Flag Fault-tolerant Error Correction for Cyclic CSS Codes

by

Theerapat Tansuwannont

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Physics (Quantum Information)

Waterloo, Ontario, Canada, 2018

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Fault-tolerant quantum error correction (FTEC) protocol is one of the most important components for a fault-tolerant simulation of a quantum circuit. It helps prevent error accumulation during the process which may cause the computation to fail. In a physical implementation of fault-tolerant circuits, the number of required ancilla qubits for the FTEC protocol can be a major concern because it grows exponentially as the level of concatenation increases whereas the resources are limited. Recently, the idea of using flag qubits to detect high-weight errors caused by fewer faults in syndrome extraction circuits has been proposed. The use of flag qubits allows the construction of fault-tolerant protocols with the fewest number of ancillas known to-date. In this work, some critical properties of CSS codes constructed from classical cyclic codes that enable the construction of a flag fault-tolerant error correction scheme are proved, and a fault-tolerant protocol along with a family of circuits for flag fault-tolerant error correction are given. The flag-FTEC protocol proposed in this work requires only four ancilla qubits and applicable to CSS codes of distance 3 constructed from classical cyclic codes.

Acknowledgements

This work cannot be done without helps of my supervisor, Prof. Debbie Leung, and my collaborator, Christopher Chamberland. I gratefully acknowledge them for their helpful suggestions and comments. Also, I would like to extend thanks to Prof. Roger Melko, Prof. Bei Zeng, and Prof. Raymond Laflamme for being parts of advisory committee and defense committee. My master's study is supported by the Queen Sirikit Scholarship under The Royal Patronage of Her Majesty Queen Sirikit of Thailand. Lastly, I would like to thank my family and friends for their love and encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

Information technology has been improved drastically in recent decades. Computing devices become a part of our daily life, and they are getting more powerful every year. More powerful computation is a result from the fact that the number of transistors on a processor chip is getting larger since the size of each transistor is getting smaller. However, it is predicted that the development of transistor will reach its limit soon. This is because quantum effects of the system increase as the size of system decreases, and some bad effects such as quantum tunneling can cause the operation of transistors to fail. In order to improve the power of computation beyond this limit, a new platform of computation must be developed.

The platform of quantum computer is a strong candidate for a computer of the next generation. In quantum theory, some extraordinary properties such as superposition of states and quantum entanglement have been proposed and experimentally observed. With these properties, the way that information is stored in a quantum system is much different from that of a classical system. The studies of quantum information can lead to new technologies such as quantum cryptography, quantum communication, and quantum computation.

The idea of quantum computer was first proposed in 1980's. However, it has become public interest after the breakthrough of Shor's factorization algorithm in 1994 [18]. The algorithm suggests that solving the factorization problem by a quantum computer can be much faster than solving by a classical computer. In particular, the factorization problem can be solved in polynomial time using quantum computer while a classical algorithm for the same problem works in exponential time. Since the breakthrough, many studies have shown that quantum computation can outperform classical computation in a large class of problems.

Although many quantum algorithms promise more efficient computation compared to their classical counterparts, the physical implementation of such algorithms can be difficult. This is because the natures of classical and quantum systems are different. A good physical system for universal quantum computation must meet the requirements proposed by DiVincenzo in [6] as followings:

1. The system must be a scalable system with well-characteristic qubit.
2. The system must have the ability to initialize the state of the qubits to a simple state.
3. The system must have decoherence time much longer than the gate operation time.
4. A universal set of quantum gates can be implemented in the system.
5. The system must have a qubit-specific measurement capability.

Various types of physical systems have been proposed to be a good candidate for universal quantum computation. Some examples of candidate systems are nuclear magnetic resonance (NMR), atomic traps, superconducting circuits, and quantum optics.

In the physical implementation of quantum circuits, reducing error rate is a big concern. Several experimental techniques have been introduced to cope with errors. However, even small errors can propagate through the circuit and become worse in the later part of computation, causing an algorithm to fail. One of mathematical techniques which can be used to deal with errors is quantum error correction (QEC). This can be done by encoding quantum data into a codeword of a quantum error correcting code (QECC), then errors will be detected and corrected. By code concatenation, i.e., encoding the data to many levels, it is possible to make error rate arbitrarily small.

Even though QEC can be used to reduce errors during computation, the QEC itself can be faulty in a non-ideal situation. Moreover, if the total number of errors grows beyond the capability of a QECC, the code may not correct errors properly. The theory of fault tolerance has been developed to solve these problems. In the theory, quantum gadgets are used to simulate the operations of ideal circuit elements such as quantum gate operation, state preparation, qubit measurement, and error correction. Such gadgets must satisfy several conditions in order to be fault tolerant. These conditions ensure that a fault-tolerant gadget will work properly and the errors from its faults will not accumulate beyond the capability of the code. One of important results from the theory of fault tolerance is the threshold theorem. It suggests that arbitrarily long quantum computation is possible if the error rate of a physical system is below some threshold value depending on the fault-tolerant protocols being used.

Fault-tolerant error correction (FTEC) protocol is an important part of the fault-tolerant simulation of a quantum circuit since it will be applied as frequently as possible to prevent error accumulation. In general, some ancilla qubits are required in the FTEC protocol, and the number of required qubits grows exponentially as the number of concatenation levels increases. This can be a big challenge for a real-world experiment since the number of reliable qubits are very limited in practice. For this reason, finding a FTEC protocol which requires only small number of ancilla qubits can be an interesting task.

Recently, a new technique of FTEC called flag error correction has been proposed in [5, 4]. In the flag-FTEC protocol, some ancilla qubits are used to detect faults in the protocol which can cause errors of higher weight. The flag-FTEC protocols in [5, 4] require only small number of ancilla qubits and can be applied to several families of QECCs. The main goal of this work is to extend the technique of flag-FTEC to another family of QECCs, which is the family of CSS codes constructed from two classical cyclic codes.

This work is organized as follows: first, the basics of QEC will be discussed in Chapter 2. Afterwards, some classical linear codes and the construction of quantum codes from classical codes will be explained in Chapter 3. The theory of fault tolerance and flag FTEC will be described later in Chapter 4. Finally, the flag-FTEC protocol for cyclic CSS codes which is the main result of this work will be discussed in Chapter 5.

Chapter 2

Quantum Error Correction

In classical computation, the interaction between a computer and environment may cause some errors. It is possible that the error from one process may affect other subsequent processes, resulting in incorrect outcome at the end of the computation. One of the techniques that can be used to prevent errors from spreading throughout the computation is error correction. This can be done by encoding a bit of information into a larger string of physical bits so that if the number of occurred error is small enough, the remaining information is sufficient to recover the original data. The idea of error correction for quantum settings is quite similar. However, some techniques from classical error correction cannot be directly applied because of the differences in the nature of physical systems being used.

In this chapter, we will start by describing some basics of quantum error correction including the classical 3-bit repetition code, the 9-qubit code, and the criteria for quantum error correction in Section 2.1. Afterwards, the stabilizer formalism which is a convenient way to describe QECCs will be introduced in Section 2.2. Finally, some examples of QECCs will be given in Section 2.3.

2.1 Basics of quantum error correction

2.1.1 3-bit repetition code

Before we study QECCs, it is good to know one of the most fundamental classical error-correcting codes, the 3-bit repetition code. Suppose that we would like to send a bit of information, either 0 or 1, through a classical channel. However, the channel can cause a bit-flip error which will transform 0 to 1 or vice versa with probability p and transmit the correct information with probability $1-p$. Our goal is to find a good coding strategy so that the probability of obtaining the wrong information is less than p. Consider the following encoding scheme:

$$0 \mapsto 000,$$
$$1 \mapsto 111.$$

After the sender transmits the data string through a noisy channel, the receiver will decode the string to the number that appear the most; for example, the string 001 will be decoded to 0. Suppose that a bit-flip error can occur on at most one bit, this decoding scheme will always give the correct encoded information. It is possible that the errors might occur on two bits or more. However, the probability of such case is $3(1-p)p^2+p^3$ which is less than p if $p \leq 1/2$. This means that if the channel is not too noisy, encoding the data using the 3-bit repetition code will reduce the error rate.

In the quantum settings, the smallest unit of information is called quantum bit or 'qubit'. A qubit can be represented by any two-level quantum system such as polarization of a photon or spin of an electron. Normally, we write two orthogonal states of a qubit as state $|0\rangle$ and state $|1\rangle$. The state of a qubit is different from the state of a classical bit; not just $|0\rangle$ or $|1\rangle$, a qubit can also be in a superposition state of $|0\rangle$ and $|1\rangle$. In general, a state of a qubit can be described by,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.1}$$

where α and β are arbitrary constants satisfying $|\alpha|^2 + |\beta|^2 = 1$. If we measure a superposition state of this form in 0/1 basis, we will obtain outcome 0 with probability $|\alpha|^2$ or obtain outcome 1 with probability $|\beta|^2$. After measurement, the state of a qubit will collapse to the state corresponding to the measurement outcome.

It is useful to define Pauli operators X, Y, and Z as following:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.2}$$

Please note that $Y = iXZ$. Normally, state $|0\rangle$ and $|1\rangle$ of a qubit are chosen to be the $+1$ and -1 eigenstates of operator Z, respectively. We sometimes call the error corresponding to the X operator as a bit-flip error and call the error corresponding to the Z operator as a phase-flip error.

Although the 3-bit repetition code can be used in classical error correction, it cannot be directly applied to the quantum settings because of the following reasons:

1. The no cloning theorem states that we cannot duplicate the unknown state of qubits [15]. Thus, encoding an arbitrary state $|\psi\rangle$ to $|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$ is impossible.

2. In the decoding procedure of the 3-bit repetition code, we have to measure the state of all three bits to see whether 0 or 1 appears the most. In quantum settings, however, the data to be encoded can be in the superposition state of 0 and 1. In that case, the measurement in 0/1 basis will destroy the superposition state, resulting in wrong decoding result.

3. In classical settings, the bit-flip error is the only type of errors that can occur. However, the errors in quantum settings are continuous; they can be errors corresponding to any linear operators.

In the next section, the 9-qubit code which is the first QECC will be described. The encoding and decoding schemes using in the 9-qubit code is quite similar to those of the 3-bit repetition code except some changes are introduced to solve the aforementioned issues.

2.1.2 The 9-qubit code

The 9-qubit code is the first QECC invented by Peter Shor in 1995 [16]. Let us consider the following encoding scheme:

$$|\bar{0}\rangle = \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right), \tag{2.3}$$

$$|\bar{1}\rangle = \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right). \tag{2.4}$$

Suppose that the input data is state $\alpha|0\rangle + \beta|1\rangle$ where $|\alpha|^2 + |\beta|^2 = 1$. This state will be encoded as $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$. Note that with this encoder, we did not clone the unknown quantum state. Thus, the encoding scheme does not contradict the no-cloning theorem. Information from each block of three qubits will be used to correct an X error occurred in that block while information from all three blocks will be used to correct a Z error. We will consider the X-error correction and Z-error correction separately as followings.

X-error correction:

Consider the case that the encoder is

$$|\tilde{0}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}, \tag{2.5}$$

$$|\tilde{1}\rangle = \frac{|000\rangle - |111\rangle}{\sqrt{2}}. \tag{2.6}$$

An arbitrary state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is encoded as,

$$|\tilde{\psi}\rangle = \alpha\left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right) + \beta\left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right). \tag{2.7}$$

Suppose that there is an X error on at most one qubit. The state $|\tilde{\psi}\rangle$ will be mapped to one of the following states

$$
\begin{aligned}
I|\tilde{\psi}\rangle &= \alpha\left(\frac{|000\rangle+|111\rangle}{\sqrt{2}}\right)+\beta\left(\frac{|000\rangle-|111\rangle}{\sqrt{2}}\right),\\
X_1|\tilde{\psi}\rangle &= \alpha\left(\frac{|100\rangle+|011\rangle}{\sqrt{2}}\right)+\beta\left(\frac{|100\rangle-|011\rangle}{\sqrt{2}}\right),\\
X_2|\tilde{\psi}\rangle &= \alpha\left(\frac{|010\rangle+|101\rangle}{\sqrt{2}}\right)+\beta\left(\frac{|010\rangle-|101\rangle}{\sqrt{2}}\right),\\
X_3|\tilde{\psi}\rangle &= \alpha\left(\frac{|001\rangle+|110\rangle}{\sqrt{2}}\right)+\beta\left(\frac{|001\rangle-|110\rangle}{\sqrt{2}}\right).
\end{aligned} \tag{2.8}
$$

where X_i is an X error on the i^{th} qubit. We would like to distinguish between these states so that the qubit on which the X error occurred can be determined. Anyhow, we should be aware that the distinguishing process must not give any information about the plus sign (in the $|\tilde{0}\rangle$ term) or minus sign (in the $|\tilde{1}\rangle$ term); otherwise the superposition will be destroyed. One of the techniques that can be used here is to measure the parity between the first and the second qubits, and the parity between the second and the third qubits. These measurement results can tell the location of the error without giving any information about the plus or minus signs. For example, if the parity measurement results tell us that the first and second qubits have the same parities and the second and the third qubits have different parities, the state must be $X_3|\tilde{\psi}\rangle$. Applying X_3 on the state will give the original state $|\tilde{\psi}\rangle$ which can be decoded using the inverse operation of the encoder. The error correction for other cases are similar. Please note that measuring the parity between the first and the second qubits and the parity between the second and the third qubits are equivalent to measuring eigenvalues of operators Z_1Z_2 and Z_2Z_3, respectively. A circuit for measuring bit parity is shown in Fig. 2.1. The actual encoding states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ as in Eq. (2.3) and Eq. (2.4) can be obtained by encoding $|0\rangle$ to $|\tilde{0}\tilde{0}\tilde{0}\rangle$ and $|1\rangle$ to $|\tilde{1}\tilde{1}\tilde{1}\rangle$. The X error correction can be applied in the same fashion on each block of three qubits in the 9-qubit code.

Z-error correction:

Consider the encoding of $|\bar{0}\rangle$ and $|\bar{1}\rangle$ as in Eq. (2.3) and Eq. (2.4). Suppose that a Z error occurs on at most one qubit. The plus sign in the block of $|\bar{0}\rangle$ on which the error occurred

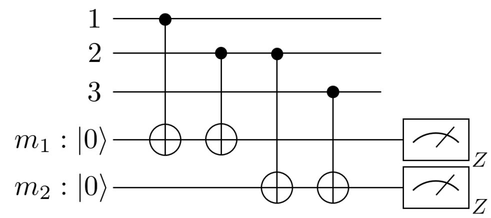
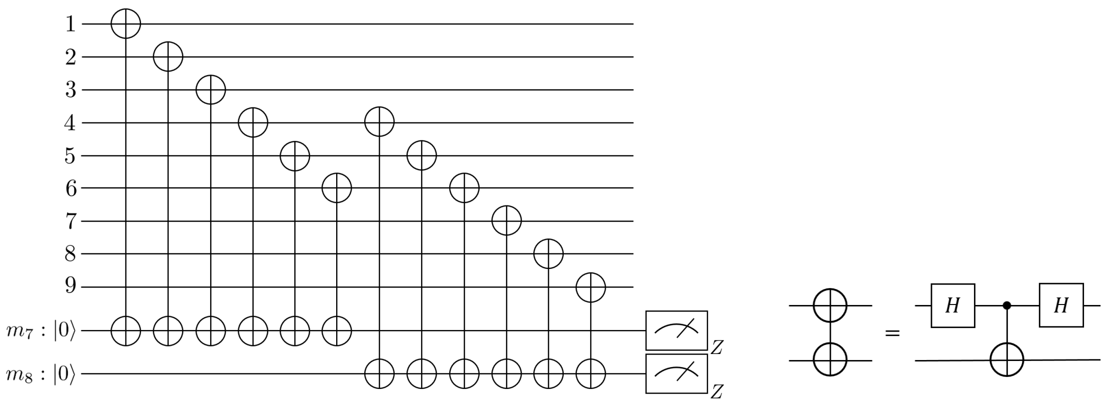


Figure 2.1: A circuit for measuring bit parities in each block of the 9-qubit code. In this figure, three data qubits and two ancilla qubits are shown. CNOT gates are used to couple the data qubits and the ancilla qubits, where the ancilla qubits are measured in Z basis (0/1 basis) at the end. The parity between the first and the second qubits is obtained by the measurement of ancilla m_0, and the parity between the second and the third qubits is obtained by the measurement of ancilla m_1.

will become a minus sign, and the minus sign in the same block of $|\bar{1}\rangle$ will become a plus sign. By measuring the phase parity between the first and the second blocks and the phase parity between the second and the third blocks, i.e., comparing the plus and minus signs between each block, we can determine the block in which Z error occurred. By applying Z operation on any qubit in the block in which the error occurred, the original state will be recovered. A circuit for measuring phase parities is shown in Fig. 2.2. Please note that measuring phase parities does not give any information about the encoding state, thus this error correction method can also be applied to a superposition state of the form $\alpha|\bar{0}\rangle+\beta|\bar{1}\rangle$. The measurements of the phase parity between the first and second blocks and the phase parity between the second and the third blocks are equivalent to the measurement of eigenvalues of $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$, respectively.

Error syndrome measurement

In the error correction procedure previously explained, the method of measuring bit parity or phase parity is called error syndrome measurement. This step can determine the type of occurred error and the qubit on which it occurred. The information from the full syndrome measurement will be used to recover the original encoded state. For the 9-qubit code, the operators whose eigenvalue is being measured in the full syndrome measurement are,



(a) (b)

Figure 2.2: A circuit for measuring phase parities in Z error correction of the 9-qubit code is shown in Fig. 2.2a, where a quantum gate involving X operator measurement is related to a CNOT gate as shown in Fig. 2.2b. The phase parity between the first and the second blocks is obtained by the measurement of m_7, and the phase parity between the second and the third blocks is obtained by the measurement of m_8.

$$\begin{array}{lccccccccc}
g_1: & Z & Z & I & I & I & I & I & I & I \\
g_2: & I & Z & Z & I & I & I & I & I & I \\
g_3: & I & I & I & Z & Z & I & I & I & I \\
g_4: & I & I & I & I & Z & Z & I & I & I \\
g_5: & I & I & I & I & I & I & Z & Z & I \\
g_6: & I & I & I & I & I & I & I & Z & Z \\
g_7: & X & X & X & X & X & X & I & I & I \\
g_8: & I & I & I & X & X & X & X & X & X
\end{array}.$$

The operators $g_1, \ldots, g_8$ correspond to the stabilizer generators of the 9-qubit code, discussed later in Section 2.2.

