Editors' Suggestion

Fault-tolerant quantum error correction using error weight parities

Theerapat Tansuwannont [1,*] and Debbie Leung [2,3,†]

[1] *Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada*

[2] *Institute for Quantum Computing and Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada*

[3] *Perimeter Institute for Theoretical Physics, Waterloo, Ontario, N2L 2Y5, Canada*

(Received 16 June 2021; accepted 31 August 2021; published 7 October 2021)

In quantum error correction using imperfect primitives, errors of high weight arising from a few faults are major concerns since they might not be correctable by the quantum error correcting code. Fortunately, some errors of different weights are logically equivalent, and the same correction procedure is applicable to all equivalent errors; thus correcting high-weight errors is sometimes possible. In this work, we introduce a technique called weight parity error correction (WPEC) which can correct Pauli error of any weight in some stabilizer codes provided that the parity of the weight of the error is known. We show that the technique is applicable to concatenated codes constructed from the $[\![7, 1, 3]\!]$ Steane code or the $[\![23, 1, 7]\!]$ Golay code. We also provide a fault-tolerant error correction protocol using WPEC for the $[\![49, 1, 9]\!]$ concatenated Steane code which can correct up to three faults and requires only two ancillas.

DOI: 10.1103/PhysRevA.104.042410

I. INTRODUCTION

One crucial component for large-scale quantum computers is fault-tolerant error correction (FTEC), which suppresses error propagation throughout the circuits. An arbitrarily small logical error rate can be achieved through code concatenation, given that the physical error rate is below some constant threshold value [1–5]. However, increasing overheads are needed for decreasing logical error rate [6–9]. Conventional FTEC schemes require many ancillas during error syndrome measurements. For example, the Shor-style [1,10] and the Knill-style [11] error corrections, which apply to any stabilizer code, require as many ancillas as the maximum weight of the stabilizer generators and twice the blocklength, respectively. Steane-style error correction [12,13] which applies to any CSS code requires one code block of ancillas.

Recently, several FTEC schemes that use only a few ancillas and are applicable to the $[\![7, 1, 3]\!]$ Steane code [14] have been proposed. The scheme due to Yoder and Kim for the $[\![7, 1, 3]\!]$ code uses two ancillas (nine qubits in total) [15]. Their ideas are further developed to a "flag FTEC" scheme which, for the $[\![7, 1, 3]\!]$ code, also uses two ancillas [16]. A flag FTEC scheme for any $[\![n, k, d]\!]$ stabilizer code requires $d + 1$ ancillas [17], where the schemes for some specific code families may require fewer [16,18–21]. The flag technique that uses a few ancillas to detect high-weight errors can also be applied to various fault-tolerant schemes [22–32]. Another FTEC scheme applicable to the $[\![7, 1, 3]\!]$ code was proposed by Reichardt; the scheme extracts several syndrome bits at once and requires no ancillas, provided that there are at least two code blocks (so at least 14 qubits are required in total) [33].

In order to achieve an arbitrarily low error rate through code concatenation, the FTEC scheme used with the code must be modified accordingly. One way to do this is replacing all physical qubits with code blocks and replacing all physical gates with corresponding logical gates [5]. For the $[\![7, 1, 3]\!]$ code, each qubit (including each ancilla qubit) required in an FTEC scheme will become a block of seven physical qubits in the modified scheme. Following this modification, the schemes in [15,16] applied to the $[\![49, 1, 9]\!]$ concatenated Steane code will require 63 qubits in total. Meanwhile, the scheme in [33] requires 98 qubits in total, encoding two logical qubits. Note that the maximum weight for the stabilizer generators increases quickly with concatenation. These difficulties motivate our main question: how to reduce the number of ancillas required for an FTEC scheme for a concatenated code?

