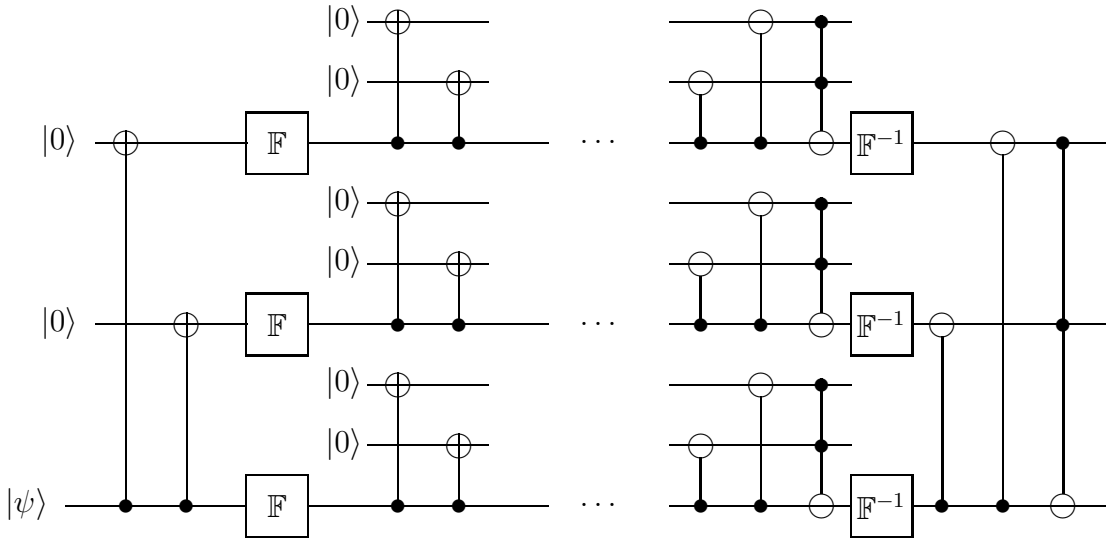


Figure 8.1: Encoding and decoding circuit for the Shor qu**d**it code.

has three input values (a, b, c). The values a and b are the control elements while c is the target element. The action of Toffoli gate on (a, b, c) is given by $(a, b, c \oplus ab)$ which therefore leaves both control elements unchanged while the target element is set to $c \oplus ab$.

Secondly, the controlled operations of the decoding circuit are given by X_j^{-1} for the control qu**d**it $|j\rangle$. Suppose the error $E_{k,l}$, for some $(k, l) \in \mathbb{Z}_{\mathbf{d}} \times \mathbb{Z}_{\mathbf{d}}$, occurs on the third cluster of qu**d**its and on the last qu**d**it. The state of the encoded qu**d**it (8.1.1) is then written as

$$\frac{1}{\mathbf{d}^{3/2}} \sum_{i=0}^{\mathbf{d}-1} \alpha_i \left(\left(\sum_{j=0}^{\mathbf{d}-1} \omega^{ij} \, |jjj\rangle \right)^{\otimes 2} \left(\sum_{j=0}^{\mathbf{d}-1} \omega^{i(j+l)} \, |jjj+k\rangle \right) \right). \tag{8.1.2}$$

The state (8.1.2) is then presented for decoding whereby we calculate successive states arising from the state (8.1.2) input to the decoding circuit of Figure

8.1. In particular,

$$\begin{aligned}
&\frac{1}{\mathbf{d}^{3/2}}\sum_{i=0}^{\mathbf{d}-1}\alpha_i\left(\left(\sum_{j=0}^{\mathbf{d}-1}\omega^{ij}\left|j0j\right\rangle\right)^{\otimes 2}\left(\sum_{j=0}^{\mathbf{d}-1}\omega^{i(j+l)}\left|j0j+k\right\rangle\right)\right)\\
&\mapsto\frac{1}{\mathbf{d}^{3/2}}\sum_{i=0}^{\mathbf{d}-1}\alpha_i\left(\left(\sum_{j=0}^{\mathbf{d}-1}\omega^{ij}\left|j00\right\rangle\right)^{\otimes 2}\left(\sum_{j=0}^{\mathbf{d}-1}\omega^{i(j+l)}\left|j0k\right\rangle\right)\right)\\
&\mapsto\frac{1}{\mathbf{d}^{3/2}}\sum_{i=0}^{\mathbf{d}-1}\alpha_i\left(\left(\sum_{j=0}^{\mathbf{d}-1}\omega^{ij}\left|j00\right\rangle\right)^{\otimes 2}\left(\sum_{j=0}^{\mathbf{d}-1}\omega^{i(j+l)}\left|j0k\right\rangle\right)\right)\\
&\mapsto\sum_{i=0}^{\mathbf{d}-1}\alpha_i\Big(\left|i00\right\rangle\left|i00\right\rangle\left|i+l0k\right\rangle\Big). \qquad (8.1.3)
\end{aligned}$$

Following application of the inverse Fourier transform on each cluster, we have it that the state of the system is given as $\sum_{i=0}^{\mathbf{d}-1}\alpha_i\left(\left|i00\right\rangle\left|i00\right\rangle\left|i+l0k\right\rangle\right)$. For the error analysis that follows the completion of the decoding stage, we note that the state of the system $\sum_{i=0}^{\mathbf{d}-1}\alpha_i\left(\left|i00\right\rangle\left|i00\right\rangle\left|i+l0k\right\rangle\right)$ may be written as $\sum_{i=0}^{\mathbf{d}-1}\alpha_i\left(\left|i\right\rangle\left|00\right\rangle\left|i\right\rangle\left|00\right\rangle\left|i+l\right\rangle\left|0k\right\rangle\right)$. Completing the decoding stage, we have it that

$$\begin{aligned}
&\sum_{i=0}^{\mathbf{d}-1}\alpha_i\Big(\left|i\right\rangle\left|00\right\rangle\left|i\right\rangle\left|00\right\rangle\left|i+l\right\rangle\left|0k\right\rangle\Big)\\
&\quad\mapsto\sum_{i=0}^{\mathbf{d}-1}\alpha_i\Big(\left|i\right\rangle\left|00\right\rangle\left|0\right\rangle\left|00\right\rangle\left|i+l\right\rangle\left|0k\right\rangle\Big)\\
&\quad\mapsto\sum_{i=0}^{\mathbf{d}-1}\alpha_i\Big(\left|i\right\rangle\left|00\right\rangle\left|0\right\rangle\left|00\right\rangle\left|l\right\rangle\left|0k\right\rangle\Big)\\
&\quad\mapsto\sum_{i=0}^{\mathbf{d}-1}\alpha_i\Big(\left|i\right\rangle\left|00\right\rangle\left|0\right\rangle\left|00\right\rangle\left|l\right\rangle\left|0k\right\rangle\Big). \qquad (8.1.4)
\end{aligned}$$

The syndrome representing the location and magnitude of the qu**d**it and phase combinations of error is given by the final eight qu**d**its of outcome (8.1.4). The set of qu**d**it pairs in the syndrome relate to the qu**d**it bit error while the final

couple of single qu**d**it states correspond to the phase error. Should a qu**d**it error occur in any of the three clusters then the corresponding pair of syndrome qu**d**its will differ from the all zero syndrome. Similarly, a phase error on a cluster of qu**d**its reveals that the pair of single qu**d**it differs from the all zero pair. There are three cases to consider to detect and correct the combinations of qu**d**it and phase error. Firstly, the syndrome pair $|0k\rangle$, for $k \in \mathbb{Z}_{\mathbf{d}}$, illustrates that a qu**d**it error occurs in position three of the corrupted cluster of qu**d**its and can be corrected by applying X_k^{-1} on that qu**d**it. Similarly, a phase error occurs on the third cluster of qu**d**its if the states of the single qu**d**its are $|0\rangle\,|l\rangle$, for $l \in \mathbb{Z}_{\mathbf{d}}$, with correction on that cluster given by Z_l^{-1}. On the other hand, the syndrome pair $|k0\rangle$ would indicate a qu**d**it error lies in position two of the corrupted cluster while the pair of single states given by $|l\rangle\,|0\rangle$ reveal a phase error on the second cluster. Correction of the corrupted state is implemented by applying X_k^{-1} on the corrupted qu**d**it and by applying Z_l^{-1} on corrupted cluster. Finally, the syndrome pair $|kk\rangle$ illustrates that a qu**d**it error occurs in position one of the corrupted cluster while the pair of single state $|l\rangle\,|l\rangle$ reveals a phase error in the first cluster on encoded qu**d**its. Applying X_k and Z_l on the corrupted qu**d**it or cluster corrects the error associated with $|kk\rangle$ and $|l\rangle\,|l\rangle$ respectively.