Now let us consider the case that Y error occurs on any qubit. Since $Y = iXZ$, it can be detected by both bit parity and phase parity measurements. Therefore, the original state can still be recovered using the information from the full syndrome measurement. Anyhow, a possible one-qubit error is not restricted to Pauli errors. In general, an arbitrary one-qubit operator A can be written as,

$$A = a_0 I + a_1 X + a_2 Y + a_3 Z, \tag{2.9}$$

for some constants a_0, a_1, a_2, a_3. This kind of errors will cause the ancilla qubits for the syndrome measurement to be in a superposition state. Fortunately, the state of all ancilla qubits will collapse to the state corresponding to either I, X, Y, or Z error occurred on one of the nine qubits after syndrome measurement. Therefore, the error will be discretized by the syndrome measurement, and the error correction can be performed as previously explained.

2.1.3 Criteria for quantum error correction

The 9-qubit code described in Section 2.1.2 is an example of QECCs for correcting arbitrary single-qubit error. We can also see that all possible one-qubit Pauli errors are distinguishable using syndrome measurement. (To be precise, Z errors in the same block are not distinguishable but they lead to the same syndromes and the error correction is still possible in that case.) In this section, we will formulate the criteria for general QECCs.

Observe that any two quantum states are perfectly distinguishable iff they are orthogonal, following the Holevo-Helstrom theorem [23]. For this reason, all distinguishable errors must preserve the orthogonality between two codewords. In particular, let $\{|c_1\rangle, |c_2\rangle, \ldots, |c_k\rangle\}$ be a basis of coding subspace and let $\mathcal{E}_c$ be the set to all correctable errors. It is required that,

$$\langle c_i | E_a^\dagger E_b | c_j \rangle = 0, \tag{2.10}$$

for all $E_a, E_b \in \mathcal{E}_c$ and for all $i, j \in \{1, \ldots, k\}$ where $i \neq j$. This is the first condition of QEC. The second condition is that the outcome of the syndrome measurement must not give any information about the codewords, otherwise the superposition state will collapse.

This property is can be written as,

$$\langle c_i|E_a^\dagger E_b|c_i\rangle = \langle c_j|E_a^\dagger E_b|c_j\rangle, \tag{2.11}$$

for all $E_a, E_b \in \mathcal{E}_c$ and for all $i, j \in \{1, \ldots, k\}$. The criteria Eq. (2.10) and Eq. (2.11) can be combined to the following equation:

$$\langle c_i|E_a^\dagger E_b|c_j\rangle = C_{ab}\delta_{ij}, \tag{2.12}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$ [12]. Here C_{ab} are constants independent of the codewords. Since $\langle c_i|E_a^\dagger E_b|c_i\rangle = (\langle c_i|E_b^\dagger E_a|c_i\rangle)^*$ for all i, we may write C_{ab} as a Hermitian matrix. If C_{ab} has maximal rank, we say that the code is *non-degenerate*. Otherwise, if C_{ab} has non-maximal rank, we say that the code is *degenerate*. A non-degenerate code refers to the case that each error in the set $\mathcal{E}_c$ corresponds to a unique syndrome, while in the case of degenerate code, there exist two errors in $\mathcal{E}_c$ whose syndromes are the same. For example, the 9-qubit code is a degenerate code since the errors Z_1, Z_2, and Z_3 all have the same syndromes. Anyhow, applying Z operation on any qubit in the first block can correct these errors.

For any n-qubit Pauli operator, the *weight* $\mathrm{wt}(E)$ of Pauli operator E is defined as a number of non-identity tensor factors of E, and the *distance* of the code is the minimum weight of Pauli operators that do not satisfy Eq. (2.12). For the code of distance d, the syndromes of all errors of weight less than d except identity operator are non-trivial, thus the code can detect up to weight $d-1$ errors. Now let us consider an QECC that corrects up to weight-t errors. For any two errors $E_a, E_b \in \mathcal{E}_c$, the error $E = E_a^\dagger E_b$ is also in $\mathcal{E}_c$. Thus, the distance of the code which can correct up to weight-t errors is at least $d = 2t+1$.

2.1.4 Quantum Hamming bound

We have seen in Section 2.1.2 that the 9-qubit code encodes one logical qubit to nine physical qubits and can correct up to one error. Let T be a quantum code that can correct up to t errors. One may ask, “what is the smallest number of required qubits for such T?”.

For non-degenerate codes, the question can be answered by the following analysis.

Suppose that quantum code T is a non-degenerate code that encodes k logical qubits into n physical qubits and can correct up to t errors. There will be $\sum_{s=0}^{t} 3^s \binom{n}{s}$ possible Pauli errors. We need to distinguish all erroneous basis states from the original basis states of the coding subspace as described in Section 2.1.3. Here the identity operator is also considered as an error corresponding to the case of $s = 0$. Thus, the correctable error set will consist of $\sum_{s=0}^{t} 3^s \binom{n}{s}$ Pauli errors including identity operator. The dimension of the original coding subspace is 2^k, which means that the dimension of the erroneous coding subspace is $\left(\sum_{s=0}^{t} 3^s \binom{n}{s}\right) 2^k$. The coding subspace must contain in the Hilbert space $\mathcal{H}_{2^n}$. Therefore, it holds that,

$$\left(\sum_{s=0}^{t} 3^s \binom{n}{s}\right) 2^k \leq 2^n,$$

or

$$\sum_{s=0}^{t} 3^s \binom{n}{s} \leq 2^{n-k}. \tag{2.13}$$

Eq. (2.13) is called quantum Hamming bound [7]. For a quantum error-correcting code with $k = 1$ that can correct up to 1-qubit error, Eq. (2.13) is simplified to,

$$3n + 1 \leq 2^{n-1}. \tag{2.14}$$

The smallest n that satisfies Eq. (2.14) is 5, which is means that every quantum error-correcting code must have at least 5 qubits. The 5-qubit code which is the smallest QECC will be described later in Section 2.3.3.

2.2 Quantum error correction in stabilizer formalism

In previous sections, we consider the error correction properties by looking at the code-words of the quantum error-correcting codes. However, this may not be inconvenient in calculation because the codewords might consist of many terms, in general. In this section, we will formulate another way to describe the QECC in terms of Pauli operators that fix the coding subspace. This formalism is called *stabilizer formalism* [8].

2.2.1 Pauli group

Before we explain the stabilizer formalism, it is good to know the Pauli group and some of its properties. The Pauli group is defined as follows.

Definition 2.1 *The* 1-qubit Pauli group $\mathcal{P}_1$ *is a set of* I, X, Y *and* Z *with an overall phase of* ± 1 *or* $\pm i$*. That is,*

$$\mathcal{P}_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \tag{2.15}$$

The n-qubit Pauli group $\mathcal{P}_n$ *is a set of tensor products of elements in* $\mathcal{P}_1$ *on* n *qubits,*

$$\mathcal{P}_n = \{P_1 \otimes \cdots \otimes P_n : P_i \in \mathcal{P}_1 \ \text{for all} \ i = 1, \ldots, n\}. \tag{2.16}$$

The followings are some important properties of the Pauli group:

1. Pauli group is a group. This means that for all $P_1, P_2 \in \mathcal{P}_n$, Operator $P_1 P_2$ is also in $\mathcal{P}_n$.

2. Operator $P \in \mathcal{P}_n$ is either Hermitian ($P^\dagger = P$) or anti-Hermitian ($P^\dagger = -P$); here we already know that the Pauli operators (I, X, Y and Z) are all Hermitian. However, since we allow a complex phase in the Pauli group, some elements might be anti-Hermitian; e.g., $(iX)^\dagger = -iX$.

3. $P^2 = \pm I$ for all $P \in \mathcal{P}_n$ since all elements in Pauli group are unitary ($P^\dagger = P^{-1}$). For Hermitian elements we have $P^2 = P^\dagger P = I$, and for anti-Hermitian elements we have $P^2 = (-P^\dagger)P = -I$.

4. Any two operators $P_1, P_2 \in \mathcal{P}_n$ either commute ($[P, Q] = PQ - QP = 0$) or anti-commute ($\{P, Q\} = PQ + QP = 0$). This is from the fact that any two non-trivial

Pauli operators acting on the same qubit anticommute, and any two Pauli operators acting on the different qubits trivially commute.

2.2.2 Definition and properties of stabilizer

Stabilizer is a group of Pauli operators that stabilize the codewords of the QECC. The definition of stabilizer is as follows:

Definition 2.2 *Let $T \subseteq \mathcal{H}_{2^n}$ be a coding subspace. The* stabilizer *S corresponding to T is*

$$S(T) = \{M \in \mathcal{P}_n : M|\psi\rangle = |\psi\rangle \ \textit{for all} \ |\psi\rangle \in T\}. \tag{2.17}$$

Note that every valid codeword $|\psi\rangle \in T$ is a $+1$ eigenvector of all operators in S. For a given set S of Pauli operators, it is also possible to define a coding subspace $\mathcal{T}(S)$ where all states in $\mathcal{T}(S)$ are stabilized by all element of S.

Definition 2.3 *Let $S \subseteq \mathcal{P}_n$ be an Abelian subgroup of $\mathcal{P}_n$. The coding subspace $\mathcal{T}(S) \subseteq \mathcal{H}_{2^n}$ corresponding to S is defined as,*

$$\mathcal{T}(S) = \{|\psi\rangle : M|\psi\rangle = |\psi\rangle \ \textit{for all} \ M \in S\}. \tag{2.18}$$

In general, T is a subset of $\mathcal{T}(S(T))$. In case that T is a QECC with $T = \mathcal{T}(S(T))$, it is called a *stabilizer code*.

Here are some interesting properties of stabilizer:

1. Stabilizer is a group; if M and N are operators in S, then MN is also an operator in S since MN also fix all states $|\psi\rangle \in T$. That is,

$$MN|\psi\rangle = M|\psi\rangle = |\psi\rangle. \tag{2.19}$$

   We can easily verify that $I \in S$, and $M^{-1} \in S$ for all $M \in S$.

2. Stabilizer is Abelian; observe that

$$MN|\psi\rangle = NM|\psi\rangle = |\psi\rangle, \tag{2.20}$$

for all $M, N \in S$. Therefore, $(MN - NM)|\psi\rangle = 0$ for all $|\psi\rangle \in T$. This implies that $[M, N] = 0$.

3. $-I$ is not in the stabilizer since the eigenvalues of $-I$ are all -1's and the operator cannot fix any states. The consequence of this property is that we cannot have the same Pauli operators with different phases (such as X and $-X$) in the stabilizer since the multiplication of such operators will be an operator outside the group.

4. The size of the stabilizer S is $|S| = 2^r$ where r is some positive integer since any operator $M \in S$ can be written in the form $g_1^{a_1} g_2^{a_2} \cdots g_r^{a_r}$ for some choices of $\{g_1, g_2, \ldots, g_r\}$ where $a_i \in \{0, 1\}$. This is the consequences of Property 1-3. The operators $g_1, g_2, \ldots, g_r$ are called generators of the stabilizer S.

5. The dimension of the coding subspace is $\dim(T(S)) = 2^{n-r}$. Let $\{g_1, \ldots, g_r\}$ be a generating set of stabilizer S. Observe that any generator g_i is a non-identity Pauli operator. Thus, half of its eigenvalues are 1's and another half are -1's. Since all valid codewords are $+1$ eigenvectors of all generators, adding one more generator to the generating set of S results in dividing the dimension of the coding subspace by 2. In addition, $k = n - r$ is the number of logical qubits that can be encoded into the stabilizer code of n physical qubits.

2.2.3 Syndrome measurement in stabilizer formalism

In the description of the 9-qubit code in Section 2.1.2, the information of errors occurred on the codeword is carried out by syndrome measurement. In this section, the explanation of the syndrome measurement will be rephrased in the stabilizer formalism.

Consider any Pauli error $E \in \mathcal{P}_n$, operator $M \in S$, and state $|\psi\rangle \in T(S)$. We know that either $[E, M] = 0$ or $\{E, M\} = 0$ from the property of the Pauli group. Observe that

$[E, M] = 0$ iff

$$ME|\psi\rangle = EM|\psi\rangle = E|\psi\rangle, \tag{2.21}$$

for all $|\psi\rangle \in T(S)$. In this case, $E|\psi\rangle$ is a $+1$ eigenvector of M. On the other hand, $\{E, M\} = 0$ iff

$$ME|\psi\rangle = -EM|\psi\rangle = -E|\psi\rangle, \tag{2.22}$$

for all $|\psi\rangle \in T(S)$. The state $E|\psi\rangle$ is a -1 eigenvector of M in this case. From the fact that valid codewords are $+1$ eigenvectors of all elements in S, the Pauli error E is detectable iff it anticommutes with at least one element of S. However, since any element in the stabilizer can be written as a product of generators $g_1, \ldots, g_r$, E is detectable iff it anticommutes with at least one generator of S.

The formal definition of error syndrome is as follows:

Definition 2.4 *Let S be a stabilizer with generators $g_1, g_2, \ldots, g_r$. The* error syndrome *$s(E) \in \{0, 1\}^r$ of a Pauli operator $E \in \mathcal{P}_n$ is an r-bit string with $s(E)_i = 0$ if $[E, g_i] = 0$ and $s(E)_i = 1$ if $\{E, g_i\} = 0$, where $s(E)_i$ is the i^{th} component of $s(E)$.*

From Definition 2.4, we can easily prove that $s(EF) = s(E) \oplus s(F)$ for all operators $E, F \in \mathcal{P}_n$, where $\oplus$ is addition modulo 2.

In Section 2.1.2, we described that the syndrome measurement corresponds to the measurement of eigenvalues of the stabilizer generators. Note that this is equivalent to determining the commutation and anticommutation relations between the error occurred on a codeword and the stabilizer generators. For a non-degenerate stabilizer code, all correctable errors must have unique syndromes.

2.2.4 Normalizer

From the previous section, we know that error $E \in \mathcal{P}_n$ is detectable iff it anticommutes with at least one generator of S. In contrast, if $[E, g_i] = 0$ for all $i \in \{1, \ldots, r\}$, the error is not detectable. Suppose that $E \in S$, we find that error correction is not necessary since the error does not change the codeword. The interesting case is the case that E commutes with all generators but E is not in S. It is possible to define a set of operators that commute with all elements in the stabilizer as follows:

Definition 2.5 *Let S be a stabilizer. The* normalizer *$N(S)$ is a set of operators that commute with all elements of S; i.e.,*

$$N(S) = \{A \in \mathcal{P}_n : AM = MA \ \text{for all} \ M \in S\}. \tag{2.23}$$

Since S is Abelian, it holds that $S \subseteq N(S)$.

In Section 2.1.3, the distance of the QECC is given by the minimum weight of Pauli operators that do not satisfy the error correction condition. For a stabilizer code, the distance can be defined as follows:

Definition 2.6 *Let S be a stabilizer and $N(S)$ be the normalizer of S. The* distance *d of the stabilizer code corresponding to S is the minimum weight of Pauli operators in $N(S) - S$.*

Consider Pauli operator $A \in N(S)$, stabilizer element $M \in S$, and valid codeword $|\psi\rangle \in T$. The operation of A on $|\psi\rangle$ gives,

$$A|\psi\rangle = AM|\psi\rangle = MA|\psi\rangle. \tag{2.24}$$

Here we can see that the state $A|\psi\rangle$ is a $+1$ eigenvector of any operator $M \in S$. Thus, the state $A|\psi\rangle$ is also a valid codeword. If $A \in S$, it acts trivially on the codeword. However, if $A \notin S$, the operation of A maps a valid codeword $|\psi\rangle$ to another valid codeword $A|\psi\rangle$. Here we can interpret operators in the normalizer $N(S)$ as *logical operators*. If operators A and B in the normalizer map a valid codeword in the same way, i.e., $A|\psi\rangle = B|\psi\rangle$, we say that A and B are *logically equivalent*.

2.2.5 Cosets inside normalizer and Pauli group

Let S be a stabilizer with generating set $\{g_1, \ldots, g_r\}$. From the fact that any two Pauli operators either commute or anticommute, we know that every Pauli operator must correspond to some error syndrome in $\{0,1\}^r$. It is possible to partition a Pauli group into subsets where operators in each subset have the same syndromes. Now let us consider the error syndrome of operator $A \in N(S)$. Since A commutes with all generators, we have that $s(A) = 0$ for all $A \in N(S)$. Using the fact that $s(EF) = s(E) + s(F)$ for all operators $E, F \in \mathcal{P}_n$, we find that multiplying operator $E \in \mathcal{P}_n$ with operator $A \in N(S)$ results in another operator which have the same error syndrome as E. Therefore, the quotient group $\mathcal{P}_n/N(S)$ and $\{0,1\}^r$ are isomorphic. This can be rephrased into the following preposition:

Proposition 2.1 *Let $E, F \in \mathcal{P}_n$ be Pauli operators and let S be a stabilizer. Then, E and F are in the same coset of $N(S)$ iff E and F have the same syndrome.*

Proof:
($\Rightarrow$) Assume that E and F are in the same coset of $N(S)$, then we can write $F = EA$ for some operator $A \in N(S)$. Let $g_1, \ldots, g_r$ be generators of S. The commutation relation between F and g_i is,

$$[F, g_i] = [EA, g_i] = EAg_i - g_iEA = Eg_iA - g_iEA = [E, g_i]A. \tag{2.25}$$

Hence, $[F, g_i] = 0$ iff $[E, g_i] = 0$. This is true for all generators g_i. Therefore, E and F have the same error syndromes.

($\Leftarrow$) Assume that E and F have the same error syndromes. Let $A = E^\dagger F$. We have that $[A, g_i] = 0$ for all generators g_i. Since A commutes with all generators, we know that A is in $N(S)$. Therefore, E and $F = EA$ are in the same coset of $N(S)$. $\square$

Form the fact that $\mathcal{P}_n/N(S) \cong \{0,1\}^r$, we have that

$$|N(S)| = 4 \cdot 2^{n+k}, \tag{2.26}$$

where $k = n - r$ is the number of logical qubits.

