Reduced Stabilizer Measurements for Flag-Based Syndrome Extraction Circuits for the [[5,1,3]] Code and the Steane Code

Master Thesis

Dhruv Bhatnagar

TU Delft

Reduced Stabilizer Measurements for Flag-Based Syndrome Extraction Circuits for the [[5,1,3]] Code and the Steane Code

Master Thesis

by

Dhruv Bhatnagar

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended on July 17th, 2023 at 09:00

Student number:	5119804
Thesis committee:	Dr Sebastian Feld (Supervisor)
	Prof. Barbara M. Terhal
	Dr Johannes Borregaard
Daily Supervisor:	Matthew Steinberg, MSc
Project Duration:	October, 2022 - June, 2023
Faculty:	Applied Sciences
Programme:	MSc Applied Physics
Track:	Physics for Quantum Devices and Quantum Computing

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Applied Sciences · Delft University of Technology

Acknowledgements

I would like to express sincere gratitude to my supervisor, Prof. Sebastian Feld, for his mature, methodical, and invaluable guidance. His kind mentorship and encouragement have been immensely helpful in my work. I would like to thank my daily supervisor, Matthew Steinberg, for his invaluable guidance, inspiration, and for the insightful introduction to the vital field of fault-tolerant quantum computing. I am deeply grateful to them for guiding me on an impactful and challenging problem, while giving me great autonomy. I am thankful to them for making the project an enjoyable and valuable learning experience, and for inspiring me with their passionate outlook on research.

I am sincerely grateful to Prof. David Elkouss and Prof. Carmen G. Almudéver for their valuable comments.

I am grateful to the committee members for dedicating their time to review my work.

Many thanks to the group members of the QML group, for all fascinating discussions, group meetings, and enjoyable interactions.

I thank my friends and colleagues from the Applied Physics programme, my residence, and the QCE division, for the immense support and motivation I received from them.

I am immensely grateful to my family for their unconditional support.

Dhruv Bhatnagar
5119804.

Abstract

Achieving universal and scalable quantum computing with reliably low error rates, despite the presence of unreliable circuit components, requires fault-tolerant quantum error correction. In general, quantum error correction imposes a significant overhead on the computation, motivating exploration of opportunities for optimization. Flag fault tolerance protocols have emerged as important schemes to realize fault tolerance experiments in the near term, because of their low qubit overhead, and absence of strict requirement for elaborate ancillary state preparation, relative to traditional schemes. However, the existing fast-reset, single-flag protocols for small codes generally employ a measurement of all stabilizer generators with unflagged circuits to distinguish a limited set of errors via the syndrome, leading to high circuit depth. In addition, the flagged measurement outcomes play a limited role in differentiating these errors. This motivates the possibility of reducing the circuit depth fault-tolerantly in flag-based syndrome extraction circuits. In this thesis, flag protocols with significantly reduced number of stabilizer measurements are constructed for the $[\![5,1,3]\!]$ code and the Steane code. The new protocols are divided into two classes. In the first class, the reduction is achieved by a dynamic choice of unflagged stabilizer measurements, based on past syndromes, and the utilization of the complete stabilizer group, to distinguish restricted sets of errors signalled by respective flagged measurements. In the second class, the reduction is achieved by measuring three high-weight flagged stabilizers, with the capability to detect a single input error, for the Steane code. The reduced stabilizer sequences are methodically constructed to yield unique and nontrivial syndromes for the relevant error set. This ensures that the fundamental condition of errors being detectable and distinguishable, which is the principal factor for the existing flag protocols to be fault-tolerant, is preserved. Pseudothresholds competitive with the existing flag protocols are established via Monte Carlo simulations under an error model consisting of two-qubit gate depolarizing errors, state preparation errors and measurement errors. Additionally, computer search programs are developed to obtain analogous reduced stabilizer sequences for both classes. These programs are also employed to assist in identifying certain mathematical properties of the high-weight Steane code stabilizers which can detect a single input error: namely, these stabilizers belong to different cosets of the X-stabilizer subgroup, and arise from 8-element subgroups within the stabilizer group. Furthermore, examples of such stabilizer sequences are constructed for few other codes. This thesis highlights the potential of employing parity measurements from the complete stabilizer group and extending beyond conventional adaptive measurements to improve the resource efficiency of fault-tolerant quantum error correction.

Contents

1 Introduction

Quantum computers, i.e. information processing devices based on the principles of quantum physics, are believed to offer speedup in solving certain computational tasks [1]. In the present day, quantum information processing devices are in the Noisy Intermediate-Scale Quantum (**NISQ**) era [2]. From the perspective of hardware, this means the following: Present-day qubits are fragile and are in constant interaction with their environment, which leads to uncontrolled evolution of the state over time. Therefore, qubits decohere, leading to errors in the quantum state. Additionally, quantum gates, i.e. the operations used to perform a desired evolution of qubit states, are imperfect and are also a source of noise. Furthermore, current devices are able to coherently control only a limited number of qubits (in the range 10-100)[3], so the number of qubits is a precious resource. Additionally, the number of gates which can be performed is limited, because every gate may add noise, and the time for which qubits can be coherently controlled is limited [2]. Besides, a significant number of quantum computing platforms only offer limited connectivity, so it may not be possible to perform two-qubit gates between every possible pair of qubits accurately [4][5].

In particular, **qubit decoherence** and **noise due to quantum gates** lead to loss of information being computed on. Both of these effects can cause the fidelity of a qubit's state with its initial state to decay exponentially with time [6][2].As a consequence, phase relations and entanglement between states, necessary for meaningful computation, may not remain preserved over time. This is markedly different from "classical" computers (i.e. those for which the computation is based on bits), for which contemporary error rates are in the range 10^{-17} [1], making the computation highly reliable. In comparison, current quantum devices have error rates in the range $10^{-3} - 10^{-2}$ [3]. A quantum advantage as powerful as an exponential speedup may be rendered unsuccessful if the result of the computation becomes progressively unreliable due to accumulation of errors, at a rate exponential in the depth of the circuit [2]. Therefore, to achieve the full potential of quantum computing, the problem of errors and reliability of the computation needs to be addressed [7].

There are multiple approaches which the research community has taken towards addressing errors in quantum computing. The hardware solution utilizes advances in nanofabrication, materials science and physics to develop better devices with lower error rates, and to improve hardwarw control techniques [8],[9]. Modern approaches also focus on harnessing the predicted phenomenon of Majorana bound states, expected to provide intrinsic protection from errors [10],[11]. In a different approach, error mitigation seeks to analyze several samples of output from a noisy quantum circuit to derive insights about the errors, and to compensate for or eliminate the effect of these errors on the result of the computation [7]. Distinct from these approaches, **Quantum Error Correction** (QEC) is a theory of mathematical techniques to actively keep track of, and correct, errors during the computation. However, QEC, being a circuit operation in itself, may introduce errors due to imperfect gates. The theory of **fault tolerance** goes beyond QEC to construct protocols to correct errors which may be introduced by the error correction circuits themselves. The existence of **threshold theorems** makes QEC and fault tolerance mathematically rigorous theories [12],[13]. These theorems prove that an arbitrarily low (encoded) error rate can be achieved for an arbitrarily long computation, as long as the error rate of any unprotected component is below a certain accuracy threshold value, with an overhead which does not scale inefficiently [1],[12]. While the significance of other approaches for addressing the problem of errors in quantum computation needs to be acknowledged, it is crucial to recognize that QEC is the only rigorous approach by which, in theory, scalable and universal

fault-tolerant quantum computation can be achieved in the long term [12],[2],[7]. Therefore, QEC and fault tolerance will be the main focus of this thesis.

Quantum error correction presents unique challenges compared to classical error correction. This arises from the fundamental impossibility of cloning arbitrary quantum states with arbitrary accuracy, the requirement of preserving superpositions of encoded states, and the continuous nature of quantum errors [1]. The Pauli **stabilizer formalism** is the most widely used framework to analyze and develop quantum error-correcting codes [14],[4]. In this framework, a quantum state is encoded into a subspace of a larger state space, defined via a commuting subgroup of Pauli operators called the stabilizer group, to protect information through redundancy. The subspace is called the codespace of a quantum error-correcting code (QECC). The generators of the stabilizer group are measured as parity checks to extract error syndromes, and decoded to identify and correct errors. Additionally, the choice of generators is not unique, and other operators from the complete stabilizer group may also be used as parity measurements.

The study of quantum error-correcting codes is divided into many classes, the major ones broadly being **block codes**[1] and topological codes[15]. Historically, among the block codes, the Shor code was the first code to be developed, which corrects single-qubit bit flips, phase flips, and combined bit and phase flips [16]. Another major development was the five qubit code, the smallest code which can correct arbitrary errors [17]. The Calderbank-Shor-Steane (**CSS**) construction, which allows constructing quantum codes from classical ones, and the smallest code of this type, the Steane code, were developed in [18],[19]. In the topological family, the toric code was a major development, inspired by spins on a lattice [20]. Since then, the field has seen many developments, leading to early experimental demonstrations of quantum error correction in the present era [3],[8],[21]. The focus of this thesis is limited to fault-tolerant syndrome extraction for small block codes, namely, the five qubit code and the Steane code, because their low qubit requirement make them favourably suitable as candidates for near-term experimental demonstrations.

Physical implementation of QEC is challenging because of extremely low error rate required, and the excessive **overhead** [2] in terms of qubits, gates, and ancillary state preparation, induced by traditional fault-tolerance techniques, developed by Shor[22], Steane[23] and Knill[24]. Additionally, an error-corrected quantum computation is envisioned to consist of multiple error correction cycles interleaved with algorithmic computation steps [1]. Due to the resulting overhead, it is advantageous to perform error correction with as little gate overhead as possible [2]. This motivates exploring methods to reduce this overhead by reducing the number of stabilizer measurements in fault tolerance schemes. Such a reduction has the potential to lead to savings in hardware resources and improving the efficiency of these protocols. Additionally, since gates themselves may introduce errors, a reduction in their number leads to a reduction in the number of locations where errors may occur. This may lead to an improvement in the accuracy threshold, relaxing the constraints on accuracy of components.

Due to the possibility of significant improvement in hardware efficiency, and the potential for relaxed accuracy requirements, this thesis focuses on developing techniques to reduce the number of stabilizer measurements in syndrome extraction circuits for a particular class of fault tolerance protocols called flag protocols [25]. **Flag fault tolerance protocols** are suitable for experimental implementation on near-term hardware [21],[26], because of they induce lower overhead in terms of number of qubits and ancillary state preparation circuits [25], compared to traditional schemes [22],[19],[24].

Flag techniques work by coupling an additional flag qubit to the ancillary qubits to detect and subsequently correct errors which can propagate to uncorrectable errors. In general, these protocols require fast measurement and qubit reset. In terms of their functioning, flag protocols involve repeating all stabilizer generator measurements with unflagged circuits to distinguish a restricted set of errors indicated by the respective flagged measurement. Furthermore, the measurement outcomes from flagged stabilizer measurements are only used to detect that an error has occurred, and are not directly used for differentiating these error, which is done via unflagged measurements. These observations on restricted error sets and limited role of flagged measurement outcomes on differentiating these errors imply that stabilizer sequences employed in flag protocols may not be optimal, and offer a possibility for improving the efficiency of syndrome extraction. Therefore, this thesis develops flag protocols with reduced number of stabilizer measurements in syndrome extraction circuits, as compared to existing flag protocols [25], while preserving detectability and distinguishability of errors, which is the principal reason for the existing flag protocols to be fault-tolerant [25]. The new protocols are constructed for the $[\![5,1,3]\!]$ code and the Steane code.

The **questions** addressed in this thesis are the following: To unambiguously distinguish between a

limited set of Pauli errors arising in flag protocols, is it necessary to measure all stabilizer generators with unflagged circuits? To achieve fault-tolerance, could it be advantageous to utilize operators from the complete stabilizer group, instead of the standard stabilizer generators? In addition, can a decision tree approach for distinguishing errors, where the choice of which stabilizer to be measured is based on past measurement outcomes, lead to shorter, fault-tolerant constructions? It will turn out that these questions can be answered positively, and it will be shown that these new approaches lead to flag protocols having lower depth overhead and competitive pseudothreshold under the specified error model, as compared to the existing protocols [25].

As a preview, the new protocols are divided into two classes. The **Split-and-Diagnose** protocols uniquely identify a detected error by constructing shorter sequences specifically designed to distinguish a given set of errors. Further, the stabilizers to be measured may be selected depending on measurement outcomes from previously measured stabilizers. The **Detect-and-Diagnose** protocol is based on utilizing combinations of high-weight stabilizers, capable of detecting a single input error on any data qubit, for flagged stabilizer measurements.

1.1. Outline

The remainder of this thesis is organized in the following sections:

- In **Sec. 2**, the mathematical preliminaries for this thesis are reviewed.
- In **Sec. 3**, the contextual background for the thesis is presented. The stabilizer formalism, the $[\![5,1,3]\!]$ code and the Steane code, on which the flag protocols in thesis are based [25], are presented. This is followed by the Gottesman-Knill theorem for efficient simulation of stabilizer codes, and the error propagation rules, which are used throughout this thesis. Fault-tolerant syndrome extraction for distance-3 codes is summarized, and the Knill's error model, used in this thesis, is presented. Traditional fault tolerance schemes are reviewed to highlight the relative advantages of flag protocols, with emphasis on the Shor's scheme [22] due to a fundamental reference [27] for the present work (see Sec. 5.1).
- In **Sec. 4**, the flag gadget, and the standard flag fault tolerance protocols for the $[\![5,1,3]\!]$ code and the Steane code, developed by Chao and Reichardt [25], are presented. These protocols serve as the fundamental basis for the construction of reduced stabilizer sequences in this thesis. Fault tolerance of these protocols is analyzed.
- In **Sec. 5**, the main contributions of this thesis, consisting of syndrome extraction circuits for flag protocols with reduced number of stabilizer measurements for the $[\![5,1,3]\!]$ code and the Steane code, are presented. The methodology developed is presented, following which the new constructions and theoretical analysis for fault tolerance are presented. In particular, one new protocol for the $[\![5,1,3]\!]$ code, and two new protocols for the Steane code, with significant reduction in gate overhead, and fault-tolerant error correction rules under the specified error model, are developed.
- In **Sec. 6**, numerical simulation results and analyses for the new flag protocols are presented. Pseudothresholds are established for the new protocols under an error model consisting of two-qubit gate depolarizing errors, preparation errors, and measurement errors [25],[28], via monte-carlo simulations.
- In **Sec. 7**, other instances of reduced stabilizer sequences, applicability to other codes, and observations on potential mathematical properties of such sequences are explored, primarily via computer-aided search.
- In **Sec. 8**, concluding remarks and future outlook are presented.

In addition, the thesis contains the following appendices:

- In **Appendix A**, definitions from classical error correction are presented as a reference for the CSS construction.
- In **Appendix B**, the complete stabilizer groups for the $[\![5,1,3]\!]$ code and the Steane code, referred throughout the thesis, are presented.
- In **Appendix C**, errors arising during flagged measurements in the protocols presented in Sec. 4 are tabulated.

- In **Appendix D**, syndromes obtained for the protocols with reduced stabilizer sequences, developed in Sec. 5, are tabulated.
- In **Appendix E**, look-up tables used for syndrome decoding for the protocols in Sec. 4 and Sec. 5 are presented.
- In **Appendices F and G**, reduced stabilizer sequences obtained via computer search, analogous in structure to the sequences developed in Sec. 5, are tabulated.

2

Definitions

This section reviews the terms and notation used in the thesis. We work in the Schrödinger picture of quantum mechanics, and it is assumed that the reader is familiar with notions of linear algebra and set theory. The main reference for this section is [1].

Quantum Computing

This work assumes familiarity with the concept of qubits as the elementary unit of quantum information, qubit states in finite-dimensional Hilbert spaces, and the Dirac notation. The standard notation is followed, for example, bra vectors in the dual space are denoted $\langle\psi| = (|\psi\rangle)^\dagger$, where $\dagger$ is the conjugate transpose, and the n-qubit Hilbert space is spanned by $\{|0\rangle, |1\rangle\}^{\otimes n}$. This work only considers pure states. Standard quantum gates, like the Hadamard, CNOT, CZ, will be heavily used, and may be looked up in standard references, like [1].

Definition 2.1 *The single-qubit **Pauli operators** are unitary operators $\{I, X, Y, Z\}$, with the following matrix representation:*

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.1}$$

*Depending on the context, the identity operator may be excluded from the Paulis. Pauli operators acting on n qubits are formed by taking n-fold tensor products of single-qubit Pauli operators. Multi-qubit Pauli operators are also called **Pauli strings**.*

For example, the operator $X \otimes I \otimes X \otimes Z \otimes Z$, denoted compactly as $XIXZZ$, is a Pauli operator acting on 5 qubits.

For quantum error correction (QEC), the Pauli operators can be viewed as (discrete) errors acting on a qubit. For example, for a state $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$, the Paulis affect it as:

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle, \quad Y(\alpha|0\rangle + \beta|1\rangle) = i(\alpha|1\rangle - \beta|0\rangle), \quad Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle. \tag{2.2}$$

Thus, X may be interpreted as a bit-flip error operator, Z as phase-flip error, and Y as a combined bit-and-phase-flip. Some useful properties of Pauli operators are reviewed below:

The Pauli operators square to identity. The single-qubit Pauli operators other than I have two eigenvalues, $\{+1, -1\}$, with the respective eigenvectors determined as

$$X|+\rangle = |+\rangle, \quad X|-\rangle = -|-\rangle; \quad Y|+i\rangle = |+i\rangle, \quad Y|-i\rangle = -|-i\rangle; \quad Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle; \tag{2.3}$$

with

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle), \quad |\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle). \tag{2.4}$$

An n-qubit Pauli operator has degenerate eigenvalues, with 2^{n-1} of them equal to $+1$ and 2^{n-1} of them being -1.

The single-qubit Pauli operators satisfy the following relations

$$XY = iZ, \quad YZ = iX, \quad ZX = iY \tag{2.5}$$

These give rise to the following commutation relations (where $[A, B] = AB - BA$):

$$[X, Y] = 2iZ, \quad [Y, Z] = 2iX, \quad [Z, X] = 2iY, \tag{2.6}$$

and the following anticommutation relations (where $\{A, B\} = AB + BA$):

$$\{X, Y\} = \{Y, Z\} = \{Z, X\} = 0, \quad \{X, X\} = \{Y, Y\} = \{Z, Z\} = 2I. \tag{2.7}$$

For example, $XY = -YX$, and hence, Pauli X is said to anticommute with Pauli Y. Pauli strings anticommute when they anticommute at an odd number of qubit positions, otherwise they commute (the identity commutes with every operator).

Definition 2.2 *The **support** of an n-qubit Pauli operator is the set of qubits on which it acts nontrivially, i.e. where the operator acting on the qubit differs from I.*

For example, the operator $XIXZZ$ has support on qubits 1, 3, 4 and 5, where the qubits are counted from the left, with indices starting from 1. This notation will be consistently used throughout this thesis.

Another convention used in this thesis is that the leftmost qubit corresponds to the top qubit in a circuit, while the rightmost qubit appears at the bottom.

Definition 2.3 *The **weight** of an n-qubit Pauli operator is the number of qubits on which it has support.*

For example, the operator $XIXZZ$ has weight equal to 4.

Group Theory

The notions in this part form the background for stabilizer codes and the CSS construction (see Sec. 3.1). The main references are [1] and [29].

Definition 2.4 *A **group** is a set $\mathcal{G}$, along with a binary operation or a composition rule, $\circ$, denoted $\{\mathcal{G}, \circ\}$, which obey the following axioms:*

1. *Closure: $\forall u, v \in \mathcal{G}, u \circ v \in G$.*
2. *Associativity: $\forall u, v, w \in \mathcal{G}, (u \circ v) \circ w = u \circ (v \circ w) = u \circ v \circ w$.*
3. *Existence of an identity element: $\forall u \in \mathcal{G}, \exists e \in \mathcal{G}$ such that $e \circ u = u$ and $u \circ e = u$.*
4. *Existence of inverse: $\forall u \in \mathcal{G}, \exists v \in \mathcal{G}$ such that $u \circ v = e$ and $v \circ u = e$. The inverse of $u \in \mathcal{G}$ is denoted as u^{-1}.*

The group can also be referred to by the symbol for the underlying set, for example $\mathcal{G}$ instead of $\{\mathcal{G}, \circ\}$. The composition symbol can also be suppressed to write $u \circ v$ as uv, for $u, v \in \mathcal{G}$.

Definition 2.5 *A **subgroup** of a group $\{\mathcal{G}, \circ\}$ is a subset $\mathcal{H}$ of $\mathcal{G}$, such that $\{\mathcal{H}, \circ\}$ satisfies the axioms of a group.*

Definition 2.6 *An **abelian group** or a **commutative group** is a group $\{\mathcal{G}, \circ\}$ such that the resultant of a composition does not depend on the order of composing the elements. That is, $\forall u, v \in \mathcal{G}, u \circ v = v \circ u$.*

Definition 2.7 *Let $\mathcal{G}$ be a group, and $\mathcal{H}$ be a subgroup of $\mathcal{G}$, under the operation $\circ$. Consider the following construction, where a new set is formed by composing an element of $\mathcal{G}$ with the entirety of $\mathcal{H}$ from the left, like so:*

$$u\mathcal{H} = \{u \circ h : h \in \mathcal{H}\}, \text{ for } u \in \mathcal{G}. \tag{2.8}$$

*Subsets of this form are the **left cosets** of $\mathcal{H}$ in $\mathcal{G}$. Analogously, the **right cosets** of $\mathcal{H}$ in $\mathcal{G}$ are defined as*

$$\mathcal{H}u = \{h \circ u : h \in \mathcal{H}\}, \text{ for } u \in \mathcal{G}. \tag{2.9}$$

Left cosets of different elements in $\mathcal{G}$ are either the same set or mutually disjoint. Similarly, this holds for the right cosets.

Theorem 2.0.1 ***Lagrange's theorem****: The number of elements in a subgroup $\mathcal{H}$ divides the number of elements in the group $\mathcal{G}$, with the ratio being equal to the number of cosets of $\mathcal{H}$ in $\mathcal{G}$.*

This is used to determine the number of codewords of a CSS code (see Section 3.2).

Definition 2.8 *Let S be a subset of $\mathcal{G}$. The **centralizer**, $\mathcal{C}_{\mathcal{G}}(S)$, of S in $\mathcal{G}$ is a subgroup of $\mathcal{G}$ containing all elements of $\mathcal{G}$ which commute with every element of S. Formally, this may be written as $\mathcal{C}_{\mathcal{G}}(S) = \{u \in \mathcal{G} | \forall s \in S, us = su\}$.*

Definition 2.9 *Let S be a subset of $\mathcal{G}$. The **normalizer**, $\mathcal{N}_{\mathcal{G}}(S)$, of S in $\mathcal{G}$ is a subgroup of $\mathcal{G}$, such that any element of S, conjugated by any element of the normalizer, gives an element of S. Formally, $\mathcal{N}_{\mathcal{G}}(S) = \{u \in \mathcal{G} | \forall s \in S, usu^{-1} = s',$ for $s' \in S\}$.*

Definition 2.10 *A subgroup $\mathcal{N}$ of $\mathcal{G}$ is a **normal subgroup** if $unu^{-1} \in \mathcal{N}, \forall u \in \mathcal{G}, \forall n \in \mathcal{N}$. That is, $\mathcal{N}$ is invariant under conjugation by elements of $\mathcal{G}$.*

Definition 2.11 *Given a normal subgroup $\mathcal{N}$ of group $\mathcal{G}$, define the set $\mathcal{G}/\mathcal{N} = \{u \circ \mathcal{N} : u \in \mathcal{G}\}$, i.e. the set of left cosets of $\mathcal{N}$ in $\mathcal{G}$. Define a composition rule on the elements of $\mathcal{G}/\mathcal{N}$ using the composition rule of $\mathcal{G}$ as $(u \circ \mathcal{N}) \circ_{\mathcal{G}/\mathcal{N}} (v \circ \mathcal{N}) := (u \circ v) \circ \mathcal{N}$. Then, $\{\mathcal{G}/\mathcal{N}, \circ_{\mathcal{G}/\mathcal{N}}\}$ obeys the group axioms and forms a group, known as the **Quotient group**.*

Definition 2.12 *Given a group $\mathcal{G}$, and a set S, $\mathcal{G}$ is said to act on S (from the left) if there exists a map $\phi : \mathcal{G} \times S \to S$, such that, $\forall s \in S$, the following axioms are satisfied:*

1. *Identity: $\phi(e, s) = s$, where e is the identity element in $\mathcal{G}$*
2. *Compatibility: $\phi(u, \phi(v, s)) = \phi(u \circ v, s), \forall u, v \in \mathcal{G}$.*

*The map ϕ defines the **group action** of $\mathcal{G}$ on S.*

Definition 2.13 *The **orbit** of an element $s \in S$ under a group action $\phi : \mathcal{G} \times S \to S$ is the set of all elements in S to which s can be moved by the group action. That is, the orbit of s is $\{\phi(u, s) : u \in \mathcal{G}\}$.*

Definition 2.14 *Given a group action ϕ on group $\mathcal{G}$ and set S. $s \in S$ is a **fixed point** of some $u \in \mathcal{G}$ if $\phi(u, s) = s$, i.e. s is not moved by u. For every $s \in S$, the **stabilizer**, $stab_{\mathcal{G}}(s)$ of s is the set of all elements in $\mathcal{G}$ which fix s, i.e.*

$$stab_{\mathcal{G}}(s) = \{u \in \mathcal{G} : \phi(u, s) = s\}. \tag{2.10}$$

The stabilizer is a subgroup of $\mathcal{G}$ [1].

As mentioned previously, the main references for this subsection are [1] and [29].

Classical Error-Correcting Codes

Elementary notions from classical error correction [1] are useful as a reference for the CSS construction (see Sec. 3.2), and are reviewed in Sec. A, to keep this section brief.

3

Background: Quantum Error Correction And Fault Tolerance

This background section briefly review of quantum error correction using stabilizer codes in Sec. 3.1, followed by the CSS construction, which is used to construct the Steane code, in Sec. 3.2. Following these, the codes which are the main focus of this thesis, the $[\![5,1,3]\!]$ code and the Steane code, are described in Sec. 3.3. This is followed by description of efficient simulation of stabilizer codes via the Gottesman-Knill theorem in Sec. 3.4. Following this, the error propagation rules used throughout this thesis are presented in Sec. 3.5. Sec. 3.6 to Sec. 3.11 form the second part of this section, and review the notions of fault tolerance. In particular, the Knill's error model, used for pseudothreshold simulations (see Sec. 6), is presented in Sec. 3.9.

This section is primarily intended for establishing the contextual framework for the upcoming sections, and for completeness. The experienced reader may prefer to focus on error propagation rules (Sec. 3.5), refresh the terminology on fault tolerance (Sec. 3.8 and Sec. 3.10.3), and the Knill's error model (Sec. 3.9).

The references for this section are [14],[1],[4],[30],[31],[13],[12].

3.1. Stabilizer Codes

In quantum error correction, a state of one or more qubits is mapped to an entangled state in a larger Hilbert space. Consequently, distributing the information into a state of a larger number of qubits provides redundancy to protect the information against errors.

As introduced in Sec. 1, a QECC is a k-dimensional subspace of an n-dimensional Hilbert space, $k < n$. Mutually orthogonal vectors which span the codespace form a basis, and a codeword is a superposition of these basis vectors. In this thesis, we'll work with Pauli errors, defined in Sec. 2. The **distance** of a quantum error-correcting code is the minimum weight of a Pauli operator which transforms one logical codeword into another. Such a quantum error-correcting code is denoted as a $[\![n,k,d]\!]$ code. The code can correct up to $t = \lceil \frac{d-1}{2} \rceil$ errors. The unencoded n qubits are called **physical qubits**, and the encoded k qubits are called **logical qubits** [14],[1].

For example, the 3-qubit quantum repetition code to protect against quantum bit-flip (Pauli X) errors is the map

$$|0\rangle \mapsto |000\rangle\,, \quad |1\rangle \mapsto |111\rangle\,. \tag{3.1}$$

Fig. 3.1 illustrates the elements of quantum error correction, namely, encoding, errors, stabilizer measurements, syndrome decoding, and correction, for the 3-qubit repetition code for bit-flip (Pauli X) errors [1].

Ths stabilizer formalism is the most widely used framework to construct quantum error correcting codes[14]. Therefore, some background on the Pauli operators is presented, which form the basis for the stabilizer formalism. This is followed by description of the aforementioned elements of error correction in the context of stabilizer codes.

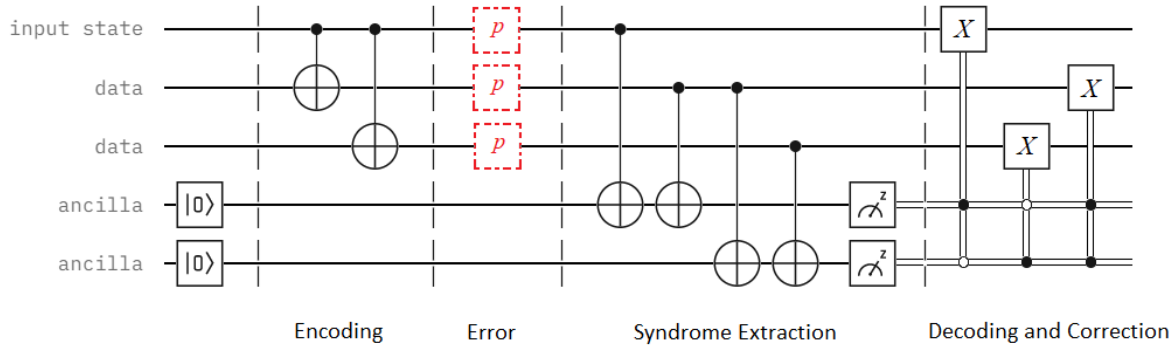

Figure 3.1: Encoding, stabilizer measurement, decoding and correction for the 3-qubit repetition code for bit flip errors.

3.1.1. The Pauli Group and the Binary Symplectic Vector Representation

The 1-qubit Pauli operators, with operator composition $\circ$, form the 1-qubit Pauli group, $(\mathcal{G}_1, \circ)$, or simply $\mathcal{G}_1$, consisting of the elements:

$$\mathcal{G}_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \tag{3.2}$$

The composition rule is matrix multiplication (see Eq. 2.5). The phase factors are included for group closure. The n-qubit Pauli group $(\mathcal{G}_n, \circ)$, or simply $\mathcal{G}_n$, consists of all n-qubit Pauli strings, along with the phase factors $\{\pm 1, \pm i\}$:

$$\mathcal{G}_n = \{pP_1 \otimes P_2 \otimes \cdots \otimes P_n | P_j \in \{I, X, Y, Z\}, j = 1, 2, \cdots, n, p \in \{\pm 1, \pm i\}\}. \tag{3.3}$$

The group composition rule derived from G_1 is:

$$(pP_1P_2 \cdots P_n) \circ (qQ_1Q_2 \cdots Q_n) = pq(P_1 \circ Q_1)(P_2 \circ Q_2) \cdots (P_n \circ Q_n), \text{ for } pP_1P_2 \cdots P_n, qQ_1Q_2 \cdots Q_n \in \mathcal{G}_n. \tag{3.4}$$

The tensor product notation has been suppressed. In the context of error correction, it is useful to consider the Pauli group with elements without the phase factors. Under the equivalence relation that two elements of $\mathcal{G}_n$ are equivalent if they only differ in an overall phase, the said group is then the quotient group under this equivalence. Modulo the phase factors, the Paulis commute, forming the commutative Pauli group.

The Binary Symplectic Vector Representation

Elements of the commuting n-qubit Pauli group can be represented as $2n$-dimensional binary vectors with addition modulo 2 [1],[14].

The vector is constructed as follows: if an X operator is present on the j^{th} qubit in an n-qubit Pauli string, the j^{th} entry in the vector is 1, and the $(n+j)^{th}$ entry is 0. If a Z is present on the j^{th} qubit, the j^{th} entry is 0, and the $(n+j)^{th}$ entry is 1. For a Y, both the j^{th} and $(n+j)^{th}$ element of the vector are 1, and I is represented by both these elements being 0. Thus, the vector has separate X and Z components, and is of the form

$$(x|z) = (x_1, x_2, \ldots, x_n | z_1, z_2, \ldots, z_n). \tag{3.5}$$

For example, $XZYIX$ is represented as $(1, 0, 1, 0, 1|0, 1, 1, 0, 0)$. Multiplication of Pauli strings corresponds to addition of respective binary vector representations modulo 2. Commutation relations among Pauli operators P and Q, having vector representations $v = (a|b)$ and $w = (c|d)$, respectively, can be determined as

$$PQ = (-1)^{ad-bc} QP. \tag{3.6}$$

More succinctly, $ad - bc = v\Lambda w^T$ is the symplectic product between v and w, with $\Lambda = \begin{bmatrix} I_n & 0 \\ 0 & I_n \end{bmatrix}$.

3.1.2. Stabilizers and Stabilizer Generators

The stabilizer formalism[14] provides a foundation to construct quantum error-correcting codes, based on the Pauli group, $\mathcal{G}_n$. The resulting codes are called stabilizer codes. A subgroup $\mathcal{S}$ of $\mathcal{G}_n$, called the stabilizer group of the code, is constructed according to the following conditions:

1. $\mathcal{S}$ is an Abelian subgroup, i.e. all elements of $\mathcal{S}$ commute.
2. $\mathcal{S}$ does not contain the element $-I$.

Then, the codespace of the stabilizer code is the Hilbert space formed by the intersection of eigenspaces corresponding to the $+1$ eigenvalue of all elements of $\mathcal{S}$, that is, the simultaneous $+1$-eigenspace of stabilizers.

The two conditions have the following significance. The codespace is a joint eigenspace of operators, and for the operators to be diagonalizable in the same basis, they need to commute. Also, it can be seen that there is no non-trivial state which is a $+1$ eigenstate of $-I$, because the solution of

$$-I\left|\psi\right\rangle = \left|\psi\right\rangle \tag{3.7}$$

is the null vector, which does not correspond to a physical state. Therefore, constructing a codespace with $-I$ as a possible element of $\mathcal{S}$ leads to a trivial codespace, hence it is excluded. Operators which give $-I$ under composition are also excluded.

Stabilizer Generators

A generating set of the stabilizer group is the smallest subset of independent stabilizers, such that, upon removing even one element from the generating set, one cannot construct the entire stabilizer group via element composition. The elements of the generating set are called stabilizer generators, or, generators. For a stabilizer subgroup of the Pauli group having 2^r elements, there are r generators, denoted $\{g_i\}, 1 \leq i \leq r$. Formally,

$$\mathcal{S} = \langle g_1, g_2, \ldots, g_r \rangle, \tag{3.8}$$

and, an arbitrary stabilizer $s_j \in \mathcal{S}$ can be expressed in terms of the generators as

$$s_j = g_1^{j_1} g_2^{j_2} \ldots g_r^{j_r}, \tag{3.9}$$

where each $j_k \in \{0, 1\}, 1 \leq k \leq r$. States within the codespace satisfy $s\left|\psi\right\rangle = \left|\psi\right\rangle, \forall s \in \mathcal{S}$.

The generating set is not unique, and other stabilizers may be chosen as generators.

Check Matrix of a Stabilizer Code

A stabilizer code can be specified by a binary check matrix of size $(n-k) \times 2n$, whose rows are formed by the binary vector representations of stabilizer generators:

$$H = \left[\begin{array}{c|c} H_1 & H_2 \end{array}\right], \tag{3.10}$$

where H_1 and H_2 are $(n-k) \times n$ binary matrices representing the X and Z components of generators. This representation is used in deriving the encoding circuit for a stabilizer code (see Sec. 3.1.4).

3.1.3. The Significance of Pauli Errors

Although general quantum errors can be continuous, it is sufficient to work with discrete Pauli errors, which form a basis in which a general error can be expressed, [1]. Upon a stabilizer measurement, a general error can be discretized to get a state on which either no error or a nontrivial Pauli error has occurred, and is correctable if the Pauli errors can be corrected, via the Knill-Laflamme theorem [1]. Stabilizer codes can be analyzed efficiently under Pauli errors via the Gottesman-Knill theorem (see Sec. 3.4).

The following sections describe how to perform error correction with stabilizer codes. State vectors will be assumed to be normalized, or, depending on the context, normalization factors will be ignored.

3.1.4. Encoding to the Codespace

An encoding map, realized with an encoding circuit, encodes the state to be protected from errors in the higher-dimensional Hilbert space.

Encoding Circuit for a Stabilizer Code
The encoding circuits for the $[\![5,1,3]\!]$ code and the Steane code are used in the wavefunction simulations in this thesis, described in Sec. 6. Therefore, their derivation is summarized.

The encoding circuit for a stabilizer code can be constructed from its check matrix and binary vectors representing logical operators expressed in standard form [14],[32],[1].

For $k = 1$, the encoding maps $|0\rangle$ and $|1\rangle$ to orthogonal states in the codespace as

$$|0\rangle \mapsto \Big(\sum_{s\in S} s\Big)|0\rangle^{\otimes n}, \quad |1\rangle \mapsto \Big(\sum_{s\in S} s\Big)\bar{X}\,|0\rangle^{\otimes n}, \tag{3.11}$$

where $\bar{X}$ represents the logical X operator. To implement the circuit corresponding to sum of stabilizers, note that

$$\sum_{s\in S} s = (I+g_1)(I+g_2)\dots(I+g_{n-k}), \tag{3.12}$$

where $g_1, g_2, \dots, g_{n-k}$ are the stabilizer generators, and this composition of operators projects onto the simultaneous $+1$ eigenspace of the generators [14],[32].

To implement the circuit corresponding to each $(I+g_j)$, the row corresponding to g_j in the standard form of the check matrix is considered. First, a hadamard gate is applied to the qubit j. If the same qubit has a 1 in the Z-component, Z gate is applied to it. For other qubits which have a 1 in the Z or X components, a controlled-Z or controlled-X gate is applied respectively, with qubit j as control. There is no gate applied corresponding to 1 in the X-component of qubit j.

To implement $\bar{X}$ unitarily, depending on whether the 1-qubit state is a $|0\rangle$ or $|1\rangle$, a qubit on which $\bar{X}$, in standard form, has support is chosen as the input qubit, and the remaining qubits are initialized in the $|0\rangle$ state. Controlled X or Z gates are applied from the input qubit to other qubits corresponding to the entries in the standard form of the binary vector representing $\bar{X}$.

The encoding circuits for the $[\![5,1,3]\!]$ code and the Steane code are presented in Sec. 3.3.

3.1.5. Stabilizer Measurement and The Syndrome

The syndrome is a string of classical bits, extracted from the encoded state by measuring stabilizer generators, to diagnose a possible error.

The syndrome signifies the eigenvalue of the stabilizer with respect to the state. By definition, the space of error-free codewords is the simultaneous $+1$ eigenspace of the stabilizers [1]. Further, each stabilizer equally divides the n-qubit Hilbert space into its $+1$ and -1 eigenspaces.

If a detectable error occurs, the state gets modified and moves to the -1 eigenspace of one or more stabilizers.

The eigenvalues of all generators forms the required parity information, i.e. the syndrome, to diagnose the error. It is sufficient to measure all stabilizer generators, because they are a minimal set which can represent all stabilizers [14].

Indirect Measurement
Stabilizer measurement on the data qubits is performed indirectly through an extra ancillary qubit, to preserve a superposition of codewords. The circuit in Fig. 3.2 implements an indirect measurement of unitary U with eigenvalues $\{-1,+1\}$ [1].

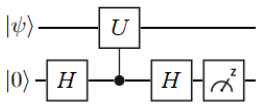

Figure 3.2: Indirect measurement of U

The circuit transforms an input state $|0\rangle |\psi\rangle$, as shown in Fig. 3.2, as:

$$|0\rangle |\psi\rangle \rightarrow \frac{1}{2}(|0\rangle ((I+U)|\psi\rangle) + |1\rangle ((I-U)|\psi\rangle)). \tag{3.13}$$

It may be verified that the circuit works as follows [1]. $(I+U)$ projects the state onto the $+1$ eigenspace of U, which occurs when the ancilla is $|0\rangle$ after measurement, and $(I-U)$ projects it onto the -1 eigenspace of U, when the ancilla is $|1\rangle$ after the measurement. Since the encoding qubits (data qubits) are projected to an eigenspace of a stabilizer generator, the data qubits and the ancilla are not entangled. Therefore, the extracted syndrome depends on the error and not on the underlying state.

For brevity, the mapping $+1 \mapsto 0, -1 \mapsto 1$ from eigenvalues to bits will be used to express the syndrome, so, for example, the syndrome $\{+1, -1\}$ will be mapped to 01.

Syndrome and Commutation Relations
The syndrome for a given Pauli error may be determined by calculating the commutation relation between the error and stabilizer generators. In Eq. 3.13, consider the state $|\psi\rangle$ in the codespace. Since the Pauli operators either commute or anticommute, a Pauli error $\mathcal{E}$, which commutes with a generator g_1, gives the $+1$ eigenvalue (syndrome bit 0) on measurement , because

$$\frac{I+g_1}{2}\mathcal{E}|\psi\rangle = \mathcal{E}\frac{I+g_1}{2}|\psi\rangle = \mathcal{E}|\psi\rangle, \quad \frac{I-g_1}{2}\mathcal{E}|\psi\rangle = \mathcal{E}\frac{I-g_1}{2}|\psi\rangle = 0, \tag{3.14}$$

while if $\mathcal{E}$ anticommutes with a generator g_2, the measurement gives the -1 eigenvalue (syndrome bit 1):

$$\frac{I+g_2}{2}\mathcal{E}|\psi\rangle = \mathcal{E}\frac{I-g_2}{2}|\psi\rangle = 0, \quad \frac{I-g_2}{2}\mathcal{E}|\psi\rangle = \mathcal{E}\frac{I+g_2}{2}|\psi\rangle = \mathcal{E}|\psi\rangle. \tag{3.15}$$

3.1.6. Decoding

Although decoding may refer to the inverse of encoding, in this thesis, decoding the syndrome refers to determining the codeword closest to a given state, given the error syndrome. Once a syndrome has been reliably obtained, it is decoded to infer the most likely Pauli error which could have resulted in that syndrome [14]. The 'most likely' error is the error which can occur with highest probability corresponding to the given syndrome.

In general, there could be multiple errors which lead to the same syndrome, which may occur with a lower probability than the most likely error [13]. Decoding assuming the most likely error may corrupt the state further and lead to a logical error, if the cause of the syndrome was an inequivalent, lower probability error.

The set of inequivalent errors the code is expected to correct must lead to distinct syndromes, i.e., they need to be distinguishable, so that they can be unambiguously decoded. A look-up table (LUT) which maps the syndrome to the most likely error suffices for decoding for the codes in this thesis.

3.1.7. Correction

Once the syndrome has been decoded, the most likely error (or the minimum weight equivalent up to stabilizer multiplication) is applied as the correction. If the decoding accurately identifies the error (up to stabilizer multiplication), this correction restores the state to the one before the error. However, if the decoding fails to infer the error correctly, the correction may lead to a logical error.

In practice, a physical correction via gates need not be applied [33].Instead, the cumulative error can be tracked by updating a Pauli frame [33]. This means keeping track of the subspace of the n qubit space, via the syndrome, to which the error has mapped the encoded state. The Pauli operator which restores the state to codespace at the end of the computation can be classically accounted for. Therefore, the corrections can be taken to be noiseless.

3.1.8. Operations on Encoded Qubits

In addition to quantum error correction cycles to preserve a state, to achieve a reliable quantum computation, logical operations, belonging to the centralizer of the stabilizer (Sec. 2), are performed while the quantum state is encoded in the QECC. The logical operators obey the commutation relations between the respective unencoded operations. At the end of the desired computation, a logical measurement on the encoded qubits extracts the result. These elements of the computation, however, are beyond the scope of this thesis.

3.2. The CSS Code Construction

Stabilizer codes, and in particular, the Steane code, can be constructed using two classical codes (see Appendix A) through the CSS construction.

Let C_1 be a $[n, k_1, d_1]$ (classical) binary linear code, and C_2 be a $[n, k_2, d_2]$ (classical) binary linear code, with parity check matrices H_{C_1} and H_{C_2} respectively, and with $C_2 \subseteq C_1$, i.e., C_2 is a subspace (or subcode) of C_1. Further, let $C_2^\perp$ be the dual of C_2, with parity check matrix $H_{C_2^\perp}$, and distance $d_2^\perp$. C_1 and $C_2^\perp$ correct up to t_1 and $t_2^\perp$ errors respectively, where $t_1 = \lceil \frac{d_1-1}{2} \rceil, t_2^\perp = \lceil \frac{d_2^\perp-1}{2} \rceil$. Then, the CSS construction defines a $[\![n, k_1 - k_2, d]\!]$ quantum code, $CSS(C_1, C_2)$, with $d \geq min(d_1, d_2^\perp)$ [18],[19].

Basis states for the codespace of $CSS(C_1, C_2)$ are constructed from the classical codewords of C_1 and C_2, which are binary vectors in $\mathbb{F}_2^n$. An equivalence relation is defined on C_1: two elements in C_1 are equivalent if they differ (modulo 2) by an element of C_2, i.e., for $x_i, x_j \in C_1, x_i \sim x_j$ if $x_i = x_j + v,$ for some $v \in C_2$. This equivalence relation partitions C_1 into equivalence classes, each of which is used to form a basis state of the quantum code. This may be formulated in group-theoretic terms: $\mathbb{F}_2^n$ forms a group under addition modulo 2, denoted by $+$. Then, C_2 is a subgroup of C_1, and the equivalence classes are cosets of C_2 in C_1.

For $x \in C_1$, the coset of C_2 with respect to x is denoted as $x + C_2 = \{x + v | v \in C_2\}$. The basis codeword for the quantum code, $|\bar{x}\rangle$, labelled by the coset representative x, is defined as a uniform superposition of ket vectors labelled by elements of the coset to which x belongs:

$$|\bar{x}\rangle = |x + C_2\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{v \in C_2} |x + v\rangle \,. \tag{3.16}$$

States corresponding to different cosets are orthogonal. This is because, for x_i, x_j in different cosets, the inner product $\langle x_i + C_2 | x_j + C_2 \rangle$ does not contain any terms which have a non-zero overlap, and is hence equal to zero.

The number of such orthogonal states is equal to the number of cosets, which is $2^{k_1 - k_2}$, by Lagrange's theorem, presented in Sec. 2. These form a basis for the $CSS(C_1, C_2)$ codespace [4],[1]. The check matrix of $CSS(C_1, C_2)$ is constructed from the parity check matrices of the classical codes as:

$$H = \left[\begin{array}{c|c} H_{C_2^\perp} & 0 \\ 0 & H_{C_1} \end{array}\right] \tag{3.17}$$

Therefore, the standard stabilizer generators for CSS codes contain either only X or only Z operators.

3.2.1. Correcting quantum errors using CSS codes

In CSS codes, bit-flip (Pauli X) errors and phase-flip (Pauli Z) errors can be decoded and corrected separately [1]. If a bit-flip error occurs, it affects the CSS code's codword analagous to a classical bit-flip on codewords of C_1. Hence, a bit-flip error on a CSS code can be corrected by measuring Z-type stabilizers constructed from parity checks of C_1. To correct phase-flip errors with a CSS code, applying the Hadamard transform on the codeword, which entails applying the Hadamard gate on all data qubits, transforms phase-flip errors into bit-flip errors on codewords of the dual code $C_2^\perp$. Hence, a phase-flip error can be corrected by measuring X-type stabilizers constructed from parity checks of $C_2^\perp$. The Hadamard transform is applied again to restore the original state. A Pauli Y error is corrected as a bit-flip followed by a phase-flip. Additionally, the separate decoding can also correct bit-flips and phase-flips occurring on different qubits. The details of these standard results may be found in the references [1], [31], and [4].

3.3. The $[\![5, 1, 3]\!]$ Code and the Steane Code

The $[\![\mathbf{5}, \mathbf{1}, \mathbf{3}]\!]$ **code**, or the five qubit code, is the smallest possible quantum error-correcting code which can correct an arbitrary error on a single qubit [17]. It is a perfect code, because every eigenspace of the stabilizers in the 5-qubit Hilbert space other than the codespace corresponds to a different single qubit error [1]. The stabilizer group of the $[\![5, 1, 3]\!]$ code is generated by:

$$\langle XZZXI, IXZZX, XIXZZ, ZXIXZ \rangle \tag{3.18}$$

and can be obtained from one of these generators by cyclic permutation the qubits. This symmetry is also reflected in the basis states for the codespace [1].

The check matrix with the generators from Eq. 3.18 is

$$H = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]. \tag{3.19}$$

The operators $XXXXX$ and $ZZZZZ$ serve as the logical operators $\bar{X}$ and $\bar{Z}$, respectively.

The circuit to extract syndromes corresponding to the stabilizer generators on different ancilla qubits, equivalent to the one derived from Fig. 3.2, is depicted in Fig. 3.3.

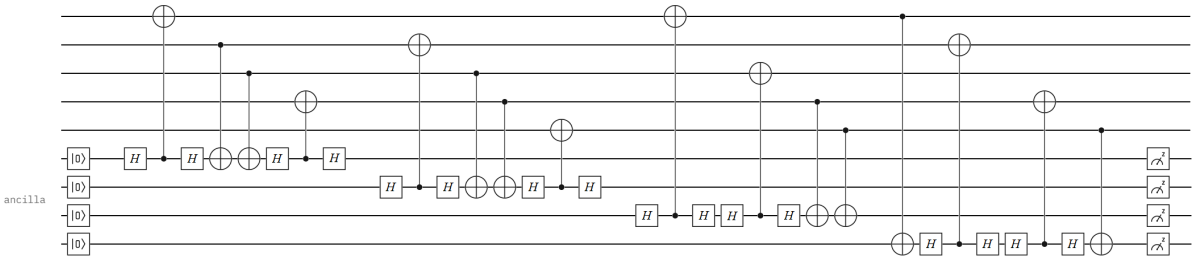

Figure 3.3: Circuit for measuring stabilizer generators of the $[\![5,1,3]\!]$ code.

Using row operations, the check matrix in Eq. 3.19 can be brought into the standard form:

$$H = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}\right]. \tag{3.20}$$

Using the procedure described in Sec. 3.1.4, the circuit to encode a single qubit state in the codespace of the $[\![5,1,3]\!]$ code, after some simplifications, is shown in Fig. 3.4.

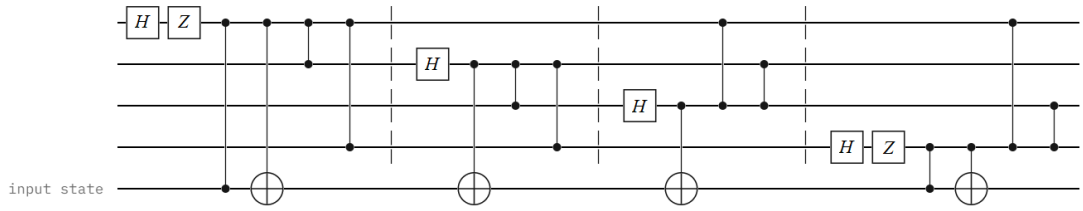

Figure 3.4: Encoding circuit for the $[\![5,1,3]\!]$ code. The fifth qubit is prepared in the qubit state to be encoded.

The look-up table, which maps syndromes, obtained by measuring the generators presented in eq. 3.18, to weight-1 input errors is presented in Table 3.1.

The **Steane code** is the smallest distance-3 CSS quantum error-correcting code (for preliminaries on classical error correcting codes with reference to CSS codes, see Appendix A). It is constructed, with reference to the CSS construction, by taking C_1 as the $[7,4,3]$ Hamming code, presented in Eq. A.1, so $k_1 = 4$, and $C_2 = C_1^{\perp}$ [19],[1]. $C_1^{\perp}$ is a $k_2 = 3$-dimensional subspace of C_1, hence, it satisfies the conditions of the CSS construction. The resulting quantum code has distance-3 and parameters

Syndrome	Most likely error	Syndrome	Most likely error
0000	No error	1000	$IXIII$
0001	$XIIII$	1001	$IIIZI$
0010	$IIZII$	1010	$ZIIII$
0011	$IIIIX$	1011	$YIIII$
0100	$IIIIZ$	1100	$IIXII$
0101	$IZIII$	1101	$IYIII$
0110	$IIIXI$	1110	$IIYII$
0111	$IIIIY$	1111	$IIIYI$

Table 3.1: Look-up table for decoding the syndrome for the $[\![5,1,3]\!]$ code.

$[\![n, k_1 - k_2, d]\!] = [\![7,1,3]\!]$. It is also the smallest code of the quantum Hamming code family [1]. Since $C_2^\perp = (C_1^\perp)^\perp = C_1$, the check matrix of the Steane code is

$$H = \left[\begin{array}{c|c} H_{C_2^\perp} & 0 \\ 0 & H_{C_1} \end{array}\right] = \left[\begin{array}{ccccccc|ccccccc} 0&0&0&1&1&1&1&0&0&0&0&0&0&0 \\ 0&1&1&0&0&1&1&0&0&0&0&0&0&0 \\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&0&0&1&1 \\ 0&0&0&0&0&0&0&1&0&1&0&1&0&1 \end{array}\right]. \tag{3.21}$$

The stabilizer group is generated by

$$\langle IIIXXXX, IXXIIXX, XIXIXIX, IIIZZZZ, IZZIIZZ, ZIZIZIZ\rangle. \tag{3.22}$$

The codewords of $C_1^\perp$ are $\{0000000, 1010101, 0110011, 1100110, 0001111, 1011010, 0111100, 1101001\}$. The cosets of $C_1^\perp$ in C_1 are used to form the basis of the codespace [1]. Due to separate decoding of X and Z errors, the Steane code can correct two single-qubit errors which are composed of a single X and a single Z occuring on different data qubits. The operators $XXXXXXX$ and $ZZZZZZZ$ serve as the logical operators $\bar{X}$ and $\bar{Z}$, respectively. The circuit to measure the stabilizer generators in parallel using 6 ancilla qubits can be derived from the general circuit in Section 3.1.5 and analogous to the circuit for the $[\![5,1,3]\!]$ code, presented in Sec. 3.3.

The check matrix in Eq. (3.21) can be brought into the following standard form (using qubit permutation) [1]:

$$H = \left[\begin{array}{ccccccc|ccccccc} 1&0&0&0&1&1&1&0&0&0&0&0&0&0 \\ 0&1&0&1&0&1&1&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&1&1&0&0&1 \\ 0&0&0&0&0&0&0&0&1&1&0&1&0&1 \\ 0&0&0&0&0&0&0&1&1&1&0&0&1&0 \end{array}\right]. \tag{3.23}$$

The encoding circuit for the Steane code derived from this matrix, and after reversing the qubit permutation and some simplifications, is shown in Fig. 3.5. The look-up table for Z-type stabilizer measurements (in the order $IIIZZZZ$,$IZZIIZZ$,$ZIZIZIZ$) is shown in Table 3.2. It can be derived analogously for X-type stabilizers (see Tables E.4 and E.3).

3.4. The Gottesman-Knill Theorem

The Gottesman-Knill theorem [34] is an important result in the stabilizer formalism, which allows efficient classical simulation of stabilizer codes under Pauli errors. This method of simulation, along with the binary symplectic vector representation of Pauli operators, is the basis for the stabilizer formalism simulations

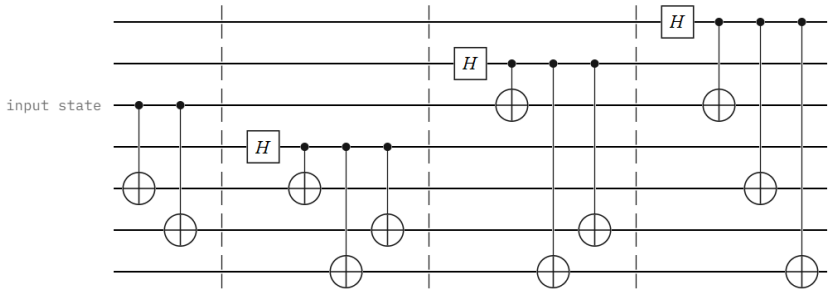

Figure 3.5: Encoding circuit for the Steane code. The third qubit is prepared in the qubit state to be encoded.

Syndrome	Most likely error
000	No error
001	$XIIIIII$
010	$IXIIIII$
011	$IIXIIII$
100	$IIIXIII$
101	$IIIIXII$
110	$IIIIIXI$
111	$IIIIIIX$

Table 3.2: Look-up table for decoding the syndrome obtained by Z-type stabilizer measurements for the Steane code. It can be derived analogously for the syndrome obtained from X-type stabilizers.

in this thesis (see Sec. 6). In general, simulating the evolution statevector of a quantum system scales exponentially in the number of qubits, and this result shows that for circuits comprising of gates only from the Clifford group, which is the normalizer of the Pauli group, and Pauli measurements, the simulation can be performed classically in polynomial time.

Theorem 3.4.1 *Any quantum computer performing only: a) Clifford group gates, b) measurements of Pauli group operators, and c) Clifford group operations conditioned on classical bits, which may be the results of earlier measurements, can be perfectly simulated in polynomial time on a probabilistic classical computer [34].*

Circuits for syndrome extraction in stabilizer codes consist only of Clifford group elements, therefore, the simulation can be performed as follows. Suppose there is a Pauli error P on a qubit, and it is followed by a gate, C, from the Clifford group. Being the normalizer, defined in Sec. 2, this gives $CPC^\dagger = P'$, where P' is also a Pauli operator.

Therefore, the effect of the gate C is to transform the Pauli P operator into another Pauli operator P', which may or may not be the same as P, depending on the conjugation rule. This analysis can be performed for all gates in the circuit, obtaining the transformed Pauli operator after every gate via conjugation, which forms the input Pauli operator for the next gate. This method requires the knowledge of how the Paulis are mapped under conjugation by a given Clifford. Once this process of "pushing the errors through the gates" is done for all gates in the circuit, the resulting Pauli operator acting on the data qubits can be used to determine whether there has been a logical error or not: if it commutes with all logical operators, then it is not a logical error. A measurement in the $Z(X)$ basis gives the -1 eigenvalue if the $X(Z)$ component of the error acting on that qubit is 1, and gives the $+1$ eigenvalue otherwise.

3.5. Notation and Error Propagation Rules

This section presents the notation for quantum gates used in this thesis, and the error propagation rules for these gates. These rules are derived by conjugating the Pauli errors by the gate, which belongs to the Clifford group, giving the Pauli error after propagation through the gate.

The XNOT gate (see Fig. 3.6(a)), or dual CZ, is defined as the composite gate $(H \otimes I)(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X)(H \otimes I)$, or equivalently, as $(I \otimes H)(I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|)(I \otimes H)$ [25]. The YNOT gate (see Fig. 3.6(b)) is defined as the composite gate $(I \otimes H)(I \otimes |0\rangle\langle 0| + Y \otimes |1\rangle\langle 1|)(I \otimes H)$, i.e. the left qubit is the target of the controlled-Y.

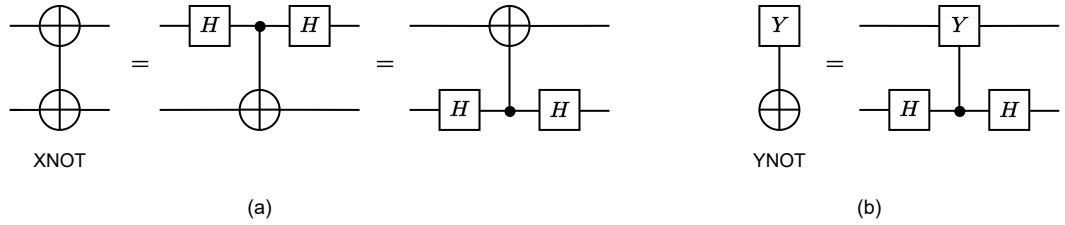

Figure 3.6: Definition and notation for the XNOT gate (a) and the YNOT (b) gate.

Although the XNOT and YNOT gates may not be commonly discussed in standard literature [1], they are employed in analysis of flag fault tolerance protocols [25], [35]. In particular, the YNOT gate and associated error propagation rules have been presented here because of its use in measuring Y parities in Sec. 5, (in particular, Sec. 5.5).

Fig. 3.7, 3.8, 3.9 and 3.10 review the propagation rules for input Pauli errors on one qubit through two-qubit gates.

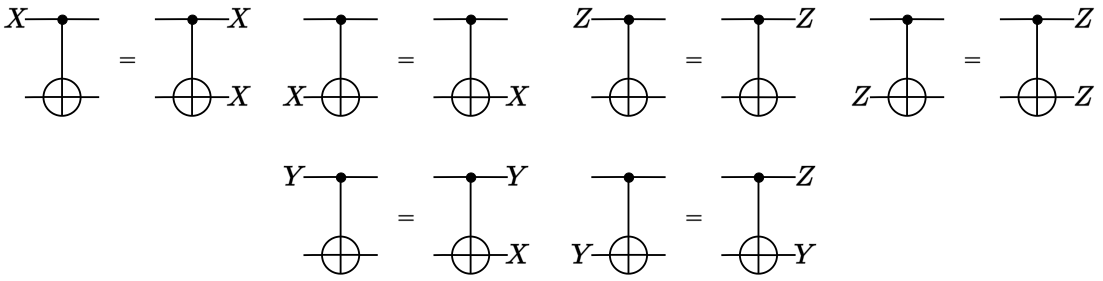

Figure 3.7: Error propagation rules for the CNOT gate.

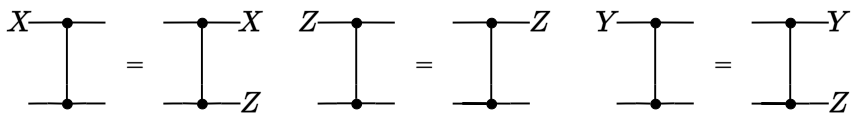

Figure 3.8: Error propagation rules for the CZ gate.

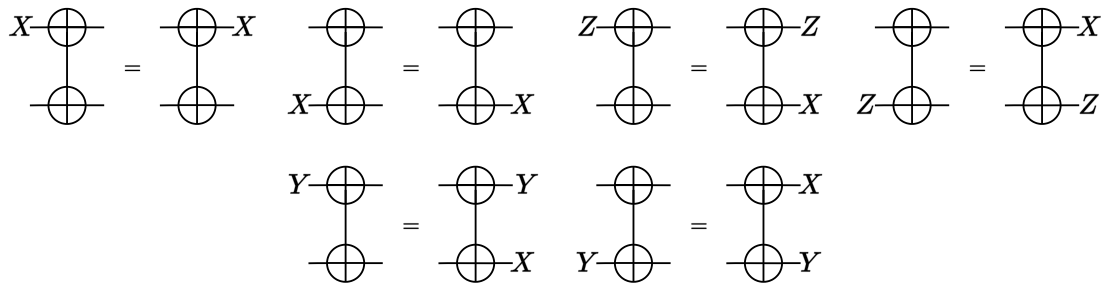

Figure 3.9: Error propagation rules for the XNOT gate.

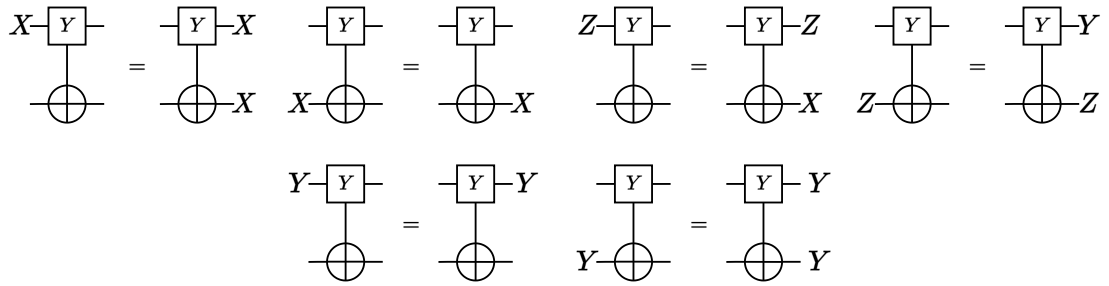

Figure 3.10: Error propagation rules for the YNOT gate

3.6. Fault-Tolerant Quantum Error Correction

In present-day computation, every circuit component, including state preparation, quantum gates, measurements, qubit reset, and occasion for the qubit to be idle, may introduce errors [28]. This necessitates the construction of a fault tolerance protocol based on a quantum error correcting code. The subject of fault tolerance addresses how to perform an arbitrarily long quantum computation reliably, with arbitrary accuracy, using quantum error correction, even if the individual components used to perform error correction are themselves noisy [1],[12],[13]. An introductory definition of a fault-tolerant quantum error correction circuit for distance-3 codes is that a single input error, or a single component failure occurring anywhere in the circuit, should lead to at most a single qubit error in the output, so that the code is able to recover from its effect [1].

This part of the background presents the effect of noisy circuit components on syndrome extraction in Sec. 3.7, followed by standard terminology in Sec. 3.8, and fault tolerance conditions and the threshold theorem in Sec. 3.10. This is followed by a brief review of traditional fault tolerance schemes in Sec. 3.11. The main focus of the thesis will only be on fault-tolerant syndrome measurement (and not on other elements of a fault-tolerant quantum computation, namely, encoding, logical measurement, and logical operations). The main references are [13],[1],[14],[4],[12].

3.7. Noisy Circuit Components and Syndrome Extraction

If circuit components are potential sources of error, the ability to perform syndrome extraction reliably is impacted, and fault tolerance may get compromised. In general, the errors may be coherent. By the Pauli error model argument, discussed in Sec. 3.1, only Pauli errors are considered. Thus, when discrete errors may occur, the components are said to have a probability of failure. A single component may fail with a probability $\mathcal{O}(p)$.

The component failures are described in the list below [1],[25]. See Fig. 3.11. These errors are assumed to be uniformly distributed over the relevant error set. Following this description, the impact on syndrome extraction is discussed.

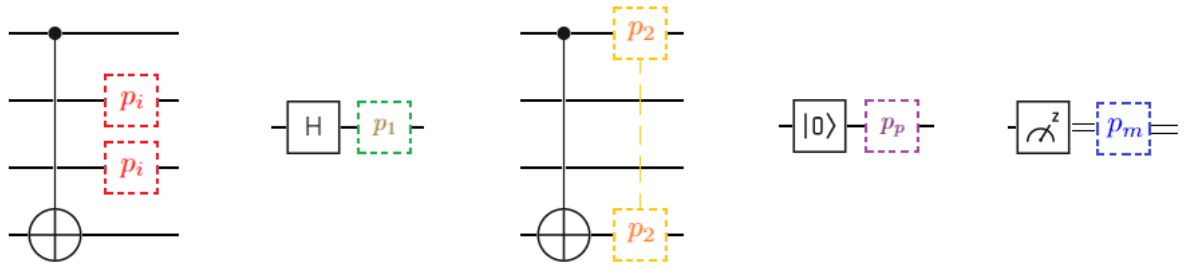

Figure 3.11: Possible errors due to component failures. From the left: idling error (shown in red, occurring with probability p_i), single-qubit gate error (shown in green, occurring with probability p_1), two-qubit gate error (shown in yellow, occurring with probability p_2), preparation error (shown in purple, occurring with probability p_p), and measurement error (shown in blue, occurring with probability p_m), respectively. The dashed line connecting the errors after the two qubit gate signifies that the errors on the two qubits are not independent.