8.2 Qudit Stabilizer Codes

The codewords associated with a quantum code $\mathcal{Q}$ are given by $|\overline{c_1 c_2}\ldots\overline{c_k}\rangle = \overline{X}_1^{c_1}\overline{X}_2^{c_2}\ldots\overline{X}_k^{c_k}\sum_{\mathcal{M}\in S}\mathcal{M}\,|000000000\rangle$. A stabilizer matrix for the quantum $[\![5,1,3]\!]_3$ code $\mathcal{Q}$ is represented in Table 8.1. The basis codewords for this code

$\mathcal{M}_1$	X_1	Z_1	Z_2	X_2	I
$\mathcal{M}_2$	I	X_1	Z_1	Z_2	X_2
$\mathcal{M}_3$	X_2	I	X_1	Z_1	Z_2
$\mathcal{M}_4$	Z_2	X_2	I	X_1	Z_1
$\overline{X}_1$	Z_1	I	I	Z_1	X_1

Table 8.1: The stabilizer for a five qutrit code.

are given by

$$\begin{aligned} \left|\overline{0}\right\rangle &= \sum_{M\in\mathcal{S}} \mathcal{M}\left|00000\right\rangle \\ \left|\overline{1}\right\rangle &= \overline{X}\left|\overline{0}\right\rangle \\ \left|\overline{2}\right\rangle &= \overline{X}^2\left|\overline{0}\right\rangle . \end{aligned} \tag{8.2.1}$$

Hence,

$$\begin{aligned} \left|\overline{0}\right\rangle &= \left|00000\right\rangle + \mathcal{M}_1\left|00000\right\rangle + \mathcal{M}_2\left|00000\right\rangle + \mathcal{M}_3\left|00000\right\rangle + \mathcal{M}_4\left|00000\right\rangle \\ &+ \mathcal{M}_1^2\left|00000\right\rangle + \mathcal{M}_2^2\left|00000\right\rangle + \mathcal{M}_3^2\left|00000\right\rangle + \mathcal{M}_4^2\left|00000\right\rangle \\ &+ \mathcal{M}_1\mathcal{M}_2\left|00000\right\rangle + \mathcal{M}_1\mathcal{M}_2^2\left|00000\right\rangle + \mathcal{M}_1^2\mathcal{M}_2\left|00000\right\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\left|00000\right\rangle \\ &+ \mathcal{M}_1\mathcal{M}_3\left|00000\right\rangle + \mathcal{M}_1\mathcal{M}_3^2\left|00000\right\rangle + \mathcal{M}_1^2\mathcal{M}_3\left|00000\right\rangle + \mathcal{M}_1^2\mathcal{M}_3^2\left|00000\right\rangle \\ &+ \mathcal{M}_1\mathcal{M}_4\left|00000\right\rangle + \mathcal{M}_1\mathcal{M}_4^2\left|00000\right\rangle + \mathcal{M}_1^2\mathcal{M}_4\left|00000\right\rangle + \mathcal{M}_1^2\mathcal{M}_4^2\left|00000\right\rangle \\ &+ \mathcal{M}_2\mathcal{M}_3\left|00000\right\rangle + \mathcal{M}_2\mathcal{M}_3^2\left|00000\right\rangle + \mathcal{M}_2^2\mathcal{M}_3\left|00000\right\rangle + \mathcal{M}_2^2\mathcal{M}_3^2\left|00000\right\rangle \\ &+ \mathcal{M}_2\mathcal{M}_4\left|00000\right\rangle + \mathcal{M}_2\mathcal{M}_4^2\left|00000\right\rangle + \mathcal{M}_2^2\mathcal{M}_4\left|00000\right\rangle + \mathcal{M}_2^2\mathcal{M}_4^2\left|00000\right\rangle \\ &+ \mathcal{M}_3\mathcal{M}_4\left|00000\right\rangle + \mathcal{M}_3\mathcal{M}_4^2\left|00000\right\rangle + \mathcal{M}_3^2\mathcal{M}_4\left|00000\right\rangle + \mathcal{M}_3^2\mathcal{M}_4^2\left|00000\right\rangle \\ &+ \mathcal{M}_1\mathcal{M}_2\mathcal{M}_3\left|00000\right\rangle + \mathcal{M}_1\mathcal{M}_2\mathcal{M}_3^2\left|00000\right\rangle + \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_3\left|00000\right\rangle \end{aligned}$$

$$
\begin{aligned}
&+ \quad \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_3\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_3\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_3^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_3^2\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_3^2\,|00000\rangle \\
&+ \quad \mathcal{M}_2\mathcal{M}_3\mathcal{M}_4\,|00000\rangle + \mathcal{M}_2\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_2\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle \\
&+ \quad \mathcal{M}_2^2\mathcal{M}_3\mathcal{M}_4\,|00000\rangle + \mathcal{M}_2^2\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_2^2\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_2\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_2^2\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1\mathcal{M}_2\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_4\,|00000\rangle \\
&+ \quad \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_3\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_1\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle \\
&+ \quad \mathcal{M}_1^2\mathcal{M}_3\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2\mathcal{M}_3\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1\mathcal{M}_2\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_3\mathcal{M}_4\,|00000\rangle \\
&+ \quad \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_3\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_3\mathcal{M}_4\,|00000\rangle \\
&+ \quad \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_1\mathcal{M}_2\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_3^2\mathcal{M}_4\,|00000\rangle \\
&+ \quad \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_3\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle \\
&+ \quad \mathcal{M}_1\mathcal{M}_2^2\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle + \mathcal{M}_1^2\mathcal{M}_2^2\mathcal{M}_3^2\mathcal{M}_4^2\,|00000\rangle \qquad (8.2.2) \\
&= \quad |00000\rangle + |10020\rangle + |01002\rangle + |20100\rangle + |02010\rangle \\
&+ \quad |20010\rangle + |02001\rangle + |10200\rangle + |01020\rangle + \omega\,|11022\rangle \\
&+ \quad \omega^2\,|12021\rangle + \omega^2\,|21012\rangle + \omega\,|22011\rangle + \omega^2\,|00120\rangle
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad \omega\,|20220\rangle + \omega\,|10110\rangle + \omega\,|20220\rangle + \omega^2\,|12000\rangle \\
&+ \quad \omega\,|11010\rangle + \omega\,|22020\rangle + \omega^2\,|21000\rangle + \omega\,|21102\rangle + \omega^2\,|11202\rangle \\
&+ \quad \omega^2\,|22101\rangle + \omega\,|12201\rangle + \omega^2\,|00012\rangle + \omega\,|02022\rangle + \omega\,|02011\rangle \\
&+ \quad \omega^2\,|00021\rangle + \omega\,|22110\rangle + \omega^2\,|21120\rangle + \omega^2\,|12210\rangle + \omega\,|11220\rangle \\
&+ \quad \omega\,|01122\rangle + \omega\,|21222\rangle + \omega^2\,|02121\rangle + \omega\,|11112\rangle + \omega\,|12111\rangle \\
&+ \quad |01212\rangle + \omega\,|22221\rangle + \omega\,|02210\rangle + \omega\,|22110\rangle + \omega\,|21122\rangle \\
&+ \quad \omega^2\,|10212\rangle + \omega\,|21111\rangle + \omega\,|11211\rangle + |20121\rangle + \omega\,|12222\rangle \\
&+ \quad |10221\rangle + \omega^2\,|10002\rangle + |12012\rangle + \omega^2\,|11001\rangle + \omega^2\,|20022\rangle \\
&+ \quad |21021\rangle + \omega^2\,|22002\rangle + \omega^2\,|20011\rangle + \omega^2\,|00001\rangle + \omega\,|01100\rangle \\
&+ \quad \omega^2\,|01110\rangle + \omega^2\,|22200\rangle + \omega\,|12120\rangle + \omega^2\,|02220\rangle + \omega^2\,|11100\rangle \\
&+ \quad \omega^2\,|21210\rangle + \omega^2\,|01200\rangle + |00100\rangle + \omega\,|01112\rangle + \omega^2\,|10212\rangle \\
&+ \quad \omega\,|21111\rangle + \omega^2\,|10122\rangle + \omega\,|11121\rangle + \omega^2\,|00222\rangle + |12102\rangle \\
&+ \quad \omega^2\,|00111\rangle + \omega\,|22212\rangle + |21201\rangle + \omega\,|01221\rangle + \omega\,|10101\rangle \\
&+ \quad \omega\,|02202\rangle + \omega\,|22211\rangle + |00201\rangle\,. \qquad (8.2.3)
\end{aligned}
$$

$$
\begin{aligned}
\left|\overline{1}\right\rangle \quad &= \quad |00001\rangle + |10021\rangle + |01000\rangle + \omega^2\,|20101\rangle + \omega\,|02011\rangle \\
&+ \quad |20011\rangle + |02002\rangle + \omega\,|10201\rangle + \omega^2\,|01021\rangle + \omega\,|11020\rangle \\
&+ \quad \omega^2\,|12022\rangle + \omega^2\,|21010\rangle + \omega\,|22012\rangle + \omega\,|00121\rangle \\
&+ \quad \omega^2\,|20221\rangle + |10111\rangle + \omega^2\,|20221\rangle + |12001\rangle \\
&+ \quad |11011\rangle + \omega^2\,|22021\rangle + \omega\,|21001\rangle + |21100\rangle + |11200\rangle \\
&+ \quad \omega\,|22102\rangle + \omega^2\,|12202\rangle + |00010\rangle + |02020\rangle + \omega^2\,|02012\rangle \\
&+ \quad \omega\,|00022\rangle + \omega\,|22111\rangle + |21121\rangle + \omega\,|12211\rangle + \omega\,|11221\rangle \\
&+ \quad |01120\rangle + \omega^2\,|21220\rangle + \omega\,|02122\rangle + |11110\rangle + |12112\rangle \\
&+ \quad \omega\,|01210\rangle + \omega^2\,|22222\rangle + \omega\,|02211\rangle + \omega\,|22111\rangle + \omega^2\,|21120\rangle
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad \omega\,|10210\rangle + \omega\,|21112\rangle + |11212\rangle + \omega\,|20122\rangle + \omega\,|12220\rangle \\
&+ \quad |10222\rangle + |10000\rangle + \omega^2\,|12010\rangle + |11002\rangle + |20020\rangle \\
&+ \quad \omega\,|21022\rangle + \omega\,|22000\rangle + \omega^2\,|20012\rangle + \omega^2\,|00002\rangle + \omega\,|01101\rangle \\
&+ \quad |01111\rangle + \omega\,|22201\rangle + \omega\,|12121\rangle + \omega\,|02221\rangle + |11101\rangle \\
&+ \quad \omega^2\,|21212\rangle + \omega^2\,|01201\rangle + |00101\rangle + \omega^2\,|01110\rangle + \omega\,|10210\rangle \\
&+ \quad \omega\,|21112\rangle + \omega^2\,|10120\rangle + \omega\,|11122\rangle + \omega\,|00220\rangle + \omega\,|12100\rangle \\
&+ \quad |00112\rangle + \omega\,|22210\rangle + \omega^2\,|21202\rangle + |01222\rangle + \omega\,|10102\rangle \\
&+ \quad \omega\,|02200\rangle + \omega\,|22212\rangle + |00202\rangle\,. \qquad (8.2.4)
\end{aligned}
$$