Now let us consider the coset of S inside $N(S)$. Since any operator $M \in S$ acts trivially on valid codeword $|\psi\rangle$ and any operator $A \in N(S)$ acts on $|\psi\rangle$ as a logical operator, multiplying A with M will give another operator which is logically equivalent to A. With similar analysis, we can partition the normalizer into subsets where operators in the same subset are logically equivalent. This can be described by the following proposition:

Proposition 2.2 *Let $A_1, A_2 \in N(S)$ be operators in the normalizer of stabilizer S. Then, N_1 and N_2 are in the same coset of S iff $N_1|\psi\rangle = N_2|\psi\rangle$ for all codewords $|\psi\rangle \in T(S)$.*

Proof:
($\Rightarrow$) Assume that N_1 and N_2 are in the same coset of S, then we can write $N_2 = N_1 M$ for some operator $M \in S$. Thus we have,

$$N_2|\psi\rangle = N_1 M|\psi\rangle = N_1|\psi\rangle, \tag{2.27}$$

for all $|\psi\rangle \in T(S)$.

($\Leftarrow$) Assume that $N_1|\psi\rangle = N_2|\psi\rangle$. Let $M = N_1^\dagger N_2$. We find that $M|\psi\rangle = |\psi\rangle$. Thus, M is in the stabilizer. This implies that N_1 and $N_2 = N_1 M$ are in the same coset of S. □

From Eq. (2.26) and the size of S explained in Section 2.2.2, we have that the number of cosets in $N(S)$ is $4 \cdot 4^k$. The quotient group $N(S)/S$ is in fact isomorphic to the Pauli group $\mathcal{P}_k$. This agrees with the fact that all operators in the same coset are logically equivalent.

2.2.6 Binary symplectic representation

In previous sections, we can see that many descriptions of QECCs in stabilizer formalism are related to commutation and anticommutation relations between operators. We also know that every Pauli operator can be written as a product of X-type operator and Z-type operator with some phase factor, for example, $Y = iXZ$. However, the phase factor does not involve in the the commutation and anticommutation relations. For this reason, it is sometimes more convenient to represent Pauli operator P in terms of a binary vector representing the number of X-type and Z-type factors of P acting on each qubit and ignore the phase factor. This representation of Pauli operators is called binary symplectic representation. The formal definition is as follows:

Definition 2.7 *Let $P \in \mathcal{P}_n$ be a Pauli operator and let v_P be a vector in $\{0,1\}^n \times \{0,1\}^n$. The* binary symplectic representation *of $\mathcal{P}_n/\{\pm 1, \pm i\}$ is an isomorphism between $\mathcal{P}_n/\{\pm 1, \pm i\}$ and $\{0,1\}^n \times \{0,1\}^n$: $P \leftrightarrow v_P = (x_P|z_P)$. Let $P = P_1 \otimes \cdots \otimes P_n$ where P_i is $I, X, Y,$ or Z. The i^{th} component of x_P is 1 if P_i is X or Y and 0 if P_i is I or Z. The i^{th} component of z_P is 1 if P_i is Y or Z and 0 if P_i is I or X.*

For example, an operator $P = X \otimes Y \otimes Z$ can be written in symplectic form as $v_P = (1\ 1\ 0|0\ 1\ 1)$. Note that the phase factor of a Pauli operator is ignored.

In binary symplectic representation, the multiplication of two Pauli operators P and Q is equivalent to the addition of binary vectors v_P and v_Q. In addition, the commutation and anticommutation relations between two Pauli operators can be written as a symplectic product as follows:

Definition 2.8 *Let $v_1 = (x_1|z_1)$ and $v_2 = (x_2|z_2)$ be vectors in $\{0,1\}^n \times \{0,1\}^n$. A* symplectic product *between v_1 and v_2 is $v_1 \odot v_2 = (x_1 \cdot z_2) \oplus (x_2 \cdot z_1)$, where dot product is the scalar product and $\oplus$ is addition modulo 2.*

Proposition 2.3 *Let $P, Q \in \mathcal{P}_n$ be Pauli operators with binary symplectic representation v_P and v_Q, respectively, and let $c : \mathcal{P}_n \times \mathcal{P}_n \to \{0,1\}$ be a function,*

$$c(P,Q) = \begin{cases} 0 & \text{if } [P,Q] = 0, \\ 1 & \text{if } \{P,Q\} = 0. \end{cases}$$

Then, $v_P \odot v_Q = c(P,Q)$.

Throughout this thesis, we will sometimes use the regular representation and the symplectic representation interchangeably. In Chapter 3, the symplectic representation will be useful in the construction of quantum codes from classical codes.

2.3 Examples of stabilizer codes

Stabilizer code is a quantum code that can be represented by Pauli operators which stabilize the coding subspace. Not every quantum code is a stabilizer code. However, if the code is a stabilizer code, the analysis of error correction properties can be much easier. For a stabilizer code, the error detection can be considered by the commutation and anticommutation relations between errors and stabilizer generators, as explained in Section 2.2.3. It is good to know a common notation for a stabilizer code as following.

Definition 2.9 *Let T be a stabilizer code which encodes k logical qubits to n physical qubits and let d be the distance of T. We say that T is an $[\![n,k,d]\!]$ code.*

In this section, some examples of stabilizer codes will be described.

2.3.1 The 9-qubit code

Previously described in Section 2.1.2, the 9-qubit code can be represented by a codeword basis $\{|\bar{0}\rangle, |\bar{1}\rangle\}$ as in Eq. (2.3) and Eq. (2.4). Moreover, any vector in the coding subspace spanned by $|\bar{0}\rangle$ and $|\bar{1}\rangle$ is an $+1$ eigenvector of all operators corresponding the error syndrome measurement. For this reason, the stabilizer S corresponding to the 9-qubit code is generated by operators $g_1, \ldots, g_8$, where

$$\begin{array}{cccccccccc}
g_1: & Z & Z & I & I & I & I & I & I & I \\
g_2: & I & Z & Z & I & I & I & I & I & I \\
g_3: & I & I & I & Z & Z & I & I & I & I \\
g_4: & I & I & I & I & Z & Z & I & I & I \\
g_5: & I & I & I & I & I & I & Z & Z & I \\
g_6: & I & I & I & I & I & I & I & Z & Z \\
g_7: & X & X & X & X & X & X & I & I & I \\
g_8: & I & I & I & X & X & X & X & X & X
\end{array}. \tag{2.28}$$

The 9-qubit code is a $[\![9,1,3]\!]$ code. Let E be any Pauli operator of weight 1. We can easily verify that E anticommutes with at least one generator, thus all errors of weight 1 are detectable. In addition, all syndromes of weight-1 Pauli errors are distinct. This means that the 9-qubit code can correct up to 1-qubit error as we expected. By applying weight-1 Pauli operator corresponding to the syndrome measurement, we can recover the original encoded state.

Now let us consider a Pauli error of weight 2. We can also verify that it anticommutes with at least one generator, therefore it is detectable. However, some weight-2 Pauli operators have the same syndromes as that of weight-1 Pauli errors. For example, X_1X_2 and X_3 have the same syndromes $(0, 1, 0, 0, 0, 0, 0, 0)$. Suppose that the error X_1X_2 occurs, the error syndrome will tell us to apply X_3 to the data for error correction. This results in error $X_1X_2X_3$ on the data. We may say that an error of weight-2 is detectable but not correctable in this case.

There are some Pauli operators of weight-3 that commute with all generators, for example, $X_1X_2X_3$. This means that such Pauli operators are in the normalizer $N(S)$. By the definition of the distance of a stabilizer code, the distance of the 9-qubit code is 3.

2.3.2 The 7-qubit code

The 7-qubit code is a stabilizer code where the stabilizer S is generated by the following generators:

$$
\begin{array}{cccccccc}
g_1: & I & I & I & X & X & X & X \\
g_2: & I & X & X & I & I & X & X \\
g_3: & X & I & X & I & X & I & X \\
g_4: & I & I & I & Z & Z & Z & Z \\
g_5: & I & Z & Z & I & I & Z & Z \\
g_6: & Z & I & Z & I & Z & I & Z
\end{array}. \tag{2.29}
$$

Since the number of physical qubits is $n = 7$ and the number of generators is $r = 6$, this code can encode $k = n - r = 1$ qubit. Similar to the argument previously described for the 9-qubit code, we can verify that this code can correct up to weight-1 Pauli errors and the distance of the code is 3. Thus, the 7-qubit code is a $[\![7,1,3]\!]$ code.

We can see from Eq. (2.29) that the X-type and and Z-type generators are of the same form. In fact, the 7-qubit is constructed from the 7-bit classical Hamming code with the following parity check matrix,

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{2.30}$$

The 7-qubit code is a code in the family of quantum Hamming codes, which will be described later in Section 3.3. The construction of a quantum code from classical codes will be discussed in details in Chapter 3.

2.3.3 The 5-qubit code

The 5-qubit code is a stabilizer code whose stabilizer generators are

$$\begin{array}{rccccc} g_1: & X & Z & Z & X & I \\ g_2: & I & X & Z & Z & X \\ g_3: & X & I & X & Z & Z \\ g_4: & Z & X & I & X & Z \end{array}. \tag{2.31}$$

The 5-qubit code is the smallest QECC which satisfies the quantum Hamming bound, discussed in Section 2.1.4. This code is a $[\![5,1,3]\!]$ code. It can correct up to 1-qubit errors and the code distance is 3. In addition, this code is cyclic in the sense that a cyclic permutation of any stabilizer element is another stabilizer element.

Chapter 3

Constructing Quantum Codes from Classical Codes

When quantum data is sent through a noisy channel, there is a probability that the noise from the channel may cause errors on the data so badly that it is unable to be recovered. By encoding the data using a QECC, some small errors can be corrected and the success probability to recover the original data is improved. In Chapter 2, we have seen that some QECCs can be represented by Pauli operators that fix the coding subspace. Such QECCs are call stabilizer codes. There are many families of stabilizer codes, and some of them may have advantages or disadvantages compared to others. Constructing a new family of stabilizer codes can be a challenging task.

In this chapter, we will consider the construction of stabilizer codes from classical linear codes. The chapter is organized as follows: first we will study basics of classical linear codes in Section 3.1. Some examples of classical linear codes will be discussed in details in Section 3.2. Finally, CSS construction which is the construction of a stabilizer code from two classical linear codes will be explained in Section 3.3.

3.1 Classical linear codes

A classical code is a set of bit-strings or codewords. Throughout this work, we will only discuss a binary code where its codewords are strings of 0's and 1's. In particular, if a classical code C has a property that the sum of any two codewords in C is also a codeword in C, the code is a classical linear code. The definition of classical linear code is as follows:

Definition 3.1 *Let u, v be codewords in $\{0,1\}^n$. A* classical linear code *C of length n is a code in $\{0,1\}^n$ for which, whenever $u, v \in C$, $au + bv \in C$ for all $a, b \in \{0,1\}$. That is, C is a linear subspace of $\{0,1\}^n$.*

Similar to quantum codes, if C is a classical linear code of length n with $\dim(C) = k$, then C encodes k logical bits to n physical bits.

Consider any classical linear code C, we know from Definition 3.1 that C is actually a linear subspace of $\{0,1\}^n$. This means that it is possible to represent C by a set of vectors spanning C. In particular, C can be represented by a generator matrix, given by the following definition:

Definition 3.2 *Let C be a classical linear code encoding k logical bits to n physical bits. A* generator matrix *G of the code C is a $k \times n$ binary matrix such that the rows of G are basis vectors of C.*

Note that the choice of generator matrix for C is not unique.

It is also possible to describe a classical linear code by a matrix determining the error correction property of the code. Such matrix is called a parity check matrix, defined as follows:

Definition 3.3 *Let C be a classical linear code encoding k logical bits to n physical bits. A* parity check matrix *H of the code C is a matrix consisting of rows w_i where $\{w_i\}$ is a maximal linearly independent set of vectors satisfying $w_i \cdot v = 0$ for all $v \in C$.*

By definition, it holds that $Hv^T = 0$ for all $v \in C$, which implies that $HG^T = 0$. Let w_i be a row of H. We also have that $Gw_i^T = 0$, which is a system of k linear equations on n

bits. The solution space of this system has dimension $n-k$. This implies that the parity check matrix H is an $(n-k)\times n$ matrix.

Let v be a codeword in C. Suppose that there is an error $e\in\{0,1\}^n$ on v, the erroneous codeword is $v+e$. Since we have $vH^T=0$, error e is detectable iff $eH^T\neq 0$. We can define the error syndrome of a vector e as following:

Definition 3.4 *Let C be a classical linear code with parity check matrix H, and let $e\in\{0,1\}^n$ be an error vector. The* error syndrome $s(e)$ *of vector e is $s(e)=eH^T$.*

Suppose that erroneous codeword $v+e$ is given. Note that

$$(v+e)H^T = 0 + eH^T = s(e), \tag{3.1}$$

for every $v\in C$. This means that $s(e)$ provides information of the error regardless of the codeword it acts on. Therefore, the information from the error syndrome can be used in error correction. In particular, let $\mathcal{E}\subseteq\{0,1\}^n$ be a set of all correctable errors. It is required that the syndromes $s(e_i)$ of all errors $e_i\in\mathcal{E}$ are distinct.

Given a classical linear code C, observe that all vectors which are orthogonal to all $v\in C$ form a new subspace. Thus, the set of such vectors is also a classical linear code. Such code is called the dual code of C. The formal definition of dual code is as follows:

Definition 3.5 *The* dual code $C^\perp$ *of a linear code C is*

$$C^\perp = \{w\in\{0,1\}^n : v\cdot w = 0 \ \text{for all} \ v\in C\}. \tag{3.2}$$

It holds that $(C^\perp)^\perp = C$. In addition, if C is a linear code with generator matrix G and parity check matrix H, then $C^\perp$ is a linear code with generator matrix H and parity check matrix G. We say that a classical linear code satisfying $C^\perp = C$ is *self-dual*, and a classical linear code satisfying $C^\perp\subseteq C$ is *weakly self-dual*.

Distance of a classical linear code is a parameter that tells the number of errors that the classical code can detect and correct. The distance of a classical code is related to the Hamming weight of its codewords, defined as follows:

Definition 3.6 *The* Hamming weight *of a codeword v is the number of non-zero entries of v.*

The following is the definition of the distance of a classical linear code.

Definition 3.7 *Let C be a classical linear code. The* distance *of the code C is the minimum Hamming weight of codewords in C.*

The number of errors that a classical linear code can detect and correct is determined by the following theorem.

Theorem 3.1 *Let C be a classical linear code of distance d. Then, C can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors. In particular, if d is even, then the code can simultaneously correct up to $\frac{d-2}{2}$ errors and detect up to $\frac{d}{2}$ errors [14].*

Similar to Definition 2.9, important parameters of a classical linear code can be expressed using the following notation:

Definition 3.8 *Let C be a classical linear code of length n with $dim(C) = k$, and let d be the distance of the code C. We say that C is an $[n, k, d]$ code.*

We should emphasize that the distance of a classical linear code in Definition 3.7 and the distance of a stabilizer code in Definition 2.6 are different. This fact will be important in the proofs of the main lemmas and theorem of this work, presented in Chapter 5.

3.2 Examples of classical linear codes

There are many classical linear codes which are error correcting codes. One of examples is the 3-bit repetition code described in Section 2.1.1. In this section, two families of classical linear codes which play major roles in this work will be described. These are the families of Hamming codes and cyclic codes.

3.2.1 Hamming codes

Previously in Section 2.3.2, we mentioned that the 7-qubit code is constructed from a parity check matrix of the 7-bit classical Hamming code in the form

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{3.3}$$

We can define a general Hamming code as follows.

Definition 3.9 *Let C be a classical linear code of length $n = 2^r - 1$ for some integer $r \geq 2$. We say that C is a* classical Hamming code *if C has parity check matrix H whose columns consist of all non-zero binary vectors of length r. Such Hamming code is a $[2^r - 1, 2^r - 1 - r, 3]$ code [14].*

Another example of Hamming code is the $[15, 7, 3]$ code where $r = 4$. A parity check matrix of the $[15, 7, 3]$ code is,

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{3.4}$$

Let H be a parity check matrix of a general Hamming code. Observe that all rows of H are mutually orthogonal and have even weight. Consider vector v in $C^{\perp}$, which is generated by H. We find that $Hv^T = 0$ for all $v \in C^{\perp}$. By Definition 3.3, we have $C^{\perp} \subseteq C$. That is, every Hamming code is weakly self-dual. This property will be useful in the construction of quantum Hamming codes, described later in Section 3.3.

3.2.2 Cyclic codes

Cyclic code is a classical linear code in which cyclic permutation preserves the coding subspace [14]. It can be defined as follows:

Definition 3.10 *Let C be a classical linear code of length n. C is* cyclic *if any cyclic shift of a codeword is also a codeword, i.e., if $(c_0, c_1, \ldots, c_n)$ is a codeword, then so is $(c_n, c_0, \ldots, c_{n-1})$.*

One of interesting properties of a cyclic code is that we can choose its generator matrix and its parity check matrix to be in a special form. Let C be a classical cyclic code of length n. There exists a unique generator polynomial $g(x) = \sum_{i=1}^{\alpha} g_i x^i$ which is also a unique monic polynomial of minimal degree in C such that C is generated by the generator matrix

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \ldots & g_\alpha & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & \ldots & g_{\alpha-1} & g_\alpha & \ldots & 0 \\ \ldots & & & & & & & \ldots \\ 0 & \ldots & g_0 & \ldots & & & \ldots & g_\alpha \end{pmatrix}. \tag{3.5}$$

Let $h(x)$ be the polynomial $h(x) = (x^n - 1)/g(x) = \sum_{i=1}^{\beta} h_i x^i$. $h(x)$ is called the check polynomial of C. The parity check matrix of C is

$$H = \begin{pmatrix} h_\beta & h_{\beta-1} & \ldots & h_1 & h_0 & 0 & \ldots & 0 \\ 0 & h_\beta & \ldots & h_2 & h_1 & h_0 & \ldots & 0 \\ \ldots & & & & & & & \ldots \\ 0 & \ldots & h_\beta & \ldots & & & \ldots & h_0 \end{pmatrix}. \tag{3.6}$$

Hamming codes and cyclic codes are related by the following theorem.