- If a single qubit gate has a failure probability p_1, the noiseless gate is followed by either X, Y or Z on the qubit, each with probability $\frac{p_1}{3}$. The probability that the gate does not fail is $1 - p_1$.
- If a two qubit-gate has a failure probability p_2, the noiseless gate is followed by a two-qubit Pauli string $P_i \otimes P_j \neq I \otimes I, P_i, P_j \in \{I, X, Y, Z\}$ on the two qubits, each with probability $\frac{p_2}{15}$. This is because a two qubit gate may introduce errors on both qubits with a probability linear in p_2, as opposed to a probability quadratic in p_1, which corresponds to two independent single-qubit errors [28].
- An error in preparation of the state $|0\rangle$ (or $|+\rangle$) results in the erroneously prepared state $|1\rangle$ (or $|-\rangle$, respectively), with probability p_p [25].
- A measurement error results in the output classical bit to get flipped with probability p_m [25].
- A qubit which is 'resting', i.e. undergoing no operation, may get affected by errors. This may occur when another qubit is being prepared, measured, or acted upon by a gate. This is called an idling error [35]. The idling location may get affected by an X, Y or Z error on the qubit, each with probability $\frac{p_i}{3}$.

Due to these errors, in general, syndrome extraction with the circuits presented so far (in Sec. 3.1) is insufficient to correct all possible errors arising from single component failures. This is illustrated with some examples below, to demonstrate that additional circuit constructions are needed to make syndrome extraction fault tolerant.

The first example, shown in Fig. 3.12, shows that fault tolerance requires that inequivalent errors resulting from single component failures need to be unambiguously distinguishable by their syndromes. Consider syndrome measurement for the Steane code with an input X error on the first qubit. The syndrome obtained by measuring the 3 Z-type stabilizer generators $IIIZZZZ$, $IZZIIZZ$, $ZIZIZIZ$ is 001. However, another syndrome measurement scenario with no input error but an X error on the third qubit after measuring $IZZIIZZ$ results in the same syndrome, 001. This error may have resulted from a two qubit gate error or an idling error. When the same syndrome arises from inequivalent errors, it will be referred to as 'syndrome collision'.

If the syndrome 001 is erroneously interpreted as an input error $XIIIIII$, whereas its actual occurrence corresponds to the second scenario, the correction would lead to the application of the weight-2 operator $XIXIIII$ on the data qubits. This operator differs from a logical error by only a weight-1 operator, because the distance of the Steane code is 3. Therefore, application of this correction would further contribute an error to the state. Thus, this syndrome extraction sequence is not fault-tolerant, and the resulting syndrome is not reliable for decoding.

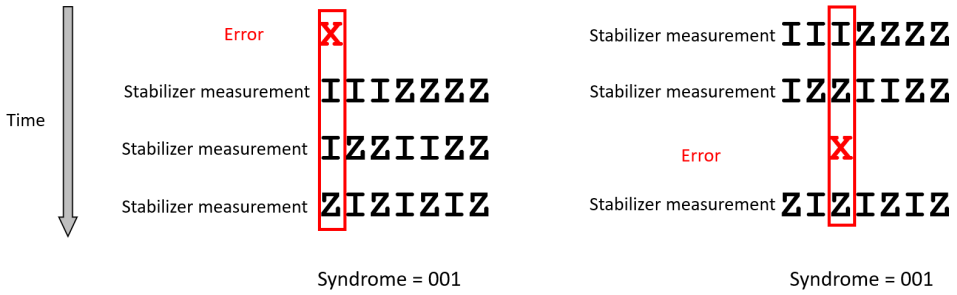

Figure 3.12: The syndrome collision scenario described in the text for syndrome extraction for the Steane code. The errors are coloured in red, and the qubits on which they act are enclosed in a box. The syndrome obtained is the same, namely 001, if an X error happens on the first qubit before the first stabilizer measurement, or on the third qubit before the third stabilizer measurement. Since these errors are not equivalent, the collision of syndromes implies this stabilizer sequence is not fault-tolerant.

The second circumstance is propagation of errors to multiple qubits, which forms an impediment to fault tolerance. The circuit for a single stabilizer measurement for the Steane code, shown in Fig. 3.13, is not fault-tolerant, because of error propagation through the shared ancilla qubit. If the second CNOT gate fails with a Z error on the ancilla qubit, this error forms an input Z error for the target of the next CNOT gate. This results in a $Z \otimes Z$ error, due to conjugation of the error by CNOT. That is, a Z error on the target of a CNOT propagates to its control qubit. Subsequently, the Z error on the ancilla qubit propagates to another data qubit throught the fourth CNOT gate. This results in a weight-2 Z error on the data qubits, which is not correctable by the Steane code. Therefore, by the introductory definition of fault tolerance presented in the Sec. 3.6, the circuit is not fault-tolerant because propagation of a single error may lead to uncorrectable errors.

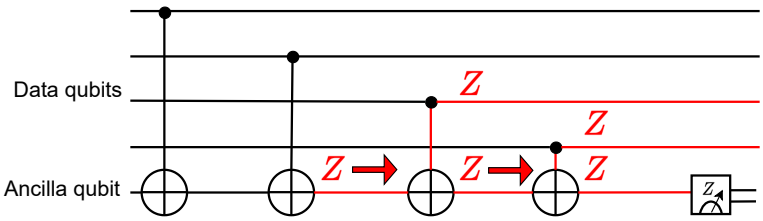

Figure 3.13: Error propagation in the circuit for measuring a Z stabilizer of the Steane code. A Z error on the ancilla after the second CNOT propagates to a weight-2 error on the data qubits. Therefore, this circuit is not fault-tolerant. Data qubits on which the stabilizer does not have support are not shown.

As the third example, preparation and measurement errors on ancilla qubits may render the measured syndrome unreliable for decoding. If a measurement error leads to an uncorrectable error, the circuit is not fault-tolerant.

3.8. Terminology: Fault Tolerance

This section presents standard terminology used in fault tolerance literature [12],[13],[35],[36].

- The notion of a **fault** is used to denote the failure of a component within a circuit. In contrast, an **input error** is an error occurring before a quantum error correction circuit. In general, the term ‘error’ will also be used to denote the resulting error from a fault.
- Every noisy gate, state preparation, measurement, and occasion for a qubit to be idle is a potential fault **location**.
- The type and probability of errors at these locations, or whether some locations are considered noiseless, is specified by the **error model**.

- A **gadget** is a composite construction using elementary components (gates, state preparation and measurement) to create a fault-tolerant operation for a gate, state preparation, measurement, or idling location. Its purpose could be to reduce the propagation of errors, or, as in the case of the flag stabilizer measurement gadget (chapter 4), first detect and then correct a propagated error. A quantum error correction gadget is an fault-tolerant encoded gadget to perform stabilizer measurements fault-tolerantly.
- In this thesis, a **fault-tolerant protocol** will be the quantum error correction gadget, to measure syndromes and correct errors fault-tolerantly. In general, a fault-tolerant protocol additionally contains gadgets corresponding to gate locations in the original circuit [13].
- For distance-3 codes, a location is **bad** if a single fault at that location has the potential to lead to an error with minimum weight at least 2 (i.e. an uncorrectable error), where the minimum is over all equivalent errors obtained by multiplication with stabilizers [35].
- Circuit components are considered to fail linearly in the parameter p, the **physical error rate**. The **logical error rate** will be the probability that the encoded state can undergo a logical error in an iteration of error correction.
- A single error or fault, assumed to occur with probability which scales linearly in p, will be referred to as an $\mathcal{O}(p)$ error. Independent errors resulting from two locations, will be assumed to occur with probability which scales quadratically in p, will be referred to as $\mathcal{O}(p^2)$ errors. Errors which occur with probability scaling as $p^2, p^3, \ldots$ will be referred to as higher order errors.
- In the context of distance-3 codes, pairs of fault locations which lead to a logical error are called **malignant pairs**, and those which don't cause a logical error are **benign**.

3.9. Pauli Error Models

This section defines the Pauli error models for circuits, used to assess fault tolerance of a circuit in QEC literature [15] via evaluation of the (pseudo)threshold, described in Sec. 3.10, under the specific noise model. In this thesis, only Pauli errors are considered on qubits, and these models assume a uniform distribution of errors, which are independently and identically distributed (i.i.d.) across fault locations.

Besides Pauli error models, more sophisticated error models are also possible, like biased noise and continuous noise based on physical processes [1]. Working with Pauli errors for stabilizer codes provides an efficient method of analysis due to the Gottesman-Knill theorem. (see Sec. 3.4). In addition, as described in Sec. 3.1.3, analyzing Pauli error is sufficient to analyze QEC with continuous noise models, which are the most accurate, but computationally expensive to simulate [1].

This thesis follows Knill's error model for majority of the analysis, in line with the main reference [25].

3.9.1. Code Capacity Noise

In this error model [15], all state preparations, measurements, and single and two-qubit gates in the quantum circuit are noiseless, and only the error correction capacity of the code is tested. Ancilla qubits are also noiseless at all time steps. The data qubits can only experience an error with a probability p at one time step, which is after encoding and before any stabilizer measurement.

The code capacity noise model is shown in Fig. 3.14. The error locations are shown with red boxes.

3.9.2. Phenomenological Noise

This error model [15] adds measurement errors to code capacity noise model. With probability p_m, a measurement outcome is flipped. The structure of the phenomenological noise model is shown in Fig. 3.15. Measurement errors are shown with blue boxes, and data qubit errors with red boxes.

3.9.3. Knill's Error Model

To model failure of quantum operations, Knill introduced an error model in [28]. The parameter p is used to specify the relative error rates. The structure of the Knill's error model is shown in Fig. 3.16. The component error rates under Knill's error model are described in the list below, with reference to the figure.

- State preparation of $|0\rangle$ ($|+\rangle$) incorrectly prepares the state in $|1\rangle$ ($|-\rangle$) with probability $p_p = \frac{4p}{15}$ (shown in purple).

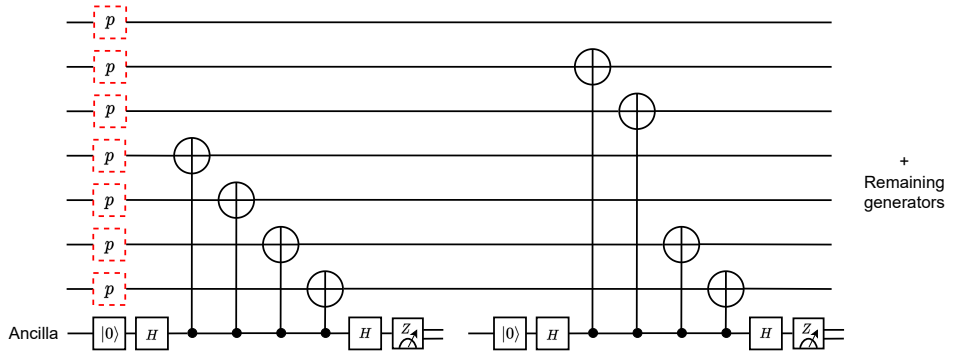

Figure 3.14: The code capacity noise model, for a part of the circuit where the first two stabilizers of the Steane code (beginning with $IIIXXXX$ and then $IXXIIXX$) are measured. Errors (i.i.d., shown in red) act on data qubits prior to the first stabilizer measurement of a QEC cycle.

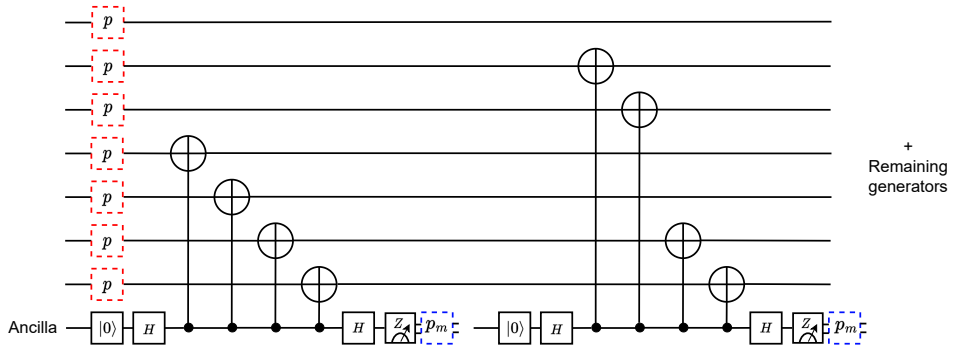

Figure 3.15: Phenomenological noise, for a part of the circuit where the first two two stabilizers of the Steane code are measured. This noise model adds measurement errors (blue boxes) in addition to errors occurring on data qubits present in the code capacity model (red boxes).

- A single-qubit gate is followed by one of $\{X, Y, Z\}$, each with probability $\frac{p_1}{3}$, where $p_1 = \frac{4p}{5}$ (shown in green).
- A two-qubit gate is followed by one of the 15 non-identity two-qubit Pauli operators, i.e. from $\{P_i \otimes P_j\} \setminus I \otimes I, P_i, P_j \in \{I, X, Y, Z\}$, each with probability $\frac{p_2}{15}$, where $p_2 = p$ (shown in orange).
- A single qubit measurement outcome is flipped with probability $p_m = \frac{4p}{15}$ (shown in blue).
- There are no errors on idling locations.

Knill's error model is considered in this thesis for pseudothreshold simulations (see Sec. 6), and introductory treatment of the flag gadget (see Sec. 4), to establish a comparison of preliminary fault tolerance analysis under identical conditions with the main reference on flag fault tolerance protocols for this work [25], wherein this model has been employed.

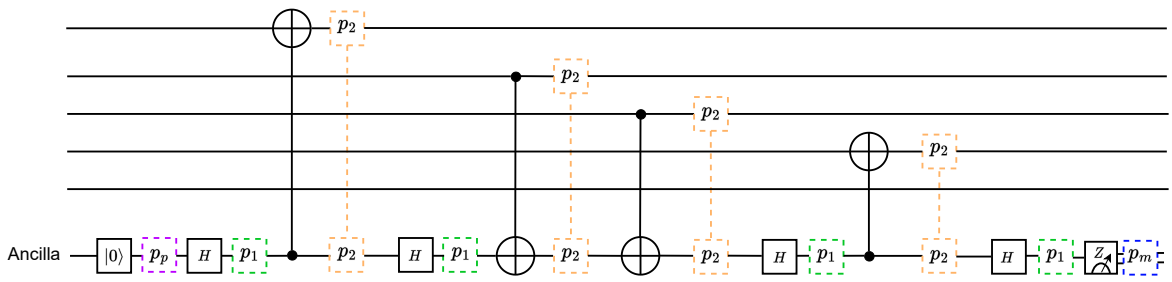

Figure 3.16: Knill's error model, for a stabilizer measurement ($XZZXI$) for the $[\![5, 1, 3]\!]$ code. This error model includes state preparation errors (shown in purple), single qubit gate errors (green), two-qubit gate errors (orange), and measurement errors (shown in blue).

3.9.4. Circuit Level Noise

This is the most stringent Pauli error model, as every circuit location can contribute a potential fault [15]. The structure of the circuit level noise model is shown in Fig. 3.17. The component errors under circuit level noise are described in the list below, with reference to the figure.

- State preparation of $|0\rangle$ ($|+\rangle$) incorrectly prepares the state in $|1\rangle$ ($|-\rangle$) with probability p_p (shown in purple).
- A single-qubit gate is followed by one of $\{X, Y, Z\}$, each with probability $\frac{p_1}{3}$ (shown in green).
- A two-qubit gate is followed by one of the 15 non-identity two-qubit Pauli operators, i.e. from $\{P_i \otimes P_j\} \setminus I \otimes I, P_i, P_j \in \{I, X, Y, Z\}$, each with probability $\frac{p_2}{15}$ (shown in orange).
- A single qubit measurement outcome is flipped with probability p_m (shown in blue).
- For every occasion for a qubit to lie idle, for instance, when an operation occurs on another qubit, it experiences a uniformly depolarizing error, i.e. one of $\{X, Y, Z\}$, each with probability $\frac{p_i}{3}$ (shown in red).

3.10. Conditions for Fault Tolerance and the Threshold Theorem

The threshold theorem [12], [13], [1] is the central result in fault-tolerance which guarantees that suppression of errors using quantum error correction is possible, even if all circuit components can introduce errors, provided the conditions of the theorem are satisfied. The introductory definition of fault tolerance for distance-3 codes, presented in the Sec. 3.6, implies that the logical error rate should scale at least as $\mathcal{O}(p^2)$ [1]. If this holds, the error rate after encoding is suppressed, provided p is below a certain value, called the accuracy threshold, or simply the threshold. If p is above the threshold, then the noisy circuit contributes more errors than the circuit can correct. If the logical error rate has an $\mathcal{O}(p)$ component, then the construction does not suppress errors [1]. In that case, it does not have a non-trivial threshold, which means there is no value of p below which the logical error rate is suppressed, and is not fault-tolerant.

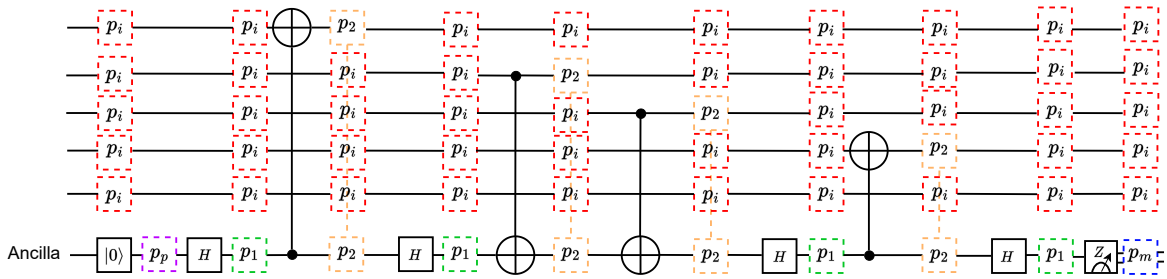

Figure 3.17: Circuit level noise, for a stabilizer measurement ($XZZXI$) for the $[\![5,1,3]\!]$ code. This error model includes state preparation errors (shown in purple), single qubit gate errors (green), two-qubit gate errors (orange), measurement errors (blue), and idling errors (red).

3.10.1. Fault Tolerance Conditions

For block codes, two conditions for fault-tolerant error correction are required [13],[35],[27],[12]:

1. If the state input to the error correction gadget has r errors, and the gadget introduces s faults, such that $r + s \leq t$, for a t-correcting QEC code, then the codeword closest to the output state is the same codeword as the one closest to the input state. This means that correctable errors should not spread to errors which are uncorrectable. For distance-3 codes, it means 1 input error or 1 fault should not lead to a logical error.
2. If the state input to the error correction gadget has an arbitrary number of errors, and the gadget introduces s faults, where $s \leq t$, then the output state is at most a weight s error away from some codeword, so that it can be returned to the codespace by a subsequent error correction cycle. The $[\![5,1,3]\!]$ code and the Steane code, the main focus of this thesis, satisfy this condition, because every corrupted state for these codes is a weight-1 error away from some codeword [27],[35].

3.10.2. The Threshold theorem

Theorem 3.10.1 *Assuming errors act randomly and independently on circuit locations for a quantum computation of size polynomial in the number of qubits, it is possible to perform the same computation using a fault-tolerant protocol with an arbitrarily low error rate, with polylogarithmic overhead, provided the error rate per physical component is less than a certain threshold value [12],[13],[1].*

The threshold sets a target for the error rate of every component, because the logical error rate is suppressed only if the physical error rates of all components are below the threshold.

For block codes, the theorem can be proved via code concatenation [1]. In code concatenation, the encoded (logical) qubits of one code are again encoded into the logical qubits of another code. For a given physical error rate, the logical error rate can be made arbitrarily low by increasing the distance of the code via recursive concatenation. The suppression of error rate scales as an exponential of the distance [1].

For example, consider the 3-qubit repetition code for bit-flip errors, with the same error model (1 bit flip) as in Fig. 3.1. The logical error rate is be plotted against the physical error rate in Fig. 3.18.

For $p < 0.5$, the logical error rate is lower than the physical error rate, seen by the intersection point with the $y = x$ line in Fig. 3.18. As the distance of the repetition code is increased, this suppression still happens for $p < 0.5$ (not shown in Fig. 3.18), hence this is the threshold under the specified error model[1].

Going beyond the abstract statement, the threshold theorem [12] implies certain requirements need to be satisfied to achieve fault tolerance [37]. The error rate of each component must be lower than the accuracy threshold, and propagation of errors needs to be controlled. Additionally, the errors should affect individual qubits independently.

3.10.3. Threshold vs Pseudothreshold

In general, different circuit components may fail with different error rates, and are replaced with different gadgets in a fault tolerance protocol. The suppression of logical error rate may be analyzed as a function

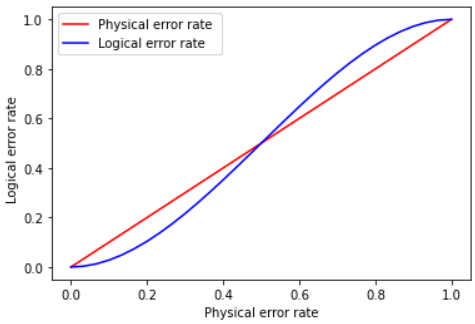

Figure 3.18: Logical error rate (blue curve) for the 3-qubit repetition code for bit-flip errors, assuming a single, i.i.d. bit-flip error on data qubits, after encoding and before stabilizer measurements. This curve can be obtained by either carrying out a Monte Carlo simulation, or by enumerating the malignant set of errors under this error model. The threshold is the value of physical error rate (red curve) at which the physical error rate and the logical error rate curves cross, which is 0.5. Below the threshold, the logical error rate is lower than the physical error rate.

of increasing code distance, achieved via code concatenation [13]. As a consequence of differential component error rates, the logical error rate curves for different code distance values may not intersect the physical error rate curve at the same point [38]. The threshold is the point where the logical error rate curves for different concatenation levels asymptotically converge.

In this thesis, logical error rates are analyzed for a single level of the code, without further concatenation. Therefore, following the terminology in [38], the intersection point of the logical error rate curve with the physical error rate curve is denoted as the pseudothreshold. This may be different from the asymptotic threshold, and is a measure of performance for the single level of the code. A qualitative illustration of the distinction between the concepts of threshold and pseudothreshold is shown in Fig. 3.19, adapted from [38].

In addition, the pseudothreshold and the threshold values are contingent upon the specific error model and failure probabilities associated with individual circuit components [15].

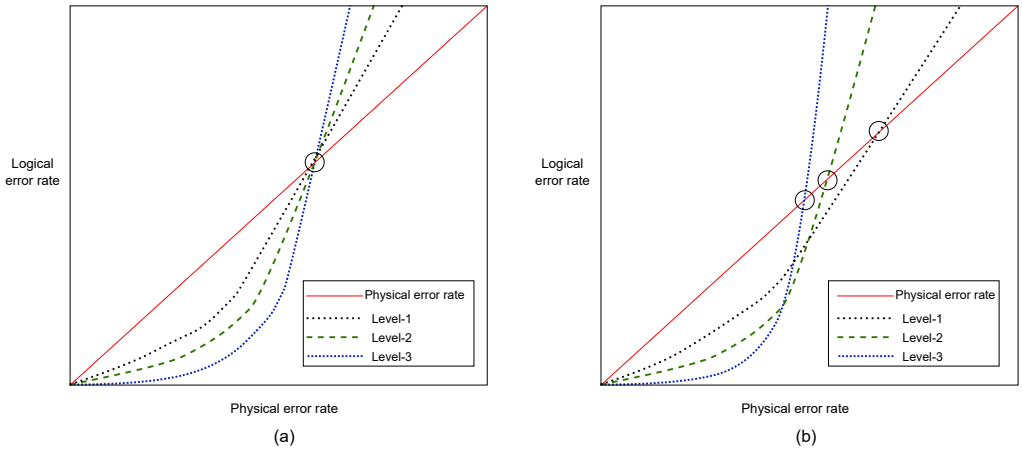

Figure 3.19: Qualitative illustration of the distinction between the concepts of threshold and pseudothreshold. In (a), the logical error rate curves for different levels of code concatenation converge to intersect the physical error rate curve (in red) at the same point (encircled). This corresponding physical error rate value is referred to as the threshold. In (b), the logical error rate curves for different levels of code concatenation intersect the physical error rate curve at different points (encircled). Each of these points represents a pseudothreshold. The threshold may be computed from the point to which these curves asymptotically converge. This figure is adapted from [38].

3.11. Traditional Fault Tolerance Schemes

A variety of fault-tolerant syndrome extraction schemes, or fault-tolerant quantum error correction gadgets, have been developed in the literature [13]. The traditional fault tolerance schemes developed by Shor [22], Steane [23] and Knill [24] form three core families of these schemes. These established schemes continue to hold significance for construction, analysis, and as a comparative reference for performance of modern fault tolerance schemes [25]. Therefore, these are reviewed in this section. The extensive discussion of Shor's scheme is motivated by a fundamental reference [27] for the present work (see Sec. 5). The schemes due to Steane and Knill, though not of direct relevance, are described briefly for completeness.

3.11.1. Shor Error Correction

In a stabilizer measurement circuit, if multiple two-qubit gates interact with a single ancilla qubit, a single error on the ancilla can propagate through the gates to multiple data qubits, which may compromise fault tolerance (see Sec. 3.7). Shor error correction [22],[13] restricts error propagation during stabilizer measurement by replacing the ancilla qubit by an ancillary system consisting of multiple qubits, prepared in a cat state. The number of ancilla qubits is equal to the weight of the stabilizer, and the cat state is presented below:

$$\frac{|00\ldots0\rangle + |11\ldots1\rangle}{\sqrt{2}}. \tag{3.24}$$

For fault tolerance, the cat state preparation needs to be checked for errors. This is done by measuring the parity of a pair of qubits, and the cat state is verified to be correct if the parity is even. The Shor error correction scheme (or the Shor fault tolerance scheme) is described in Fig. 3.20.

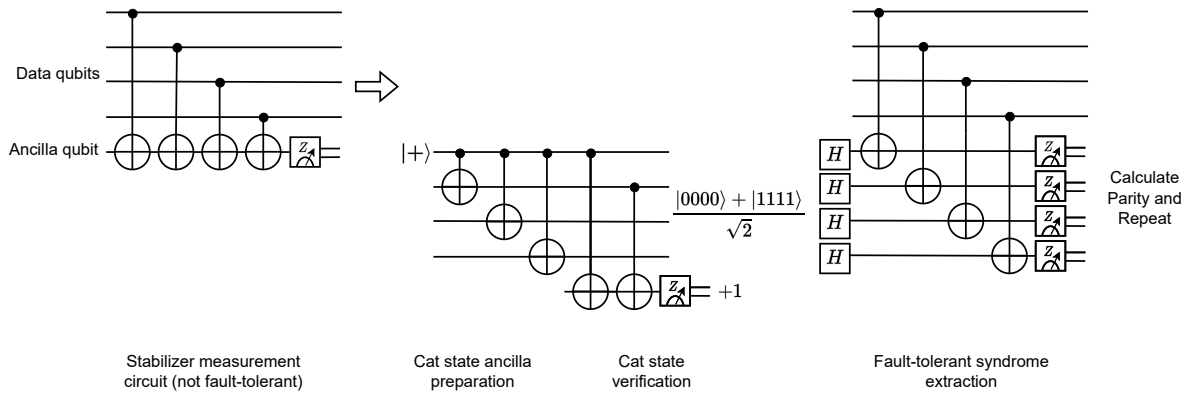

Figure 3.20: Schematic of the Shor error correction scheme [22], [13]. Fault tolerance is guaranteed by suppressing error propagation via transversal two-qubit gates between data qubits and ancilla qubits prepared in a verified cat state, and repetition of stabilizer measurements.

Syndrome measurement is carried out by measuring one stabilizer at a time, by allowing each two-qubit gate from a data qubit to interact with only one of the ancilla qubits of the (verified) cat state (i.e., in a transversal manner). As a consequence, even if a two qubit gate fails, the resulting error on the ancilla qubit does not propagate to other data qubits, because the ancilla qubit is not acted upon by a subsequent quantum gate.

The cat state is used because interaction with this state only extracts parity information about the errors, and does not leave the data qubits entangled with the ancillas. Syndrome extraction is performed as follows. For measuring a stabilizer with Z operators, the cat state is first converted to a state with superposition of all even parity terms via hadamard gates, then CNOT gates are applied transversally between data qubits and the ancillas, and the ancillas are measured in the Z basis. The parity (of the number of ones) of the measured values forms the syndrome bit. X-type stabilizers can be measured analogously, without using the hadamard gates, and by measuring the ancillas in the X basis. This controls propagation of errors during syndrome extraction, because they do not spread to more than 1 qubit.

To guarantee fault tolerance, the stabilizer measurements through the cat state need to be repeated

until the same syndrome is obtained a sufficient number of times [27]. Consider the same example presented in Sec. 3.7 (see Fig. 3.12), namely, the occurrence of an X error on qubit 3 after measuring the second stabilizer generator of the Steane code.The measurement of a stabilizer only once, even with the use of a cat state ancilla, is still vulnerable to such a syndrome collision scenario. To unambiguously diagnose errors, syndrome measured needs to be repeated, and the observation of the same syndrome consecutively signifies that no error has occurred in between stabilizer measurement (with high probability). The syndrome obtained consecutively from repetition of measurements is reliable for fault tolerance. For every repetition, the process of cat state preparation, verification, and syndrome extraction is repeated, to prevent syndrome collision.

Additionally, the scheme is fault-tolerant, through repetition of stabilizer measurements, against error propagation during ancilla verification, because this may lead to at most a weight-1 error on a data qubits. The repetition also guarantees fault-tolerance against measurement errors (see Fig. 3.20).

Shor error correction applies to any stabilizer code, and for measuring a weight w stabilizer, it requires a qubit overhead of $w+1$, due to the qubits in the cat state, and the verification qubit.

3.11.2. Steane and Knill Error Correction

Steane error correction [23] [13] applies specifically to CSS codes. In this scheme, the ancilla qubit is encoded in the same quantum error-correcting code, and the ancilla preparation is verified by measuring stabilizer generators as parity checks on the ancilla state, via an additional qubit [23],[13]. To measure the Z-type stabilizers, the ancilla is prepared in the encoded state $|\bar{+}\rangle = \frac{|\bar{0}\rangle+|\bar{1}\rangle}{\sqrt{2}}$, and CNOT gates are applied transversally from data qubits to the (verified) ancilla block. The ancilla qubits are measured, and the syndrome is obtained by classically calculating the parities corresponding to stabilizers from the measurement outcomes. The X-type parities are extracted analogously by verified preparation of the ancillas in $|\bar{0}\rangle$, applying CNOT gates transversally from ancilla qubits to data qubits, and measuring in the X basis.

With Steane error correction, syndromes corresponding to stabilizers of the same type (X or Z) can be extracted in parallel. As a measure of qubit overhead, for an $[\![n,k,d]\!]$ CSS code, Steane error correction requires $2n$ ancillas per round of stabilizer measurements.

Knill error correction [24][13] applies to any stabilizer code, and employs verified preparation of ancillas in the $2n$-qubit logical Bell state $\frac{|\bar{0}\bar{0}\rangle+|\bar{1}\bar{1}\rangle}{\sqrt{2}}$. A Bell basis measurement of the data qubit block and one logical qubit from the logical Bell pair teleports the encoded state to the other half of the Bell pair, and the measurement outcome is used to determine the syndrome [24].

Steane and Knill error correction allow syndromes to be extracted in parallel, i.e. with lower circuit depth, in general, they may require more ancilla qubits than Shor error correction. This leads to an increase in fault locations for the ancilla qubits. Preparation and verification of ancillary states also adds to circuit overhead.

4

Flag Fault Tolerance Protocols

This section presents the flag gadget, in Sec. 4.1, followed by developing a terminology for flag fault tolerance protocols in Sec. 4.2, and the flag fault tolerance protocols for the $[\![5,1,3]\!]$ code and the Steane code, in Sec. 4.3, as originally developed by Chao and Reichardt [25]. The main reference for this section is [25]. Flag fault tolerance protocols, based on using an extra qubit (called the flag) to detect badly propagated errors, have emerged as powerful schemes [25], [35], because they are applicable to small quantum error-correcting codes, require few extra ancilla/flag qubits, and do not require elaborate ancilla state preparation, relative to traditional schemes [25]. Fault tolerance of these protocols is verified in Sec. 4.4, assuming at most 1 error or 1 fault, since the codes have distance 3. This section concludes with an optional and brief outlook on the flag protocols in Sec. 4.5.

4.1. The Flag Gadget

This section presents the motivation to construct a flag gadget, namely, to detect propagation of errors from bad gates during a stabilizer measurement in Sec. 4.1.1. This is followed by construction of the flagged stabilizer measurement circuit in Sec. 4.1.2, and analysis to show that the flag gates do not add bad locations in Sec. 4.1.3.

4.1.1. Motivation

The circuit to measure the stabilizer generator $XZZXI$ of the $[\![5,1,3]\!]$ code (see Fig. 4.1, expressed using the XNOT gate (see Fig. 3.6), is not fault-tolerant due to interaction with a shared ancilla. The two CNOTs are bad locations (see Fig. 4.1(a)), because a single fault after these can propagate to an uncorrectable error (analogous to Fig. 3.13).

The first XNOT gate is not a bad location, because a fault after this gate propagates to an error with weight at most 1, up to stabilizer multiplication (see Fig. 4.1(b)). The fourth XNOT gate is also not a bad location, because a fault after it leads to at most a weight-1 error (see Fig. 4.1(c)).

4.1.2. Flagged Stabilizer Measurement Circuit

The main idea behind flag fault tolerance protocols, developed in [25], is to measure stabilizer generators using circuits with an additional flag qubit [25],[39],[35], to detect propagated errors resulting from bad locations. Note that these are $\mathcal{O}(p)$ errors, and need to be corrected to ensure fault tolerance. This circuit will be called a flagged circuit. Faults after two-qubit gates are allowed to propagate through subsequent two-qubit gates connected to the shared ancilla. The circuit implements interactions between the ancilla qubit with the flag, which causes faults from bad locations, or bad faults, to be detected by measuring the flag. This will be referred to as flagging the bad locations, and the resulting stabilizer measurement circuit as a flagged circuit.

Interaction of the ancilla qubit with the flag refers to the flag first being entangled with the ancilla, via a two-qubit gate, and subsequently being disentangled via another two-qubit gate. These gates are placed such that when a fault propagates from a bad location, it also propagates to the flag qubit, and is detected by measuring the flag qubit.

Upon detecting a nontrivial flag measurement outcome, all stabilizer generators are measured again, this time without the flag (i.e., with unflagged circuits). The resulting syndrome is used to decode and

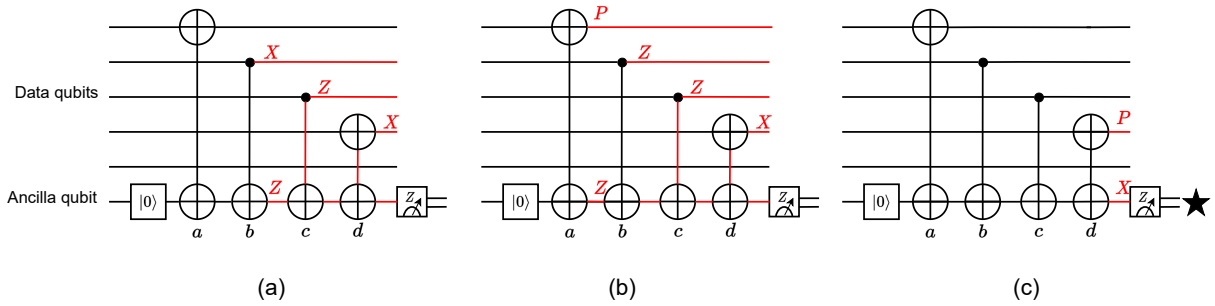

Figure 4.1: The circuit to measure the generator $XZZXI$ for the $[\![5,1,3]\!]$ code, without any fault tolerance constructions. The letters beneath the gates are labels for reference. In (a), an XZ fault after CNOT b propagates to an error on multiple qubits, $IXZXI$. This is equivalent to a weight-2 error $XYIII$ up to stabilizer multiplication, because $IXZXI \circ XZZXI = XYIII$. Therefore, CNOT b is a bad location, and the circuit is not fault-tolerant. Similarly, CNOT c is also a bad location (not shown). In (b), XNOT a fails with an error PZ, where $P \in \{I, X, Y, Z\}$. The resulting error on the data qubits due to propagation is $PZZXI$, which is only a weight-1 error on the data qubits, because $PZZXI \circ XZZXI = (P \circ X)IIII$. Therefore, CNOT a is not a bad location. In (c), a PX fault occurs after XNOT d, and it does not propagate to an error on multiple data qubits through subsequent stabilizer measurement circuits, due to the error propagation rules (section 3.5). The black star indicates a nontrivial syndrome.

correct the propagated error. Importantly, even though these errors may have weight greater than 1, they can still be corrected, if the gates in the flagged circuit are placed in an order which leads to propagated errors which are not logical errors, and result in distinct syndromes for inequivalent errors. Thus, the additional flag measurement increases the space of syndrome information, allowing to correct errors resulting from a single bad location.

In addition to these errors, it is also possible that an error which occurred prior to or during a flagged stabilizer measurement leads to a nontrivial syndrome without the flag getting triggered. It is also possible that a measurement error occurs during the flagged measurement. In both these cases as well, measuring all stabilizer generators with unflagged circuits leads to syndromes which can be used to unambiguously decode these errors. Thus, the use of flagged stabilizer measurements, followed by repeating stabilizer measurements upon a nontrivial syndrome or flag measurement, makes these protocols fault-tolerant.

In addition, it is sufficient to measure the stabilizers subsequent to a nontrivial flag or syndrome outcome without a flag qubit, i.e. with unflagged circuits, to achieve fault tolerance with a distance-3 code. This is because when the unflagged measurements are initiated, the flagged measurement has already indicated the occurrence of a single error, which is the number of required to be corrected with a distance-3 code. The use of another flagged circuit is not necessary, because the flag qubit may detect another error, which is not strictly required to be detected for fault tolerance.

As an example, consider the flagged measurement of $XZZXI$, a stabilizer of the $[\![5,1,3]\!]$ code, shown in Fig. 4.2. The circuit is set up as follows: The ancilla qubit is initialized in $|0\rangle$, and the flag qubit in $|+\rangle$. The two qubit gates between the data qubits and the ancilla extract the required $XZZXI$ parity information. Two CNOT gates, which entangle and disentangle the flag to the ancilla, enclose the bad CNOT gates between the data qubits and the ancilla. Through the CNOT gates with the flag qubits, errors resulting from bad faults also propagate to the flag, as explained below. After these gates, the ancilla qubit is measured in the Z basis, and the flag qubit is measured in the X basis.

Faults after CNOTs b and c can propagate to multiple data qubits, and it is only necessary to consider the cases with a Z error on the ancilla qubit [25]. These errors are enumerated in Table 4.1. The only faults after bad two-qubit gates which need to be considered are those with a Z on the ancilla. This is because an X on the ancilla does not propagate to data qubits, and a Y propagates to data qubits in the same manner as a Z (Refer to the error propagation rules in Sec. 3.5).

The faults in Table 4.1 also propagate to the flag qubit (see Fig. 4.2). The ancilla is also the target qubit of the second CNOT gate between the flag and the ancilla, therefore, the Z error propagates to its

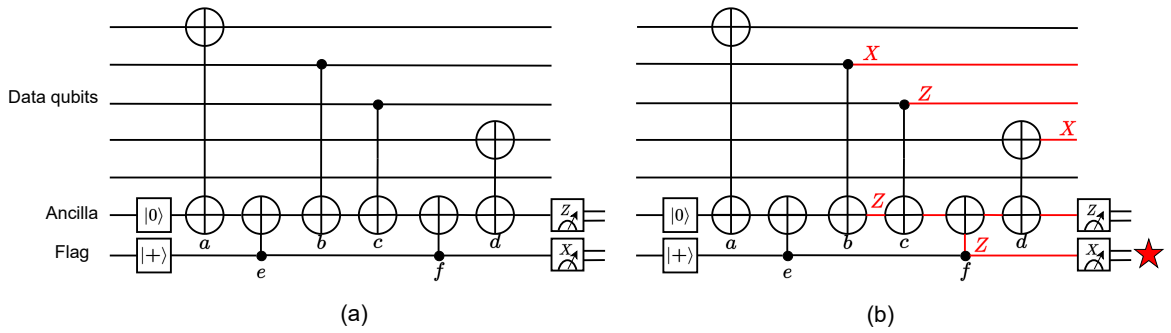

Figure 4.2: (a): Flagged circuit to measure $XZZXI$, a stabilizer generator for the $[\![5,1,3]\!]$ code. (b): As an example, an XZ fault after CNOT b in the same circuit propagates to the flag and gives the -1 outcome after measuring the flag qubit, and can thus be detected. This is denoted by the red star.

Causal fault	Propagated error on data qubits	Syndrome from unflagged stabilizer measurements
IZ fault after CNOT b or ZZ fault after CNOT c	$IIZXI$	0100
XZ fault after CNOT b	$IXZXI$	1100
YZ fault after CNOT b	$IYZXI$	1001
ZZ fault after CNOT b	$IZZXI$	0001
IZ fault after CNOT c	$IIIXI$	0110
XZ fault after CNOT c	$IIXXI$	1010
YZ fault after CNOT c	$IIYXI$	1000

Table 4.1: Errors on data qubits from bad gates during flagged $XZZXI$ measurement, stabilizer generator of the $[\![5,1,3]\!]$ code, using the circuit shown in Fig. 4.2. The first column denotes the fault which can cause the flag to get triggered. As explained in the text, only those faults with a Pauli Z acting on the ancilla need to be considered. The second column denotes the resulting error on data qubits due to propagation. The third column denotes the corresponding syndrome from measuring all stabilizer generators with unflagged circuits. Since these are unique and non-trivial, they can be used to identify each of these errors unambiguously, and also to distinguish them from a flag measurement error (which gives a trivial syndrome).

control, i.e. the flag, due to the propagation rules (see Sec. 3.5). The resulting Z error on the flag qubit anticommutes with the X operator, which is the basis of measurement. Therefore, the Z error shows up as a nontrivial flag measurement outcome (i.e., the flag gets triggered). As an example, an XZ fault after CNOT b during the flagged $XZZXI$ measurement circuit is shown in Fig. 4.2. Faults analogous to those in Table 4.1, but with a Y in place of the Z on the ancilla qubit also propagate to the flag qubit in the same manner, due to the error propagation rules in Sec. 3.5.

If such a propagated error has occurred and all stabilizer generators are measured by unflagged circuits, these errors can be decoded via the resulting syndrome. This is because of two reasons. First, although some of these errors are high-weight errors, none of these errors is a logical error. Therefore, they are detectable by stabilizer measurements. Second, the syndrome obtained by measuring all generators is unique for each (inequivalent) error, and is a nontrivial (i.e. non-zero) value. Therefore, these errors are detectable and can be distinguished unambiguously. A nontrivial syndrome means these errors can be distinguished from a measurement error on the flag as well, because this error gives a trivial syndrome from unflagged measurements. This leads to fault-tolerant error correction rules for these errors, in the form of a look-up table (LUT) in Table 4.2. This is constructed by inverting the error-to-syndrome map in Table 4.1. The corrections may be the actual propagated errors on the data qubits, or equivalent minimum-weight

corrections. These are calculated by multiplying the corrections with stabilizers, and selecting one of the possible operators with minimum weight.

Syndrome from unflagged stabilizer measurements	Correction for most likely error	Minimum-weight equivalent correction, up to stabilizer multiplication
0100	$IIZXI$	$IIZXI$
1100	$IXZXI$	$XYIII$
1001	$IYZXI$	$XXIII$
0001	$IZZXI$	$XIIII$
0110	$IIIXI$	$IIIXI$
1010	$IIXXI$	$IIXXI$
1000	$IIYXI$	$IIYXI$

Table 4.2: The look-up table (LUT) for corrections, given the syndrome resulting from subsequent unflagged stabilizer generator measurements, when the flag gets triggered during flagged $XZZXI$ measurement. This is a stabilizer generator of the $[\![5,1,3]\!]$ code. Equivalent minimum-weight corrections are calculated by multiplying the corrections with stabilizers, and selecting one of the possible operators with minimum weight.

Therefore, faults due to bad locations can be detected by the flagged circuit, and subsequently corrected using all unflagged stabilizer measurements. This also analogously holds for other flagged stabilizer generator measurement circuits for the $[\![5,1,3]\!]$ code. The detectability and distinguishability of errors from bad gates during the other flagged stabilizer generator measurement circuits can be seen in Table C.2, Table C.3 and Table C.4. The errors presented in this table are used to derive the respective lookup-tables for decoding.

A nontrivial syndrome outcome from this circuit without the flag getting triggered is also followed by measuring all (unflagged) stabilizer generators. This leads to the flag fault tolerance protocol for the $[\![5,1,3]\!]$ code, described in Sec. 4.3.

4.1.3. Flag CNOTs Do Not Add Bad Locations

Vital to fault tolerance of the flag schemes is that the two CNOT gates between the ancilla and flag qubits (in the following called the 'flag CNOTs') do not add new bad locations. This can be seen by some examples which are representative of the error propagation behaviour in the flagged $XZZXI$ stabilizer generator measurement, depicted in Fig. 4.3. These are explained below. Following this, fault tolerance against the remaining two-qubit gate faults is discussed.

As the first example, consider an IX fault after CNOT e (the first flag CNOT) in the circuit of Fig. 4.2. This is shown in Fig. 4.3(a). It does not propagate to data qubits, and propagates to the ancilla qubit via the second flag CNOT. This will show up as a nontrivial syndrome, and subsequent measurement of all unflagged stabilizers will yield the trivial syndrome. Thus, no correction is applied, based on decoding via the usual look-up table (Table 3.1). Further, this fault is equivalent to a measurement error on the ancilla, and also to an X fault on the ancilla after any of the two-qubit gates with an I on the other participating qubit, including the flag CNOTs. Thus, the circuit can tolerate this fault.

The second example is a ZX fault after CNOT e, which propagates to give the same error on data qubits as a ZZ fault after CNOT b (see Fig. 4.3(b)). The Z on the ancilla propagates to Z on the flag, and the X on the flag propagates to X on the ancilla, leading to nontrivial measurement outcomes for both. The flag protocol dictates measuring all unflagged stabilizer generators upon this occurrence, and decoding according to the LUT in Table 4.2 corrects this error.

The third example is a ZZ fault after the second flag CNOT (CNOT f), which propagates to a weight-1 error on the data qubits, and also triggers the flag (see Fig. 4.3(c)). This is equivalent to an IZ fault after CNOT c (refer to Table 4.1). Therefore, it can be decoded by measuring all unflagged stabilizers and using the LUT in Table 4.2.

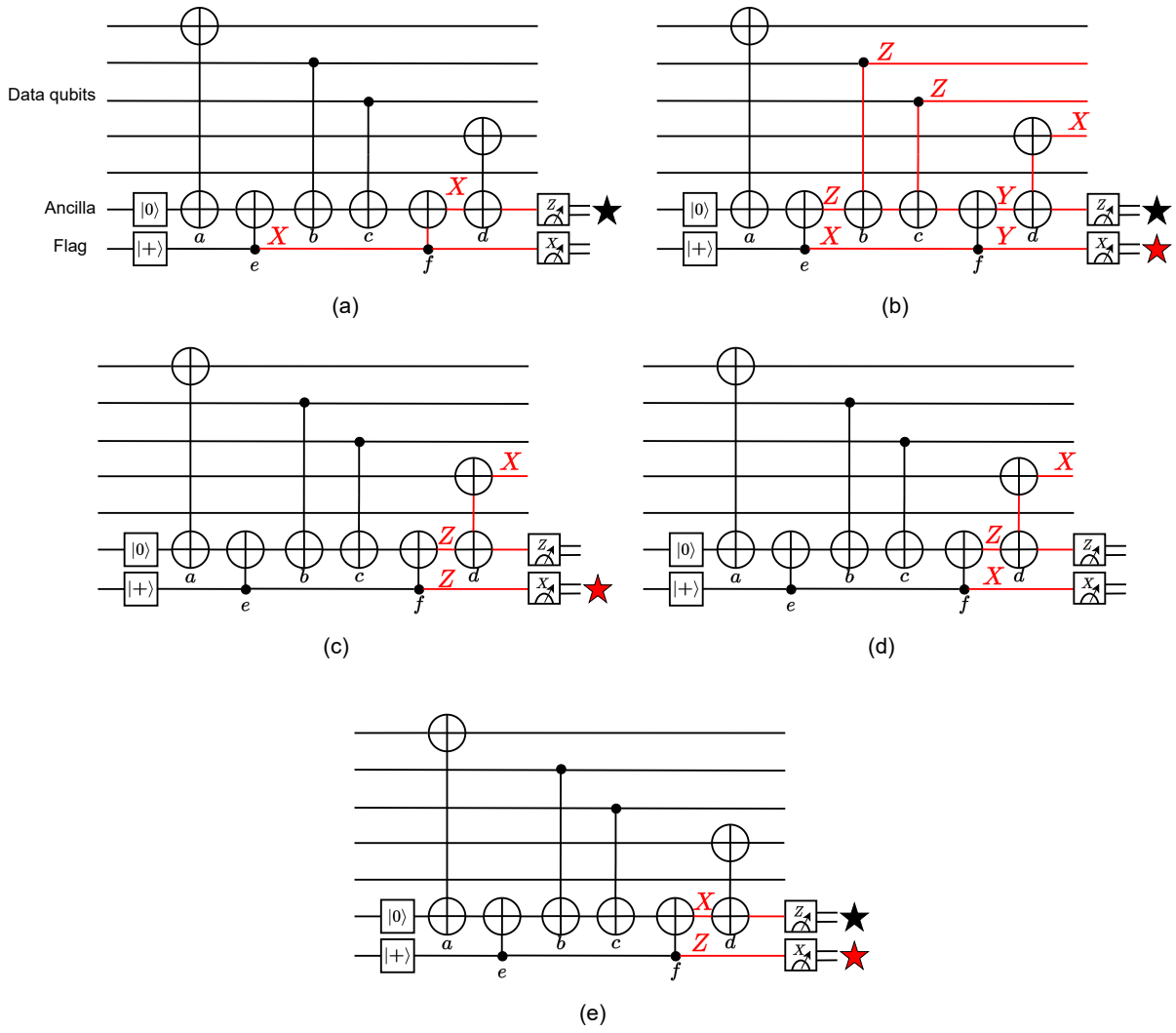

Figure 4.3: Different faults after flag CNOTs described in the text. (a): IX fault after the first flag CNOT (CNOT e), (b): ZX fault after the first flag CNOT, (c): ZZ fault after the second flag CNOT (CNOT f), (d): ZX fault after the second flag CNOT, and (e): an XZ fault after the second flag CNOT.

The fourth example is a ZX fault after the second flag CNOT (see Fig. 4.3(d)). It propagates to a weight-1 X error on the data qubits, and does not lead to a nontrivial flag or ancilla measurement outcome. This error can be detected by a subsequent flagged measurement as a data qubit error. It is shown that this can be corrected in Sec. 4.3, where the complete flag protocol for the $[\![5,1,3]\!]$ code is presented.

The fifth example is an XZ fault after the second flag CNOT. It does not propagate to the data qubits, and leads to both the flag and ancilla measurement giving a nontrivial outcome (see Fig. 4.3(e)). The effect is the same as a measurement error on the flag qubit, as is shown in Sec. 4.3. Thus, this fault also does not lead to an uncorrectable error. The effect of an XZ fault after the first flag CNOT is also the same. Measuring all unflagged stabilizer generators because of the nontrivial flag outcome gives the trivial syndrome, and using the LUT in Table 4.2, no correction needs to be applied.

The explanation for fault tolerance against the remaining two-qubit Pauli errors after CNOT gates e and f in the circuit of Fig. 4.2 is summarized here, since the behaviour of the circuit has already been explained graphically in Fig. 4.3. XX, IZ, XZ, IY, XY, XI faults after CNOT e do not propagate to multiple data qubits. A YX after CNOT e propagates to data qubits in the same manner as a ZX, which has already been explained in Fig. 4.3. YZ, ZZ, YY, ZY, YI, ZI faults after CNOT e propagate to weight-1 errors up to stabilizer multiplication. With respect to CNOT f, the IZ, IX, IY, XX, XY and XI faults after this gate do not propagate to data qubits. ZI, ZY, YI, YX, YY, and YZ faults propagate to weight-1 errors on the data qubits. Therefore, combining all cases discussed in this section, there are at most weight-1 errors on data qubits resulting from at most 1 two-qubit gate fault after both the flag CNOT gates, which signifies these do not add bad locations. Such analysis also extends to the remaining flagged stabilizer generator measurement circuits for the $[\![5,1,3]\!]$ code, as well as for the Steane code.

4.2. Terminology: Flag Fault Tolerance

The flag gadget for a single stabilizer measurement is used to construct fault tolerance protocols for a QEC cycle. Prior to the description of these protocols, in this section, we develop a terminology to describe various elements of flag protocols. These are also illustrated in the figure for the flag fault tolerance protocol for the $[\![5,1,3]\!]$ code (see Fig. 4.4).

- A stabilizer measurement with a flag gadget will be called a **flagged stabilizer**, and one without the flag would be called an **unflagged stabilizer**.
- When a propagated error causes a flag measurement to give a nontrivial measurement outcome, the flag will be said to have been **triggered**.
- In a flagged stabilizer measurement, if a flag has not been triggered, but the ancilla measurement gives the -1 eigenvalue, it will be referred to as a **nontrivial syndrome without flag**.
- Flag fault tolerance protocols (to follow) may require repeating stabilizer generator measurements, in order to arrive at the final syndrome suitable for decoding [25],[35]. Every time a new sequence of stabilizer generator measurements is initiated, it will be called a new **subround**.
- The protocols in [25] start with flagged stabilizer measurements, which vary between one and all generators being measured. This will be called the **first subround**. If a flag gets triggered or a nontrivial syndrome without flag is observed, protocols in [25] require all stabilizer generators to be measured once, with unflagged circuits. This will be called the **second subround**. The motivation for this terminology is that more advanced flag protocols may require further repetitions of stabilizer measurements [35]. Each of these may be numbered as consecutive subrounds.
- When the syndrome has been completely extracted using the required number of subrounds, including flagged measurements, all measurements leading up to this stage will constitute a **round**.
- Once the syndrome has been decoded and the correction applied/tracked, all operations up to this stage, starting from the first flagged measurement, will complete a **cycle**. When a new cycle begins, it will start with the first subround (i.e. flagged measurements).
- The possible sequences of operations in a single cycle of a flag protocol will be denoted as a **decision tree**. This is because the protocols require different subsequent circuits, depending on flagged stabilizer measurement outcomes, at many stages.
- The possible circuits to be applied based on a decision on the measurement outcomes will form **branches** of the tree. The unflagged stabilizers to be measured following a nontrivial flag outcome

or a nontrivial syndrome without flag will also be referred to by variables representing measurement outcomes leading to those circuits. The variables f and s represent a flag and ancilla measurement outcome, respectively. The branches will be denoted as the $\mathbf{f} = \mathbf{1}$ **branch**, when a flag getting triggered is the cause to initiate unflagged measurements, and the $(\mathbf{s}, \mathbf{f}) = (\mathbf{1}, \mathbf{0})$ **branch**, when a nontrivial syndrome without flag is the cause to initiate unflagged measurements.

4.3. Chao and Reichardt's Flag Fault Tolerance Protocols for Syndrome Extraction

Flag fault tolerance protocols for several distance-3 codes were developed by Chao and Reichardt in [25]. These have the advantage that they can achieve fault tolerance with only 2 extra qubits, which is, in general, significantly lower than traditional schemes, for the codes considered. In this thesis, a subset of the schemes developed in [25] are studied, in which the stabilizers are measured one at a time, for the $[\![5, 1, 3]\!]$ code and the Steane code. From among the traditional fault tolerance schemes (see Sec. 3.11), these flag protocols are most similar to the Shor fault tolerance scheme. This is because, in both schemes, stabilizers are measured one at a time, and measurements need to be repeated. However, there are important differences as well: in flag techniques, faults at bad locations are allowed to propagate to possible high-weight errors on data qubits, and are later corrected, while they are not allowed to propagate to more than one data qubit in Shor fault tolerance, due to transversal interactions between data qubits and the ancillary cat state.

In the protocols of [25], flagged stabilizers are measured to correct errors. If a flag is triggered or a nontrivial syndrome without flag is obtained, the flagged measurements are discontinued, and all stabilizer generators are measured with unflagged circuits. This feature, i.e. the need to actively decide what is to be measured, and which circuit is to be used to measure it, depending on a measurement outcome, is not present in the traditional fault tolerance schemes (see Sec. 3.11). Such a measurement, conditioned on a past measurement outcome, or the resulting protocol, is referred to as **adaptive** [27].

This work builds on the protocols for the $[\![5, 1, 3]\!]$ code and the Steane code developed in [25]. These are presented in the following sections, and, as an example, the protocol for the $[\![5, 1, 3]\!]$ code is analyzed for fault-tolerance with examples.

4.3.1. Flag Fault Tolerance Protocol for the $[\![5, 1, 3]\!]$ Code

The flag fault tolerance protocol for the $[\![5, 1, 3]\!]$ code [25] is shown in Fig. 4.4, which represents the decision tree for a single cycle of the protocol.

The protocol may be described as follows. A cycle of the protocol starts with the measurement of the first stabilizer generator, $XZZXI$, with a flagged circuit. If the flag gets triggered or a nontrivial syndrome without flag is observed, the flagged measurements are discontinued, and all stabilizer generators (including the first) are measured, this time with unflagged circuits. The resulting second subround syndrome can be decoded to correct the error, as explained previously. If both the flag and syndrome measurement result in trivial outcomes, the second flagged stabilizer generator, $IXZZX$, is measured. If there is a nontrivial outcome, all stabilizers are measured with unflagged circuits, and a syndrome is obtained, otherwise, the next flagged stabilizer generator is measured. The process of measuring flagged generators, and subsequently measuring all unflagged stabilizer generators on a nontrivial measurement outcome, or measuring the next flagged generator otherwise, continues and ends in either of two ways:

1. All flagged generator measurements were carried out and gave trivial outcomes. In this case, nothing is measured further, and no correction is applied. This completes the cycle.
2. A syndrome was obtained via unflagged generator measurements. This syndrome is then decoded, and the correction completes the cycle.

The syndrome, resulting from unflagged measurements, is decoded as follows.

- If a nontrivial flag outcome (regardless of ancilla measurement outcome) was the result of initiating the unflagged stabilizer measurements, the syndrome is decoded according the the look-up table corresponding to faults which trigger the flag and propagate to data qubits for the respective flagged stabilizer measurement. This will be referred to as the **flag LUT**. Thus, Table 4.2 is a flag LUT for

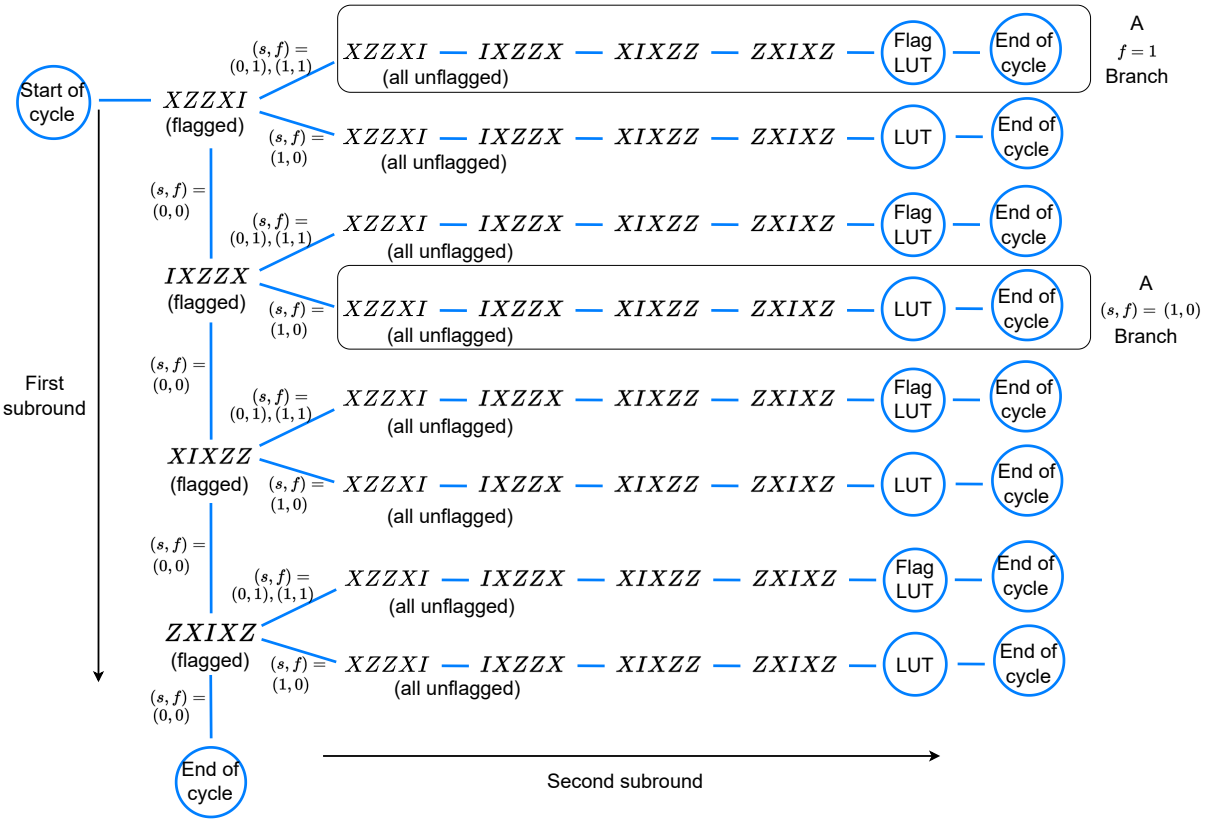

Figure 4.4: Diagram for the flag fault tolerance protocol for syndrome extraction for the $[\![5, 1, 3]\!]$ code, developed in [25]. The QEC cycle starts with the block labeled 'Start' and ends with decoding via either of the LUTs, or in the 'End' block, where no correction is applied. The diagram can be traversed left to right or bottom, and the lines denote which stabilizers to measure in the sequence. Annotations specify whether the measurement is flagged or not, and the measurement outcomes leading to specific branches of the protocol. Corrections for errors which trigger the flags may have weight greater than 1, inferred from flag LUTs. LUTs other than the flag LUTs apply the usual, weight-1 corrections. Some terminology from Sec. 4.2 is also shown.

flagged $XZZXI$ measurement. In general, the flag LUT may contain high-weight (i.e. weight-2) corrections.

- If, on the other hand, a nontrivial syndrome was the cause of initiating the unflagged measurements, and the flag was trival, the syndrome is decoded like a weight-1 input error for the $[\![5, 1, 3]\!]$ code (Table 3.1). This will be referred to as the **usual LUT**, or simply the **LUT**.

The circuits for measuring the flagged stabilizer generators $XZZXI$, $IXZZX$, $XIXZZ$, and $ZXIXZ$ are shown in Fig. 4.2 and Fig. 4.5. The resulting propagated errors on the data qubits, resulting from bad two-qubit gate faults in these circuits, which cause the flags to get triggered during each of these measurements, are listed in Table 4.1, Table C.2, Table C.3 and Table C.4.

The protocol requires only two extra qubits, which are the ancilla and the flag, because these qubits are reset after every measurement. That is, the ancilla qubit and the flag are reset after every flagged measurement, and the ancilla is reset after every unflagged measurement. This allows achieving fault-tolerance with only two extra qubits via qubit reuse. Therefore, measurement and reset should be sufficiently fast, so that more errors do not accumulate further during this time, and also because of the coherence time limit.

Decoding the syndrome for the protocol in Fig. 4.4 requires a composite LUT, because a QEC cycle can end in multiple ways in either a flag LUT or the weight-1 correction LUT. This LUT requires a composite

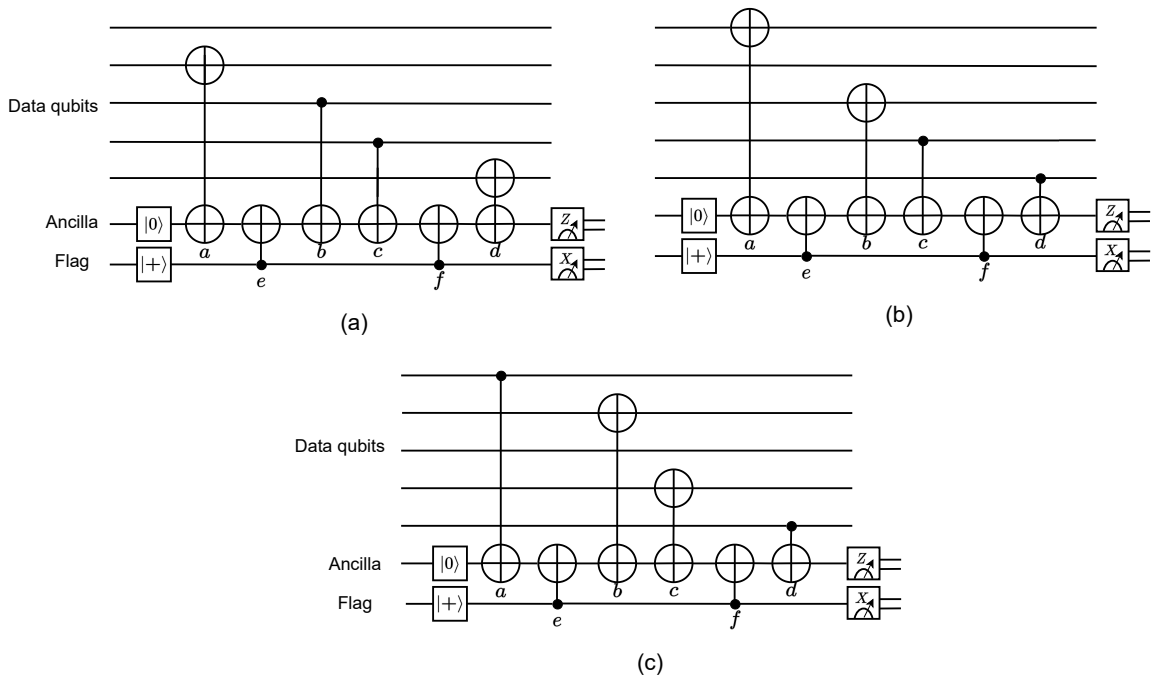

Figure 4.5: Circuits for flagged stabilizer generator measurement for the $[\![5,1,3]\!]$ code. (a): $IXZZX$, (b): $XIXZZ$ and (c): $ZXIXZ$.

syndrome consisting of possible measurement outcomes from flagged measurements and the possible measurement outcomes from subsequent unflagged measurements. Given this as input, the output of the composite LUT is a correction corresponding to the most likely error given the syndrome. Additionally, a particular choice of a minimum weight equivalent correction, up to multiplication by a stabilizer, may be used as the correction. The composite LUT for the protocol in Fig. 4.4, using circuits shown in Fig. 4.2 and Fig. 4.5, is presented in the Appendix (Table E.2).