$$
\begin{aligned}
\left|\overline{2}\right\rangle \;=\;& |00002\rangle + |10022\rangle + |01001\rangle + \omega\,|20102\rangle + \omega^2\,|02012\rangle \\
&+ \quad |20012\rangle + |02000\rangle + \omega^2\,|10202\rangle + \omega\,|01022\rangle + \omega\,|11021\rangle \\
&+ \quad \omega^2\,|12020\rangle + \omega^2\,|21011\rangle + \omega\,|22010\rangle + |00122\rangle \\
&+ \quad |20222\rangle + \omega^2\,|10112\rangle + |20222\rangle + \omega\,|12002\rangle \\
&+ \quad \omega^2\,|11012\rangle + |22022\rangle + |21002\rangle + \omega^2\,|21101\rangle + \omega\,|11201\rangle \\
&+ \quad |22100\rangle + |12200\rangle + \omega\,|00011\rangle + \omega^2\,|02021\rangle + |02010\rangle \\
&+ \quad |00020\rangle + \omega\,|22112\rangle + \omega\,|21122\rangle + |12212\rangle + \omega\,|11222\rangle \\
&+ \quad \omega^2\,|01121\rangle + |21221\rangle + |02120\rangle + \omega^2\,|11111\rangle + \omega^2\,|12110\rangle \\
&+ \quad \omega^2\,|01211\rangle + |22220\rangle + |02212\rangle + \omega\,|22112\rangle + \omega^2\,|21121\rangle \\
&+ \quad |10211\rangle + \omega\,|21110\rangle + \omega^2\,|11210\rangle + \omega\,|20120\rangle + \omega\,|12221\rangle \\
&+ \quad |10220\rangle + \omega\,|10001\rangle + \omega\,|12011\rangle + \omega\,|11000\rangle + \omega\,|20021\rangle \\
&+ \quad \omega\,|21020\rangle + |22001\rangle + \omega^2\,|20010\rangle + \omega^2\,|00000\rangle + \omega\,|01102\rangle \\
&+ \quad \omega\,|01112\rangle + |22202\rangle + \omega\,|12122\rangle + |02222\rangle + \omega\,|11102\rangle \\
&+ \quad \omega^2\,|21212\rangle + \omega^2\,|01202\rangle + |00102\rangle + |01111\rangle + |10211\rangle
\end{aligned}
$$

$\mathcal{M}_1$	X_1	I	Z_2	$Y_{2,1}$	$Y_{1,2}$	$Y_{2,2}$	X_2	Z_1	$Y_{1,1}$
$\mathcal{M}_2$	I	X_1	$Y_{1,1}$	$Y_{2,2}$	Z_1	$Y_{2,1}$	X_2	$Y_{1,2}$	Z_2
$\mathcal{M}_3$	Z_1	I	$Y_{2,2}$	X_1	X_2	$Y_{2,1}$	Z_2	$Y_{1,1}$	$Y_{1,2}$
$\mathcal{M}_4$	I	Z_1	$Y_{1,2}$	$Y_{2,1}$	$Y_{1,1}$	X_1	Z_2	X_2	$Y_{2,2}$
$\overline{X}_1$	X_1	X_2	I	X_2	X_1	I	I	I	I
$\overline{X}_2$	X_2	X_1	I	X_2	I	X_1	I	I	I
$\overline{X}_3$	X_1	I	X_2	X_2	I	I	X_1	I	I
$\overline{X}_4$	X_2	I	X_1	X_2	I	I	I	X_1	I
$\overline{X}_5$	I	X_2	X_1	X_2	I	I	I	I	X_1

Table 8.2: The stabilizer for a nine qutrit code.

$$
\begin{aligned}
&+ \quad \omega\,|21110\rangle + \omega^2\,|10121\rangle + \omega\,|11120\rangle + |00221\rangle + \omega^2\,|12101\rangle \\
&+ \quad \omega\,|00110\rangle + \omega\,|22211\rangle + \omega\,|21200\rangle + \omega^2\,|01220\rangle + |10100\rangle \\
&+ \quad \omega\,|02201\rangle + \omega\,|22210\rangle + |00200\rangle\,. \qquad (8.2.5)
\end{aligned}
$$

A quantum code $\mathcal{Q}$ that encodes five qutrits in nine qutrits is represented in Table 8.2 along with the set of generators representing $\mathcal{N}(\mathcal{S}) - \mathcal{S}$.

8.3 Constructing the Normaliser $\mathcal{N}(\mathcal{S})$

It is well-documented [36, 60] that the stabilizer $\mathcal{S}$ of an $[\![n, k, d]\!]$ quantum code $\mathcal{Q}$ is an Abelian subgroup of the error group $\mathcal{G}^n$. Whence, a stabilizer code may be viewed as the common eigenspace of a minimal set of generators. We have shown that all elements $E_{i,j} \in \mathcal{G}^{\otimes n}$ can be written as $E_{i,j} = \omega^{\alpha}(X_{i_1}Z_{j_1}) \otimes (X_{i_2}Z_{j_2}) \otimes \cdots \otimes (X_{i_n}Z_{j_n})$, $\alpha \in \mathbb{Z}_{\mathbf{d}}$ and $((i_1, j_1), (i_2, j_2), \ldots, (i_n, j_n)) \in \mathbb{Z}_{\mathbf{d}}^n \times \mathbb{Z}_{\mathbf{d}}^n$. Furthermore, we have it that $\mathcal{G}^n$ contains elements within the normaliser $\mathcal{N}(\mathcal{S})$ that maintain the common eigenspace of a quantum stabilizer code $\mathcal{Q}$ but

which are not elements of the stabilizer $\mathcal{S}$. This is a troublesome fact for a coding theorist since codewords may be corrupted by errors that lie within $\mathcal{N}(\mathcal{S})$. Such corruption cannot be corrected or detected but instead is past on under the guise of some other codeword. The encoded information is lost and the coding theorist is caught unawares and unknowing.

While elements of $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ act non-trivially within the common eigenspace of the coding space $\mathcal{C}(\mathcal{S})$, $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ permits the full description of the set of quantum codewords. The $2k$ elements of $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ are described by the sets $\overline{X_i}$ and $\overline{Z_i}$ where $i = 1, \ldots, k$; see section 4.5. Lifting an $[\![n, k, d]\!]_{\mathbf{d}}$ quantum code to higher dimensions requires that the set $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ be determined according to conditions;

1. $\overline{X}_i$ and $\overline{Z}_j$ form independent and commuting sets that commute with with each $\mathcal{M} \in \mathcal{S}$.

2. $\overline{X}_i$ commutes with $\overline{Z}_j$ for $i \neq j$.

3. $\overline{X}_i$ does not commute with $\overline{Z}_j$ for $i = j$.

The following theorem determines commutativity between basis elements of $\mathbb{C}^{\mathbf{d}}$ which is then extended to the n-qu**d**it setting.

Theorem 26. The operators $E_{i,j}$ and $E_{k,l}$ commute if and only if $\sum_{z=1}^{n} (j_z k_z - i_z l_z)$ vanishes.