Theorem 3.2 *Every Hamming code can be made cyclic [14].*

For example, the $[15, 7, 3]$ Hamming code in cyclic form has the following parity check matrix,

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{3.7}$$

We can readily verify that H satisfies the condition in Definition 3.9, thus it is exactly a parity check matrix of the $[15, 7, 3]$ Hamming code. In particular, parity check matrices in Eqs. (3.4) and (3.7) are equivalent up to permutation of columns. The proof of Theorem 3.2 is quite complicated since it involves the theory of finite fields [14], so it will not be given in this thesis.

3.3 CSS construction

There are several methods to construct a stabilizer code. One method is using classical linear codes. CSS construction is the construction of a stabilizer code from two classical linear codes invented by Calderbank, Shor, and Steane [3, 19, 21]. In this section, we will explain the CSS construction in details.

First, let us define a CSS code as follows:

Definition 3.11 *An $[\![n, k, d]\!]$ stabilizer code is a* CSS code *if there exists a choice of generators such that the binary symplectic representation of generators is of the form*

$$\begin{pmatrix} A & | & 0 \\ 0 & | & B \end{pmatrix}, \tag{3.8}$$

where A is an $r_x \times n$ matrix and B is an $r_z \times n$ matrix for some r_x and r_z with $r_x + r_z = n - k$. A and B are called X and Z stabilizer matrices [3, 19].

From Definition 3.11, we can see that a stabilizer generator of a CSS code is either X-type or Z-type Pauli operator. With this property, X-type errors can be detected using Z-type generators alone, and vice versa.

A CSS code can be constructed from two classical linear codes by the following theorem:

Theorem 3.3 *Let C_x be an $[n, k_x, d_x]$ classical linear code with parity check matrix H_x and let C_z be an $[n, k_z, d_z]$ classical linear code with parity check matrix H_z. Suppose that $C_x^\perp \subseteq C_z$. Let C be the CSS code with stabilizer generators,*

$$\begin{pmatrix} H_x & | & 0 \\ 0 & | & H_z \end{pmatrix}, \tag{3.9}$$

in the symplectic representation. Then C is an $[[n, k, d]]$ quantum code with $k = k_x + k_z - n$ and $d \geq \min\{d_x, d_z\}$. This quantum code construction is called CSS construction. [21]

Proof:
We would like to verify that the Pauli operators specified by Eq. (3.9) indeed form a generating set for the stabilizer of the code C; in particular, the Pauli operators form a commuting set. Since all X-type (or Z-type) generators trivially commute, all we have to do is verifying that all X-type generators commute with all Z-type generators. Observe that the symplectic representation of any X-type generator is of the form $(x|0)$ where $x \in C_x^\perp$ since H_x is a generator matrix of $C_x^\perp$. Also, any Z-type generator is of the form $(0|z)$ where $z \in C_z^\perp$. Then, $(x|0) \odot (0|z) = x \cdot z$. By Proposition 2.3, X-type generators and Z-type generators commute iff

$$x \cdot z = 0 \tag{3.10}$$

for all $x \in C_x^\perp$ and $z \in C_z^\perp$. Assume that $C_x^\perp \subseteq C_z$, then $x \in C_x^\perp$ is also a vector in C_z. By the definition of dual code in Definition 3.5, we find that the condition in Eq. (3.10) is satisfied.

Now we will consider parameters k and d of the CSS code. Since H_x has $n - k_x$ rows and H_z has $n - k_z$ rows, the number of stabilizer generators is $2n - k_x - k_z$. Thus, $k = n - r = k_x + k_z - n$. The CSS code can detect X and Z errors separately, therefore it can detect X errors up to weight $d_z - 1$ and detect Z errors up to weight $d_x - 1$. Hence, the distance d of the CSS code is at least $\min\{d_x, d_z\}$. □

Quantum Hamming code

The 7-qubit code introduced in Section 2.3.2 is an example of a stabilizer code constructed by Theorem 3.3. We know from Section 3.2.1 that a general Hamming code C is weakly self-dual. Thus, a construction with $C_1 = C_2 = C$ is always possible. A stabilizer code constructed from a classical Hamming code is called quantum Hamming code. In particular, quantum Hamming code is a $[\![2^r-1, 2^r-1-2r, 3]\!]$ code. Please note that quantum Hamming codes discussed here are not the quantum codes that saturate the quantum Hamming bound presented in Section 2.1.4.

In addition to classical Hamming codes, classical cyclic codes satisfying Theorem 3.3 can be found in [13].

Chapter 4

Theory of Fault Tolerance and Flag Fault-tolerant Error Correction

In quantum computation, a single fault on a quantum gate may cause some errors which can spread through the entire circuit. If the error rate is high, the computation might fail most of the time. We have discussed in Chapter 2 that the error rate can be reduced by using a QECC to encode the data and applying error correction. However, the error correction procedure previously discussed are assumed to be perfect. What will happen if the QEC itself is faulty? Can we still recover the original data?

The theory of fault tolerance is developed to find solutions to such problems. In this chapter, we will study the conditions that the fault-tolerant gadgets must satisfy so that they can be used to fault-tolerantly simulate an ideal circuit. These conditions guarantee that errors caused by the gadgets will not accumulate too much so that the error correction can still be applied. One of important results from the theory of fault tolerance is the threshold theorem. It says that if the error rate of a physical system is below some threshold value, arbitrarily long fault-tolerant quantum computation is possible.

The main result of this work discussed later in Chapter 5 is a new fault-tolerant error correction protocol. It is based on a fault-tolerant error correction (FTEC) technique called flag fault-tolerant error correction. This technique will be discussed in details in the last part of this chapter.

This chapter is organized as follows: we will first discuss the Clifford group and error propagation in Section 4.1. Afterwards, the theory of fault tolerance including error models, the properties of fault tolerant protocols, and the threshold theorem will be discussed in Section 4.2. We will finally review the flag-FTEC in Section 4.3.

4.1 Clifford group and error propagation

From Chapter 2, we already know that a QECC of distance d can correct up to weight-t error, where $t = \lfloor \frac{d-1}{2} \rfloor$. Suppose that there is an error occurred during computation, the weight of the error may increase after the application of quantum gates. Anyhow, such error is still correctable as long as its weight is $\leq t$. For this reason, we intend to apply the error correction as frequently as possible in the computation to avoid error accumulation.

To study how errors change after applying quantum gates, the Clifford group might be introduced. The Clifford group is a set of unitary operators which preserve the Pauli group under conjugation, defined as follows:

Definition 4.1 *Let $\mathcal{P}_n$ be the Pauli group on n qubits and let $\mathcal{U}(2^n)$ be the unitary group. The* Clifford group *$\mathcal{C}_n$ on n qubits is,*

$$\mathcal{C}_n = \{U \in \mathcal{U}(2^n) : UPU^\dagger \in \mathcal{P}_n \ \textit{for all} \ P \in \mathcal{P}_n\}. \tag{4.1}$$

The Clifford group on n qubits can be generated by Hadamard gate H, $R_{\pi/4}$ gate, and CNOT gate, where

$$H = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_{\pi/4} = e^{-i\pi/4}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{4.2}$$

Consider a set of elementary operations consisting of preparations in $|0\rangle$ state, Clifford gate operations, and qubit measurements corresponding to the eigenvalues of X, Y or Z operator. The Gottesman-Knill theorem states that if a quantum circuit consists of only these elementary operations, the operation of the circuit can be efficiently simulated by a classical computer [9]. This means that if we want to construct a quantum computer that fully exploits the power of quantum computation, only aforementioned operations are not sufficient.

It is known that magic state preparation along with elementary operations previously described can simulate universal quantum computation [2]. In particular, magic states are special ancillary states which have the following properties:

1. Universality:
   The preparation of magic states along with $|0\rangle$ state preparation, Clifford gate operations, and qubit measurement in X, Y or Z basis can simulate universal quantum computation.

2. Distillability:
   Imperfect magic states can be purified using only $|0\rangle$ state preparation, Clifford gate operations, and qubit measurement in X, Y or Z basis.

There are many types of magic states. Two of them which are studied the most are T-type magic state $|T\rangle$ and H-type magic state $|H\rangle$. The density operators of these two magic states are,

$$\begin{aligned} |T\rangle\langle T| &= \frac{1}{2}\left[I + \frac{1}{\sqrt{3}}(X+Y+Z)\right], \\ |H\rangle\langle H| &= \frac{1}{2}\left[I + \frac{1}{\sqrt{2}}(X+Z)\right]. \end{aligned} \tag{4.3}$$

As a result of the universality property, we may sometimes assume that all quantum gates in the circuit are Clifford gates.

Now let us consider the propagation of Pauli errors. Let T be a stabilizer code, $|\psi\rangle \in T$ be a valid codeword, $U \in \mathcal{U}(2^n)$ be any unitary operator, and $E \in \mathcal{P}_n$ be a Pauli error. The operation of U on an erroneous codeword $E|\psi\rangle$ gives,

$$UE|\psi\rangle = (UEU^\dagger)U|\psi\rangle. \tag{4.4}$$

Observe that $U|\psi\rangle$ is a state from the operation of U on state $|\psi\rangle$. Thus, the R.H.S. of Eq. (4.4) can be interpreted as state $U|\psi\rangle$ with error $UEU^\dagger$. Here we can see that the error E becomes $UEU^\dagger$ after the operation of U.

We may assume that all quantum gates are Clifford gates, i.e., $U \in \mathcal{C}_n$. With this assumption, any Pauli error is mapped to another Pauli error under the conjugation of U. To see how Pauli errors are mapped, it is sufficient to consider only the transformation of generators of $\mathcal{P}_n$ under the conjugation of the generators of $\mathcal{C}_n$. The operation of H transforms X error to Z error and vice versa as shown in Eq. (4.5). The transformations of Pauli errors under the operation of $R_{\pi/4}$ and CNOT are shown in Eqs. (4.6) and (4.7).

$$H: \begin{array}{ccc} X & \mapsto & Z \\ Z & \mapsto & X \end{array} \tag{4.5}$$

$$R_{\pi/4}: \begin{array}{ccc} X & \mapsto & Y \\ Y & \mapsto & -X \\ Z & \mapsto & Z \end{array} \tag{4.6}$$

$$\text{CNOT}: \begin{array}{ccc} X \otimes I & \mapsto & X \otimes X \\ I \otimes X & \mapsto & I \otimes X \\ Z \otimes I & \mapsto & Z \otimes I \\ I \otimes Z & \mapsto & Z \otimes Z \end{array} \tag{4.7}$$

In Eq. (4.7), we can see that X error propagates from the control qubit of a CNOT gate to the target qubit, while Z error propagates from the target qubit to the control qubit. The propagation of errors discussed here will play an important role in the analysis of flag error correction in Section 4.3.

4.2 Theory of fault tolerance

In a practical quantum computer, every component of the quantum circuit can be faulty and errors may occur at any point during computation. Quantum error correction described in Chapter 2 can be used to reduce the error rate. However, it may not work properly if the number of errors grows beyond the capability of the code before error correction is applied. In order to build a quantum computer which is fault tolerant, we have to make sure that an error occurred at any point will not propagate badly so that the error correction can be applied in time.

In this section, we will start by describing an independent local stochastic model. Afterwards, the conditions which fault-tolerant gadgets must satisfy will be explained. Finally, the threshold theorem for an independent local stochastic model will be discussed.

4.2.1 Error model

Before we look at a particular error model, let us define the smallest unit of a quantum circuit as follows:

Definition 4.2 *A* location *in the quantum circuit is a single action which takes 1 time-step. This can be preparation location, gate location, measurement location, wait location, or classical computation location.*

In a physical quantum computer, any locations can be faulty and the probability that a fault occurs may differ depending on the types of locations. In addition, these faults might be correlated in some way; for example, a faulty quantum gate may induce errors on its neighboring quantum gates in the circuit even if they are not applied on the same qubits. Anyhow, the correlated error model is quite complicated so we will assume in this work that all faults are independent. The model using in this work is defined as follows:

Definition 4.3 *An error model is an* independent local stochastic model *if each location L performs the correct action with probability $1 - p_L$ and performs anything else with* *probability p_L, where p_L only depends on the type of locations. In particular, if each location performs the correct action followed by Pauli errors, the error model is an* independent Pauli error model*.*

4.2.2 Properties of fault-tolerant protocols

Suppose that we have a particular ideal quantum circuit and we want to build a fault-tolerant computer simulating the circuit, one way to do this is encoding the data using a QECC and replacing each ideal component by a fault-tolerant gadget, defined as follows:

Definition 4.4 *Let L be a specific type of location. A* gadget *for L is a QECC T with a circuit G_L such that if the qubits involved with L are encoded into T, then applied by G_L, then decoded without errors in any process, this gives the same effect as performing the action of L directly on the unencoded qubits.*

We have to make sure that each gadget in the circuit satisfies two properties: the gadget must perform the same logical operation as the corresponding component does, and it must not propagate errors too badly. To study these two properties, some graphical notations might be introduced.

First, let us define the r-filter as follows:

Definition 4.5 *Let T be a QECC. The* r-filter *for T is the projector onto the subspace spanned by,*

$$\{E|\psi\rangle : |\psi\rangle \in T, E \in \mathcal{P}_n \;\; \textit{with} \;\; wt(E) \leq r\}. \tag{4.8}$$

A graphical representation of the $r-$filter is shown in Fig. 4.1 where the rectangle represents the r-filter and the horizontal bold line represents a block of code.

From Definition 4.5, we can see that the output state from the r-filter differs from a valid codeword by an error of weight at most r.

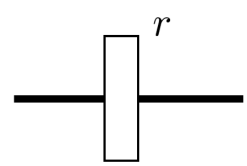


Figure 4.1: r-filter

In order to transform an ideal quantum circuit to its corresponding fault-tolerant circuit, we will use the following notations:

Definition 4.6 *Let T be an QECC. Ideal gadgets representing quantum gate, state preparation, qubit measurement, and error correction acting on a single qubit are shown in Fig. 4.2a, Fig. 4.2b, Fig. 4.2c, and Fig. 4.2d, respectively. Let s be a number of faults. Noisy gadgets with exactly s faults simulating ideal quantum gate, ideal state preparation, ideal qubit measurement, and ideal error correction are shown in Fig. 4.2e, Fig. 4.2f, Fig. 4.2g, and Fig. 4.2h, respectively. A horizontal thin line represents a single qubit while a horizontal bold line represents a block of code.*

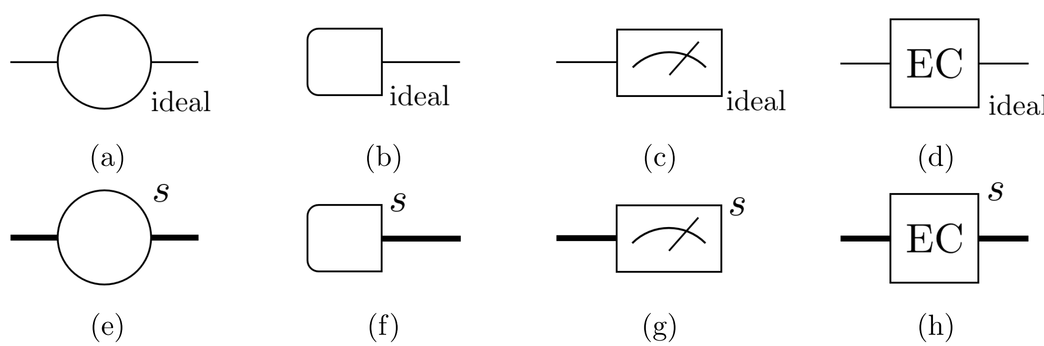


Figure 4.2: Graphical representations of ideal and noisy gadgets: (a) ideal gate gadget, (b) ideal preparation gadget, (c) ideal measurement gadget, (d) ideal error correction gadget, (e) noisy gate gadget, (f) noisy preparation gadget, (g) noisy measurement gadget, (h) noisy error correction gadget. All noisy gadgets described here have exactly s faults.

Lastly, we will define the ideal decoder which can perfectly decode the erroneous codewords regardless of the weight of errors as following:

Definition 4.7 *The* ideal decoder *for a QECC T is the quantum channel that takes an erroneous state encoded in T, corrects the errors, and decodes the logical state without any faults during the process. A graphical representation of the ideal decoder is shown in Fig. 4.3. The triangle represents the ideal decoder, and the horizontal bold line and horizontal thin line represent a block of code and a single qubit, respectively.*

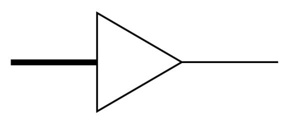


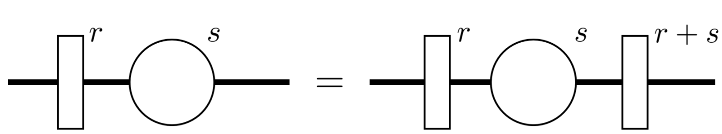
Figure 4.3: Ideal decoder

Now we are ready to consider the properties that fault-tolerant gadgets must satisfy [1, 10].