4.3.2. Flag Fault Tolerance Protocol for the Steane Code

Fig. 4.6 shows the decision tree for a single cycle of the flag fault tolerance protocol for the Steane code, as developed in [25]. The essential idea is the same as for the protocol for the $[\![5,1,3]\!]$ code. For a single cycle of the protocol, the first subround of the protocol starts by measuring all stabilizer generators by flagged circuits. If a flag or ancilla measurement gives a nontrivial outcome, the flagged measurements are discontinued, and the second subround of measurements is carried out. This consists of measuring all stabilizer generators with unflagged circuits, including those already measured. This completes a round. The resulting syndrome is then decoded to determine the correction. Applying or keeping track of the correction completes a cycle. If a flagged measurement yields trivial measurement outcomes for both the flag and the ancilla, the next flagged stabilizer is measured, with analogous subsequent stabilizer measurements as described. If none of the flagged measurements gives a nontrivial flag or ancilla measurement outcome, the first subround gets completed, there is no second subround consisting of unflagged measurements, there is no correction applied, and the cycle ends.

Note, again, that the protocol requires only two extra qubits. The ancilla qubit and the flag are reset after every flagged measurement, and the ancilla is reset after every unflagged measurement. This allows achieving fault-tolerance with only two extra qubits via qubit reuse [25]. Therefore, the fast measurement and reset requirement also applies here.

The details which follow are specific to the protocol for the Steane code. The circuits for the 6 flagged stabilizer generator measurements are shown in Fig. 4.7. The resulting propagated errors on data qubits due to faults which trigger the flag in a flagged stabilizer measurement, along with the resulting unique and

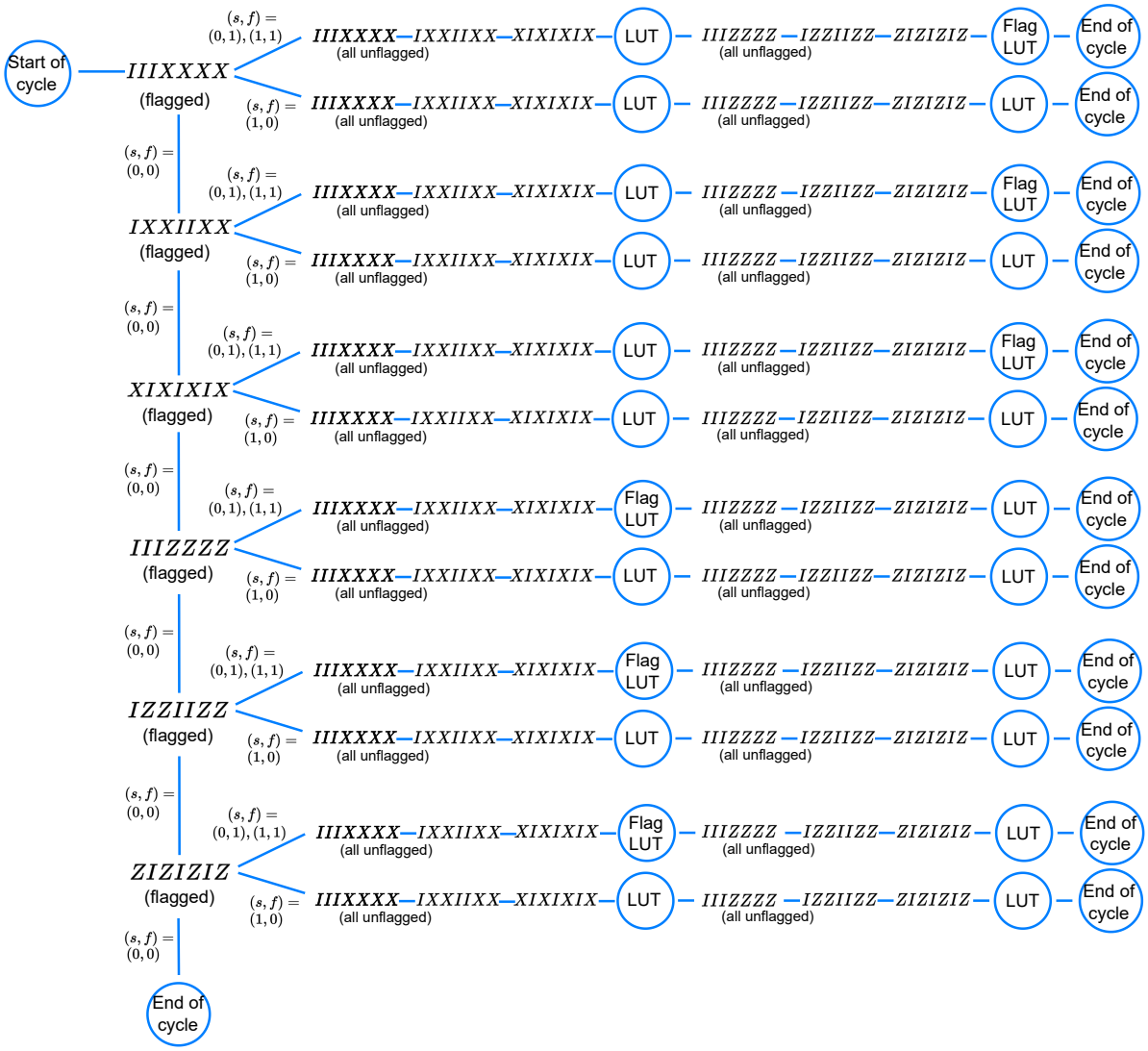

Figure 4.6: Diagram for the flag fault tolerance protocol for syndrome extraction for the Steane code, developed in [25]. The diagram may be interpreted analogously as Fig. 4.6. The X and Z errors are decoded separately.

nontrivial second subround syndromes, are presented in Table 4.3, Table C.6, Table C.7, Table C.8, Table C.9, and Table C.10.

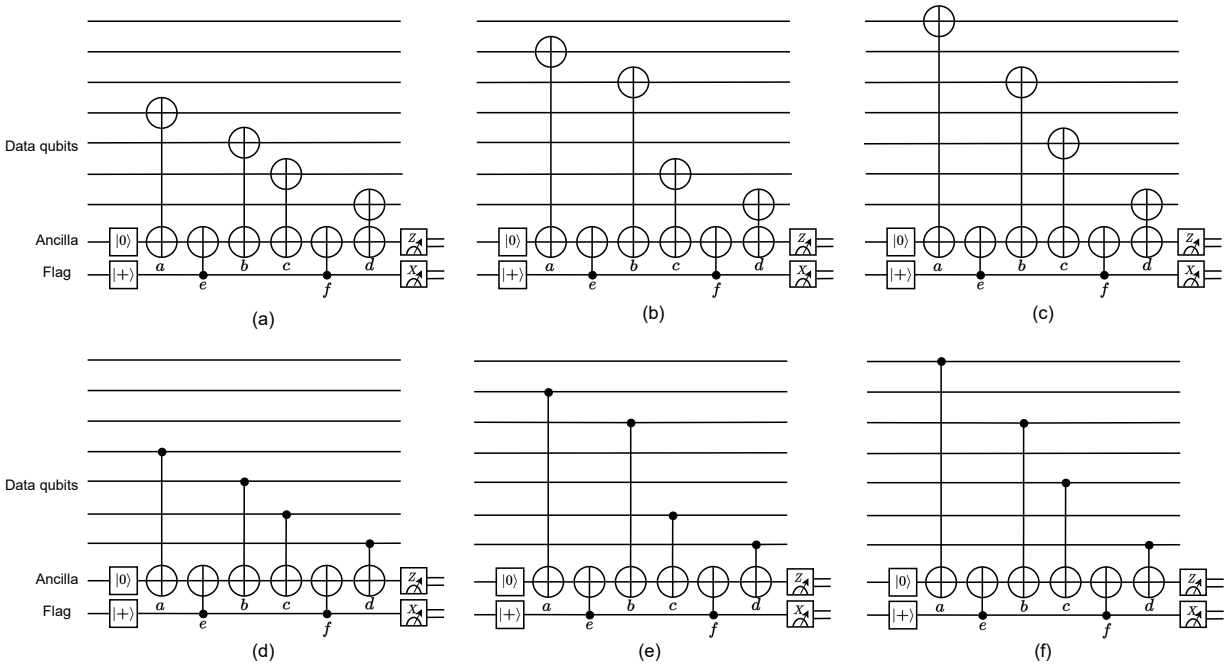

Figure 4.7: Circuits for flagged stabilizer generator measurement for the Steane code. (a): $IIIXXXX$, (b): $IXXIIXX$, (c): $XIXIXIX$, (d): $IIIZZZZ$, (e): $IZZIIZZ$, and (f): $ZIZIZIZ$. The letters below the gates are annotations to refer to specific gates.

Causal fault	Propagated error on data qubits	Syndrome from unflagged X-type stabilizer measurements	Syndrome from unflagged Z-type stabilizer measurements
IZ fault after XNOT b or XZ fault after XNOT c	$IIIIIXX$	000	001
XZ fault after XNOT b	$IIIIXXX$	000	100
YZ fault after XNOT b	$IIIIYXX$	101	100
ZZ fault after XNOT b	$IIIIZXX$	101	001
IZ fault after XNOT c	$IIIIIIX$	000	111
YZ fault after XNOT c	$IIIIIYX$	110	001
ZZ fault after XNOT c	$IIIIIZX$	110	111

Table 4.3: Errors on data qubits from bad gates during flagged $IIIXXXX$ measurement, stabilizer generator of the Steane code, using the circuit and gate annotations shown in Fig. 4.7(a). Syndromes from X- and Z- type stabilizer generator measurements are shown separately.

As mentioned before, if a flagged measurement gives a nontrivial flag or syndrome outcome, the generators are measured with unflagged circuits, in the order $IIIXXXX$, $IXXIIXX$, $XIXIXIX$, $IIIZZZZ$, $IZZIIZZ$ and $ZIZIZIZ$. This gives the syndrome for the second subround. Decoding this syndrome is more nuanced in the case of the Steane code, and it is useful to look at it in more detail. The decoding is performed separately for X-type stabilizers (which leads to Z corrections) and Z-type stabilizers (which leads to X corrections). This is true for the $f = 1$ branches, as well as the $(s, f) = (1, 0)$ branches.

Syndrome from unflagged X-type stabilizer measurements	Z correction	Syndrome from unflagged Z-type stabilizer measurements	X correction	Minimum-weight equivalent X correction	Corresponding propagated error
000	None	001	$IIIIIXX$	$IIIIIXX$	$IIIIIXX$
000	None	100	$IIIIXXX$	$IIIXIII$	$IIIIXXX$
101	$IIIIZII$	100	$IIIIXXX$	$IIIXIII$	$IIIIYXX$
101	$IIIIZII$	001	$IIIIIXX$	$IIIIIXX$	$IIIIZXX$
000	None	111	$IIIIIIX$	$IIIIIIX$	$IIIIIIX$
110	$IIIIIZI$	001	$IIIIIXX$	$IIIIIXX$	$IIIIIYX$
110	$IIIIIZI$	111	$IIIIIIX$	$IIIIIIX$	$IIIIIYX$

Table 4.4: The flag LUT for $IIIXXXX$, i.e., look-up table to decode the syndromes arising from unflagged measurements, when a flagged $IIIXXXX$ measurement results in the flag being triggered. The syndromes are the same as in Table 4.3, and are decoded separately for X and Z corrections. The Z corrections are at most weight-1, and are determined using the usual input error LUT (see Table 3.2). The possibly higher-weight X corrections are determined by inverting the syndrome-to-error map in Table 4.3. The minimum weight correction is an equivalent up to multiplication with a stabilizer. The last column represents the propagated error appearing in Table 4.3, which gives this syndrome, repeated here for reference. The corrections correspond to most likely errors, given the syndrome. As mentioned in the text, due to separate decoding of X and Z syndromes, this table can be simplified, as presented in Table E.5 and Table E.6.

There is a further observation about this protocol, which influences the LUTs required for decoding. For the faults which trigger the flag during flagged measurements for X-type stabilizer generators (see Fig. 4.7), the propagated errors on the data qubits require at most weight-1 Z corrections determined using the usual LUT (see Table 3.2), as well as corrections for possibly higher-weight X errors, determined using a flag LUT. For a flagged Z-type stabilizer generator measurement, the propagated errors require at most weight-1 X corrections, and corrections for possibly higher-weight Z errors. This trend can be observed by enumerating these errors which result from bad locations during flagged stabilizer generator measurements. These are presented in Table 4.3, Table C.6, Table C.7, Table C.8, Table C.9, and Table C.10.

The implication of this may be seen by an example: the faults occurring during a flagged stabilizer measurement of $IIIXXXX$, using the circuit shown in Fig. 4.7, consist either of errors consisting of Pauli X with weight ≥ 1, or a weight-1 Z error composed with errors consisting of Pauli X with weight ≥ 1 (see Table 4.3).The possible weight-1 Z error is corrected by decoding the syndrome from the 3 X-type unflagged stabilizer measurements using the usual LUT. The possibly higher-weight X component is corrected by decoding the syndrome from the 3 Z-type unflagged stabilizer generators and using a flag LUT. The resulting flag LUT for $IIIXXXX$ is presented in Table 4.4.

Decoding the syndromes for all possible branches requires a composite LUT, consisting of flag LUTs and weight-1 correction LUTs. Besides, the syndromes for X-type and Z-type unflagged stabilizer generator measurements are decoded separately, since the Steane code is a CSS code. This also means that, the flag LUT in Table 4.4, shown only for illustration, is implemented as separate X and Z correction LUTs. The composite Z correction and X correction LUTs for the protocol in Fig. 4.6, using circuits shown in Fig. 4.7, are presented in the Appendix (Table E.5 and Table E.6). These LUTs requires a composite syndrome consisting of possible measurement outcomes from flagged measurements and the possible measurement outcomes from subsequent unflagged measurements. Given this as input, the output of the composite LUTs is a Z or X correction corresponding to the most likely error given the syndrome. Additionally, a particular choice of a minimum weight equivalent correction, up to multiplication by a stabilizer, may be used as the correction. Both Table E.5 and Table E.6 need to be used to determine X and Z corrections separately, to correct all weight-1 input errors, or possibly high-weight errors resulting from a single fault.

4.4. Analysis of Fault Tolerance

In this section, fault tolerance of the flag fault tolerance protocols is verified by analyzing particular errors, assuming at most 1 error or fault, for the protocol for the $[\![5,1,3]\!]$ code as an example (see Fig. 4.4). The arguments will also analogously apply to the protocol for the Steane code.

In Sec. 4.1, it was discussed that propagated errors which trigger the flag can be decoded unambiguously, and that flag CNOTs do not add new bad locations. In addition to these, it is also useful to consider other possible errors.

The $[\![5,1,3]\!]$ code is a distance-3 code, hence, $t = 1$. Therefore, it is sufficient to analyze the protocol in Fig. 4.4 for fault tolerance in the case of at most 1 input error or at most 1 internal fault, based on the fault tolerance conditions (see Sec. 3.10.1). If the combined number of errors and faults exceeds 1 and the protocol is unable to correct them, this does not violate the fault tolerance conditions for a distance-3 code. This is also the reason why, if a flag gets triggered or a nontrivial syndrome without flag is obtained, it is sufficient to measure the stabilizer generators with unflagged circuits (i.e. in the second subround). This is because, such an outcome already indicates that an error or a fault has occurred (including preparation and measurement errors), and the flag qubit is no longer strictly necessary to determine if another fault (detectable by the flag) has occurred.

The behaviour of the protocol under some examples of errors, representative of different scenarios which may occur during protocol operation, is considered.

First, consider an input Z error on the 5^{th} qubit of Fig. 4.8. The first subround begins by measuring $XZZXI$ with a flagged circuit. The error commutes with this stabilizer generator, so the measurement outcomes are trivial. The protocol proceeds by measuring the second stabilizer generator, $IXZZX$, with a flagged circuit. The error anticommutes with this stabilizer, so a nontrivial syndrome outcome (without flag) is obtained, denoted by the black star in Fig. 4.8. Following the course of the protocol, this completes the first subround, because flagged measurements are discontinued. This leads to the second subround, in which all stabilizer generators are measured with unflagged circuits. This gives the syndrome 0100, which completes this round, and this can be correctly decoded by the usual (weight-1) LUT (see Table 3.1). Applying/tracking the correction completes the cycle. Such an analysis can also be carried out for other weight-1 input errors, which shows that the protocol can correct any input error on 1 qubit.

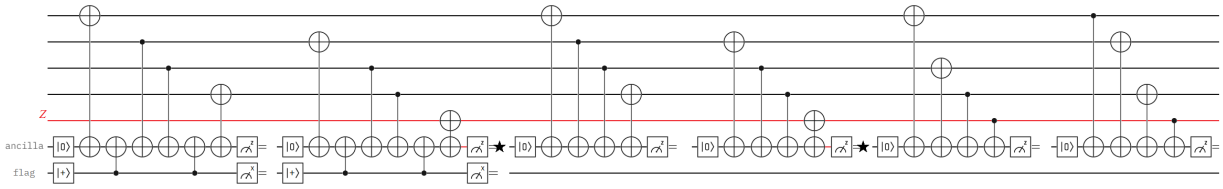

Figure 4.8: Behaviour of the protocol in Fig. 4.4 under an input Z error on the 5^{th} qubit, as described in the text. The black star denotes a nontrivial measurement outcome from the ancilla. The decoding and correction after the last measurement are not shown.

As the second example, consider a situation which, in the absence of repeated stabilizer measurements, could have led to a syndrome collision: an X error happening on the third qubit after the first flagged stabilizer generator measurement, but before the second flagged measurement. This is shown in Fig. 4.9. This error may have been caused by a two-qubit gate failure, or, by idling. This error anticommutes with the first stabilizer generator, $XZZXI$, and would have given a nontrivial syndrome from this measurement, had it occurred before it. However, it happens after it can be detected by the first measurement, so the syndrome corresponding to the first flagged measurement will be 0. Subsequently, this error gives a nontrivial syndrome from the second flagged measurement, $IXZZX$. Following the course of the protocol, the second subround (unflagged stabilizer measurements) gives a syndrome 1100, which can be correctly decoded by Table 3.1. The nontrivial syndromes are denoted by the black stars in Fig. 4.9. This highlights the use of measuring all stabilizer generators, including those already measured with flagged circuits. To unambiguously diagnose errors which can occur in between stabilizers, especially those which would have given a nontrivial syndrome for a stabilizer measured previously, this protocol [25] repeats all stabilizer measurements, including those already measured, with unflagged circuits.

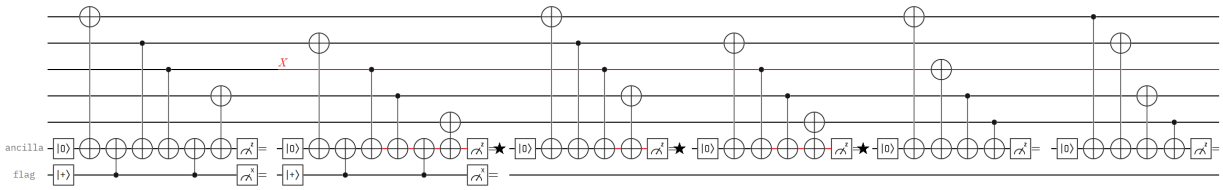

Figure 4.9: Behaviour of the protocol in Fig. 4.4 under an X error on the 3^{rd} qubit, after the first but before the second flagged stabilizer generator measurement, as described in the text. The decoding and correction after the last measurement are not shown.

The third example is a measurement error on the ancilla qubit, shown in Fig. 4.10 by the dashed blue box. Consider the second flagged stabilizer generator measurement, $IXZZX$, incorrectly giving a nontrivial syndrome, denoted by the black star in Fig. 4.10. Upon observing this outcome, the flagged measurements are discontinued, and the second subround, consisting of unflagged stabilizer generator measurements, is carried out. Under the same assumption of at most 1 error or fault, the second subround will give the trivial syndrome, 0000. This is decoded via Table 3.1, and no correction needs to be applied.

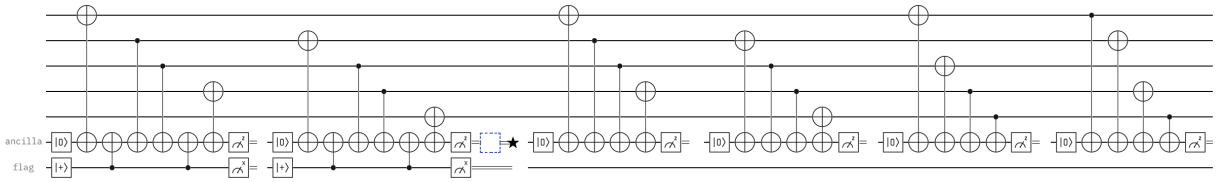

Figure 4.10: Behaviour of the protocol in Fig. 4.4 under a measurement error on the ancilla qubit during the second flagged stabilizer measurement, denoted by the dashed blue box, as described in the text.

A preparation error on the ancilla gives the same analysis, so there is no need to analyze it separately. Further, only the case of a measurement error giving the -1 outcome instead of the $+1$ outcome needs to be considered. This is because, if a measurement error gives a $+1$ outcome when the actual value is -1, it is an $\mathcal{O}(p^2)$ error, because the nontrivial outcome before the error would already have resulted from an error. Since the protocol for the $[\![5,1,3]\!]$ code does not strictly need to correct an $\mathcal{O}(p^2)$ error, this case does not need to be considered.

The fourth example is an idling error, or an error resulting from a two-qubit gate failure, on a data qubit, which does not anticommute with any (flagged) stabilizer yet to be measured. This is shown in Fig. 4.11. Consider a Z error on the 5^{th} qubit after the second flagged stabilizer generator measurement (see Fig. 4.11). This does not anticommute with the remaining stabilizer generators, $XIXZZ$ and $ZXIXZ$. This error does not propagate to other qubits during these subsequent flagged measurements (see the error propagation rules in Sec. 3.5). Therefore, the measurement outcomes in the round consisting only of flagged measurements will all be trivial, and no correction will be applied. The error persists, and is a weight-1 input error for the next QEC cycle. Therefore, it can be corrected in the same way as an input error, as discussed earlier.

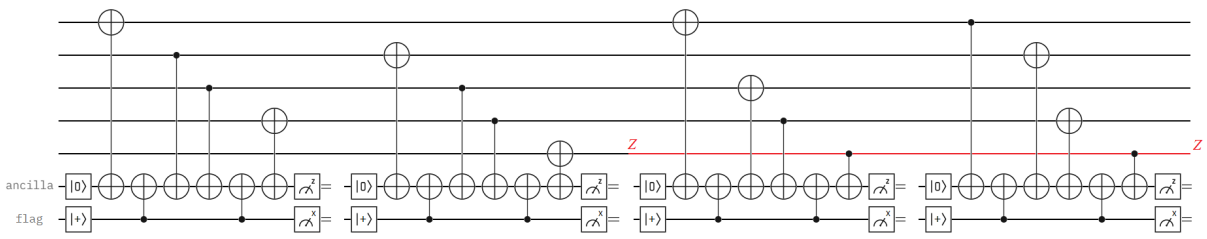

Figure 4.11: Behaviour of the protocol in Fig. 4.4 under a Z error on the 5^{th} qubit after the second flagged stabilizer generator measurement, as described in the text. This is a residual $\mathcal{O}(p)$ error for this cycle, as no correction is applied. It is corrected because it is an input error for the next QEC cycle.

Therefore, a single error on a data qubit either leads to a nontrivial syndrome without flag, or becomes a weight-1 input error for the next QEC cycle.

Now, the errors which will trigger the flag are considered. In Sec. 4.1, it is explained how faults from bad two-qubit gates in the original circuits lead to flag getting triggered, and can be corrected by measuring all stabilizer generators by unflagged circuits. As the fifth example, the course of the protocol when an XZ fault happens after CNOT b during the first flagged stabilizer measurement is shown in Fig. 4.12. The red star in the figure denotes the flag being triggered.

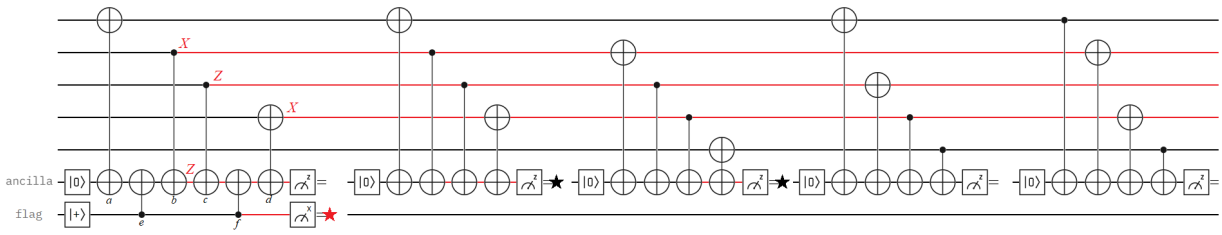

Figure 4.12: Behaviour of the protocol in Fig. 4.4 when an XZ fault happens after CNOT b during the first flagged stabilizer measurement. This triggers the flag, shown by a red star, and measuring all unflagged stabilizer generators gives the syndrome. A black star indicates nontrivial syndrome bit. Decoding and correction after the last measurement are not shown.

Subsequent unflagged stabilizer measurements give the syndrome 1100, and the nontrivial syndromes are denoted be black stars in Fig. 4.12. The flag LUT for $XZZXI$ (Table 4.2), which maps this syndrome to the correction for a propagated error due to faults after bad gates is used to decode this error. The flag LUTs for decoding such errors during the same or a different flagged measurement may be constructed by inverting the error-to-syndrome map in Tables 4.1, Table C.2, Table C.3 and Table C.4. These are incorporated in the composite LUT, presented in Table E.2.

The sixth example is a measurement error on the flag qubit during the second flagged stabilizer measurement, shown in Fig. 4.13 by the dashed blue box and red star. Like the case with a measurement error on the flag, the first subround (flagged stabilizer measurements) is stopped, and the second subround (unflagged stabilizer measurements) gives the trivial syndrome. Referring to the list of propagated errors from bad locations during the flagged second stabilizers measurement in Table C.2, no error corresponds to the trivial syndrome. Hence, no correction needs to be applied, which is consistent with the fact that there is no error on the data qubits.

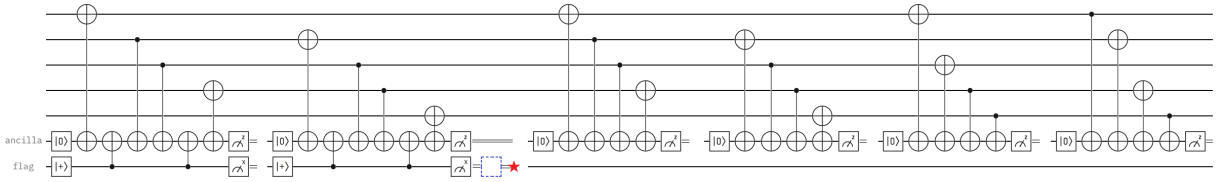

Figure 4.13: Behaviour of the protocol in Fig. 4.4 under a measurement error on the flag qubit during the second flagged stabilizer measurement, denoted by the dashed blue box, as described in the text.

This shows that if a measurement error happens on the ancilla qubit or on the flag qubit, and it is the only error to have occurred, the syndrome used for decoding will be the trivial syndrome (i.e., all zeros). This is an important consideration for the protocols developed in Sec. 5, because a different error indicated by a flagged measurement should not lead to the same syndrome as a measurement error.

Finally, it was already shown in Sec. 4.1 that a single fault after the flag CNOTs does not lead to an uncorrectable error. In addition, it can be shown that idling errors on the flag qubit either propagate to give the same set of errors on data qubits as described in 4.1, or have the same effect as a measurement error on the flag. Thus, the protocol can tolerate these faults as well.

This analysis, along with the cases analyzed in Sec. 4.1, may be extended to all errors of the type discussed here, to show that the protocol is fault-tolerant against a single input error or internal fault. It

may also be analogously extended to show that the protocol for the Steane code is fault-tolerant. A few remarks on propagated errors and decoding conclude this discussion.

A fault with a Y error on the ancilla qubit propagates to give the same error on the data qubit as the same fault with a Z on the ancilla. Fig. 4.14 shows an example. The Y error on ancilla causes both the ancilla and flag measurement outcomes to be nontrivial, and the $f = 1$ branch of the protocol needs to be followed to measure the syndromes, which is presented in Fig. 4.6. The decoding is done via the flag LUT. Therefore, these errors do not need to be analyzed separately.

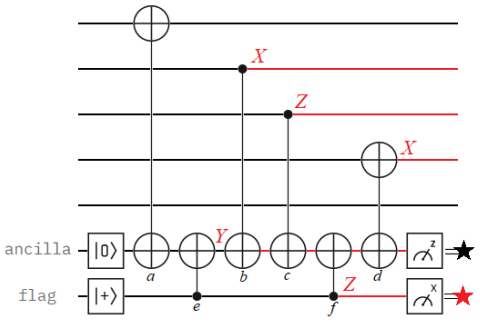

Figure 4.14: An XY fault after CNOT b during flagged measurement of $XZZXI$ gives the same error on data qubits after propagation as an XZ fault after the same gate (see Fig. 4.2).

In addition, note that the faults which cause a Z or a Y error on the ancilla are the only errors which can propagate to the flag and trigger it. Errors on data qubits, possible due to idling, don't propagate to flag, because they can at most propagate to an X error on the ancilla qubit (see the error propagation rules in Sec. 3.5). Therefore, while considering the set of errors to be corrected when a flag gets triggered, one only needs to consider faults with a Z on the ancilla after the bad two-qubit gates, and errors/measurement error on the flag qubit.

It is important to note that the set of possible propagated errors on data qubits resulting from a fault triggering the flag in a flagged stabilizer measurement depends on the order in which gates are applied between data qubits and the ancilla qubit. For example, these gates in Fig. 4.2 can be applied in a different order between the qubits, and still extract the same parity, and lead to a different error on data qubits after propagation. Since the propagated errors change, so does the corresponding flag LUT. Therefore, when describing flag protocols, it is useful to specify the order of applying the gates in a flagged measurement, with which fault tolerance is guaranteed.

Finally, decoding via the flag LUT following a nontrivial flag outcome gives the correction corresponding to the most likely error for that syndrome. It is possible that the same syndrome can result from some higher order error, however, those do not necessarily need to be corrected in order to maintain fault tolerance (for a protocol for a distance-3 code).

4.5. Outlook on Flag Protocols

This section briefly describes related flag protocols and research directions in the field. It may be skipped by readers who prefer to focus on the main content, without loss of continuity.

In [25], besides the flag protocols presented previously, other flag-based syndrome extraction circuits have also been developed. These include protocols for other distance-3 codes (in particular, quantum Hamming codes), $[\![n, n-2, 2]\!]$ error-detecting codes, protocols with multiple flag qubits to localize faults in smaller sections of the stabilizer measurement circuit, and protocols to measure more than one stabilizer generators with a shared flag qubit. In addition, flag-based circuits for fault-tolerant encoding and logical measurements have been developed in [25]. In [40], flag-based fault-tolerant logical operations have been developed. A unified reference is [39].

In [35], the flag fault-tolerant protocols have been expanded to a bigger class of distance-3 codes, and codes of arbitrary large distances. Non-adaptive flag protocols, which use a larger number of flag qubits for a flagged measurement, have been developed in [41].

In [5], flag circuits are extended incorporate qubit connectivity constraints. In [42], extensions of the flag gadget have been explored. In [36], a fault tolerance quantum error correction scheme based on flag qubits and tailored to a physically realistic noise model has been developed. In [43], a computer algorithm has been proposed to carry out X stabilizer (and Z stabilizer) measurements in parallel using a single flag qubit, for CSS codes with distance 3, to reduce depth overhead. In [44], classical error-correcting codes have been used to reduce the number of flag qubits in the protocols of [41]. These developments align with the direction of overhead reduction in flag fault tolerance schemes, pursued in this thesis.

5

Reduced Stabilizer Measurements For Flag-Based Syndrome Extraction Circuits For The $[\![5, 1, 3]\!]$ Code And The Steane Code

As discussed in the preceding section, flag fault tolerance protocols require lower overhead in terms of number of qubits and ancillary state preparation circuits, relative to traditional fault tolerance schemes [25]. At the same time, there is significant benefit in resource efficiency to be gained in exploring if quantum error correction can be carried out fault-tolerantly with as little gate overhead as possible [2].A reduction in gate overhead, if possible, may allow for more efficient use of hardware resources to correct errors, and also has the potential to lead to an increase in the pseudothreshold, because of reduction in the number of fault locations.

This forms the motivation for the main contribution of this thesis, described in this section: constructing flag qubit-based syndrome extraction circuits with reduced stabilizer measurements for flag protocols developed in [25], for the $[\![5, 1, 3]\!]$ code and the Steane code. 'Reduced stabilizer measurements' refers to a fewer number of these measurements in comparison with the original protocols. As explained in the introduction (Sec. 1), the possibility such a reduction in flag-based syndrome extraction circuits is motivated by the use of all stabilizer generators to distinguish a limited set of errors via unflagged measurements, and the limited role of flagged measurement outcomes in differentiating these errors in existing flag protocols [25].

In particular, one new protocol for the $[\![5, 1, 3]\!]$ code, and two new protocols for the Steane code, requiring fewer stabilizer measurements, are developed. These protocols achieve a very significant reduction in two-qubit gate overhead, in comparison to the original protocols [25]. These constructions are presented in Sec. 5.3, Sec. 5.4 and Sec. 5.5. The main techniques (Sec. 5.2) used to achieve this reduction are utilising operators from the complete stabilizer group of the code, especially stabilizers with weight higher than the generators, carrying out stabilizer measurements conditioned on past measurement outcomes, and constructing stabilizer sequences to distinguish errors belonging to a restricted (i.e. finite and limited) set.

These techniques are adapted from and inspired by the significantly shorter syndrome extraction sequences developed by Delfosse and Reichardt [27] for the Shor fault tolerance scheme (see Sec. 3.11). The constructions in this thesis extend these techniques to flag protocols, while taking into account their distinctive features. A review of the schemes in [27] is presented in Sec. 5.1.

The protocols with reduced stabilizer sequences in this thesis are methodically constructed to yield unique and nontrivial syndromes for the relevant error set signalled by the respective flagged stabilizer measurement. This ensures that the fundamental condition of errors being detectable and distinguishable by flag protocols, which is the principal factor for the original protocols to be fault-tolerant, is preserved. Theoretical arguments are developed to demonstrate fault tolerance of the new protocols when errors may be present at every location (Sec. 5.3, Sec. 5.4 and Sec. 5.5)

The new protocols are divided into two classes. The 'Split-and-Diagnose' protocols uniquely identify a detected error by constructing shorter sequences specifically designed to distinguish a given set of errors. Further, the stabilizers to be measured may be selected depending on measurement outcomes from previously measured stabilizers. The new protocol for the $[\![5,1,3]\!]$ code (Sec. 5.3) and one new protocol for the Steane code (Sec. 5.4) belong to this type. Therefore, these protocols lead to gate overhead reduction in unflagged measurements (i.e. the second subround). The third new protocol for the Steane code belongs to the 'Detect-and-Diagnose' class (Sec. 5.5). The main idea behind this scheme is to identify combinations of 3 high-weight stabilizers, which are fewer in number than the (6) generators of the Steane code stabilizer group, which can detect, but, by themselves, not necessarily correct for, any input error on any data qubit. Such high-weight stabilizers are used for flagged stabilizer measurements. Upon detecting a nontrivial measurement outcome, measuring all standard stabilizer generators with unflagged circuits is then used to uniquely correct such a single qubit error, or propagated errors from fault resulting from bad locations in flagged measurements. This protocol demonstrates a gate overhead reduction in flagged measurements.

In this section, after a review of stabilizer reductions presented in [27] for Shor fault tolerance, it is discussed that these techniques may not be directly applicable in the context of flag protocols, and the methodology developed in this thesis for this extentsion is presented in Sec. 5.2. This is followed by concrete constructions of the split-and-diagnose protocols for the $[\![5,1,3]\!]$ code (Sec. 5.3) and the Steane code (Sec. 5.4). Subsequent to this, the detect-and-diagnose protocol for the Steane code is presented (Sec. 5.5).

5.1. Delfosse and Reichardt's Reduced Shor-Style Stabilizer Sequences

In [27], fault-tolerant Shor quantum error correction has been optimized to reduce the number of stabilizer measurements. Fault tolerance of the schemes developed in [27] has been argued based on their capability to distinguish input errors from internal faults, and using Shor-style syndrome extraction. The basis for these reductions is to utilize stabilizers other than the standard generators, investigating how many times the stabilizers need to be measured, measuring these stabilizers in a particular order, and deciding which stabilizers to measure next based on observed syndromes.

This approach is shown to give substantial reductions in the number of stabilizers required for fault-tolerant syndrome extraction. For standard Shor error correction with a distance d code, stabilizer measurement circuits need to be repeated until the same syndrome is observed $\frac{d+1}{2}$ times, to guarantee fault tolerance. This requires at most $\left(\frac{d+1}{2}\right)^2$ repetitions of stabilizer measurements [45]. As an example, the Steane code has 6 stabilizer generators, and if each measurement needs to be repeated $\left(\frac{d+1}{2}\right)^2 = 4$ times, then 24 stabilizer measurements need to be carried out in the worst case. Using the schemes in [27], this number has been reduced to 10 with a non-adaptive scheme using only X-type and Z-type stabilizers, to 8 if Y-type stabilizers are also used, and to 7 in the best case, if high-weight stabilizers (which measure X, Y and Z Paulis on different qubits in the same operator) are used. These great reductions approximately lie in the range of 58% to 70.8%, making such investigations useful for NISQ applications.

From the schemes in [27], we only focus on the schemes for distance-3 codes, CSS codes, and specifically for the Steane code. In this section, some representative schemes are reviewed, to present the underlying principles as a reference for the new protocols, and arguments for their fault tolerance, as presented in [27], are briefly discussed.

Fig. 5.1 shows a non-adaptive stabilizer sequence for syndrome extraction using Shor error correction for a distance-3 CSS code [27]. The code is assumed to have r_Z Z-type stabilizer generators, denoted by $g_1, \ldots, g_{r_Z}$, and r_X X-type stabilizer generators. The sequence is shown only for Z-type stabilizer generators, and an analogous sequence is implied for the X-type stabilizer generators. The sequence starts by measuring all Z-type generators once, followed by measuring all except the last generator again. Importantly, this sequence only requires measuring a fixed number $(2r_Z - 1)$ Z-type generators, instead of measuring at most $4r_Z$ Z-type generators (required by Shor error correction). The resulting reduction in the number of stabilizer measurement circuits (and, by extension, of the number of gates and measurements required) by about 58% over the worst case of standard Shor error correction.

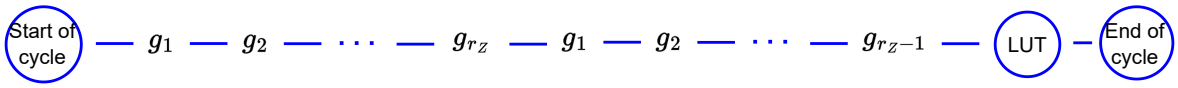

Figure 5.1: Non-adaptive stabilizer sequence for fault-tolerant syndrome syndrome extraction using Shor error correction for a distance-3 CSS code [27] The sequence is shown only for Z-type stabilizer generators $g_1, \ldots, g_{r_Z}$, and the X-type stabilizer generators need to be measured in an analogous sequence (not shown).

In [27], the schemes are argued to be fault-tolerant if there is no internal fault which gives the same syndrome as an (inequivalent) input error, to make them distinguishable. An internal fault is assumed to occur in between two stabilizer measurements. These errors, and measurement errors, are the only errors which are considered, because error propagation to multiple data qubits is assumed to be suppressed by virtue of Shor error correction, and errors on data qubits due to idling or gates can be treated as occurring in between two stabilizer measurements. Measurement errors cause at most a weight-1 correction to be applied, hence, they are not included in the analysis in [27]. It is not directly evident from [27] whether the schemes are intended to give different syndromes for inequivalent internal faults as well (refer to Sec. III B and IV A of [27]). If all input errors can be distinguished from internal faults and from each other, and no corrections are applied for syndromes which signal an internal fault not equivalent to an input error, the fault becomes an input error for the subsequent cycle. In that case, it can be corrected and the protocol is then fault-tolerant. This, however, is not elaborated upon in [27]. In the following explanation, a stabilizer sequence will be considered fault-tolerant (following the arguments in [27]) if, assuming at most 1 input error or fault between two stabilizer measurements occurs, input errors have syndromes distinct from internal faults. The syndrome now consists of all measurement outcomes.

For the particular sequence in Fig. 5.1, if a weight-1 input X error occurs, the syndrome obtained by measuring $g_1, \ldots, g_{r_Z-1}$ the first time will be the same as that obtained by measuring them the second time. If a weight-1 internal X fault occurs among the first r_Z measurements, then there are two possibilities. If the error resulting from the fault does not anticommute with any generator measured before it has occurred, its effect is equivalent to that of an input error on the same qubit, and these do not need to be distinguished. If it anticommutes with one or more generators measured before it, then the syndrome obtained by the first $r_Z - 1$ measurements is different from that obtained by the last $r_Z - 1$ measurements. Thus, an internal X fault among the first r_Z measurements can be distinguished from an input error not equivalent to it. If the fault occurs after the first r_Z measurements, the syndrome from first r_Z measurements will be trivial, and can be distinguished from inequivalent input error. It is also worth discussing why g_{r_Z} needs not be measured again. If measuring g_{r_Z} gives a nontrivial syndrome, it will again give a nontrivial syndrome, if measured again, after the $(2r_Z - 1)^{th}$ measurement. If it gives a trivial syndrome, then it means no error which anticommutes with g_{r_Z} occurred before it. Thus, the syndromes obtained from this stabilizer sequence for input errors are different from those for internal faults. Thus, this sequence is considered fault-tolerant. The same arguments can be extended to the analogous X-type stabilizer sequence for Z errors.

The next sequence, shown in Fig. 5.2, reduces the sequence of Fig. 5.1 further by introducing adaptive behaviour: deciding which stabilizers to measure based on observed syndromes. The sequence for Z-type stabilizers starts by measuring the stabilizers. If a nontrivial syndrome outcome is observed, all stabilizers are measured again, except for the one which gave the nontrivial syndrome. For example, if $g_1, g_2, \ldots g_{j-1}$ all resulted in the syndrome 0, and g_j resulted in the nontrivial outcome, then the stabilizers measured subsequent to this outcome are $g_1, g_2, \ldots g_{j-1}, g_{j+1}, g_{j+2}, \ldots g_{r_Z}$. An analogous sequence of X-type stabilizers also needs to be measured.

This sequence is fault-tolerant according to [27], because input errors (of weight up to t) can be distinguished from each other because of the code, as every generator is measured at least once. Input errors can be distinguished from internal faults, because, an internal fault which occurs before and anticommutes with g_j, and occurs after any stabilizer measurement with which it anticommutes, gives different syndromes when $g_1, \ldots g_{j-1}$ are measured before g_j and then after the nontrivial outcome from

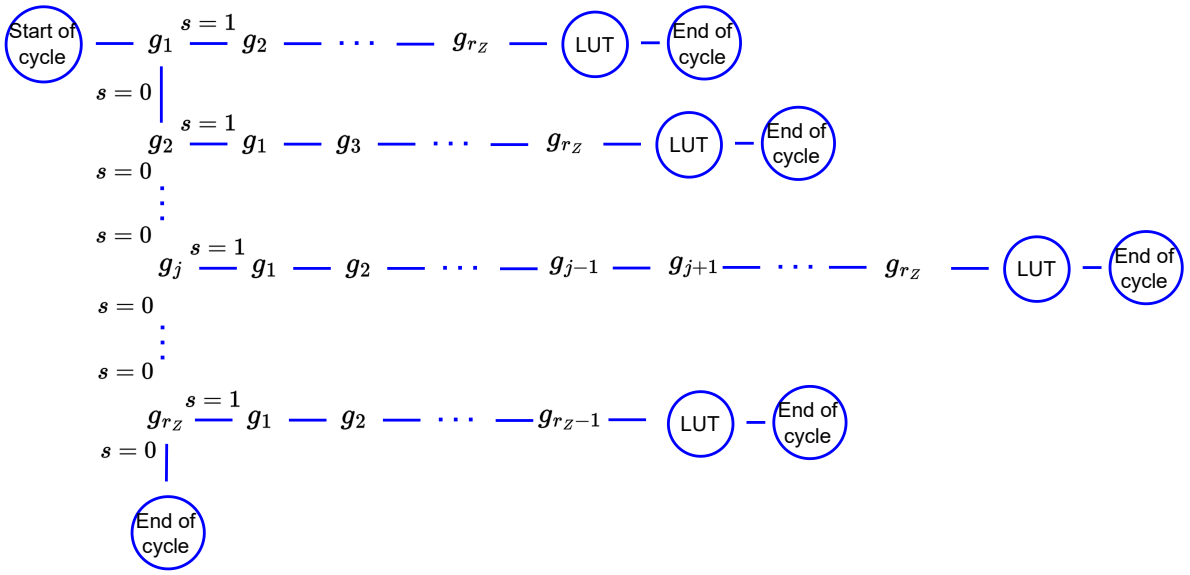

Figure 5.2: Adaptive stabilizer sequence for fault-tolerant syndrome syndrome extraction using Shor error correction for a distance-3 CSS code [27]. The values at the edges denote the syndrome bit for which the following sequence of stabilizers is to be measured. The sequence is shown only for Z-type stabilizer generators $g_1, \ldots, g_{r_Z}$, and the X-type stabilizer generators need to be measured in an analogous sequence (not shown).

g_j. In contrast, the syndromes for an input error will be the same for these two sets of measurements, and hence input errors can be distinguished from internal faults.

The sequence of Z-type stabilizers requires a minimum of r_Z and a maximum of $2r_Z - 1$ measurements, giving a variable reduction of $58\% - 75\%$ in the number of syndrome extractions over the worst case of standard Shor error correction.

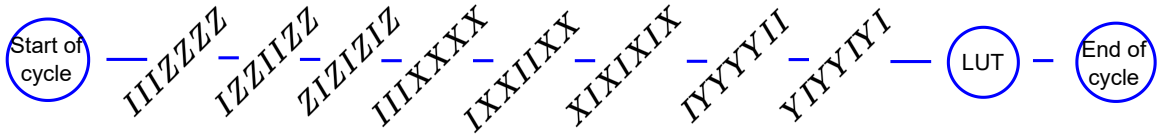

Figure 5.3: Stabilizer sequence for fault-tolerant syndrome syndrome extraction using Shor error correction for the Steane code, utilizing Y-type stabilizers [27].

In [27], it is also shown that, for CSS codes, stabilizers belonging to the complete stabilizer group and different from the standard stabilizer generators can be used to construct short syndrome extraction sequences for Shor error correction. Along these lines, the sequence shown in 5.3[27] for the Steane code measures the 3 standard Z-type generators, followed by the 3 standard X-type generators, and then measures 2 Y-type stabilizers. The Y-type stabilizers are used so that syndromes from internal faults turn out to be different from those of input errors.

Since this scheme always requires 8 stabilizers to be measured, it gives a reduction of 66.7% over the worst case of standard Shor error correction, which, as explained before, requires 24 stabilizer measurements.

The stabilizer group of the Steane code (eq. B.2) also contains operators of weight 6, which is higher

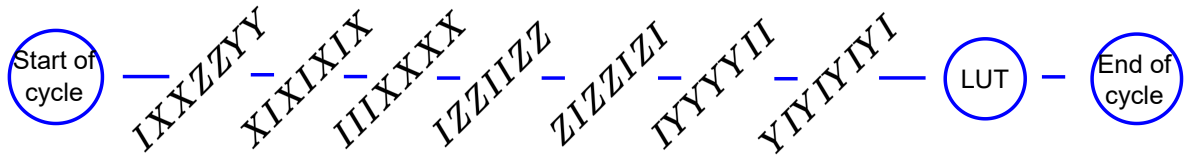

Figure 5.4: Stabilizer sequence for fault-tolerant syndrome syndrome extraction using Shor error correction for the Steane code, utilizing a stabilizer of weight higher than the generators, and Y-type stabilizers [27].

than 4, the weight of the generators. A reduced sequence is constructed in [27] (shown in Fig.5.4) by using 1 high-weight stabilizer, 2 X-type stabilizers, 2 Z-type stabilizers, along with 2 Y-type stabilizers. Using the high-weight stabilizer has the advantage that it contains 2 non-identity Paulis of each type, and has support on a larger number of qubits than the generators. Hence, it extracts a different kind of parity than the generators.The high-weight stabilizer used in Fig. 5.4 can detect a larger number of input errors earlier in the sequence as compared to a weight-4 generator, and makes it easier to distinguish input errors from internal faults by making it possible to use a shorter sequence.

The scheme shown in Fig.5.4 is the sequence requiring the fewest stabilizer measurements (7) for the Steane code, among those proposed in [27]. It gives a reduction of 70.8% in the number of measurements, over the worst case of standard Shor error correction. Since a high-weight stabilizer is used, which requires more gates than a generator, the reduction in the number of gates, though quite significant, will not be as high as the reduction in the number of measurements.

In [27], it is argued that the sequences in Fig. 5.3 and Fig. 5.4 are fault-tolerant, because the syndromes for input errors are different from those of internal faults, which may be verified by an exhaustive calculation of all syndromes due to a single input error or internal fault.

As a concluding remark, although theoretical arguments have been provided for fault-tolerance in [27], fault-tolerant pseudothresholds have not been established for the sequences in [27]. Finally, the idea to use operators from the full stabilizer group for fault-tolerant syndrome extraction has been explored in earlier work as well [46].

5.2. Approach and Methodology

The sequences developed in [27] are applicable to Shor error correction, and may not extend to flag protocols in a straightforward manner. Since Shor error correction uses transversal interactions with a cat state ancilla, there are no gates acting between different data qubits and a shared ancilla qubit (see Fig. 3.20). Consequently, it is assumed in [27] that faults after two-qubit gates do not propagate to other data qubits. In contrast, in flag protocols of [25], errors are allowed to propagate to multiple data qubits through the shared ancilla qubit in the flag protocols. These errors need to be distinguished from each other, as well as from measurement errors in flag protocols. Therefore, the criteria for fault tolerance in [27] may not apply to flag protocols.

Based on these considerations, a naïve application of the sequences in [27] to the flag protocol for the Steane code may be attempted as follows. There are at most 6 flagged measurements in the original flag protocol for the Steane code (see Fig. 4.6). However, the sequences presented in Sec. 5.1 require between 7 and 10 stabilizer measurements. If the flagged stabilizers in Fig. 4.6 are replaced by the stabilizers used in the sequences of Fig. 5.1, Fig. 5.3, or Fig. 5.4, the resulting protocol may, in general, require more stabilizer measurements than the original flag protocol. These flagged measurements may be redundant, like the Y-type stabilizers in Fig. 5.3, because they are measured in addition to the 6 standard generators. Furthermore, if using the sequence containing the high-weight stabilizer in Fig. 5.4, fault tolerance requires that the errors resulting from faults at bad locations in these stabilizers (especially for the weight-6 stabilizer) are detectable and distinguishable by subsequent unflagged stabilizer measurements. Consequently, using the sequences from [45] directly for the flagged measurements is not expected to lead to a shorter fault-tolerant protocol.

Using the sequences of [27] for the second subround (unflagged measurements) is also redundant. This is because, the function of the unflagged measurements is to obtain a syndrome to determine which error has occurred, assuming a single input error or fault has already occurred in the preceding flagged circuit. However, in addition to giving distinct syndromes for input errors, the stabilizer sequences of Sec. 5.1 are capable of distinguishing input errors from internal faults, which is, strictly speaking, is a feature not required for the second subround. Moreover, if the unflagged measurements are replaced by sequences in Fig. 5.1, Fig. 5.2, Fig. 5.3, or Fig. 5.4, it is expected to lead to protocols with a larger number of stabilizer measurements than the original flag protocol. Therefore, these sequences are not directly applicable for reducing the second subround.

While Delfosse and Reichardt's sequences may not directly lead to reduced stabilizers in flag protocol, the underlying principles can be extended to flag protocols. The approach and methods developed to achieve this are an important contribution of this thesis, and are presented in the list below.

1. **The Complete Stabilizer Group:** The first technique inspired by [27] is to employ the complete stabilizer group of the code for flag-based syndrome extraction circuits. These elements enable the extraction of a different kind of parity information from the state, as compared to the stabilizer generators. If the syndrome bit obtained by measuring a generator g_1 is s_1, and by measuring g_2 is s_2, the syndrome bit obtained by measuring the stabilizer formed by their product, i.e. $g_1 g_2$, is $s_1 + s_2$, where the addition is modulo 2. This may be seen from the binary symplectic vector representation [1]. Therefore, measuring products of generators gives the ability to extract linear combinations of parities. This extends the space of information which can be extracted via a single measurement. The existing flag protocols developed in [25] use only the standard stabilizer generators for both flagged and unflagged measurements. In addition, the complete stabilizer group offers possibilities for more diverse stabilizer sequences. The new constructions presented in this section demonstrate that it is possible to use operators from the full stabilizer group for both flagged and unflagged measurements to construct flag-based syndrome extraction circuits with reduced stabilizer measurements for the flag protocols for the $[\![5, 1, 3]\!]$ code and the Steane code. The complete stabilizer groups of these codes are presented in Eq. B.1 and Eq. B.2.
2. **Restricted Error Sets:** The second technique is based on analyzing the set of errors to be distinguished in the flag protocol, when a flag gets triggered or there is a nontrivial syndrome without the flag. These are restricted sets of errors; for example, there are 7 errors to be distinguished via a syndrome when the flag gets triggered for the flagged $XZZXI$ measurement for the $[\![5, 1, 3]\!]$ code (see Table 4.1). The constructions in this thesis demonstrate that it is not necessary to measure the entire sequence of standard stabilizer generators with unflagged circuits to distinguish errors belonging to such restricted sets. For fault tolerance, it is only required to measure a sequence of stabilizers which gives distinct syndromes for inequivalent errors in the set, with the added requirement that these syndromes are nontrivial (i.e., non-zero bit strings). Such sequences may have fewer stabilizers than the generators. In particular, we pose a problem about the efficiency of the extracted syndrome bits: given a restricted set of M inequivalent errors which are not logical errors, is it possible to distinguish them with unique and nontrivial syndromes, having the minimum length dictated by the number of bits required to represent $M + 1$ quantities classically? This lower bound of $\lceil log_2(M + 1) \rceil$ bits is placed by the Shannon information contained in these quantities, and syndrome shorter than this may not be able to distinguish these errors with nontrivial syndromes [1]. This is because, a syndrome with fewer bits would necessarily map inequivalent errors to the same syndrome, according to the pigeon hole principle. The $+1$ signifies that the trivial syndrome is always reserved for measurement errors, and is not assigned to any nontrivial error. For the sets of errors pertaining to the flag protocols for the $[\![5, 1, 3]\!]$ code and the Steane code, presented in the previous section, the constructions in this thesis show that stabilizer sequences have the potential to achieve this limit, using the first and third techniques. As a particular example, the 7 errors appearing in Table 4.1 can be distinguished using a sequence of 3 stabilizers, instead of the 4 generators.
3. **Dynamically Selected Stabilizers:** The third technique, inspired by [27], is to shorten the decision tree of a flag protocol, by selecting stabilizers to be measured based on previously observed outcomes. This is the principle behind the adaptive sequence of Fig. 5.2, in which, once it is known that there is an error, the existing sequence is discontinued, and all stabilizers (excluding the one which gave a nontrivial outcome) are measured. Such an adaptive decision-making is already present in the existing flag protocols [25], for example, when a flag gets triggered. However, in these protocols, the

adaptive measurement only takes place when stabilizer measurements need to be repeated with unflagged circuits. The constructions presented in this thesis take conventional adaptiveness beyond this assumption, in the sense that the choice of subsequent unflagged stabilizers is conditioned on past measurement outcomes. Such a dynamic stabilizer sequence more intelligently utilizes the information already extracted from the state, to measure as little as possible, while maintaining distinguishability of errors. This is the main idea behind the split-and-diagnose protocols.

4. **High-Weight Stabilizers:** Although a subset of using the complete stabilizer group, the fourth technique is inspired from the high-weight stabilizer sequence for the Steane code in Fig. 5.4. The constructions in this thesis shows that it is possible to employ high-weight stabilizers for flagged measurements, to construct a shorter protocol. This technique forms the basis of the detect-and-diagnose protocol for the Steane code (see Sec. 5.5).

The new constructions are now described in detail.

5.3. Split-and-Diagnose Protocol for the $[\![5,1,3]\!]$ Code with Reduced Unflagged Measurements

The first new flag protocol developed in this thesis is the split-and-diagnose flag protocol for the $[\![5,1,3]\!]$ code, which requires 3 stabilizers to be measured instead of 4, when a nontrivial flag measurement outcome has been observed, while still rendering the errors which trigger the flag detectable and distinguishable. The unflagged measurements in the case of a nontrivial syndrome without flag, as well as the flagged measurements, are the same as in the original protocol (Fig. 4.4). This protocol is shown in Fig. 5.5. This protocol will also be referred to as the flag protocol with $f = 1$ reduction for the $[\![5,1,3]\!]$ code.

5.3.1. Protocol Description

The protocol works as follows.

The cycle starts by measuring the flagged stabilizer $XZZXI$. If the flag has been triggered, the $f = 1$ branch is followed, in which $XZZXI$ is measured again with an unflagged circuit, followed by unflagged measurement of the stabilizer $YXXYI$, which is not a standard stabilizer generator (see Eq. B.1). The next stabilizer to be measured is decided as follows: if the $YXXYI$ measurement returned the syndrome bit 0, the stabilizer $ZIZYY$ is measured, which is also different from the standard generators, otherwise, the stabilizer $XIXZZ$ is measured. Further, the next stabilizer to be measured is decided dynamically (i.e. going beyond conventional adaptivenes), based on an already observed outcome, which underlies the nomenclature: 'split-and-diagnose'. The resulting 3-bit syndrome can be used to distinguish the errors resulting in the triggered flag, including measurement errors, presented in Sec. 5.3.2.

If, instead, a nontrivial syndrome without the flag has been observed, the 4 unflagged generators are measured, as the original protocol (see Fig. 4.4).

The stabilizer sequences for the $f = 1$ branches for the remaining flagged stabilizers follow an analogous pattern: measuring the same generator, followed by a different stabilizer, and a stabilizer which depends on the previous measured value, all with unflagged circuits. Stabilizer sequences to be measured when a nontrivial syndrome is observed without a flag are unchanged from Fig. 4.4. Note that, like the flag protocols presented previously, the ancilla qubit and the flag are reset after every flagged measurement, and the ancilla is reset after every unflagged measurement. The cycle ends in either a decoding step, either by a flag LUT or the input errors LUT, and applying a correction, or by applying no correction if all flagged measurements gave trivial outcomes. The decoding of syndromes is performed unambiguously by a composite LUT, presented in Table E.7.

5.3.2. Construction of the Stabilizer Sequences

The basis for this reduction and choice of stabilizers in the sequence is explained in this section.

The propagated errors from bad locations in flagged stabilizer measurement circuits are the same as for the original protocol in Fig. 4.4, because they employ the same circuits as in Fig. 4.2 and Fig. 4.5. For example, the propagated errors from bad gates which cause the flag to get triggered for the flagged $XZZXI$ measurement, and occur with probability $\mathcal{O}(p)$, presented in Table 4.1, are presented in Eq. 5.1.

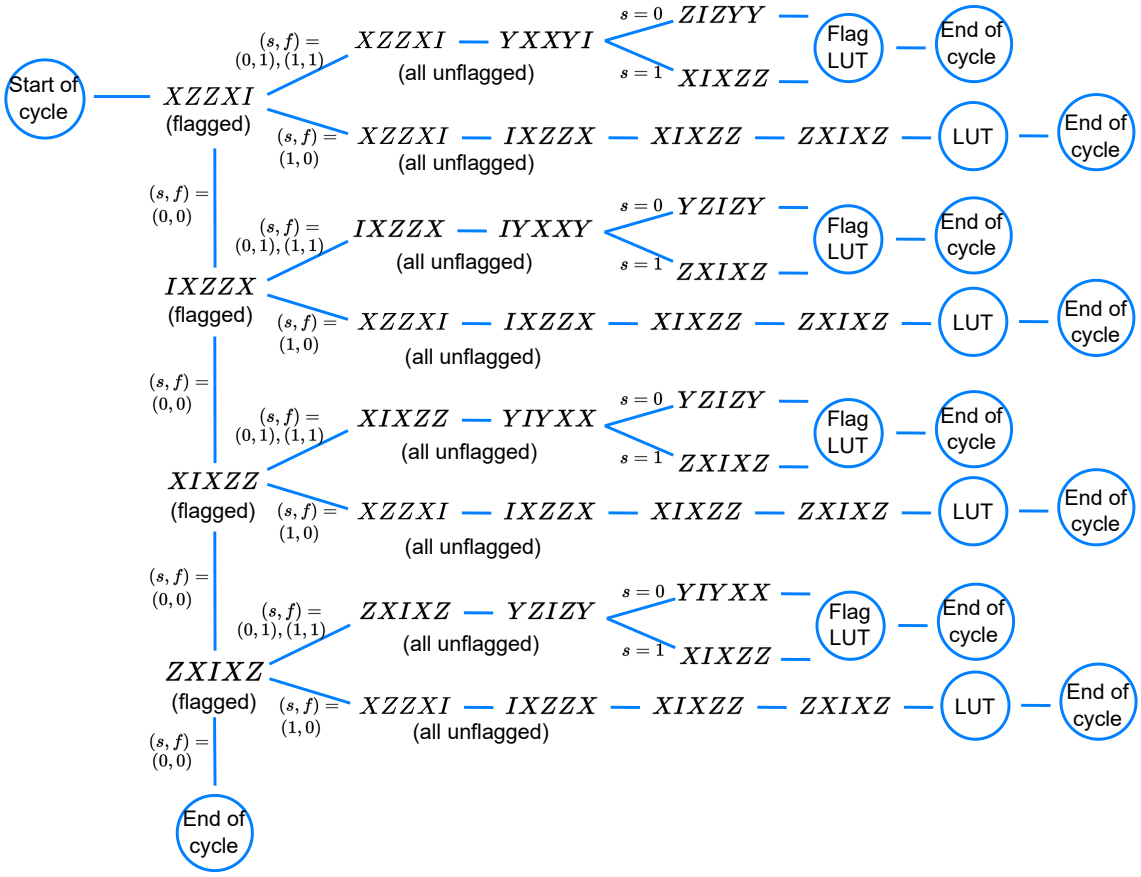

Figure 5.5: The split-and-diagnose flag protocol for the $[\![5,1,3]\!]$ code. This protocol requires 3 unflagged stabilizer measurements, instead of the 4 required by the original protocol of Fig. 4.4, when a flag gets triggered.

$$IIZXI, IXZXI, IYZXI, IZZXI, IIIXI, IIXXI, IIYXI. \tag{5.1}$$

These form a restricted or a finite set of 7 errors. The original flag protocol (Fig. 4.4) requires measuring the 4 stabilizer generators to distinguish these errors. However, a 4-bit syndrome has the capability to represent up to $2^4 = 16$ different errors (including the no error case). This indicates a possibility for reduction in syndrome bits, because some 4-bit syndrome strings may be unutilized, and motivates constructing a stabilizer sequence consisting of $\lceil log_2(7+1) \rceil = 3$ stabilizers. Such a sequence provides 8 possible syndromes, so that the 7 nontrivial syndromes can be mapped to each one of the errors in Eq. 5.1, and the trivial syndrome to a flag measurement error. Thus, these constructions are motivated by making measurements which are sufficient to distinguish a restricted set of errors.

To construct this sequence, some observations about the errors are employed. The support of the errors in Eq. 5.1 on qubits 2 and 3 is highlighted in red, because these are the only two qubits where the support of these operators varies. In the following discussion, these qubits will be called the **deciding qubits**. The operators acting on the other qubits are observed to be the same for all these strings (Eq. 5.1). Therefore, the key factor in selecting stabilizers to distinguish these errors is the Pauli operators acting on the deciding qubits, because this is the only information that can be used to distinguish these errors.

For the sake of argument, as an example, if 2 operators, both of which measure XI on the deciding qubits, are measured, both of them would given the same information about the errors (up to the actual syndrome bit). It would be redundant to measure one if the other has been measured. As another example, an operator which measures II on the deciding qubits, would give the same syndrome for all errors in Eq. 5.1, and it would be of no use to measure it.

Therefore, to achieve a sequence with the number of stabilizers as close as possible to a number logarithmic in the number of errors, it is useful to choose stabilizers which partition the set of errors into nearly half at each step. The classical algorithm for binary search of sorted items also motivates this approach [1]. Therefore, the first (unflagged) stabilizer to distinguish the errors in Eq. 5.1 is chosen to be the same stabilizer ($XZZXI$) as the flagged measurement, because it has support on both the deciding qubits (3 and 4). Further, the resulting syndromes split the set in Eq. 5.1 into two nearly equal halves: a set of 3 errors (corresponding to syndrome bit 0), shown in rows $1, 4, 5$, column 2 in Table 5.1, and 4 errors (corresponding to syndrome bit 1), shown in rows $2, 3, 6, 7$, column 2 in Table 5.1.

Propagated error on data qubits	Syndrome from $XZZXI$ measurement	Syndrome from $YXXYI$ measurement	Syndrome from third stabilizer measurement
$IIZXI$	0	0	$(ZIZYY) : 1$
$IXZXI$	1	0	$(ZIZYY) : 1$
$IYZXI$	1	1	$(XIXZZ) : 0$
$IZZXI$	0	1	$(XIXZZ) : 0$
$IIIXI$	0	1	$(XIXZZ) : 1$
$IIXXI$	1	1	$(XIXZZ) : 1$
$IYXXI$	1	0	$(ZIZYY) : 0$

Table 5.1: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $XZZXI$ measurement (see Table 4.1), using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the $[\![5,1,3]\!]$ code in Fig. 5.5. Pauli operators acting on the deciding qubits have been shown in red.