Proof: Commutativity between the error operators $E_{i,j}$ and $E_{k,l}$ is established following conjugation of $E_{i,j}$ with $E_{k,l}$. In particular, we have it that

$$\begin{aligned}
&(X_{i_z} Z_{j_z}) (X_{k_z} Z_{l_z}) (X_{i_z} Z_{j_z})^{-1} \\
&\quad = \quad (\sum_{x=0}^{\mathbf{d}-1} \omega^{j_z x} |x + i_z\rangle \langle x|)(\sum_{y=0}^{\mathbf{d}-1} \omega^{l_z y} |y + k_z\rangle \langle y|)(\sum_{z=0}^{\mathbf{d}-1} \omega^{j_z z} |z + i_z\rangle \langle z|)^{-1}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{x=0}^{\mathbf{d}-1} \omega^{j_z x} |x+i_z\rangle\langle x| \sum_{y=0}^{\mathbf{d}-1} \omega^{l_z y} |y+k_z\rangle\langle y| \sum_{z=0}^{\mathbf{d}-1} \omega^{-j_z z} |z\rangle\langle z+i_z| \\
&= \sum_{x=0}^{\mathbf{d}-1} \omega^{j_z x} |x+i_z\rangle\langle x| \sum_{y,z=0}^{\mathbf{d}-1} \omega^{l_z y}\omega^{-j_z z} |y+k_z\rangle\langle y|z\rangle\langle z+i_z| \\
&= \sum_{x=0}^{\mathbf{d}-1} \omega^{j_z x} |x+i_z\rangle\langle x| \sum_{y=0}^{\mathbf{d}-1} \omega^{l_z y - j_z y} |y+k_z\rangle\langle y+i_z| \\
&= \sum_{x,y=0}^{\mathbf{d}-1} \omega^{j_z x + (l_z - j_z) y} |x+i_z\rangle\langle x|y+k_z\rangle\langle y+i_z| \\
&= \sum_{y=0}^{\mathbf{d}-1} \omega^{j_z (y+k_x) + (l_z - j_z) y} |y+k_z+i_z\rangle\langle y+i_z| \\
&= \sum_{y=0}^{\mathbf{d}-1} \omega^{j_z k_z + l_z y} |y+k_z+i_z\rangle\langle y+i_z| \\
&= \sum_{y=0}^{\mathbf{d}-1} \omega^{j_z k_z + l_z (y - i_z)} |y+k_z\rangle\langle y| \\
&= \omega^{j_z k_z - i_z l_z} \sum_{y=0}^{\mathbf{d}-1} \omega^{l_z y} |y+k_z\rangle\langle y| \\
&= \omega^{j_z k_z - i_z l_z} X_{k_z} Z_{l_z}. \qquad (8.3.1)
\end{aligned}
$$

Hence $E_{i,j} E_{k,l} E_{i,j}^{-1} = \otimes_{z=1}^{n} \omega^{j_z k_z - i_z l_z} X_{k_z} Z_{l_z}$. Consequently, $E_{i,j}$ commutes with $E_{k,l}$ over $\mathbb{C}^{\mathbf{d}^n}$ if and only if $\sum_{z=1}^{n} (j_z k_z - i_z l_z)$ vanishes.

The general form of the set $\overline{X}_i$ is given by $(\beta_{3_i}, \beta_{2_i}, \beta_{1_i} \mid \alpha_{3_i}, \alpha_{2_i}, \alpha_{1_i})^T \in \mathbb{Z}_{\mathbf{d}}^{2n}$ where α_1 and β_1 are r-dimensional vectors, α_2 and β_2 are $n-k-r$-dimensional vectors, and α_3 and β_3 are k-dimensional vectors. Correspondingly $\overline{Z}_j$ has the general form $(\beta'_{3_i}, \beta'_{2_i}, \beta'_{1_i} \mid \alpha'_{3_i}, \alpha'_{2_i}, \alpha'_{1_i})^T$. To establish the $2k$ operators of $\mathcal{N}(\mathcal{S}) - \mathcal{S}$, we take the standard form of $\mathcal{S}$ and the general forms of $\overline{X}_i$ and $\overline{Z}_j$, and together with the conditions on $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ over $\mathbb{C}^{\mathbf{d}^n}$, we have it that

the set $\overline{X}_i$ is determined by

$$\begin{aligned} \beta_1^T + A_2\beta_3^T - B_5\alpha_3^T &= 0 \\ -\alpha_2^T - C_5\alpha_3^T &= 0 \\ \alpha_3\beta_3^T - \alpha_3^T\beta_3 &= 0 \end{aligned} \tag{8.3.2}$$

while the set $\overline{Z}_j$ is given by

$$\begin{aligned} \beta'^T_1 + A_2\beta'^T_3 - B_5\alpha'^T_3 &= 0 \\ -\alpha'^T_2 - C_5\alpha'^T_3 &= 0 \end{aligned} \tag{8.3.3}$$

and the condition that $\overline{X}_i$ does not commute with $\overline{Z}_j$ imposes the further requirement that

$$\alpha'_3\beta_3^T - \alpha_3^T\beta'_3 \neq 0. \tag{8.3.4}$$

Let us suppose that $\alpha_3^T = 1$ therefore $\alpha_3 = 1$, hence, $\alpha_2 = -C_5^T$. By Theorem 26 and constraint (8.3.2), we have $\alpha_1 = \beta_2 = 0$. If we allow $\alpha'_3 = \beta_3 = 0$ then $\beta'_3 = 1$ and $\beta'_1 = -A_2^T$. Consequently, we have $\beta_1 = B_5$. Whence, the set $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ consisting of encoded $\overline{X}_i$ and $\overline{Z}_j$ vectors in a qu**d**it quantum system is given by

$$\overline{X} = \begin{pmatrix} 0 \\ 0 \\ B_5^T \\ -- \\ I \\ -C_5^T \\ 0 \end{pmatrix} \qquad \overline{Z} = \begin{pmatrix} I \\ 0 \\ -A_2^T \\ -- \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{8.3.5}$$

8.4 Encoding a Qudit Stabilizer Code

Channel encoding permits the transmission of information through a quantum system by embedding a $(\mathbb{C}^{\mathbf{d}})^{\otimes k}$ subspace within a $(\mathbb{C}^{\mathbf{d}})^{\otimes n}$ quantum system.

Cleve and Gottesman [19, 21] introduced an efficient scheme to compute the encoding network for any qubit stabilizer code. For an n-qu**d**it quantum code defined by a set of $n-k$ stabilizer elements, an encoding network transforms a k-qu**d**it information set into an n-qu**d**it codeword. The basis states for the coding space are obtained by implementing Gaussian elimination on the $n-k$ stabilizer set and appending the subsequent stabilizer set with a set of logical-X operators. Finally, a set of Fourier operators is targeted on the redundant elements of the encoding to ensure that the information content of the codewords is defined by the $(\mathbb{C}^{\mathbf{d}})^{\otimes k}$ subspace. The requirement of orthogonality among codewords implies that the embedding is necessarily a unitary transformation. Therefore, a decoding network can be implemented following error-correcting techniques by reversing the order of the encoding transformation and applying the set of inverse gates that correspond to those gates of the set of controlled operations. Table 8.3 illustrates an encoding scheme for a qu**d**it quantum code. A justification for the algorithm follows Table 8.3.

Table 8.3: Algorithm to compute an encoding network for a qudit code.

Input: Standard form of a code stabilizer, $\mathcal{S}$.

Output: Encoding network for a qudit code.

1. $\mathcal{A} \leftarrow \emptyset$
2. for b from 1 to $(n-k)-1$ do
3. for a from $b+1$ to $(n-k)$ do
4. $M_{a,b} := \mathcal{S}_{a,b} \times {\mathcal{S}_{b,b}}^{-1}$
5. for c from $(n-k)+1$ to $2n$ do
6. $\mathcal{S}_{a,c} \leftarrow \mathcal{S}_{a,c} - M_{a,b} \times \mathcal{S}_{b,c}$
7. end
8. for d from $b+1$ to $(n-k)$ do
9. $\mathcal{S}_{a,d} \leftarrow \mathcal{S}_{a,d} - M_{a,b} \times \mathcal{S}_{b,d}$
10. end
11. end
12. end
13. for a' from 2 to $(n-k)$ do
14. for b' from $a'-1$ downto 1 do

15. $M_{b',a'} := \mathcal{S}_{b',a'} \times {\mathcal{S}_{a',a'}}^{-1}$

16. for c' from $(n-k)+1$ to $2n$ do

17. $\mathcal{S}_{b',c'} \leftarrow \mathcal{S}_{b',c'} - M_{b',a'} \times \mathcal{S}_{a',c'}$

18. end

19. for d' from $a'+1$ to $(n-k)$ do

20. $\mathcal{S}_{b',d'} \leftarrow \mathcal{S}_{b',d'} - M_{b',a'} \times \mathcal{S}_{a',d'}$

21. end

22. end

23. end

24. $r :=$ rank of $\mathcal{S}$

25. for b from $n+r+1$ to $2n-k-1$ do

26. for a from $b-n+1$ to $(n-k)$ do

27. $M_{a,b} := \mathcal{S}_{a,b} \times {\mathcal{S}_{b-n,b}}^{-1}$

28. for c from $n+r+2$ to $2n$ do

29. $\mathcal{S}_{a,c} \leftarrow \mathcal{S}_{a,c} - M_{a,b} \times \mathcal{S}_{b-n,c}$

30. end

31. for d from $n+1$ to $n+r$ do

32. $\mathcal{S}_{a,d} \leftarrow \mathcal{S}_{a,d} - M_{a,b} \times \mathcal{S}_{b-n,d}$

33. end

34. end

35. end

36. for b' from $n + r + 2$ to $2n - k$ do

37. for a' from $r + 1$ to $b' - n - 1$ do

38. $M_{a',b'} := \mathcal{S}_{a',b'} \times {S_{b'-n,b'}}^{-1}$

39. for c' from $b' + 1$ to $2n$ do

40. $\mathcal{S}_{a',c'} \leftarrow \mathcal{S}_{a',c'} - M_{a',b'} \times \mathcal{S}_{a'+1,c'}$

41. end

42. for d' from $n + 1$ to $n + r$ do

43. $\mathcal{S}_{a',d'} \leftarrow \mathcal{S}_{a',d'} - M_{a',b'} \times \mathcal{S}_{a'+1,d'}$

44. end

45. end

46. end

47. for a from 1 to $(n - k)$

48. for b from 1 to $2n$ do

49. $\mathcal{S}_{a,b} \leftarrow S_{2n-b+1,(n-k)-a+1}$

50. end

51. end

52. for a' from $2n$ downto $n + 1$ do

53. find the first column $l \notin \mathcal{A}$ where $\mathcal{S}_{a',l} = 1$

54. include l into $\mathcal{A}$

55. end

56. for a from 1 to n do

57. for b from 1 to $(n-k)$ do

58. $\mathcal{S}_{a,b} \leftarrow \mathcal{S}_{a,b} \times \mathcal{S}_{n+a,b}$

59. end

60. end

61. Append k columns of $\bar{X}$ to $\mathcal{S}_{a,b}$

62. $\mathbb{F} \leftarrow \otimes_{l \in \mathcal{A}} \mathbb{F}^{(l)}$

63. return $(\mathbb{F}, \overline{X}, \mathcal{S})$.