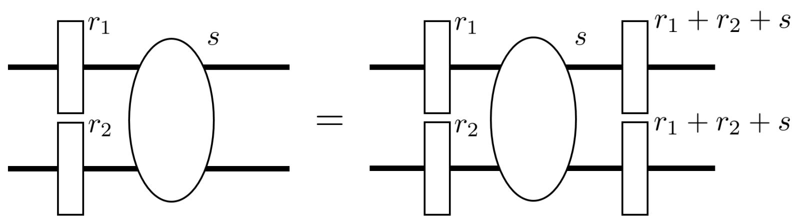
Fault-tolerant quantum gates:

A fault-tolerant gate gadget must satisfy two conditions: its action on an encoded state must be logically equivalent to the action of the corresponding ideal gate on a logical state, and the gadget must not propagate errors too badly. The formal definitions of these two conditions are as followings:

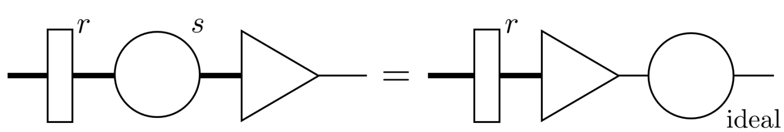
Definition 4.8 *Let G be a single-qubit gate gadget associated with a QECC that can correct up to t errors, and let r be the number of input errors and s be the number of faults in G. G satisfies the* Gate Error Propagation Property (GPP) *if whenever $r + s \leq t$,*



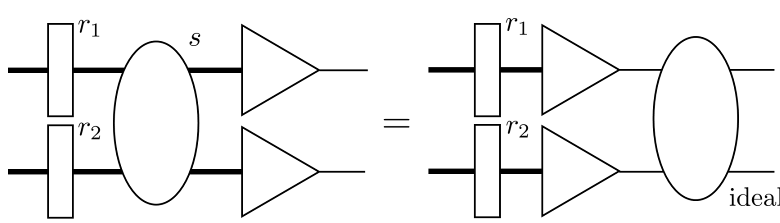
For a 2-qubit gate gadget, let r_1, r_2 be the numbers of input errors in blocks 1 and 2, respectively. The GPP is satisfied whenever $r_1 + r_2 + s \leq t$,



Definition 4.9 *Let G be a single-qubit gate gadget associated with a QECC that can correct up to t errors, and let r be the number of input errors and s be the number of faults in G. G satisfies the* Gate Error Correctness Property (GCP) *if whenever $r + s \leq t$,*



For a 2-qubit gate gadget, let r_1, r_2 be the number of input errors in blocks 1 and 2, respectively. The GCP is satisfied whenever $r_1 + r_2 + s \leq t$,



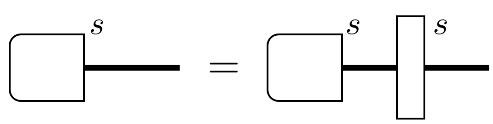
The GPP for a single-qubit gate guarantees that if the input of the gate gadget has r errors and the gadget has s faults, the output will have at most $r + s$ errors whenever $r + s \leq t$. This means that error correction still works after the operation of the gate gadget. For a 2-qubit gate gadget, it is possible that errors might propagate from the first block to the second block, or vice versa. The GPP guarantees that the output of any block will not have errors more than the number of total input errors plus the number of faults whenever $r_1 + r_2 + s \leq t$. In addition, the GCP guarantees that a gate gadget performs the correct operation whenever the number of total input errors plus the number of faults is not greater than t.

We have seen in Eq. (4.7) that errors can propagate from one qubit to another qubit when a 2-qubit gate is applied. It is not easy to find quantum gate gadgets which satisfy both GPP and GCP. Fortunately, we find that if two blocks of code are coupled by applying 2-qubit gates only between the i^{th} qubit of one block and the i^{th} qubit of another block, the GPP is satisfied. A quantum gate being applied in this kind of coupling is called *transversal gate*. In addition, if applying operator U transversally is equivalent to applying logical U (or another logical operator), we say that U is *transversal*. There are some stabilizer codes in which all generators of the Clifford group are transversal. For example, in the 7-qubit code, $H^{\otimes 7}$ is logical H, $R_{\pi/4}^{\otimes 7}$ is logical $R_{-\pi/4}$, and $\text{CNOT}^{\otimes 7}$ is logical CNOT. This comes from the fact that the 7-qubit code is a CSS code constructed from a weakly self-dual code.

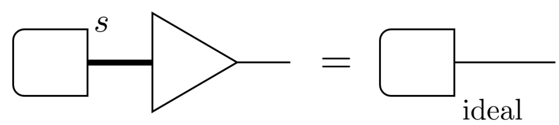
Fault-tolerant state preparation:

The conditions for a state preparation gadget is quite similar to the conditions for a gate gadget. We will define the conditions as follows:

Definition 4.10 *Let G be a preparation gadget associated with a QECC that can correct up to t errors, and let s be the number of faults in G. G satisfies the* Fault-tolerant Preparation Error Propagation Property (PPP) *if whenever $s \leq t$,*



Definition 4.11 *Let G be a preparation gadget associated with a QECC that can correct up to t errors, and let s be the number of faults in G. G satisfies the* Fault-tolerant Preparation Correctness Property (PCP) *if whenever $s \leq t$,*

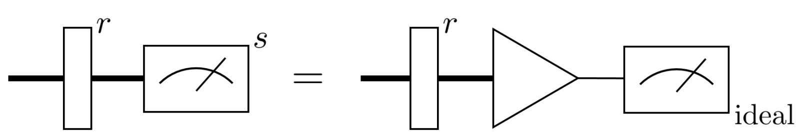


The PPP guarantees that the output of a preparation gadget have errors no more that the number of faults, while the PCP guarantees that the gadget prepares exactly the same state as desired.

Fault-tolerant measurement:

For a fault-tolerant measurement gadget, the error propagation property is not required since the measurement produces classical data and the classical error correction is assumed to be perfect. Thus, we have only correctness property defined as follows:

Definition 4.12 *Let G be a measurement gadget associated with a QECC that can correct up to t errors, and let r be the number of input errors and s be the number of faults in G. G satisfies the* Fault-tolerant Measurement Correctness Property (MCP) *if whenever $r + s \leq t$,*

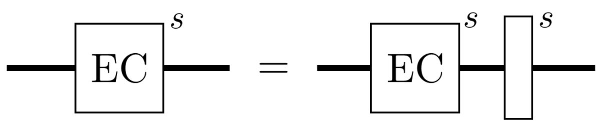


The MCP verifies that whenever $r+s \leq t$, the measurement gadget gives the correct result.

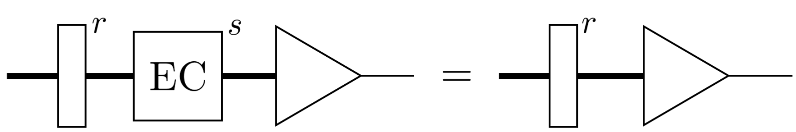
Fault-tolerant error correction:

The last fault-tolerant gadget that will be discussed is an error correction gadget. This gadget is important since it will be applied before and after other gadgets as frequently as possible to make sure that errors will not build up beyond the capability of the QECC. A fault-tolerant error correction gadget must satisfy the following conditions:

Definition 4.13 *Let G be an error correction gadget associated with a QECC that can correct up to t errors, and let s be the number of faults in G. G satisfies the* Fault-tolerant Error Correction Recovery Property (ECRP) *if whenever $s \leq t$,*



Definition 4.14 *Let G be an error correction gadget associated with a QECC that can correct up to t errors, and let r be the number of input errors and s be the number of faults in G. G satisfies the* Fault-tolerant Error Correction Correctness Property (ECCP) *if whenever $r + s \leq t$,*



The ECRP is different from the propagation properties of other gadgets; there is no restriction on the weight of the input error. The ECRP guarantees that if the error correction gadget has s faults, the weight of the output error will be at most s. This means that if the input state has many errors, the output state may have a logical error but it must not differ from a valid codeword more than s errors. The ECCP is similar to other correctness properties. It verifies that if the input error has weight r and the gadget has s faults where $r + s \leq t$, the error correction gadget will perform correctly.

In general, a fault-tolerant protocol may contain more than one gadgets. In that case, every gadget in the protocol must satisfy the conditions previously discussed. The definition of a fault-tolerant protocol is as follows:

Definition 4.15 *A* fault-tolerant protocol *consists of a QECC with the following types of fault-tolerant gadgets:*

- *Gate gadget satisfying the GPP in Definition 4.8 and the GCP in Definition 4.9,*
- *state preparation gadget satisfying the PPP in Definition 4.10 and the PCP in Definition 4.11,*
- *qubit measurement gadget satisfying the MCP in Definition 4.12,*
- *error correction gadget satisfying the ECRP in Definition 4.13 and the ECCP in Definition 4.14.*

4.2.3 Threshold theorem

In the previous section, we developed all properties that each fault-tolerant gadget must satisfy. These gadgets are the main ingredients for the construction of a fault-tolerant circuit simulating any desired ideal circuit. The simulation of a quantum circuit is defined as follows:

Definition 4.16 *Let C be a quantum circuit and let T be a QECC. A* fault-tolerant simulation *$\tilde{C}$ of the circuit C associated with given fault-tolerant protocols is a circuit obtained by replacing each location of C by the corresponding fault-tolerant protocol, replacing each qubit by a block of code T, and inserting error correction protocols after every state preparation gadget and gate gadget.*

Our goal is to build a reliable quantum computer where the error rate can be made arbitrarily small. In Chapter 2 where we assume that all error correction protocols are perfect, the error rate can be reduced by encoding the data into codewords and performing error correction; for example, the 9-qubit code reduces the error rate from p to $O(p^2)$. Thus if we concatenate the quantum codes, i.e., the data is encoded repeatedly, the error rate can be suppressed to a very small positive number when the number of concatenation level becomes large. The idea of code concatenation can also be applied to the fault-tolerant simulation in which every gadget satisfies the fault-tolerant properties. That is, the error rate can be made arbitrarily small by repeatedly simulating the quantum circuit. With this error rate suppression idea, it is possible to prove the following threshold theorem:

Theorem 4.1 *Let C be an ideal quantum circuit. Suppose that the system is subjected to the independent local stochastic model with error probability p for any type of location. Then there exists threshold error rate p_T such that if $p < p_T$, for any $\epsilon > 0$, there exists a fault-tolerant simulation $\tilde{C}$ of C such that the output probability distribution of $\tilde{C}$ has statistical distance at most ϵ from the output of C. The threshold error rate p_T depends on the fault-tolerant protocols. In addition, if C has k locations, then $\tilde{C}$ has $O(k \cdot \mathit{polylog}(k/\epsilon))$ locations [10].*

The threshold theorem is one of the most important results from the theory of fault tolerance. It implies that for given fault-tolerant protocols, if we can experimentally devise all types of locations with error probability less than p_T, a physical reliable quantum computer can be built.

We point out that the threshold theorem discussed here is based on the following assumptions:

1. All quantum gates being used in any fault-tolerant protocol are Clifford gates.
2. 2-qubit gates can be implemented on any pair of qubits.
3. State $|0\rangle$ can be prepared at any point during computation.
4. State measurement can be performed at any point during computation, and classical computation involving measurement results is reliable.
5. Numerous tasks can be operated in parallel.
6. The physical system is subjected to independent local stochastic error model.

Note that in some studies where some of these assumptions are relaxed, other versions of threshold theorem might be obtained.

4.3 Flag fault-tolerant error correction

The threshold theorem discussed in Section 4.2.3 suggested that whenever $p < p_T$, the statistical distance ϵ between the ideal circuit C and the circuit simulation $\tilde{C}$ can be made arbitrarily small. However, we can see from Theorem 4.1 that the number of required resources will become larger as ϵ decreases. Since resources in a real-world experiment are very limited, developing a new protocol in which fewer resources are required is an important task. In this work, the number of ancilla qubits being used in fault-tolerant error correction (FTEC) protocols is the main concern.

There are three well-known FTEC protocols which can be applied to a large family of stabilizer codes: Shor EC [17], Steane EC [20], and Knill EC [11]. The family of codes in which each protocol is applicable and the number of required ancilla qubits for each protocol are as followings:

- Shor EC is applicable to any stabilizer code. For a measurement of weight-m stabilizer generator, the protocol requires m ancilla qubits. In addition, a large number of measurement repetitions is required.
- Steane EC is only applicable to CSS codes. It requires fewer repetitions than Shor EC does. However, Steane EC requires a full block of code as ancilla qubits, i.e., if the code being used has length n, then n ancilla qubits are required.
- Knill EC is applicable to any stabilizer code and requires fewer repetitions than Shor EC does. Anyhow, two additional blocks of code are required in the protocol.

The drawback of these protocols is that the number of required ancilla qubits grows as the weight of stabilizer generators or the block length becomes larger. Hence, these protocols might not be a good choice for a physical implementation in the systems where ancilla qubits are limited.

Recently, a technique called flag-FTEC has been proposed by Chao and Reichardt [5]. It requires only two ancilla qubits and it is applicable to numerous QECCs of distance 3 such as the 5-qubit code and quantum Hamming codes. In their protocol, one ancilla qubit is

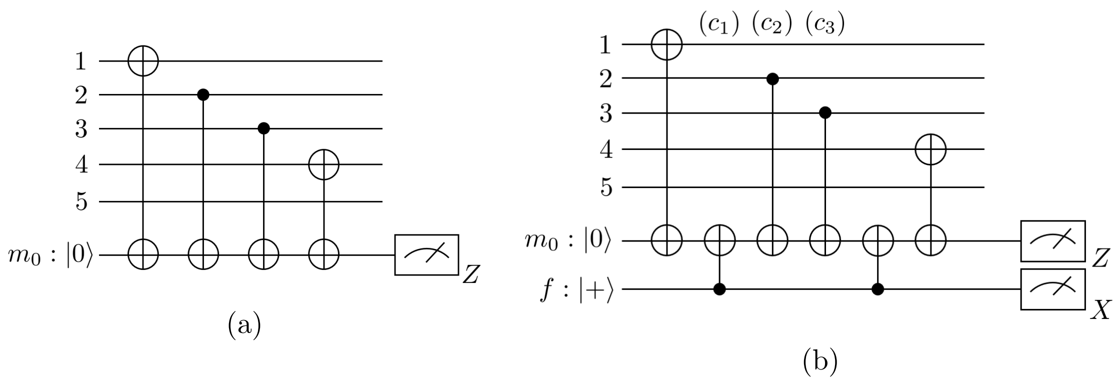


Figure 4.4: Circuits for measuring generator $XZZXI$ of the 5-qubit code, where the quantum gate involving X measurement is defined in Fig. 2.2b. The ideal circuit in Fig. 4.4a is not fault tolerant since some fault can lead to an error of weight > 1. In Fig. 4.4b, a corresponding 1-flag circuit has been shown. Here, a flag qubit is introduced to detect an error of higher weight caused by gate (c_1), (c_2) or (c_3).

used in the syndrome measurement and another qubit is used as a 'flag' to detect a single fault which can cause error of higher weight. The idea of flag-FTEC is generalized later by Chamberland and Beverland [4]. In this section, we will review some basics of flag-FTEC.

Consider a circuit for measuring stabilizer generator $XZZXI$ of the 5-qubit code (discussed in Section 2.3.3) as shown in Fig. 4.4a. In the independent Pauli error model defined in Definition 4.3, we assume that a faulty 2-qubit gate will perform the correct action followed by an error of the form $P \otimes Q$, where P is a Pauli error on the control qubit and Q is a Pauli error on the target qubit (which is an ancilla qubit for the syndrome measurement in this case). Observe that if the error from a single fault is $P \otimes X$ where $P \in \{I, X, Y, Z\}$, the data error will have weight at most 1 in any case since X error cannot propagate from the target qubit to the control qubit. On the other hand, if the error from a single fault is of the form $P \otimes Z$ or $P \otimes Y$ where $P \in \{I, X, Y, Z\}$, it is possible to have an error of weight > 1 on the data since Z error can propagate back to the control qubit. For example, if the second gate in Fig. 4.4a fails and causes IZ error, we will get an $IIZXI$ error on the data, which is equivalent to a $XZIII$ error.

Now let us consider a circuit in Fig. 4.4b which has another ancilla qubit initially prepared in state $|+\rangle$. This 'flag' qubit is used to detect a single fault which can cause data error

Location	Failure	Data error
(c_1)	IZ	$IZZXI \sim XIIII$
(c_2)	IZ	$IIZXI \sim XZIII$
	XZ	$IXZXI \sim XYIII$
	YZ	$IYZXI \sim XXIII$
	ZZ	$IZZXI \sim XIIII$
(c_3)	IZ	$IIIXI \sim XZZII$
	XZ	$IIXXI \sim XZYII$
	YZ	$IIYXI \sim XZXII$
	ZZ	$IIZXI \sim XZIII$
f measurement		$IIIII$

Table 4.1: All possible single faults which can cause the circuit in Fig. 4.4b to flag and their corresponding data errors. Any pair of errors in this table are either logically equivalent or corresponding to different syndromes.

of weight > 1. Observe that if the gate (c_1), (c_2) or (c_3) in Fig. 4.4b causes an error of the form $P \otimes Z$ or $P \otimes Y$, Z error on qubit m_0 will propagate to qubit f and the state will become $|-\rangle$. This can be distinguished from other cases by measuring qubit m_0 in X basis. When the measurement result gives eigenvalue -1, we say that the circuit flags. All possible faults that can cause the circuit to flag and the corresponding data errors are listed in Table 4.1. In addition, these errors have distinct syndromes, thus there are distinguishable by subsequent syndrome measurements and the error correction can be applied. The circuit in Fig. 4.4b is called a 1-flag circuit. The formal definition of a general t-flag circuit is as follows:

Definition 4.17 *Let P be a Pauli operator. A circuit $C(P)$ is a t-*flag circuit *if*

1. *the circuit implements a projective measurement of P without flagging when fault-free, and*

2. *for any sets of v faults at up to t locations in $C(P)$ resulting in an error E with $min(wt(E), wt(EP)) > v$, the circuit flags.*

In addition, the set of all errors that must be distinguished if the circuit flags is defined as following:

Definition 4.18 *Let $C(g_i)$ be a 1-flag circuit corresponding the measurement of operator g_i. The* flag error set *$\mathcal{E}(g_i)$ is the set of all errors that caused $C(g_i)$ to flag.*

In order to distinguish all errors causing the circuit to flag so that the error correction can be performed, the following condition must be satisfied.

Definition 4.19 *Let $S = \langle g_1, g_2, \dots, g_r\rangle$ be a stabilizer, $\{C(g_1), C(g_2), \dots, C(g_r)\}$ be a set of corresponding 1-flag circuits, and $\mathcal{E}(g_i)$ the flag error set corresponding to $C(g_i)$. The* Flag 1-FTEC condition *is satisfied if for every generator g_i, all pairs of elements $E, E' \in \mathcal{E}(g_i)$ satisfy $s(E) \neq s(E')$ or E and E' are logically equivalent (written as $E \sim E'$).*

Now we will consider the error correction procedure for the flag-FTEC. Recall the ECRP in Definition 4.13. In the case that the circuit has a single fault ($s = 1$), we know that the output state cannot differ from a valid codeword with error of weight greater than 1 even if the input error has higher weight. We may define the minimal weight error correction as follows:

Definition 4.20 *Let s be the syndrome of a Pauli operator. The* minimal weight error correction *$E_{min}(s)$ is the error correction of the Pauli operator of minimum weight which corresponds to syndrome s.*

Let S be a stabilizer and $|\psi\rangle \in T(S)$ be a valid codeword. Consider the following cases of error correction:

1. If errors E and E' are logically equivalent; i.e., $E' = EM$ for some operator $M \in S$, then applying the error correction of E which is $E^\dagger$ on state $E'|\psi\rangle$ gives

$$E^\dagger E'|\psi\rangle = M|\psi\rangle = |\psi\rangle. \tag{4.9}$$

   In this case, the original codeword is recovered.