The second stabilizer is chosen to be $YXXYI$, because it extracts a different kind of parity information (XX, as opposed to ZZ from the previous stabilizer). With reference to Table 5.1, column 3, this measurement partitions the set of errors which resulted in syndrome bit 0 from $XZZXI$ into two sets, consisting of 1 error (row 1) and 2 errors (rows $4, 5$) respectively, and, at the same time, it partitions the set of errors which resulted in syndrome bit 1 from $XZZXI$ into two sets consisting of 2 errors each (rows $2, 7$ and rows $3, 6$).

The third stabilizer in the sequence is chosen to extract yet another type of parity, not measured yet. This stabilizer has support on only one of the deciding qubits. Further, this stabilizer depends on the measurement outcome from $YXXYI$ (in the third column of Table 5.1): if the outcome is 0, $ZIZYY$ is measured (see rows $1, 2, 7$, column 3), otherwise, $XIXZZ$ is measured (see rows $3, 4, 5, 6$, column 3). This choice also splits the errors having same syndromes from the previous measurement into two nearly equally sized sets (rows $1, 2, 5, 6$, and rows $3, 4, 7$, column 4). The dynamic choice of the stabilizer based on a previous measurement outcome lifts the restriction that the same stabilizer is expected to efficiently partition all errors with same syndrome suffixes into nearly equal halves, and enables shorter stabilizer sequences. The resulting syndromes (see Table 5.1) are unique and nontrivial for the errors in Eq. 5.1, and thus, this stabilizer sequence renders these errors detectable and distinguishable.

Analogous stabilizer sequences are constructed for errors which cause the flag to get triggered for the remaining flagged measurements, and give unique and nontrivial syndromes: for the $f = 1$ branches, the first stabilizer is chosen as the same generator with an unflagged circuit, because it has support on both deciding qubits. The second stabilizer is chosen to be different from a generator, and has support on both deciding qubits. The third stabilizer to be measured has support on only one deciding qubit, depending on the previous outcome. The resulting syndromes are unique and nontrivial, and are presented in Tables D.2, D.3, D.4. The composite LUT for syndrome decoding for the complete protocol (Table E.7) is constructed using Tables 5.1, D.2, D.3, D.4.

5.3.3. Analysis of Fault Tolerance

This section presents an analysis of the error-correction capabilities of the protocol in Fig. 5.7.

It is sufficient to analyze the correction of errors which trigger the flag in a flagged measurement. This is because, the syndrome extraction circuits in this protocol are the same for the errors which yield a nontrivial syndrome without the flag as the original protocol (see Fig. 4.4). Therefore, the new protocol inherits error correction capability for these errors from the existing protocol. These may be weight-1 input errors, weight-1 errors on data qubits occurring in between stabilizer measurements (due to idling or gate faults), or measurement errors on the ancilla during a flagged measurement. In addition, the CNOT gates between the ancilla and the flag do not add bad locations, as explained before in Sec. 4.1.

The proposed reduction in $f = 1$ branches is sufficient to distinguish errors which trigger the flag because of the following reasons. Crucially, the errors presented in Tables 4.1, C.2, C.3 and C.4 are the only inequivalent errors which are indicated by a triggered flag in the respective flagged measurement (and need to be distinguished) for this protocol, although they may arise from different fault locations [25][39]. Although the analysis is based on Knill's error model as in [25], it is expected to apply upon inclusion of idling errors as well, since these error sets remain the same (see Sec. 4 and [25]).

For this analysis, recall that the purpose of the second subround (i.e. flagged measurements) in the protocol in Fig. 4.4 is to yield unique syndromes for inequivalent errors indicated by a triggered flag. This is the (first) requirement that the errors are distinguishable. This is also true for the reduced sequences used in the protocol in Fig. 5.5, as seen from Tables 5.1, D.2, D.3, D.4: the sequences have been designed so that the resulting syndromes are distinct for different errors. This is required for fault tolerance, so that syndrome collision is avoided, and the the ($\mathcal{O}(p)$) error can be unambiguously identified. The stabilizer sequences used in the new protocol of Fig. 5.5 for the $f = 1$ branches preserve this property, while being shorter than the sequence of 4 stabilizer generators used in the original protocol (Fig. 4.4), and using stabilizers of the same weight.

The second requirement from the second subround measurements is that these errors, which are not logical errors (see Sec. 4), are detectable. The stabilizer sequence used to distinguish them should not give the trivial syndrome for any of these errors. This is because, if, instead, a measurement error is the cause of the flag getting triggered, the subsequent syndromes would be all zeros, irrespective of the stabilizers used in the sequence (assuming at most 1 error or fault). If an error indicated by a triggered flag also gives the trivial syndrome, it may not be distinguishable from a measurement error on the flag, and the resulting error correction rules will not be fault-tolerant. The requirement of nontrivial syndromes is also satisfied by the stabilizer sequences used in the $f = 1$ branches in the new protocol of Fig. 5.5, as seen from Tables 5.1, D.2, D.3, D.4. Therefore, in the new protocol, measurement errors can be identified by the trivial syndrome in the second subround.

Therefore, the rigorously derived LUT (see Table E.7), using Tables 5.1, D.2, D.3, D.4, demonstrates that the derived stabilizer sequences can correct the errors which trigger the flags, due to the resulting unique and non-trivial syndromes.

Furthermore, as presented in Sec. 4.4, a single error on a data qubit, possibly due to idling, either leads to a nontrivial syndrome without flag, or becomes a weight-1 input error for the next error correction cycle. These errors are also present in the Knill's error model, due to two-qubit gate depolarizing errors. Since the flagged measurements, and the unflagged measurements in the split-and-diagnose protocol (Fig. 5.5) used to correct them are the same as those in the original protocol (Fig. 4.4), the new protocol can correct these errors as well. This analysis demonstrates the fault tolerance of the split-and-diagnose protocol in Fig. 5.5, under the errors discussed here.

5.3.4. Overhead Reduction

Given that a flag has been triggered, the protocol in Fig. 5.5 requires 3 stabilizer measurements instead of 4, which is a reduction of 25% for the $f = 1$ branches. Assuming that gates do not need to be decomposed further, in line with [25],[35], a sequence of 3 unflagged stabilizers requires $3 \times 4 = 12$ two-qubit gates, while 4 unflagged generators require $4 \times 4 = 16$ two-qubit gates, which is also a reduction of 25%. This assumes circuits of the form shown in Fig. 3.3 for individual stabilizer measurements.

If one views the second subround (unflagged measurements) collectively, then the number of measurements and two qubit gates in the $f = 1$ branches goes down from 4 to 3, while this number is unchanged (4) for the $(s, f) = (1, 0)$ branches. Therefore, at an average, the number of stabilizer measurements gets reduced by 12.5% in the second subround.

When viewed as a whole, the decision tree of Fig. 5.5 requires a minimum of 4 measurements, which occurs for the $f = 1$ branch after the flagged $XZZXI$ measurement. The minimum number of measurements for the original protocol (see Fig. 4.4) is 5, which occurs in the same scenario. Therefore, the new protocol gives a reduction of 20% on this metric. The maximum number of measurements required for both protocols is 8, which occurs when a nontrivial syndrome without flag is obtained from the flagged $ZXIXZ$ measurement. Therefore, there is no reduction in this number.

5.3.5. Miscellaneous Remarks

It is worth pointing out that an analogous attempt to reduce the stabilizer sequences by analyzing the errors for the $(s, f) = (1, 0)$ branches may not lead to shorter sequences. For instance, the weight-1 errors which can cause the outcome of a nontrivial syndrome without flag from the flagged $XZZXI$ measurement, based on anticommutation relations, and regardless of the location, are

$$ZIIII, YIIII, IXIII, IYIII, IIXII, IIYII, IIIZI, IIIYI, \tag{5.2}$$

in addition to a measurement error. These are a total of 9 quantities to be distinguished, which requires $\lceil log_2(9) \rceil = 4$ syndrome bits. The sequence used in the original protocol (see Fig. 4.4) already uses 4 generators, and a further reduction in this number, based on this approach, may not be possible.

5.4. Split-and-Diagnose Protocol for the Steane Code with Reduced Unflagged Measurements

The second new protocol developed in this thesis is the flag protocol for the Steane code, with more extensive reductions in the number of unflagged stabilizers over the original protocol [25] (see Fig. 4.6), as compared to the reduction offered by the new protocol for the $[\![5, 1, 3]\!]$ code (see Fig. 5.5). This protocol for the Steane code requires 3 unflagged stabilizers to be measured instead of 6, when a nontrivial flag measurement outcome has been observed. In addition, it requires fewer unflagged stabilizers to be measured for the cases when a nontrivial syndrome without flag is observed, down to 4 from 6. These reduced stabilizer sequences are capable of distinguishing $\mathcal{O}(p)$ errors during flagged measurements leading to these branches. This protocol is shown in Fig. 5.6. This protocol will also be referred to as the flag protocol with $f = 1$ and $s = 1$ reduction for the Steane code.

5.4.1. Protocol Description

The cycle begins with flagged measurements of the 6 standard stabilizer generators. If the first flagged measurement $IIIXXXX$ gives a nontrivial flag outcome, a sequence of three unflagged stabilizers is

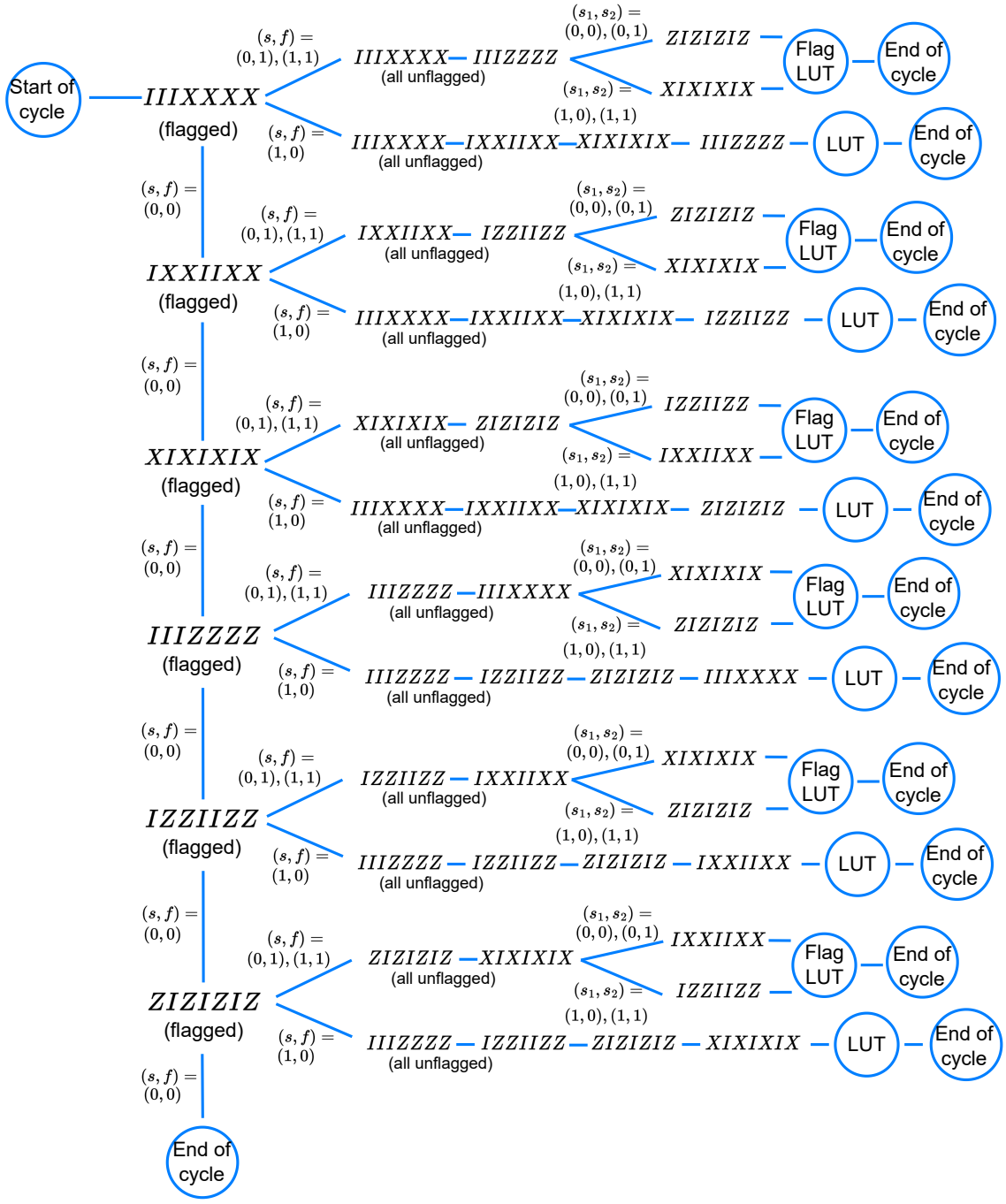

Figure 5.6: The split-and-diagnose flag protocol for the Steane code. This protocol requires 3 unflagged stabilizer measurements, instead of the 6 required by the original protocol of Fig. 4.4, when a flag gets triggered, and 4 unflagged measurements instead of 6 when a nontrivial syndrome without flag is observed. The measurement outcomes of the first two unflagged measurements in the $f = 1$ branches are denoted by the variables (s_1, s_2).

measured in the $f = 1$ branch to correct the possible ($\mathcal{O}(p)$) error which would have caused this outcome. The sequence consists of measuring the same generator again with an unflagged circuit, analogous to the protocol in Fig. 5.5. This is followed by the corresponding conjugate Z-type generator $IIIZZZZ$. The third stabilizer is chosen to be $ZIZIZIZ$ if the unflagged $IIIXXXX$ measurement gave a 0, otherwise, the stabilizer $XIXIXIX$ is measured. This adaptive measurement depends on the measurement outcome of the first unflagged stabilizer, which is a difference from the protocol in Fig. 5.5, where it is based on the second measurement. In Fig. 5.6, the measurement outcomes of the first two unflagged measurements in the $f = 1$ branches are denoted by the variables (s_1, s_2).

As opposed to conventional CSS decoding where the X and Z corrections are applied separately even for the $f = 1$ branches (see Fig. 4.6), there is a single flag LUT used for decoding the syndrome resulting from the $f = 1$ branch of the flagged $IIIXXXX$ measurement. This LUT contains a minimum number of corrections needed to unambiguously correct these errors, to achieve a reduction in the number of stabilizer measurements.

If, instead, the flagged $IIIXXXX$ measurement results in a nontrivial syndrome without the flag, the $(s, f) = (1, 0)$ branch is followed, and this consists of the 3 unflagged X-type generators, and one Z-type unflagged generator $IIIZZZZ$, which is the conjugate to $IIIXXXX$. The resulting 4-bit syndrome is decoded via an LUT which applies weight-1 corrections, incorporated in a composite LUT (see Table E.8 and E.9). However, the syndrome-to-error mapping in this LUT is different from the ones for input errors (which are presented in Table E.3 and Table E.4).

If the flagged $IIIXXXX$ measurement results in trivial flag and syndrome, the next flagged generator, $IXXIIXX$ is measured. Analogous stabilizer sequences are measur,ed for the branches following this and the remaining flagged measurements, depending on the flagged measurement outcomes. Note that, like the flag protocols presented previously, the ancilla qubit and the flag are reset after every flagged measurement, and the ancilla is reset after every unflagged measurement. The cycle ends in either a decoding step, either by a flag LUT or a weight-1 correction LUT, and applying a correction, or by applying no correction if all flagged measurements gave trivial outcomes. The decoding of syndromes is performed unabiguously via a composite LUT, presented in Table E.8 and Table E.9.

5.4.2. Construction of the Stabilizer Sequences

This section discusses the basis for the stabilizer reductions in the protocol in Fig. 5.6.

The sequences for the $f = 1$ branches are constructed analogous to the previous protocol in Fig. 5.5, and will only be described briefly. The propagated errors from bad locations in flagged stabilizer measurements are the same as for the protocol in Fig. 4.6, because they employ the same circuits as in Fig. 4.7. As an example, these errors due to faults which trigger the flag in a flagged $IIIXXXX$ measurement (see Table 4.3), are presented in Eq. 5.3, with the deciding qubits highlighted in red:

$$IIII\color{red}{IX}X, IIII\color{red}{XX}X, IIII\color{red}{YX}X, IIII\color{red}{ZX}X, IIII\color{red}{II}X, IIII\color{red}{IY}X, IIII\color{red}{IZ}X. \tag{5.3}$$

These are a restricted set of 7 errors, and, a 6-bit syndrome, capable of distinguishing up to $2^6 = 64$ errors, as per the original protocol in Fig. 4.6, may be inefficient to distinguish these. This motivates constructing a stabilizer sequence consisting of $\lceil log_2(7 + 1) \rceil = 3$ stabilizers.

Analogous to the previous protocol (Fig. 5.5), the unflagged stabilizers for $f = 1$ branches are chosen according to their support on the deciding qubits, and in a manner that a stabilizer measurement (not necessarily with the same stabilizer) divides the errors having the same syndrome from the previous measurement into two nearly equal subsets. It is observed, for example, that the same stabilizer as the flagged measurement, $IIIXXXX$, has support on both deciding qubits for the errors in Eq. 5.3 (and this also holds for other flagged measurements, see Tables 4.3, C.6, C.7, C.8, C.9 and C.10). Thus, $IIIXXXX$ is chosen as the first stabilizer in the $f = 1$ sequence resulting from the flagged $IIIXXXX$ measurement. A convenient choice for the second stabilizer is the corresponding conjugate Z-type stabilizer $IIIZZZZ$, because it also has support on the deciding qubits, and extracts a different kind of parity. The third stabilizer is chosen such that it has support on only one of the deciding qubits. If the unflagged $IIIXXXX$ measurement gave a 0, $ZIZIZIZ$ is measured, otherwise, $XIXIXIX$ is measured. The resulting syndromes are unique and nontrivial (see Table 5.2, which may be read in the same manner as Table 5.1, explained before).

Analogous stabilizer sequences are constructed for $f = 1$ branches of the remaining flagged measurements, and lead to unique and nontrivial syndromes for the relevant set of errors, as shown in Tables D.6, D.7,D.8,D.9, and D.10. These are used to form the composite LUT presented in Table E.8 and Table E.9, and are used to unambiguously correct these errors.

Propagated error on data qubits	Syndrome from $IIIXXXX$ measurement	Syndrome from $IIIZZZZ$ measurement	Syndrome from third stabilizer measurement
$IIIIIXX$	0	0	$(ZIZIZIZ) : 1$
$IIIIXXX$	0	1	$(ZIZIZIZ) : 0$
$IIIIYXX$	1	1	$(XIXIXIX) : 1$
$IIIIZXX$	1	0	$(XIXIXIX) : 1$
$IIIIIIX$	0	1	$(ZIZIZIZ) : 1$
$IIIIIYX$	1	0	$(XIXIXIX) : 0$
$IIIIIZX$	1	1	$(XIXIXIX) : 0$

Table 5.2: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $IIIXXXX$ measurement (see Table 4.3) using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the Steane code in Fig. 5.6. The Pauli operators acting on the deciding qubits have been shown in red.

In addition to the stabilizer count reductions in $f = 1$ branches, it is also possible to shorten the $(s, f) = (1, 0)$ branches, i.e. when the flagged measurement gives nontrivial syndrome without flag, in the protocol for the Steane code (Fig. 4.6). For example, the $\mathcal{O}(p)$ errors, which can cause the $(s, f) = (1, 0)$ outcome from a flagged $IIIXXXX$ measurement (in addition to a measurement error) are presented in Eq. 5.4:

$$IIIZIII, IIIIZII, IIIIIZI, IIIIIIZ, IIIYIII, IIIIYII, IIIIIYI, \text{ and } IIIIIIY. \tag{5.4}$$

Independent of whether these errors occur due to idling or gate faults, these are the only $\mathcal{O}(p)$ errors on data qubits which can cause this outcome, by anticommutation relations. Including the measurement error, these are 9 errors to be distinguished, which suggests the shortest possible sequence of stabilizers may have size $\lceil log_2(8 + 1) \rceil = 4$. This sequence is constructed as follows: first, all 3 X-type stabilizers are measured to determine the Z component of the errors in Eq. 5.4. These 3 measurements locate the qubit on which the error has occurred. Subsequently, to determine whether the error contains a Z or a Y operator (because these occur at the same locations, see Eq. 5.4), the conjugate Z-type stabilizer corresponding to the flagged stabilizer (here, $IIIZZZZ$) is measured. The resulting syndromes, presented in Table 5.3, are unique and nontrivial.

Note that, although measuring $IIIXXXX$ again gives the same syndrome for all errors (bit 1), as presented in Table 5.3, it is measured again to distinguish these errors from a measurement error. This is because a measurement error which gives a nontrivial syndrome without flag for the flagged $IIIXXXX$ measurement will give a trivial syndrome when measuring $IIIXXXX$ again with an unflagged circuit.

The same analysis can be applied to construct reduced stabilizer sequences for the $(s, f) = (1, 0)$ branches of the remaining flagged measurements, and the syndromes are presented in Tables D.12, D.13, D.14, D.15, and D.16. The corrections applied due to these syndromes are always weight-1. These are used to form the composite LUT presented in Table E.8 and Table E.9, and correct the relevant error set unambiguously, by virtue of the unique and nontrivial syndromes.

5.4.3. Analysis of Fault Tolerance

This section presents an analysis of the error-correction capabilities of the split-and-diagnose protocol in Fig. 5.6.

Similar to the arguments for the new protocol for the $[\![5, 1, 3]\!]$ code, it is discussed here that the reduced unflagged stabilizer sequences for the protocol in Fig. 5.6 render the errors signalled by flagged measurements detectable and distinguishable.

Error	Syndrome from $IIIXXXX$ measurement	Syndrome from $IXXIIXX$ measurement	Syndrome from $XIXIXIX$ measurement	Syndrome from $IIIZZZZ$ measurement
$IIIZIII$	1	0	0	0
$IIIIZII$	1	0	1	0
$IIIIIZI$	1	1	0	0
$IIIIIIZ$	1	1	1	0
$IIIYIII$	1	0	0	1
$IIIIYII$	1	0	1	1
$IIIIIYI$	1	1	0	1
$IIIIIIY$	1	1	1	1

Table 5.3: Unique and nontrivial syndromes computed using the reduced stabilizer sequence used in the split-and-diagnose protocol in Fig. 5.6 for errors which give a nontrivial syndrome without flag from the flagged $IIIXXXX$ measurement, and are diagnosed by the $(s, f) = (1, 0)$ branch.

First, the propagated errors which lead to a flag getting triggered are considered, distinguished by the $f = 1$ branches of the protocol. Tables 5.2, D.6, D.7, D.8, D.9, and D.10 demonstrate that the second subround syndromes obtained for these errors for all $f = 1$ stabilizer sequences used in Fig. 5.6 are unique and nontrivial. Crucially, the errors presented in these tables are the only inequivalent errors which are indicated by a triggered flag in the respective flagged measurement (and need to be distinguished) for this protocol, although they may arise from different faults [25],[39]. Since the syndromes are unique, the relevant errors can be detected and distinguished unambiguously. Since they are nontrivial, measurement errors on the flag can be identified by the trivial syndrome in the second subround. Therefore, these reduced stabilizer sequences can unambiguously correct the errors from these sets.

Secondly, the same argument is used to show that the protocol can correct weight-1 errors which lead to nontrivial syndrome without flag from a flagged measurement. These may be input errors, or weight-1 data qubit errors which occur between stabilizer measurements due to idling or gate faults. Tables 5.3, D.12, D.13, D.14, D.15, and D.16 show that the second subround syndromes obtained for all such errors are unique and nontrivial. As before, unique syndromes mean that the relevant errors can be detected and distinguished unambiguously, and nontrivial syndromes mean that a measurement error on the ancilla can be identified by the trivial syndrome in the second subround.

Furthermore, note that the data qubit errors appearing in these tables are the only errors which could have caused a nontrivial syndrome without the flag, regardless of the location of the fault (such as a two-qubit gate or idling location) leading to these errors. This is based on calculating all 7-qubit Pauli strings with support on 1 qubit, which anticommute with, and can hence be signalled by, the respective (flagged) stabilizer vis the nontrivial syndrome without flag outcome. Additionally, an error may occur in the cycle where it cannot be detected by any subsequent flagged measurement (see Sec. 4.4). For example, consider an X error occurring on the second qubit before the flagged $ZIZIZIZ$ measurement. This error does not anticommute with this stabilizer, so the second subround does not get initiated. This error also does not get propagated to multiple data qubits (due to the error propagation rules presented in Sec. 3.5). Thus, this error becomes a weight-1 input error at the beginning of the next cycle, and can be corrected likewise. These errors are also present in the Knill's error model, due to two-qubit gate depolarizing errors. Since the flagged stabilizer measurements to detect input errors are the same for the split-and-diagnose protocol (Fig. 5.6) as for the original protocol (Fig. 4.6), and the reduced unflagged stabilizer sequences in the $(s, f) = (1, 0)$ branches yield unique and nontrivial syndromes for these errors (see Tables 5.3, D.12, D.13, D.14, D.15, and D.16), the new protocol is capable of correcting a weight-1 error on a data qubits.

Therefore, the rigorously derived LUT (see Table E.8 and Table E.9), which demonstrate that the derived stabilizer sequences lead to unique and non-trivial syndromes for error sets signalled by particular flagged measurements, and the preceding analysis, demonstrate the fault tolerance of the split-and-diagnose protocol in Fig. 5.6 under the errors discussed here, which includes a single input error or internal fault.

5.4.4. Overhead Reduction

Given that a flag has been triggered, the protocol in Fig. 5.6 requires 3 stabilizer measurements instead of 6, which is a reduction of 50%. Assuming that gates do not need to be decomposed further, in line with [25],[35], a sequence of 3 unflagged stabilizers requires $3 \times 4 = 12$ two-qubit gates, while 6 unflagged generators require $6 \times 4 = 24$ two-qubit gates, which is also a reduction of 50%. This assumes circuits of the form shown in Fig. 3.3 for individual stabilizer measurements.

Given that a nontrivial syndrome without flag has resulted from a flagged measurement, the protocol in Fig. 5.6 requires 4 stabilizer measurements instead of 6, which is a reduction of 33.33%. Assuming that gates do not need to be decomposed further, a sequence of 4 unflagged stabilizers requires $4 \times 4 = 16$ two-qubit gates, while 6 unflagged generators require $6 \times 4 = 24$ two-qubit gates, which is also a reduction of 33.33%. This assumes circuits of the form shown in Fig. 3.3 for individual stabilizer measurements.

If one views the second subround (unflagged measurements) collectively, then the number of measurements and two qubit gates in the $f = 1$ branches goes down from 6 to 3, and from 6 to 4 for the $(s, f) = (1, 0)$ branches. Therefore, at an average, the number of stabilizer measurements gets reduced by 41.67% in the second subround.

When viewed as a whole, the decision tree of Fig. 5.6 requires a minimum of 4 measurements, which occurs for the $f = 1$ branch after the flagged $IIIXXXX$ measurement. The minimum number of measurements for the original protocol (see Fig. 4.6) is 7, which occurs in the same scenario. Therefore, the new protocol gives a reduction of 42.86% on this metric. The maximum number of measurements required in Fig. 5.6 are 10, which occurs when the $(s, f) = (1, 0)$ branch of the flagged $ZIZIZIZ$ measurement is followed. The maximum number of measurements for the original protocol (see Fig. 4.6) is 12, which occurs in the same scenario. Therefore, the new protocol gives a reduction of 16.67% on this metric.

5.4.5. Miscellaneous Remarks

The reduction in the number of stabilizer measurements obtained for the Steane code, using the protocol in Fig. 5.6, is much more substantial than that for the $[\![5, 1, 3]\!]$ code. For the $f = 1$ branches, this is because the original flag protocol for the Steane code (see Fig. 4.6) requires 6 stabilizer measurements in these branches, while the original flag protocol for the $[\![5, 1, 3]\!]$ code (see Fig. 4.4) requires 4 stabilizer measurements in these branches. At the same time, the number of errors to be distinguished by $f = 1$ branches are the same for both protocols, namely, 7 (for example, those in Eq. 5.1 and Eq. 5.3). The number of errors is the same because the number of bad two-qubit gates for stabilizer generators of both codes is the same (2), because the generators for both codes have weight 4. The original Steane code protocol (see Fig. 4.6) measures an even larger number of stabilizers than the $[\![5, 1, 3]\!]$ code protocol (see Fig. 4.4) to distinguish the same number of errors, leading to larger superfluity and hence more possibility for reduction. Further, the possibility of stabilizer reduction for the $(s, f) = (1, 0)$ branches for the Steane code protocol, which is not applicable to the $[\![5, 1, 3]\!]$ code protocol, is another contributing factor in the higher reduction.

As an added advantage, the reduced stabilizer sequences used in the $f = 1$ branches, as well as in the $(s, f) = (1, 0)$ branches, in the new protocol for the Steane code (see Fig. 5.6) only make use of weight-4 stabilizers. It is not necessarily required to use the weight-6 Steane code stabilizers, whose circuits would require a larger number of gates than the weight-4 stabilizers, to achieve the split-and-diagnose-style reduction. Additionally, no Y-type stabilizers are used and, in fact, only the standard stabilizer generators are used in all unflagged measurements, which may be advantageous for physical implementation.

The substantial stabilizer reduction in the Steane code protocol also comes with a price, that the CSS-style decoding of X- and Z-type stabilizer measurements is no longer followed. The impact this has can be seen by an example. Suppose a single fault from a bad location in the flagged $ZIZIZIZ$ measurement propagates to the error $IIIIZIZ$ on the data qubits (see Table C.10). This error contains only Z operators, and can be corrected using the syndrome from X-type stabilizer measurements only. Now suppose that along with this fault, a single qubit error $XIIIIII$ has also occurred, which makes the cumulative error to be $XIIIZIZ$. This event occurs with probability $\mathcal{O}(p^2)$. Although this error does not appear in Table C.10, it can still be corrected by the original protocol, by correcting the X and Z components separately. Thus, the original protocol has the capability to correct some higher order errors. The new protocol (see Fig. 5.6) may not have the capability to correct such higher order errors, because the stabilizer sequences have been designed to correct only $\mathcal{O}(p)$ errors.

However, this does not prevent the new protocol (see Fig. 5.6) from being fault-tolerant, because the protocol can still correct $\mathcal{O}(p)$ errors, discussed in the previous section, which is required for fault tolerance.

Finally, the split-and-diagnose-style stabilizer reductions, in which reduced stabilizer sequences are constructed by analyzing the set of errors to be distinguished, are possible for flag protocols, because different flagged measurements neatly and effectively separate the errors occurring in different parts of the circuit into restricted sets.

5.5. Detect-and-Diagnose Protocol for the Steane Code with Reduced, High-Weight Flagged Measurements

The third new protocol developed in this thesis is the flag protocol for the Steane code, in which flagged measurements are carried out with high-weight stabilizers, instead of the stabilizer generators as in the original protocol (see Fig. 4.6). This protocol is an extension of the technique behind Delfosse and Reichardt's sequence presented in Fig. 5.4 to flag protocols. This new protocol for the Steane code requires 3 flagged measurements instead of 6, and this reduction happens in every cycle, regardless of whether an error is detected or not. The reduced stabilizer sequences are capable of detecting, and subsequently correcting a single weight-1 data qubit error, as well as detecting and distinguishing faults which trigger the flag during flagged measurements. This protocol is shown in Fig. 5.7. This protocol will also be referred to as the flag protocol with first subround reduction for the Steane code.

Figure 5.7: The detect-and-diagnose flag protocol for the Steane code. This protocol requires 3 weight-6 flagged stabilizer measurements, instead of the 6 weight-4 generators required by the original protocol in Fig. 4.6.

5.5.1. Protocol Description

The cycle begins by measuring 3 weight-6 (or high-weight) operators $IZZXXYY$, $XIXYZYZ$ and $ZXYYXZI$ with flagged circuits. If the flag or syndrome measurement outcome indicates that a fault propagation from a bad location or an error has occurred, the flagged measurements are discontinued, and all 6 generators are measured with unflagged circuits. The ancilla qubit and the flag are reset after every flagged measurement, and the ancilla is reset after every unflagged measurement. The resulting syndromes from X- and Z- type stabilizer measurements are decoded separately, following the conventional CSS-style decoding. The cycle ends when either a syndrome is decoded via look-up tables and the correction applied, which could have resulted from flag or ancilla measurement outcomes, or, if all flagged measurement outcomes are trivial, indicating that no ($\mathcal{O}(p)$) error has occurred, in which case, no correction is applied.

The syndrome values on which adaptive measurements in Fig. 5.7 depend are explained as follows.

The operators chosen for flagged measurements differ from the Steane code stabilizers by a negative sign, see Eq. B.2. For example, the operator $-IZZZXXYY$ is a stabilizer of the Steane code, while the operator measured is $IZZXXYY$. Consequently, when no error has occurred, or the error commutes with the high-weight stabilizer, the syndrome obtained is bit value 1, or the -1 eigenvalue. When the error anticommutes with the stabilizer, the syndrome obtained is bit value 0. This is opposite to the convention for measurement outcomes for the protocols discussed so far. Thus, only with respect to this protocol, the phrase 'nontrivial syndrome without flag' from flagged measurements will refer to the $(s, f) = (0, 0)$ outcome, and a trivial syndrome from a flagged measurement will refer to the $s = 1$ outcome. This is also depicted on the edges after adaptive measurements in Fig. 5.7. The convention for the flag measurement stays the same: a flag getting triggered refers to the bit value 1, or the -1 eigenvalue. The convention for the second subround (unflagged measurements) also stays the same: the trivial syndrome means the bit value 0.

It may be worth mentioning that the initial state is still encoded in the codespace of the Steane code, i.e. the simultaneous $+1$ eigenspace of all stabilizer generators, and not only the space which is the simultaneous $+1$ eigenspace of the 3 high-weight stabilizers alone. Further, when the measurement of one of the weight-6 operators gives a syndrome bit 1, the state is projected to the -1 eigenspace of that operator, which is the $+1$ eigenspace of the corresponding stabilizer. Hence, if all flagged measurements give the trivial outcome, the state is in the codespace.

The decoding of syndromes is performed using an Z correction LUT and a X correction LUT used in conjunction, and presented in Table E.10 and Table E.10, respectively. A difference from the original protocol (see Fig. 4.6) is that both X and Z correction flag LUTs in the new protocol in Fig. 5.7 may contain corrections of weight ≥ 1. This is explained by the propagated errors resulting from faults at bad locations in flagged measurements.

5.5.2. Construction of the Stabilizer Sequences

The motivation for this construction is the observation that the syndrome information obtained from flagged measurements in the original flag protocols (see Fig. 4.6, Fig. 4.4) is not directly used to distinguish the actual error that has occurred. The role of the flagged measurements is only to detect and signal that an error or fault at a bad location, belonging to a particular limited set, has occurred. This motivates replacing the flagged stabilizer generator measurements by fewer, possibly high-weight flagged stabilizer measurements, which are sufficient detect an arbitrary single-qubit input error. These stabilizers may or may not necessarily be able to correct these errors by themselves. If such stabilizers can be found from the complete stabilizer group, they may be used for flagged measurements, and the error can be diagnosed by measuring all stabilizer generators with unflagged circuits in the second subround. The new protocol in Fig. 5.7 relies on the realization that such stabilizers can indeed be found for the Steane code. This justifies the name: 'detect-and-diagnose'.

The weight-6 stabilizers of the Steane code have the structure that each Pauli operator (X, Y, Z) has support on 2 qubits (see Eq. B.2). Consider the following triplet of stabilizers of the Steane code:

$$\begin{aligned} &-IZZXXYY, \\ &-XIXYZYZ, \\ &-ZXYYXZI. \end{aligned} \tag{5.5}$$

These stabilizers measure at least two different non-identity Pauli operators on every qubit. For example, on qubit 7, the operators measured are Y, Z, and I. This gives these stabilizers the capability to detect any single-qubit input error. This is because a Pauli operator, representing the error, must anticommute with at least one of the two different non-identity Paulis which are a part of the parity check, due to anticommutation relations (see Eq. 2.7). For the example above, an input X error on qubit 7 anticommutes with both Y and Z in the stabilizers, an input Y error anticommutes with the Z only, and an input Z error anticommutes with the Y only. Thus, every input error can be detected by at least one of the weight-6 stabilizer measurements. This may also be seen from Table 5.4.

Once the input error is detected, the actual error can be identified by measuring all 6 stabilizer generators in the second subround (the $(s, f) = (0, 0)$ branch, Fig. 5.7). The correction is then determined from the usual weight-1 input error LUTs (see Tables E.3 and E.4).

To achieve fault tolerance, it is necessary that the inequivalent propagated errors resulting from bad locations (indicated by the flag getting triggered) have unique and nontrivial syndromes, when a sequence of unflagged stabilizers is measured in the second subround. This holds for the three weight-6 stabilizers, which is discussed here. First, the circuits used to measure the high-weight operators in Fig. 5.7 are shown in Fig. 5.8.

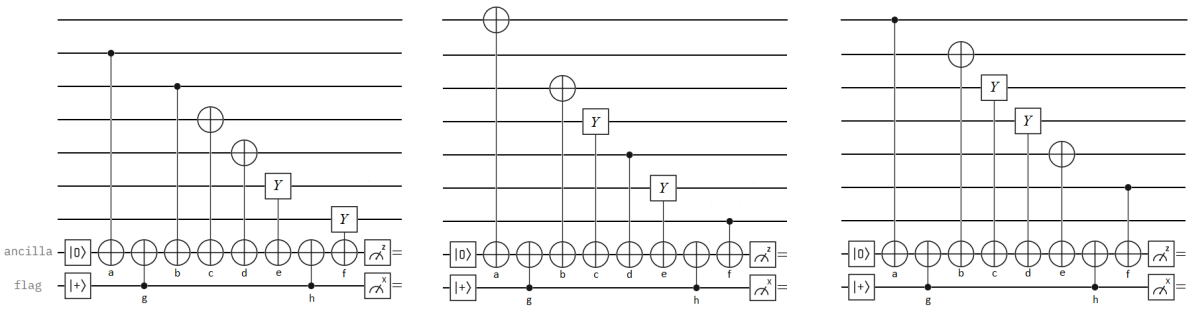

Figure 5.8: Flagged circuits for measuring the operators $IZZXXYY$, $XIXYZYZ$, and $ZXYYXZI$ respectively, used in the detect-and-diagnose protocol in Fig. 5.7.

It is important to specify the circuits used for flagged measurements, because the propagated errors from bad locations depend on the ordering of gates between data qubits and the ancilla. From Fig. 5.8, it may be seen that since each weight-6 operator measurement circuit requires 6 two-qubit gates between the data qubits and the ancilla, there are 4 bad gates, resulting in 13 inequivalent propagated errors, for each circuit. The resulting propagated errors, and the faults which cause them, are specified in Tables D.17, D.18 and D.19, for the circuits in Fig. 5.8 for $IZZXXYY$, $XIXYZYZ$ and $ZXYYXZI$ respectively.

A rigorous calculation of the second subround syndromes for these propagated errors shows that the syndromes obtained are unique for inequivalent errors, and further, this holds separately for the X and Z components of the errors. This is also presented in Tables D.17, D.18 and D.19. In addition, the syndromes from X- and Z-type stabilizers are nontrivial when the error has a non-identity Z or X component, respectively. The 3 stabilizers $-IZZXXYY$, $-XIXYZYZ$, and $-ZXYYXZI$ have been specifically chosen so that the circuits in Fig. 5.8 lead to such syndromes. Therefore, these errors are detectable and distinguishable, and hence can be corrected by the protocol in Fig. 5.7.

From Tables D.17, D.18 and D.19, it may be observed that both the X corrections and Z corrections may have weight ≥ 1 when the flag gets triggered (see columns 5 and 8). An example of this may be seen in column 5, row 1 and column 8, row 11 in Table D.17. Decoding of syndromes using stabilizer sequence used in Fig. 5.7 is done using the composite LUTs in Tables E.10 and E.11, which need to be used in conjunction, because they apply the X and Z corrections separately. These LUTs incorporate the the previously references tables for $f = 1$ branches, and the usual weight-1 LUTs (see Tables E.3 and E.4) for $(s, f) = (0, 0)$ branches.

5.5.3. Analysis of Fault Tolerance

First, it is essential to analyze that the protocol can correct any weight-1 input error. All such possible error strings are presented in Table 5.4. The table also shows whether these errors anticommute with the weight-6 stabilizers (indicated by 1) or not (0).

It may be observed that every input error anticommutes with at least one of the weight-6 stabilizers, because there are no errors with the entry 0 for all three stabilizers. This is because the weight-6 stabilizers measure at least 2 different Pauli operators on each data qubit, and can thus detect any weight-1 input error. In the protocol in Fig. 5.7, a weight-1 input error will get detected by the flagged measurement of the first weight-6 stabilizer it anticommutes with. This will give a nontrivial syndrome outcome without the flag (for this protocol, that corresponds to $(s, f) = (0, 0)$), and initiate the second subround, consisting of unflagged measurements of all stabilizer generators. Since all generators are measured, the resulting syndrome can be decoded using the standard weight-1 correction LUTs for the Steane code (see Tables E.3 and E.4). Thus, any weight-1 input error can be corrected.

Input error	Anticommutation with $-IZZXXYY$	Anticommutation with $-XIXYZYZ$	Anticommutation with $-ZXYYXZI$
$XIIIIII$	0	0	1
$IXIIIII$	1	0	0
$IIXIIII$	1	0	1
$IIIXIII$	0	1	1
$IIIIXII$	0	1	0
$IIIIIXI$	1	1	1
$IIIIIIX$	1	1	0
$ZIIIIII$	0	1	0
$IZIIIII$	0	0	1
$IIZIIII$	0	1	1
$IIIZIII$	1	1	1
$IIIIZII$	1	0	1
$IIIIIZI$	1	1	0
$IIIIIIZ$	1	0	0
$YIIIIII$	0	1	1
$IYIIIII$	1	0	1
$IIYIIII$	1	1	0
$IIIYIII$	1	0	0
$IIIIYII$	1	1	1
$IIIIIYI$	0	0	1
$IIIIIIY$	0	1	0

Table 5.4: Table showing that every weight-1 input error anticommutes with at least one of the high-weight stabilizers ($IZZXXYY$, $XIXYZYZ$, and $ZXYYXZI$) measured in the detect-and-diagnose protocol for the Steane code in Fig. 5.7. This is because the 3 stabilizers chosen have the property that at least two different Paulis are measured on every qubit. The entry in the columns is a 1 if the error anticommutes with the stabilizer, and 0 otherwise.

The next case to be analyzed is when a weight-1 error happens on a data qubit, when one flagged stabilizer measurement has been completed and the next one is yet to commence (i.e. in-between stabilizer measurements). This may be a result of a gate fault, or due to idling. If this error anticommutes with any of the remaining stabilizer measured with a flagged circuit, it gets detected and gives the nontrivial syndrome outcome, and can be corrected via the syndrome obtained from unflagged measurements.

If the error does not anticommute with any remaining flagged stabilizer, the error goes undetected, and becomes a weight-1 input error for the next QEC cycle, as presented in Sec. 4.4. This is because it does not propagate to multiple data qubits due to the subsequent circuits (it can at most propagate to the ancilla and flip its measurement outcome in the case the error anticommutes with the stabilizer; see the error propagation rules in Sec. 3.5). Thus, in this case, the error can be corrected like a weight-1 input error, as described before. This means the protocol is fault-tolerant against weight-1 errors on data qubits in such locations in the circuit.

Fault tolerance against faults propagating from bad two-qubit gates in flagged stabilizer measurements has been described in the previous section, and is sketched here for completeness. The propagated errors on data qubits resulting from these faults are presented in Tables D.17, D.18 and D.19. Analogous to the protocols in Sec. 4, it can be argued that these are the only errors to be distinguished when a flag gets triggered during the respective flagged stabilizer measurement, although they may arise from different fault locations. The triggered flag causes the protocol to enter an $f = 1$ branch. This yields a syndrome from measuring all 6 (unflagged) stabilizer generators. The tables show that these syndromes are unique for inequivalent errors, separately for the X-type and Z-type measurements. Further, the syndromes from X- or Z-type stabilizer measurements are nontrivial when the error has a Z or X component, respectively. These tables are used to construct fault-tolerant flag LUTs, which are used to decode the second subround syndrome. Therefore, these errors are detectable and distinguishable, and they can be unambiguously identified and corrected by the protocol in Fig. 5.7.

To analyze measurement errors, note that non-identity errors which trigger the flag, or cause the flag to get triggered, lead to nontrivial syndromes from the second subround (unflagged measurements). This may be seen from Tables E.3 and E.4 for the $(s, f) = (0, 0)$ branches, and Tables D.17, D.18 and D.19 for the $f = 1$ branches. Thus, a measurement error on the ancilla or the flag qubit results in the trivial syndrome (all zeros) during the second subround. Since no other error corresponds to this syndrome, measurement errors can be identified unambiguously, and the LUTs dictate that no correction needs to be applied. Thus, the protocol can recover from single measurement errors as well.

The remaining analysis for fault tolerance, such as the fact that flag CNOTs do not add bad locations, remains the same as the original flag protocols, discussed in Sec. 4.4.

The tables referred to in the preceding paragraphs are used to construct the composite decoding LUTs for the protocol in Fig. 5.7. These are presented in Tables E.10 and E.11, separately for Z and X corrections, and are used in conjunction. Therefore, the preceding analysis, and the rigorously derived LUTs, with unique and nontrivial syndromes, render the relevant errors detectable and distinguishable, signify the fault tolerance of this protocol against the specified errors.

5.5.4. Overhead Reduction

The protocol in Fig. 5.7 requires at most 3 flagged measurements instead of at most 6 required by the original protocol, which is a reduction of up to 50% in the number of flagged measurements. Assuming no further decomposition of two-qubit gates, in line with [25],[35], the reduction in the number of two-qubit gates in flagged measurements is calculated as follows. Each flagged measurement in the original protocol (see Fig. 4.6) requires 4 gates between the data qubits and the ancilla, and 2 gates between the ancilla and the flag, leading to a total of at most 36 two-qubit gates in the first subround for the 6 generators. Each high-weight stabilizer measurement in the new protocol (see Fig. 5.7) requires 6 gates between the data qubits and the ancilla (because the stabilizer weight is 6), and 2 gates between the ancilla and the flag. This leads to a maximum of 24 two-qubit gates in the first subround, for 3 stabilizers (this is a maximum because all flagged stabilizers may not get measured in every cycle, due to the nature of the protocol). Assuming all flagged measurements are carried out for both protocols, this is a reduction of 33.33% in the number of two-qubit gates.

The decision tree of Fig. 5.7 requires a minimum of 7 measurements, which occurs when the protocol enters the $f = 1$ or the $(s, f) = (0, 0)$ branch after the first flagged measurement. This is also the minimum

number of measurements required by the original protocol (see Fig. 4.6), which occurs in the same scenario. Therefore, there is no reduction in this metric. The maximum number of measurements required by the new protocol is 9, which occurs in the $f = 1$ or $(s, f) = (0, 0)$ branches after the last flagged measurement, while this number is 12 for the original protocol, which occurs in the same scenario. Therefore, the new protocol gives a reduction of 25% on this metric.

The distinguishing feature of this protocol as compared to the other two new protocols proposed (see Fig. 5.5 and Fig. 5.6) is that the savings offered by this protocol are present in every cycle, including (especially) the ones where no errors are detected, while in the other two protocols, the savings are present only when an error has been detected.

5.5.5. Miscellaneous Remarks

As the first remark, analogous to the argument for the split-and-diagnose protocol for the Steane code (see Sec. 5.4.5), the detect-and-diagnose protocol may not be able to correct some higher order errors (such as an input error with an X and a Z error on different qubits), which are correctable by the original protocol [25]. This is because, the flagged stabilizer sequences have been selected to detect a single input error, and may not be able to detect every higher order error (in general). This may be viewed as a trade-off in exchange for the reduced circuit depth, and is highlighted in the future outlook (see Sec. 8).

The second remark is about the number of weight-6 stabilizers used for a detect-and-diagnose-style protocol (like Fig. 5.7) for the Steane code. A combination of 2 weight-6 stabilizers does not satisfy the requirement of measuring at least two different Paulis on every qubit. This is because, every weight-6 stabilizer has no support on one qubit (an identity operator acts on it), and choosing only 2 weight-6 stabilizers would mean that on those qubits, only at most one Pauli operator can be measured.

On the other hand, if 4 weight-6 stabilizers were required to meet the same condition, it would significantly reduce the savings in the number of gates for flagged measurements. This is because, measuring 4 flagged weight-6 requires $4(6 + 2) = 32$ two-qubit gates, while the original protocol measures the 6 weight-4 flagged generators, leading to a two-qubit gate count of $6(4 + 2) = 36$. This is only a reduction of 11.11%, as opposed to 33.33% with 3 weight-6 stabilizers, calculated earlier.

Therefore, having to measure 3 weight-6 stabilizers provides the desired functionality, and still provides a significant reduction.

The third remark is about stabilizer sequences used for unflagged measurements in the protocol in Fig. 5.7.These have not been reduced, based on distinguishing only the relevant set of errors, as has been done for the split-and-diagnose protocols. Instead, the 6 unflagged stabilizer generators have been retained. This is because, due to the increased weight of the flagged stabilizer, a larger number of possible errors are indicated when a flag is triggered, or a nontrivial syndrome is observed. In principle, it may be possible to find reduced stabilizer sequences for unflagged measurements. However, this has not been addressed in this thesis, because of the added complexity due to distinguishing an increased number of errors. This is also highlighted in the future outlook (see Sec. 8).

6

Numerical Results

The noise performance of the flag protocols developed in this thesis is analyzed by Monte Carlo simulations to determine the pseudothreshold values under Knill's error model, as used in [25], and compared to that of the original protocols [25]. These results are presented in this section. Two types of simulation have been performed. The first type simulates a single cycle of error correction per shot, in which only the error correction gadget has faults. The second type simulates multiple (consecutive) cycles of error correction, which allows weight-1 input errors, resulting from a previous error correction gadget, to be present before the next cycle. This style has been followed in [25], and serves to verify the fault tolerance conditions under at most 1 input error or 1 internal fault.

The simulations are implemented as follows. The protocols which are simulated, along with their respective circuits, are presented in Sec. 4 (Fig. 4.4 and Fig. 4.6, the original protocols) and Sec. 5 (Fig. 5.5, Fig. 5.6, and Fig. 5.7, the protocls constructed in this thesis). For every Monte Carlo sample, i.i.d. errors are applied probabilistically in the circuit for the relevant flag protocol, at locations dictated by the specific noise model. The circuit is simulated, and according to the protocol, the syndromes are obtained and possible corrections applied, if errors are indicated. This is followed by an ideal decoding round, consisting of another cycle of error correction with the same flag protocol without errors. This is done to remove the residual $O(p)$ errors, as followed in [25], [35], and [36]. The purpose of the ideal decoder is to verify the first fault tolerance condition presented in Subsec. 3.10. The ideal decoding step projects the resulting state to the codespace, which is analyzed to determine if a logical error has occurred [35],[36]. The logical error rate for a given physical error rate is calculated from the ratio of the total count of logical errors observed in the Monte Carlo samples, to the number of samples. The physical error rate is varied to determine the pseudothreshold value, below which the protocol can correct the applied errors (see Subsec. 3.10).

All simulations are carried out on the DelftBlue supercomputer [47]. The system is employed to parallelize the identical Monte Carlo samples over multiple cores via the message passing interface (MPI) protocol. The logical error counts resulting from different cores are combined at the end of the simulation. This enables execution of the required number of samples within meaningful timeframe. In particular, 384 cores from Intel Xeon compute nodes are used, each of 3.0 GHz, 192 GB of memory, and 480 GB of hard-drive space. The simulations were developed from scratch using Python, some of which employ Qiskit [48] , incorporating ease-of-use and software capability for potential future development as a flag fault tolerance simulation tool. The source code for the simulation of the flag protocol for the $[\![5,1,3]\!]$ code may be found in the public github repository: https://github.com/dhruvbhq/lowdepthflagqec.

The pseudothresholds are simulated under the Knill's error model, described in Sec. 3.9 in Fig. 3.16, which is the same model employed in [25]. The simulations allow establishing a comparison with the original protocols in [25] under the same error model. The relative error probability values are taken as $\frac{4p}{15}$ for state preparation errors, $\frac{4p}{15}$ form measurement errors, and p for two-qubit gate depolarizing error, which are be identical to those in [25]. In line with [25], single-qubit gate depolarizing error probability is taken to be 0, since two-qubit gates have been simulated without further decomposition. This error model does not include idling errors. The pseudothreshold values are established relative to the two-qubit gate error probability, p.

The first simulation is based on wavefunction evolution, and is performed with a simulation developed using Qiskit [48] as a base. In these simulations, the circuit is reset at the end of simulation for each sample,

to simulate a single error correction gadget [35]. These simulations make use of noiseless encoding circuits for the $[\![5,1,3]\!]$ code, presented in Fig. 3.4, and the Steane code, presented in Fig. 3.5. The encoding circuits are assumed to be noiseless to restrict the scope of this simulation to assess pseudothresholds when only the error correction gadget (syndrome extraction circuit) has faults. The initial state before encoding is prepared by the Y rotation $R_y(\pi/3)\,|0\rangle$, so that all logical single-qubit Pauli errors can be observed in the final state. The values of the physical error rate, p, are taken to be 13 equally, logarithmically spaced values between 6.31×10^{-4} and 10^{-2}, both included. For each value of physical error rate $p \leq 10^{-3}$, 10^7 Monte Carlo samples are taken, and 10^6 samples are taken for physical error rate $p > 10^{-3}$, to make efficient use of available processing capabilities.

The pseudothreshold simulation results under Knill's error model, using wavefunction simulations and resetting the circuit after every sample, are shown in Fig. 6.1. The logical error rate data are fit to curves polynomial in the physical error rate [1], and pseudothreshold values are computed from intersection points of these curves with the physical error rate line. The pseudothresholds, error bar values and polynomial coefficients for simulation under Knill's error model, using wavefunction simulations and resetting the circuit after every sample, are presented in Table 6.1.

Flag Protocol	Pseudothreshold	Best fit polynomial
$[\![5,1,3]\!]$ code, unmodified protocol	$3.573\times10^{-3}\pm1.293\times10^{-4}$	$(-7.129\times10^{-2})p + (3.332\times10^{2})p^2+\mathcal{O}(p^3)$
$[\![5,1,3]\!]$ code, $f=1$ reduction	$3.703\times10^{-3}\pm1.319\times10^{-4}$	$(-1.179\times10^{-1})p + (3.462\times10^{2})p^2+\mathcal{O}(p^3)$
Steane code, unmodified protocol	$2.192\times10^{-3}\pm9.647\times10^{-5}$	$(-1.637\times10^{-1})p + (5.515\times10^{2})p^2+\mathcal{O}(p^3)$
Steane code, $f=1$ and $s=1$ reduction	$2.430\times10^{-3}\pm1.064\times10^{-4}$	$(7.752\times10^{-2})p + (3.941\times10^{2})p^2+\mathcal{O}(p^3)$
Steane code, first subround reduction	$2.361\times10^{-3}\pm1.422\times10^{-4}$	$(2.254\times10^{-1})p + (3.389\times10^{2})p^2+\mathcal{O}(p^3)$

Table 6.1: Computed pseudothreshold values and best fit polynomial coefficients for the simulation results presented in Fig. 6.1 (under Knill's error model, using wavefunction simulations and resetting the circuit after every sample). The error bars have been calculated from calculating pseudothresholds for $\pm2\sigma$ values of logical error rate.

The error in computed pseudothresholds, presented in Table 6.1, is calculated using curves fit to logical error rates $\pm2\sigma$ values, where $\sigma=\sqrt{\frac{p_L(1-p_L)}{N}}$, for a logical error rate p_L computed using N i.i.d. samples. In addition, the coefficients of the best fit curves (presented in Table 6.1) demonstrate quadratic scaling of logical error rate with physical error rate, with the coefficient of the $\mathcal{O}(p)$ terms being smaller than 1 and at least 3 orders of magnitude below the $\mathcal{O}(p^2)$ coefficient for each protocol. This suppression of $\mathcal{O}(p)$ errors signifies fault tolerance with distance-3 codes, as described in Subsec. 3.10.

It is observed that there is an increase in the pseudothreshold values for the new flag protocols calculated from Fig. 6.1, with respect to the originally developed protocols [25]. The pseudothreshold for the split-and-diagnose protocol ($f=1$ reduction) for the $[\![5,1,3]\!]$ code is increased by 3.71%, for the split-and-diagnose protocol ($f=1$ and $s=1$ reduction) for the Steane code, the improvement is 10.85%, and for the detect-and-diagnose protocol (first subround reduction) for the Steane code, the improvement is 7.71% with respect to the original protocols for the $[\![5,1,3]\!]$ code and the Steane code, respectively. This increase in pseudothreshold values is attributed to a reduction in the number of fault locations in the new protocols, which reduces the possible opportunities for errors in the circuit.

The second type of simulations are carried out to replicate the exact simulation style in [25], which is different from the method implemented above. In these simulations, the accumulated error after the syndrome extraction circuit is not reset after every sample. Rather, it is only reset if a logical error is detected in a cycle via the ideal decoder. As explained in [25], this models consecutive QEC cycles. In this simulation style, if a logical error has not been detected by the ideal decoder, then the state made

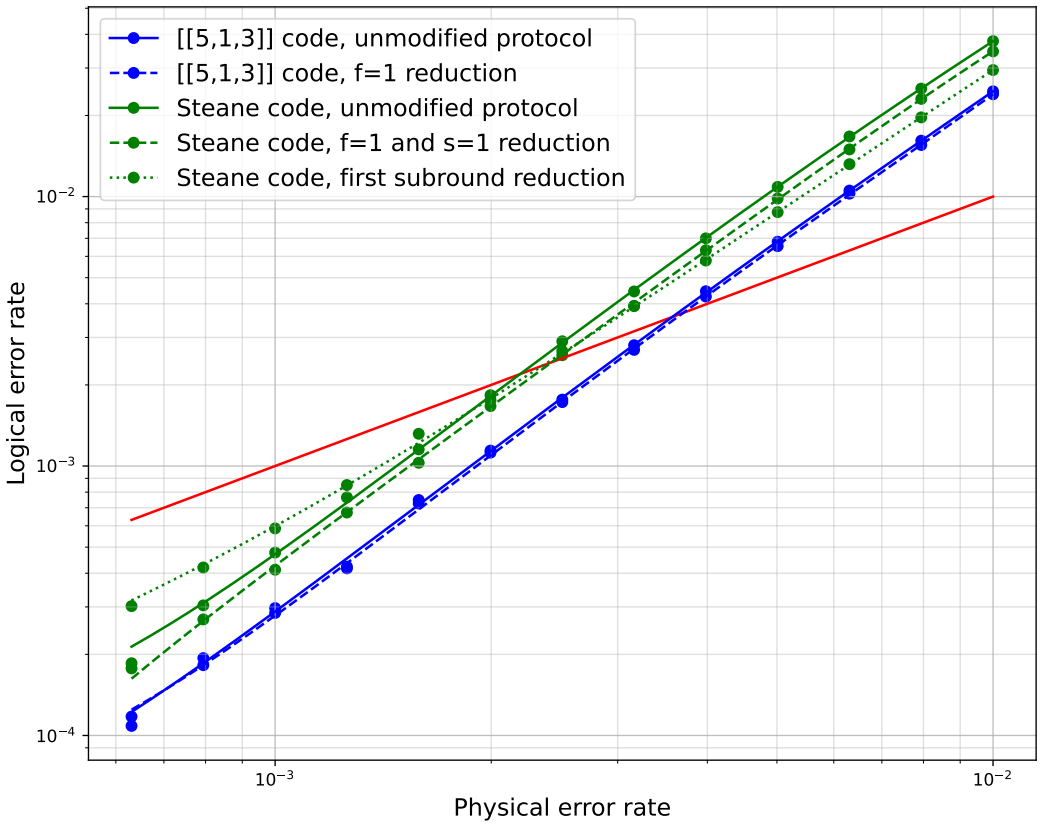

Figure 6.1: Logical error rates, fit to polynomial curves, resulting from pseudothreshold simulations under Knill's error model (described in Subsec. 3.9), using wavefunction simulations and resetting the circuit after every sample. The line coloured in red represents the physical error rate. The original protocols for the $[\![5, 1, 3]\!]$ code and the Steane code [25] are labeled as 'unmodified'. The split-and-diagnose protocol for the $[\![5, 1, 3]\!]$ code (see Fig. 5.5) is labeled as '$[\![5, 1, 3]\!]$ code, $f = 1$ reduction'. The split-and-diagnose protocol for the Steane code (see Fig. 5.6) is labeled as 'Steane code, $f = 1$ and $s = 1$ reduction'. The detect-and-diagnose protocol for the Steane code (see Fig. 5.7) is labeled as 'Steane code, first subround reduction'. The curves for the 2 protocols for the $[\![5, 1, 3]\!]$ code are coloured in blue, and those for the 3 protocols for the Steane code are coloured in green, for visual clarity. The values of the physical error rate, p, are taken to be 13 equally, logarithmically spaced values between 6.31×10^{-4} and 10^{-2}, both included. For each value of physical error rate $p \leq 10^{-3}$, 10^7 Monte Carlo samples are taken, and 10^6 samples are taken for physical error rate $p > 10^{-3}$. The mean pseudothreshold values obtained are: 3.573×10^{-3} for the unmodified protocol for the $[\![5, 1, 3]\!]$ code, 3.703×10^{-3} for the $[\![5, 1, 3]\!]$ code with $f = 1$ reduction, 2.192×10^{-3} for the unmodified protocol for the Steane code, 2.430×10^{-3} for the Steane code with $f = 1$ and $s = 1$ reduction, and 2.361×10^{-3} for the Steane code with first subround reduction. The pseudothresholds, error bar values and polynomial coefficients are presented in Table 6.1.

available as input to the next cycle is the one without the correction applied by the ideal decoder. This provides an enhancement over the previous simulation, in the sense that error correction for input errors is evaluated. These simulations are carried out to so that the new protocols may be compared with the original protocols by a pseudothreshold simulation comparable to [25].

The second type of simulations are implemented from scratch in Python as stabilizer simulations based on the Gottesman-Knill theorem, discussed in Sec. 3.4, and the binary symplectic vector representation, described in Sec. 3, for computational efficiency. These simulations do not make use of the encoding circuits presented in Sec. 3. The values of the physical error rate, p, are taken to be 13 equally, logarithmically spaced values between 6.31×10^{-4} and 10^{-2}, both included. For each value of physical error rate $p \leq 10^{-3}$, 10^7 Monte Carlo samples are taken, and 10^6 samples are taken for physical error rate $p > 10^{-3}$, to compare using the same number of samples as in [25].

The pseudothreshold simulation results under Knill's error model, using stabilizer simulations and only resetting the circuit in case a logical error is detected, are shown in Fig. 6.2. The logical error rate data are fit to curves polynomial in the physical error rate [1], and pseudothreshold values computed from intersections of these curves with the physical error rate line. The pseudothresholds, error bar values and polynomial coefficients for simulation under Knill's error model, using wavefunction simulations and only resetting the circuit in case a logical error is detected, are presented in Table 6.2.

Flag Protocol	Pseudothreshold	Best fit polynomial
$[\![5,1,3]\!]$ code, unmodified protocol	$3.127\times10^{-3}\pm1.289\times10^{-4}$	$(3.679 \times 10^{-1})p + (1.648 \times 10^{2})p^2 + \mathcal{O}(p^3)$
$[\![5,1,3]\!]$ code, $f = 1$ reduction	$3.177\times10^{-3}\pm1.266\times10^{-4}$	$(2.504 \times 10^{-1})p + (2.176 \times 10^{2})p^2 + \mathcal{O}(p^3)$
Steane code, unmodified protocol	$1.904\times10^{-3}\pm8.684\times10^{-5}$	$(-1.753 \times 10^{-1})p + (6.344 \times 10^{2})p^2 + \mathcal{O}(p^3)$
Steane code, $f = 1$ and $s = 1$ reduction	$1.828\times10^{-3}\pm8.757\times10^{-5}$	$(-1.546 \times 10^{-1})p + (6.578 \times 10^{2})p^2 + \mathcal{O}(p^3)$
Steane code, first subround reduction	$1.901\times10^{-3}\pm9.052\times10^{-5}$	$(-4.097 \times 10^{-1})p + (7.076 \times 10^{2})p^2 + \mathcal{O}(p^3)$

Table 6.2: Computed pseudothreshold values and best fit polynomial coefficients for Fig. 6.2 (under Knill's error model, using stabilizer simulations, and only resetting the circuit in case a logical error is detected). The error bars have been calculated from calculating pseudothresholds for $\pm2\sigma$ values of logical error rate.

For the second type of simulations, the pseudothresholds obtained for the new protocols are comparable to the corresponding unmodified original protocols, up to statistical error. Further analysis of this simulation method may explain this result. The second simulation style is expected to yield lower pseudothresholds than the first, because of additional errors possible at the input of an error correction gadget. In addition, the coefficients of the best fit curves demonstrate quadratic scaling of logical error rate with physical error rate, with the coefficient of the $O(p)$ terms being smaller than 1 and at least 3 orders of magnitude below the $O(p^2)$ coefficient for each protocol (see Table 6.2).

In conclusion, the three protocols developed in this thesis, which offer significant gate overhead reduction, demonstrate nontrivial pseudothresholds under the restricted error model, both when a single cycle is simulated, as well as when multiple consecutive cycles with the possibility of weight-1 input errors are simulated. This demonstrates that the new protocols can correct errors applied by the specified error model, which contains two-qubit depolarizing errors, preparation, and measurement errors, without idling errors. As discussed in Sec. 3.10, an increase in pseudothreshold, which may need to be verified with more extensive simulations, may be advantageous because it relaxes the physical error rate value to be achieved for fault tolerance. Additionally, the pseudothresholds obtained from the second simulations of multiple cycles are competitive with the original protocols, and demonstrate the capability of the new protocols to correct input errors.