We show the correctness of our algorithm. We can always perform Gaussian elimination on S, the $n-k \times 2n$ stabilizer matrix, without changing the system it represents. This corresponds to Table 8.3 steps $2-12$. Let us assume that the X matrix, the first $n-k \times n$ submatrix of $\mathcal{S}$, has rank $r \leq n-k$, then S has the following form,

$$\left(\begin{array}{c|c} A & B \\ 0 & C \end{array}\right) \tag{8.4.1}$$

where A is an $(r \times n)$ submatrix of X. The Z submatrix of $\mathcal{S}$ takes no particular form. We continue the algorithm by iterating on a' over steps $13-23$. This allows us to obtain A in row-echleon form,

$$\left(\begin{array}{cc|c} I & A_1 & B \\ 0 & 0 & C \end{array}\right) \tag{8.4.2}$$

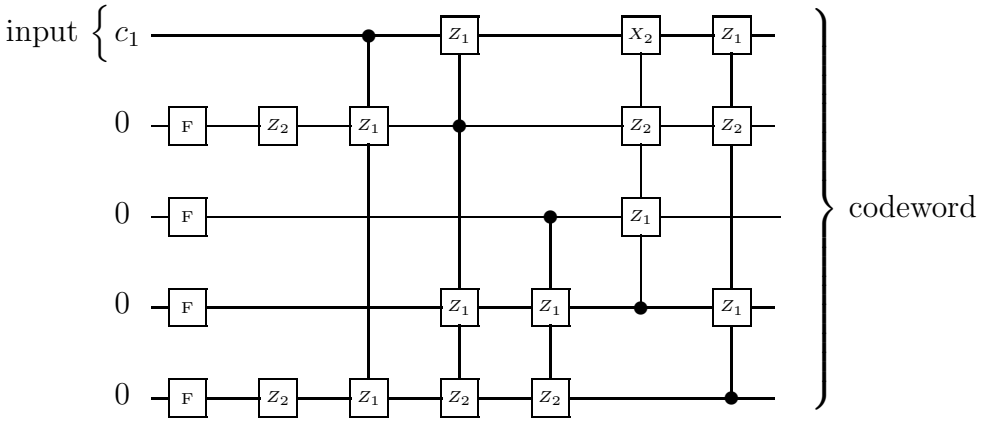


Figure 8.2: Encoding circuit for a five qutrit code.

As the rows are linearly independent, C must have rank $n-k-r$, hence it contains an $(n-k-r \times n-k-r)$ submatrix C_2 of maximal rank. Thus, we may produce a matrix whose form is given by,

$$\left(\begin{array}{ccc|ccc} I & A_1 & A_2 & B_1 & B_2 & B_3 \\ 0 & 0 & 0 & C_1 & C_2 & C_3 \end{array}\right) \tag{8.4.3}$$

In a similar manner to the row operations targeted on X, we begin another stage of iteration on the submatrix C_2. The algorithm steps $25-46$ mirror those of steps $2-23$, and returns the submatrix C_2 in row-echleon form. Steps $25-46$ also act on the submatrices C_1 and C_3, transforming them to C_4 and C_5 respectively. Given $C_2 = I$, we may reduce B_2 to the all zero matrix. Further to this, submatrices C_4 and C_5 condition themselves on B_1 and B_3 respectively. The resulting stabilizer matrix is

$$\left(\begin{array}{ccc|ccc} I & A_1 & A_2 & B_4 & 0 & B_5 \\ 0 & 0 & 0 & C_4 & I & C_5 \end{array}\right) \tag{8.4.4}$$

Steps $47-51$ transpose the stabilizer and reverse the subsequent order of columns. Algorithm steps $52-55$ search for any columns $l \notin \mathcal{A}$ such that

$\mathcal{S}_{a',l} = 1$, and updates $\mathcal{A}$. The X and Z portions of the stylised $\mathcal{S}$ are amalgamated in steps $56 - 60$, before finally appending the encoded $\overline{X}$,

$$\begin{pmatrix} I \\ -C_5^T \\ B_5^T \end{pmatrix}, \tag{8.4.5}$$

and Fourier set $\otimes_{l \in \mathcal{A}} \mathbb{F}^{(l)}$ to $\mathcal{S}$.

8.5 Correcting Procedure

The stabilizer formalism is well suited to meet the requirements of quantum coding theory since it provides a compact basis to characterise the errors that a code can detect and correct. This is qualified since the $n - k$ stabilizers $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{n-k}$ can be seen to act as *check operators* for the code. Alternatively, the stabilizer forms a set of collective observables that provide non-demolition measurements which are used in the diagnosis of errors. The set of non-demolition measurements is also known as the error *syndrome* for the code. We consider a more formal approach to the calculation of the error syndrome.

For stabilizer generators $\mathcal{M}_\alpha$ and errors E_β, we have

$$E_\beta \mathcal{M}_\alpha E_\beta^{-1} |\psi\rangle = \varsigma_{\alpha\beta} \mathcal{M}_\alpha |\psi\rangle \tag{8.5.1}$$

for all $|\psi\rangle \in \mathcal{C}$. The set $\varsigma_{\alpha\beta}$, for $\alpha = 1, \ldots, n-k$, constitutes the error syndrome for E_β since $\varsigma_{\alpha\beta}$ is the eigenvalue of M_α obtained by measuring M_α against E_β. In particular, we have it that $\varsigma_{\alpha\beta} = \omega^{\sum_{z=1}^{n} j_z k_z - i_z l_z}$. Recall that the error correcting condition (2.3.2) for errors $E_\beta, E_{\beta'} \in \mathcal{E}^n$ is given by

$$\langle\psi| E_{\beta'}^{-1} E_\beta |\psi\rangle = c_{\beta'\beta}. \tag{8.5.2}$$

This condition is satisfied provided that one of the following holds

1. $E_{\beta'}^{-1}E_{\beta} \in \mathcal{S}$
2. There exists some $\mathcal{M}_{\alpha} \in \mathcal{S}$ such that $\mathcal{M}_{\alpha}$ does not commute with $E_{\beta'}^{-1}E_{\beta}$.

Proof. Suppose $E_{\beta'}^{-1}E_{\beta} \in \mathcal{S}$ then $\langle\psi| E_{\beta'}^{-1}E_{\beta} |\psi\rangle = \langle\psi|\psi\rangle = 1$. Next, suppose that there is some $\mathcal{M}_{\alpha} \in \mathcal{S}$ that does not commute with $E_{\beta'}^{-1}E_{\beta}$. Therefore,

$$(E_{\beta'}^{-1}E_{\beta})\mathcal{M}_{\alpha}(E_{\beta'}^{-1}E_{\beta})^{-1} = \omega^{\sum_{z=1}^{n} j_z(k_z-k'_z)-i_z(l_z-l'_z)}\mathcal{M}_{\alpha} \tag{8.5.3}$$

with $\sum_{z=1}^{n} j_z(k_z - k'_z) - i_z(l_z - l'_z) \neq 0$. Then

$$\begin{aligned}
\langle\psi| E_{\beta'}^{-1}E_{\beta} |\psi\rangle &= \langle\psi| E_{\beta'}^{-1}E_{\beta}\mathcal{M}_{\alpha} |\psi\rangle \\
&= \omega^{\sum_{z=1}^{n} j_z(k_z-k'_z)-i_z(l_z-l'_z)} \langle\psi| \mathcal{M}_i E_{\beta'}^{-1}E_{\beta} |\psi\rangle \\
&= \omega^{\sum_{z=1}^{n} j_z(k_z-k'_z)-i_z(l_z-l'_z)} \langle\psi| E_{\beta'}^{-1}E_{\beta} |\psi\rangle
\end{aligned} \tag{8.5.4}$$

is maintained when $\langle\psi| E_{\beta'}^{-1}E_{\beta} |\psi\rangle = 0$. Thus, $E_{\beta}, E_{\beta'}$ form a set of correctable errors for the stabilizer code $\mathcal{C}$.

Should an error E_{β} occur on the coding space $\mathcal{C}$ having an associated error syndrome $\varsigma_{\alpha\beta} = \omega^{\sum_{z=1}^{n} j_z(k_z-k'_z)-i_z(l_z-l'_z)}$ for $\alpha = 1, \ldots, n-k$, then recovery is achieved by applying $E_{\beta}^{-1} = E_{-\beta}$. A quantum code is called nondegenerate if there is a 1-1 correspondence between errors E_{β} and syndrome $\varsigma_{\alpha\beta}$, otherwise, the code is said to be degenerate. In the case of a degenerate code, there are syndromes $\varsigma_{\alpha\beta}$ and $\varsigma_{\alpha\beta'}$ for which $\varsigma_{\alpha\beta} = \varsigma_{\alpha\beta'}$ where $E_{\beta} \neq E_{\beta'}$. Correction will induce a nontrivial action on the coding space $\mathcal{C}$ by mistakingly applying $E_{\beta'}^{-1}$ on a corrupted state of E_{β}^{-1}.