2. If errors E and E' are not logically equivalent but $s(E) = s(E')$, we know from Proposition 2.1 that E and E' are in the same coset of $N(S)$; i.e., $E' = EA$ for some operator $A \in N(S)$. Assume that E is the Pauli operator of minimum weight corresponding to syndrome s, then $E_{min}(s) = E^\dagger$. By applying $E_{min}(s)$ on state $E'|\psi\rangle$, we have

$$E^\dagger E'|\psi\rangle = A|\psi\rangle \in T(S). \tag{4.10}$$

   We can see that even if the original codeword is not recovered, the output state is still a valid codeword.

The procedure discussed above is a crucial part of the error correction protocol that makes it satisfy the ECRP. The full FTEC protocol for 1-flag circuit is described as follows:

Flag 1-FTEC Protocol

Let T be a stabilizer code with stabilizer $S = \langle g_1, g_2 \ldots, g_r\rangle$ and corresponding 1-flag circuits $C(g_1), C(g_2), \ldots, C(g_r)$, and let s_i be the syndrome obtained from round i. Repeat the syndrome measurement using the 1-flag circuits until one of the following is satisfied.

1. If the syndrome s is repeated twice in a row and the circuit does not flag in both rounds (i.e., the measurement of the flag qubit gives $+1$ result), apply $E_{min}(s)$.
2. If the syndromes s_1 and s_2 are different and the circuit does not flag in both rounds, repeat the syndrome measurement using non-flagged circuits to obtain s_3. Apply the error correction $E_{\min}(s_3)$.
3. If a circuit $C(g_i)$ flags, stop and repeat the syndrome measurement using non-flag circuits to obtain syndrome s. If there is an element E in the flag error set $E(g_i)$ satisfying $s(E) = s$, apply $E^\dagger$. Otherwise, apply $E_{min}(s)$.

The flag-FTEC condition and the flag-FTEC protocol can also be generalized for a general t-flag circuit. The generalization and analysis can be found in [4].

Previously, the 1-flag circuit for measuring generator $XZZXI$ of the 5-qubit code has been discussed. In that case, quantum gates for syndrome measurement is in the *normal permutation*; i.e., they are applied from the top qubit to the bottom qubit as in Fig. 4.4b. Note that in general, a flag circuit with normal permutation might not satisfy the flag 1-FTEC condition but a flag circuit with another permutation of gates might do.

Let us consider the 1-flag circuit for measuring generator $X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}X_{15}$ of the 15-qubit code as an example. The circuit with normal permutation is shown in Fig. 4.5a. Suppose that a fault from the CNOT gate $(*)$ causes an error IZ, the data error will be $X_{12}X_{13}X_{14}X_{15}$. This error gives the trivial syndrome $s = 0$, which means it cannot be distinguished from the case of faulty flag measurement which causes no error on the data. In particular, $X_{12}X_{13}X_{14}X_{15}$ and I are not logically equivalent. Thus, the flag 1-FTEC condition is not satisfied. Now let us consider the circuit with another permutation of CNOT gates in Fig. 4.5b. We can verify that every error in the flag error set leads to a distinct syndrome. Thus, the flag 1-FTEC condition is satisfied and fault-tolerant error correction is possible. In addition, it has been proved in [5] that a flag circuit for syndrome extraction in a general quantum Hamming code satisfies the flag 1-FTEC condition for some permutation of quantum gates, and such permutation can always be constructed. The theorem is as follows:

Theorem 4.2 *There exists a permutation of quantum gates in the syndrome measurement circuits for a $[\![2^r-1, 2^r-1-2r, 3]\!]$ quantum Hamming code such that fault-tolerant error correction can be performed using only two ancilla qubits.*

Besides this, the flag-FTEC is also applicable to any stabilizer code satisfying the sufficient condition provided in [4]. The condition is as following:

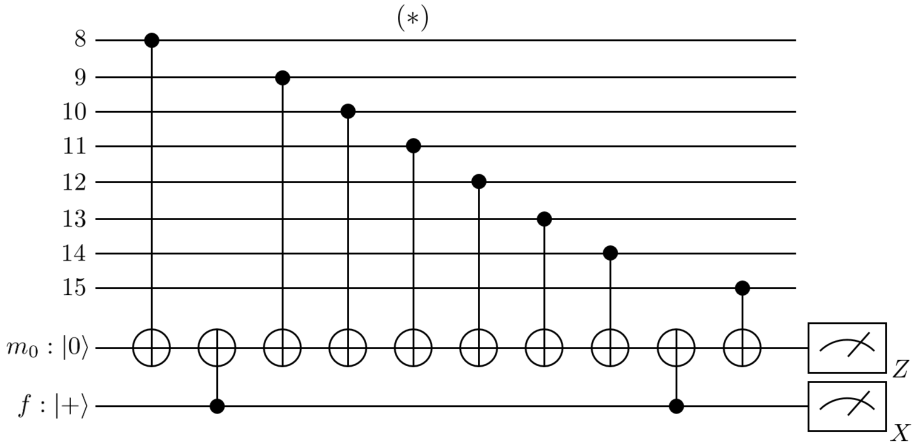
Theorem 4.3 *Consider a stabilizer code of distance $d > 1$ with stabilizer $S = \langle g_1, g_2, \dots, g_r\rangle$. Suppose that for all $v \in \{0, 1, \dots, t\}$, all choices Q_{t-v} of $2(t-v)$ qubits, and all subsets of v stabilizer generators $\{g_{i_1}, \dots, g_{i_v}\} \subset \{g_1, \dots, g_r\}$, there is no logical operator $l \in N(S) - S$ such that,*

$$supp(l) \subset supp(g_{i_1}) \cup \cdots \cup supp(g_{i_v}) \cup Q_{t-v}, \tag{4.11}$$

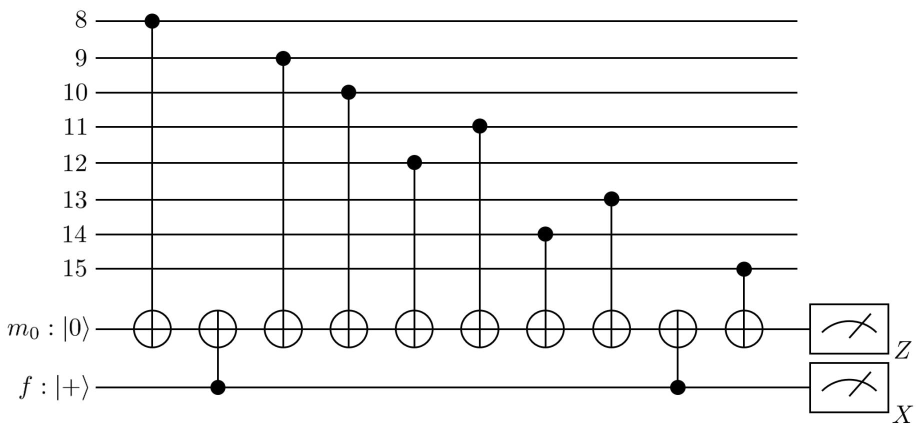
where supp(P) denotes the set of supporting qubits of operator P. Then, the flag t-FTEC condition is satisfied for any choice of t-flag circuits.

Examples of stabilizer codes satisfying Theorem 4.3 are $[\![2^m-1,1,d]\!]$ quantum Reed-Muller code for every integer $m \geq 3$, $[\![d^2-1,1,d]\!]$ rotated surface code for all odd integer d, and $[\![(3d^2+1)/4,1,d]\!]$ hexagonal color code [4]. For every code satisfying Theorem 4.3, the permutation of gates in the flag circuit does not affect the fault-tolerant error correction properties.

The idea of flag-FTEC developed by [5] and [4] allows us to find a fault-tolerant protocol using a constant number of ancilla qubits. In the next chapter, we will further develop this idea in order to construct a new flag-FTEC protocol which is applicable to cyclic CSS codes.



(a)



(b)

Figure 4.5: Circuits for measuring generator $X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}X_{15}$ of the 15-qubit code. A 1-flag circuit in Fig. 4.5a does not satisfy the flag 1-FTEC condition since a fault IZ from the CNOT gate $(*)$ can cause data error $X_{12}X_{13}X_{14}X_{15}$ which is not equivalent to I but has the trivial syndrome. In Fig. 4.5b, another permutation of CNOT gates gives a 1-flag circuit which satisfies the flag 1-FTEC condition in Definition 4.19.

Chapter 5

Flag Fault-tolerant Error Correction for Cyclic CSS Codes

Fault-tolerant error correction protocol is one of the most important parts of fault-tolerant circuit simulation since the protocol helps prevent error accumulation. In Section 4.3, flag-FTEC protocols developed by [5] and [4] have been discussed. The advantage of the flag-FTEC protocols is that only small number of ancilla qubits is required. In this chapter, the idea of flag-FTEC will be extended to a family of CSS codes constructed from two classical cyclic codes. This work was done jointly with Christopher Chamberland and Debbie Leung and is submitted to the arXiv as [22].

5.1 Flag circuit for general CSS codes

In Section 4.3, the 1-flag circuits for the 5-qubit code and the 15-qubit code have been discussed, and we have seen that not every 1-flag circuit satisfies the flag 1-FTEC condition in Definition 4.19. The crucial point is that any pair of errors which can cause the circuit to flag must be either logically equivalent or correspond to different syndromes. In particular, let $C(g_i)$ be a 1-flag circuit with flag error set $\mathcal{E}(g_i)$ and let E_1 and E_2 be errors in $\mathcal{E}(g_i)$. If E_1 and E_2 are not logically equivalent, they must be distinguishable by the subsequent syndrome measurement in the flag-FTEC protocol. It is possible to give a formal definition of distinguishable errors as follows:

Definition 5.1 *Let C be an $[\![n,k,d]\!]$ stabilizer code and let E_1 and E_2 be Pauli errors with syndromes $s(E_1)$ and $s(E_2)$, respectively. We say that E_1 and E_2 are* distinguishable *by C if $s(E_1) \neq s(E_2)$. Otherwise we say that they are* indistinguishable. *In addition, if any pair of errors from an error set $\mathcal{E}$ are distinguishable by C, we say that $\mathcal{E}$ is distinguishable by C.*

Our goal here is to find a family of codes and their 1-flag circuits such that the flag 1-FTEC condition is satisfied and the flag-FTEC technique is applicable.

Let us consider the flag-FTEC corresponding to a CSS code (defined in Definition 3.11) such as the 15-qubit code discussed in Section 4.3. The analysis of error distinguishability for CSS codes can be much easier compared to the case of non-CSS codes since X-type errors can be distinguished using only Z-type generators or vice versa. For a circuit measuring Z-type generator, we already know that the permutation of CNOT gates affects the distinguishability of errors in the flag error set. Note that permuting CNOT gates is equivalent to permuting the columns of the stabilizer matrices. Thus, in order to find a family of CSS codes such that the flag technique can be applied, we may fix the CNOT gates in the normal permutation (i.e., applying CNOT gates from top to bottom in the syndrome extraction circuit) and find conditions which are needed to be satisfied by the X and Z stabilizer matrices instead. In particular, for an X-type or Z-type generator which has support on m qubits where $m \in \{1,\dots,n\}$, we may assume that the operator being measured is in the form $X^{\otimes m} \otimes I^{\otimes n-m}$ or $Z^{\otimes m} \otimes I^{\otimes n-m}$. The 1-flag circuit for measuring weight-m Z-type generator is shown in Fig. 5.1.

Suppose that the system is subjected to the independent Pauli error model (defined in Definition 4.3), we will assume that a faulty CNOT gate can cause an error of the form $P \otimes Q$ where $P, Q \in \{I,X,Y,Z\}$ are Pauli errors on the control and the target qubits, respectively. Consider a circuit for measuring operator $Z^{\otimes m} \otimes I^{\otimes n-m}$ as in Fig. 5.1. A single fault at a CNOT location can result in the following types of errors:

(a) If an error from a faulty CNOT gate is of the form $P \otimes Q$ where $P \in \{I,X,Y,Z\}$

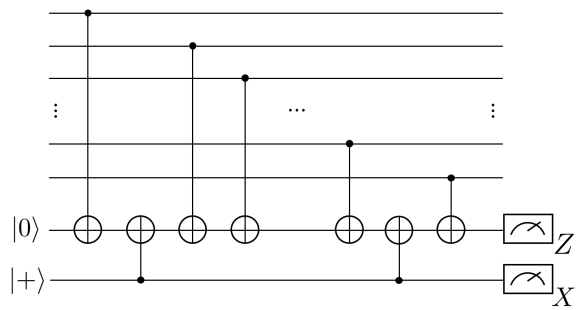


Figure 5.1: A 1-flag circuit for measuring operator $Z^{\otimes m} \otimes I^{\otimes n-m}$ in the normal permutation.

and $Q \in \{I, X\}$, then the data error is of weight ≤ 1 and the flag outcome is $+1$.

(b) If an error from a faulty CNOT gate is $P \otimes Q$ where $P = I$ and $Q \in \{Y, Z\}$, the data error is of the form $Z^{\otimes p} \otimes I^{n-p}$ where $p \in \{1, \cdots, m\}$. In the cases where the data error has weight > 1, the flag outcome is -1.

(c) If an error from a faulty CNOT is $P \otimes Q$ where $P \in \{X, Y, Z\}$ and $Q \in \{Y, Z\}$, the data error is of the form $Z^{\otimes p-1} \otimes (PZ) \otimes I^{n-p}$ where $p \in \{1, \cdots, m\}$. In the cases where the data error has weight > 1, the flag outcome is -1.

Data errors of the form (b) or (c) arise due to the propagation of Z errors from the target qubit to the control qubit of CNOT gates as explained in Section 4.1. Each of such errors is equivalent to an operator of the form $Z^{\otimes p} \otimes I^{n-p}$ or $Z^{\otimes p-1} \otimes (PZ) \otimes I^{n-p}$ up to the multiplication of stabilizer generator $Z^{\otimes m} \otimes I^{\otimes n-m}$. In addition, if the error caused by a faulty CNOT is $Z \otimes Z$, this error can be viewed as an error $I \otimes Z$ caused by the preceding CNOT gate.

Let $\mathcal{E}_+$ and $\mathcal{E}_-$ be sets of errors corresponding to the cases that circuit flags (flag outcome is $+1$) and does not flag (flag outcome is -1), respectively. Consider an $[\![n, k, d]\!]$ CSS code C constructed from $[n, k_x, d_x]$ classical code C_x and $[n, k_z, d_z]$ classical code C_z using Theorem 3.3. Since all errors in $\mathcal{E}_+$ have weight at most 1, it is clear that $\mathcal{E}_+$ is distinguishable by C whenever $d \geq 3$. Now let us consider Z errors of the form (b) in $\mathcal{E}_-$. The distinguishability of these errors only depends on the code C_x. In addition, an error of the form (c) in $\mathcal{E}_-$ can be written as a product of an error of the form (b) and a weight-1

X-type error. Thus, if C_z is a classical code with $d_z \geq 3$ and all errors of the form (b) are distinguishable by C_x, then $\mathcal{E}_-$ is distinguishable by C. The similar argument is also applicable to a circuit for measuring X-type generator of the form $X^{\otimes m} \otimes I^{\otimes n-m}$.

5.2 Consecutive errors and cyclic symmetry

In the previous section, we find that the ability of the CSS code to distinguish errors of the form $Z^{\otimes p} \otimes I^{n-p}$ is crucial because it will make the 1-flag circuit satisfy the flag 1-FTEC condition. We may define a set of errors of the form $Z^{\otimes p} \otimes I^{n-p}$ or $X^{\otimes p} \otimes I^{n-p}$ as follows:

Definition 5.2 *A* consecutive-Z error set $\mathcal{E}_n^z$ *and a* consecutive-X error set $\mathcal{E}_n^x$ *are sets of the form,*

$$\mathcal{E}_n^z = \{Z^{\otimes p} \otimes I^{\otimes n-p} : p \in \{0, 1, \ldots, n-1\}\}, \tag{5.1}$$

$$\mathcal{E}_n^x = \{X^{\otimes p} \otimes I^{\otimes n-p} : p \in \{0, 1, \ldots, n-1\}\}. \tag{5.2}$$

By Definitions 5.1 and 5.2, we can prove the following lemma.