To further investigate the fault tolerance of these protocols, pseudothresholds need to be established under more more extensive noise models, including circuit level noise with varying levels of idling error

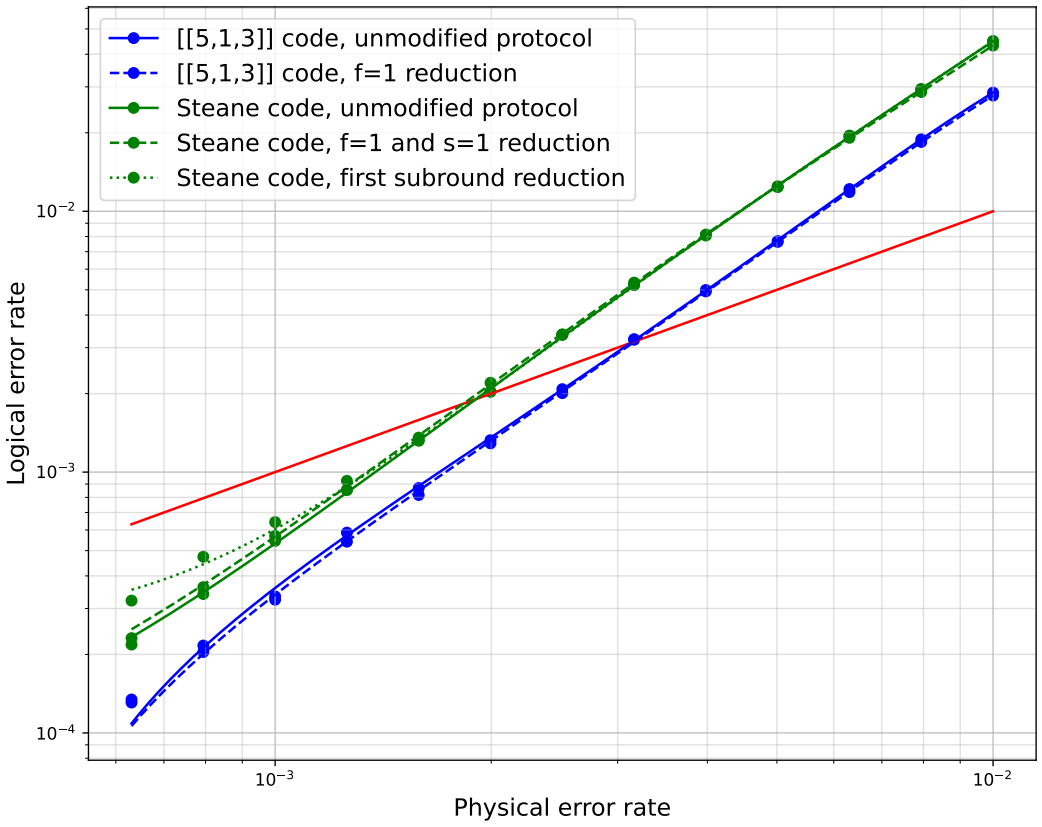

Figure 6.2: Logical error rates, fit to polynomial curves, resulting from pseudothreshold simulations under Knill's error model (described in Sec. 3.9), using stabilizer simulations and resetting the circuit only in case a logical error is detected. The line coloured in red represents the physical error rate. The original protocols for the $[\![5,1,3]\!]$ code and the Steane code [25] are labeled as 'unmodified'. The original protocols for the $[\![5,1,3]\!]$ code and the Steane code [25] are labeled as 'unmodified'. The split-and-diagnose protocol for the $[\![5,1,3]\!]$ code (see Fig. 5.5) is labeled as '$[\![5,1,3]\!]$ code, $f=1$ reduction'. The split-and-diagnose protocol for the Steane code (see Fig. 5.6) is labeled as 'Steane code, $f=1$ and $s=1$ reduction'. The detect-and-diagnose protocol for the Steane code (see Fig. 5.7) is labeled as 'Steane code, first subround reduction'. The curves for the 2 protocols for the $[\![5,1,3]\!]$ code are coloured in blue, and those for the 3 protocols for the Steane code are coloured in green, for visual clarity. The values of the physical error rate, p, are taken to be 13 equally, logarithmically spaced values between 6.31×10^{-4} and 10^{-2}, both included. For each value of physical error rate $p \leq 10^{-3}$, 10^7 Monte Carlo samples are taken, and 10^6 samples are taken for physical error rate $p > 10^{-3}$. The mean pseudothreshold values obtained are: 3.127×10^{-3} for the unmodified protocol for the $[\![5,1,3]\!]$ code, 3.177×10^{-3} for the $[\![5,1,3]\!]$ code with $f=1$ reduction, 1.904×10^{-3} for the unmodified protocol for the Steane code, 1.828×10^{-3} for the Steane code with $f=1$ and $s=1$ reduction, and 1.901×10^{-3} for the Steane code with first subround reduction. The pseudothresholds, error bar values and polynomial coefficients are presented in Table 6.2.

rate [15], as well as under physically realistic noise models. This investigation is beyond the scope of the present work.

7

Towards Generalization of Reduced Stabilizer Sequences

The development of particular instances of reduced stabilizer sequences in Fig. 5.5, Fig. 5.6 and Fig. 5.7, which have been derived analytically in Sec. 5, motivates exploration of other instances of reduced stabilizer sequences with fault-tolerant error correction rules, applicability to other codes, and a potential for mathematical description for such sequences.

This section briefly addresses performing a brute-force computer search for reduced stabilizer sequences, to assist in such generalization, and construction of stabilizer sequences of the detect-and-diagnose type for certain other codes. These programs, developed with assistance from Qiskit [48], exhaustively search over the complete stabilizer group to construct reduced stabilizer sequences. This approach yields numerous candidate stabilizer sequences with the same general structure as the protocols developed in this thesis. This is presented in Sec. 7.1 and Sec. 7.2. Further, computer search is used to identify potentially interesting mathematical properties detect-and-diagnose-style stabilizer sequences, capable of detecting 1 input error, for the Steane code (see Sec. 7.2). Finally, in Sec. 7.3, some high-weight stabilizer sequences with the capability to detect 1 input error are presented for the $[\![15, 7, 3]\!]$ quantum Hamming code and the $[\![9, 1, 3]\!]$ Shor code [1], indicating potential for generalization to other codes.

These preliminary observations and analyses may be useful to verify a theoretical formulation of these sequences, as well as provide candidate stabilizer sequences which more suited to certain physical implementations than others to construct fault-tolerant protocols with reduced overhead.

7.1. Computer Search for Split-and-Diagnose-Style Stabilizer Sequences

The starting point for the computer searches is the flag protocol for the $[\![5, 1, 3]\!]$ code. The aim of the program is to find reduced stabilizer sequences for the $f = 1$ branch of the protocol which yield unique and nontrivial syndromes for the errors which trigger the flag. The flagged measurements are left unchanged, and so are the unflagged measurements for the $(s, f) = (1, 0)$ (nontrivial syndrome without flag) branches.

The first ansatz, or constellation, of unflagged stabilizers for $f = 1$ branches is chosen as the one in which the first and second stabilizers do not depend on the past syndrome values, while the third depends on the second measurement outcome, analogous to the structure used in the new protocol presented in Fig. 5.5. An indexed notation is used to label the stabilizers being searched for as follow. If the i^{th} flagged measurement results in a triggered flag, the first two stabilizers measured are $S_{i,1}$ and $S_{i,2}$ respectively. If the measurement of $S_{i,2}$ gives a syndrome bit 0, the stabilizer $S_{i,2,1}$ is measured, otherwise $S_{i,2,2}$ is measured. In this notation, an additional index is added when a new adaptive measurement is carried out.

The algorithm for the computer search iterates over the complete stabilizer group of the $[\![5, 1, 3]\!]$ code for stabilizer sequences consisting of a single adaptive stabilizer measurement in the $f = 1$ branches of the protocol, analogous to the structure of the protocol in Fig. 5.5, calculates the syndromes for the error set via anticommutation relations, and returns as output those stabilizer sequences which give unique and nontrivial syndromes for the relevant error set, according to the criterion presented in Sec. 5.

The resulting reduced stabilizer sequences for the $[\![5,1,3]\!]$ code are presented in the appendix in Tables F.1, F.2, F.3, F.4, F.5, F.6, F.7, and F.8. For every stabilizer sequence in a $f=1$ branch, the search iterates over the full stabilizer group for a total of 4 stabilizers in the constellation (although, for diagnosing an error, only 3 need to be measured, since the sequence is adaptive). The size of the search space for every $f=1$ branch is $16^4 = 65536$, out of which, 256 stabilizer sequences giving unique and nontrivial syndromes are obtained for every $f=1$ branch. There are 4 $f=1$ branches, and any of the solutions for any of these branches is a potential candidate for the reduced stabilizer measurement sequence. The particular sequences developed in the protocol in Fig. 5.5 are also part of these solutions, and are coloured in red in these tables.

In view the derivation of the sequences in Fig. 5.5, some preliminary observations from these sequences are made. The stabilizers $S_{i,1}$ and $S_{i,2}$ for every $f=1$ sequence $(i \in 1,2,3,4)$ always have support on the deciding qubits for the particular flagged stabilizer measurement. Although the subsequent stabilizers $S_{i,2,1}$ and $S_{i,2,2}$ may have support on only one or both the deciding qubits, they measure a parity different from $S_{i,1}$ and $S_{i,2}$.

Analogously, the computer searches are also carried out to find other reduced, unflagged stabilizer sequences for the Steane code for the $f=1$ branches of the protocol in Fig. 4.6. These errors are presented in Tables 4.3, C.6, C.7, C.8, C.9, and C.10. The stabilizer sequences used in the $f=1$ branch resulting from the i^{th} flagged measure the unflagged stabilizers $S_{i,1}$ and $S_{i,2}$, followed by the third unflagged stabilizer, $S_{i,2,1}$ or $S_{i,2,2}$, depending on the outcome of the $S_{i,2}$ $(i \in \{1,2,\ldots,6\})$. This is equivalent in structure as in the newly-developed protocol in Fig. 5.6, with an inconsequential difference that the stabilizer measurement on whose outcome the third stabilizer is decided is measured at a different position, to make the program simpler. This only amounts to a change in the order of measuring $S_{i,1}$ and $S_{i,2}, i \in \{1,2,3,4,5,6\}$, and the same criterion of unique and nontrivial syndromes is used to select the reduced stabilizer sequences.

This search is restricted to the weight-4 stabilizers of the Steane code, to keep the size of the search space and solution set tractable. The stabilizers used for flagged measurements, as well as those for the $(s,f)=(1,0)$ branches are kept the same as the original protocol (see Fig. 4.6).

The reduced, weight-4 stabilizer sequences which yield unique and nontrivial syndromes for the $f=1$ branches of the Steane code, resulting from the computer search, are presented in the appendix in Tables F.9, F.10, F.11, F.12, F.13, and F.14. The size of the search space for every $f=1$ branch is $21^4 = 194481$ stabilizer constellations, because there are 21 weight-4 stabilizers (see Eq. B.2). Out of these, 112 stabilizer sequences giving unique and nontrivial syndromes are obtained for every $f=1$ branch. For the 6 $f=1$ branches, any of these solutions may be a potential candidate to construct a protocol with reduced stabilizer sequences. The particular sequences developed in the protocol in Fig. 5.6 are also part of these solutions, shown in red in these tables.

A trend is observed in the reduced stabilizer sequences, which yield unique and nontrivial synromesfor errors which trigger the flag, resulting from the search over weight-4 stabilizers. For $i \in \{1,2,3\}$, i.e. when the flagged measurement is an X-type stabilizer, one of the stabilizers $S_{i,1}$ and $S_{i,2}$ is always either a Z-type or a Y-type stabilizer, having the same support as the stabilizer used for the flagged measurement. For $i \in \{4,5,6\}$, one of $S_{i,1}$ and $S_{i,2}$ is always either a X-type or a Y-type stabilizer, having the same support as the stabilizer used for the flagged measurement. As before, these observations may be helpful for further analysis of these sequences.

Finally, it is explored whether non-adaptive sequences of 3 unflagged stabilizers in the $f=1$ branches of the $[\![5,1,3]\!]$ code and Steane code protocols can distinguish the errors which trigger the flag. These errors are presented in Tables presented in Tables 4.1, C.2, C.3 and C.4 for the protocol for the $[\![5,1,3]\!]$ codes, and in Tables 4.3, C.6, C.7, C.8, C.9, and C.10 for the protocol for the Steane code. Interestingly, even with the complete stabilizer group, the computer searches do not yield any non-adaptive sequences of 3 unflagged stabilizers, of the form $\{S_{i,1}, S_{i,2}, S_{i,3}\}$, for the $f=1$ branches which can give unique and nontrivial syndromes for the relevant propagated errors which are indicated by a triggered, for both these codes. Therefore, based on computer searches, a fault-tolerant protocol with such a descision tree may not exist for these codes. This may also indicate that the problem of designing a stabilizer sequence reaching the lower bound of $log_2\lceil(M+1)\rceil$ sequences to distinguish a given set of M inequivalent errors [1] with unique, nontrivial syndromes, posed in Sec. 5, and the use of adaptive sequences for this purpose, may have a nontrivial mathematical explanation.

7.2. Computer Search for Detect-and-Diagnose-Style Stabilizer Sequences

The detect-and-diagnose protocol for the Steane code relies on using 3 high-weight stabilizers for flagged measurements (see Fig. 5.7). These stabilizers measure at least two different Pauli operators on every qubit. This gives them the capability to detect any single-qubit input error. This motivates constructing other stabilizer triplets for the Steane code with the same property, to determine more such candidate stabilizer sequences for these protocols for the Steane code.

The search algorithm iterates over the entire stabilizer group, and numerous triplets of weight-6 stabilizers of the Steane code, which measure at least 2 different Pauli operators on every qubit, are found as solutions. More precisely, they are 1344 in number (stabilizer permutations are not counted as separate), out of the $64^3 = 262144$ possible stabilizer triplets. These are presented in the appendix in Tables G.1-G.16. The exact sequences used in the newly-developed protocol in Fig. 5.7 are also recovered in these solutions, and are coloured red in Table G.9, part of the previously referenced tables.

The computer search is also used to assist in identifying some mathematically interesting properties of a potential underlying structure of these stabilizer triplets. To explore this, the fact that the Steane code's stabilizer group is divided into subgroups of X-type and Z-type stabilizers is employed in a preliminary approach. This division motivates exploring whether the triplets of weight-6 stabilizers follow a structure based on this. To that end, a partition of the stabilizer group is constructed via the cosets of the X-type stabilizer subgroup. It is observed that, some triplets of weight-6 stabilizers follow the structure that the 3 stabilizers belong to different cosets of the X-type stabilizer subgroup. This is depicted in Fig. 7.1.

IIIIIII,IIIXXXX, IXXIIXX,XIXIXIX, IXXXXII,XIXXIXI, XXIIXXI,XXIXIIX (0)	IIIZZZZ,IIIYYYY, -IXXZZYY,-XIXZYZY, -IXXYYZZ,-XIXYZYZ, -XXIZYYZ,-XXIYZZY (1)	IZZIIZZ,-IZZXXYY, IYYIIYY,-XZYIXZY, -IYYXXZZ,-XZYXIYZ, -XYZIXYZ,-XYZXIZY (2)	ZIZIZIZ,-ZIZXYXY, -ZXYIZXY,YIYIYIY, -ZXYXYIZ,-YIYXZXZ, -YXZIYXZ,-YXZXZIY (3)
IZZZZII,-IZZYYXX, -IYYZZXX,-XZYZYIX, IYYYYII,-XZYYZXI, -XYZZYXI,-XYZYZIX (4)	ZIZZIZI,-ZIZYXYX, -ZXYZIYX,-YIYZXZX, -ZXYYXZI,YIYYIYI, -YXZZXYI,-YXZYIZX (5)	ZZIIZZI,-ZZIXYYX, -ZYXIZYX,-YZXIYZX, -ZYXXYZI,-YZXXZYI, YYIIYYI,-YYIXZZX (6)	ZZIZIIZ,-ZZIYXXY, -ZYXZIXY,-YZXZXIY, -ZYXYXIZ,-YZXYIXZ, -YYIZXXZ,YYIYIIY (7)

Figure 7.1: Graphic showing the stabilizer group of the Steane code partitioned into cosets of the X-type stabilizer subgroup, outlined in red. Some stabilizer triplets, which measure at least 2 different Pauli operators on each qubit are highlighted, and observed to belong to different cosets. On the bottom right of each block containing the elements of the coset, an arbitrarily assigned number is written, which is used to reference the coset later.

This observation is verified for all such stabilizer triplets via a computer program: all triplets of weight-6 stabilizers of the Steane code found by the search, which measure at least 2 different Pauli operators on every qubit, belong to different cosets of the X-type stabilizer subgroup within the stabilizer group. This may be verified from Tables G.1-G.16, in which the cosets to which each stabilizer belongs are presented. The particular coset is identified by an arbitrarily assigned number, according to Fig. 7.1. Further, no stabilizer in the triplet belongs to the coset numbered (0) in Fig. 7.1, because it is the subgroup of X-type stabilizers, and only contains weight-4 operators.

This insight motivates further exploration into this group structure. A computer program is used to verify that these stabilizer triplets which can detect 1 input error belong to different subgroups. If the 8-element subgroups generated by each of these stabilizer triplets are computed, the number of distinct subgroups to which these 1344 stabilizer triplets belong come out to be 48, via the program. This is also presented in Tables G.1-G.16, in which the stabilizer triplets are organized according to the subgroup from which they arise. This may also signify an underlying structure, identified with computer assistance.

Finally, a computer search is used to check whether any triplets of the stabilizers of the $[\![5, 1, 3]\!]$ code, which measure at least 2 different Pauli operators on every qubit, exist. The result is that, even when iterated over the complete stabilizer group, there are no sequences of 3 stabilizers found for the $[\![5, 1, 3]\!]$ code which satisfy this criteria. Thus, by the computer search, it is concluded that a detect-and-diagnose-

style protocol, of the form developed for the Steane code (see Fig. 5.7), may not exist for the $[\![5,1,3]\!]$ code with fewer than 4 flagged stabilizer measurements.

7.3. Detect-and-Diagnose-Style Stabilizers for Other Codes

Motivated by the existence of high-weight stabilizer sequences which can detect 1 input error for the Steane code, such sequences for two other codes are explored and are presented here.

The Shor code [1] is a $[\![9,1,3]\!]$ stabilizer code, with the following 8 stabilizer generators:

$$\{ZZIIIIIII, IZZIIIIII, IIIZZIIII, IIIIZZIII, IIIIIIZZI, IIIIIIIZZ, \\ XXXXXXIII, IIIIIIXXX\}. \tag{7.1}$$

By observation, the following set of 3 stabilizers is constructed by composing stabilizer generators, which has the property that at least 2 different Pauli operators are measured on every qubit:

$$\begin{aligned} ZZIIIIIII \circ IIIZZIIII \circ IIIIIIIZZ \circ IIIXXXXXX \circ XXXXXXIII &= YYXZZIXYY \\ ZZIIIIIII \circ IIIZZIIII \circ IIIIIIZZI \circ IIIXXXXXX &= ZZIYYXYYX \\ IZZIIIIII \circ IIIIZZIII \circ IIIIIIIZZ \circ XXXXXXIII &= XYYXYYIZZ \end{aligned} \tag{7.2}$$

The second code considered is the $[\![15,7,3]\!]$ code belonging to the quantum Hamming code family [1],[25], to which the Steane code also belongs. This code has the following 8 weight-8 stabilizer generators:

$$\{IIIIIIIZZZZZZZZ, IIIZZZZIIIIZZZZ, IZZIIZZIIZZIIZZ, ZIZIZIZIZIZIZIZ, \\ IIIIIIXXXXXXXX, IIIXXXXIIIIXXXX, IXXIIXXIIXXIIXX, XIXIXIXIXIXIXIX.\} \tag{7.3}$$

For this code as well, certain sequences of 4 stabilizers can be constructed, which measure at least 2 different Paulis on most qubits, presented below:

$$\{IIIZZZZXXXXYYYY, \\ ZYXZIXYZIXYIZYX, \\ IZZZZIIYYXXXXYY, \\ YYIYIIYXZZXZXXZ\} \tag{7.4}$$

$$\{IIIIIIIYYYYYYYY, \\ YYIXZZXZXXZYIIY, \\ YXZIYXZZXYIZXYI, \\ XZYZYIXIXZYZYIX\} \tag{7.5}$$

These examples and the group-theoretic observations presented previously may be useful to explore a potential generalization to other codes.

8

Conclusion and Outlook

8.1. Conclusion

In this work, three new flag fault tolerance protocols with reduced stabilizer measurements have been developed for the $[\![5,1,3]\!]$ code and the Steane code. The reductions have been achieved by adapting the techniques behind the shorter stabilizer sequences for Shor-style error correction proposed in [27] – namely, adaptive stabilizer measurements, utilizing the complete stabilizer group, particularly high-weight stabilizers, and designing a fault-tolerant ordering for stabilizer measurements –- in the context of flag fault tolerance protocols for these codes developed in [25]. The main contributions of this thesis are summarized.

In Sec. 5, two new and shorter flag protocols of the split-and-diagnose type, one for the $[\![5,1,3]\!]$ code and one for the Steane code have been derived. These shorter constructions have been achieved by tailoring unflagged stabilizer sequences to distinguish specific sets of errors signaled by respective flagged measurements, with nontrivial syndromes. These stabilizers are decided based on the support of the error set, and may belong to the full stabilizer group. The key feature of the split-and-diagnose protocols is the unique approach to dynamically select a different stabilizer to be measured, conditioned on a previous stabilizer measurement outcome. This enables identifying the eigenspaces of stabilizer generators, resulting from different errors, more efficiently by requiring fewer stabilizer measurements.

Additionally, in Sec. 5, the third new and shorter flag fault tolerance protocol of this thesis, the detect-and-diagnose protocol for the Steane code, has been derived. This protocol achieves a reduction in flagged stabilizer measurements by identifying and utilizing fewer high-weight stabilizers, sufficient to detect 1 input error, for flagged measurements. This protocol introduces a new approach to performing fault-tolerant quantum error correction, wherein errors are initially detected by (flagged) stabilizer measurements without the complete ability to correct them, and subsequently corrected via unflagged stabilizer generator measurements.

Fault-tolerant error correction rules of the new protocols, under the specified error model, have been established by the rigorously calculated look-up tables presented in the Appendix E. The newly developed protocols offer significant reduction in circuit depth, as described in Sec. 5. In Sec. 6, results from numerical Monte Carlo simulations are used to demonstrate nontrivial pseudothresholds for the three new protocols under Knill's error model. The pseudothresholds of new flag fault tolerance protocols are shown to be competitive with those of the existing protocols [25].

In Sec. 7, the construction of additional reduced fault tolerant sequences, for protocols analogous in structure to the new protocols, and for other codes, has been briefly investigated. By employing computer searches as an assisting tool, numerous other candidate stabilizer sequences for split-and-diagnose protocols have been obtained for the $[\![5,1,3]\!]$ code and the Steane code. Further, computer searches have been used identify other stabilizer triplets which are capable of detecting a single input error for the Steane code, and form candidate stabilizer sequences for the detect-and-diagnose protocol. Additionally, these stabilizer triplets may be of interest independent of flag protocols, motivating further analysis. These programs have also been used to assist in making certain interesting observations about mathematical properties of such stabilizers for the Steane code. In addition, examples of high-weight stabilizers capable of detecting 1 error have also been presented for 2 other distance-3 codes.

In conclusion, the development of the protocols in this thesis highlights the potential of employing parity measurements from the complete stabilizer group, and extending beyond conventional adaptive stabilizer measurements present in existing flag protocols [25], to reduce the circuit depth overhead in fault-tolerant quantum error correction. These constructions serve as a preliminary step in improving the resource efficiency of quantum error correction, which is particularly valuable for applications in the NISQ era, where resources are limited, as discussed earlier in Sec. 1.

8.2. Outlook

While the flag-based syndrome extraction circuits developed in this thesis may be valuable from the perspective of resource efficiency, there are other aspects which may need to be addressed, and are beyond the scope of the present work. These may form future research directions arising from this work, or challenges yet to be overcome.

First, the pseudothresholds for the new protocols have been established under Knill's error model, under a single cycle, as well as consecutive cycles of error correction. This error model contains state preparation errors, measurement errors, two-qubit gate depolarizing errors (which may lead to error propagation, as well as weight-1 errors on data qubits), and can be used to assess fault tolerance under all these errors. However, it is not a commonly used error model [15], and has been used in this thesis to establish a comparison with the main reference [25]. Additionally, theoretical arguments have been developed in Sec. 5 to demonstrate fault tolerance of the new protocols when errors may be present at every location. To further investigate the fault tolerance of these protocols, pseudothresholds need to be established under more more extensive noise models, including circuit level noise with varying levels of idling error rate [15], as well as under physically realistic noise models.

Additionally, the proposed protocols have been developed from an abstract standpoint, and do not incorporate constraints or limitations on implementation posed by physical hardware [5]. For instance, while stabilizers belonging to the full group allow measuring parities other than the generators, they may involve Pauli operations which may not be natively supported by the hardware. Implementing such operations would require a decomposition into the supported gate set, which may contribute to an increase in gate overhead. This may be analyzed further for specific hardware platforms.

Another consideration for physical implementation is that the additional adaptive stabilizer measurement in split-and-diagnose protocols may increase the time required for one cycle of error correction, due to the delay required for the deciding measurement outcome to become available. This may consume precious qubit coherence time. At the outset, however, this does not seem to pose a major drawback to the implementation of the additional adaptive measurement. This is because the existing flag fault tolerance protocols [25] already require waiting upon flagged measurement outcomes to decide subsequent stabilizer measurements, and thus rely on fast measurement and reset times. A possible solution for the increased cycle time in case of the split-and-diagnose protocols may be to schedule the measurement of the stabilizer used to decide the subsequent measurement at an earlier time in the sequence.

In addition to taking into account implementation considerations, there is a broader scope for investigation from a higher level of abstraction. The reductions obtained via the shorter, fault-tolerant stabilizer sequences developed for flag protocols have been investigated independently for flagged and unflagged measurements. It may be interesting to study whether a simultaneous reduction of both flagged and unflagged measurements can lead to an even shorter flag protocol. This has not been addressed in the detect-and-diagnose protocol for the Steane code, because of the added complexity due to increased number of errors indicated by a triggered flag, or a nontrivial syndrome. A further question which may be addressed is whether the obtained sequences are optimal, and how optimality of a stabilizer sequence may be defined in the context of fault tolerance.

Moreover, an interesting extension would be a mathematical description which explains the protocols developed. This may be used as a starting point for generalizing such constructions for other codes, including those with higher distance, and for other flag protocols developed in literature (see Sec. 4.5). In particular, the extension to topological codes [15] may be interesting, because of the potential impact of adaptive or high-weight stabilizer measurements on syndrome decoding algorithms.

In addition, further improvements are possible in the performance analyses carried out for new protocols. The analysis for gate overhead reduction, presented in Sec. 5, presents scope for further enhancement.

In the presented analysis, the reduction for split-and-diagnose protocols is computed conditioned on a flag having been triggered or a nontrivial syndrome measurement without flag having been observed. On the other hand, the reduction for the detect-and-diagnose protocols is calculated with respect to number of flagged measurements. A more realistic estimate of gate count reductions for both types of protocols would be derived from the average number of flagged and unflagged stabilizers required to be measured in the limit of large number of error correction cycles. For high-weight stabilizers, the increase in gate count may need to be incorporated into such a calculation to get an accurate estimate of gate overhead reduction.

Additionally, for analytically computing the expected improvements in pseudothreshold values, the reduction in the number of malignant pairs of fault locations may be analyzed, as has been performed in [36]. This may be used as a reference for values obtained by simulation, and is beyond the scope of the present work.

As discussed in Sec. 5, the two new flag protocols for the Steane may not be able to correct some $\mathcal{O}(p^2)$ errors, consisting of an $\mathcal{O}(p)$ X-type error and an $\mathcal{O}(p)$ Z-type error. These may be correctable by the original flag protocol for the Steane code [25]. This may be viewed as a trade-off in exchange for the reduced circuit depth. In the present analysis, the circuit overhead reduction (discussed in Sec. 5) and the competitive pseudothresholds under Knill's error model (see Sec. 6) indicate this expense may be justified for the small codes studied under the present conditions. However, more detailed analysis of error correction performance [15] may be required for further insight.

In addition, the use of computer searches to design fault-tolerant stabilizer sequences has also been briefly explored. These were limited to sequences consisting of few stabilizers. As the size of the sequence or the stabilizer group is increased, these searches may become computationally intractable. This prohibits performing a brute force search for the $[\![15, 7, 3]\!]$ quantum Hamming code, which was briefly discussed. This motivates exploring more sophisticated, automated search algorithms for assisting in developing stabilizer sequences for fault tolerance protocols. These approaches may also be useful in reducing the number of unflagged stabilizers in the detect-and-diagnose protocol for the Steane code (Fig. 5.7).

Therefore, multiple avenues worthy of exploration emerge, spanning theoretical, applied, as well as implementation-focused directions.

References

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[2] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79. DOI: 10.22331/q-2018-08-06-79.

[3] Zijun Chen et al. “Exponential suppression of bit or phase errors with cyclic error correction”. In: *Nature* 595.7867 (July 2021), pp. 383–387. DOI: 10.1038/s41586-021-03588-y. URL: https://doi.org/10.1038/s41586-021-03588-y.

[4] John Preskill. *Lecture Notes for Physics 229: Quantum Information and Computation. Chapter 7: Quantum Error Correction.* California Institute of Technology, 1998. URL: http://theory.caltech.edu/~preskill/ph229/notes/.

[5] Lingling Lao and Carmen G. Almudever. “Fault-tolerant quantum error correction on near-term quantum processors using flag and bridge qubits”. In: *Phys. Rev. A* 101 (3 2020), p. 032333. DOI: 10.1103/PhysRevA.101.032333. URL: https://link.aps.org/doi/10.1103/PhysRevA.101.032333.

[6] E. Knill et al. “Randomized benchmarking of quantum gates”. In: *Phys. Rev. A* 77 (1 2008), p. 012307. DOI: 10.1103/PhysRevA.77.012307. URL: https://link.aps.org/doi/10.1103/PhysRevA.77.012307.

[7] Zhenyu Cai et al. *Quantum Error Mitigation*. 2022. DOI: https://arxiv.org/abs/2210.00921.

[8] Rajeev Acharya et al. “Suppressing quantum errors by scaling a surface code logical qubit”. In: *Nature* 614.7949 (2023), pp. 676–681. DOI: 10.1038/s41586-022-05434-1.

[9] J. F. Marques et al. “Logical-qubit operations in an error-detecting surface code”. In: *Nature Physics* 18.1 (2021), pp. 80–86. DOI: 10.1038/s41567-021-01423-9.

[10] Denis Paiste. *Researchers develop error correction method for quantum computing based on Majorana fermions*. 2016. URL: https://phys.org/news/2016-02-error-method-quantum-based-majorana.html.

[11] Katherine Wright. *Evidence found for a Majorana “cousin”*. 2023. URL: https://physics.aps.org/articles/v16/24.

[12] Panos Aliferis, Daniel Gottesman, and John Preskill. “Quantum Accuracy Threshold for Concatenated Distance-3 Codes”. In: *Quantum Info. Comput.* 6.2 (2006), pp. 97–165.

[13] Daniel Eric Gottesman. *An introduction to quantum error correction and fault-tolerant quantum computation*. Proceedings of Symposia in Applied Mathematics, 2010. DOI: https://doi.org/10.1090/psapm/068. URL: https://arxiv.org/abs/0904.2557.

[14] Daniel Eric Gottesman. “Stabilizer codes and quantum error correction”. PhD thesis. 1997. DOI: https://doi.org/10.48550/arXiv.quant-ph/9705052.

[15] Ashley M. Stephens. “Fault-tolerant thresholds for quantum error correction with the surface code”. In: *Phys. Rev. A* 89 (2 2014), p. 022321. DOI: 10.1103/PhysRevA.89.022321.

[16] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* 52 (4 1995), R2493–R2496. DOI: 10.1103/PhysRevA.52.R2493. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[17] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. “Perfect Quantum Error Correcting Code”. In: *Phys. Rev. Lett.* 77 (1 1996), pp. 198–201. DOI: 10.1103/PhysRevLett.77.198. URL: https://link.aps.org/doi/10.1103/PhysRevLett.77.198.

[18] A. R. Calderbank and Peter W. Shor. “Good quantum error-correcting codes exist”. In: *Phys. Rev. A* 54 (2 1996), pp. 1098–1105. DOI: 10.1103/PhysRevA.54.1098. URL: https://link.aps.org/doi/10.1103/PhysRevA.54.1098.

[19] Andrew Steane. “Multiple-particle interference and quantum error correction”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577. DOI: 10.1098/rspa.1996.0136.

[20] A.Yu. Kitaev. “Fault-tolerant quantum computation by anyons”. In: *Annals of Physics* 303.1 (2003), pp. 2–30. DOI: https://doi.org/10.1016/S0003-4916(02)00018-0. URL: https://www.sciencedirect.com/science/article/pii/S0003491602000180.

[21] C. Ryan-Anderson et al. “Realization of Real-Time Fault-Tolerant Quantum Error Correction”. In: *Phys. Rev. X* 11 (4 2021), p. 041058. DOI: 10.1103/PhysRevX.11.041058. URL: https://link.aps.org/doi/10.1103/PhysRevX.11.041058.

[22] Peter W. Shor. “Fault-tolerant quantum computation”. In: *Proceedings of 37th Conference on Foundations of Computer Science* (1996), pp. 56–65.

[23] Andrew M. Steane. “Active Stabilization, Quantum Computation, and Quantum State Synthesis”. In: *Phys. Rev. Lett.* 78 (11 1997), pp. 2252–2255. DOI: 10.1103/PhysRevLett.78.2252. URL: https://link.aps.org/doi/10.1103/PhysRevLett.78.2252.

[24] Enamuel Knill. “Scalable quantum computing in the presence of large detected-error rates”. In: *Phys. Rev. A* 71 (4 2005), p. 042322. DOI: 10.1103/PhysRevA.71.042322. URL: https://link.aps.org/doi/10.1103/PhysRevA.71.042322.

[25] Rui Chao and Ben W. Reichardt. “Quantum Error Correction with Only Two Extra Qubits”. In: *Phys. Rev. Lett.* 121 (5 2018), p. 050502. DOI: 10.1103/PhysRevLett.121.050502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.050502.

[26] Lukas Postler et al. “Demonstration of fault-tolerant universal quantum gate operations”. In: *Nature* 605 (2022), pp. 675–680. URL: https://www.nature.com/articles/s41586-022-04721-1.

[27] Nicolas Delfosse and Ben W. Reichardt. *Short Shor-style syndrome sequences*. 2020. DOI: https://arxiv.org/abs/2008.05051.

[28] E. Knill. “Quantum computing with realistically noisy devices”. In: *Nature* 434.7029 (2005), pp. 39–44. DOI: 10.1038/nature03350.

[29] Joseph A. Gallian. *Contemporary abstract algebra*. CRC, Taylor & Francis Group, 2021.

[30] Andrew Steane. *A tutorial on quantum error correction*. Proceedings of the International School of Physics “Enrico Fermi”, course CLXII, “Quantum Computers, Algorithms, Chaos”, G. Casati, D. L. Shepelyansky, and P. Zoller, eds., pp. 1–32 (IOS Press, Amsterdam 2006). URL: https://www2.physics.ox.ac.uk/sites/default/files/ErrorCorrectionSteane06.pdf.

[31] John Watrous. *CPSC 519/619: Quantum Computation. Lecture Notes*. University of Calgary, 2006. URL: https://cs.uwaterloo.ca/~watrous/QC-notes/.

[32] Daniel Eric Gottesman. *Errata in “Stabilizer Codes and Quantum Error Correction”. Known errata in PhD thesis.* URL: https://www2.perimeterinstitute.ca/personal/dgottesman/thesis-errata.html.

[33] L. Riesebos et al. “Pauli frames for Quantum Computer Architectures”. In: *Proceedings of the 54th Annual Design Automation Conference 2017* (2017). DOI: 10.1145/3061639.3062300.

[34] Daniel Eric Gottesman. *The Heisenberg Representation of Quantum Computers*. Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics, eds. S. P. Corney, R. Delbourgo, and P. D. Jarvis, pp. 32-43 (Cambridge, MA, International Press, 1999). DOI: https://doi.org/10.48550/arXiv.quant-ph/9807006.

[35] Christopher Chamberland and Michael E. Beverland. “Flag fault-tolerant error correction with arbitrary distance codes”. In: *Quantum* 2 (2018), p. 53. DOI: 10.22331/q-2018-02-08-53.

[36] Akshaya Jayashankar, My Duy Hoang Long, Hui Khoon Ng, and Prabha Mandayam. “Achieving fault tolerance against amplitude-damping noise”. In: *Phys. Rev. Res.* 4 (2 2022), p. 023034. DOI: 10.1103/PhysRevResearch.4.023034. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.4.023034.

[37] Andrew M. Steane. “Space, Time, Parallelism and Noise Requirements for Reliable Quantum Computing”. In: *Quantum Computing, S.L. Braunstein (Ed.)* (1999), pp. 137–151. DOI: https://doi.org/10.1002/3527603093.ch8.

[38] K.M. Svore, A.W. Cross, I.L. Chuang, and A.V. Aho. “A flow-map model for analyzing pseudothresholds in fault-tolerant quantum computing”. In: *Quantum Information and Computation* 6.3 (2006), pp. 193–212. DOI: 10.26421/qic6.3-1.

[39] Rui Chao. *Flag The Faults For Reliable Quantum Computing*. Dissertation Presented to the Faculty Of The Graduate School, University of Southern California, 2020.

[40] Rui Chao and Ben W. Reichardt. “Fault-tolerant quantum computation with few qubits”. In: *npj Quantum Information* 4.1 (2018). DOI: 10.1038/s41534-018-0085-z.

[41] Rui Chao and Ben W. Reichardt. “Flag Fault-Tolerant Error Correction for any Stabilizer Code”. In: *PRX Quantum* 1 (1 2020), p. 010302. DOI: 10.1103/PRXQuantum.1.010302. URL: https://link.aps.org/doi/10.1103/PRXQuantum.1.010302.

[42] Dripto M. Debroy and Kenneth R. Brown. “Extended flag gadgets for low-overhead circuit verification”. In: *Phys. Rev. A* 102 (5 2020), p. 052409. DOI: 10.1103/PhysRevA.102.052409. URL: https://link.aps.org/doi/10.1103/PhysRevA.102.052409.

[43] Pei-Hao Liou and Ching-Yi Lai. “Parallel syndrome extraction with shared flag qubits for Calderbank-Shor-Steane codes of distance three”. In: *Phys. Rev. A* 107 (2 2023), p. 022614. DOI: 10.1103/PhysRevA.107.022614. URL: https://link.aps.org/doi/10.1103/PhysRevA.107.022614.

[44] Benjamin Anker and Milad Marvian. “Flag Gadgets based on Classical Codes”. In: (2022). DOI: https://doi.org/10.48550/arXiv.2212.10738.

[45] Nicolas Delfosse, Ben W. Reichardt, and Krysta M. Svore. “Beyond Single-Shot Fault-Tolerant Quantum Error Correction”. In: *IEEE Transactions on Information Theory* 68.1 (2022), pp. 287–301. DOI: 10.1109/TIT.2021.3120685.

[46] Yuichiro Fujiwara. “Ability of stabilizer quantum error correction to protect itself from its own imperfection”. In: *Phys. Rev. A* 90 (6 2014), p. 062304. DOI: 10.1103/PhysRevA.90.062304. URL: https://link.aps.org/doi/10.1103/PhysRevA.90.062304.

[47] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 1)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1. 2022.

[48] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.

A

Definitions: Classical Error-Correcting Codes

These definitions, useful as a reference for the CSS construction (see Sec. 3.2), are presented here. The main reference is [1].

Definition A.1 *A **binary linear code** C, encoding strings of k bits into strings of n bits, with $k < n$, is a specifically chosen k-dimensional subspace of the linear vector space $\mathbb{F}_2^n$ over the binary field $\mathbb{F}_2$. The binary field is a set of 2 elements, typically denoted by 0 and 1, equipped with the operations of addition and multiplication, both modulo 2. The encoded bit strings are called codewords, and a linear combination of codewords gives another codeword.*

Definition A.2 *A binary linear code is specified by a **generator matrix** $G \in \mathbb{F}_2^{n\times k}$, which maps a k-bit string y as Gy. The map is called an **encoding**. The rows of G form a basis for C. Equivalently, the code can be defined as the null space/kernel of a **parity check matrix** $H \in \mathbb{F}_2^{(n-k)\times n}$, i.e. the code is $C = \{x \in \mathbb{F}_2^n : Hx = 0\}$. For a given code, these matrices satisfy $HG = 0$.*

Definition A.3 *The **Hamming weight** of an n-bit string is the number of non-zero entries in the string. The **Hamming distance** between two n-bit strings, a and b, denoted $d(a, b)$, is the Hamming weight of $(a + b)$, where the addition is modulo 2.*

Definition A.4 *The **distance** of a code C, denoted by d, is the minimum Hamming distance between any two codewords. For a linear code, this is equal to the minimum Hamming weight of a non-zero codeword. A distance d code can correct up to $t = \lceil \frac{d-1}{2} \rceil$ errors; the code is then said to be t-error-correcting.*

Definition A.5 *A code C encoding k bits of information into n bits, and having distance d is denoted as an $[n, k, d]$ code.*

The dual of a code is used in the CSS construction (see Sec. 3.2).

Definition A.6 *The **dual** of a code $C \subset \mathbb{F}_2^n$, denoted $C^\perp$, is the orthogonal complement C, i.e. $C^\perp = \{w : w^T x = 0, \forall x \in C\}$, where $w^T x = \sum_{i=1}^n w_i x_i$ is an inner product, with the addition being defined modulo 2 over the binary field. $C^\perp$ has a dimension equal to $n - dim(C)$. It holds that $(C^\perp)^\perp = C$.*

Definition A.7 *A linear code C is **self-dual**, if it is equal to its dual, i.e. $C = C^\perp$. A linear code C is **weakly self-dual**, if it is contained in its dual, i.e. $C \subseteq C^\perp$.*

Definition A.8 *(Classical, binary) **Hamming codes** are a family of $[2^r - 1, 2^r - r - 1, 3]$ binary linear codes for $r \geq 2$. Hamming codes are perfect codes.*

For example, the $[7, 4, 3]$ Hamming code $(r = 3)$, used in the construction of Steane code (see Sec. 3.3), has the following parity check matrix:

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{A.1}$$

B

Stabilizer Groups: the $[\![5, 1, 3]\!]$ code and the Steane code

The 16 elements of the full stabilizer group for the $[\![5, 1, 3]\!]$ code are presented in eq. B.1.

$$IIIII, XZZXI, IXZZX, XIXZZ, ZXIXZ, XYIYX, IZYYZ, YYZIZ, XXYIY, ZIZYY, YXXYI, IYXXY, YZIZY, ZYYZI, YIYXX, ZZXIX. \quad \text{(B.1)}$$

The 64 elements of the full stabilizer group for the Steane code are presented in eq. B.2. They have been labelled according to their type (the trivial stabilizer, stabilizers containing only X, Y or Z operators, and high-weight stabilizers (which contain 2 non-identity Paulis of each type)).

trivial:

$IIIIIII$

X-type:

$IIIXXXX, IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI, XXIIXXI, XXIXIIX$

Z-type:

$IIIZZZZ, IZZIIZZ, ZIZIZIZ, IZZZZII, ZIZZIZI, ZZIIZZI, ZZIZIIZ$

Y-type:

$IIIYYYY, IYYIIYY, YIYIYIY, IYYYYII, YIYYIYI, YYIIYYI, YYIYIIY$

mixed/high-weight:

$$-IZZXXYY, -ZIZXYXY, -IXXZZYY, -ZXYIZXY, -XIXZYZY, -XZYIXZY,$$
$$-IXXYYZZ, -IYYXXZZ, -ZXYXYIZ, -XIXYZYZ, -XZYXIYZ, -YIYXZXZ,$$
$$-IZZYYXX, -ZIZYXYX, -ZZIXYYX, -XXIZYYZ, -XYZIXYZ, -YXZIYXZ,$$
$$-IYYZZXX, -ZXYZIYX, -ZYXIZYX, -XZYZYIX, -YIYZXZX, -YZXIYZX,$$
$$-XXIYZZY, -XYZXIZY, -YXZXZIY, -ZXYYXZI, -ZYXXYZI, -XZYYZXI,$$
$$-YZXXZYI, -ZZIYXXY, -XYZZYXI, -YXZZXYI, -ZYXZIXY, -YZXZXIY,$$
$$-XYZYZIX, -YXZYIZX, -YYIXZZX, -ZYXYXIZ, -YZXYIXZ, -YYIZXXZ. \quad \text{(B.2)}$$

C

Errors Arising in Flag Protocols

Causal fault	Propagated error on data qubits	Syndrome from unflagged stabilizer measurements
IZ fault after CNOT b or ZZ fault after CNOT c	$IIZXI$	0100
XZ fault after CNOT b	$IXZXI$	1100
YZ fault after CNOT b	$IYZXI$	1001
ZZ fault after CNOT b	$IZZXI$	0001
IZ fault after CNOT c	$IIIXI$	0110
XZ fault after CNOT c	$IIXXI$	1010
YZ fault after CNOT c	$IIYXI$	1000

Table C.1: Errors on data qubits from bad gates during flagged $XZZXI$ measurement, stabilizer generator of the $[\![5,1,3]\!]$ code, using the circuit shown in Fig. 4.2. The first column denotes the fault which can cause the flag to get triggered. The second column denotes the resulting error on data qubits due to propagation. The third column denotes the corresponding syndrome from measuring all stabilizer generators with unflagged circuits.

Causal fault	Propagated error on data qubits	Syndrome from unflagged stabilizer measurements
IZ fault after CNOT b or ZZ fault after CNOT c	$IIIZX$	1010
XZ fault after CNOT b	$IIXZX$	0110
YZ fault after CNOT b	$IIYZX$	0100
ZZ fault after CNOT b	$IIZZX$	1000
IZ fault after CNOT c	$IIIIX$	0011
XZ fault after CNOT c	$IIIXX$	0101
YZ fault after CNOT c	$IIYXX$	1100

Table C.2: Errors on data qubits from bad gates during flagged $IXZZX$ measurement, stabilizer generator of the $[\![5,1,3]\!]$ code, using the circuit and gate annotations shown in Fig. 4.5.

Causal fault	Propagated error on data qubits	Syndrome from unflagged stabilizer measurements
IZ fault after XNOT b or ZZ fault after CNOT c	$IIIZZ$	1101
XZ fault after XNOT b	$IIXZZ$	0001
YZ fault after XNOT b	$IIYZZ$	0011
ZZ fault after XNOT b	$IIZZZ$	1111
IZ fault after CNOT c	$IIIIZ$	0100
XZ fault after CNOT c	$IIIXZ$	0010
YZ fault after CNOT c	$IIIYZ$	1011

Table C.3: Errors on data qubits from bad gates during flagged $XIXZZ$ measurement, stabilizer generator of the $[\![5,1,3]\!]$ code, using the circuit and gate annotations shown in Fig. 4.5.

Causal fault	Propagated error on data qubits	Syndrome from unflagged stabilizer measurements
IZ fault after XNOT b or XZ fault after XNOT c	$IIIXZ$	0010
XZ fault after XNOT b	$IXIXZ$	1010
YZ fault after XNOT b	$IYIXZ$	1111
ZZ fault after XNOT b	$IZIXZ$	0111
IZ fault after XNOT c	$IIIIZ$	0100
YZ fault after XNOT c	$IIIYZ$	1011
ZZ fault after XNOT c	$IIIZZ$	1101

Table C.4: Errors on data qubits from bad gates during flagged $ZXIXZ$ measurement, stabilizer generator of the $[\![5,1,3]\!]$ code, using the circuit and gate annotations shown in Fig. 4.5.

Causal fault	Propagated error on data qubits	Syndrome from unflagged X-type stabilizer measurements	Syndrome from unflagged Z-type stabilizer measurements
IZ fault after XNOT b or XZ fault after XNOT c	$IIIIIXX$	000	001
XZ fault after XNOT b	$IIIIXXX$	000	100
YZ fault after XNOT b	$IIIIYXX$	101	100
ZZ fault after XNOT b	$IIIIZXX$	101	001
IZ fault after XNOT c	$IIIIIIX$	000	111
YZ fault after XNOT c	$IIIIIYX$	110	001
ZZ fault after XNOT c	$IIIIIZX$	110	111

Table C.5: Errors on data qubits from bad gates during flagged $IIIXXXX$ measurement, stabilizer generator of the Steane code, using the circuit and gate annotations shown in Fig. 4.7(a). Syndromes from X- and Z- type stabilizer generator measurements are shown separately.

Causal fault	Propagated error on data qubits	Syndrome from unflagged X-type stabilizer measurements	Syndrome from unflagged Z-type stabilizer measurements
IZ fault after XNOT b or XZ fault after XNOT c	$IIIIIXX$	000	001
XZ fault after XNOT b	$IIXIIXX$	000	010
YZ fault after XNOT b	$IIYIIXX$	011	010
ZZ fault after XNOT b	$IIZIIXX$	011	001
IZ fault after XNOT c	$IIIIIIX$	000	111
YZ fault after XNOT c	$IIIIIYX$	110	001
ZZ fault after XNOT c	$IIIIIZX$	110	111

Table C.6: Errors on data qubits from bad gates during flagged $IXXIIXX$ measurement, stabilizer generator of the Steane code, using the circuit and gate annotations shown in Fig. 4.7. Syndromes from X- and Z- type stabilizer generator measurements are shown separately.

Causal fault	Propagated error on data qubits	Syndrome from unflagged X-type stabilizer measurements	Syndrome from unflagged Z-type stabilizer measurements
IZ fault after XNOT b or XZ fault after XNOT c	$IIIIXIX$	000	010
XZ fault after XNOT b	$IIXIXIX$	000	001
YZ fault after XNOT b	$IIYIXIX$	011	001
ZZ fault after XNOT b	$IIZIXIX$	011	010
IZ fault after XNOT c	$IIIIIIX$	000	111
YZ fault after XNOT c	$IIIIYIX$	001	010
ZZ fault after XNOT c	$IIIIZIX$	101	111

Table C.7: Errors on data qubits from bad gates during flagged $XIXIXIX$ measurement, stabilizer generator of the Steane code, using the circuit and gate annotations shown in Fig. 4.7. Syndromes from X- and Z- type stabilizer generator measurements are shown separately.

Causal fault	Propagated error on data qubits	Syndrome from unflagged X-type stabilizer measurements	Syndrome from unflagged Z-type stabilizer measurements
IZ fault after CNOT b or ZZ fault after CNOT c	$IIIIIZZ$	001	000
XZ fault after CNOT b	$IIIIXZZ$	001	101
YZ fault after CNOT b	$IIIIYZZ$	100	101
ZZ fault after CNOT b	$IIIIZZZ$	100	000
IZ fault after CNOT c	$IIIIIIZ$	111	000
XZ fault after CNOT c	$IIIIIXZ$	111	110
YZ fault after CNOT c	$IIIIIYZ$	001	110

Table C.8: Errors on data qubits from bad gates during flagged $IIIZZZZ$ measurement, stabilizer generator of the Steane code, using the circuit and gate annotations shown in Fig. 4.7. Syndromes from X- and Z- type stabilizer generator measurements are shown separately.

Causal fault	Propagated error on data qubits	Syndrome from unflagged X-type stabilizer measurements	Syndrome from unflagged Z-type stabilizer measurements
IZ fault after CNOT b or ZZ fault after CNOT c	$IIIIIZZ$	001	000
XZ fault after CNOT b	$IIXIIZZ$	001	011
YZ fault after CNOT b	$IIYIIZZ$	010	011
ZZ fault after CNOT b	$IIZIIZZ$	010	000
IZ fault after CNOT c	$IIIIIIZ$	111	000
XZ fault after CNOT c	$IIIIIXZ$	111	110
YZ fault after CNOT c	$IIIIIYZ$	001	110

Table C.9: Errors on data qubits from bad gates during flagged $IZZIIZZ$ measurement, stabilizer generator of the Steane code, using the circuit and gate annotations shown in Fig. 4.7. Syndromes from X- and Z- type stabilizer generator measurements are shown separately.

Causal fault	Propagated error on data qubits	Syndrome from unflagged X-type stabilizer measurements	Syndrome from unflagged Z-type stabilizer measurements
IZ fault after CNOT b or ZZ fault after CNOT c	$IIIIZIZ$	010	000
XZ fault after CNOT b	$IIXIZIZ$	010	011
YZ fault after CNOT b	$IIYIZIZ$	001	011
ZZ fault after CNOT b	$IIZIZIZ$	001	000
IZ fault after CNOT c	$IIIIIIZ$	111	000
XZ fault after CNOT c	$IIIIXIZ$	111	101
YZ fault after CNOT c	$IIIIYIZ$	010	101

Table C.10: Errors on data qubits from bad gates during flagged $ZIZIZIZ$ measurement, stabilizer generator of the Steane code, using the circuit and gate annotations shown in Fig. 4.7. Syndromes from X- and Z- type stabilizer generator measurements are shown separately.

D

Syndromes from Reduced Stabilizer Sequences

Propagated error on data qubits	Syndrome from $XZZXI$ measurement	Syndrome from $YXXYI$ measurement	Syndrome from third stabilizer measurement
$IIZXI$	0	0	$(ZIZYY):1$
$IXZXI$	1	0	$(ZIZYY):1$
$IYZXI$	1	1	$(XIXZZ):0$
$IZZXI$	0	1	$(XIXZZ):0$
$IIIXI$	0	1	$(XIXZZ):1$
$IIXXI$	1	1	$(XIXZZ):1$
$IYXXI$	1	0	$(ZIZYY):0$

Table D.1: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $XZZXI$ measurement (see Table 4.1), using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the $[\![5, 1, 3]\!]$ code in Fig. 5.5. Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $IXZZX$ measurement	Syndrome from $IYXXY$ measurement	Syndrome from third stabilizer measurement
$IIIZX$	0	0	$(YZIZY):1$
$IIXZX$	1	0	$(YZIZY):1$
$IIYZX$	1	1	$(ZXIXZ):0$
$IIZZX$	0	1	$(ZXIXZ):0$
$IIIIX$	0	1	$(ZXIXZ):1$
$IIIXX$	1	1	$(ZXIXZ):1$
$IIIYX$	1	0	$(YZIZY):0$

Table D.2: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $IXZZX$ measurement (see Table C.2), using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the $[\![5, 1, 3]\!]$ code in Fig. 5.5. Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $XIXZZ$ measurement	Syndrome from $YIYXX$ measurement	Syndrome from third stabilizer measurement
$IIIZZ$	0	0	$(YZIZY):1$
$IIXZZ$	0	1	$(ZXIXZ):1$
$IIYZZ$	1	0	$(YZIZY):1$
$IIZZZ$	1	1	$(ZXIXZ):1$
$IIIIZ$	0	1	$(ZXIXZ):0$
$IIIXZ$	1	1	$(ZXIXZ):0$
$IIIYZ$	1	0	$(YZIZY):0$

Table D.3: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $XIXZZ$ measurement (see Table C.3), using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the $[\![5, 1, 3]\!]$ code in Fig. 5.5. Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $ZXIXZ$ measurement	Syndrome from $YZIZY$ measurement	Syndrome from third stabilizer measurement
$IIIXZ$	0	0	$(YIYXX):1$
$IXIXZ$	0	1	$(XIXZZ):1$
$IYIXZ$	1	1	$(XIXZZ):1$
$IZIXZ$	1	0	$(YIYXX):1$
$IIIIZ$	0	1	$(XIXZZ):0$
$IIIYZ$	1	0	$(YIYXX):0$
$IIIZZ$	1	1	$(XIXZZ):0$

Table D.4: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $ZXIXZ$ measurement (see Table C.4), using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the $[\![5, 1, 3]\!]$ code in Fig. 5.5. Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $IIIXXXX$ measurement	Syndrome from $IIIZZZZ$ measurement	Syndrome from third stabilizer measurement
$IIIIIXX$	0	0	$(ZIZIZIZ):1$
$IIIIXXX$	0	1	$(ZIZIZIZ):0$
$IIIIYXX$	1	1	$(XIXIXIX):1$
$IIIIZXX$	1	0	$(XIXIXIX):1$
$IIIIIIX$	0	1	$(ZIZIZIZ):1$
$IIIIIYX$	1	0	$(XIXIXIX):0$
$IIIIIZX$	1	1	$(XIXIXIX):0$

Table D.5: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $IIIXXXX$ measurement (see Table 4.3) using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the Steane code in Fig. 5.6. The Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $IXXIIXX$ measurement	Syndrome from $IZZIIZZ$ measurement	Syndrome from third stabilizer measurement
$IIIIIXX$	0	0	$(ZIZIZIZ):1$
$IIXIIXX$	0	1	$(ZIZIZIZ):0$
$IIYIIXX$	1	1	$(XIXIXIX):1$
$IIZIIXX$	1	0	$(XIXIXIX):1$
$IIIIIIX$	0	1	$(ZIZIZIZ):1$
$IIIIIYX$	1	0	$(XIXIXIX):0$
$IIIIIZX$	1	1	$(XIXIXIX):0$

Table D.6: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $IXXIIXX$ measurement (see Table C.6) using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the Steane code in Fig. 5.6. The Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $XIXIXIX$ measurement	Syndrome from $ZIZIZIZ$ measurement	Syndrome from third stabilizer measurement
$IIIXIX$	0	0	$(IZZIIZZ):1$
$IIXIXIX$	0	1	$(IZZIIZZ):0$
$IIYIXIX$	1	1	$(IXXIIXX):1$
$IIZIXIX$	1	0	$(IXXIIXX):1$
$IIIIIIX$	0	1	$(IZZIIZZ):1$
$IIIIYIX$	1	0	$(IXXIIXX):0$
$IIIIZIX$	1	1	$(IXXIIXX):0$

Table D.7: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $XIXIXIX$ measurement (see Table C.7) using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the Steane code in Fig. 5.6. The Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $IIIZZZZ$ measurement	Syndrome from $IIIXXXX$ measurement	Syndrome from third stabilizer measurement
$IIIIIZZ$	0	0	$(XIXIXIX):1$
$IIIIXZZ$	1	0	$(ZIZIZIZ):1$
$IIIIYZZ$	1	1	$(ZIZIZIZ):1$
$IIIIZZZ$	0	1	$(XIXIXIX):0$
$IIIIIIZ$	0	1	$(XIXIXIX):1$
$IIIIIXZ$	1	1	$(ZIZIZIZ):0$
$IIIIIYZ$	1	0	$(ZIZIZIZ):0$

Table D.8: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $IIIZZZZ$ measurement (see Table C.8) using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the Steane code in Fig. 5.6. The Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $IZZIIZZ$ measurement	Syndrome from $IXXIIXX$ measurement	Syndrome from third stabilizer measurement
$IIIIIZZ$	0	0	$(XIXIXIX):1$
$IIXIIZZ$	1	0	$(ZIZIZIZ):1$
$IIYIIZZ$	1	1	$(ZIZIZIZ):1$
$IIZIIZZ$	0	1	$(XIXIXIX):0$
$IIIIIIZ$	0	1	$(XIXIXIX):1$
$IIIIIXZ$	1	1	$(ZIZIZIZ):0$
$IIIIIYZ$	1	0	$(ZIZIZIZ):0$

Table D.9: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $IZZIIZZ$ measurement (see Table C.9) using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the Steane code in Fig. 5.6. The Pauli operators acting on the deciding qubits have been shown in red.

Propagated error on data qubits	Syndrome from $ZIZIZIZ$ measurement	Syndrome from $XIXIXIX$ measurement	Syndrome from third stabilizer measurement
$IIIZIZ$	0	0	$(IXXIIXX):1$
$IIXIZIZ$	1	0	$(IZZIIZZ):1$
$IIYIZIZ$	1	1	$(IZZIIZZ):1$
$IIZIZIZ$	0	1	$(IXXIIXX):0$
$IIIIIIZ$	0	1	$(IXXIIXX):1$
$IIIIXIZ$	1	1	$(IZZIIZZ):0$
$IIIIYIZ$	1	0	$(IZZIIZZ):0$

Table D.10: Unique and nontrivial syndromes obtained for errors which trigger the flag during flagged $ZIZIZIZ$ measurement (see Table C.10) using the reduced stabilizer sequence used in the corresponding $f = 1$ branch of the split-and-diagnose protocol for the Steane code in Fig. 5.6. The Pauli operators acting on the deciding qubits have been shown in red.

Error	Syndrome from $IIIXXXX$ measurement	Syndrome from $IXXIIXX$ measurement	Syndrome from $XIXIXIX$ measurement	Syndrome from $IIIZZZZ$ measurement
$IIIZIII$	1	0	0	0
$IIIIZII$	1	0	1	0
$IIIIIZI$	1	1	0	0
$IIIIIIZ$	1	1	1	0
$IIIYIII$	1	0	0	1
$IIIIYII$	1	0	1	1
$IIIIIYI$	1	1	0	1
$IIIIIIY$	1	1	1	1

Table D.11: Unique and nontrivial syndromes computed using the reduced stabilizer sequence used in the split-and-diagnose protocol in Fig. 5.6 for errors which give a nontrivial syndrome without flag from the flagged $IIIXXXX$ measurement, and are diagnosed by the $(s, f) = (1, 0)$ branch.

Error	Syndrome from $IIIXXXX$ measurement	Syndrome from $IXXIIXX$ measurement	Syndrome from $XIXIXIX$ measurement	Syndrome from $IZZIIZZ$ measurement
$IZIIIII$	0	1	0	0
$IIZIIII$	0	1	1	0
$IIIIIZI$	1	1	0	0
$IIIIIIZ$	1	1	1	0
$IYIIIII$	0	1	0	1
$IIYIIII$	0	1	1	1
$IIIIIYI$	1	1	0	1
$IIIIIIY$	1	1	1	1

Table D.12: Unique and nontrivial syndromes computed using the reduced stabilizer sequence used in the split-and-diagnose protocol in Fig. 5.6 for errors which give a nontrivial syndrome without flag from the flagged $IXXIIXX$ measurement, and are diagnosed by the $(s, f) = (1, 0)$ branch.

Error	Syndrome from $IIIXXXX$ measurement	Syndrome from $IXXIIXX$ measurement	Syndrome from $XIXIXIX$ measurement	Syndrome from $ZIZIZIZ$ measurement
$ZIIIIII$	0	0	1	0
$IIZIIII$	0	1	1	0
$IIIIZII$	1	0	1	0
$IIIIIIZ$	1	1	1	0
$YIIIIII$	0	0	1	1
$IIYIIII$	0	1	1	1
$IIIIYII$	1	0	1	1
$IIIIIIY$	1	1	1	1

Table D.13: Unique and nontrivial syndromes computed using the reduced stabilizer sequence used in the split-and-diagnose protocol in Fig. 5.6 for errors which give a nontrivial syndrome without flag from the flagged $XIXIXIX$ measurement, and are diagnosed by the $(s, f) = (1, 0)$ branch.

Error	Syndrome from $IIIZZZZ$ measurement	Syndrome from $IZZIIZZ$ measurement	Syndrome from $ZIZIZIZ$ measurement	Syndrome from $IIIXXXX$ measurement
$IIIXIII$	1	0	0	0
$IIIIXII$	1	0	1	0
$IIIIIXI$	1	1	0	0
$IIIIIIX$	1	1	1	0
$IIIYIII$	1	0	0	1
$IIIIYII$	1	0	1	1
$IIIIIYI$	1	1	0	1
$IIIIIIY$	1	1	1	1

Table D.14: Unique and nontrivial syndromes computed using the reduced stabilizer sequence used in the split-and-diagnose protocol in Fig. 5.6 for errors which give a nontrivial syndrome without flag from the flagged $IIIZZZZ$ measurement, and are diagnosed by the $(s, f) = (1, 0)$ branch.

Error	Syndrome from $IIIZZZZ$ measurement	Syndrome from $IZZIIZZ$ measurement	Syndrome from $ZIZIZIZ$ measurement	Syndrome from $IXXIIXX$ measurement
$IXIIIII$	0	1	0	0
$IIXIIII$	0	1	1	0
$IIIIIXI$	1	1	0	0
$IIIIIIX$	1	1	1	0
$IYIIIII$	0	1	0	1
$IIYIIII$	0	1	1	1
$IIIIIYI$	1	1	0	1
$IIIIIIY$	1	1	1	1

Table D.15: Unique and nontrivial syndromes computed using the reduced stabilizer sequence used in the split-and-diagnose protocol in Fig. 5.6 for errors which give a nontrivial syndrome without flag from the flagged $IZZIIZZ$ measurement, and are diagnosed by the $(s, f) = (1, 0)$ branch.

Error	Syndrome from $IIIZZZZ$ measurement	Syndrome from $IZZIIZZ$ measurement	Syndrome from $ZIZIZIZ$ measurement	Syndrome from $XIXIXIX$ measurement
$XIIIIII$	0	0	1	0
$IIXIIII$	0	1	1	0
$IIIIXII$	1	0	1	0
$IIIIIIX$	1	1	1	0
$YIIIIII$	0	0	1	1
$IIYIIII$	0	1	1	1
$IIIIYII$	1	0	1	1
$IIIIIIY$	1	1	1	1

Table D.16: Unique and nontrivial syndromes computed using the reduced stabilizer sequence used in the split-and-diagnose protocol in Fig. 5.6 for errors which give a nontrivial syndrome without flag from the flagged $ZIZIZIZ$ measurement, and are diagnosed by the $(s, f) = (1, 0)$ branch.

Causal fault	Propagated error	Syndrome from X-type stabilizer measurements	Z correction	Minimum weight equivalent	Syndrome from Z-type stabilizer measurements	X correction	Minimum weight equivalent
IZ after gate b, or, XZ after gate c	$IIIXXYY$	001	$IIIIIZZ$	$IIIIIZZ$	000	$IIIXXXX$	$IIIIIII$
XZ after gate b	$IIXXXYY$	001	$IIIIIZZ$	$IIIIIZZ$	011	$IIXXXXX$	$IIXIIII$
YZ after gate b	$IIYXXYY$	010	$IIZIIZZ$	$IZIIIII$	011	$IIXXXXX$	$IIXIIII$
ZZ after gate b	$IIZXXYY$	010	$IIZIIZZ$	$IZIIIII$	000	$IIIXXXX$	$IIIIIII$
IZ after gate c, or, XZ after gate d	$IIIIXYY$	001	$IIIIIZZ$	$IIIIIZZ$	100	$IIIIXXX$	$IIIXIII$
YZ after gate c	$IIIYXYY$	101	$IIIZIZZ$	$IIIIZII$	000	$IIIXXXX$	$IIIIIII$
ZZ after gate c	$IIIZXYY$	101	$IIIZIZZ$	$IIIIZII$	100	$IIIIXXX$	$IIIXIII$
IZ after gate d, or, YZ after gate e	$IIIIIYY$	001	$IIIIIZZ$	$IIIIIZZ$	001	$IIIIIXX$	$IIIIIXX$
YZ after gate d	$IIIIYYY$	100	$IIIIZZZ$	$IIIZIII$	100	$IIIIXXX$	$IIIXIII$
ZZ after gate d	$IIIIZYY$	100	$IIIIZZZ$	$IIIZIII$	001	$IIIIIXX$	$IIIIIXX$
IZ after gate e	$IIIIIIY$	111	$IIIIIIZ$	$IIIIIIZ$	111	$IIIIIIX$	$IIIIIIX$
XZ after gate e	$IIIIIXY$	111	$IIIIIIZ$	$IIIIIIZ$	001	$IIIIIXX$	$IIIIIXX$
ZZ after gate e	$IIIIIZY$	001	$IIIIIZZ$	$IIIIIZZ$	111	$IIIIIIX$	$IIIIIIX$

Table D.17: Propagated errors on data qubits from faults after bad gates in $IZZXXYY$ measurement, the corresponding syndromes obtained by measuring the 6 standard Steane code generators, and the corrections to be applied. The propagated errors result from the circuit shown in Fig. 5.8, and the gate labels b, c, d, e are referred from the same figure. This table shows that the syndromes obtained are unique for inequivalent errors, and further, this holds separately for the X and Z components of the errors. In addition, the syndromes from X- and Z-type stabilizers are nontrivial when the error has a non-identity Z or X component, respectively. Therefore, the protocol in Fig. 5.7 can correct these propagated errors.