8.6 Complexity of a Qudit Stabilizer Code

Complexity measures the degree of efficiency to which an information preserving function can be solved. We consider the computational complexity of some problem function where focus is placed on the growth of a solution time required against the size of the problem, and whose aim is to furnish an appropriate result abstracted away from any particular circuitry architecture simulating the function.

Consider the set of Boolean functions in $LB_{k,n}$ that describe the set of all linear transformation from $\mathbb{F}_q^k$ to $\mathbb{F}_q^n$. The gate complexity of a function $f \in LB_{k,n}$, given by $f(x) = Ax$, is denoted $C(A)$. In such case, A is a $k \times n$ matrix with entries in $\mathbb{F}_q$, and $x = (x_1, \ldots, x_n)^t$ is a vector with elements in $\mathbb{F}_q$. The complexity of computing a Boolean function can be formalised by the number of CNOT gates a circuit possesses. Consequently, $C(A)$ is then defined as the minimum number of CNOT gates needed to realise f. This number is also referred to as the gate complexity of A. Byrne *et al.* [16] determined the CNOT gate complexity of $\mathcal{H}_k$, where $\mathcal{H}_k$ is the parity-check matrix of the $[2^k - 1, 2^k - k - 1]$ Hamming code. Furthermore, a general upper bound on the complexity of functions in $LB_{n,n}$ has also been determined from the complexity of the transpose of $\mathcal{H}_k$ [16].

Quantum computation is realised by a set of unitary operators that exhibit a reversible nature. Thus, to illustrate equivalence between quantum computation and the Boolean function f, we require that f be a reversible function. Therefore, recall the reversible function $\tilde{f}$ given by $\tilde{f} : {\mathbb{F}_{\mathbf{d}}}^{n+k} \mapsto {\mathbb{F}_{\mathbf{d}}}^{n+k}$ where $\tilde{f}(\alpha; 0^{(n)}) = (\alpha; f(\alpha))$. Then, the computation of $\tilde{f}$ corresponds to a unitary operator $U_{\tilde{f}} : \mathbb{C}^{\mathbf{d}^{n+k}} \mapsto \mathbb{C}^{\mathbf{d}^{n+k}}$ in a quantum system. As a consequence, we have the following result.

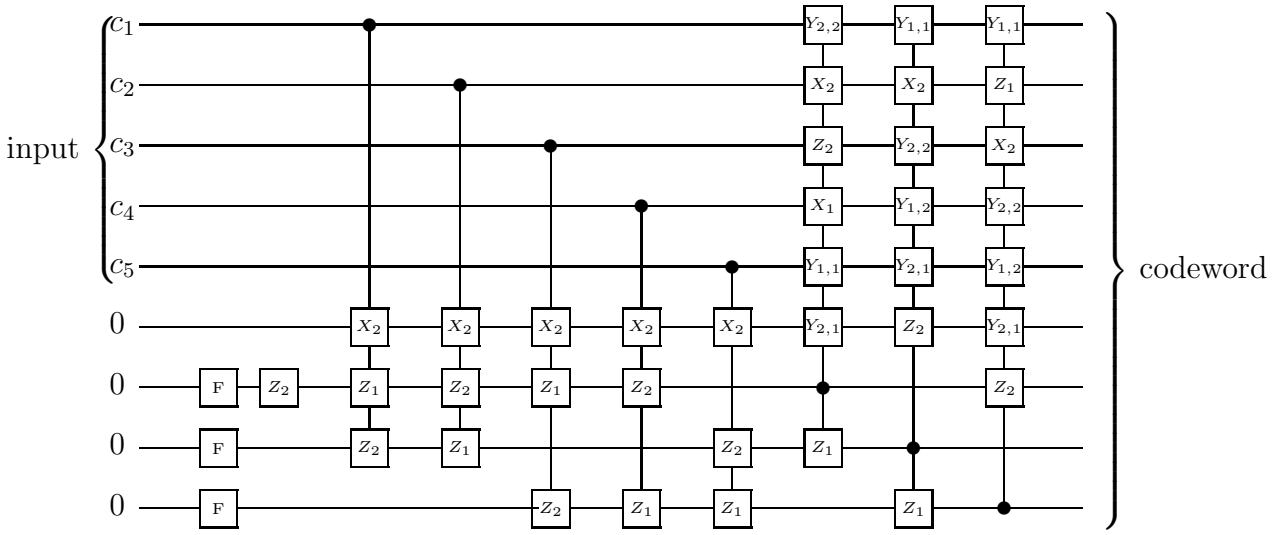


Figure 8.3: Encoding circuit for a nine qutrit code.

Theorem 27. An upper bound on the complexity associated with the implementation of a qu**d**it stabilizer code is $2n^2/\log_2 n$.

Proof A qu**d**it stabilizer code has an encoding circuit network of depth n and width n which consists of CNOT and Phase gates. Byrne *et al.* [16] have shown that $2n^2/\log_2 n$ is an upper bound on the gate complexity of functions in $LB_{n,n}$. As a consequence of quantum parallelism, the complexity associated with implementing a stabilizer encoding is an instance of the CNOT complexity of computing a function f in $LB_{n,n}$, and the result follows.

$\mathcal{M}_1$	X_1	Z_1	Z_{d-1}	X_{d-1}	I
$\mathcal{M}_2$	I	X_1	Z_1	Z_{d-1}	X_{d-1}
$\mathcal{M}_3$	X_{d-1}	I	X_1	Z_1	Z_{d-1}
$\mathcal{M}_4$	Z_{d-1}	X_{d-1}	I	X_1	Z_1
$\overline{X}_1$	Z_1	I	I	Z_1	X_1
$\overline{Z}_1$	Z_1	Z_1	Z_1	Z_1	Z_1

Table 8.4: The stabilizer for a five qu**d**it code derived from the five qubit code.

Appendix

MAPLE PROGRAM: EXAMPLES

Calculate the roots α_j, the coefficients β_j and test the sequence $a_j = \sum_{i=0}^{j/d} \binom{j-(d-1)i}{i}$ against the closed form $\sum_{l=1}^{\mathbf{d}} \beta_l \alpha_l^j$.

> restart;

Example 1. Test the closed form for the sum $\sum_{i=0}^{j/4} \binom{j-3i}{i}$. The associated generating function is $1 - z - z^4$. The reciprocals of the roots of $1 - z - z^4$ are obtained from

> factor($z^4 - z^3 - 1$, complex);

$(z+.8191725134)(z-.2194474721+.9144736630I)(z-.2194474721-.9144736630I)(z-1.380277569)$

We calculate the values of the beta coefficients. As $\beta_l = \frac{-\alpha_l}{B'(1/\alpha_l)}$, we determine each beta coefficient to be

> $\beta_1 = -(-.8191725134)/(-1 - 4 * (1/(-.8191725134))^3)$;

$\beta_1 = .1305102698$

> $\beta_2 = -(.2194474721-.9144736630I)/(-1-4*(1/(.2194474721-.9144736630I))^3)$;

$\beta_2 = .1610008758 + .1534011260I$

> $\beta_3 = -(.2194474721+.9144736630I)/(-1-4*(1/(.2194474721+.9144736630I))^3)$;

$\beta_3 = .1610008758 - .1534011260I$

> $\beta_4 = -(1.380277569)/(-1 - 4 * (1/(1.380277569))^3)$;

$\beta_4 = .5474879784$

We put the values of α_l and β_l from above into the closed form expression $\sum_{l=1}^{d} \beta_l \alpha_l^j$ and generate instances of this expression.

> f := proc (j) options operator, arrow; $(.1305102698) * (-.8191725134)^j +$

$(.1610008758+.1534011260I)*(.2194474721-.9144736630I)^j+(.1610008758-$
$.1534011260I)*(.2194474721+.9144736630I)^j+(.5474879784)*(1.380277569)^j;$

end proc;

$f := j -> .1305102698 * (-.8191725134)^j + (.1610008758 + .1534011260I) *$
$(.2194474721-.9144736630I)^j+(.1610008758-.1534011260I)*(.2194474721+$
$.9144736630I)^j + .5474879784 * 1.380277569^j$

> for j from 0 to 15 do; round $(f(j))$; end;

$1, 1, 1, 1, 2, 3, 4, 5, 7, 10, 14, 19, 26, 36, 50, 69$

> restart;

Example 2. Test the closed form for the sum $\sum_{i=0}^{j/8} \binom{j-7i}{i}$. The associated generating function is $1 - z - z^8$. The reciprocals of the roots of $1 - z - z^8$ are obtained from

> factor($z^8 - z^7 - 1$,complex);

$(z + .9115923535) * (z + .6157823065 + .6871957511I) * (z + .6157823065 -$
$.6871957511I)*(z-.1033089835+.9564836042I)*(z-.1033089835-.9564836042I)*$
$(z - .8522421840 + .6352622030I) * (z - .8522421840 - .6352622030I) * (z -$
$1.232054631)$

We calculate the values of the β_l coefficients. As $\beta_l = \frac{-\alpha_l}{B'(1/\alpha_l)}$, we determine each β_l coefficient to be

> $\beta_1 = -(-.9115923535)/(-1 - 8 * (1/(-.9115923535))^7);$

$\beta_1 = .06378010282$

> $\beta_2 = -(-.6157823065 - .6871957511I)/(-1 - 8 * (1/(-.6157823065 -$
$.6871957511I))^7);$

$\beta_2 = .06449005934 + .02789285455I$

$> \beta_3 = -(-.6157823065 + .6871957511I)/(-1 - 8 * (1/(-.6157823065 + .6871957511I))^7);$

$\beta_3 = .06449005934 - .02789285455I$

$> \beta_4 = -(.1033089835-.9564836042I)/(-1-8*(1/(.1033089835-.9564836042I))^7);$

$\beta_4 = .06911712233 + .06926484155I$

$> \beta_5 = -(.1033089835+.9564836042I)/(-1-8*(1/(.1033089835+.9564836042I))^7);$

$\beta_5 = .06911712233 - .06926484155I$

$> \beta_6 = -(.8522421840-.6352622030I)/(-1-8*(1/(.8522421840-.6352622030I))^7);$

$\beta_6 = .1188399306 + .1719523210I$

$> \beta_7 = -(.8522421840+.6352622030I)/(-1-8*(1/(.8522421840+.6352622030I))^7);$

$\beta_7 = .1188399306 - .1719523210I$

$> \beta_8 = -(1.232054631)/(-1 - 8 * (1/(1.232054631))^7);$

$\beta_8 = .4313256714$

Put the values of α_l and β_l from above into the closed form expression $\sum_{l=1}^{d} \beta_l \alpha_l^j$ and generate instances of this form.