In this paper, we introduce a technique called weight parity error correction (WPEC) and construct an FTEC scheme for the $[\![49, 1, 9]\!]$ concatenated Steane code using only *two* ancilla qubits. The scheme relies on the fact that, for the $[\![7, 1, 3]\!]$ code, errors with the same syndrome and weight parity differ by the multiplication of some stabilizer; these errors are thus logically equivalent and need not be distinguished from one another. Therefore, error correction on each subblock of seven qubits in the $[\![49, 1, 9]\!]$ code can be accomplished using only two ingredients: the error syndrome and the weight parity of error in each subblock. Most importantly, the weight parity for each subblock of seven qubits in the $[\![49, 1, 9]\!]$ code can be obtained from the full syndrome measurement. Using this idea in conjunction with two ancilla qubits, our FTEC protocol for the $[\![49, 1, 9]\!]$ code can correct up to three faults. As a result, our protocol can suppress the error rate from p to $O(p^4)$ using 51 qubits in total.

*ttansuwannont@uwaterloo.ca

†wcleung@uwaterloo.ca

 ©2021 American Physical Society

The paper is organized as follows: In Sec. II we observe the aforementioned equivalence between errors of any weight with the same syndrome and weight parity, and describe WPEC. In Sec. III we provide sufficient conditions for WPEC, and then we provide syndrome extraction circuits and an FTEC protocol for the $[\![49, 1, 9]\!]$ concatenated Steane code using only two ancilla qubits. In Sec. IV WPEC is extended to the $[\![23, 1, 7]\!]$ Golay code and concatenated Steane codes with more than two levels of concatenation. Last, we discuss our results and directions for future works in Sec. V.

II. WEIGHT PARITY ERROR CORRECTION FOR THE STEANE CODE

The Steane code [14], also known as the $[\![7, 1, 3]\!]$ code, is a quantum error correcting code that encodes one logical qubit into seven physical qubits and can correct any error on up to one qubit. It has several desirable properties for fault-tolerant quantum computation, e.g., logical Clifford operations are transversal [1]. The Steane code is a code in the Calderbank-Shor-Steane (CSS) code family [14,34] where X- and Z-type errors can be detected and corrected separately. The Steane code in the stabilizer formalism can be constructed from the parity check matrix of the classical [7,4,3] Hamming code through the CSS construction [35]. In addition, it is known that any classical Hamming code can be rearranged into a cyclic code, a binary linear code in which any cyclic shift of a codeword is also a codeword [36]. We can describe the Steane code in cyclic form with the following stabilizer generators:

$$\begin{aligned} g_1^x &: X\,I\,X\,X\,X\,I\,I, & g_1^z &: Z\,I\,Z\,Z\,Z\,I\,I,\\ g_2^x &: I\,X\,I\,X\,X\,X\,I, & g_2^z &: I\,Z\,I\,Z\,Z\,Z\,I,\\ g_3^x &: I\,I\,X\,I\,X\,X\,X, & g_3^z &: I\,I\,Z\,I\,Z\,Z\,Z. \end{aligned} \tag{1}$$

The generators of a stabilizer code define not only the codespace, but also the measurements that give rise to the error syndrome. When these measurements are imperfect, different sets of generators for the same code can have different fault-tolerant properties. The use of the Steane code in cyclic form gives some advantages in distinguishing high-weight errors in consecutive form [19] (see Sec. V for more details). We can choose the logical X and logical Z operators to be $X^{\otimes 7}S$ and $Z^{\otimes 7}T$ for any stabilizer operators S, T. With this convention, we state the following crucial property of the Steane code that goes into our construction.

Fact 1. Let M be any Z-type operator (a tensor product of Is and Zs) defined on seven qubits. Suppose M commutes with all X-type generators of the $[\![7, 1, 3]\!]$ code. If M has even weight, then it is a logical I; otherwise, if M has odd weight, then it is a logical Z.

Proof. Because M is a Z-type operator that commutes with all X-type generators, M is either a stabilizer of Z type or a logical Z operator. Let E_1 and E_2 be Z-type operators with weights w_1 and w_2. Then E_1E_2 is an operator of weight $w_1 + w_2 - 2c$, where c is the number of qubits supported by both E_1 and E_2. Observe that all stabilizer generators of the Steane code have even weight, and a multiplication of two operators with even weight always gives an operator with even weight. Thus, all stabilizers of Z type (which are logical I operators) have even weight. Moreover, a Z-type operator which is a logical Z operator is of the form $Z^{\otimes 7}T$ where T is a stabilizer of Z type. Therefore, all logical Z operators of Z type have odd weight. ■