Causal fault	Propagated error	Syndrome from X-type stabilizer measurements	Z correction	Minimum weight equivalent	Syndrome from Z-type stabilizer measurements	X correction	Minimum weight equivalent
IZ after gate b, or, YZ after gate c	$IIIYZYZ$	000	$IIIZZZZ$	$IIIIIII$	010	$IIIXIXI$	$IIIXIXI$
XZ after gate b	$IIXYZYZ$	000	$IIIZZZZ$	$IIIIIII$	001	$IIXXIXI$	$XIIIIII$
YZ after gate b	$IIYYZYZ$	011	$IIZZZZZ$	$IIZIIII$	001	$IIXXIXI$	$XIIIIII$
ZZ after gate b	$IIZYZYZ$	011	$IIZZZZZ$	$IIZIIII$	010	$IIIXIXI$	$IIIXIXI$
IZ after gate c, or, ZZ after gate d	$IIIIZYZ$	100	$IIIIZZZ$	$IIIZIII$	110	$IIIIIXI$	$IIIIIXI$
XZ after gate c	$IIIXZYZ$	100	$IIIIZZZ$	$IIIZIII$	010	$IIIXIXI$	$IIIXIXI$
ZZ after gate c	$IIIZZYZ$	000	$IIIZZZZ$	$IIIIIII$	110	$IIIIIXI$	$IIIIIXI$
IZ after gate d, or, YZ after gate e	$IIIIIYZ$	001	$IIIIIZZ$	$IIIIIZZ$	110	$IIIIIXI$	$IIIIIXI$
XZ after gate d	$IIIIXYZ$	001	$IIIIIZZ$	$IIIIIZZ$	011	$IIIIXXI$	$IIIIXXI$
YZ after gate d	$IIIIYYZ$	100	$IIIIZZZ$	$IIIZIII$	011	$IIIIXXI$	$IIIIXXI$
IZ after gate e	$IIIIIIZ$	111	$IIIIIIZ$	$IIIIIIZ$	000	$IIIIIII$	$IIIIIII$
XZ after gate e	$IIIIIXZ$	111	$IIIIIIZ$	$IIIIIIZ$	110	$IIIIIXI$	$IIIIIXI$
ZZ after gate e	$IIIIIZZ$	001	$IIIIIZZ$	$IIIIIZZ$	000	$IIIIIII$	$IIIIIII$

Table D.18: Propagated errors on data qubits from faults after bad gates in $XIXYZYZ$ measurement, the corresponding syndromes obtained by measuring the 6 standard Steane code generators, and the corrections to be applied. The propagated errors result from the circuit shown in Fig. 5.8, and the gate labels b, c, d, e are referred from the same figure. This table shows that the syndromes obtained are unique for inequivalent errors, and further, this holds separately for the X and Z components of the errors. In addition, the syndromes from X- and Z-type stabilizers are nontrivial when the error has a non-identity Z or X component, respectively. Therefore, the protocol in Fig. 5.7 can correct these propagated errors.

Causal fault	Propagated error	Syndrome from X-type stabilizer measurements	Z correction	Minimum weight equivalent	Syndrome from Z-type stabilizer measurements	X correction	Minimum weight equivalent
IZ after gate b, or, YZ after gate c	$IIYYXZI$	001	$IIZZIZI$	$ZIIIIII$	010	$IIXXXII$	$IXIIIII$
XZ after gate b	$IXYYXZI$	001	$IIZZIZI$	$ZIIIIII$	000	$IXXXXII$	$IIIIIII$
YZ after gate b	$IYYYXZI$	011	$IZZZIZI$	$IIIZIIZ$	000	$IXXXXII$	$IIIIIII$
ZZ after gate b	$IZYYXZI$	011	$IZZZIZI$	$IIIZIIZ$	010	$IIXXXII$	$IXIIIII$
IZ after gate c, or, YZ after gate d	$IIIYXZI$	010	$IIIZIZI$	$IIIZIZI$	001	$IIIXXII$	$IIIXXII$
XZ after gate c	$IIXYXZI$	010	$IIIZIZI$	$IIIZIZI$	010	$IIXXXII$	$IXIIIII$
ZZ after gate c	$IIZYZXI$	001	$IIZZIZI$	$ZIIIIII$	001	$IIIXXII$	$IIIXXII$
IZ after gate d, or, YZ after gate e	$IIIIXZI$	110	$IIIIIZI$	$IIIIIZI$	101	$IIIIXII$	$IIIIXII$
XZ after gate d	$IIIXXZI$	110	$IIIIIZI$	$IIIIIZI$	001	$IIIXXII$	$IIIXXII$
ZZ after gate d	$IIIZXZI$	010	$IIIZIZI$	$IIIZIZI$	101	$IIIIXII$	$IIIIXII$
IZ after gate e	$IIIIIZI$	110	$IIIIIZI$	$IIIIIZI$	000	$IIIIIII$	$IIIIIII$
XZ after gate e	$IIIIXZI$	110	$IIIIIZI$	$IIIIIZI$	101	$IIIIXII$	$IIIIXII$
ZZ after gate e	$IIIIZZI$	011	$IIIIZZI$	$IIIIZZI$	000	$IIIIIII$	$IIIIIII$

Table D.19: Propagated errors on data qubits from faults after bad gates in $ZXYYXZI$ measurement, the corresponding syndromes obtained by measuring the 6 standard Steane code generators, and the corrections to be applied. The propagated errors result from the circuit shown in Fig. 5.8, and the gate labels b, c, d, e are referred from the same figure. This table shows that the syndromes obtained are unique for inequivalent errors, and further, this holds separately for the X and Z components of the errors. In addition, the syndromes from X- and Z-type stabilizers are nontrivial when the error has a non-identity Z or X component, respectively. Therefore, the protocol in Fig. 5.7 can correct these propagated errors.

E

Look-up Tables for Decoding

Look-up tables for protocols discussed in the thesis are presented here. The mapping $1 \mapsto 0, -1 \mapsto 1$ from measured eigenvalues to bits is assumed.

The syndromes and flag measurement outcomes are formatted in a manner different from before, to be able to represent them in a consistent notation along with flagged measurement syndrome outcomes. For example, a syndrome 010 resulting from unflagged measurements is represented as the tuple $(0, 1, 0)$, enclosed in parentheses and with individual syndrome bits represented with commas. This will be used to represent the second subround syndromes.

Sequences of flagged measurement outcomes are represented as a tuple of tuples, where a constituent tuple has two elements, the syndrome measurement outcome, followed by the flag measurement outcome. For example, a syndrome value 1 and a flag outcome 0 from a single flagged measurement is represented as $(1, 0)$. A $(-, -)$ is used if no flagged measurement was made. Consider an example: for the $[\![5, 1, 3]\!]$ code, if the first flagged measurement gave trivial outcomes for the syndrome and the flag, the second flagged measurement gave a trivial syndrome but with the flag triggered, and the remaining two flagged measurements were not carried out (due to the rules of the protocols discussed in this thesis), this sequence of measurement outcomes is represented as $((0, 0), (0, 1), (-, -), (-, -))$. This format will be used to denote measurement outcomes from the first subround.

In this appendix, composite LUTs for flag protocols are presented. A composite LUT contains the first subround syndrome (i.e. the sequence of flagged measurement outcomes), followed by the second subround syndrome obtained from unflagged measurements, given that the first subround syndrome specified before has been observed.

Syndrome	Correction
$(0,0,0,0)$	None
$(0,0,0,1)$	$XIIII$
$(0,0,1,0)$	$IIZII$
$(0,0,1,1)$	$IIIIX$
$(0,1,0,0)$	$IIIIZ$
$(0,1,0,1)$	$IZIII$
$(0,1,1,0)$	$IIIXI$
$(0,1,1,1)$	$IIIIY$
$(1,0,0,0)$	$IXIII$
$(1,0,0,1)$	$IIIZI$
$(1,0,1,0)$	$ZIIII$
$(1,0,1,1)$	$YIIII$
$(1,1,0,0)$	$IIXII$
$(1,1,0,1)$	$IYIII$
$(1,1,1,0)$	$IIYII$
$(1,1,1,1)$	$IIIYI$

Table E.1: Look-up table for input errors with weight-1 corrections (i.e., the usual LUT) for the $[\![5,1,3]\!]$ code. This is the same as Table 3.1, and is reproduced here for convenience. The syndrome corresponds to measuring the generators $XZZXI$, $IXZZX$, $XIXZZ$, and $ZXIXZ$, in this order. Equivalent weight-1 corrections are omitted because they are the same as the correction in the second column.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged stabilizer measurement (second subround)	Correction	Equivalent minimum-weight correction
$((0,1),(-,-),(-,-),(-,-))$, or, $((1,1),(-,-),(-,-),(-,-))$	$(0,1,0,0)$	$IIZXI$	$IIZXI$
	$(1,1,0,0)$	$IXZXI$	$XYIII$
	$(1,0,0,1)$	$IYZXI$	$XXIII$
	$(0,0,0,1)$	$IZZXI$	$XIIII$
	$(0,1,1,0)$	$IIIXI$	$IIIXI$
	$(1,0,1,0)$	$IIXXI$	$IIXXI$
	$(1,0,0,0)$	$IIYXI$	$IIYXI$
$((1,0),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 corrections (Table E.1)		
$((0,0),(0,1),(-,-),(-,-))$, or, $((0,0),(1,1),(-,-),(-,-))$	$(1,0,1,0)$	$IIIZX$	$IIIZX$
	$(0,1,1,0)$	$IIXZX$	$XIIIY$
	$(0,1,0,0)$	$IIYZX$	$IXXII$
	$(1,0,0,0)$	$IIZZX$	$IXIII$
	$(0,0,1,1)$	$IIIIX$	$IIIIX$
	$(0,1,0,1)$	$IIIXX$	$IIIXX$
	$(1,1,0,0)$	$IIIYX$	$IIIYX$
$((0,0),(1,0),(-,-),(-,-))$	Usual LUT with weight-1 corrections (Table E.1)		
$((0,0),(0,0),(0,1),(-,-))$, or, $((0,0),(0,0),(1,1),(-,-))$	$(1,1,0,1)$	$IIIZZ$	$IIIZZ$
	$(0,0,0,1)$	$IIXZZ$	$XIIII$
	$(0,0,1,1)$	$IIYZZ$	$XIZII$
	$(1,1,1,1)$	$IIZZZ$	$IXIIY$
	$(0,1,0,0)$	$IIIIZ$	$IIIIZ$
	$(0,0,1,0)$	$IIIXZ$	$IIIXZ$
	$(1,0,1,1)$	$IIIYZ$	$IIIYZ$
$((0,0),(0,0),(1,0),(-,-))$	Usual LUT with weight-1 corrections (Table E.1)		
$((0,0),(0,0),(0,0),(0,1))$, or, $((0,0),(0,0),(0,0),(1,1))$	$(0,0,1,0)$	$IIIXZ$	$IIIXZ$
	$(1,0,1,0)$	$IXIXZ$	$ZIIII$
	$(1,1,1,1)$	$IYIXZ$	$ZZIII$
	$(0,1,1,1)$	$IZIXZ$	$ZYIII$
	$(0,1,0,0)$	$IIIIZ$	$IIIIZ$
	$(1,0,1,1)$	$IIIYZ$	$IIIYZ$
	$(1,1,0,1)$	$IIIZZ$	$IIIZZ$
$((0,0),(0,0),(0,0),(1,0))$	Usual LUT with weight-1 corrections (Table E.1)		

Table E.2: Composite LUT for the flag fault tolerance protocol for the $[\![5,1,3]\!]$ code (Fig. 4.4), for the flagged circuits shown in Fig. 4.2 and Fig. 4.5.

Syndrome	Correction
$(0,0,0)$	None
$(0,0,1)$	$ZIIIIII$
$(0,1,0)$	$IZIIIII$
$(0,1,1)$	$IIZIIII$
$(1,0,0)$	$IIIZIII$
$(1,0,1)$	$IIIIZII$
$(1,1,0)$	$IIIIIZI$
$(1,1,1)$	$IIIIIIZ$

Table E.3: Look-up table for input errors with weight-1 Z corrections (i.e., the usual LUT) for the Steane code. This can be derived analogously as Table 3.2. The syndrome corresponds to measuring the generators $IIIXXXX$, $IXXIIXX$ and $XIXIXIX$, in this order. Equivalent weight-1 corrections are omitted because they are the same as the correction in the second column.

Syndrome	Correction
$(0,0,0)$	None
$(0,0,1)$	$XIIIIII$
$(0,1,0)$	$IXIIIII$
$(0,1,1)$	$IIXIIII$
$(1,0,0)$	$IIIXIII$
$(1,0,1)$	$IIIIXII$
$(1,1,0)$	$IIIIIXI$
$(1,1,1)$	$IIIIIIX$

Table E.4: Look-up table for input errors with weight-1 X corrections (i.e., the usual LUT) for the Steane code. This is the same as Table 3.2, and is reproduced here for convenience. The syndrome corresponds to measuring the generators $IIIZZZZ$, $IZZIIZZ$ and $ZIZIZIZ$, in this order. Equivalent weight-1 corrections are omitted because they are the same as the correction in the second column.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged X-type stabilizer measurement (second subround)	Z Correction	Equivalent minimum-weight Z correction
$((0,1),(-,-),(-,-),(-,-),(-,-),(-,-))$, or, $((1,1),(-,-),(-,-),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((1,0),(-,-),(-,-),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,1),(-,-),(-,-),(-,-),(-,-))$, or, $((0,0),(1,1),(-,-),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(1,0),(-,-),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,0),(0,1),(-,-),(-,-),(-,-))$, or, $((0,0),(0,0),(1,1),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,0),(1,0),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,0),(0,0),(0,1),(-,-),(-,-))$, or, $((0,0),(0,0),(0,0),(1,1),(-,-),(-,-))$	$(0,0,1)$	$IIIIIZZ$	$IIIIIZZ$
	$(1,0,0)$	$IIIIZZZ$	$IIIZIII$
	$(1,1,1)$	$IIIIIIZ$	$IIIIIIZ$
$((0,0),(0,0),(0,0),(1,0),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,0),(0,0),(0,0),(0,1),(-,-))$, or, $((0,0),(0,0),(0,0),(0,0),(1,1),(-,-))$	$(0,0,1)$	$IIIIIZZ$	$IIIIIZZ$
	$(0,1,0)$	$IIZIIZZ$	$IZIIIII$
	$(1,1,1)$	$IIIIIIZ$	$IIIIIIZ$
$((0,0),(0,0),(0,0),(0,0),(1,0),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,0),(0,0),(0,0),(0,0),(0,1))$, or, $((0,0),(0,0),(0,0),(0,0),(0,0),(1,1))$	$(0,0,1)$	$IIZIZIZ$	$ZIIIIII$
	$(0,1,0)$	$IIIIZIZ$	$IIIIZIZ$
	$(1,1,1)$	$IIIIIIZ$	$IIIIIIZ$
$((0,0),(0,0),(0,0),(0,0),(0,0),(1,0))$	Usual LUT with weight-1 Z corrections (Table E.3)		

Table E.5: Composite LUT for Z corrections for the flag fault tolerance protocol for the Steane code (Fig. 4.6), for the flagged circuits shown in Fig. 4.7. This table needs to be used with Table E.6 to correct all weight-1 input errors or errors resulting from a single fault.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged Z-type stabilizer measurement (second subround)	X Correction	Equivalent minimum-weight X correction
$((0,1),(-,-),(-,-),(-,-),(-,-),(-,-))$, or, $((1,1),(-,-),(-,-),(-,-),(-,-),(-,-))$	$(0,0,1)$	$IIIIIXX$	$IIIIIXX$
	$(1,0,0)$	$IIIIXXX$	$IIIXIII$
	$(1,1,1)$	$IIIIIIX$	$IIIIIIX$
$((1,0),(-,-),(-,-),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,1),(-,-),(-,-),(-,-),(-,-))$, or, $((0,0),(1,1),(-,-),(-,-),(-,-),(-,-))$	$(0,0,1)$	$IIIIIXX$	$IIIIIXX$
	$(0,1,0)$	$IIXIIXX$	$IXIIIII$
	$(1,1,1)$	$IIIIIIX$	$IIIIIIX$
$((0,0),(1,0),(-,-),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,1),(-,-),(-,-),(-,-))$, or, $((0,0),(0,0),(1,1),(-,-),(-,-),(-,-))$	$(0,0,1)$	$IIXIXIX$	$XIIIIII$
	$(0,1,0)$	$IIIIXIX$	$IIIIXIX$
	$(1,1,1)$	$IIIIIIX$	$IIIIIIX$
$((0,0),(0,0),(1,0),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,0),(0,1),(-,-),(-,-))$, or, $((0,0),(0,0),(0,0),(1,1),(-,-),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,0),(1,0),(-,-),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,0),(0,0),(0,1),(-,-))$, or, $((0,0),(0,0),(0,0),(0,0),(1,1),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,0),(0,0),(1,0),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,0),(0,0),(0,0),(0,1))$, or, $((0,0),(0,0),(0,0),(0,0),(0,0),(1,1))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,0),(0,0),(0,0),(1,0))$	Usual LUT with weight-1 X corrections (Table E.4)		

Table E.6: Composite LUT for X corrections for the flag fault tolerance protocol for the Steane code (Fig. 4.6), for the flagged circuits shown in Fig. 4.7. This table needs to be used with Table E.5 to correct all weight-1 input errors or errors resulting from a single fault.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged stabilizer measurement (second subround)	Correction	Equivalent minimum-weight correction
$((0,1),(-,-),(-,-),(-,-))$, or, $((1,1),(-,-),(-,-),(-,-))$	$(0,0,1)$	$IIZXI$	$IIZXI$
	$(1,0,1)$	$IXZXI$	$XYIII$
	$(1,1,0)$	$IYZXI$	$XXIII$
	$(0,1,0)$	$IZZXI$	$XIIII$
	$(0,1,1)$	$IIIXI$	$IIIXI$
	$(1,1,1)$	$IIXXI$	$IIXXI$
	$(1,0,0)$	$IIYXI$	$IIYXI$
$((1,0),(-,-),(-,-),(-,-))$	Usual LUT with weight-1 corrections (Table E.1)		
$((0,0),(0,1),(-,-),(-,-))$, or, $((0,0),(1,1),(-,-),(-,-))$	$(0,0,1)$	$IIIZX$	$IIIZX$
	$(1,0,1)$	$IIXZX$	$IXYII$
	$(1,1,0)$	$IIYZX$	$IXXII$
	$(0,1,0)$	$IIZZX$	$IXIII$
	$(0,1,1)$	$IIIIX$	$IIIIX$
	$(1,1,1)$	$IIIXX$	$IIIXX$
	$(1,0,0)$	$IIIYX$	$IIIYX$
$((0,0),(1,0),(-,-),(-,-))$	Usual LUT with weight-1 corrections (Table E.1)		
$((0,0),(0,0),(0,1),(-,-))$, or, $((0,0),(0,0),(1,1),(-,-))$	$(0,0,1)$	$IIIZZ$	$IIIZZ$
	$(0,1,1)$	$IIXZZ$	$XIIII$
	$(1,0,1)$	$IIYZZ$	$XIZII$
	$(1,1,1)$	$IIZZZ$	$XIYII$
	$(0,1,0)$	$IIIIZ$	$IIIIZ$
	$(1,1,0)$	$IIIXZ$	$IIIXZ$
	$(1,0,0)$	$IIIYZ$	$IIIYZ$
$((0,0),(0,0),(1,0),(-,-))$	Usual LUT with weight-1 corrections (Table E.1)		
$((0,0),(0,0),(0,0),(0,1))$, or, $((0,0),(0,0),(0,0),(1,1))$	$(0,0,1)$	$IIIXZ$	$IIIXZ$
	$(0,1,1)$	$IXIXZ$	$ZIIII$
	$(1,1,1)$	$IYIXZ$	$ZZIII$
	$(1,0,1)$	$IZIXZ$	$ZYIII$
	$(0,1,0)$	$IIIIZ$	$IIIIZ$
	$(1,0,0)$	$IIIYZ$	$IIIYZ$
	$(1,1,0)$	$IIIZZ$	$IIIZZ$
$((0,0),(0,0),(0,0),(1,0))$	Usual LUT with weight-1 corrections (Table E.1)		

Table E.7: Composite LUT for the split-and-diagnose protocol for the $[\![5,1,3]\!]$ code (reduced unflagged measurements for $f=1$ branches, Fig. 5.5), for the flagged circuits shown in Fig. 4.2 and Fig. 4.5.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged stabilizer measurement (second subround)	Correction	Equivalent minimum-weight correction
$((0,1),(-,-),(-,-),(-,-),(-,-),(-,-))$, or, $((1,1),(-,-),(-,-),(-,-),(-,-),(-,-))$	$(0,0,1)$	$IIIIIXX$	$IIIIIXX$
	$(0,1,0)$	$IIIIXXX$	$IIIXIII$
	$(1,1,1)$	$IIIIYXX$	$IIIXZII$
	$(1,0,1)$	$IIIIZXX$	$IIIXYII$
	$(0,1,1)$	$IIIIIIX$	$IIIIIIX$
	$(1,0,0)$	$IIIIIYX$	$IIIIIYX$
	$(1,1,0)$	$IIIIIZX$	$IIIIIZX$
$((1,0),(-,-),(-,-),(-,-),(-,-),(-,-))$	$(1,0,0,0)$	$IIIZIII$	$IIIZIII$
	$(1,0,1,0)$	$IIIIZII$	$IIIIZII$
	$(1,1,0,0)$	$IIIIIZI$	$IIIIIZI$
	$(1,1,1,0)$	$IIIIIIZ$	$IIIIIIZ$
	$(1,0,0,1)$	$IIIYIII$	$IIIYIII$
	$(1,0,1,1)$	$IIIIYII$	$IIIIYII$
	$(1,1,0,1)$	$IIIIIYI$	$IIIIIYI$
	$(1,1,1,1)$	$IIIIIIY$	$IIIIIIY$
$((0,0),(0,1),(-,-),(-,-),(-,-),(-,-))$, or, $((0,0),(1,1),(-,-),(-,-),(-,-),(-,-))$	$(0,0,1)$	$IIIIIXX$	$IIIIIXX$
	$(0,1,0)$	$IIXIIXX$	$IXIIIII$
	$(1,1,1)$	$IIYIIXX$	$IXZIIII$
	$(1,0,1)$	$IIZIIXX$	$IXYIIII$
	$(0,1,1)$	$IIIIIIX$	$IIIIIIX$
	$(1,0,0)$	$IIIIIYX$	$IIIIIYX$
	$(1,1,0)$	$IIIIIZX$	$IIIIIZX$
$((0,0),(1,0),(-,-),(-,-),(-,-),(-,-))$	$(0,1,0,0)$	$IZIIIII$	$IZIIIII$
	$(0,1,1,0)$	$IIZIIII$	$IIZIIII$
	$(1,1,0,0)$	$IIIIIZI$	$IIIIIZI$
	$(1,1,1,0)$	$IIIIIIZ$	$IIIIIIZ$
	$(0,1,0,1)$	$IYIIIII$	$IYIIIII$
	$(0,1,1,1)$	$IIYIIII$	$IIYIIII$
	$(1,1,0,1)$	$IIIIIYI$	$IIIIIYI$
	$(1,1,1,1)$	$IIIIIIY$	$IIIIIIY$
$((0,0),(0,0),(0,1),(-,-),(-,-),(-,-))$, or, $((0,0),(0,0),(1,1),(-,-),(-,-),(-,-))$	$(0,0,1)$	$IIIIXIX$	$IIIIXIX$
	$(0,1,0)$	$IIXIXIX$	$XIIIIII$
	$(1,1,1)$	$IIYIXIX$	$XIZIIII$
	$(1,0,1)$	$IIZIXIX$	$XIYIIII$
	$(0,1,1)$	$IIIIIIX$	$IIIIIIX$
	$(1,0,0)$	$IIIIYIX$	$IIIIYIX$
	$(1,1,0)$	$IIIIZIX$	$IIIIZIX$
$((0,0),(0,0),(1,0),(-,-),(-,-),(-,-))$	$(0,0,1,0)$	$ZIIIIII$	$ZIIIIII$
	$(0,1,1,0)$	$IIZIIII$	$IIZIIII$
	$(1,0,1,0)$	$IIIIZII$	$IIIIZII$
	$(1,1,1,0)$	$IIIIIIZ$	$IIIIIIZ$
	$(0,0,1,1)$	$YIIIIII$	$YIIIIII$
	$(0,1,1,1)$	$IIYIIII$	$IIYIIII$
	$(1,0,1,1)$	$IIIIYII$	$IIIIYII$
	$(1,1,1,1)$	$IIIIIIY$	$IIIIIIY$

Table E.8: Composite LUT for the split-and-diagnose protocol for the Steane code (reduced unflagged measurements for $f = 1$ as well as $(s, f) = (1, 0)$ branches, Fig. 5.6), for the flagged circuits shown in Fig. 4.7. This is the first half of the table, with the remainder being in Table E.9.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged stabilizer measurement (second subround)	Correction	Equivalent minimum-weight correction
$((0,0),(0,0),(0,0),(0,1),(-,-),(-,-))$, or, $((0,0),(0,0),(0,0),(1,1),(-,-),(-,-))$	$(0,0,1)$	$IIIIIZZ$	$IIIIIZZ$
	$(1,0,1)$	$IIIIXZZ$	$IIIZYII$
	$(1,1,1)$	$IIIIYZZ$	$IIIZXII$
	$(0,1,0)$	$IIIIZZZ$	$IIIZIII$
	$(0,1,1)$	$IIIIIIZ$	$IIIIIIZ$
	$(1,1,0)$	$IIIIIXZ$	$IIIIIXZ$
	$(1,0,0)$	$IIIIIYZ$	$IIIIIYZ$
$((0,0),(0,0),(0,0),(1,0),(-,-),(-,-))$	$(1,0,0,0)$	$IIIXIII$	$IIIXIII$
	$(1,0,1,0)$	$IIIIXII$	$IIIIXII$
	$(1,1,0,0)$	$IIIIIXI$	$IIIIIXI$
	$(1,1,1,0)$	$IIIIIIX$	$IIIIIIX$
	$(1,0,0,1)$	$IIIYIII$	$IIIYIII$
	$(1,0,1,1)$	$IIIIYII$	$IIIIYII$
	$(1,1,0,1)$	$IIIIIYI$	$IIIIIYI$
	$(1,1,1,1)$	$IIIIIIY$	$IIIIIIY$
$((0,0),(0,0),(0,0),(0,0),(0,1),(-,-))$, or, $((0,0),(0,0),(0,0),(0,0),(1,1),(-,-))$	$(0,0,1)$	$IIIIIZZ$	$IIIIIZZ$
	$(1,0,1)$	$IIXIIZZ$	$IZYIIII$
	$(1,1,1)$	$IIYIIZZ$	$IZXIIII$
	$(0,1,0)$	$IIZIIZZ$	$IZIIIII$
	$(0,1,1)$	$IIIIIIZ$	$IIIIIIZ$
	$(1,1,0)$	$IIIIIXZ$	$IIIIIXZ$
	$(1,0,0)$	$IIIIIYZ$	$IIIIIYZ$
$((0,0),(0,0),(0,0),(0,0),(1,0),(-,-))$	$(0,1,0,0)$	$IXIIIII$	$IXIIIII$
	$(0,1,1,0)$	$IIXIIII$	$IIXIIII$
	$(1,1,0,0)$	$IIIIIXI$	$IIIIIXI$
	$(1,1,1,0)$	$IIIIIIX$	$IIIIIIX$
	$(0,1,0,1)$	$IYIIIII$	$IYIIIII$
	$(0,1,1,1)$	$IIYIIII$	$IIYIIII$
	$(1,1,0,1)$	$IIIIIYI$	$IIIIIYI$
	$(1,1,1,1)$	$IIIIIIY$	$IIIIIIY$
$((0,0),(0,0),(0,0),(0,0),(0,0),(0,1))$, or, $((0,0),(0,0),(0,0),(0,0),(0,0),(1,1))$	$(0,0,1)$	$IIIIZIZ$	$IIIIZIZ$
	$(1,0,1)$	$IIXIZIZ$	$ZIYIIII$
	$(1,1,1)$	$IIYIZIZ$	$ZIXIIII$
	$(0,1,0)$	$IIZIZIZ$	$ZIIIIII$
	$(0,1,1)$	$IIIIIIZ$	$IIIIIIZ$
	$(1,1,0)$	$IIIIXIZ$	$IIIIXIZ$
	$(1,0,0)$	$IIIIYIZ$	$IIIIYIZ$
$((0,0),(0,0),(0,0),(0,0),(0,0),(1,0))$	$(0,0,1,0)$	$XIIIIII$	$XIIIIII$
	$(0,1,1,0)$	$IIXIIII$	$IIXIIII$
	$(1,0,1,0)$	$IIIIXII$	$IIIIXII$
	$(1,1,1,0)$	$IIIIIIX$	$IIIIIIX$
	$(0,0,1,1)$	$YIIIIII$	$YIIIIII$
	$(0,1,1,1)$	$IIYIIII$	$IIYIIII$
	$(1,0,1,1)$	$IIIIYII$	$IIIIYII$
	$(1,1,1,1)$	$IIIIIIY$	$IIIIIIY$

Table E.9: (Continued) Composite LUT for the split-and-diagnose protocol for the Steane code (reduced unflagged measurements for $f = 1$ as well as $(s, f) = (1, 0)$ branches, Fig. 5.6), for the flagged circuits shown in Fig. 4.7. This is the second half of the table, with the remainder being in Table E.8.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged X-type stabilizer measurement (second subround)	Z Correction	Equivalent minimum-weight Z correction
$((0,1),(-,-),(-,-))$, or, $((1,1),(-,-),(-,-))$	$(0,0,1)$	$IIIIIZZ$	$IIIIIZZ$
	$(0,1,0)$	$IIZIIZZ$	$IZIIIII$
	$(1,0,1)$	$IIIZIZZ$	$IIIIZII$
	$(1,0,0)$	$IIIIZZZ$	$IIIZIII$
	$(1,1,1)$	$IIIIIIZ$	$IIIIIIZ$
$((1,0),(-,-),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,1),(-,-))$, or, $((0,0),(1,1),(-,-))$	$(0,1,1)$	$IIZIIII$	$IIZIIII$
	$(1,0,0)$	$IIIZIII$	$IIIZIII$
	$(0,0,1)$	$IIIIIZZ$	$IIIIIZZ$
	$(1,1,1)$	$IIIIIIZ$	$IIIIIIZ$
$((0,0),(1,0),(-,-))$	Usual LUT with weight-1 Z corrections (Table E.3)		
$((0,0),(0,0),(0,1))$, or, $((0,0),(0,0),(1,1))$	$(0,0,1)$	$IIZZIZI$	$ZIIIIII$
	$(0,1,1)$	$IZZZIZI$	$ZZIIIII$
	$(0,1,0)$	$IIIZIZI$	$IIIZIZI$
	$(1,1,0)$	$IIIIIZI$	$IIIIIZI$
$((0,0),(0,0),(1,0))$	Usual LUT with weight-1 Z corrections (Table E.3)		

Table E.10: Composite LUT for Z corrections for the detect-and-diagnose protocol for the Steane code (reduced flagged measurements, Fig. 5.7), for the flagged circuits shown in Fig. 5.8. This table needs to be used with Table E.11 to correct all weight-1 input errors or errors resulting from a single fault.

Flagged stabilizer measurement outcome (first subround)	Syndrome from unflagged Z-type stabilizer measurement (second subround)	X Correction	Equivalent minimum-weight X correction
$((0,1),(-,-),(-,-))$, or, $((1,1),(-,-),(-,-))$	$(0,1,1)$	$IIXXXXX$	$IIXIIII$
	$(1,0,0)$	$IIIIXXX$	$IIIXIII$
	$(0,0,1)$	$IIIIIXX$	$IIIIIXX$
	$(1,1,1)$	$IIIIIIX$	$IIIIIIX$
$((1,0),(-,-),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,1),(-,-))$, or, $((0,0),(1,1),(-,-))$	$(0,1,0)$	$IIIXIXI$	$IIIXIXI$
	$(0,0,1)$	$IIXXIXI$	$XIIIIII$
	$(1,1,0)$	$IIIIIXI$	$IIIIIXI$
	$(0,1,1)$	$IIIIXXI$	
$((0,0),(1,0),(-,-))$	Usual LUT with weight-1 X corrections (Table E.4)		
$((0,0),(0,0),(0,1))$, or, $((0,0),(0,0),(1,1))$	$(0,1,0)$	$IIXXXII$	$IXIIIII$
	$(0,0,1)$	$IIIXXII$	$IIIXXII$
	$(1,0,1)$	$IIIIXII$	$IIIIXII$
$((0,0),(0,0),(1,0))$	Usual LUT with weight-1 X corrections (Table E.4)		

Table E.11: Composite LUT for X corrections for the detect-and-diagnose protocol for the Steane code (reduced flagged measurements, Fig. 5.7), for the flagged circuits shown in Fig. 5.8. This table needs to be used with Table E.10 to correct all weight-1 input errors or errors resulting from a single fault.

F

Split-and-Diagnose-Style Stabilizer Sequences

This section presents the reduced stabilizer sequences for $f = 1$ branches, which yield unique and nontrivial syndromes for errors which trigger the flag, obtained by computer search for the split-and-diagnose-style flag protocol for the $[\![5, 1, 3]\!]$ code and the Steane code.

$[\![5, 1, 3]\!]$ code

The stabilizer sequences are tabulated separately for the $f = 1$ branches resulting from the 4 flagged measurements, in Tables F.1-F.8. In the $f = 1$ branch following the i^{th} flagged measurement, $i \in \{1, 2, 3, 4\}$, the stabilizers $S_{i,1}$ and $S_{i,2}$ are measured, followed by $S_{i,2,1}$ if $S_{i,2}$ gives the outcome 0, or $S_{i,2,2}$ if $S_{i,2}$ gives the outcome 1.

$S_{1,1}$	$S_{1,2}$	$S_{1,2,1}$	Possible choices for $S_{1,2,2}$
$XZZXI$	$YXXYI$	$IXZZX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$YXXYI$	$XYIYX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$YXXYI$	$XXYIY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$YXXYI$	$ZIZYY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$YXXYI$	$IYXXY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$YXXYI$	$YZIZY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$YXXYI$	$YIYXX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$YXXYI$	$ZZXIX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$XZZXI$	$ZYYZI$	$XXYIY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$XZZXI$	$ZYYZI$	$ZIZYY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$XZZXI$	$ZYYZI$	$IYXXY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$XZZXI$	$ZYYZI$	$YZIZY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$IZYYZ$	$YYZIZ$	$IXZZX$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$IZYYZ$	$YYZIZ$	$IYXXY$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$IZYYZ$	$YYZIZ$	$YZIZY$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$IZYYZ$	$YYZIZ$	$YIYXX$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$IZYYZ$	$YXXYI$	$IXZZX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$IZYYZ$	$YXXYI$	$XYIYX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$IZYYZ$	$YXXYI$	$XXYIY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$IZYYZ$	$YXXYI$	$ZIZYY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$

Table F.1: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the first flagged stabilizer measurement ($XZZXI$) for the $[\![5, 1, 3]\!]$ code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.5 are coloured in red. Continued in Table F.2.

$S_{1,1}$	$S_{1,2}$	$S_{1,2,1}$	Possible choices for $S_{1,2,2}$
$IZYYZ$	$YXXYI$	$IYXXY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$IZYYZ$	$YXXYI$	$YZIZY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$IZYYZ$	$YXXYI$	$YIYXX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$IZYYZ$	$YXXYI$	$ZZXIX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$IZYYZ$	$XYIYX$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YYZIZ$	$IZYYZ$	$XXYIY$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YYZIZ$	$IZYYZ$	$ZIZYY$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YYZIZ$	$IZYYZ$	$ZZXIX$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YYZIZ$	$YXXYI$	$IXZZX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$YXXYI$	$XYIYX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$YXXYI$	$XXYIY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$YXXYI$	$ZIZYY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$YXXYI$	$IYXXY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$YXXYI$	$YZIZY$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$YXXYI$	$YIYXX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YYZIZ$	$YXXYI$	$ZZXIX$	$XZZXI, XIXZZ, ZXIXZ, ZYYZI$
$YXXYI$	$XZZXI$	$IXZZX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$YXXYI$	$XZZXI$	$XYIYX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$YXXYI$	$XZZXI$	$YIYXX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$YXXYI$	$XZZXI$	$ZZXIX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$YXXYI$	$IZYYZ$	$XYIYX$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YXXYI$	$IZYYZ$	$XXYIY$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YXXYI$	$IZYYZ$	$ZIZYY$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YXXYI$	$IZYYZ$	$ZZXIX$	$IXZZX, IYXXY, YZIZY, YIYXX$
$YXXYI$	$YYZIZ$	$IXZZX$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$YXXYI$	$YYZIZ$	$IYXXY$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$YXXYI$	$YYZIZ$	$YZIZY$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$YXXYI$	$YYZIZ$	$YIYXX$	$XYIYX, XXYIY, ZIZYY, ZZXIX$
$YXXYI$	$ZYYZI$	$XXYIY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$YXXYI$	$ZYYZI$	$ZIZYY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$YXXYI$	$ZYYZI$	$IYXXY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$YXXYI$	$ZYYZI$	$YZIZY$	$IXZZX, XYIYX, YIYXX, ZZXIX$
$ZYYZI$	$XZZXI$	$IXZZX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$ZYYZI$	$XZZXI$	$XYIYX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$ZYYZI$	$XZZXI$	$YIYXX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$ZYYZI$	$XZZXI$	$ZZXIX$	$XXYIY, ZIZYY, IYXXY, YZIZY$
$ZYYZI$	$YXXYI$	$IXZZX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$ZYYZI$	$YXXYI$	$XYIYX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$ZYYZI$	$YXXYI$	$XXYIY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$ZYYZI$	$YXXYI$	$ZIZYY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$ZYYZI$	$YXXYI$	$IYXXY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$ZYYZI$	$YXXYI$	$YZIZY$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$ZYYZI$	$YXXYI$	$YIYXX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$
$ZYYZI$	$YXXYI$	$ZZXIX$	$XIXZZ, ZXIXZ, IZYYZ, YYZIZ$

Table F.2: Continued from Table F.1.

$S_{2,1}$	$S_{2,2}$	$S_{2,2,1}$	Possible choices for $S_{2,2,2}$
$IXZZX$	$IZYYZ$	$YYZIZ$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IXZZX$	$IZYYZ$	$YXXYI$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IXZZX$	$IZYYZ$	$YZIZY$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IXZZX$	$IZYYZ$	$YIYXX$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IXZZX$	$IYXXY$	$XZZXI$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IXZZX$	$IYXXY$	$XIXZZ$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IXZZX$	$IYXXY$	$XYIYX$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IXZZX$	$IYXXY$	$YYZIZ$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IXZZX$	$IYXXY$	$XXYIY$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IXZZX$	$IYXXY$	$YXXYI$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IXZZX$	$IYXXY$	$YZIZY$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IXZZX$	$IYXXY$	$YIYXX$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IXZZX$	$XZZXI$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IZYYZ$	$IXZZX$	$XIXZZ$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IZYYZ$	$IXZZX$	$XYIYX$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IZYYZ$	$IXZZX$	$XXYIY$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IZYYZ$	$IYXXY$	$XZZXI$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IYXXY$	$XIXZZ$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IYXXY$	$XYIYX$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IYXXY$	$YYZIZ$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IYXXY$	$XXYIY$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IYXXY$	$YXXYI$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IYXXY$	$YZIZY$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$IZYYZ$	$IYXXY$	$YIYXX$	$ZXIXZ, ZIZYY, ZYYZI, ZZXIX$
$ZIZYY$	$IYXXY$	$XZZXI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$XIXZZ$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$XYIYX$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$YYZIZ$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$XXYIY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$YXXYI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$YZIZY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$YIYXX$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$

Table F.3: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the second flagged stabilizer measurement ($IXZZX$) for the $[\![5, 1, 3]\!]$ code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.5 are coloured in red. Continued in Table F.4.

$S_{2,1}$	$S_{2,2}$	$S_{2,2,1}$	Possible choices for $S_{2,2,2}$
$ZIZYY$	$ZYYZI$	$XIXZZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZIZYY$	$ZYYZI$	$XYIYX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZIZYY$	$ZYYZI$	$YYZIZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZIZYY$	$ZYYZI$	$YIYXX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$IYXXY$	$IXZZX$	$XZZXI$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IYXXY$	$IXZZX$	$XIXZZ$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IYXXY$	$IXZZX$	$XYIYX$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IYXXY$	$IXZZX$	$XXYIY$	$YYZIZ, YXXYI, YZIZY, YIYXX$
$IYXXY$	$IZYYZ$	$YYZIZ$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IYXXY$	$IZYYZ$	$YXXYI$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IYXXY$	$IZYYZ$	$YZIZY$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IYXXY$	$IZYYZ$	$YIYXX$	$XZZXI, XIXZZ, XYIYX, XXYIY$
$IYXXY$	$ZIZYY$	$XZZXI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$XXYIY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$YXXYI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$YZIZY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZYYZI$	$XIXZZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$IYXXY$	$ZYYZI$	$XYIYX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$IYXXY$	$ZYYZI$	$YYZIZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$IYXXY$	$ZYYZI$	$YIYXX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZYYZI$	$ZIZYY$	$XZZXI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$XXYIY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$YXXYI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$YZIZY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$IYXXY$	$XZZXI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$XIXZZ$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$XYIYX$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$YYZIZ$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$XXYIY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$YXXYI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$YZIZY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$YIYXX$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$

Table F.4: Continued from Table F.3.

$S_{3,1}$	$S_{3,2}$	$S_{3,2,1}$	Possible choices for $S_{3,2,2}$
$XIXZZ$	$ZIZYY$	$XZZXI$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$ZIZYY$	$IXZZX$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$ZIZYY$	$ZXIXZ$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$ZIZYY$	$IZYYZ$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$ZIZYY$	$XXYIY$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$ZIZYY$	$YXXYI$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$ZIZYY$	$YZIZY$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$ZIZYY$	$ZZXIX$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$XIXZZ$	$YIYXX$	$XZZXI$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$XIXZZ$	$YIYXX$	$IZYYZ$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$XIXZZ$	$YIYXX$	$YZIZY$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$XIXZZ$	$YIYXX$	$ZZXIX$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$ZIZYY$	$XIXZZ$	$IXZZX$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$ZIZYY$	$XIXZZ$	$ZXIXZ$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$ZIZYY$	$XIXZZ$	$XXYIY$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$ZIZYY$	$XIXZZ$	$YXXYI$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$ZIZYY$	$IYXXY$	$XZZXI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$XXYIY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$YXXYI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$IYXXY$	$YZIZY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZIZYY$	$ZYYZI$	$IXZZX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZIZYY$	$ZYYZI$	$ZXIXZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZIZYY$	$ZYYZI$	$IZYYZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZIZYY$	$ZYYZI$	$ZZXIX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZIZYY$	$YIYXX$	$XZZXI$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$ZIZYY$	$YIYXX$	$IZYYZ$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$ZIZYY$	$YIYXX$	$YZIZY$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$ZIZYY$	$YIYXX$	$ZZXIX$	$IXZZX, ZXIXZ, XXYIY, YXXYI$
$IYXXY$	$ZIZYY$	$XZZXI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$IXZZX$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$ZXIXZ$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$IZYYZ$	$XIXZZ, XYIYX, YYZIZ, YIYXX$

Table F.5: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the third flagged stabilizer measurement ($XIXZZ$) for the $[\![5, 1, 3]\!]$ code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.5 are coloured in red. Continued in Table F.6.

$S_{3,1}$	$S_{3,2}$	$S_{3,2,1}$	Possible choices for $S_{3,2,2}$
$IYXXY$	$ZIZYY$	$XXYIY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$YXXYI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$YZIZY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZIZYY$	$ZZXIX$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$IYXXY$	$ZYYZI$	$IXZZX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$IYXXY$	$ZYYZI$	$ZXIXZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$IYXXY$	$ZYYZI$	$IZYYZ$	$XZZXI, XXYIY, YXXYI, YZIZY$
$IYXXY$	$ZYYZI$	$ZZXIX$	$XZZXI, XXYIY, YXXYI, YZIZY$
$ZYYZI$	$ZIZYY$	$XZZXI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$IXZZX$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$ZXIXZ$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$IZYYZ$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$XXYIY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$YXXYI$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$YZIZY$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$ZIZYY$	$ZZXIX$	$XIXZZ, XYIYX, YYZIZ, YIYXX$
$ZYYZI$	$IYXXY$	$XZZXI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$XXYIY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$YXXYI$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$ZYYZI$	$IYXXY$	$YZIZY$	$IXZZX, ZXIXZ, IZYYZ, ZZXIX$
$YIYXX$	$XIXZZ$	$IXZZX$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$YIYXX$	$XIXZZ$	$ZXIXZ$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$YIYXX$	$XIXZZ$	$XXYIY$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$YIYXX$	$XIXZZ$	$YXXYI$	$XZZXI, IZYYZ, YZIZY, ZZXIX$
$YIYXX$	$ZIZYY$	$XZZXI$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$YIYXX$	$ZIZYY$	$IXZZX$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$YIYXX$	$ZIZYY$	$ZXIXZ$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$YIYXX$	$ZIZYY$	$IZYYZ$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$YIYXX$	$ZIZYY$	$XXYIY$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$YIYXX$	$ZIZYY$	$YXXYI$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$YIYXX$	$ZIZYY$	$YZIZY$	$XYIYX, YYZIZ, IYXXY, ZYYZI$
$YIYXX$	$ZIZYY$	$ZZXIX$	$XYIYX, YYZIZ, IYXXY, ZYYZI$

Table F.6: Continued from Table F.5.

$S_{4,1}$	$S_{4,2}$	$S_{4,2,1}$	Possible choices for $S_{4,2,2}$
$XZZXI$	$IXZZX$	$YXXYI$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XZZXI$	$IXZZX$	$ZYYZI$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XZZXI$	$IXZZX$	$YIYXX$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XZZXI$	$IXZZX$	$ZZXIX$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XZZXI$	$XYIYX$	$XIXZZ$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$XZZXI$	$XYIYX$	$IZYYZ$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$XZZXI$	$XYIYX$	$XXYIY$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$XZZXI$	$XYIYX$	$YXXYI$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$XZZXI$	$XYIYX$	$IYXXY$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$XZZXI$	$XYIYX$	$ZYYZI$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$XZZXI$	$XYIYX$	$YIYXX$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$XZZXI$	$XYIYX$	$ZZXIX$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XZZXI$	$XIXZZ$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$IXZZX$	$XZZXI$	$IZYYZ$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$IXZZX$	$XZZXI$	$XXYIY$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$IXZZX$	$XZZXI$	$IYXXY$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$IXZZX$	$XYIYX$	$XIXZZ$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XYIYX$	$IZYYZ$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XYIYX$	$XXYIY$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XYIYX$	$YXXYI$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XYIYX$	$IYXXY$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XYIYX$	$ZYYZI$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XYIYX$	$YIYXX$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$IXZZX$	$XYIYX$	$ZZXIX$	$ZXIXZ, YYZIZ, ZIZYY, YZIZY$
$ZXIXZ$	$XYIYX$	$XIXZZ$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$ZXIXZ$	$XYIYX$	$IZYYZ$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$ZXIXZ$	$XYIYX$	$XXYIY$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$ZXIXZ$	$XYIYX$	$YXXYI$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$ZXIXZ$	$XYIYX$	$IYXXY$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$ZXIXZ$	$XYIYX$	$ZYYZI$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$ZXIXZ$	$XYIYX$	$YIYXX$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$ZXIXZ$	$XYIYX$	$ZZXIX$	$XZZXI, IXZZX, YYZIZ, ZIZYY$

Table F.7: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the fourth flagged stabilizer measurement ($ZXIXZ$) for the $[\![5, 1, 3]\!]$ code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.5 are coloured in red. Continued in Table F.8.

$S_{4,1}$	$S_{4,2}$	$S_{4,2,1}$	Possible choices for $S_{4,2,2}$
$ZXIXZ$	$YZIZY$	$IZYYZ$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$ZXIXZ$	$YZIZY$	$XXYIY$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$ZXIXZ$	$YZIZY$	$ZYYZI$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$ZXIXZ$	$YZIZY$	$YIYXX$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$XYIYX$	$XZZXI$	$XIXZZ$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$XYIYX$	$XZZXI$	$IZYYZ$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$XYIYX$	$XZZXI$	$XXYIY$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$XYIYX$	$XZZXI$	$IYXXY$	$YXXYI, ZYYZI, YIYXX, ZZXIX$
$XYIYX$	$IXZZX$	$YXXYI$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XYIYX$	$IXZZX$	$ZYYZI$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XYIYX$	$IXZZX$	$YIYXX$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XYIYX$	$IXZZX$	$ZZXIX$	$XIXZZ, IZYYZ, XXYIY, IYXXY$
$XYIYX$	$ZXIXZ$	$XIXZZ$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$XYIYX$	$ZXIXZ$	$YXXYI$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$XYIYX$	$ZXIXZ$	$IYXXY$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$XYIYX$	$ZXIXZ$	$ZZXIX$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$XYIYX$	$YZIZY$	$IZYYZ$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$XYIYX$	$YZIZY$	$XXYIY$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$XYIYX$	$YZIZY$	$ZYYZI$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$XYIYX$	$YZIZY$	$YIYXX$	$XIXZZ, YXXYI, IYXXY, ZZXIX$
$YZIZY$	$ZXIXZ$	$XIXZZ$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$YZIZY$	$ZXIXZ$	$YXXYI$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$YZIZY$	$ZXIXZ$	$IYXXY$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$YZIZY$	$ZXIXZ$	$ZZXIX$	$IZYYZ, XXYIY, ZYYZI, YIYXX$
$YZIZY$	$XYIYX$	$XIXZZ$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$YZIZY$	$XYIYX$	$IZYYZ$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$YZIZY$	$XYIYX$	$XXYIY$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$YZIZY$	$XYIYX$	$YXXYI$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$YZIZY$	$XYIYX$	$IYXXY$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$YZIZY$	$XYIYX$	$ZYYZI$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$YZIZY$	$XYIYX$	$YIYXX$	$XZZXI, IXZZX, YYZIZ, ZIZYY$
$YZIZY$	$XYIYX$	$ZZXIX$	$XZZXI, IXZZX, YYZIZ, ZIZYY$

Table F.8: Continued from Table F.7.

Steane code

The reduced, adaptive stabilizer sequences are tabulated separately for the $f = 1$ branches resulting from the 6 flagged measurements, in Tables F.9-F.14. The search is restricted to weight-4 stabilizers, with the notation presented in in Sec. 7.

$S_{1,1}$	$S_{1,2}$	$S_{1,2,1}$	Possible choices for $S_{1,2,2}$
$IIIXXXX$	$IIIZZZZ$	$YIYIYIY$	$IZZIIZZ, ZIZIZIZ, IZZZZII, ZIZZIZI$
$IIIXXXX$	$IIIZZZZ$	$YIYYIYI$	$IZZIIZZ, ZIZIZIZ, IZZZZII, ZIZZIZI$
$IIIXXXX$	$IIIYYYY$	$ZIZIZIZ$	$IYYIIYY, YIYIYIY, IYYYYII, YIYYIYI$
$IIIXXXX$	$IIIYYYY$	$ZIZZIZI$	$IYYIIYY, YIYIYIY, IYYYYII, YIYYIYI$
$XXIIXXI$	$IIIZZZZ$	$YIYIYIY$	$IZZIIZZ, ZIZIZIZ, IZZZZII, ZIZZIZI$
$XXIIXXI$	$IIIZZZZ$	$YIYYIYI$	$IZZIIZZ, ZIZIZIZ, IZZZZII, ZIZZIZI$
$XXIIXXI$	$IIIYYYY$	$ZIZIZIZ$	$IYYIIYY, YIYIYIY, IYYYYII, YIYYIYI$
$XXIIXXI$	$IIIYYYY$	$ZIZZIZI$	$IYYIIYY, YIYIYIY, IYYYYII, YIYYIYI$
$IIIZZZZ$	$IIIXXXX$	$ZIZIZIZ$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$IIIXXXX$	$ZIZZIZI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$IIIXXXX$	$YIYIYIY$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$IIIXXXX$	$YIYYIYI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$XXIIXXI$	$ZIZIZIZ$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$XXIIXXI$	$ZIZZIZI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$XXIIXXI$	$YIYIYIY$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$XXIIXXI$	$YIYYIYI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIZZZZ$	$IIIYYYY$	$ZIZIZIZ$	$IYYIIYY, YIYIYIY, IYYYYII, YIYYIYI$
$IIIZZZZ$	$IIIYYYY$	$ZIZZIZI$	$IYYIIYY, YIYIYIY, IYYYYII, YIYYIYI$
$IIIYYYY$	$IIIXXXX$	$ZIZIZIZ$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$IIIXXXX$	$ZIZZIZI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$IIIXXXX$	$YIYIYIY$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$IIIXXXX$	$YIYYIYI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$XXIIXXI$	$ZIZIZIZ$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$XXIIXXI$	$ZIZZIZI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$XXIIXXI$	$YIYIYIY$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$XXIIXXI$	$YIYYIYI$	$IXXIIXX, XIXIXIX, IXXXXII, XIXXIXI$
$IIIYYYY$	$IIIZZZZ$	$YIYIYIY$	$IZZIIZZ, ZIZIZIZ, IZZZZII, ZIZZIZI$
$IIIYYYY$	$IIIZZZZ$	$YIYYIYI$	$IZZIIZZ, ZIZIZIZ, IZZZZII, ZIZZIZI$

Table F.9: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the first flagged stabilizer measurement ($IIIXXXX$) for the Steane code. Refer to Sec. 7 The particular sequences derived analytically in the protocol in Fig. 5.6 are coloured in red.

$S_{2,1}$	$S_{2,2}$	$S_{2,2,1}$	Possible choices for $S_{2,2,2}$
$IXXIIXX$	$IZZIIZZ$	$YIYIYIY$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$IZZIIZZ$	$YYIIYYI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$IYYIIYY$	$ZIZIZIZ$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IXXIIXX$	$IYYIIYY$	$ZZIIZZI$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$XIXXIXI$	$IZZIIZZ$	$YIYIYIY$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$XIXXIXI$	$IZZIIZZ$	$YYIIYYI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$XIXXIXI$	$IYYIIYY$	$ZIZIZIZ$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$XIXXIXI$	$IYYIIYY$	$ZZIIZZI$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IZZIIZZ$	$IXXIIXX$	$ZIZIZIZ$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$IXXIIXX$	$ZZIIZZI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$IXXIIXX$	$YIYIYIY$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$IXXIIXX$	$YYIIYYI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$XIXXIXI$	$ZIZIZIZ$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$XIXXIXI$	$ZZIIZZI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$XIXXIXI$	$YIYIYIY$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$XIXXIXI$	$YYIIYYI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$IYYIIYY$	$ZIZIZIZ$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IZZIIZZ$	$IYYIIYY$	$ZZIIZZI$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IYYIIYY$	$IXXIIXX$	$ZIZIZIZ$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$IXXIIXX$	$ZZIIZZI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$IXXIIXX$	$YIYIYIY$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$IXXIIXX$	$YYIIYYI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$XIXXIXI$	$ZIZIZIZ$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$XIXXIXI$	$ZZIIZZI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$XIXXIXI$	$YIYIYIY$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$XIXXIXI$	$YYIIYYI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$IZZIIZZ$	$YIYIYIY$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$IZZIIZZ$	$YYIIYYI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$

Table F.10: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the second flagged stabilizer measurement ($IXXIIXX$) for the Steane code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.6 are coloured in red.

$S_{3,1}$	$S_{3,2}$	$S_{3,2,1}$	Possible choices for $S_{3,2,2}$
$XIXIXIX$	$ZIZIZIZ$	$IYYIIYY$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$ZIZIZIZ$	$YYIIYYI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$YIYIYIY$	$IZZIIZZ$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$XIXIXIX$	$YIYIYIY$	$ZZIIZZI$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$IXXXXII$	$ZIZIZIZ$	$IYYIIYY$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$IXXXXII$	$ZIZIZIZ$	$YYIIYYI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$IXXXXII$	$YIYIYIY$	$IZZIIZZ$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$IXXXXII$	$YIYIYIY$	$ZZIIZZI$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$ZIZIZIZ$	$XIXIXIX$	$IZZIIZZ$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$XIXIXIX$	$ZZIIZZI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$XIXIXIX$	$IYYIIYY$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$XIXIXIX$	$YYIIYYI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$IXXXXII$	$IZZIIZZ$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$IXXXXII$	$ZZIIZZI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$IXXXXII$	$IYYIIYY$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$IXXXXII$	$YYIIYYI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$YIYIYIY$	$IZZIIZZ$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$ZIZIZIZ$	$YIYIYIY$	$ZZIIZZI$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$YIYIYIY$	$XIXIXIX$	$IZZIIZZ$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$XIXIXIX$	$ZZIIZZI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$XIXIXIX$	$IYYIIYY$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$XIXIXIX$	$YYIIYYI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$IXXXXII$	$IZZIIZZ$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$IXXXXII$	$ZZIIZZI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$IXXXXII$	$IYYIIYY$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$IXXXXII$	$YYIIYYI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$ZIZIZIZ$	$IYYIIYY$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$ZIZIZIZ$	$YYIIYYI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$

Table F.11: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the first flagged stabilizer measurement ($XIXIXIX$) for the Steane code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.6 are coloured in red.

$S_{4,1}$	$S_{4,2}$	$S_{4,2,1}$	Possible choices for $S_{4,2,2}$
$IIIXXXX$	$IIIZZZZ$	$XIXIXIX$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$IIIZZZZ$	$XIXXIXI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$IIIZZZZ$	$YIYIYIY$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$IIIZZZZ$	$YIYYIYI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$ZZIIZZI$	$XIXIXIX$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$ZZIIZZI$	$XIXXIXI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$ZZIIZZI$	$YIYIYIY$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$ZZIIZZI$	$YIYYIYI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIXXXX$	$IIIYYYY$	$XIXIXIX$	$IYYIIYY$, $YIYIYIY$, $IYYYYII$, $YIYYIYI$
$IIIXXXX$	$IIIYYYY$	$XIXXIXI$	$IYYIIYY$, $YIYIYIY$, $IYYYYII$, $YIYYIYI$
$IIIZZZZ$	$IIIXXXX$	$YIYIYIY$	$IXXIIXX$, $XIXIXIX$, $IXXXXII$, $XIXXIXI$
$IIIZZZZ$	$IIIXXXX$	$YIYYIYI$	$IXXIIXX$, $XIXIXIX$, $IXXXXII$, $XIXXIXI$
$IIIZZZZ$	$IIIYYYY$	$XIXIXIX$	$IYYIIYY$, $YIYIYIY$, $IYYYYII$, $YIYYIYI$
$IIIZZZZ$	$IIIYYYY$	$XIXXIXI$	$IYYIIYY$, $YIYIYIY$, $IYYYYII$, $YIYYIYI$
$ZZIIZZI$	$IIIXXXX$	$YIYIYIY$	$IXXIIXX$, $XIXIXIX$, $IXXXXII$, $XIXXIXI$
$ZZIIZZI$	$IIIXXXX$	$YIYYIYI$	$IXXIIXX$, $XIXIXIX$, $IXXXXII$, $XIXXIXI$
$ZZIIZZI$	$IIIYYYY$	$XIXIXIX$	$IYYIIYY$, $YIYIYIY$, $IYYYYII$, $YIYYIYI$
$ZZIIZZI$	$IIIYYYY$	$XIXXIXI$	$IYYIIYY$, $YIYIYIY$, $IYYYYII$, $YIYYIYI$
$IIIYYYY$	$IIIXXXX$	$YIYIYIY$	$IXXIIXX$, $XIXIXIX$, $IXXXXII$, $XIXXIXI$
$IIIYYYY$	$IIIXXXX$	$YIYYIYI$	$IXXIIXX$, $XIXIXIX$, $IXXXXII$, $XIXXIXI$
$IIIYYYY$	$IIIZZZZ$	$XIXIXIX$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIYYYY$	$IIIZZZZ$	$XIXXIXI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIYYYY$	$IIIZZZZ$	$YIYIYIY$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIYYYY$	$IIIZZZZ$	$YIYYIYI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIYYYY$	$ZZIIZZI$	$XIXIXIX$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIYYYY$	$ZZIIZZI$	$XIXXIXI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIYYYY$	$ZZIIZZI$	$YIYIYIY$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$
$IIIYYYY$	$ZZIIZZI$	$YIYYIYI$	$IZZIIZZ$, $ZIZIZIZ$, $IZZZZII$, $ZIZZIZI$

Table F.12: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the first flagged stabilizer measurement ($IIIZZZZ$) for the Steane code. Refer to Sec 7. The particular sequences derived analytically in the protocol in Fig. 5.6 are coloured in red.

$S_{5,1}$	$S_{5,2}$	$S_{5,2,1}$	Possible choices for $S_{5,2,2}$
$IXXIIXX$	$IZZIIZZ$	$XIXIXIX$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$IZZIIZZ$	$XXIIXXI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$IZZIIZZ$	$YIYIYIY$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$IZZIIZZ$	$YYIIYYI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$ZIZZIZI$	$XIXIXIX$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$ZIZZIZI$	$XXIIXXI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$ZIZZIZI$	$YIYIYIY$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$ZIZZIZI$	$YYIIYYI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IXXIIXX$	$IYYIIYY$	$XIXIXIX$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IXXIIXX$	$IYYIIYY$	$XXIIXXI$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IZZIIZZ$	$IXXIIXX$	$YIYIYIY$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$IXXIIXX$	$YYIIYYI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IZZIIZZ$	$IYYIIYY$	$XIXIXIX$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IZZIIZZ$	$IYYIIYY$	$XXIIXXI$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$ZIZZIZI$	$IXXIIXX$	$YIYIYIY$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$ZIZZIZI$	$IXXIIXX$	$YYIIYYI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$ZIZZIZI$	$IYYIIYY$	$XIXIXIX$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$ZIZZIZI$	$IYYIIYY$	$XXIIXXI$	$IIIYYYY, YIYIYIY, IYYYYII, YYIIYYI$
$IYYIIYY$	$IXXIIXX$	$YIYIYIY$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$IXXIIXX$	$YYIIYYI$	$IIIXXXX, XIXIXIX, IXXXXII, XXIIXXI$
$IYYIIYY$	$IZZIIZZ$	$XIXIXIX$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$IZZIIZZ$	$XXIIXXI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$IZZIIZZ$	$YIYIYIY$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$IZZIIZZ$	$YYIIYYI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$ZIZZIZI$	$XIXIXIX$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$ZIZZIZI$	$XXIIXXI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$ZIZZIZI$	$YIYIYIY$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$
$IYYIIYY$	$ZIZZIZI$	$YYIIYYI$	$IIIZZZZ, ZIZIZIZ, IZZZZII, ZZIIZZI$

Table F.13: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the first flagged stabilizer measurement ($IZZIIZZ$) for the Steane code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.6 are coloured in red.

$S_{6,1}$	$S_{6,2}$	$S_{6,2,1}$	Possible choices for $S_{6,2,2}$
$XIXIXIX$	$ZIZIZIZ$	$IXXIIXX$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$ZIZIZIZ$	$XXIIXXI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$ZIZIZIZ$	$IYYIIYY$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$ZIZIZIZ$	$YYIIYYI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$IZZZZII$	$IXXIIXX$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$IZZZZII$	$XXIIXXI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$IZZZZII$	$IYYIIYY$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$IZZZZII$	$YYIIYYI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$XIXIXIX$	$YIYIYIY$	$IXXIIXX$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$XIXIXIX$	$YIYIYIY$	$XXIIXXI$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$ZIZIZIZ$	$XIXIXIX$	$IYYIIYY$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$XIXIXIX$	$YYIIYYI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$ZIZIZIZ$	$YIYIYIY$	$IXXIIXX$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$ZIZIZIZ$	$YIYIYIY$	$XXIIXXI$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$IZZZZII$	$XIXIXIX$	$IYYIIYY$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$IZZZZII$	$XIXIXIX$	$YYIIYYI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$IZZZZII$	$YIYIYIY$	$IXXIIXX$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$IZZZZII$	$YIYIYIY$	$XXIIXXI$	$IIIYYYY, IYYIIYY, YIYYIYI, YYIIYYI$
$YIYIYIY$	$XIXIXIX$	$IYYIIYY$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$XIXIXIX$	$YYIIYYI$	$IIIXXXX, IXXIIXX, XIXXIXI, XXIIXXI$
$YIYIYIY$	$ZIZIZIZ$	$IXXIIXX$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$ZIZIZIZ$	$XXIIXXI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$ZIZIZIZ$	$IYYIIYY$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$ZIZIZIZ$	$YYIIYYI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$IZZZZII$	$IXXIIXX$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$IZZZZII$	$XXIIXXI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$IZZZZII$	$IYYIIYY$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$
$YIYIYIY$	$IZZZZII$	$YYIIYYI$	$IIIZZZZ, IZZIIZZ, ZIZZIZI, ZZIIZZI$

Table F.14: Reduced, adaptive stabilizer sequences obtained by computer search, which yield unique and nontrivial syndromes, for the $f = 1$ branches resulting from the first flagged stabilizer measurement ($ZIZIZIZ$) for the Steane code. Refer to Sec. 7. The particular sequences derived analytically in the protocol in Fig. 5.6 are coloured in red.

G

Detect-and-Diagnose-Style Stabilizer Sequences

This section presents sequences of 3 high-weight stabilizers of the Steane code, obtained by computer search, which can detect 1 input error. These may be useful as candidate sequences for detect-and-diagnose-style flag protocols. These are presented in Tables G.1-G.16. Refer to Sec. 7.