$> f :=$ proc (j) options operator, arrow; $(.06378010282) * (-.9115923535)^j +$ $(.0644+.0278I)*(-.61578230-.68719575I)^j+(.0644-.0278I)*(-.61578230+$ $.68719575I)^j+(.0691+.0692I)*(.10330898-.95648360I)^j+(.0691-.0692I)*$ $(.10330898 + .95648360I)^j + (.1188 + .1719I) * (.85224218 - .63526220I)^j +$ $(.1188-.1719I)*(.85224218+.63526220I)^j+(.4313256714)*(1.232054631)^j;$

end proc;

$f := j -> .06378010282*(-.9115923535)^j+(.0644+.0278I)*(-.61578230-$ $.68719575I)^j+(.0644-.0278I)*(-.61578230+.68719575I)^j+(.0691+.0692I)*$ $(.10330898 - .95648360I)^j + (.0691 - .0692I) * (.10330898 + .95648360I)^j +$

$(.1188 + .1719I) * (.85224218 - .63526220I)^j + (.1188 - .1719I) * (.85224218 +$

$.63526220I)^j + .4313256714 * (1.232054631)^j$

$>$ for j from 0 to 25 do; $>$ round($f(j)$); $>$ end;

$1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 14, 18, 23, 29, 36, 44, 53, 64, 78$

Bibliography

[1] Aspect A, Dalibard J, and Roger G, *Experimental Test of Bell's Inequalities Using Time-Verying Analyzers*, Physical Review Letters, Vol. 49, 1982, pp. 1804-1807.

[2] Ashikhmin A, and Knill E, *Nonbinary Quantum Stabilizer Codes*, IEEE Trans. Inform. Theory, Vol. 47, 2001, pp. 3065-3072.

[3] Barenco A, Bennett C H, Cleve R, DiVincenzo D P, Margolus N, Shor P, Sleator T, Smolin J, and Weinfurter H, *Elementary Gates for Quantum Computation*, Physical Review A, Vol. 52, 1995, pp. 3457-3488.

[4] Bell J, *On the Einstein-Podolsky-Rosen Paradox*, Physics, Vol. 1, 1964, pp. 195-200.

[5] Bennett C H, Brassard G, Crepeau C, Jozsa R, Peres A, and Wooters A K, *Teleporting an Unknown Quantum State via Dual Classical and EPR Channels*, Physical Review Letters, Vol. 70, 1993, pp. 1895-1899.

[6] Bennett C H, *Logical Reversibility of Computation*, IBM J. Res. Develop., 17, 525, 1973.

[7] Bergholm V, Vatiainen J J, Möttönen M, and Salomaa M M, *Quantum circuits with uniformly controlled one-qubit gates*, Phys. Review A, 71, 052330, 2005.

[8] Bergholm V, Vatiainen J J, Möttönen M, and Salomaa M M, *Quantum circuits for general multiqubit gates*, Phys. Review Letters, 93, 13, 2004.

[9] Blasiak, Horzela, Penson, Solomon, and Duchamp, *Combinatorics and Boson normal ordering: A gentle introduction*, quant-ph/07043119, - 'A very elegant way of storing and tackling information about sequences is attained through their generating function.'

[10] Bose R C, and Ray-Choudhuri D K, *On a class of error correcting binary group codes*, Information and Control, Vol. 3, 1960, pp. 68-79.

[11] Bouwmeester D, Ekert A, and Zeilinger A, *The Physics of Quantum Information*, Springer, 2001, p. 163.

[12] Calderbank A R, Rains E M, Shor P W, and Sloane N J A, *Quantum Error Correction and Orthogonal Geometry*, Physical Review Letters, Vol. 78, 1995, pp. 405-408.

[13] Calderbank A R, Rains E M, Shor P W, and Sloane N J A, *Quantum Error Correction via Codes over GF(4)*, IEEE Trans. Inform. Theory, Vol. 44, 1998, pp. 1369-1387.

[14] Calderbank A R, and Shor P W, *Good Quantum Error-Correcting Codes exist*, Physical Review A, Vol. 54, 1996, pp. 1098-1105.

[15] Cameron P J, *Combinatorics: Topics, Techniques, Algorithms*, Cambridge University Press, 1994.

[16] Carmelo Interlando J, Byrne E, and Rosenthal J, *The Gate Complexity of Syndrome Decoding of Hamming Codes*, Proceedings of the Tenth International Conference on Applications of Computer Algebra, Beaumont, Texas, 2004, pp. 33-37.

[17] Cirac J L, and Zoller P, *Quantum Computations with Cold Trapped Ions*, Phys. Rev. Lett., Vol. 74, 1995, p. 4091-4094.

[18] Clarisse L, Ghosh S, Severini S, and Sudbery A, *The disentangling power of unitaries*, quant-ph/0611075, 2006.

[19] Cleve R, and Gottesman D, *Efficient computations of encodings for quantum error correction*, Phys. Rev. A, Vol. 56, 1997, pp. 76-82. LANL e-print, quant-ph/9607030.

[20] Cleve R, *Quantum Stabilizer Codes and Classical Linear Codes*, Phys. Rev. A, Vol. 55, 1997, pp. 4054-4059. LANL e-print, quant-ph/9612048.

[21] Cleve R, *An Introduction to Quantum Complexity Theory*, Collected Papers on Quantum Computation and Quantum Information Theory, eds. C. Macchiavello, G.M. Palma, and A. Zeilinger, World Scientific, Singapore, 2000. LANL e-print, quant-ph/9906111.

[22] Chuang I L, and Yamamoto Y, *Quantum Bit Regeneration*, Phys. Rev. Letters, Vol. 76, 1996, pp. 4281-4284.

[23] Chuang I L, and Yamamoto Y, *Creation of a persistent quantum bit using error correction*, Phys. Rev. A, Vol. 55, 1997, pp. 114-127.

[24] Deutsch D, *Quantum theory, the Church-Turing principle and the universal quantum computer,* Proc. Roy. Soc. Lond. A, Vol. 400, 1985, pp.97-117.

[25] Deutsch D, *Quantum Computational Networks* Proc. Roy. Soc. Lond. A, Vol. 425, 1989, pp. 73-90.

[26] Deutsch D, Barenco A, and Ekert A, *Universality in Quantum Computation*, Proc. Roy. Soc. Lond. A, Vol. 449, 1995, pp. 669-677.

[27] DiVincenzo D P, *Quantum Gates and Circuits*, Proceedings of the ITP Conference on Quantum Coherence and Decoherence, Proc. Roy. Soc. Lond. A, Vol. 454, 1998, pp. 261-276. LANL e-print, quant-ph/9705009.

[28] Erdmann K, and Wildon M J, *Introduction to Lie Algebras*, Springer, 2006.

[29] Feynman R P, *Simulating physics with computers*. Int. J. Theor. Phys. Vol. 21, 1982, pp. 467-488.

[30] Fowler A G, Dervitt S J, and Hollenberg L C L, *Implementation of Shor's algorithm on a linear nearest neighbour qubit array*, quant-ph/0402196, 2004.

[31] Gottesman D, *A Class of Quantum Error-Correcting Codes saturating the Quantum Hamming bound*, Phys. Rev. A, Vol. 54, 1996, pp. 1862-1968.

[32] Gottesman D, *Stabilizer Codes and Quantum Error Correction*, Ph.D. Thesis, Calif. Inst. Technol., Pasadena, CA, 1997.

[33] Gottesman D, *Fault-Tolerant Quantum Computation with Higher-Dimensional Systems*, Quantum Computing and Quantum Communications, Proceedings of the 1st NASA International Conference on Quantum Computing and Quantum Communications (QCQC), Palm Springs, California, ed. C. Williams, New York, NY, Springer-Verlag, 1998, pp. 302-313. LANL e-print, quant-ph/9802007.

[34] Gottesman D, *The Heisenberg Representation of Quantum Computers*, Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics, eds. S. P. Corney, R. Delbourgo, and P. D. Jarvis, Cambridge, MA, International Press, 1999, pp. 32-43. LANL e-print, quant-ph/9807006.

[35] Graham R L, Knuth D E, and Patashnik O, *Concrete Mathematics*, 3rd ed., Addison-Wesley, 1994.