Lemma 5.1 *Let C be a CSS code constructed from the classical cyclic codes C_x and C_z following Theorem 3.3 with parity check matrices H_x and H_z of the form,*

$$H_x = \begin{pmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,n} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,n} \\ \ldots & & & \ldots \\ x_{r_x,1} & x_{r_x,2} & \ldots & x_{r_x,n} \end{pmatrix}, \tag{5.3}$$

$$H_z = \begin{pmatrix} z_{1,1} & z_{1,2} & \ldots & z_{1,n} \\ z_{2,1} & z_{2,2} & \ldots & z_{2,n} \\ \ldots & & & \ldots \\ z_{r_z,1} & z_{r_z,2} & \ldots & z_{r_z,n} \end{pmatrix}, \tag{5.4}$$

and let $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$ be consecutive-Z and consecutive-X error sets, respectively. Then,

1. *$\mathcal{E}_n^z$ is distinguishable by C iff for all $p, q \in \{0, ..., n-1\}$ such that $p < q$, there exists $i \in \{1, \ldots, r_x\}$ such that $x_{i,p+1} \oplus \cdots \oplus x_{i,q} = 1$.*

2. *$\mathcal{E}_n^x$ is distinguishable by C iff for all $p, q \in \{0, ..., n-1\}$ such that $p < q$, there exists $i \in \{1, \ldots, r_z\}$ such that $z_{i,p+1} \oplus \cdots \oplus z_{i,q} = 1$.*

Proof:
We will prove that $\mathcal{E}_n^z$ is distinguishable by C iff for all $p, q \in \{0, ..., n-1\}$ such that $p < q$, there exists $i \in \{1, \ldots, r_x\}$ such that $x_{i,p+1} \oplus \cdots \oplus x_{i,q} = 1$. Consider errors $E_p = Z^{\otimes p} \otimes I^{\otimes n-p}$ and $E_q = Z^{\otimes q} \otimes I^{\otimes n-q}$ where $p, q \in \{0, 1, \ldots, n-1\}, p < q$. Let $s(E_p), s(E_q) \in \{0, 1\}^r$ be error syndromes corresponding to errors E_p and E_q, respectively. By Definition 5.1, E_p and E_q are distinguishable by C iff $s(E_p) \neq s(E_q)$, i.e., there exists $i \in \{1, 2, \cdots, r_x\}$ such that $s(E_p)_i \neq s(E_q)_i$, where $s(E_p)_i$ and $s(E_q)_i$ correspond to the i^{th} component of $s(E_p)$ and $s(E_q)$. From the parity check matrix H_x, the i^{th} component of $s(E_p)$ and $s(E_q)$ is given by

$$s(E_p)_i = x_{i,1} \oplus x_{i,2} \oplus \cdots \oplus x_{i,p}, \tag{5.5}$$

$$s(E_q)_i = x_{i,1} \oplus x_{i,2} \oplus \cdots \oplus x_{i,q}. \tag{5.6}$$

From Eqs. (5.5) and (5.6), we have that

$$\begin{aligned} s(E_p)_i \neq s(E_q)_i &\Leftrightarrow s(E_p)_i \oplus s(E_q)_i = 1 \\ &\Leftrightarrow x_{i,p+1} \oplus \cdots \oplus x_{i,q} = 1. \end{aligned} \tag{5.7}$$

Thus, $\mathcal{E}_n^z$ is distinguishable by C iff for all $p, q \in \{0, ..., n-1\}$ such that $p < q$, there exists $i \in \{1, 2, \cdots, r_x\}$ such that

$$x_{i,p+1} \oplus \cdots \oplus x_{i,q} = 1. \tag{5.8}$$

The proof of statement for $\mathcal{E}_n^x$ is similar. $\square$

Lemma 5.1 provides the sufficient and necessary conditions for distinguishing errors in the consecutive form. In addition, these conditions can be simplified if the parity check matrices H_x and H_z have some symmetries. Let us consider the case that there exists row i of H_x such that $x_{i,p+1} \oplus \cdots \oplus x_{i,q} = 1$ for given p and q, and there exists row i' of H_x which is a cyclic shift of row i. In this case, there must exist p' and q' such that $x_{i',p'+1} \oplus \cdots \oplus x_{i,q'} = 1$. This argument suggests that the cyclic symmetry of cyclic CSS codes can be used to simplify Lemma 5.1. This is because each row of the parity check matrix of a classical cyclic code as in Eq. (3.6) is a cyclic shift of other rows. The following lemma is a simplified version of Lemma 5.1 for a cyclic CSS code.

Lemma 5.2 *Let C be a CSS code constructed from the classical cyclic codes C_x and C_z following Theorem 3.3 with parity check matrices H_x and H_z of the form,*

$$H_x = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \dots & & & \dots \\ x_{r_x,1} & x_{r_x,2} & \dots & x_{r_x,n} \end{pmatrix}, \tag{5.9}$$

$$H_z = \begin{pmatrix} z_{1,1} & z_{1,2} & \dots & z_{1,n} \\ z_{2,1} & z_{2,2} & \dots & z_{2,n} \\ \dots & & & \dots \\ z_{r_z,1} & z_{r_z,2} & \dots & z_{r_z,n} \end{pmatrix}, \tag{5.10}$$

where $x_{i_1+1,(j+1) mod\ n} = x_{i_1,j}$ for all $i_1 \in \{1, 2, \dots, r_x - 1\}$ and $z_{i_2+1,(j+1) mod\ n} = z_{i_2,j}$ for all $i_2 \in \{1, 2, \dots, r_z - 1\}$, $j \in \{1, \dots, n\}$. Let $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$ be consecutive-Z and consecutive-X error sets, respectively. Then,

1. *$\mathcal{E}_n^z$ is distinguishable by C iff for all $u_x \in \{1, \dots, n-1\}$, there exists $i \in \{1, \dots, r_x\}$ such that $x_{i,1} \oplus \cdots \oplus x_{i,u_x} = 1$.*
2. *$\mathcal{E}_n^x$ is distinguishable by C iff for all $u_z \in \{1, \dots, n-1\}$, there exists $i \in \{1, \dots, r_z\}$ such that $z_{i,1} \oplus \cdots \oplus z_{i,u_z} = 1$.*

Proof:
Here we will prove that $\mathcal{E}_n^z$ is distinguishable by C iff for all $u_x \in \{1, \dots, n-1\}$, there exists $i \in \{1, \dots, r_x\}$ such that $x_{i,1} \oplus \cdots \oplus x_{i,u_x} = 1$. Applying Lemma 5.1, we would

like to prove that there exists $i \in \{1, \cdots, r_x\}$ such that $x_{i,p+1} \oplus \cdots \oplus x_{i,q} = 1$ for all $p, q \in \{0, ..., n-1\}$, $p < q$ iff there exist $i' \in \{1, \cdots, r_x\}$ such that $x_{i',1} \oplus \cdots \oplus x_{i',u_x} = 1$ for all $u_x \in \{1, 2, \ldots, n-1\}$.
($\Rightarrow$) By choosing $p = 0$, the proof is trivial.
($\Leftarrow$) Assume that there exists $i' \in \{1, \cdots, r_x\}$ such that $x_{i',1} \oplus \cdots \oplus x_{i',u_x} = 1$ for all $u_x \in \{1, 2, \ldots, n-1\}$. Let $p, q \in \{0, 1, \ldots, n-1\}$ be integers such that $p < q$. Let S be the stabilizer group corresponding to the code C. We want to find an operator $M \in S$ that can distinguish between $E_p = Z^{\otimes p} \otimes I^{\otimes n-p}$ and $E_q = Z^{\otimes q} \otimes I^{\otimes n-q}$. By assumption, there exists generator $g^x_{i'} = (x_{i',1}, \ldots, x_{i',n})$ such that $x_{i',1} \oplus \cdots \oplus x_{i',q-p} = 1$. Since C is constructed from a classical cyclic code C_x, an operator $M = (x_{i',n-p+1}, \ldots, x_{i',n}, x_{i',1}, \ldots, x_{i',q-p}, \ldots, x_{i',n-p})$ which is a cyclic shift of $g^x_{i'}$ is also in the stabilizer. Note that the $(p+1)^{\text{th}}$ component of M is $x_{i',1}$ and the q^{th} component of M is $x_{i',q-p}$. Let $s^M(E_p)$ and $s^M(E_q)$ be the measurement outcomes corresponding to the measurement of E_p and E_q by M. Then we have

$$s^M(E_p) = x_{i',n-p+1} \oplus \cdots \oplus x_{i',n}, \tag{5.11}$$

$$s^M(E_q) = x_{i',n-p+1} \oplus \cdots \oplus x_{i',n} \oplus x_{i',1} \oplus \cdots \oplus x_{i',q-p}. \tag{5.12}$$

Now given that $x_{i',1} \oplus \cdots \oplus x_{i',q-p} = 1$, we have $s^M(E_p) \neq s^M(E_q)$. Since $M \in S$, there exists a set of $a_1, \cdots, a_{r_x} \in \{0, 1\}$ such that $M = (g^x_1)^{a_1} \cdots (g^x_{r_x})^{a_{r_x}}$ where in symplectic form $g^x_i = (x_{i,1}, \ldots, x_{i,n})$. Observe that $s^M(E_p)$ and $s^M(E_q)$ can be written as

$$s^M(E_p) = \sum_{i=1}^{r_x} a_i s(E_p)_i = \sum_{i=1}^{r_x} \sum_{j=1}^{p} a_i x_{i,j}, \tag{5.13}$$

$$s^M(E_q) = \sum_{i=1}^{r_x} a_i s(E_q)_i = \sum_{i=1}^{r_x} \sum_{j=1}^{q} a_i x_{i,j}. \tag{5.14}$$

Thus, $s^M(E_p) \neq s^M(E_q)$ iff

$$\sum_{i=1}^{r_x} \sum_{j=p+1}^{q} a_i x_{i,j} = 1. \tag{5.15}$$

From Eq. (5.15), there exists an $i \in \{1, \cdots, r_x\}$ such that $a_i = 1$ and $x_{i,p+1} \oplus \cdots \oplus x_{i,q} = 1$, which means that E_p and E_q are distinguishable by some generator g_i^x. Note that this is true for all $p, q \in \{0, \cdots, n-1\}$ with $p < q$.

The proof of statement for $\mathcal{E}_n^x$ is similar. $\square$

In the following theorem, we will show that if a CSS code is constructed from two classical cyclic error-correcting codes, the conditions in Lemma 5.2 are automatically satisfied.

Theorem 5.1 *Let C be an $[\![n,k,d]\!]$ CSS code constructed from the $[n,k_x,d_x]$ classical cyclic code C_x and the $[n,k_z,d_z]$ classical cyclic code C_z, and let $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$ be consecutive-Z and consecutive-X error sets, respectively. If both $d_x, d_z \geq 3$, then $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$ are distinguishable by C.*

Proof:
Let the parity check matrix H_x of the code C_x and H_z of the code C_z be in the form

$$H_x = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \dots & & & \dots \\ x_{r_x,1} & x_{r_x,2} & \dots & x_{r_x,n} \end{pmatrix}, \tag{5.16}$$

$$H_z = \begin{pmatrix} z_{1,1} & z_{1,2} & \dots & z_{1,n} \\ z_{2,1} & z_{2,2} & \dots & z_{2,n} \\ \dots & & & \dots \\ z_{r_z,1} & z_{r_z,2} & \dots & z_{r_z,n} \end{pmatrix}, \tag{5.17}$$

where $x_{i_1+1,(j+1)\text{mod } n} = x_{i_1,j}$ for all $i_1 \in \{1,\dots,r_x-1\}$ and $z_{i_2+1,(j+1)\text{mod } n} = z_{i_2,j}$ for all $i_2 \in \{1,\dots,r_z-1\}$, $j \in \{1,\dots,n\}$ since C_x and C_z are cyclic codes. By Lemma 5.2, we know that $\mathcal{E}_n^z$ is distinguishable by C iff for all $u_x \in \{1,2,...,n-1\}$, there exists $i \in \{1,\cdots,r_x\}$ such that $x_{i,1} \oplus \cdots \oplus x_{i,u_x} = 1$. Suppose by contradiction that there exists a $u_x \in \{1,\dots,n-1\}$ such that $x_{i,1} \oplus \cdots \oplus x_{i,u_x} = 0$ for all $i \in \{1,\dots,r_x\}$. Now, let $E_{p_x} = Z^{\otimes p_x} \otimes I^{\otimes n-p_x}$, $E_{q_x} = Z^{\otimes q_x} \otimes I^{\otimes n-q_x}$ with $q_x - p_x = u_x$. Further, let $M_i^x = (x_{i,n-p_x+1},\dots,x_{i,n},x_{i,1},\dots,x_{i,q_x-p_x},\dots,x_{i,n-p_x})$ be a cyclic shift of the generator g_i^x for all

$i \in \{1, \dots, r_x\}$. By assumption, errors E_{p_x} and E_{q_x} cannot be distinguished by the operator M_i^x for all $i \in \{1, \cdots, r_x\}$ (see the proof of Lemma 5.2). Observe that $\{M_1^x, \dots, M_{r_x}^x\}$ is also a generating set for the X-stabilizers. Thus, E_{p_x} and E_{q_x} are indistinguishable. The indistinguishability of E_{p_x} and E_{q_x} by M_1^x gives

$$x_{1,1} \oplus \cdots \oplus x_{1,u_x} = 0. \tag{5.18}$$

From the cyclic property, any cyclic shift of M_1^x cannot distinguish between E_{p_x} and E_{q_x} as well. This gives the following conditions:

$$\begin{aligned} x_{1,2} \oplus \cdots \oplus x_{1,u_x+1} &= 0 \\ x_{1,3} \oplus \cdots \oplus x_{1,u_x+2} &= 0 \\ &\vdots \\ x_{1,n} \oplus x_{1,1} \oplus \cdots \oplus x_{1,u_x-1} &= 0 \end{aligned} \tag{5.19}$$

From Eqs. (5.18) and (5.19), we have that $x_{1,j} = x_{1,(u_x+j)\text{mod } n}$ for all $j \in \{1, \dots, n\}$. Let $t_x = \text{GCD}(u_x, n)$, the greatest common divisor of u_x and n. The conditions become

$$x_{1,j} = x_{1,j+t_x} = x_{1,j+2t_x} = \dots = x_{1,j+n-t_x}, \tag{5.20}$$

for all $j \in \{1, \dots, u_x\}$. Repeating the above steps for all $M_i^x \in \{M_1^x, \cdots, M_{r_x}^x\}$, we obtain

$$x_{i,j} = x_{i,j+t_x} = x_{i,j+2t_x} = \dots = x_{i,j+n-t_x}, \tag{5.21}$$

for all $i \in \{1, \dots, r_x\}$, $j \in \{1, \dots, u_x\}$.

From the above, we see that any error of the form $Z_{l_x} Z_{l_x+t_x}$ (where $l_x \in \{1, \cdots, n-t_x\}$) commutes with all stabilizer generators. Now let us consider two cases:

- **Case 1:** At least one operator of the form $Z_{l_x} Z_{l_x+t_x}$ is not in the stabilizer.

  In this case C has distance at most two. This contradicts our assumption that $\min\{d_x, d_z\} \geq 3$.

- **Case 2:** All operators of the form $Z_{l_x}Z_{l_x+t_x}$ are in the stabilizer.

  In this case, there exists a set of coefficients $a_1, \cdots, a_{r_z} \in \{0,1\}$ such that $\prod_i (g_i^z)^{a_i} = Z_{l_x}Z_{l_x+t_x}$ where g_i^z is the i^{th} row of H_z in binary symplectic form. Since H_z generates $C_z^\perp$ and $C_z^\perp \subseteq C_x$ by construction of CSS codes. Thus we have that $Z_{l_x}Z_{l_x+t_x} \in C_x$. Now since the distance of classical codes is given by the minimum weight codeword, we have that $d_x \leq 2$ which contradicts our assumption that $d_x \geq 3$.

Similarly, assume by contradiction that there exists a $u_z \in \{1, \ldots, n-1\}$ such that $z_{i,1} \oplus \cdots \oplus z_{i,u_z} = 0$ for all $i \in \{1, \ldots, r_z\}$, we will have that C has $d \leq 2$ or $d_z \leq 2$. Thus, $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$ are distinguishable by C if $d_x, d_z \geq 3$. □

Since every classical error-correcting code has distance at least 3, Theorem 5.1 implies that every CSS code constructed from two classical cyclic error-correcting codes can distinguish consecutive errors in $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$. However, the 1-flag circuit in Fig. 5.1 cannot be directly applied to the measurement of an X-type or Z-type generator of the cyclic CSS code since the data errors might not be in the consecutive form without qubit permutation. Moreover, permuting qubits to transforming the generator into the form $X^{\otimes m} \otimes I^{\otimes n-m}$ or $Z^{\otimes m} \otimes I^{\otimes n-m}$ will break the cyclic symmetry of the code, and $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$ might be no longer distinguishable. In the next section, we will use Theorem 5.1 to find a 1-flag circuit for cyclic CSS codes of distance 3 and the corresponding flag-FTEC protocol satisfying the ECRP and the ECCP in Definitions 4.13 and 4.14.

5.3 Flag circuit and fault-tolerant protocol for cyclic CSS codes

In the previous section, we prove that every CSS code constructed from two classical cyclic error-correcting codes can distinguish all errors in the consecutive error sets $\mathcal{E}_n^z$ and $\mathcal{E}_n^x$ defined in Definition 5.2. However, errors caused by a single fault in a 1-flag circuit corresponding to the measurement of a cyclic CSS code's generator might not be in the consecutive form in general. Our goal in this section is to construct a 1-flag circuit for measuring generators of a cyclic CSS code and corresponding error correction protocol such that the error distinguishability argument discussed in Section 5.1 is applicable.

Suppose that the stabilizer generator being measured is of the form

$$P = Z^{\otimes a_1} \otimes I^{\otimes b_1} \otimes Z^{\otimes a_2} \otimes I^{\otimes b_2} \otimes \cdots \otimes Z^{\otimes a_m} \otimes I^{\otimes b_m}. \tag{5.22}$$

The i^{th} block contains a_i qubits, which are from the $\sum_{j=1}^{i-1}(a_j + b_j) + 1$'th qubit to the $\sum_{j=1}^{i-1}(a_j + b_j) + a_i$'th qubit. The circuit for measuring this operator is given in Fig. 5.2. This circuit requires four ancilla qubits: one qubit for syndrome measurement and three qubits for flag measurement.