The columns on the left hand side present the actual sequences enclosed in braces and separated by commas. Sequences resulting from permutations among stabilizers are not counted separately. The number in parentheses following the stabilizer is the arbitrarily assigned coset number presented in Fig. 7.1. These numbers are presented here so that it may be verified that, for every triplet which satisfies the criteria, each stabilizer belongs to a different coset, and none belongs to the subgroup of X-type stabilizers (coset (0)). The columns on the right hand side present the 8-element subgroup of the stabilizer group generated by each of the sequences on the left (which are organized according to the distinct subgroup they generate). The weight-6 stabilizers are presented without the negative signs for brevity. The particular sequence derived analytically for the protocol in Fig. 5.7 is coloured in red (see Table G.9).

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IXXZZYY(1), XZYIXZY(2), ZXYXYIZ(3)\}$, $\{IXXZZYY(1), ZIZYXYX(5), XZYIXZY(2)\}$, $\{IXXZZYY(1), YYIXZZX(6), XZYIXZY(2)\}$, $\{IXXZZYY(1), YZXYIXZ(7), XZYIXZY(2)\}$, $\{IXXZZYY(1), ZXYXYIZ(3), XYZZYXI(4)\}$, $\{IXXZZYY(1), YYIXZZX(6), ZXYXYIZ(3)\}$, $\{IXXZZYY(1), YZXYIXZ(7), ZXYXYIZ(3)\}$, $\{IXXZZYY(1), ZIZYXYX(5), XYZZYXI(4)\}$, $\{IXXZZYY(1), YYIXZZX(6), XYZZYXI(4)\}$, $\{IXXZZYY(1), ZIZYXYX(5), YYIXZZX(6)\}$, $\{IXXZZYY(1), ZIZYXYX(5), YZXYIXZ(7)\}$, $\{XZYIXZY(2), ZXYXYIZ(3), XYZZYXI(4)\}$, $\{ZIZYXYX(5), XZYIXZY(2), ZXYXYIZ(3)\}$, $\{YZXYIXZ(7), XZYIXZY(2), ZXYXYIZ(3)\}$, $\{ZIZYXYX(5), XZYIXZY(2), XYZZYXI(4)\}$, $\{YYIXZZX(6), XZYIXZY(2), XYZZYXI(4)\}$, $\{YZXYIXZ(7), XZYIXZY(2), XYZZYXI(4)\}$, $\{ZIZYXYX(5), YYIXZZX(6), XZYIXZY(2)\}$, $\{YZXYIXZ(7), YYIXZZX(6), XZYIXZY(2)\}$, $\{ZIZYXYX(5), ZXYXYIZ(3), XYZZYXI(4)\}$, $\{YYIXZZX(6), ZXYXYIZ(3), XYZZYXI(4)\}$, $\{ZIZYXYX(5), YYIXZZX(6), ZXYXYIZ(3)\}$, $\{ZIZYXYX(5), YZXYIXZ(7), ZXYXYIZ(3)\}$, $\{YYIXZZX(6), YZXYIXZ(7), ZXYXYIZ(3)\}$, $\{ZIZYXYX(5), YZXYIXZ(7), XYZZYXI(4)\}$, $\{YYIXZZX(6), YZXYIXZ(7), XYZZYXI(4)\}$, $\{ZIZYXYX(5), YYIXZZX(6), YZXYIXZ(7)\}$	$\{IXXZZYY, XYZZYXI, XZYIXZY, YYIXZZX, YZXYIXZ, ZIZYXYX, ZXYXYIZ, IIIIIII\}$
$\{IXXZZYY(1), YIYXZXZ(3), XZYIXZY(2)\}$, $\{IXXZZYY(1), YXZYIZX(5), XZYIXZY(2)\}$, $\{IXXZZYY(1), ZZIXYYX(6), XZYIXZY(2)\}$, $\{IXXZZYY(1), ZYXYXIZ(7), XZYIXZY(2)\}$, $\{IXXZZYY(1), YIYXZXZ(3), XYZZYXI(4)\}$, $\{IXXZZYY(1), YIYXZXZ(3), ZZIXYYX(6)\}$, $\{IXXZZYY(1), YIYXZXZ(3), ZYXYXIZ(7)\}$, $\{IXXZZYY(1), YXZYIZX(5), XYZZYXI(4)\}$, $\{IXXZZYY(1), ZZIXYYX(6), XYZZYXI(4)\}$, $\{IXXZZYY(1), ZYXYXIZ(7), XYZZYXI(4)\}$, $\{IXXZZYY(1), YXZYIZX(5), ZZIXYYX(6)\}$, $\{IXXZZYY(1), YXZYIZX(5), ZYXYXIZ(7)\}$, $\{YIYXZXZ(3), XZYIXZY(2), XYZZYXI(4)\}$, $\{YIYXZXZ(3), YXZYIZX(5), XZYIXZY(2)\}$, $\{YIYXZXZ(3), ZYXYXIZ(7), XZYIXZY(2)\}$, $\{YXZYIZX(5), XZYIXZY(2), XYZZYXI(4)\}$, $\{ZZIXYYX(6), XZYIXZY(2), XYZZYXI(4)\}$, $\{ZYXYXIZ(7), XZYIXZY(2), XYZZYXI(4)\}$, $\{YXZYIZX(5), XZYIXZY(2), ZZIXYYX(6)\}$, $\{ZYXYXIZ(7), ZZIXYYX(6), XZYIXZY(2)\}$, $\{YIYXZXZ(3), YXZYIZX(5), XYZZYXI(4)\}$, $\{YIYXZXZ(3), ZZIXYYX(6), XYZZYXI(4)\}$, $\{YIYXZXZ(3), YXZYIZX(5), ZZIXYYX(6)\}$, $\{YIYXZXZ(3), YXZYIZX(5), ZYXYXIZ(7)\}$, $\{YIYXZXZ(3), ZZIXYYX(6), ZYXYXIZ(7)\}$, $\{YXZYIZX(5), ZYXYXIZ(7), XYZZYXI(4)\}$, $\{ZZIXYYX(6), ZYXYXIZ(7), XYZZYXI(4)\}$, $\{YXZYIZX(5), ZYXYXIZ(7), ZZIXYYX(6)\}$	$\{IXXZZYY, XYZZYXI, XZYIXZY, YIYXZXZ, YXZYIZX, ZYXYXIZ, ZZIXYYX, IIIIIII\}$
$\{IXXZZYY(1), ZIZXYXY(3), XZYXIYZ(2)\}$, $\{IXXZZYY(1), ZXYYXZI(5), XZYXIYZ(2)\}$, $\{IXXZZYY(1), YZXIYZX(6), XZYXIYZ(2)\}$, $\{IXXZZYY(1), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{IXXZZYY(1), ZIZXYXY(3), XYZYZIX(4)\}$, $\{IXXZZYY(1), YZXIYZX(6), ZIZXYXY(3)\}$, $\{IXXZZYY(1), ZIZXYXY(3), YYIZXXZ(7)\}$, $\{IXXZZYY(1), ZXYYXZI(5), XYZYZIX(4)\}$, $\{IXXZZYY(1), YZXIYZX(6), XYZYZIX(4)\}$, $\{IXXZZYY(1), YYIZXXZ(7), XYZYZIX(4)\}$, $\{IXXZZYY(1), ZXYYXZI(5), YZXIYZX(6)\}$, $\{IXXZZYY(1), ZXYYXZI(5), YYIZXXZ(7)\}$, $\{ZIZXYXY(3), XYZYZIX(4), XZYXIYZ(2)\}$, $\{ZXYYXZI(5), ZIZXYXY(3), XZYXIYZ(2)\}$, $\{ZIZXYXY(3), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{ZXYYXZI(5), XYZYZIX(4), XZYXIYZ(2)\}$, $\{YZXIYZX(6), XYZYZIX(4), XZYXIYZ(2)\}$, $\{XYZYZIX(4), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{ZXYYXZI(5), YZXIYZX(6), XZYXIYZ(2)\}$, $\{YZXIYZX(6), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{ZXYYXZI(5), ZIZXYXY(3), XYZYZIX(4)\}$, $\{YZXIYZX(6), ZIZXYXY(3), XYZYZIX(4)\}$, $\{ZXYYXZI(5), ZIZXYXY(3), YZXIYZX(6)\}$, $\{ZXYYXZI(5), ZIZXYXY(3), YYIZXXZ(7)\}$, $\{YZXIYZX(6), ZIZXYXY(3), YYIZXXZ(7)\}$, $\{ZXYYXZI(5), YYIZXXZ(7), XYZYZIX(4)\}$, $\{YZXIYZX(6), YYIZXXZ(7), XYZYZIX(4)\}$, $\{ZXYYXZI(5), YYIZXXZ(7), YZXIYZX(6)\}$	$\{IXXZZYY, XYZYZIX, XZYXIYZ, YYIZXXZ, YZXIYZX, ZIZXYXY, ZXYYXZI, IIIIIII\}$

Table G.1: Detect-and-diagnose-style stabilizer triplets for the Steane code, obtained by computer search. Refer to Sec. 7. The left columns present the sequences. Permutations among stabilizers are not counted separately. The number in parentheses following the stabilizer is the arbitrarily assigned coset number presented in Fig. 7.1. The right column presents the subgroup of the stabilizer group generated by each of the sequences on the left. Negative signs are dropped from weight-6 stabilizers for brevity. The particular sequence derived analytically for the protocol in Fig. 5.7 is coloured in red (see Table G.9). Continued in Tables G.2-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IXXZZYY(1), YXZIYXZ(3), XZYXIYZ(2)\}$, $\{IXXZZYY(1), YIYZXZX(5), XZYXIYZ(2)\}$, $\{IXXZZYY(1), ZYXXYZI(6), XZYXIYZ(2)\}$, $\{IXXZZYY(1), ZZIYXXY(7), XZYXIYZ(2)\}$, $\{IXXZZYY(1), YXZIYXZ(3), XYZYZIX(4)\}$, $\{IXXZZYY(1), ZYXXYZI(6), YXZIYXZ(3)\}$, $\{IXXZZYY(1), YIYZXZX(5), XYZYZIX(4)\}$, $\{IXXZZYY(1), ZYXXYZI(6), XYZYZIX(4)\}$, $\{IXXZZYY(1), ZZIYXXY(7), XYZYZIX(4)\}$, $\{IXXZZYY(1), YIYZXZX(5), ZYXXYZI(6)\}$, $\{IXXZZYY(1), YIYZXZX(5), ZZIYXXY(7)\}$, $\{YXZIYXZ(3), XYZYZIX(4), XZYXIYZ(2)\}$, $\{YIYZXZX(5), YXZIYXZ(3), XZYXIYZ(2)\}$, $\{ZZIYXXY(7), YXZIYXZ(3), XZYXIYZ(2)\}$, $\{YIYZXZX(5), XYZYZIX(4), XZYXIYZ(2)\}$, $\{ZYXXYZI(6), XYZYZIX(4), XZYXIYZ(2)\}$, $\{ZZIYXXY(7), XYZYZIX(4), XZYXIYZ(2)\}$, $\{YIYZXZX(5), ZYXXYZI(6), XZYXIYZ(2)\}$, $\{ZZIYXXY(7), ZYXXYZI(6), XZYXIYZ(2)\}$, $\{YIYZXZX(5), YXZIYXZ(3), XYZYZIX(4)\}$, $\{ZYXXYZI(6), YXZIYXZ(3), XYZYZIX(4)\}$, $\{YIYZXZX(5), YXZIYXZ(3), ZYXXYZI(6)\}$, $\{YIYZXZX(5), YXZIYXZ(3), ZZIYXXY(7)\}$, $\{ZYXXYZI(6), YXZIYXZ(3), ZZIYXXY(7)\}$, $\{YIYZXZX(5), ZZIYXXY(7), XYZYZIX(4)\}$, $\{ZZIYXXY(7), ZYXXYZI(6), XYZYZIX(4)\}$, $\{YIYZXZX(5), ZYXXYZI(6), ZZIYXXY(7)\}$	$\{IXXZZYY, XYZYZIX, XZYXIYZ, YIYZXZX, YXZIYXZ, ZYXXYZI, ZZIYXXY, IIIIIII\}$
$\{IXXZZYY(1), XYZIXYZ(2), ZIZXYXY(3)\}$, $\{IXXZZYY(1), ZXYYXZI(5), XYZIXYZ(2)\}$, $\{IXXZZYY(1), XYZIXYZ(2), YYIXZZX(6)\}$, $\{IXXZZYY(1), XYZIXYZ(2), YZXYIXZ(7)\}$, $\{IXXZZYY(1), ZIZXYXY(3), XZYZYIX(4)\}$, $\{IXXZZYY(1), YYIXZZX(6), ZIZXYXY(3)\}$, $\{IXXZZYY(1), ZIZXYXY(3), YZXYIXZ(7)\}$, $\{IXXZZYY(1), ZXYYXZI(5), XZYZYIX(4)\}$, $\{IXXZZYY(1), YYIXZZX(6), XZYZYIX(4)\}$, $\{IXXZZYY(1), YZXYIXZ(7), XZYZYIX(4)\}$, $\{IXXZZYY(1), ZXYYXZI(5), YYIXZZX(6)\}$, $\{IXXZZYY(1), ZXYYXZI(5), YZXYIXZ(7)\}$, $\{XZYZYIX(4), XYZIXYZ(2), ZIZXYXY(3)\}$, $\{ZXYYXZI(5), XYZIXYZ(2), ZIZXYXY(3)\}$, $\{XYZIXYZ(2), ZIZXYXY(3), YZXYIXZ(7)\}$, $\{XZYZYIX(4), ZXYYXZI(5), XYZIXYZ(2)\}$, $\{XZYZYIX(4), XYZIXYZ(2), YYIXZZX(6)\}$, $\{XZYZYIX(4), XYZIXYZ(2), YZXYIXZ(7)\}$, $\{ZXYYXZI(5), XYZIXYZ(2), YYIXZZX(6)\}$, $\{XYZIXYZ(2), YYIXZZX(6), YZXYIXZ(7)\}$, $\{XZYZYIX(4), ZXYYXZI(5), ZIZXYXY(3)\}$, $\{XZYZYIX(4), YYIXZZX(6), ZIZXYXY(3)\}$, $\{ZXYYXZI(5), ZIZXYXY(3), YYIXZZX(6)\}$, $\{ZXYYXZI(5), ZIZXYXY(3), YZXYIXZ(7)\}$, $\{YYIXZZX(6), ZIZXYXY(3), YZXYIXZ(7)\}$, $\{XZYZYIX(4), ZXYYXZI(5), YZXYIXZ(7)\}$, $\{XZYZYIX(4), YYIXZZX(6), YZXYIXZ(7)\}$, $\{ZXYYXZI(5), YYIXZZX(6), YZXYIXZ(7)\}$	$\{IXXZZYY, XYZIXYZ, XZYZYIX, YYIXZZX, YZXYIXZ, ZIZXYXY, ZXYYXZI, IIIIIII\}$
$\{IXXZZYY(1), YIYXZXZ(3), XYZIXYZ(2)\}$, $\{IXXZZYY(1), YXZYIZX(5), XYZIXYZ(2)\}$, $\{IXXZZYY(1), XYZIXYZ(2), ZYXXYZI(6)\}$, $\{IXXZZYY(1), XYZIXYZ(2), ZZIYXXY(7)\}$, $\{IXXZZYY(1), YIYXZXZ(3), XZYZYIX(4)\}$, $\{IXXZZYY(1), YIYXZXZ(3), ZYXXYZI(6)\}$, $\{IXXZZYY(1), YIYXZXZ(3), ZZIYXXY(7)\}$, $\{IXXZZYY(1), YXZYIZX(5), XZYZYIX(4)\}$, $\{IXXZZYY(1), ZYXXYZI(6), XZYZYIX(4)\}$, $\{IXXZZYY(1), ZZIYXXY(7), XZYZYIX(4)\}$, $\{IXXZZYY(1), YXZYIZX(5), ZYXXYZI(6)\}$, $\{IXXZZYY(1), YXZYIZX(5), ZZIYXXY(7)\}$, $\{XZYZYIX(4), YIYXZXZ(3), XYZIXYZ(2)\}$, $\{YIYXZXZ(3), YXZYIZX(5), XYZIXYZ(2)\}$, $\{YIYXZXZ(3), XYZIXYZ(2), ZZIYXXY(7)\}$, $\{XZYZYIX(4), YXZYIZX(5), XYZIXYZ(2)\}$, $\{XZYZYIX(4), XYZIXYZ(2), ZYXXYZI(6)\}$, $\{XZYZYIX(4), XYZIXYZ(2), ZZIYXXY(7)\}$, $\{YXZYIZX(5), XYZIXYZ(2), ZYXXYZI(6)\}$, $\{XYZIXYZ(2), ZYXXYZI(6), ZZIYXXY(7)\}$, $\{XZYZYIX(4), YIYXZXZ(3), YXZYIZX(5)\}$, $\{XZYZYIX(4), YIYXZXZ(3), ZYXXYZI(6)\}$, $\{YIYXZXZ(3), YXZYIZX(5), ZYXXYZI(6)\}$, $\{YIYXZXZ(3), YXZYIZX(5), ZZIYXXY(7)\}$, $\{YIYXZXZ(3), ZZIYXXY(7), ZYXXYZI(6)\}$, $\{XZYZYIX(4), YXZYIZX(5), ZZIYXXY(7)\}$, $\{XZYZYIX(4), ZZIYXXY(7), ZYXXYZI(6)\}$, $\{YXZYIZX(5), ZZIYXXY(7), ZYXXYZI(6)\}$	$\{IXXZZYY, XYZIXYZ, XZYZYIX, YIYXZXZ, YXZYIZX, ZYXXYZI, ZZIYXXY, IIIIIII\}$

Table G.2: Continued from Table G.1 and in Tables G.3-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IXXZZYY(1), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{IXXZZYY(1), ZIZYXYX(5), XYZXIZY(2)\}$, $\{IXXZZYY(1), YZXIYZX(6), XYZXIZY(2)\}$, $\{IXXZZYY(1), YYIZXXZ(7), XYZXIZY(2)\}$, $\{IXXZZYY(1), XZYYZXI(4), ZXYXYIZ(3)\}$, $\{IXXZZYY(1), YZXIYZX(6), ZXYXYIZ(3)\}$, $\{IXXZZYY(1), YYIZXXZ(7), ZXYXYIZ(3)\}$, $\{IXXZZYY(1), ZIZYXYX(5), XZYYZXI(4)\}$, $\{IXXZZYY(1), YZXIYZX(6), XZYYZXI(4)\}$, $\{IXXZZYY(1), XZYYZXI(4), YYIZXXZ(7)\}$, $\{IXXZZYY(1), YZXIYZX(6), ZIZYXYX(5)\}$, $\{IXXZZYY(1), ZIZYXYX(5), YYIZXXZ(7)\}$, $\{XZYYZXI(4), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{ZIZYXYX(5), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{YYIZXXZ(7), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{ZIZYXYX(5), XZYYZXI(4), XYZXIZY(2)\}$, $\{YZXIYZX(6), XZYYZXI(4), XYZXIZY(2)\}$, $\{XZYYZXI(4), YYIZXXZ(7), XYZXIZY(2)\}$, $\{YZXIYZX(6), ZIZYXYX(5), XYZXIZY(2)\}$, $\{YZXIYZX(6), YYIZXXZ(7), XYZXIZY(2)\}$, $\{ZIZYXYX(5), XZYYZXI(4), ZXYXYIZ(3)\}$, $\{YZXIYZX(6), XZYYZXI(4), ZXYXYIZ(3)\}$, $\{YZXIYZX(6), ZIZYXYX(5), ZXYXYIZ(3)\}$, $\{YYIZXXZ(7), ZIZYXYX(5), ZXYXYIZ(3)\}$, $\{YZXIYZX(6), YYIZXXZ(7), ZXYXYIZ(3)\}$, $\{ZIZYXYX(5), XZYYZXI(4), YYIZXXZ(7)\}$, $\{YZXIYZX(6), XZYYZXI(4), YYIZXXZ(7)\}$, $\{YZXIYZX(6), ZIZYXYX(5), YYIZXXZ(7)\}$	$\{IXXZZYY, XYZXIZY, XZYYZXI, YYIZXXZ, YZXIYZX, ZIZYXYX, ZXYXYIZ, IIIIIII\}$
$\{IXXZZYY(1), YXZIYXZ(3), XYZXIZY(2)\}$, $\{IXXZZYY(1), YIYZXZX(5), XYZXIZY(2)\}$, $\{IXXZZYY(1), ZZIXYYX(6), XYZXIZY(2)\}$, $\{IXXZZYY(1), ZYXYXIZ(7), XYZXIZY(2)\}$, $\{IXXZZYY(1), YXZIYXZ(3), XZYYZXI(4)\}$, $\{IXXZZYY(1), ZZIXYYX(6), YXZIYXZ(3)\}$, $\{IXXZZYY(1), YXZIYXZ(3), ZYXYXIZ(7)\}$, $\{IXXZZYY(1), YIYZXZX(5), XZYYZXI(4)\}$, $\{IXXZZYY(1), ZZIXYYX(6), XZYYZXI(4)\}$, $\{IXXZZYY(1), ZYXYXIZ(7), XZYYZXI(4)\}$, $\{IXXZZYY(1), ZZIXYYX(6), YIYZXZX(5)\}$, $\{IXXZZYY(1), YIYZXZX(5), ZYXYXIZ(7)\}$, $\{YXZIYXZ(3), XZYYZXI(4), XYZXIZY(2)\}$, $\{YIYZXZX(5), YXZIYXZ(3), XYZXIZY(2)\}$, $\{YXZIYXZ(3), ZYXYXIZ(7), XYZXIZY(2)\}$, $\{YIYZXZX(5), XZYYZXI(4), XYZXIZY(2)\}$, $\{ZZIXYYX(6), XZYYZXI(4), XYZXIZY(2)\}$, $\{ZYXYXIZ(7), XZYYZXI(4), XYZXIZY(2)\}$, $\{ZZIXYYX(6), YIYZXZX(5), XYZXIZY(2)\}$, $\{ZZIXYYX(6), ZYXYXIZ(7), XYZXIZY(2)\}$, $\{YIYZXZX(5), YXZIYXZ(3), XZYYZXI(4)\}$, $\{ZZIXYYX(6), YXZIYXZ(3), XZYYZXI(4)\}$, $\{ZZIXYYX(6), YIYZXZX(5), YXZIYXZ(3)\}$, $\{YIYZXZX(5), YXZIYXZ(3), ZYXYXIZ(7)\}$, $\{ZZIXYYX(6), YXZIYXZ(3), ZYXYXIZ(7)\}$, $\{ZYXYXIZ(7), YIYZXZX(5), XZYYZXI(4)\}$, $\{ZYXYXIZ(7), ZZIXYYX(6), XZYYZXI(4)\}$, $\{ZZIXYYX(6), YIYZXZX(5), ZYXYXIZ(7)\}$	$\{IXXZZYY, XYZXIZY, XZYYZXI, YIYZXZX, YXZIYXZ, ZYXYXIZ, ZZIXYYX, IIIIIII\}$
$\{IZZXXYY(2), XIXZYZY(1), ZXYXYIZ(3)\}$, $\{IZZXXYY(2), XIXZYZY(1), YXZYIZX(5)\}$, $\{IZZXXYY(2), XIXZYZY(1), ZYXIZYX(6)\}$, $\{IZZXXYY(2), XIXZYZY(1), YYIZXXZ(7)\}$, $\{XIXZYZY(1), XZYYZXI(4), ZXYXYIZ(3)\}$, $\{XIXZYZY(1), ZXYXYIZ(3), ZYXIZYX(6)\}$, $\{YYIZXXZ(7), XIXZYZY(1), ZXYXYIZ(3)\}$, $\{YXZYIZX(5), XIXZYZY(1), XZYYZXI(4)\}$, $\{XIXZYZY(1), XZYYZXI(4), ZYXIZYX(6)\}$, $\{XIXZYZY(1), XZYYZXI(4), YYIZXXZ(7)\}$, $\{YXZYIZX(5), XIXZYZY(1), ZYXIZYX(6)\}$, $\{YXZYIZX(5), XIXZYZY(1), YYIZXXZ(7)\}$, $\{IZZXXYY(2), XZYYZXI(4), ZXYXYIZ(3)\}$, $\{IZZXXYY(2), ZXYXYIZ(3), YXZYIZX(5)\}$, $\{YYIZXXZ(7), IZZXXYY(2), ZXYXYIZ(3)\}$, $\{IZZXXYY(2), XZYYZXI(4), YXZYIZX(5)\}$, $\{IZZXXYY(2), XZYYZXI(4), ZYXIZYX(6)\}$, $\{IZZXXYY(2), XZYYZXI(4), YYIZXXZ(7)\}$, $\{IZZXXYY(2), YXZYIZX(5), ZYXIZYX(6)\}$, $\{IZZXXYY(2), YYIZXXZ(7), ZYXIZYX(6)\}$, $\{YXZYIZX(5), XZYYZXI(4), ZXYXYIZ(3)\}$, $\{XZYYZXI(4), ZXYXYIZ(3), ZYXIZYX(6)\}$, $\{YXZYIZX(5), ZXYXYIZ(3), ZYXIZYX(6)\}$, $\{YYIZXXZ(7), YXZYIZX(5), ZXYXYIZ(3)\}$, $\{YYIZXXZ(7), ZXYXYIZ(3), ZYXIZYX(6)\}$, $\{YXZYIZX(5), XZYYZXI(4), YYIZXXZ(7)\}$, $\{XZYYZXI(4), YYIZXXZ(7), ZYXIZYX(6)\}$, $\{YXZYIZX(5), YYIZXXZ(7), ZYXIZYX(6)\}$	$\{IZZXXYY, XIXZYZY, XZYYZXI, YXZYIZX, YYIZXXZ, ZXYXYIZ, ZYXIZYX, IIIIIII\}$

Table G.3: Continued from Tables G.1-G.2 and in Tables G.4-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IZZXXYY(2), XIXZYZY(1), YXZIYXZ(3)\}$, $\{IZZXXYY(2), XIXZYZY(1), ZXYZIYX(5)\}$, $\{YYIXZZX(6), IZZXXYY(2), XIXZYZY(1)\}$, $\{IZZXXYY(2), XIXZYZY(1), ZYXYXIZ(7)\}$, $\{YXZIYXZ(3), XIXZYZY(1), XZYYZXI(4)\}$, $\{YYIXZZX(6), YXZIYXZ(3), XIXZYZY(1)\}$, $\{YXZIYXZ(3), XIXZYZY(1), ZYXYXIZ(7)\}$, $\{ZXYZIYX(5), XIXZYZY(1), XZYYZXI(4)\}$, $\{YYIXZZX(6), XIXZYZY(1), XZYYZXI(4)\}$, $\{ZYXYXIZ(7), XIXZYZY(1), XZYYZXI(4)\}$, $\{YYIXZZX(6), XIXZYZY(1), ZXYZIYX(5)\}$, $\{ZYXYXIZ(7), XIXZYZY(1), ZXYZIYX(5)\}$, $\{IZZXXYY(2), YXZIYXZ(3), XZYYZXI(4)\}$, $\{IZZXXYY(2), YXZIYXZ(3), ZXYZIYX(5)\}$, $\{IZZXXYY(2), YXZIYXZ(3), ZYXYXIZ(7)\}$, $\{IZZXXYY(2), ZXYZIYX(5), XZYYZXI(4)\}$, $\{IZZXXYY(2), YYIXZZX(6), XZYYZXI(4)\}$, $\{ZYXYXIZ(7), IZZXXYY(2), XZYYZXI(4)\}$, $\{IZZXXYY(2), YYIXZZX(6), ZXYZIYX(5)\}$, $\{IZZXXYY(2), YYIXZZX(6), ZYXYXIZ(7)\}$, $\{ZXYZIYX(5), YXZIYXZ(3), XZYYZXI(4)\}$, $\{YYIXZZX(6), YXZIYXZ(3), XZYYZXI(4)\}$, $\{YYIXZZX(6), YXZIYXZ(3), ZXYZIYX(5)\}$, $\{ZYXYXIZ(7), YXZIYXZ(3), ZXYZIYX(5)\}$, $\{YYIXZZX(6), YXZIYXZ(3), ZYXYXIZ(7)\}$, $\{ZYXYXIZ(7), ZXYZIYX(5), XZYYZXI(4)\}$, $\{ZYXYXIZ(7), YYIXZZX(6), XZYYZXI(4)\}$, $\{ZYXYXIZ(7), YYIXZZX(6), ZXYZIYX(5)\}$	$\{IZZXXYY, XIXZYZY, XZYYZXI, YXZIYXZ, YYIXZZX, ZXYZIYX, ZYXYXIZ, IIIIIII\}$
$\{IYYXXZZ(2), XIXZYZY(1), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), XIXZYZY(1), YXZZXYI(5)\}$, $\{IYYXXZZ(2), XIXZYZY(1), ZZIXYYX(6)\}$, $\{IYYXXZZ(2), XIXZYZY(1), YZXYIXZ(7)\}$, $\{XIXZYZY(1), XYZYZIX(4), ZXYIZXY(3)\}$, $\{ZZIXYYX(6), XIXZYZY(1), ZXYIZXY(3)\}$, $\{XIXZYZY(1), YXZZXYI(5), ZXYIZXY(3)\}$, $\{XIXZYZY(1), YXZZXYI(5), XYZYZIX(4)\}$, $\{ZZIXYYX(6), XIXZYZY(1), XYZYZIX(4)\}$, $\{XIXZYZY(1), YZXYIXZ(7), XYZYZIX(4)\}$, $\{ZZIXYYX(6), XIXZYZY(1), YXZZXYI(5)\}$, $\{YZXYIXZ(7), XIXZYZY(1), YXZZXYI(5)\}$, $\{IYYXXZZ(2), XYZYZIX(4), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), YXZZXYI(5), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), YZXYIXZ(7), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), YXZZXYI(5), XYZYZIX(4)\}$, $\{IYYXXZZ(2), ZZIXYYX(6), XYZYZIX(4)\}$, $\{IYYXXZZ(2), YZXYIXZ(7), XYZYZIX(4)\}$, $\{IYYXXZZ(2), YXZZXYI(5), ZZIXYYX(6)\}$, $\{IYYXXZZ(2), YZXYIXZ(7), ZZIXYYX(6)\}$, $\{YXZZXYI(5), XYZYZIX(4), ZXYIZXY(3)\}$, $\{ZZIXYYX(6), XYZYZIX(4), ZXYIZXY(3)\}$, $\{ZZIXYYX(6), YXZZXYI(5), ZXYIZXY(3)\}$, $\{YZXYIXZ(7), YXZZXYI(5), ZXYIZXY(3)\}$, $\{ZZIXYYX(6), YZXYIXZ(7), ZXYIZXY(3)\}$, $\{YZXYIXZ(7), YXZZXYI(5), XYZYZIX(4)\}$, $\{ZZIXYYX(6), YZXYIXZ(7), XYZYZIX(4)\}$, $\{ZZIXYYX(6), YZXYIXZ(7), YXZZXYI(5)\}$	$\{IYYXXZZ, XIXZYZY, XYZYZIX, YXZZXYI, YZXYIXZ, ZXYIZXY, ZZIXYYX, IIIIIII\}$
$\{IYYXXZZ(2), XIXZYZY(1), YXZIYXZ(3)\}$, $\{IYYXXZZ(2), XIXZYZY(1), ZXYZIYX(5)\}$, $\{YZXXZYI(6), IYYXXZZ(2), XIXZYZY(1)\}$, $\{IYYXXZZ(2), ZZIYXXY(7), XIXZYZY(1)\}$, $\{YXZIYXZ(3), XIXZYZY(1), XYZYZIX(4)\}$, $\{YZXXZYI(6), YXZIYXZ(3), XIXZYZY(1)\}$, $\{YXZIYXZ(3), ZZIYXXY(7), XIXZYZY(1)\}$, $\{XIXZYZY(1), ZXYZIYX(5), XYZYZIX(4)\}$, $\{YZXXZYI(6), XIXZYZY(1), XYZYZIX(4)\}$, $\{ZZIYXXY(7), XIXZYZY(1), XYZYZIX(4)\}$, $\{YZXXZYI(6), XIXZYZY(1), ZXYZIYX(5)\}$, $\{ZZIYXXY(7), XIXZYZY(1), ZXYZIYX(5)\}$, $\{IYYXXZZ(2), YXZIYXZ(3), XYZYZIX(4)\}$, $\{IYYXXZZ(2), YXZIYXZ(3), ZXYZIYX(5)\}$, $\{IYYXXZZ(2), ZZIYXXY(7), YXZIYXZ(3)\}$, $\{IYYXXZZ(2), ZXYZIYX(5), XYZYZIX(4)\}$, $\{IYYXXZZ(2), YZXXZYI(6), XYZYZIX(4)\}$, $\{IYYXXZZ(2), ZZIYXXY(7), XYZYZIX(4)\}$, $\{IYYXXZZ(2), YZXXZYI(6), ZXYZIYX(5)\}$, $\{IYYXXZZ(2), ZZIYXXY(7), YZXXZYI(6)\}$, $\{YXZIYXZ(3), ZXYZIYX(5), XYZYZIX(4)\}$, $\{YZXXZYI(6), YXZIYXZ(3), XYZYZIX(4)\}$, $\{YZXXZYI(6), YXZIYXZ(3), ZXYZIYX(5)\}$, $\{ZZIYXXY(7), YXZIYXZ(3), ZXYZIYX(5)\}$, $\{YZXXZYI(6), ZZIYXXY(7), YXZIYXZ(3)\}$, $\{ZZIYXXY(7), ZXYZIYX(5), XYZYZIX(4)\}$, $\{YZXXZYI(6), ZZIYXXY(7), XYZYZIX(4)\}$, $\{YZXXZYI(6), ZZIYXXY(7), ZXYZIYX(5)\}$	$\{IYYXXZZ, XIXZYZY, XYZYZIX, YXZIYXZ, YZXXZYI, ZXYZIYX, ZZIYXXY, IIIIIII\}$

Table G.4: Continued from Tables G.1-G.3 and in Tables G.5-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{ZXYIZXY(3), XIXZYZY(1), XZYXIYZ(2)\}$, $\{XIXZYZY(1), YXZZXYI(5), XZYXIYZ(2)\}$, $\{YYIXZZX(6), XIXZYZY(1), XZYXIYZ(2)\}$, $\{XIXZYZY(1), ZYXYXIZ(7), XZYXIYZ(2)\}$, $\{IZZYYXX(4), XIXZYZY(1), ZXYIZXY(3)\}$, $\{YYIXZZX(6), XIXZYZY(1), ZXYIZXY(3)\}$, $\{XIXZYZY(1), ZYXYXIZ(7), ZXYIZXY(3)\}$, $\{IZZYYXX(4), XIXZYZY(1), YXZZXYI(5)\}$, $\{IZZYYXX(4), YYIXZZX(6), XIXZYZY(1)\}$, $\{IZZYYXX(4), XIXZYZY(1), ZYXYXIZ(7)\}$, $\{YYIXZZX(6), XIXZYZY(1), YXZZXYI(5)\}$, $\{ZYXYXIZ(7), XIXZYZY(1), YXZZXYI(5)\}$, $\{IZZYYXX(4), ZXYIZXY(3), XZYXIYZ(2)\}$, $\{ZXYIZXY(3), YXZZXYI(5), XZYXIYZ(2)\}$, $\{ZXYIZXY(3), ZYXYXIZ(7), XZYXIYZ(2)\}$, $\{IZZYYXX(4), YXZZXYI(5), XZYXIYZ(2)\}$, $\{IZZYYXX(4), YYIXZZX(6), XZYXIYZ(2)\}$, $\{IZZYYXX(4), ZYXYXIZ(7), XZYXIYZ(2)\}$, $\{YYIXZZX(6), YXZZXYI(5), XZYXIYZ(2)\}$, $\{YYIXZZX(6), ZYXYXIZ(7), XZYXIYZ(2)\}$, $\{IZZYYXX(4), YXZZXYI(5), ZXYIZXY(3)\}$, $\{IZZYYXX(4), YYIXZZX(6), ZXYIZXY(3)\}$, $\{YYIXZZX(6), YXZZXYI(5), ZXYIZXY(3)\}$, $\{ZYXYXIZ(7), YXZZXYI(5), ZXYIZXY(3)\}$, $\{YYIXZZX(6), ZYXYXIZ(7), ZXYIZXY(3)\}$, $\{IZZYYXX(4), ZYXYXIZ(7), YXZZXYI(5)\}$, $\{IZZYYXX(4), YYIXZZX(6), ZYXYXIZ(7)\}$, $\{ZYXYXIZ(7), YYIXZZX(6), YXZZXYI(5)\}$	$\{IZZYYXX, XIXZYZY, XZYXIYZ, YXZZXYI, YYIXZZX, ZXYIZXY, ZYXYXIZ, IIIIIII\}$
$\{YXZXZIY(3), XIXZYZY(1), XZYXIYZ(2)\}$, $\{ZXYYXZI(5), XIXZYZY(1), XZYXIYZ(2)\}$, $\{ZYXIZYX(6), XIXZYZY(1), XZYXIYZ(2)\}$, $\{XIXZYZY(1), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{IZZYYXX(4), XIXZYZY(1), YXZXZIY(3)\}$, $\{ZYXIZYX(6), XIXZYZY(1), YXZXZIY(3)\}$, $\{XIXZYZY(1), YYIZXXZ(7), YXZXZIY(3)\}$, $\{IZZYYXX(4), ZXYYXZI(5), XIXZYZY(1)\}$, $\{IZZYYXX(4), XIXZYZY(1), ZYXIZYX(6)\}$, $\{IZZYYXX(4), XIXZYZY(1), YYIZXXZ(7)\}$, $\{ZXYYXZI(5), XIXZYZY(1), ZYXIZYX(6)\}$, $\{ZXYYXZI(5), XIXZYZY(1), YYIZXXZ(7)\}$, $\{IZZYYXX(4), YXZXZIY(3), XZYXIYZ(2)\}$, $\{ZXYYXZI(5), YXZXZIY(3), XZYXIYZ(2)\}$, $\{YXZXZIY(3), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{IZZYYXX(4), ZXYYXZI(5), XZYXIYZ(2)\}$, $\{IZZYYXX(4), ZYXIZYX(6), XZYXIYZ(2)\}$, $\{IZZYYXX(4), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{ZXYYXZI(5), ZYXIZYX(6), XZYXIYZ(2)\}$, $\{ZYXIZYX(6), YYIZXXZ(7), XZYXIYZ(2)\}$, $\{IZZYYXX(4), ZXYYXZI(5), YXZXZIY(3)\}$, $\{IZZYYXX(4), ZYXIZYX(6), YXZXZIY(3)\}$, $\{ZXYYXZI(5), ZYXIZYX(6), YXZXZIY(3)\}$, $\{ZXYYXZI(5), YYIZXXZ(7), YXZXZIY(3)\}$, $\{ZYXIZYX(6), YYIZXXZ(7), YXZXZIY(3)\}$, $\{IZZYYXX(4), ZXYYXZI(5), YYIZXXZ(7)\}$, $\{IZZYYXX(4), YYIZXXZ(7), ZYXIZYX(6)\}$, $\{ZXYYXZI(5), YYIZXXZ(7), ZYXIZYX(6)\}$	$\{IZZYYXX, XIXZYZY, XZYXIYZ, YXZXZIY, YYIZXXZ, ZXYYXZI, ZYXIZYX, IIIIIII\}$
$\{XYZIXYZ(2), XIXZYZY(1), ZXYXYIZ(3)\}$, $\{YXZYIZX(5), XYZIXYZ(2), XIXZYZY(1)\}$, $\{YZXXZYI(6), XYZIXYZ(2), XIXZYZY(1)\}$, $\{XYZIXYZ(2), XIXZYZY(1), ZZIYXXY(7)\}$, $\{IYYZZXX(4), XIXZYZY(1), ZXYXYIZ(3)\}$, $\{YZXXZYI(6), XIXZYZY(1), ZXYXYIZ(3)\}$, $\{ZZIYXXY(7), XIXZYZY(1), ZXYXYIZ(3)\}$, $\{YXZYIZX(5), IYYZZXX(4), XIXZYZY(1)\}$, $\{YZXXZYI(6), IYYZZXX(4), XIXZYZY(1)\}$, $\{IYYZZXX(4), XIXZYZY(1), ZZIYXXY(7)\}$, $\{YZXXZYI(6), YXZYIZX(5), XIXZYZY(1)\}$, $\{YXZYIZX(5), ZZIYXXY(7), XIXZYZY(1)\}$, $\{XYZIXYZ(2), IYYZZXX(4), ZXYXYIZ(3)\}$, $\{YXZYIZX(5), XYZIXYZ(2), ZXYXYIZ(3)\}$, $\{XYZIXYZ(2), ZXYXYIZ(3), ZZIYXXY(7)\}$, $\{YXZYIZX(5), XYZIXYZ(2), IYYZZXX(4)\}$, $\{YZXXZYI(6), XYZIXYZ(2), IYYZZXX(4)\}$, $\{XYZIXYZ(2), IYYZZXX(4), ZZIYXXY(7)\}$, $\{YXZYIZX(5), XYZIXYZ(2), YZXXZYI(6)\}$, $\{YZXXZYI(6), XYZIXYZ(2), ZZIYXXY(7)\}$, $\{YXZYIZX(5), IYYZZXX(4), ZXYXYIZ(3)\}$, $\{YZXXZYI(6), IYYZZXX(4), ZXYXYIZ(3)\}$, $\{YXZYIZX(5), YZXXZYI(6), ZXYXYIZ(3)\}$, $\{YXZYIZX(5), ZZIYXXY(7), ZXYXYIZ(3)\}$, $\{YZXXZYI(6), ZZIYXXY(7), ZXYXYIZ(3)\}$, $\{YXZYIZX(5), ZZIYXXY(7), IYYZZXX(4)\}$, $\{YZXXZYI(6), ZZIYXXY(7), IYYZZXX(4)\}$, $\{YXZYIZX(5), ZZIYXXY(7), YZXXZYI(6)\}$	$\{IYYZZXX, XIXZYZY, XYZIXYZ, YXZYIZX, YZXXZYI, ZXYXYIZ, ZZIYXXY, IIIIIII\}$

Table G.5: Continued from Tables G.1-G.4 and in Tables G.6-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{XYZIXYZ(2), XIXZYZY(1), YXZXZIY(3)\}$, $\{ZXYYXZI(5), XYZIXYZ(2), XIXZYZY(1)\}$, $\{ZZIXYYX(6), XYZIXYZ(2), XIXZYZY(1)\}$, $\{XYZIXYZ(2), XIXZYZY(1), YZXYIXZ(7)\}$, $\{IYYZZXX(4), XIXZYZY(1), YXZXZIY(3)\}$, $\{ZZIXYYX(6), XIXZYZY(1), YXZXZIY(3)\}$, $\{XIXZYZY(1), YZXYIXZ(7), YXZXZIY(3)\}$, $\{ZXYYXZI(5), IYYZZXX(4), XIXZYZY(1)\}$, $\{ZZIXYYX(6), IYYZZXX(4), XIXZYZY(1)\}$, $\{IYYZZXX(4), XIXZYZY(1), YZXYIXZ(7)\}$, $\{ZXYYXZI(5), ZZIXYYX(6), XIXZYZY(1)\}$, $\{ZXYYXZI(5), XIXZYZY(1), YZXYIXZ(7)\}$, $\{XYZIXYZ(2), IYYZZXX(4), YXZXZIY(3)\}$, $\{ZXYYXZI(5), XYZIXYZ(2), YXZXZIY(3)\}$, $\{XYZIXYZ(2), YZXYIXZ(7), YXZXZIY(3)\}$, $\{ZXYYXZI(5), XYZIXYZ(2), IYYZZXX(4)\}$, $\{ZZIXYYX(6), XYZIXYZ(2), IYYZZXX(4)\}$, $\{XYZIXYZ(2), IYYZZXX(4), YZXYIXZ(7)\}$, $\{ZXYYXZI(5), ZZIXYYX(6), XYZIXYZ(2)\}$, $\{ZZIXYYX(6), XYZIXYZ(2), YZXYIXZ(7)\}$, $\{ZXYYXZI(5), IYYZZXX(4), YXZXZIY(3)\}$, $\{ZZIXYYX(6), IYYZZXX(4), YXZXZIY(3)\}$, $\{ZXYYXZI(5), ZZIXYYX(6), YXZXZIY(3)\}$, $\{ZXYYXZI(5), YZXYIXZ(7), YXZXZIY(3)\}$, $\{ZZIXYYX(6), YZXYIXZ(7), YXZXZIY(3)\}$, $\{ZXYYXZI(5), IYYZZXX(4), YZXYIXZ(7)\}$, $\{ZZIXYYX(6), IYYZZXX(4), YZXYIXZ(7)\}$, $\{ZXYYXZI(5), ZZIXYYX(6), YZXYIXZ(7)\}$	$\{IYYZZXX, XIXZYZY, XYZIXYZ, YXZXZIY, YZXYIXZ, ZXYYXZI, ZZIXYYX, IIIIIII\}$
$\{IXXYYZZ(1), ZIZXYXY(3), XZYIXZY(2)\}$, $\{IXXYYZZ(1), ZXYZIYX(5), XZYIXZY(2)\}$, $\{IXXYYZZ(1), YZXXZYI(6), XZYIXZY(2)\}$, $\{IXXYYZZ(1), XZYIXZY(2), YYIZXXZ(7)\}$, $\{IXXYYZZ(1), ZIZXYXY(3), XYZYZIX(4)\}$, $\{IXXYYZZ(1), YZXXZYI(6), ZIZXYXY(3)\}$, $\{IXXYYZZ(1), ZIZXYXY(3), YYIZXXZ(7)\}$, $\{IXXYYZZ(1), ZXYZIYX(5), XYZYZIX(4)\}$, $\{IXXYYZZ(1), YZXXZYI(6), XYZYZIX(4)\}$, $\{IXXYYZZ(1), YYIZXXZ(7), XYZYZIX(4)\}$, $\{IXXYYZZ(1), YZXXZYI(6), ZXYZIYX(5)\}$, $\{IXXYYZZ(1), ZXYZIYX(5), YYIZXXZ(7)\}$, $\{ZIZXYXY(3), XZYIXZY(2), XYZYZIX(4)\}$, $\{ZXYZIYX(5), ZIZXYXY(3), XZYIXZY(2)\}$, $\{ZIZXYXY(3), XZYIXZY(2), YYIZXXZ(7)\}$, $\{ZXYZIYX(5), XZYIXZY(2), XYZYZIX(4)\}$, $\{YZXXZYI(6), XZYIXZY(2), XYZYZIX(4)\}$, $\{XZYIXZY(2), YYIZXXZ(7), XYZYZIX(4)\}$, $\{YZXXZYI(6), ZXYZIYX(5), XZYIXZY(2)\}$, $\{YZXXZYI(6), XZYIXZY(2), YYIZXXZ(7)\}$, $\{ZIZXYXY(3), ZXYZIYX(5), XYZYZIX(4)\}$, $\{YZXXZYI(6), ZIZXYXY(3), ZXYZIYX(5)\}$, $\{ZIZXYXY(3), ZXYZIYX(5), YYIZXXZ(7)\}$, $\{YZXXZYI(6), ZIZXYXY(3), YYIZXXZ(7)\}$, $\{ZXYZIYX(5), YYIZXXZ(7), XYZYZIX(4)\}$, $\{YZXXZYI(6), YYIZXXZ(7), XYZYZIX(4)\}$, $\{YZXXZYI(6), ZXYZIYX(5), YYIZXXZ(7)\}$	$\{IXXYYZZ, XYZYZIX, XZYIXZY, YYIZXXZ, YZXXZYI, ZIZXYXY, ZXYZIYX, IIIIIII\}$
$\{IXXYYZZ(1), XZYIXZY(2), YIYXZXZ(3)\}$, $\{YXZZXYI(5), IXXYYZZ(1), XZYIXZY(2)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), XZYIXZY(2)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), XZYIXZY(2)\}$, $\{IXXYYZZ(1), XYZYZIX(4), YIYXZXZ(3)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), YIYXZXZ(3)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), YIYXZXZ(3)\}$, $\{IXXYYZZ(1), YXZZXYI(5), XYZYZIX(4)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), XYZYZIX(4)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), XYZYZIX(4)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), YXZZXYI(5)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), YXZZXYI(5)\}$, $\{YIYXZXZ(3), XZYIXZY(2), XYZYZIX(4)\}$, $\{YXZZXYI(5), YIYXZXZ(3), XZYIXZY(2)\}$, $\{YIYXZXZ(3), ZYXZIXY(7), XZYIXZY(2)\}$, $\{YXZZXYI(5), XZYIXZY(2), XYZYZIX(4)\}$, $\{ZZIXYYX(6), XZYIXZY(2), XYZYZIX(4)\}$, $\{ZYXZIXY(7), XZYIXZY(2), XYZYZIX(4)\}$, $\{YXZZXYI(5), ZZIXYYX(6), XZYIXZY(2)\}$, $\{ZYXZIXY(7), ZZIXYYX(6), XZYIXZY(2)\}$, $\{YIYXZXZ(3), YXZZXYI(5), XYZYZIX(4)\}$, $\{YIYXZXZ(3), ZZIXYYX(6), XYZYZIX(4)\}$, $\{YIYXZXZ(3), ZZIXYYX(6), YXZZXYI(5)\}$, $\{YIYXZXZ(3), ZYXZIXY(7), YXZZXYI(5)\}$, $\{YIYXZXZ(3), ZZIXYYX(6), ZYXZIXY(7)\}$, $\{ZYXZIXY(7), YXZZXYI(5), XYZYZIX(4)\}$, $\{ZYXZIXY(7), ZZIXYYX(6), XYZYZIX(4)\}$, $\{ZYXZIXY(7), ZZIXYYX(6), YXZZXYI(5)\}$	$\{IXXYYZZ, XYZYZIX, XZYIXZY, YIYXZXZ, YXZZXYI, ZYXZIXY, ZZIXYYX, IIIIIII\}$

Table G.6: Continued from Tables G.1-G.5 and in Tables G.7-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IXXYYZZ(1), ZXYIZXY(3), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), ZIZYXYX(5), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), YYIXZZX(6), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), YZXZXIY(7), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), XYZZYXI(4), ZXYIZXY(3)\}$, $\{IXXYYZZ(1), YYIXZZX(6), ZXYIZXY(3)\}$, $\{IXXYYZZ(1), YZXZXIY(7), ZXYIZXY(3)\}$, $\{IXXYYZZ(1), ZIZYXYX(5), XYZZYXI(4)\}$, $\{IXXYYZZ(1), YYIXZZX(6), XYZZYXI(4)\}$, $\{IXXYYZZ(1), YZXZXIY(7), XYZZYXI(4)\}$, $\{IXXYYZZ(1), ZIZYXYX(5), YYIXZZX(6)\}$, $\{IXXYYZZ(1), ZIZYXYX(5), YZXZXIY(7)\}$, $\{ZXYIZXY(3), XYZZYXI(4), XZYXIYZ(2)\}$, $\{ZXYIZXY(3), ZIZYXYX(5), XZYXIYZ(2)\}$, $\{ZXYIZXY(3), YZXZXIY(7), XZYXIYZ(2)\}$, $\{XYZZYXI(4), ZIZYXYX(5), XZYXIYZ(2)\}$, $\{XYZZYXI(4), YYIXZZX(6), XZYXIYZ(2)\}$, $\{XYZZYXI(4), YZXZXIY(7), XZYXIYZ(2)\}$, $\{ZIZYXYX(5), YYIXZZX(6), XZYXIYZ(2)\}$, $\{YYIXZZX(6), YZXZXIY(7), XZYXIYZ(2)\}$, $\{XYZZYXI(4), ZIZYXYX(5), ZXYIZXY(3)\}$, $\{XYZZYXI(4), YYIXZZX(6), ZXYIZXY(3)\}$, $\{ZIZYXYX(5), YYIXZZX(6), ZXYIZXY(3)\}$, $\{ZIZYXYX(5), YZXZXIY(7), ZXYIZXY(3)\}$, $\{YYIXZZX(6), YZXZXIY(7), ZXYIZXY(3)\}$, $\{ZIZYXYX(5), YZXZXIY(7), XYZZYXI(4)\}$, $\{YYIXZZX(6), YZXZXIY(7), XYZZYXI(4)\}$, $\{ZIZYXYX(5), YYIXZZX(6), YZXZXIY(7)\}$	$\{IXXYYZZ, XYZZYXI, XZYXIYZ, YYIXZZX, YZXZXIY, ZIZYXYX, ZXYIZXY, IIIIIII\}$
$\{IXXYYZZ(1), YXZXZIY(3), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), YIYZXZX(5), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), ZYXIZYX(6), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), ZZIYXXY(7), XZYXIYZ(2)\}$, $\{IXXYYZZ(1), XYZZYXI(4), YXZXZIY(3)\}$, $\{IXXYYZZ(1), ZYXIZYX(6), YXZXZIY(3)\}$, $\{IXXYYZZ(1), ZZIYXXY(7), YXZXZIY(3)\}$, $\{IXXYYZZ(1), YIYZXZX(5), XYZZYXI(4)\}$, $\{IXXYYZZ(1), ZYXIZYX(6), XYZZYXI(4)\}$, $\{IXXYYZZ(1), ZZIYXXY(7), XYZZYXI(4)\}$, $\{IXXYYZZ(1), YIYZXZX(5), ZYXIZYX(6)\}$, $\{IXXYYZZ(1), YIYZXZX(5), ZZIYXXY(7)\}$, $\{XYZZYXI(4), YXZXZIY(3), XZYXIYZ(2)\}$, $\{YXZXZIY(3), YIYZXZX(5), XZYXIYZ(2)\}$, $\{YXZXZIY(3), ZZIYXXY(7), XZYXIYZ(2)\}$, $\{XYZZYXI(4), YIYZXZX(5), XZYXIYZ(2)\}$, $\{XYZZYXI(4), ZYXIZYX(6), XZYXIYZ(2)\}$, $\{XYZZYXI(4), ZZIYXXY(7), XZYXIYZ(2)\}$, $\{YIYZXZX(5), ZYXIZYX(6), XZYXIYZ(2)\}$, $\{ZYXIZYX(6), ZZIYXXY(7), XZYXIYZ(2)\}$, $\{XYZZYXI(4), YIYZXZX(5), YXZXZIY(3)\}$, $\{XYZZYXI(4), ZYXIZYX(6), YXZXZIY(3)\}$, $\{YIYZXZX(5), ZYXIZYX(6), YXZXZIY(3)\}$, $\{YIYZXZX(5), ZZIYXXY(7), YXZXZIY(3)\}$, $\{ZYXIZYX(6), ZZIYXXY(7), YXZXZIY(3)\}$, $\{YIYZXZX(5), ZZIYXXY(7), XYZZYXI(4)\}$, $\{ZYXIZYX(6), ZZIYXXY(7), XYZZYXI(4)\}$, $\{YIYZXZX(5), ZZIYXXY(7), ZYXIZYX(6)\}$	$\{IXXYYZZ, XYZZYXI, XZYXIYZ, YIYZXZX, YXZXZIY, ZYXIZYX, ZZIYXXY, IIIIIII\}$
$\{IXXYYZZ(1), XYZIXYZ(2), ZIZXYXY(3)\}$, $\{IXXYYZZ(1), XYZIXYZ(2), ZXYZIYX(5)\}$, $\{IXXYYZZ(1), XYZIXYZ(2), YYIXZZX(6)\}$, $\{IXXYYZZ(1), XYZIXYZ(2), YZXZXIY(7)\}$, $\{IXXYYZZ(1), ZIZXYXY(3), XZYYZXI(4)\}$, $\{IXXYYZZ(1), ZIZXYXY(3), YYIXZZX(6)\}$, $\{IXXYYZZ(1), ZIZXYXY(3), YZXZXIY(7)\}$, $\{IXXYYZZ(1), ZXYZIYX(5), XZYYZXI(4)\}$, $\{IXXYYZZ(1), YYIXZZX(6), XZYYZXI(4)\}$, $\{IXXYYZZ(1), YZXZXIY(7), XZYYZXI(4)\}$, $\{IXXYYZZ(1), YYIXZZX(6), ZXYZIYX(5)\}$, $\{IXXYYZZ(1), YZXZXIY(7), ZXYZIYX(5)\}$, $\{XYZIXYZ(2), ZIZXYXY(3), XZYYZXI(4)\}$, $\{XYZIXYZ(2), ZIZXYXY(3), ZXYZIYX(5)\}$, $\{XYZIXYZ(2), ZIZXYXY(3), YZXZXIY(7)\}$, $\{ZXYZIYX(5), XYZIXYZ(2), XZYYZXI(4)\}$, $\{XYZIXYZ(2), YYIXZZX(6), XZYYZXI(4)\}$, $\{YZXZXIY(7), XYZIXYZ(2), XZYYZXI(4)\}$, $\{XYZIXYZ(2), YYIXZZX(6), ZXYZIYX(5)\}$, $\{XYZIXYZ(2), YYIXZZX(6), YZXZXIY(7)\}$, $\{ZXYZIYX(5), ZIZXYXY(3), XZYYZXI(4)\}$, $\{YYIXZZX(6), ZIZXYXY(3), XZYYZXI(4)\}$, $\{YYIXZZX(6), ZIZXYXY(3), ZXYZIYX(5)\}$, $\{YZXZXIY(7), ZIZXYXY(3), ZXYZIYX(5)\}$, $\{YYIXZZX(6), ZIZXYXY(3), YZXZXIY(7)\}$, $\{YZXZXIY(7), ZXYZIYX(5), XZYYZXI(4)\}$, $\{YZXZXIY(7), YYIXZZX(6), XZYYZXI(4)\}$, $\{YZXZXIY(7), YYIXZZX(6), ZXYZIYX(5)\}$	$\{IXXYYZZ, XYZIXYZ, XZYYZXI, YYIXZZX, YZXZXIY, ZIZXYXY, ZXYZIYX, IIIIIII\}$

Table G.7: Continued from Tables G.1-G.6 and in Tables G.8-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IXXYYZZ(1), XYZIXYZ(2), YXZXZIY(3)\}$, $\{YIYZXZX(5), IXXYYZZ(1), XYZIXYZ(2)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), XYZIXYZ(2)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), XYZIXYZ(2)\}$, $\{IXXYYZZ(1), XZYYZXI(4), YXZXZIY(3)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), YXZXZIY(3)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), YXZXZIY(3)\}$, $\{IXXYYZZ(1), YIYZXZX(5), XZYYZXI(4)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), XZYYZXI(4)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), XZYYZXI(4)\}$, $\{IXXYYZZ(1), ZZIXYYX(6), YIYZXZX(5)\}$, $\{IXXYYZZ(1), ZYXZIXY(7), YIYZXZX(5)\}$, $\{XYZIXYZ(2), XZYYZXI(4), YXZXZIY(3)\}$, $\{YIYZXZX(5), XYZIXYZ(2), YXZXZIY(3)\}$, $\{ZYXZIXY(7), XYZIXYZ(2), YXZXZIY(3)\}$, $\{YIYZXZX(5), XYZIXYZ(2), XZYYZXI(4)\}$, $\{ZZIXYYX(6), XYZIXYZ(2), XZYYZXI(4)\}$, $\{ZYXZIXY(7), XYZIXYZ(2), XZYYZXI(4)\}$, $\{YIYZXZX(5), ZZIXYYX(6), XYZIXYZ(2)\}$, $\{ZYXZIXY(7), ZZIXYYX(6), XYZIXYZ(2)\}$, $\{YIYZXZX(5), XZYYZXI(4), YXZXZIY(3)\}$, $\{ZZIXYYX(6), XZYYZXI(4), YXZXZIY(3)\}$, $\{ZZIXYYX(6), YIYZXZX(5), YXZXZIY(3)\}$, $\{ZYXZIXY(7), YIYZXZX(5), YXZXZIY(3)\}$, $\{ZYXZIXY(7), ZZIXYYX(6), YXZXZIY(3)\}$, $\{ZYXZIXY(7), YIYZXZX(5), XZYYZXI(4)\}$, $\{ZYXZIXY(7), ZZIXYYX(6), XZYYZXI(4)\}$, $\{ZYXZIXY(7), YYIZXXZ(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), IXXYYZZ(1), ZXYIZXY(3)\}$, $\{IXXYYZZ(1), YZXXZYI(6), ZXYIZXY(3)\}$, $\{IXXYYZZ(1), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{XZYZYIX(4), IXXYYZZ(1), ZIZYXYX(5)\}$, $\{XZYZYIX(4), IXXYYZZ(1), YZXXZYI(6)\}$, $\{XZYZYIX(4), IXXYYZZ(1), YYIZXXZ(7)\}$, $\{IXXYYZZ(1), YZXXZYI(6), ZIZYXYX(5)\}$, $\{IXXYYZZ(1), ZIZYXYX(5), YYIZXXZ(7)\}$, $\{XZYZYIX(4), ZXYIZXY(3), XYZXIZY(2)\}$, $\{ZXYIZXY(3), ZIZYXYX(5), XYZXIZY(2)\}$, $\{ZXYIZXY(3), YYIZXXZ(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), ZIZYXYX(5), XYZXIZY(2)\}$, $\{XZYZYIX(4), YZXXZYI(6), XYZXIZY(2)\}$, $\{XZYZYIX(4), YYIZXXZ(7), XYZXIZY(2)\}$, $\{YZXXZYI(6), ZIZYXYX(5), XYZXIZY(2)\}$, $\{YZXXZYI(6), YYIZXXZ(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), ZIZYXYX(5), ZXYIZXY(3)\}$, $\{XZYZYIX(4), YZXXZYI(6), ZXYIZXY(3)\}$, $\{YZXXZYI(6), ZIZYXYX(5), ZXYIZXY(3)\}$, $\{ZIZYXYX(5), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{YZXXZYI(6), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{XZYZYIX(4), ZIZYXYX(5), YYIZXXZ(7)\}$, $\{XZYZYIX(4), YZXXZYI(6), YYIZXXZ(7)\}$, $\{YZXXZYI(6), ZIZYXYX(5), YYIZXXZ(7)\}$	$\{IXXYYZZ, XYZXIZY, XZYZYIX, YYIZXXZ, YZXXZYI, ZIZYXYX, ZXYIZXY, IIIIIII\}$
$\{IXXYYZZ(1), ZXYIZXY(3), XYZXIZY(2)\}$, $\{IXXYYZZ(1), ZIZYXYX(5), XYZXIZY(2)\}$, $\{IXXYYZZ(1), YZXXZYI(6), XYZXIZY(2)\}$, $\{IXXYYZZ(1), YYIZXXZ(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), IXXYYZZ(1), ZXYIZXY(3)\}$, $\{IXXYYZZ(1), YZXXZYI(6), ZXYIZXY(3)\}$, $\{IXXYYZZ(1), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{XZYZYIX(4), IXXYYZZ(1), ZIZYXYX(5)\}$, $\{XZYZYIX(4), IXXYYZZ(1), YZXXZYI(6)\}$, $\{XZYZYIX(4), IXXYYZZ(1), YYIZXXZ(7)\}$, $\{IXXYYZZ(1), YZXXZYI(6), ZIZYXYX(5)\}$, $\{IXXYYZZ(1), ZIZYXYX(5), YYIZXXZ(7)\}$, $\{XZYZYIX(4), ZXYIZXY(3), XYZXIZY(2)\}$, $\{ZXYIZXY(3), ZIZYXYX(5), XYZXIZY(2)\}$, $\{ZXYIZXY(3), YYIZXXZ(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), ZIZYXYX(5), XYZXIZY(2)\}$, $\{XZYZYIX(4), YZXXZYI(6), XYZXIZY(2)\}$, $\{XZYZYIX(4), YYIZXXZ(7), XYZXIZY(2)\}$, $\{YZXXZYI(6), ZIZYXYX(5), XYZXIZY(2)\}$, $\{YZXXZYI(6), YYIZXXZ(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), ZIZYXYX(5), ZXYIZXY(3)\}$, $\{XZYZYIX(4), YZXXZYI(6), ZXYIZXY(3)\}$, $\{YZXXZYI(6), ZIZYXYX(5), ZXYIZXY(3)\}$, $\{ZIZYXYX(5), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{YZXXZYI(6), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{XZYZYIX(4), ZIZYXYX(5), YYIZXXZ(7)\}$, $\{XZYZYIX(4), YZXXZYI(6), YYIZXXZ(7)\}$, $\{YZXXZYI(6), ZIZYXYX(5), YYIZXXZ(7)\}$	$\{IXXYYZZ, XYZXIZY, XZYZYIX, YYIZXXZ, YZXXZYI, ZIZYXYX, ZXYIZXY, IIIIIII\}$
$\{IXXYYZZ(1), YIYXZXZ(3), XYZXIZY(2)\}$, $\{IXXYYZZ(1), YXZZXYI(5), XYZXIZY(2)\}$, $\{IXXYYZZ(1), ZYXIZYX(6), XYZXIZY(2)\}$, $\{IXXYYZZ(1), ZZIYXXY(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), IXXYYZZ(1), YIYXZXZ(3)\}$, $\{IXXYYZZ(1), ZYXIZYX(6), YIYXZXZ(3)\}$, $\{IXXYYZZ(1), ZZIYXXY(7), YIYXZXZ(3)\}$, $\{XZYZYIX(4), IXXYYZZ(1), YXZZXYI(5)\}$, $\{XZYZYIX(4), IXXYYZZ(1), ZYXIZYX(6)\}$, $\{XZYZYIX(4), IXXYYZZ(1), ZZIYXXY(7)\}$, $\{IXXYYZZ(1), YXZZXYI(5), ZYXIZYX(6)\}$, $\{IXXYYZZ(1), ZZIYXXY(7), YXZZXYI(5)\}$, $\{XZYZYIX(4), YIYXZXZ(3), XYZXIZY(2)\}$, $\{YIYXZXZ(3), YXZZXYI(5), XYZXIZY(2)\}$, $\{YIYXZXZ(3), ZZIYXXY(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), YXZZXYI(5), XYZXIZY(2)\}$, $\{XZYZYIX(4), ZYXIZYX(6), XYZXIZY(2)\}$, $\{XZYZYIX(4), ZZIYXXY(7), XYZXIZY(2)\}$, $\{ZYXIZYX(6), YXZZXYI(5), XYZXIZY(2)\}$, $\{ZYXIZYX(6), ZZIYXXY(7), XYZXIZY(2)\}$, $\{XZYZYIX(4), YIYXZXZ(3), YXZZXYI(5)\}$, $\{XZYZYIX(4), YIYXZXZ(3), ZYXIZYX(6)\}$, $\{YIYXZXZ(3), YXZZXYI(5), ZYXIZYX(6)\}$, $\{YIYXZXZ(3), ZZIYXXY(7), YXZZXYI(5)\}$, $\{YIYXZXZ(3), ZZIYXXY(7), ZYXIZYX(6)\}$, $\{XZYZYIX(4), ZZIYXXY(7), YXZZXYI(5)\}$, $\{XZYZYIX(4), ZZIYXXY(7), ZYXIZYX(6)\}$, $\{ZZIYXXY(7), YXZZXYI(5), ZYXIZYX(6)\}$	$\{IXXYYZZ, XYZXIZY, XZYZYIX, YIYXZXZ, YXZZXYI, ZYXIZYX, ZZIYXXY, IIIIIII\}$