[36] Grassl M, Rötteler M, and Beth T, *Efficient Quantum Circuits for Non-Qubit Quantum Error-Correcting Codes*, International Journal of Foundations of Computer Science, Vol. 14, No. 5, 2003, pp. 757-775. LANL e-print, quant-ph/0211014.

[37] Gershenfeld N, and Chuang I, *Bulk Spin Resonance Quantum Computation*, Science Vol. 275, 1997, pp.350-356.

[38] http://mathworld.wolfram.com/IrreducibleRepresentation.html.

[39] Hamming R W, *Error Detecting and Error Correcting Codes*, Bell System Technical Journal, Vol. 26, 1950, pp. 147-160.

[40] Hardy Y, and Steeb W H, *Decomposing the SWAP quantum gate*, J. Phys. A: Math. Gen. 39, 2006.

[41] Herstein I N, *Topics in Algebra*, 2nd ed., Wiley and Sons, 1975.

[42] Hill C D, *Robust CNOT gates from almost any interaction*, quant-ph/0610059, 2006.

[43] Hocquenghem A, *Codes correcteurs d'erreurs*, Chiffres (Paris), Vol. 2, 1959, pp. 147-156.

[44] Hoffman D G, Leonard D A, Linder C C, Phelps K T, Rodger C A, Wall J R, *Coding Theory; The Essentials*, Marcel Dekker, Inc., 1991, p. 38.

[45] Khaneja N, Brockett R, and Glaser S J, *Time optimal control in spin systems*, Phys. Review A, 63, 032308, 2001.

[46] Klappenecker A, and Rötteler M, *Unitary Error Bases: Constructions, Equivalence and Applications*, Lecture Notes in Computer Science, 2003, 2643, pages 139-149.

[47] Klappenecker A, and Rötteler M, *Beyond Stabilizer Codes I: Nice Error Bases*, IEEE Trans. Inform. Theory, Vol. 48, 2002, pp. 2392-2395. LANL e-print, quant-ph/0010082.

[48] Klappenecker A, and Rötteler M, *Beyond Stabilizer Codes II: Clifford Codes*, IEEE Trans. Inform. Theory, Vol. 48, 2002, pp. 2396-2399. LANL e-print, quant-ph/0010076.

[49] Klappenecker A, and Rötteler M, *On the monomiality of nice error bases*, IEEE Trans. Inform. Theory, Vol. 51, 2005, pp. 1084-1089. LANL e-print, quant-ph/0301078.

[50] Knill E, *Group representations, error bases and quantum codes*, Los Alamos National Laboratory Report LAUR-96-2807, 1996.

[51] Knill E, *Non-binary Unitary Error Bases and Quantum Codes*, Los Alamos National Laboratory Report LAUR-96-2717, 1996, quant-ph/9608048.

[52] Knill E, and Laflamme R, *A theory of quantum error-correcting codes*, Physical Review A, Vol. 55, No. 2, 1997, pp. 900-911.

[53] Liang L, and Li C, *Realization of quantum SWAP gate between flying and stationary qubits*, Phys. Review A, 72, 024303, 2005.

[54] Lu F, and Marinescu D C, *Quantum Error Correction of Time-Correlated Errors*, quant-ph/06052206.

[55] Lu C J, and Tsai S C, *The Periodic Property of Binomial Coefficients Modulo m and Its Applications*, 10th SIAM Conference on Discrete Mathematics, Minneapolis, Minnesota, USA, 2000.

[56] http://www.maplesoft.com/

[57] MacWilliams F J, and Sloane N J A, *The Theory of Error-Coreecting Codes*, North-Holland, Amsterdam, 1997.

[58] Monroe C, Meekhof D, King B, Itano W, and Wineland D, *Demonstration of a Fundamental Quantum Logic Gate*, Phys. Rev. Lett., Vol 75, 1995, pp. 4714-4717.

[59] Nielsen M A, *A geometric approach to quantum circuit lower bounds*, quant-ph/0502070.

[60] Nielsen M A, and Chuang I L, *Quantum Computations and Quantum Information*, Cambridge University Press, 2000.

[61] Nielsen M A, and Chuang I L, *Quantum Computations and Quantum Information*, Cambridge University Press, 2000, p. 23.

[62] Nielsen M A, and Chuang I L, *Quantum Computations and Quantum Information*, Cambridge University Press, 2000, p. 159.

[63] Nielsen M A, and Chuang I L, *Quantum Computations and Quantum Information*, Cambridge University Press, 2000, p. 191.

[64] Nielsen M A, and Chuang I L, *Quantum Computations and Quantum Information*, Cambridge University Press, 2000, p. 205.

[65] Nielsen M A, and Chuang I L, *Quantum Computations and Quantum Information*, Cambridge University Press, 2000, p. 436.

[66] Nielsen A M, Knill E, and Laflamme R, *Complete Quantum Teleportation using Nuclear Magnetic Resonance*, Nature, Vol. 396, No. 7706, 1998, pp. 52-55.

[67] Petkovsek M, Wilf H, and Zeilberger D, $A = B$, www.cis.upenn.edu/ wilf/AeqB.html.

[68] http://en.wikipedia.org/wiki/Separable_polynomial.

[69] Plenio M B, and Knight P L, *Realistic lower bounds for the factorisation time of large numbers on a quantum computer*, Physical Review A, Vol. 53, 1996, pp. 2986-2990.

[70] Preskill J, *Quantum Computing: Pro and Con*, Proc. Roy. Soc. Lond. A Vol. 454, 1998, pp. 469-486. LANL e-print, quant-ph/9705032.

[71] Rains E M, *Nonbinary quantum codes*, IEEE Trans. Inform. Theory, Vol. 45, 1999, pp. 1827-1832.

[72] Rosen K H, *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, 2000.

[73] Schlingemann D, *Problem 6 in open problems in Quantum Information Theory*, www.imaph.tu-bs.de/problems/

[74] Shannon C E, *A Mathematical Theory of Communication*, Bell System Technical Journal, Vol. 27, 1995, pp. 379-423.

[75] Schumacher B, *Quantum Coding*, Phys. Rev. A, Vol. 51, 1995, pp. 2738-2747.

[76] Sedlák M, and Plesch M, *Towards optimization of quantum circuits*, quant-ph/0607123.

[77] Shende V, Markov I L, and Bullock S, *Synthesis of Quantum Logic Circuits*, IEEE Trans. on Computer-Aid Design, 25, no. 6, p. 100, 2006.

[78] Shende V, Markov I L, and Bullock S, *Minimal universal two-qubit controlled-NOT-based circuits*, Phys. Review A, 69, 062321, 2004.

[79] Shor P W, *Scheme for reducing decoherence in quantum computer memory*, Physical Review A, Vol. 52, 1995, pp. 2493-2496.

[80] Shor P W, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 1994.

[81] Shor P W, and Laflamme R, *Quantum Analog of the MacWilliams Identities for Classical Coding Theory*, Phys. Rev. Lett., Vol. 78, 1997, pp. 1600-1602.

[82] Steane A M, *Quantum Reed-Muller Codes*, IEEE Trans. Inform. Theory, Vol. 45, No.5, 1999, pp. 1701-1703. LANL e-print, quant-ph/9608026.

[83] Steane A M, *Multiple Particle Interference and Quantum Error Correction*, Proc. Roy. Soc. Lond. A, Vol. 452, 1996, pp. 2551-2577. LANL e-print, quant-ph/9601029.

[84] Steane A M, *How to build a 300 bit, 1 Gop quantum computer*, LANL e-print, quant-ph/0412165.

[85] Testolin M J, Hill C D, Wellard C J, and Hollenberg L C L, *A precise CNOT gate in the presence of large fabrication induced variations of the exchange interaction strength,* e-print: quant-ph/0701165.

[86] Toffoli T, *Reversible computing,* Automata Languages and Programming, Seventh Colloquium, Lecture Notes in Computer Science, Vol. 84, de Bakker J W and van Leeuwen J, eds., Springer, 1985, pp. 632-644.

[87] Turchette Q, Hood C J, Lange W, Mabuchi H, and Kimble H J, *Measurement of Conditional phase Shifts for Quantum Logic,* Phys. Rev. Lett., Vol. 75, 1995, pp. 4710-4713.

[88] Turing A M, *On computable numbers, with an application to the Entscheidungsproblem*, Proc. Lond. Math. Sco. 2, 42:230, 1936.

[89] http://en.wikipedia.org/wiki/Church-Turing_thesis

[90] Vatan F, and Williams C, *Optimal quantum circuits for general two-qubit gates*, Phys. Rev. A 69, 032315, 2004.

[91] Vidal G, and Dawson C M, *Universal quantum circuit for two-qubit tranformations with three controlled-NOT gates*, Phys. Rev. A 69, 2004.

[92] Vlasov A Y, *Algebras and universal quantum computations with higher dimensional systems*, Proc. SPIE, Vol. 5128, 2003, pp. 29-36, LANL e-print, quant-ph/0210049.

[93] Petkovsek M, Wilf H, and Zeilberger D, *A=B*, http://www.math.upenn.edu/wilf/AeqB.html.

[94] Wilf H, *generatingfunctionology*, http://www.math.upenn.edu/wilf/gfology2.pdf.

[95] Werner R, *All teleportation and dense coding schemes*, J. Phys. A, vol. 34, pp. 7081-7094, 2001, quant-ph/0003070.

[96] Weyl H, *The Thoery of Groups and Quantum Mechanics*, Dover Publications, New York, 1931.

[97] Zanardi P, Zalka C, and Faoro L, *Entangling power of quantum evolutions*, Phys. Review A, 62, 030301, 2000.