Notice that the blue, green and orange CNOT gates in the circuit of Fig. 5.2 always come in pairs. This is to ensure that when fault-free, the circuit implements a projective measurement of the stabilizer without flagging. In what follows, we will refer to the first blue, green or orange CNOT of a pair as an *open* CNOT and the second blue, green or orange CNOT as a *closed* CNOT. Given these definitions, we have the following claim:

Claim 5.1 *During the measurement of $Z^{\otimes a_1} \otimes I^{\otimes b_1} \otimes Z^{\otimes a_2} \otimes I^{\otimes b_2} \otimes \cdots \otimes Z^{\otimes a_m} \otimes I^{\otimes b_m}$ using the circuit in Fig. 5.2, the following can occur:*

1. *If there are no faults, none of the f_i ancilla qubits will flag.*

2. *A fault at a CNOT location resulting in a ZZ error is equivalent to the prior CNOT failing resulting in an IZ error (here Z acts on the target qubit).*

3. *Suppose that a fault occurs on one of the red CNOT's and causes a Z error on the ancilla m_0. If the fault occurs on block a_1, only f_1 will flag. Otherwise, if it occurs on block a_i where $i \geq 2$, only the ancillas f_1 and f_i will flag.*

4. *Suppose that a fault occurs on a blue or green CNOT. Let the control qubit be the ancilla f_i. If it is the* open *CNOT and causes a Z error on ancilla m_0, the ancillas f_1, f_i, and f_{i-1} will flag. However, if it is the* closed *CNOT and causes a Z error on the ancilla m_0, the ancillas f_1 and f_{i+1} will flag. However if the fault occurs on*

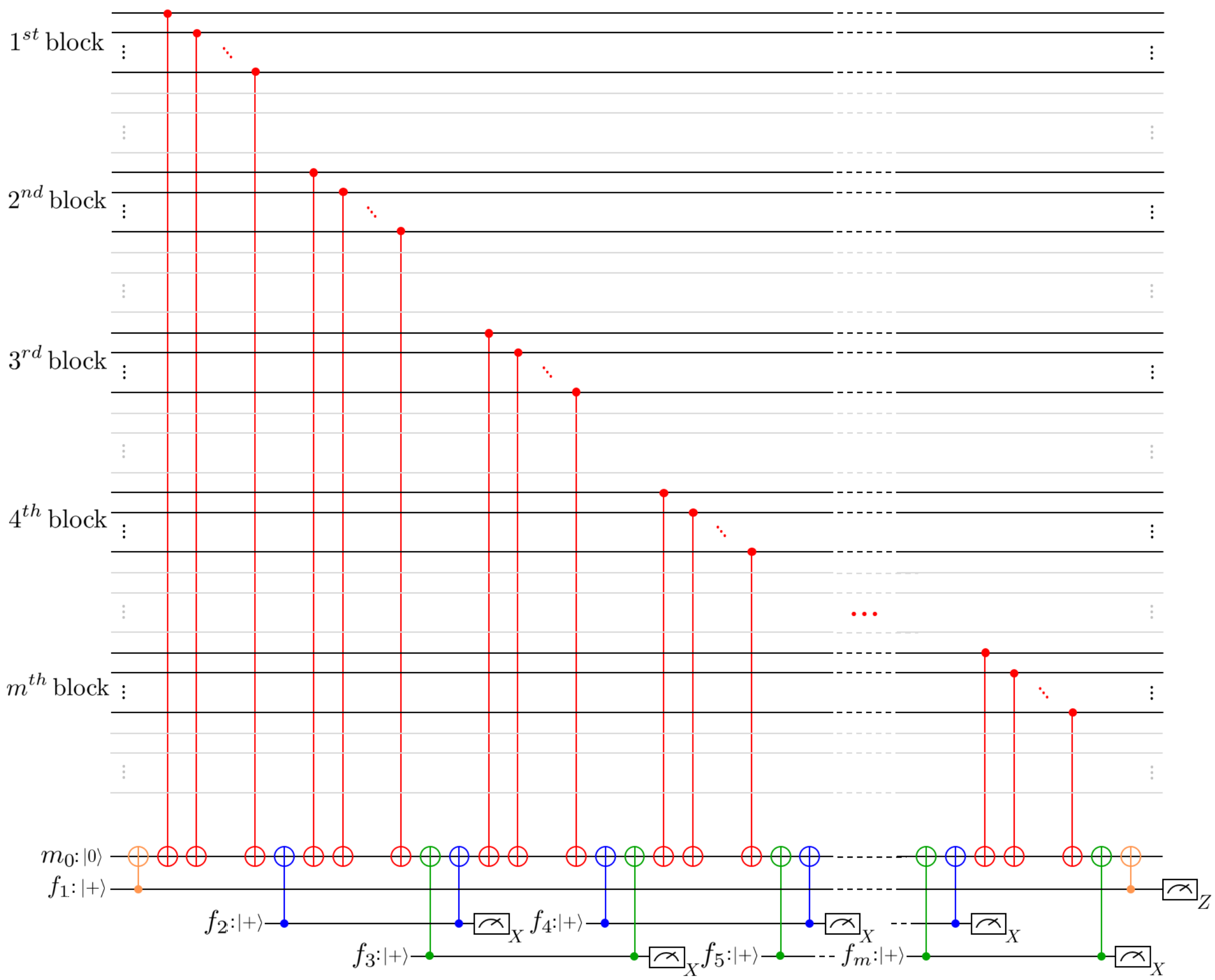


Figure 5.2: A 1-flag circuit for measuring operator of the form $Z^{\otimes a_1} \otimes I^{\otimes b_1} \otimes Z^{\otimes a_2} \otimes I^{\otimes b_2} \otimes \cdots \otimes Z^{\otimes a_m} \otimes I^{\otimes b_m}$.

a blue or green CNOT's at the boundary [1]*, if the open CNOT of f_2 is faulty, f_1 and f_2 will flag, and if the closed CNOT of f_m is faulty, only f_1 will flag.*

5. *A fault occurring at an orange CNOT gate will not cause a data qubit error (since a Z spreading to all qubits is equivalent to the stabilizer being measured). Furthermore, only the ancilla f_1 can flag in this case (depending on whether the error was of the form IZ or ZZ and also if it occurred on the open or closed orange CNOT).*

From the above claim, one can verify that a single fault resulting in a data qubit error E with $\min(\mathrm{wt}(E), \mathrm{wt}(EP)) > 1$ (where P is given by Eq. (5.22)) will always cause at least one flag qubit to flag. Thus, the circuit in Fig. 5.2 is a 1-flag circuit. Note that an analogous claim can be made for X-type generators.

Using Theorem 5.1 and Claim 5.1, we now describe a FTEC protocol that satisfies Definitions 4.13 and 4.14 for distance-three cyclic CSS codes using a procedure adapted from [4]. In what follows, we define s_i to be the syndrome obtained during round i (either using flag or non-flag circuits).

FTEC Protocol:

Let C be an $[\![n, k, d]\!]$ cyclic CSS code satisfying Theorem 5.1 with stabilizer $S = \langle g_1, \cdots, g_r \rangle$. Let $C(g_i)$ be the 1-flag circuit of Fig. 5.2 for generator g_i. Repeat the syndrome measurement (measurement of all stabilizer generators) using the 1-flag circuits until one of the following is satisfied:

1. If the syndrome is repeated twice in a row and there are no flags, apply $E_{\min}(s_1)$.

2. If there are no flags and the syndromes s_1 and s_2 differ, repeat the syndrome measurement using non-flagged circuits. Apply the correction $E_{\min}(s_3)$.

3. If f_1 doesn't flag but f_i flags (with $i \geq 2$) during round one, stop. Repeat the syndrome measurement using non-flag circuits and apply $E_{\min}(s_2)$. If there are no

[1] By boundary we are referring to either the first blue CNOT after the block a_1 or the last green CNOT after the block a_m.

flags in the first round but in round two f_i flags and f_1 doesn't flag, stop. Apply $E_{\min}(s_1)$.

4. If at anytime during the protocol f_1 flags, stop and do one of the following:

   (a) Suppose $f_i = 0$ for all $i \geq 2$. Repeat the syndrome measurement using non-flag circuits. If there is an element E in $\mathcal{E}_n^z$ or $\mathcal{E}_n^x$ that satisfies $s(E) = s$, apply E. Otherwise, apply $E_{\min}(s)$.

   (b) If there is only one i such that $f_i = 1$ (with $i \geq 2$), apply $I^{\otimes a_1} \otimes Z^{\otimes b_1} \otimes I^{\otimes a_2} \otimes Z^{\otimes b_2} \otimes \cdots \otimes I^{\otimes a_{i-1}} \otimes Z^{\otimes b_{i-1}} \otimes I^{n-k}$ to the data if the generator being measured is a Z-type generator or $I^{\otimes a_1} \otimes X^{\otimes b_1} \otimes I^{\otimes a_2} \otimes X^{\otimes b_2} \otimes \cdots \otimes I^{\otimes a_{i-1}} \otimes X^{\otimes b_{i-1}} \otimes I^{n-k}$ if it is an X-type generator (where $k = \sum_{j=0}^{i-1}(a_j + b_j)$). Repeat the syndrome measurement using non-flag circuits yielding syndrome s. If there is an element E in $\mathcal{E}_n^z$ or $\mathcal{E}_n^x$ that satisfies $s(E) = s$, apply E. Otherwise, apply $E_{\min}(s)$.

   (c) Suppose there is an i such that $f_i = 1$ and $f_{i+1} = 1$. Apply $I^{\otimes a_1} \otimes Z^{\otimes b_1} \otimes I^{\otimes a_2} \otimes Z^{\otimes b_2} \otimes \cdots \otimes I^{\otimes a_{i-1}} \otimes Z^{\otimes b_{i-1}} \otimes I^{n-k}$ to the data if the generator being measured is a Z-type generator or $I^{\otimes a_1} \otimes X^{\otimes b_1} \otimes I^{\otimes a_2} \otimes X^{\otimes b_2} \otimes \cdots \otimes I^{\otimes a_{i-1}} \otimes X^{\otimes b_{i-1}} \otimes I^{n-k}$ if it is an X-type generator (where $k = \sum_{j=0}^{i-1}(a_j + b_j)$). Repeat the syndrome measurement using non-flag circuits yielding syndrome s. If there is an element E in $\mathcal{E}_n^z$ or $\mathcal{E}_n^x$ that satisfies $s(E) = s$, apply E. Otherwise, apply $E_{\min}(s)$.

To see that the above protocol satisfies Definitions 4.13 and 4.14, we will assume that there is at most one fault during the protocol. If a fault in any of the CNOT gates introduces a Z error on ancilla m_0, then f_1 will flag (unless the first orange CNOT introduces an error of the form ZZ or the last orange CNOT introduces an error of the form IZ which in both cases, there will be no data qubit error). Furthermore, if f_1 doesn't flag but f_i flags with $i \geq 2$, then the fault could either have been caused by a measurement error, idle qubit error on the ancilla f_i, or an error on the control qubit of the CNOT gate interacting with f_i. However in all three cases, the error could not have spread to the data. By repeating the syndrome measurement and applying $E_{\min}(s)$, both criteria of Definitions 4.13 and 4.14 will be satisfied. Note that if f_i flags during round two, then the syndrome obtained during round one corresponds to the data qubit error (since there could not have been a measurement error giving the wrong syndrome during the first round), so correcting using s_1 will again satisfy Definitions 4.13 and 4.14.

Next, let us consider the case where none of the f_i ancillas flag. By the circuit construction, a single fault can introduce an error E with $\text{wt}(E) \leq 1$. If the same syndrome is repeated twice in a row, i.e., $s_1 = s_2 = s$, then applying $E_{\min}(s)$ can result in a data error of weight at most one. If $s_1 \neq s_2$, then a fault occurred in either the first or second round. Thus repeating the syndrome measurement a third time and applying $E_{\min}(s_3)$ will remove the data errors or project the code back to the codespace (following Proposition 2.1).

Next we consider the case where a fault happens on a red CNOT introducing a Z error on the ancilla m_0. If the fault occurred in the first block (a_1), then only f_1 will flag. If there is no input error, then the data qubit error will belong to either $\mathcal{E}_n^z$ or $\mathcal{E}_n^x$. By Theorem 5.1, errors in the set $\mathcal{E}_n^z$ or $\mathcal{E}_n^x$ can be distinguished. Thus applying the correction in 4 a) of the protocol will remove the error if there are no input errors. If there is an input error, then applying $E_{\min}(s)$ will project the code back to the codespace. Now if the fault occurs on any other block, then f_1 will flag and there will be only one $i \geq 2$ such that f_i flags. Applying $I^{\otimes a_1} \otimes Z^{\otimes b_1} \otimes I^{\otimes a_2} \otimes Z^{\otimes b_2} \otimes \cdots \otimes I^{\otimes a_{i-1}} \otimes Z^{\otimes b_{i-1}} \otimes I^{n-k}$ to the data if the generator being measured is a Z-type generator or $I^{\otimes a_1} \otimes X^{\otimes b_1} \otimes I^{\otimes a_2} \otimes X^{\otimes b_2} \otimes \cdots \otimes I^{\otimes a_{i-1}} \otimes X^{\otimes b_{i-1}} \otimes I^{n-k}$ if it is an X-type generator guarantees that the resulting error belongs to the set $\mathcal{E}_n^z$ or $\mathcal{E}_n^x$ (since the Z error will spread to all qubits in block a_i to a_m). Repeating the same arguments as above and using 4 b) will ensure that both criteria of Definitions 4.13 and 4.14 are satisfied.

Lastly, if a fault occurs on a blue or green CNOT, then from Claim 5.1 either the case in 4 b) or 4 c) will be satisfied. However in both cases, the Z error will spread to the data in the same way. Hence the correction proposed in 4 c) will satisfy the fault-tolerance criteria of Definitions 4.13 and 4.14.

5.4 Discussions and conclusion

In this work we used the symmetries of CSS codes constructed from classical cyclic codes in order to prove that errors written in consecutive form (as in Definition 5.2) can be distinguished. From these properties we were able to obtain a 1-flag circuit along with a flag-FTEC protocol which satisfies the fault-tolerance criteria of Definitions 4.13 and 4.14 when there is at most one fault. The 1-flag circuit requires only four ancilla qubits. This number does not grow as the block length gets larger, making our protocol advantageous in the implementation where resources are limited. We note that not all cyclic CSS codes are Hamming codes and therefore the methods in [5] cannot be directly applied, thus providing further motivation for our work.

In general, cyclic CSS codes do not satisfy the sufficient condition required for flag fault-tolerance presented in [4] (one example is the family of Hamming codes which can be made cyclic). Nevertheless, using the techniques presented in this work, a flag fault-tolerant protocol can still be achieved.

Note that for all CSS codes, the stabilizer generators being measured are of the form $X^{\otimes m} \otimes I^{\otimes n-m}$ or $Z^{\otimes m} \otimes I^{\otimes n-m}$ up to qubit permutations. Thus data qubit errors arising from faulty CNOT gates will be expressed in consecutive form. The errors of this form are distinguishable iff the sub-matrices of the X and Z stabilizers satisfy Lemma 5.1. In our work, we use the symmetry of the cyclic codes to simplify Lemma 5.1 into Lemma 5.2. We believe that Lemma 5.1 can be simplified by using symmetries found in other families of quantum codes. With appropriate t-flag circuits and operations dependent on the flag measurement outcome, this may lead to new flag-FTEC protocols.

Another interesting avenue is finding non-cyclic quantum codes for which a version of Theorem 5.1 can be applied. We note that for such codes, the same 1-flag circuit as in Fig. 5.2 along with the flag-FTEC protocol of Section 5.3 can be used. The reason is that the key property used by these schemes is based on the distinguishability of consecutive errors.

Note that there are quantum cyclic codes which are not CSS codes for which flag fault-tolerant schemes are still possible. For instance, a flag-FTEC protocol for the $[\![5, 1, 3]\!]$ code was devised in [5]. We believe it could be interesting to generalize the ideas presented in this work to non-CSS cyclic quantum codes. However, we leave this problem for future work.

The flag-FTEC protocol for cyclic CSS codes presented in this work is based on the same assumptions as discussed in Section 4.2.3. One of the important assumptions is that the qubit measurement and state preparation must be fast since we reuse some flag qubits in the protocol (as we can see in Fig. 5.2). Some examples of physical systems where preparation and measurement can be quickly performed are ion traps and superconducting qubits. If we do not reuse flag qubits, however, the number of required ancillas will be $m+1$ for an operator being measured of the form $P = Z^{\otimes a_1} \otimes I^{\otimes b_1} \otimes Z^{\otimes a_2} \otimes I^{\otimes b_2} \otimes \cdots \otimes Z^{\otimes a_m} \otimes I^{\otimes b_m}$ instead of 4.

One important feature of flag-FTEC protocols is that the number of required ancillas is very small compared to other FTEC schemes. We believe that if fewer ancillas are required, the accuracy threshold p_T will increase since the number of locations will decrease in total. However, we should point out that subsequent syndrome measurements are also required in a flag-FTEC protocol and may increase the total number of locations in the protocol. The answer of whether the accuracy threshold for a flag-FTEC protocol is greater or smaller compared to other FTEC schemes is still unknown.

Lastly, we point out that cyclic CSS codes which satisfy the condition in Theorem 5.1 are not limited to distance-three codes. Therefore, interesting future work would be to use the methods of [4] to obtain flag-FTEC schemes for higher distance codes. In particular, the main challenge stems from finding t-flag circuits as in Fig. 5.2 for $t > 1$.

Bibliography

[1] Panos Aliferis, Daniel Gottesman, and John Preskill. Quantum accuracy threshold for concatenated distance-3 codes. *Quantum Info. Comput.*, 6(2):97–165, March 2006.

[2] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005.

[3] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996.

[4] Christopher Chamberland and Michael E. Beverland. Flag fault-tolerant error correction with arbitrary distance codes. *Quantum*, 2:53, February 2018.

[5] Rui Chao and Ben W. Reichardt. Quantum error correction with only two extra qubits. *arXiv:quant-ph/1705.02329*, 2017.

[6] David P DiVincenzo et al. The physical implementation of quantum computation. *arXiv preprint quant-ph/0002077*, 2000.

[7] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Physical Review A*, 54(3):1862, 1996.

[8] Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv preprint quant-ph/9705052*, 1997.

[9] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.

[10] Daniel Gottesman. Lecture notes in quantum error correction and fault tolerance, January 2018.

[11] E. Knill. Scalable quantum computing in the presence of large detected-error rates. *Phys. Rev. A*, 71:042322, Apr 2005.

[12] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900–911, Feb 1997.

[13] Ruihu Li and Xueliang Li. Quantum codes constructed from binary cyclic codes. *International Journal of Quantum Information*, 02(02):265–272, 2004.

[14] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland, New York, 1977.

[15] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[16] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[17] Peter W. Shor. Fault-tolerant quantum computation. *Proceedings., 37th Annual Symposium on Foundations of Computer Science*, pages 56–65, 1996.

[18] P.W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings., 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[19] A. M. Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, 1996.

[20] A. M. Steane. Active stabilization, quantum computation, and quantum state synthesis. *Phys. Rev. Lett.*, 78:2252–2255, Mar 1997.

[21] A. M. Steane. Enlargement of calderbank-shor-steane quantum codes. *IEEE Transactions on Information Theory*, 45(7):2492–2495, Nov 1999.

[22] Theerapat Tansuwannont, Christopher Chamberland, and Debbie Leung. Flag fault-tolerant error correction for cyclic css codes. *arXiv preprint arXiv:1803.09758*, 2018.

[23] John Watrous. *Theory of Quantum Information*. Cambridge University Press, 2018.