Table G.8: Continued from Tables G.1-G.7 and in Tables G.9-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IZZXXYY(2), XIXYZYZ(1), ZXYIZXY(3)\}$, $\{IZZXXYY(2), XIXYZYZ(1), YXZYIZX(5)\}$, $\{IZZXXYY(2), ZYXXYZI(6), XIXYZYZ(1)\}$, $\{IZZXXYY(2), XIXYZYZ(1), YYIZXXZ(7)\}$, $\{XZYZYIX(4), XIXYZYZ(1), ZXYIZXY(3)\}$, $\{ZYXXYZI(6), XIXYZYZ(1), ZXYIZXY(3)\}$, $\{XIXYZYZ(1), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{XZYZYIX(4), YXZYIZX(5), XIXYZYZ(1)\}$, $\{XZYZYIX(4), ZYXXYZI(6), XIXYZYZ(1)\}$, $\{XZYZYIX(4), XIXYZYZ(1), YYIZXXZ(7)\}$, $\{YXZYIZX(5), ZYXXYZI(6), XIXYZYZ(1)\}$, $\{YXZYIZX(5), XIXYZYZ(1), YYIZXXZ(7)\}$, $\{XZYZYIX(4), IZZXXYY(2), ZXYIZXY(3)\}$, $\{IZZXXYY(2), YXZYIZX(5), ZXYIZXY(3)\}$, $\{IZZXXYY(2), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{XZYZYIX(4), IZZXXYY(2), YXZYIZX(5)\}$, $\{XZYZYIX(4), IZZXXYY(2), ZYXXYZI(6)\}$, $\{XZYZYIX(4), IZZXXYY(2), YYIZXXZ(7)\}$, $\{IZZXXYY(2), ZYXXYZI(6), YXZYIZX(5)\}$, $\{IZZXXYY(2), ZYXXYZI(6), YYIZXXZ(7)\}$, $\{XZYZYIX(4), YXZYIZX(5), ZXYIZXY(3)\}$, $\{XZYZYIX(4), ZYXXYZI(6), ZXYIZXY(3)\}$, $\{YXZYIZX(5), ZYXXYZI(6), ZXYIZXY(3)\}$, $\{YXZYIZX(5), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{ZYXXYZI(6), YYIZXXZ(7), ZXYIZXY(3)\}$, $\{XZYZYIX(4), YXZYIZX(5), YYIZXXZ(7)\}$, $\{XZYZYIX(4), ZYXXYZI(6), YYIZXXZ(7)\}$, $\{YXZYIZX(5), ZYXXYZI(6), YYIZXXZ(7)\}$	$\{IZZXXYY, XIXYZYZ, XZYZYIX, YXZYIZX, YYIZXXZ, ZXYIZXY, ZYXXYZI, IIIIIII\}$
$\{IZZXXYY(2), YXZIYXZ(3), XIXYZYZ(1)\}$, $\{ZXYYXZI(5), IZZXXYY(2), XIXYZYZ(1)\}$, $\{IZZXXYY(2), YYIXZZX(6), XIXYZYZ(1)\}$, $\{ZYXZIXY(7), IZZXXYY(2), XIXYZYZ(1)\}$, $\{XZYZYIX(4), YXZIYXZ(3), XIXYZYZ(1)\}$, $\{YYIXZZX(6), YXZIYXZ(3), XIXYZYZ(1)\}$, $\{ZYXZIXY(7), YXZIYXZ(3), XIXYZYZ(1)\}$, $\{XZYZYIX(4), ZXYYXZI(5), XIXYZYZ(1)\}$, $\{XZYZYIX(4), YYIXZZX(6), XIXYZYZ(1)\}$, $\{XZYZYIX(4), ZYXZIXY(7), XIXYZYZ(1)\}$, $\{ZXYYXZI(5), YYIXZZX(6), XIXYZYZ(1)\}$, $\{ZXYYXZI(5), ZYXZIXY(7), XIXYZYZ(1)\}$, $\{XZYZYIX(4), IZZXXYY(2), YXZIYXZ(3)\}$, $\{ZXYYXZI(5), IZZXXYY(2), YXZIYXZ(3)\}$, $\{ZYXZIXY(7), IZZXXYY(2), YXZIYXZ(3)\}$, $\{XZYZYIX(4), ZXYYXZI(5), IZZXXYY(2)\}$, $\{XZYZYIX(4), IZZXXYY(2), YYIXZZX(6)\}$, $\{XZYZYIX(4), ZYXZIXY(7), IZZXXYY(2)\}$, $\{ZXYYXZI(5), IZZXXYY(2), YYIXZZX(6)\}$, $\{ZYXZIXY(7), IZZXXYY(2), YYIXZZX(6)\}$, $\{XZYZYIX(4), ZXYYXZI(5), YXZIYXZ(3)\}$, $\{XZYZYIX(4), YYIXZZX(6), YXZIYXZ(3)\}$, $\{ZXYYXZI(5), YXZIYXZ(3), YYIXZZX(6)\}$, $\{ZXYYXZI(5), ZYXZIXY(7), YXZIYXZ(3)\}$, $\{YYIXZZX(6), ZYXZIXY(7), YXZIYXZ(3)\}$, $\{XZYZYIX(4), ZXYYXZI(5), ZYXZIXY(7)\}$, $\{XZYZYIX(4), ZYXZIXY(7), YYIXZZX(6)\}$, $\{ZXYYXZI(5), ZYXZIXY(7), YYIXZZX(6)\}$	$\{IZZXXYY, XIXYZYZ, XZYZYIX, YXZIYXZ, YYIXZZX, ZXYYXZI, ZYXZIXY, IIIIIII\}$
$\{ZXYXYIZ(3), XIXYZYZ(1), XZYIXZY(2)\}$, $\{YXZZXYI(5), XIXYZYZ(1), XZYIXZY(2)\}$, $\{YYIXZZX(6), XIXYZYZ(1), XZYIXZY(2)\}$, $\{ZYXZIXY(7), XIXYZYZ(1), XZYIXZY(2)\}$, $\{IZZYYXX(4), XIXYZYZ(1), ZXYXYIZ(3)\}$, $\{YYIXZZX(6), XIXYZYZ(1), ZXYXYIZ(3)\}$, $\{ZYXZIXY(7), XIXYZYZ(1), ZXYXYIZ(3)\}$, $\{IZZYYXX(4), YXZZXYI(5), XIXYZYZ(1)\}$, $\{IZZYYXX(4), YYIXZZX(6), XIXYZYZ(1)\}$, $\{IZZYYXX(4), ZYXZIXY(7), XIXYZYZ(1)\}$, $\{YXZZXYI(5), YYIXZZX(6), XIXYZYZ(1)\}$, $\{YXZZXYI(5), ZYXZIXY(7), XIXYZYZ(1)\}$, $\{IZZYYXX(4), XZYIXZY(2), ZXYXYIZ(3)\}$, $\{YXZZXYI(5), XZYIXZY(2), ZXYXYIZ(3)\}$, $\{ZYXZIXY(7), XZYIXZY(2), ZXYXYIZ(3)\}$, $\{IZZYYXX(4), YXZZXYI(5), XZYIXZY(2)\}$, $\{IZZYYXX(4), YYIXZZX(6), XZYIXZY(2)\}$, $\{IZZYYXX(4), ZYXZIXY(7), XZYIXZY(2)\}$, $\{YXZZXYI(5), YYIXZZX(6), XZYIXZY(2)\}$, $\{ZYXZIXY(7), YYIXZZX(6), XZYIXZY(2)\}$, $\{IZZYYXX(4), YXZZXYI(5), ZXYXYIZ(3)\}$, $\{IZZYYXX(4), YYIXZZX(6), ZXYXYIZ(3)\}$, $\{YYIXZZX(6), YXZZXYI(5), ZXYXYIZ(3)\}$, $\{ZYXZIXY(7), YXZZXYI(5), ZXYXYIZ(3)\}$, $\{ZYXZIXY(7), YYIXZZX(6), ZXYXYIZ(3)\}$, $\{IZZYYXX(4), ZYXZIXY(7), YXZZXYI(5)\}$, $\{IZZYYXX(4), ZYXZIXY(7), YYIXZZX(6)\}$, $\{ZYXZIXY(7), YYIXZZX(6), YXZZXYI(5)\}$	$\{IZZYYXX, XIXYZYZ, XZYIXZY, YXZZXYI, YYIXZZX, ZXYXYIZ, ZYXZIXY, IIIIIII\}$

Table G.9: Continued from Tables G.1-G.8 and in Tables G.10-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{XIXYZYZ(1), XZYIXZY(2), YXZXZIY(3)\}$, $\{ZXYZIYX(5), XIXYZYZ(1), XZYIXZY(2)\}$, $\{ZYXXYZI(6), XIXYZYZ(1), XZYIXZY(2)\}$, $\{YYIZXXZ(7), XIXYZYZ(1), XZYIXZY(2)\}$, $\{IZZYYXX(4), XIXYZYZ(1), YXZXZIY(3)\}$, $\{ZYXXYZI(6), XIXYZYZ(1), YXZXZIY(3)\}$, $\{XIXYZYZ(1), YYIZXXZ(7), YXZXZIY(3)\}$, $\{IZZYYXX(4), ZXYZIYX(5), XIXYZYZ(1)\}$, $\{IZZYYXX(4), ZYXXYZI(6), XIXYZYZ(1)\}$, $\{IZZYYXX(4), XIXYZYZ(1), YYIZXXZ(7)\}$, $\{ZXYZIYX(5), ZYXXYZI(6), XIXYZYZ(1)\}$, $\{ZXYZIYX(5), XIXYZYZ(1), YYIZXXZ(7)\}$, $\{IZZYYXX(4), XZYIXZY(2), YXZXZIY(3)\}$, $\{ZXYZIYX(5), XZYIXZY(2), YXZXZIY(3)\}$, $\{XZYIXZY(2), YYIZXXZ(7), YXZXZIY(3)\}$, $\{IZZYYXX(4), ZXYZIYX(5), XZYIXZY(2)\}$, $\{IZZYYXX(4), ZYXXYZI(6), XZYIXZY(2)\}$, $\{IZZYYXX(4), XZYIXZY(2), YYIZXXZ(7)\}$, $\{ZXYZIYX(5), ZYXXYZI(6), XZYIXZY(2)\}$, $\{ZYXXYZI(6), XZYIXZY(2), YYIZXXZ(7)\}$, $\{IZZYYXX(4), ZXYZIYX(5), YXZXZIY(3)\}$, $\{IZZYYXX(4), ZYXXYZI(6), YXZXZIY(3)\}$, $\{ZYXXYZI(6), ZXYZIYX(5), YXZXZIY(3)\}$, $\{ZXYZIYX(5), YYIZXXZ(7), YXZXZIY(3)\}$, $\{ZYXXYZI(6), YYIZXXZ(7), YXZXZIY(3)\}$, $\{IZZYYXX(4), ZXYZIYX(5), YYIZXXZ(7)\}$, $\{IZZYYXX(4), ZYXXYZI(6), YYIZXXZ(7)\}$, $\{ZYXXYZI(6), ZXYZIYX(5), YYIZXXZ(7)\}$	$\{IZZYYXX, XIXYZYZ, XZYIXZY, YXZXZIY, YYIZXXZ, ZXYZIYX, ZYXXYZI, IIIIIII\}$
$\{IYYXXZZ(2), XIXYZYZ(1), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), XIXYZYZ(1), YXZYIZX(5)\}$, $\{IYYXXZZ(2), XIXYZYZ(1), ZZIXYYX(6)\}$, $\{YZXZXIY(7), IYYXXZZ(2), XIXYZYZ(1)\}$, $\{XYZZYXI(4), XIXYZYZ(1), ZXYIZXY(3)\}$, $\{ZZIXYYX(6), XIXYZYZ(1), ZXYIZXY(3)\}$, $\{YZXZXIY(7), XIXYZYZ(1), ZXYIZXY(3)\}$, $\{YXZYIZX(5), XIXYZYZ(1), XYZZYXI(4)\}$, $\{ZZIXYYX(6), XIXYZYZ(1), XYZZYXI(4)\}$, $\{YZXZXIY(7), XIXYZYZ(1), XYZZYXI(4)\}$, $\{YXZYIZX(5), XIXYZYZ(1), ZZIXYYX(6)\}$, $\{YZXZXIY(7), YXZYIZX(5), XIXYZYZ(1)\}$, $\{XYZZYXI(4), IYYXXZZ(2), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), YXZYIZX(5), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), YZXZXIY(7), ZXYIZXY(3)\}$, $\{IYYXXZZ(2), YXZYIZX(5), XYZZYXI(4)\}$, $\{IYYXXZZ(2), ZZIXYYX(6), XYZZYXI(4)\}$, $\{IYYXXZZ(2), YZXZXIY(7), XYZZYXI(4)\}$, $\{IYYXXZZ(2), YXZYIZX(5), ZZIXYYX(6)\}$, $\{IYYXXZZ(2), YZXZXIY(7), ZZIXYYX(6)\}$, $\{XYZZYXI(4), YXZYIZX(5), ZXYIZXY(3)\}$, $\{XYZZYXI(4), ZZIXYYX(6), ZXYIZXY(3)\}$, $\{YXZYIZX(5), ZZIXYYX(6), ZXYIZXY(3)\}$, $\{YXZYIZX(5), YZXZXIY(7), ZXYIZXY(3)\}$, $\{ZZIXYYX(6), YZXZXIY(7), ZXYIZXY(3)\}$, $\{YXZYIZX(5), YZXZXIY(7), XYZZYXI(4)\}$, $\{ZZIXYYX(6), YZXZXIY(7), XYZZYXI(4)\}$, $\{YXZYIZX(5), YZXZXIY(7), ZZIXYYX(6)\}$	$\{IYYXXZZ, XIXYZYZ, XYZZYXI, YXZYIZX, YZXZXIY, ZXYIZXY, ZZIXYYX, IIIIIII\}$
$\{IYYXXZZ(2), XIXYZYZ(1), YXZXZIY(3)\}$, $\{IYYXXZZ(2), ZXYZIYX(5), XIXYZYZ(1)\}$, $\{YZXIYZX(6), IYYXXZZ(2), XIXYZYZ(1)\}$, $\{IYYXXZZ(2), ZZIYXXY(7), XIXYZYZ(1)\}$, $\{XYZZYXI(4), XIXYZYZ(1), YXZXZIY(3)\}$, $\{YZXIYZX(6), XIXYZYZ(1), YXZXZIY(3)\}$, $\{ZZIYXXY(7), XIXYZYZ(1), YXZXZIY(3)\}$, $\{ZXYZIYX(5), XIXYZYZ(1), XYZZYXI(4)\}$, $\{YZXIYZX(6), XIXYZYZ(1), XYZZYXI(4)\}$, $\{ZZIYXXY(7), XIXYZYZ(1), XYZZYXI(4)\}$, $\{YZXIYZX(6), ZXYZIYX(5), XIXYZYZ(1)\}$, $\{ZXYZIYX(5), XIXYZYZ(1), ZZIYXXY(7)\}$, $\{XYZZYXI(4), IYYXXZZ(2), YXZXZIY(3)\}$, $\{IYYXXZZ(2), ZXYZIYX(5), YXZXZIY(3)\}$, $\{IYYXXZZ(2), ZZIYXXY(7), YXZXZIY(3)\}$, $\{IYYXXZZ(2), ZXYZIYX(5), XYZZYXI(4)\}$, $\{YZXIYZX(6), IYYXXZZ(2), XYZZYXI(4)\}$, $\{IYYXXZZ(2), ZZIYXXY(7), XYZZYXI(4)\}$, $\{YZXIYZX(6), IYYXXZZ(2), ZXYZIYX(5)\}$, $\{YZXIYZX(6), IYYXXZZ(2), ZZIYXXY(7)\}$, $\{XYZZYXI(4), ZXYZIYX(5), YXZXZIY(3)\}$, $\{YZXIYZX(6), XYZZYXI(4), YXZXZIY(3)\}$, $\{YZXIYZX(6), ZXYZIYX(5), YXZXZIY(3)\}$, $\{ZZIYXXY(7), ZXYZIYX(5), YXZXZIY(3)\}$, $\{YZXIYZX(6), ZZIYXXY(7), YXZXZIY(3)\}$, $\{ZZIYXXY(7), ZXYZIYX(5), XYZZYXI(4)\}$, $\{YZXIYZX(6), ZZIYXXY(7), XYZZYXI(4)\}$, $\{YZXIYZX(6), ZZIYXXY(7), ZXYZIYX(5)\}$	$\{IYYXXZZ, XIXYZYZ, XYZZYXI, YXZXZIY, YZXIYZX, ZXYZIYX, ZZIYXXY, IIIIIII\}$

Table G.10: Continued from Tables G.1-G.9 and in Tables G.11-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{XIXYZYZ(1), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{YXZZXYI(5), XIXYZYZ(1), XYZXIZY(2)\}$, $\{YZXIYZX(6), XIXYZYZ(1), XYZXIZY(2)\}$, $\{ZZIYXXY(7), XIXYZYZ(1), XYZXIZY(2)\}$, $\{IYYZZXX(4), XIXYZYZ(1), ZXYXYIZ(3)\}$, $\{YZXIYZX(6), XIXYZYZ(1), ZXYXYIZ(3)\}$, $\{ZZIYXXY(7), XIXYZYZ(1), ZXYXYIZ(3)\}$, $\{YXZZXYI(5), IYYZZXX(4), XIXYZYZ(1)\}$, $\{YZXIYZX(6), IYYZZXX(4), XIXYZYZ(1)\}$, $\{ZZIYXXY(7), IYYZZXX(4), XIXYZYZ(1)\}$, $\{YXZZXYI(5), YZXIYZX(6), XIXYZYZ(1)\}$, $\{YXZZXYI(5), ZZIYXXY(7), XIXYZYZ(1)\}$, $\{IYYZZXX(4), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{YXZZXYI(5), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{ZZIYXXY(7), ZXYXYIZ(3), XYZXIZY(2)\}$, $\{IYYZZXX(4), YXZZXYI(5), XYZXIZY(2)\}$, $\{YZXIYZX(6), IYYZZXX(4), XYZXIZY(2)\}$, $\{ZZIYXXY(7), IYYZZXX(4), XYZXIZY(2)\}$, $\{YZXIYZX(6), YXZZXYI(5), XYZXIZY(2)\}$, $\{YZXIYZX(6), ZZIYXXY(7), XYZXIZY(2)\}$, $\{IYYZZXX(4), YXZZXYI(5), ZXYXYIZ(3)\}$, $\{YZXIYZX(6), IYYZZXX(4), ZXYXYIZ(3)\}$, $\{YZXIYZX(6), YXZZXYI(5), ZXYXYIZ(3)\}$, $\{ZZIYXXY(7), YXZZXYI(5), ZXYXYIZ(3)\}$, $\{YZXIYZX(6), ZZIYXXY(7), ZXYXYIZ(3)\}$, $\{ZZIYXXY(7), IYYZZXX(4), YXZZXYI(5)\}$, $\{YZXIYZX(6), ZZIYXXY(7), IYYZZXX(4)\}$, $\{YZXIYZX(6), ZZIYXXY(7), YXZZXYI(5)\}$	$\{IYYZZXX, XIXYZYZ, XYZXIZY, YXZZXYI, YZXIYZX, ZXYXYIZ, ZZIYXXY, IIIIIII\}$
$\{XYZXIZY(2), XIXYZYZ(1), YXZIYXZ(3)\}$, $\{XYZXIZY(2), XIXYZYZ(1), ZXYYXZI(5)\}$, $\{XYZXIZY(2), ZZIXYYX(6), XIXYZYZ(1)\}$, $\{XYZXIZY(2), XIXYZYZ(1), YZXZXIY(7)\}$, $\{IYYZZXX(4), XIXYZYZ(1), YXZIYXZ(3)\}$, $\{ZZIXYYX(6), XIXYZYZ(1), YXZIYXZ(3)\}$, $\{XIXYZYZ(1), YZXZXIY(7), YXZIYXZ(3)\}$, $\{IYYZZXX(4), XIXYZYZ(1), ZXYYXZI(5)\}$, $\{ZZIXYYX(6), IYYZZXX(4), XIXYZYZ(1)\}$, $\{IYYZZXX(4), XIXYZYZ(1), YZXZXIY(7)\}$, $\{ZZIXYYX(6), XIXYZYZ(1), ZXYYXZI(5)\}$, $\{XIXYZYZ(1), ZXYYXZI(5), YZXZXIY(7)\}$, $\{XYZXIZY(2), IYYZZXX(4), YXZIYXZ(3)\}$, $\{XYZXIZY(2), ZXYYXZI(5), YXZIYXZ(3)\}$, $\{XYZXIZY(2), YZXZXIY(7), YXZIYXZ(3)\}$, $\{XYZXIZY(2), IYYZZXX(4), ZXYYXZI(5)\}$, $\{ZZIXYYX(6), XYZXIZY(2), IYYZZXX(4)\}$, $\{XYZXIZY(2), IYYZZXX(4), YZXZXIY(7)\}$, $\{XYZXIZY(2), ZZIXYYX(6), ZXYYXZI(5)\}$, $\{XYZXIZY(2), ZZIXYYX(6), YZXZXIY(7)\}$, $\{IYYZZXX(4), ZXYYXZI(5), YXZIYXZ(3)\}$, $\{ZZIXYYX(6), IYYZZXX(4), YXZIYXZ(3)\}$, $\{ZZIXYYX(6), ZXYYXZI(5), YXZIYXZ(3)\}$, $\{YZXZXIY(7), ZXYYXZI(5), YXZIYXZ(3)\}$, $\{ZZIXYYX(6), YZXZXIY(7), YXZIYXZ(3)\}$, $\{IYYZZXX(4), YZXZXIY(7), ZXYYXZI(5)\}$, $\{ZZIXYYX(6), IYYZZXX(4), YZXZXIY(7)\}$, $\{ZZIXYYX(6), YZXZXIY(7), ZXYYXZI(5)\}$	$\{IYYZZXX, XIXYZYZ, XYZXIZY, YXZIYXZ, YZXZXIY, ZXYYXZI, ZZIXYYX, IIIIIII\}$
$\{XXIZYYZ(1), ZXYIZXY(3), IZZXXYY(2)\}$, $\{XXIZYYZ(1), IZZXXYY(2), YIYZXZX(5)\}$, $\{XXIZYYZ(1), IZZXXYY(2), ZYXXYZI(6)\}$, $\{XXIZYYZ(1), YZXYIXZ(7), IZZXXYY(2)\}$, $\{XXIZYYZ(1), ZXYIZXY(3), XYZYZIX(4)\}$, $\{XXIZYYZ(1), ZXYIZXY(3), ZYXXYZI(6)\}$, $\{XXIZYYZ(1), YZXYIXZ(7), ZXYIZXY(3)\}$, $\{XXIZYYZ(1), XYZYZIX(4), YIYZXZX(5)\}$, $\{XXIZYYZ(1), XYZYZIX(4), ZYXXYZI(6)\}$, $\{XXIZYYZ(1), YZXYIXZ(7), XYZYZIX(4)\}$, $\{XXIZYYZ(1), ZYXXYZI(6), YIYZXZX(5)\}$, $\{XXIZYYZ(1), YZXYIXZ(7), YIYZXZX(5)\}$, $\{ZXYIZXY(3), IZZXXYY(2), XYZYZIX(4)\}$, $\{ZXYIZXY(3), IZZXXYY(2), YIYZXZX(5)\}$, $\{YZXYIXZ(7), ZXYIZXY(3), IZZXXYY(2)\}$, $\{XYZYZIX(4), IZZXXYY(2), YIYZXZX(5)\}$, $\{XYZYZIX(4), IZZXXYY(2), ZYXXYZI(6)\}$, $\{YZXYIXZ(7), XYZYZIX(4), IZZXXYY(2)\}$, $\{IZZXXYY(2), YIYZXZX(5), ZYXXYZI(6)\}$, $\{YZXYIXZ(7), IZZXXYY(2), ZYXXYZI(6)\}$, $\{YIYZXZX(5), ZXYIZXY(3), XYZYZIX(4)\}$, $\{ZXYIZXY(3), ZYXXYZI(6), XYZYZIX(4)\}$, $\{ZXYIZXY(3), ZYXXYZI(6), YIYZXZX(5)\}$, $\{YZXYIXZ(7), ZXYIZXY(3), YIYZXZX(5)\}$, $\{YZXYIXZ(7), ZXYIZXY(3), ZYXXYZI(6)\}$, $\{YZXYIXZ(7), XYZYZIX(4), YIYZXZX(5)\}$, $\{YZXYIXZ(7), XYZYZIX(4), ZYXXYZI(6)\}$, $\{YZXYIXZ(7), ZYXXYZI(6), YIYZXZX(5)\}$	$\{IZZXXYY, XXIZYYZ, XYZYZIX, YIYZXZX, YZXYIXZ, ZXYIZXY, ZYXXYZI, IIIIIII\}$

Table G.11: Continued from Tables G.1-G.10 and in Tables G.12-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{XXIZYYZ(1), IZZXXYY(2), YIYXZXZ(3)\}$, $\{XXIZYYZ(1), ZXYYXZI(5), IZZXXYY(2)\}$, $\{XXIZYYZ(1), IZZXXYY(2), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XXIZYYZ(1), IZZXXYY(2)\}$, $\{XXIZYYZ(1), XYZYZIX(4), YIYXZXZ(3)\}$, $\{XXIZYYZ(1), YIYXZXZ(3), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XXIZYYZ(1), YIYXZXZ(3)\}$, $\{XXIZYYZ(1), XYZYZIX(4), ZXYYXZI(5)\}$, $\{XXIZYYZ(1), XYZYZIX(4), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XXIZYYZ(1), XYZYZIX(4)\}$, $\{XXIZYYZ(1), ZXYYXZI(5), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XXIZYYZ(1), ZXYYXZI(5)\}$, $\{XYZYZIX(4), IZZXXYY(2), YIYXZXZ(3)\}$, $\{ZXYYXZI(5), IZZXXYY(2), YIYXZXZ(3)\}$, $\{ZYXZIXY(7), IZZXXYY(2), YIYXZXZ(3)\}$, $\{ZXYYXZI(5), XYZYZIX(4), IZZXXYY(2)\}$, $\{XYZYZIX(4), IZZXXYY(2), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XYZYZIX(4), IZZXXYY(2)\}$, $\{ZXYYXZI(5), IZZXXYY(2), YZXIYZX(6)\}$, $\{ZYXZIXY(7), IZZXXYY(2), YZXIYZX(6)\}$, $\{XYZYZIX(4), ZXYYXZI(5), YIYXZXZ(3)\}$, $\{XYZYZIX(4), YIYXZXZ(3), YZXIYZX(6)\}$, $\{ZXYYXZI(5), YIYXZXZ(3), YZXIYZX(6)\}$, $\{ZYXZIXY(7), ZXYYXZI(5), YIYXZXZ(3)\}$, $\{ZYXZIXY(7), YIYXZXZ(3), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XYZYZIX(4), ZXYYXZI(5)\}$, $\{ZYXZIXY(7), XYZYZIX(4), YZXIYZX(6)\}$, $\{ZYXZIXY(7), ZXYYXZI(5), YZXIYZX(6)\}$	$\{IZZXXYY, XXIZYYZ, XYZYZIX, YIYXZXZ, YZXIYZX, ZXYYXZI, ZYXZIXY, IIIIIII\}$
$\{XZYIXZY(2), XXIZYYZ(1), ZIZXYXY(3)\}$, $\{YXZYIZX(5), XZYIXZY(2), XXIZYYZ(1)\}$, $\{XZYIXZY(2), XXIZYYZ(1), YZXXZYI(6)\}$, $\{XZYIXZY(2), XXIZYYZ(1), ZYXYXIZ(7)\}$, $\{XXIZYYZ(1), ZIZXYXY(3), IYYZZXX(4)\}$, $\{XXIZYYZ(1), YZXXZYI(6), ZIZXYXY(3)\}$, $\{XXIZYYZ(1), ZIZXYXY(3), ZYXYXIZ(7)\}$, $\{YXZYIZX(5), XXIZYYZ(1), IYYZZXX(4)\}$, $\{XXIZYYZ(1), YZXXZYI(6), IYYZZXX(4)\}$, $\{XXIZYYZ(1), ZYXYXIZ(7), IYYZZXX(4)\}$, $\{YXZYIZX(5), XXIZYYZ(1), YZXXZYI(6)\}$, $\{YXZYIZX(5), XXIZYYZ(1), ZYXYXIZ(7)\}$, $\{XZYIXZY(2), ZIZXYXY(3), IYYZZXX(4)\}$, $\{YXZYIZX(5), XZYIXZY(2), ZIZXYXY(3)\}$, $\{XZYIXZY(2), ZIZXYXY(3), ZYXYXIZ(7)\}$, $\{YXZYIZX(5), XZYIXZY(2), IYYZZXX(4)\}$, $\{XZYIXZY(2), YZXXZYI(6), IYYZZXX(4)\}$, $\{XZYIXZY(2), ZYXYXIZ(7), IYYZZXX(4)\}$, $\{YXZYIZX(5), XZYIXZY(2), YZXXZYI(6)\}$, $\{XZYIXZY(2), YZXXZYI(6), ZYXYXIZ(7)\}$, $\{YXZYIZX(5), IYYZZXX(4), ZIZXYXY(3)\}$, $\{IYYZZXX(4), YZXXZYI(6), ZIZXYXY(3)\}$, $\{YXZYIZX(5), YZXXZYI(6), ZIZXYXY(3)\}$, $\{YXZYIZX(5), ZIZXYXY(3), ZYXYXIZ(7)\}$, $\{YZXXZYI(6), ZIZXYXY(3), ZYXYXIZ(7)\}$, $\{YXZYIZX(5), IYYZZXX(4), ZYXYXIZ(7)\}$, $\{IYYZZXX(4), YZXXZYI(6), ZYXYXIZ(7)\}$, $\{YXZYIZX(5), YZXXZYI(6), ZYXYXIZ(7)\}$	$\{IYYZZXX, XXIZYYZ, XZYIXZY, YXZYIZX, YZXXZYI, ZIZXYXY, ZYXYXIZ, IIIIIII\}$
$\{XZYIXZY(2), XXIZYYZ(1), YXZXZIY(3)\}$, $\{XZYIXZY(2), XXIZYYZ(1), ZIZYXYX(5)\}$, $\{XZYIXZY(2), XXIZYYZ(1), ZYXXYZI(6)\}$, $\{XXIZYYZ(1), YXZXZIY(3), IYYZZXX(4)\}$, $\{XXIZYYZ(1), YXZXZIY(3), ZYXXYZI(6)\}$, $\{XXIZYYZ(1), ZIZYXYX(5), IYYZZXX(4)\}$, $\{XXIZYYZ(1), ZYXXYZI(6), IYYZZXX(4)\}$, $\{XXIZYYZ(1), YZXYIXZ(7), IYYZZXX(4)\}$, $\{XXIZYYZ(1), ZIZYXYX(5), ZYXXYZI(6)\}$, $\{XZYIXZY(2), YXZXZIY(3), IYYZZXX(4)\}$, $\{XZYIXZY(2), ZIZYXYX(5), YXZXZIY(3)\}$, $\{XZYIXZY(2), YZXYIXZ(7), YXZXZIY(3)\}$, $\{XZYIXZY(2), ZIZYXYX(5), IYYZZXX(4)\}$, $\{XZYIXZY(2), ZYXXYZI(6), IYYZZXX(4)\}$, $\{XZYIXZY(2), YZXYIXZ(7), IYYZZXX(4)\}$, $\{XZYIXZY(2), ZIZYXYX(5), ZYXXYZI(6)\}$, $\{XZYIXZY(2), YZXYIXZ(7), ZYXXYZI(6)\}$, $\{IYYZZXX(4), ZIZYXYX(5), YXZXZIY(3)\}$, $\{IYYZZXX(4), YXZXZIY(3), ZYXXYZI(6)\}$, $\{ZIZYXYX(5), YXZXZIY(3), ZYXXYZI(6)\}$, $\{YZXYIXZ(7), ZIZYXYX(5), YXZXZIY(3)\}$, $\{YZXYIXZ(7), YXZXZIY(3), ZYXXYZI(6)\}$, $\{IYYZZXX(4), YZXYIXZ(7), ZIZYXYX(5)\}$, $\{IYYZZXX(4), YZXYIXZ(7), ZYXXYZI(6)\}$, $\{YZXYIXZ(7), ZIZYXYX(5), ZYXXYZI(6)\}$	$\{IYYZZXX, XXIZYYZ, XZYIXZY, YXZXZIY, YZXYIXZ, ZIZYXYX, ZYXXYZI, IIIIIII\}$

Table G.12: Continued from Tables G.1-G.11 and in Tables G.13-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IYYXXZZ(2), XXIZYYZ(1), ZIZXYXY(3)\}$, $\{IYYXXZZ(2), XXIZYYZ(1), YXZYIZX(5)\}$, $\{IYYXXZZ(2), XXIZYYZ(1), ZYXIZYX(6)\}$, $\{IYYXXZZ(2), XXIZYYZ(1), YZXZXIY(7)\}$, $\{XZYYZXI(4), XXIZYYZ(1), ZIZXYXY(3)\}$, $\{XXIZYYZ(1), ZYXIZYX(6), ZIZXYXY(3)\}$, $\{XXIZYYZ(1), YZXZXIY(7), ZIZXYXY(3)\}$, $\{XZYYZXI(4), XXIZYYZ(1), YXZYIZX(5)\}$, $\{XZYYZXI(4), XXIZYYZ(1), ZYXIZYX(6)\}$, $\{XZYYZXI(4), XXIZYYZ(1), YZXZXIY(7)\}$, $\{YXZYIZX(5), XXIZYYZ(1), ZYXIZYX(6)\}$, $\{YXZYIZX(5), XXIZYYZ(1), YZXZXIY(7)\}$, $\{IYYXXZZ(2), XZYYZXI(4), ZIZXYXY(3)\}$, $\{IYYXXZZ(2), ZIZXYXY(3), YXZYIZX(5)\}$, $\{IYYXXZZ(2), YZXZXIY(7), ZIZXYXY(3)\}$, $\{IYYXXZZ(2), XZYYZXI(4), YXZYIZX(5)\}$, $\{IYYXXZZ(2), XZYYZXI(4), ZYXIZYX(6)\}$, $\{IYYXXZZ(2), XZYYZXI(4), YZXZXIY(7)\}$, $\{IYYXXZZ(2), ZYXIZYX(6), YXZYIZX(5)\}$, $\{IYYXXZZ(2), ZYXIZYX(6), YZXZXIY(7)\}$, $\{XZYYZXI(4), ZIZXYXY(3), YXZYIZX(5)\}$, $\{XZYYZXI(4), ZYXIZYX(6), ZIZXYXY(3)\}$, $\{YXZYIZX(5), ZYXIZYX(6), ZIZXYXY(3)\}$, $\{YXZYIZX(5), YZXZXIY(7), ZIZXYXY(3)\}$, $\{ZYXIZYX(6), ZIZXYXY(3), YZXZXIY(7)\}$, $\{XZYYZXI(4), YZXZXIY(7), YXZYIZX(5)\}$, $\{XZYYZXI(4), ZYXIZYX(6), YZXZXIY(7)\}$, $\{YXZYIZX(5), ZYXIZYX(6), YZXZXIY(7)\}$	$\{IYYXXZZ, XXIZYYZ, XZYYZXI, YXZYIZX, YZXZXIY, ZIZXYXY, ZYXIZYX, IIIIIII\}$
$\{IYYXXZZ(2), XXIZYYZ(1), YXZXZIY(3)\}$, $\{IYYXXZZ(2), XXIZYYZ(1), ZIZYXYX(5)\}$, $\{IYYXXZZ(2), XXIZYYZ(1), YZXIYZX(6)\}$, $\{IYYXXZZ(2), ZYXZIXY(7), XXIZYYZ(1)\}$, $\{XZYYZXI(4), XXIZYYZ(1), YXZXZIY(3)\}$, $\{XXIZYYZ(1), YXZXZIY(3), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XXIZYYZ(1), YXZXZIY(3)\}$, $\{XZYYZXI(4), XXIZYYZ(1), ZIZYXYX(5)\}$, $\{XZYYZXI(4), XXIZYYZ(1), YZXIYZX(6)\}$, $\{XZYYZXI(4), ZYXZIXY(7), XXIZYYZ(1)\}$, $\{XXIZYYZ(1), ZIZYXYX(5), YZXIYZX(6)\}$, $\{ZYXZIXY(7), XXIZYYZ(1), ZIZYXYX(5)\}$, $\{IYYXXZZ(2), XZYYZXI(4), YXZXZIY(3)\}$, $\{IYYXXZZ(2), ZIZYXYX(5), YXZXZIY(3)\}$, $\{IYYXXZZ(2), ZYXZIXY(7), YXZXZIY(3)\}$, $\{IYYXXZZ(2), XZYYZXI(4), ZIZYXYX(5)\}$, $\{IYYXXZZ(2), XZYYZXI(4), YZXIYZX(6)\}$, $\{IYYXXZZ(2), ZYXZIXY(7), XZYYZXI(4)\}$, $\{IYYXXZZ(2), ZIZYXYX(5), YZXIYZX(6)\}$, $\{IYYXXZZ(2), ZYXZIXY(7), YZXIYZX(6)\}$, $\{XZYYZXI(4), ZIZYXYX(5), YXZXZIY(3)\}$, $\{XZYYZXI(4), YXZXZIY(3), YZXIYZX(6)\}$, $\{ZIZYXYX(5), YXZXZIY(3), YZXIYZX(6)\}$, $\{ZYXZIXY(7), ZIZYXYX(5), YXZXZIY(3)\}$, $\{ZYXZIXY(7), YXZXZIY(3), YZXIYZX(6)\}$, $\{XZYYZXI(4), ZYXZIXY(7), ZIZYXYX(5)\}$, $\{XZYYZXI(4), ZYXZIXY(7), YZXIYZX(6)\}$, $\{ZYXZIXY(7), ZIZYXYX(5), YZXIYZX(6)\}$	$\{IYYXXZZ, XXIZYYZ, XZYYZXI, YXZXZIY, YZXIYZX, ZIZYXYX, ZYXZIXY, IIIIIII\}$
$\{XYZXIZY(2), XXIZYYZ(1), ZXYIZXY(3)\}$, $\{XYZXIZY(2), XXIZYYZ(1), YIYZXZX(5)\}$, $\{XYZXIZY(2), XXIZYYZ(1), YZXXZYI(6)\}$, $\{XYZXIZY(2), XXIZYYZ(1), ZYXYXIZ(7)\}$, $\{IZZYYXX(4), XXIZYYZ(1), ZXYIZXY(3)\}$, $\{XXIZYYZ(1), YZXXZYI(6), ZXYIZXY(3)\}$, $\{XXIZYYZ(1), ZXYIZXY(3), ZYXYXIZ(7)\}$, $\{IZZYYXX(4), XXIZYYZ(1), YIYZXZX(5)\}$, $\{IZZYYXX(4), XXIZYYZ(1), YZXXZYI(6)\}$, $\{IZZYYXX(4), XXIZYYZ(1), ZYXYXIZ(7)\}$, $\{XXIZYYZ(1), YZXXZYI(6), YIYZXZX(5)\}$, $\{XXIZYYZ(1), ZYXYXIZ(7), YIYZXZX(5)\}$, $\{IZZYYXX(4), XYZXIZY(2), ZXYIZXY(3)\}$, $\{XYZXIZY(2), ZXYIZXY(3), YIYZXZX(5)\}$, $\{XYZXIZY(2), ZXYIZXY(3), ZYXYXIZ(7)\}$, $\{IZZYYXX(4), XYZXIZY(2), YIYZXZX(5)\}$, $\{IZZYYXX(4), XYZXIZY(2), YZXXZYI(6)\}$, $\{IZZYYXX(4), XYZXIZY(2), ZYXYXIZ(7)\}$, $\{XYZXIZY(2), YZXXZYI(6), YIYZXZX(5)\}$, $\{XYZXIZY(2), YZXXZYI(6), ZYXYXIZ(7)\}$, $\{IZZYYXX(4), ZXYIZXY(3), YIYZXZX(5)\}$, $\{IZZYYXX(4), YZXXZYI(6), ZXYIZXY(3)\}$, $\{YZXXZYI(6), ZXYIZXY(3), YIYZXZX(5)\}$, $\{ZXYIZXY(3), ZYXYXIZ(7), YIYZXZX(5)\}$, $\{YZXXZYI(6), ZXYIZXY(3), ZYXYXIZ(7)\}$, $\{IZZYYXX(4), ZYXYXIZ(7), YIYZXZX(5)\}$, $\{IZZYYXX(4), YZXXZYI(6), ZYXYXIZ(7)\}$, $\{YZXXZYI(6), ZYXYXIZ(7), YIYZXZX(5)\}$	$\{IZZYYXX, XXIZYYZ, XYZXIZY, YIYZXZX, YZXXZYI, ZXYIZXY, ZYXYXIZ, IIIIIII\}$

Table G.13: Continued from Tables G.1-G.12 and in Tables G.14-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{XYZXIZY(2), XXIZYYZ(1), YIYXZXZ(3)\}$, $\{XYZXIZY(2), XXIZYYZ(1), ZXYYXZI(5)\}$, $\{XYZXIZY(2), XXIZYYZ(1), ZYXIZYX(6)\}$, $\{XYZXIZY(2), XXIZYYZ(1), YZXZXIY(7)\}$, $\{IZZYYXX(4), XXIZYYZ(1), YIYXZXZ(3)\}$, $\{XXIZYYZ(1), ZYXIZYX(6), YIYXZXZ(3)\}$, $\{XXIZYYZ(1), YZXZXIY(7), YIYXZXZ(3)\}$, $\{IZZYYXX(4), XXIZYYZ(1), ZXYYXZI(5)\}$, $\{IZZYYXX(4), XXIZYYZ(1), ZYXIZYX(6)\}$, $\{IZZYYXX(4), XXIZYYZ(1), YZXZXIY(7)\}$, $\{XXIZYYZ(1), ZYXIZYX(6), ZXYYXZI(5)\}$, $\{XXIZYYZ(1), YZXZXIY(7), ZXYYXZI(5)\}$, $\{IZZYYXX(4), XYZXIZY(2), YIYXZXZ(3)\}$, $\{XYZXIZY(2), ZXYYXZI(5), YIYXZXZ(3)\}$, $\{XYZXIZY(2), YZXZXIY(7), YIYXZXZ(3)\}$, $\{IZZYYXX(4), XYZXIZY(2), ZXYYXZI(5)\}$, $\{IZZYYXX(4), XYZXIZY(2), ZYXIZYX(6)\}$, $\{IZZYYXX(4), XYZXIZY(2), YZXZXIY(7)\}$, $\{XYZXIZY(2), ZYXIZYX(6), ZXYYXZI(5)\}$, $\{XYZXIZY(2), ZYXIZYX(6), YZXZXIY(7)\}$, $\{IZZYYXX(4), ZXYYXZI(5), YIYXZXZ(3)\}$, $\{IZZYYXX(4), ZYXIZYX(6), YIYXZXZ(3)\}$, $\{ZYXIZYX(6), ZXYYXZI(5), YIYXZXZ(3)\}$, $\{YZXZXIY(7), ZXYYXZI(5), YIYXZXZ(3)\}$, $\{ZYXIZYX(6), YZXZXIY(7), YIYXZXZ(3)\}$, $\{IZZYYXX(4), YZXZXIY(7), ZXYYXZI(5)\}$, $\{IZZYYXX(4), ZYXIZYX(6), YZXZXIY(7)\}$, $\{ZYXIZYX(6), ZXYYXZI(5), YZXZXIY(7)\}$	$\{IZZYYXX, XXIZYYZ, XYZXIZY, YIYXZXZ, YZXZXIY, ZXYYXZI, ZYXIZYX, IIIIIII\}$
$\{XXIYZZY(1), ZXYXYIZ(3), IZZXXYY(2)\}$, $\{XXIYZZY(1), IZZXXYY(2), YIYZXZX(5)\}$, $\{XXIYZZY(1), ZYXIZYX(6), IZZXXYY(2)\}$, $\{XXIYZZY(1), YZXYIXZ(7), IZZXXYY(2)\}$, $\{XXIYZZY(1), ZXYXYIZ(3), XYZZYXI(4)\}$, $\{XXIYZZY(1), ZXYXYIZ(3), ZYXIZYX(6)\}$, $\{XXIYZZY(1), ZXYXYIZ(3), YZXYIXZ(7)\}$, $\{XXIYZZY(1), XYZZYXI(4), YIYZXZX(5)\}$, $\{XXIYZZY(1), XYZZYXI(4), ZYXIZYX(6)\}$, $\{XXIYZZY(1), XYZZYXI(4), YZXYIXZ(7)\}$, $\{XXIYZZY(1), ZYXIZYX(6), YIYZXZX(5)\}$, $\{XXIYZZY(1), YZXYIXZ(7), YIYZXZX(5)\}$, $\{ZXYXYIZ(3), XYZZYXI(4), IZZXXYY(2)\}$, $\{ZXYXYIZ(3), IZZXXYY(2), YIYZXZX(5)\}$, $\{ZXYXYIZ(3), YZXYIXZ(7), IZZXXYY(2)\}$, $\{XYZZYXI(4), IZZXXYY(2), YIYZXZX(5)\}$, $\{XYZZYXI(4), ZYXIZYX(6), IZZXXYY(2)\}$, $\{XYZZYXI(4), YZXYIXZ(7), IZZXXYY(2)\}$, $\{ZYXIZYX(6), IZZXXYY(2), YIYZXZX(5)\}$, $\{YZXYIXZ(7), ZYXIZYX(6), IZZXXYY(2)\}$, $\{ZXYXYIZ(3), XYZZYXI(4), YIYZXZX(5)\}$, $\{ZXYXYIZ(3), ZYXIZYX(6), XYZZYXI(4)\}$, $\{ZXYXYIZ(3), ZYXIZYX(6), YIYZXZX(5)\}$, $\{ZXYXYIZ(3), YZXYIXZ(7), YIYZXZX(5)\}$, $\{YZXYIXZ(7), ZXYXYIZ(3), ZYXIZYX(6)\}$, $\{XYZZYXI(4), YZXYIXZ(7), YIYZXZX(5)\}$, $\{YZXYIXZ(7), XYZZYXI(4), ZYXIZYX(6)\}$, $\{YZXYIXZ(7), ZYXIZYX(6), YIYZXZX(5)\}$	$\{IZZXXYY, XXIYZZY, XYZZYXI, YIYZXZX, YZXYIXZ, ZXYXYIZ, ZYXIZYX, IIIIIII\}$
$\{XXIYZZY(1), IZZXXYY(2), YIYXZXZ(3)\}$, $\{XXIYZZY(1), ZXYZIYX(5), IZZXXYY(2)\}$, $\{XXIYZZY(1), IZZXXYY(2), YZXIYZX(6)\}$, $\{XXIYZZY(1), IZZXXYY(2), ZYXYXIZ(7)\}$, $\{XXIYZZY(1), XYZZYXI(4), YIYXZXZ(3)\}$, $\{XXIYZZY(1), YIYXZXZ(3), YZXIYZX(6)\}$, $\{XXIYZZY(1), ZYXYXIZ(7), YIYXZXZ(3)\}$, $\{XXIYZZY(1), XYZZYXI(4), ZXYZIYX(5)\}$, $\{XXIYZZY(1), XYZZYXI(4), YZXIYZX(6)\}$, $\{XXIYZZY(1), XYZZYXI(4), ZYXYXIZ(7)\}$, $\{XXIYZZY(1), ZXYZIYX(5), YZXIYZX(6)\}$, $\{XXIYZZY(1), ZXYZIYX(5), ZYXYXIZ(7)\}$, $\{XYZZYXI(4), IZZXXYY(2), YIYXZXZ(3)\}$, $\{ZXYZIYX(5), IZZXXYY(2), YIYXZXZ(3)\}$, $\{ZYXYXIZ(7), IZZXXYY(2), YIYXZXZ(3)\}$, $\{XYZZYXI(4), ZXYZIYX(5), IZZXXYY(2)\}$, $\{XYZZYXI(4), IZZXXYY(2), YZXIYZX(6)\}$, $\{XYZZYXI(4), IZZXXYY(2), ZYXYXIZ(7)\}$, $\{ZXYZIYX(5), IZZXXYY(2), YZXIYZX(6)\}$, $\{IZZXXYY(2), ZYXYXIZ(7), YZXIYZX(6)\}$, $\{XYZZYXI(4), ZXYZIYX(5), YIYXZXZ(3)\}$, $\{XYZZYXI(4), YIYXZXZ(3), YZXIYZX(6)\}$, $\{ZXYZIYX(5), YIYXZXZ(3), YZXIYZX(6)\}$, $\{ZXYZIYX(5), ZYXYXIZ(7), YIYXZXZ(3)\}$, $\{ZYXYXIZ(7), YIYXZXZ(3), YZXIYZX(6)\}$, $\{XYZZYXI(4), ZXYZIYX(5), ZYXYXIZ(7)\}$, $\{XYZZYXI(4), ZYXYXIZ(7), YZXIYZX(6)\}$, $\{ZXYZIYX(5), ZYXYXIZ(7), YZXIYZX(6)\}$	$\{IZZXXYY, XXIYZZY, XYZZYXI, YIYXZXZ, YZXIYZX, ZXYZIYX, ZYXYXIZ, IIIIIII\}$

Table G.14: Continued from Tables G.1-G.13 and in Tables G.15-G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{IYYXXZZ(2), XXIYZZY(1), ZIZXYXY(3)\}$, $\{IYYXXZZ(2), XXIYZZY(1), YXZZXYI(5)\}$, $\{IYYXXZZ(2), XXIYZZY(1), ZYXIZYX(6)\}$, $\{IYYXXZZ(2), XXIYZZY(1), YZXYIXZ(7)\}$, $\{XXIYZZY(1), ZIZXYXY(3), XZYZYIX(4)\}$, $\{XXIYZZY(1), ZYXIZYX(6), ZIZXYXY(3)\}$, $\{XXIYZZY(1), YZXYIXZ(7), ZIZXYXY(3)\}$, $\{XXIYZZY(1), XZYZYIX(4), YXZZXYI(5)\}$, $\{XXIYZZY(1), ZYXIZYX(6), XZYZYIX(4)\}$, $\{XXIYZZY(1), YZXYIXZ(7), XZYZYIX(4)\}$, $\{XXIYZZY(1), ZYXIZYX(6), YXZZXYI(5)\}$, $\{XXIYZZY(1), YZXYIXZ(7), YXZZXYI(5)\}$, $\{IYYXXZZ(2), ZIZXYXY(3), XZYZYIX(4)\}$, $\{IYYXXZZ(2), ZIZXYXY(3), YXZZXYI(5)\}$, $\{IYYXXZZ(2), YZXYIXZ(7), ZIZXYXY(3)\}$, $\{IYYXXZZ(2), XZYZYIX(4), YXZZXYI(5)\}$, $\{IYYXXZZ(2), ZYXIZYX(6), XZYZYIX(4)\}$, $\{IYYXXZZ(2), YZXYIXZ(7), XZYZYIX(4)\}$, $\{IYYXXZZ(2), ZYXIZYX(6), YXZZXYI(5)\}$, $\{IYYXXZZ(2), YZXYIXZ(7), ZYXIZYX(6)\}$, $\{ZIZXYXY(3), XZYZYIX(4), YXZZXYI(5)\}$, $\{ZYXIZYX(6), ZIZXYXY(3), XZYZYIX(4)\}$, $\{ZYXIZYX(6), ZIZXYXY(3), YXZZXYI(5)\}$, $\{YZXYIXZ(7), ZIZXYXY(3), YXZZXYI(5)\}$, $\{YZXYIXZ(7), ZYXIZYX(6), ZIZXYXY(3)\}$, $\{YZXYIXZ(7), XZYZYIX(4), YXZZXYI(5)\}$, $\{YZXYIXZ(7), ZYXIZYX(6), XZYZYIX(4)\}$, $\{YZXYIXZ(7), ZYXIZYX(6), YXZZXYI(5)\}$	$\{IYYXXZZ, XXIYZZY, XZYZYIX, YXZZXYI, YZXYIXZ, ZIZXYXY, ZYXIZYX, IIIIIII\}$
$\{IYYXXZZ(2), XXIYZZY(1), YXZIYXZ(3)\}$, $\{IYYXXZZ(2), XXIYZZY(1), ZIZYXYX(5)\}$, $\{IYYXXZZ(2), XXIYZZY(1), YZXXZYI(6)\}$, $\{IYYXXZZ(2), XXIYZZY(1), ZYXZIXY(7)\}$, $\{XXIYZZY(1), XZYZYIX(4), YXZIYXZ(3)\}$, $\{XXIYZZY(1), YZXXZYI(6), YXZIYXZ(3)\}$, $\{XXIYZZY(1), ZYXZIXY(7), YXZIYXZ(3)\}$, $\{XXIYZZY(1), ZIZYXYX(5), XZYZYIX(4)\}$, $\{XXIYZZY(1), YZXXZYI(6), XZYZYIX(4)\}$, $\{XXIYZZY(1), ZYXZIXY(7), XZYZYIX(4)\}$, $\{XXIYZZY(1), YZXXZYI(6), ZIZYXYX(5)\}$, $\{XXIYZZY(1), ZIZYXYX(5), ZYXZIXY(7)\}$, $\{IYYXXZZ(2), XZYZYIX(4), YXZIYXZ(3)\}$, $\{IYYXXZZ(2), ZIZYXYX(5), YXZIYXZ(3)\}$, $\{IYYXXZZ(2), ZYXZIXY(7), YXZIYXZ(3)\}$, $\{IYYXXZZ(2), ZIZYXYX(5), XZYZYIX(4)\}$, $\{IYYXXZZ(2), YZXXZYI(6), XZYZYIX(4)\}$, $\{IYYXXZZ(2), ZYXZIXY(7), XZYZYIX(4)\}$, $\{IYYXXZZ(2), YZXXZYI(6), ZIZYXYX(5)\}$, $\{IYYXXZZ(2), ZYXZIXY(7), YZXXZYI(6)\}$, $\{ZIZYXYX(5), XZYZYIX(4), YXZIYXZ(3)\}$, $\{YZXXZYI(6), XZYZYIX(4), YXZIYXZ(3)\}$, $\{YZXXZYI(6), ZIZYXYX(5), YXZIYXZ(3)\}$, $\{ZYXZIXY(7), ZIZYXYX(5), YXZIYXZ(3)\}$, $\{ZYXZIXY(7), YZXXZYI(6), YXZIYXZ(3)\}$, $\{ZYXZIXY(7), ZIZYXYX(5), XZYZYIX(4)\}$, $\{ZYXZIXY(7), YZXXZYI(6), XZYZYIX(4)\}$, $\{ZYXZIXY(7), YZXXZYI(6), ZIZYXYX(5)\}$	$\{IYYXXZZ, XXIYZZY, XZYZYIX, YXZIYXZ, YZXXZYI, ZIZYXYX, ZYXZIXY, IIIIIII\}$
$\{XXIYZZY(1), XZYXIYZ(2), ZIZXYXY(3)\}$, $\{XXIYZZY(1), XZYXIYZ(2), YXZZXYI(5)\}$, $\{XXIYZZY(1), XZYXIYZ(2), YZXIYZX(6)\}$, $\{XXIYZZY(1), XZYXIYZ(2), ZYXYXIZ(7)\}$, $\{XXIYZZY(1), IYYZZXX(4), ZIZXYXY(3)\}$, $\{XXIYZZY(1), ZIZXYXY(3), YZXIYZX(6)\}$, $\{XXIYZZY(1), ZIZXYXY(3), ZYXYXIZ(7)\}$, $\{XXIYZZY(1), IYYZZXX(4), YXZZXYI(5)\}$, $\{XXIYZZY(1), IYYZZXX(4), YZXIYZX(6)\}$, $\{XXIYZZY(1), IYYZZXX(4), ZYXYXIZ(7)\}$, $\{XXIYZZY(1), YZXIYZX(6), YXZZXYI(5)\}$, $\{XXIYZZY(1), ZYXYXIZ(7), YXZZXYI(5)\}$, $\{IYYZZXX(4), XZYXIYZ(2), ZIZXYXY(3)\}$, $\{XZYXIYZ(2), ZIZXYXY(3), YXZZXYI(5)\}$, $\{XZYXIYZ(2), ZIZXYXY(3), ZYXYXIZ(7)\}$, $\{IYYZZXX(4), XZYXIYZ(2), YXZZXYI(5)\}$, $\{IYYZZXX(4), XZYXIYZ(2), YZXIYZX(6)\}$, $\{IYYZZXX(4), XZYXIYZ(2), ZYXYXIZ(7)\}$, $\{YZXIYZX(6), XZYXIYZ(2), YXZZXYI(5)\}$, $\{XZYXIYZ(2), ZYXYXIZ(7), YZXIYZX(6)\}$, $\{IYYZZXX(4), ZIZXYXY(3), YXZZXYI(5)\}$, $\{IYYZZXX(4), ZIZXYXY(3), YZXIYZX(6)\}$, $\{YZXIYZX(6), ZIZXYXY(3), YXZZXYI(5)\}$, $\{ZIZXYXY(3), ZYXYXIZ(7), YXZZXYI(5)\}$, $\{ZIZXYXY(3), ZYXYXIZ(7), YZXIYZX(6)\}$, $\{IYYZZXX(4), ZYXYXIZ(7), YXZZXYI(5)\}$, $\{IYYZZXX(4), ZYXYXIZ(7), YZXIYZX(6)\}$, $\{YZXIYZX(6), ZYXYXIZ(7), YXZZXYI(5)\}$	$\{IYYZZXX, XXIYZZY, XZYXIYZ, YXZZXYI, YZXIYZX, ZIZXYXY, ZYXYXIZ, IIIIIII\}$

Table G.15: Continued from Tables G.1-G.14 and in Table G.16.

Stabilizer triplets which measure at least 2 different Paulis on every qubit	Generated subgroups
$\{XXIYZZY(1), XZYXIYZ(2), YXZIYXZ(3)\}$, $\{XXIYZZY(1), XZYXIYZ(2), ZIZYXYX(5)\}$, $\{XXIYZZY(1), XZYXIYZ(2), ZYXXYZI(6)\}$, $\{XXIYZZY(1), XZYXIYZ(2), YZXZXIY(7)\}$, $\{XXIYZZY(1), IYYZZXX(4), YXZIYXZ(3)\}$, $\{XXIYZZY(1), ZYXXYZI(6), YXZIYXZ(3)\}$, $\{XXIYZZY(1), YZXZXIY(7), YXZIYXZ(3)\}$, $\{XXIYZZY(1), IYYZZXX(4), ZIZYXYX(5)\}$, $\{XXIYZZY(1), IYYZZXX(4), ZYXXYZI(6)\}$, $\{XXIYZZY(1), IYYZZXX(4), YZXZXIY(7)\}$, $\{XXIYZZY(1), ZIZYXYX(5), ZYXXYZI(6)\}$, $\{XXIYZZY(1), YZXZXIY(7), ZIZYXYX(5)\}$, $\{IYYZZXX(4), XZYXIYZ(2), YXZIYXZ(3)\}$, $\{XZYXIYZ(2), ZIZYXYX(5), YXZIYXZ(3)\}$, $\{XZYXIYZ(2), YZXZXIY(7), YXZIYXZ(3)\}$, $\{IYYZZXX(4), XZYXIYZ(2), ZIZYXYX(5)\}$, $\{IYYZZXX(4), XZYXIYZ(2), ZYXXYZI(6)\}$, $\{IYYZZXX(4), XZYXIYZ(2), YZXZXIY(7)\}$, $\{XZYXIYZ(2), ZIZYXYX(5), ZYXXYZI(6)\}$, $\{XZYXIYZ(2), ZYXXYZI(6), YZXZXIY(7)\}$, $\{IYYZZXX(4), ZIZYXYX(5), YXZIYXZ(3)\}$, $\{IYYZZXX(4), ZYXXYZI(6), YXZIYXZ(3)\}$, $\{ZIZYXYX(5), ZYXXYZI(6), YXZIYXZ(3)\}$, $\{YZXZXIY(7), ZIZYXYX(5), YXZIYXZ(3)\}$, $\{YZXZXIY(7), ZYXXYZI(6), YXZIYXZ(3)\}$, $\{IYYZZXX(4), YZXZXIY(7), ZIZYXYX(5)\}$, $\{IYYZZXX(4), YZXZXIY(7), ZYXXYZI(6)\}$, $\{YZXZXIY(7), ZIZYXYX(5), ZYXXYZI(6)\}$	$\{IYYZZXX, XXIYZZY, XZYXIYZ, YXZIYXZ, YZXZXIY, ZIZYXYX, ZYXXYZI, IIIIIII\}$
$\{XXIYZZY(1), ZXYXYIZ(3), XYZIXYZ(2)\}$, $\{XXIYZZY(1), XYZIXYZ(2), YIYZXZX(5)\}$, $\{XXIYZZY(1), YZXXZYI(6), XYZIXYZ(2)\}$, $\{XXIYZZY(1), XYZIXYZ(2), ZYXZIXY(7)\}$, $\{IZZYYXX(4), XXIYZZY(1), ZXYXYIZ(3)\}$, $\{XXIYZZY(1), ZXYXYIZ(3), YZXXZYI(6)\}$, $\{XXIYZZY(1), ZXYXYIZ(3), ZYXZIXY(7)\}$, $\{IZZYYXX(4), XXIYZZY(1), YIYZXZX(5)\}$, $\{IZZYYXX(4), XXIYZZY(1), YZXXZYI(6)\}$, $\{IZZYYXX(4), XXIYZZY(1), ZYXZIXY(7)\}$, $\{XXIYZZY(1), YZXXZYI(6), YIYZXZX(5)\}$, $\{XXIYZZY(1), ZYXZIXY(7), YIYZXZX(5)\}$, $\{IZZYYXX(4), ZXYXYIZ(3), XYZIXYZ(2)\}$, $\{ZXYXYIZ(3), XYZIXYZ(2), YIYZXZX(5)\}$, $\{ZYXZIXY(7), ZXYXYIZ(3), XYZIXYZ(2)\}$, $\{IZZYYXX(4), XYZIXYZ(2), YIYZXZX(5)\}$, $\{IZZYYXX(4), YZXXZYI(6), XYZIXYZ(2)\}$, $\{IZZYYXX(4), ZYXZIXY(7), XYZIXYZ(2)\}$, $\{YZXXZYI(6), XYZIXYZ(2), YIYZXZX(5)\}$, $\{ZYXZIXY(7), YZXXZYI(6), XYZIXYZ(2)\}$, $\{IZZYYXX(4), ZXYXYIZ(3), YIYZXZX(5)\}$, $\{IZZYYXX(4), ZXYXYIZ(3), YZXXZYI(6)\}$, $\{ZXYXYIZ(3), YZXXZYI(6), YIYZXZX(5)\}$, $\{ZYXZIXY(7), ZXYXYIZ(3), YIYZXZX(5)\}$, $\{ZYXZIXY(7), ZXYXYIZ(3), YZXXZYI(6)\}$, $\{IZZYYXX(4), ZYXZIXY(7), YIYZXZX(5)\}$, $\{IZZYYXX(4), ZYXZIXY(7), YZXXZYI(6)\}$, $\{ZYXZIXY(7), YZXXZYI(6), YIYZXZX(5)\}$	$\{IZZYYXX, XXIYZZY, XYZIXYZ, YIYZXZX, YZXXZYI, ZXYXYIZ, ZYXZIXY, IIIIIII\}$
$\{XXIYZZY(1), XYZIXYZ(2), YIYXZXZ(3)\}$, $\{XXIYZZY(1), ZXYZIYX(5), XYZIXYZ(2)\}$, $\{XXIYZZY(1), XYZIXYZ(2), ZYXXYZI(6)\}$, $\{XXIYZZY(1), YZXZXIY(7), XYZIXYZ(2)\}$, $\{IZZYYXX(4), XXIYZZY(1), YIYXZXZ(3)\}$, $\{XXIYZZY(1), ZYXXYZI(6), YIYXZXZ(3)\}$, $\{XXIYZZY(1), YZXZXIY(7), YIYXZXZ(3)\}$, $\{IZZYYXX(4), XXIYZZY(1), ZXYZIYX(5)\}$, $\{IZZYYXX(4), XXIYZZY(1), ZYXXYZI(6)\}$, $\{IZZYYXX(4), XXIYZZY(1), YZXZXIY(7)\}$, $\{XXIYZZY(1), ZXYZIYX(5), ZYXXYZI(6)\}$, $\{XXIYZZY(1), ZXYZIYX(5), YZXZXIY(7)\}$, $\{IZZYYXX(4), XYZIXYZ(2), YIYXZXZ(3)\}$, $\{ZXYZIYX(5), XYZIXYZ(2), YIYXZXZ(3)\}$, $\{YZXZXIY(7), XYZIXYZ(2), YIYXZXZ(3)\}$, $\{IZZYYXX(4), ZXYZIYX(5), XYZIXYZ(2)\}$, $\{IZZYYXX(4), XYZIXYZ(2), ZYXXYZI(6)\}$, $\{IZZYYXX(4), YZXZXIY(7), XYZIXYZ(2)\}$, $\{ZXYZIYX(5), XYZIXYZ(2), ZYXXYZI(6)\}$, $\{YZXZXIY(7), XYZIXYZ(2), ZYXXYZI(6)\}$, $\{IZZYYXX(4), ZXYZIYX(5), YIYXZXZ(3)\}$, $\{IZZYYXX(4), ZYXXYZI(6), YIYXZXZ(3)\}$, $\{ZXYZIYX(5), ZYXXYZI(6), YIYXZXZ(3)\}$, $\{ZXYZIYX(5), YZXZXIY(7), YIYXZXZ(3)\}$, $\{YZXZXIY(7), ZYXXYZI(6), YIYXZXZ(3)\}$, $\{IZZYYXX(4), ZXYZIYX(5), YZXZXIY(7)\}$, $\{IZZYYXX(4), YZXZXIY(7), ZYXXYZI(6)\}$, $\{ZXYZIYX(5), ZYXXYZI(6), YZXZXIY(7)\}$	$\{IZZYYXX, XXIYZZY, XYZIXYZ, YIYXZXZ, YZXZXIY, ZXYZIYX, ZYXXYZI, IIIIIII\}$

Table G.16: Continued from Tables G.1-G.15.