For a Pauli error E on a block of seven qubits, the syndrome is a six-bit string denoted by $s(E) = (s_x|s_z)$ where $s_x, s_z \in \mathbb{Z}_2^3$. The ith bit of s_x (or s_z) is 0 if E commutes with g_i^x (or g_i^z), and 1 if E anticommutes with g_i^x (g_i^z). If E occurs to a codeword of the Steane code, $s(E)$ corresponds to the outcomes of measuring the six generators (0 and 1 correspond to $+1$ and -1 outcomes, respectively). The Steane code is a *perfect CSS code of distance 3* meaning that for each s_x, $(s_x|000)$ is the syndrome of a *unique* weight-1 Z-type error, which we denote as $E^z_{\mathrm{wt}-1}(s_x)$, and similarly each $(000|s_z)$ is the syndrome of a unique X-type error.[1] For CSS codes, the X- and Z-type error corrections are independent of one another. Furthermore, we focus on CSS codes in which X- and Z-type generators have the same form, and the same method applies to both types of error correction. So we focus on Z errors for simplicity. Since Z-type errors have trivial s_z, we focus on s_x from now on.

With the above notations, consider the following simple error correction procedure on the Steane code: if the syndrome is $(s_x|000)$, do nothing if s_x is trivial, apply $E^z_{\mathrm{wt}-1}(s_x)$ otherwise. We observe that if the syndrome is caused by a Z-type error, then the procedure outputs the encoded data transformed by a logical I or logical Z. This is because the actual Z-type error combined with the correction remains Z-type and commutes with all of $g^x_{1,2,3}$, so the conclusion follows from Fact 1.

If a codeword is corrupted by an arbitrary Z-type error E, the above procedure always recovers the codeword, but sometimes with an undesirable logical Z error. The technique of weight parity error correction, to be developed next, is a revised procedure that will *always* correct the error E, but it requires knowing whether E has odd or even weight. Measuring the error weight parity should not be done on a single layer of Steane code since it measures a logical operator on the Steane code. Fortunately, the parity information can be safely learnt for the constituent blocks when we concatenate the Steane code with itself. We will describe these ideas in detail in the rest of this section, and apply them for fault-tolerant error correction in the next section.

First, we use Fact 1 to show that Z-type errors with the same syndrome *and* the same weight parity (whether odd or even) differ by the multiplication of some stabilizer.

Claim 1. Logical equivalence of errors with the same syndrome and weight parity for the $[\![7, 1, 3]\!]$ code

Suppose E_1, E_2 are arbitrary Z-type errors (of any weights) on the $[\![7, 1, 3]\!]$ code with the same syndrome. Then E_1 and E_2 have the same weight parity iff $E_1 = E_2S$ for some stabilizer S.

Proof. Let w_1, w_2 be the weights of E_1, E_2, respectively. Let $N = E_1E_2$ (so $E_2 = E_1N$ as $E_1 = E_1^\dagger$). The weight of N is equal to $w_1 + w_2 - 2c$ where c is the number of qubits supported by both E_1 and E_2. As N commutes with all of

[1]This is from the fact that the $[\![7, 1, 3]\!]$ Steane code can be constructed from the classical [7,4,3] Hamming code, which is a *classical perfect code*, a code which saturates the classical Hamming bound [36].

$g^x_{1,2,3}$, from Fact 1, N is a logical I if and only if $w_1 + w_2 - 2c$ is even (when E_1 and E_2 have the same weight parity). ■

Second, we use Claim 1 to provide a method for error correction of Z-type error of arbitrary weight on the Steane code, *if the weight parity of the error is known*.

Definition 1. Weight parity error correction (WPEC) for the $[\![7, 1, 3]\!]$ code

Suppose a Z-type error E occurs to a codeword of the $[\![7, 1, 3]\!]$ code. Let s_x and w be the syndrome and the weight of E, $E^z_{\mathrm{wt}-1}(s_x)$ be the weight-1 Z-type operator with syndrome s_x, and $E^z_{\mathrm{wt}-2}(s_x)$ be any weight-2 Z-type operator with syndrome s_x, respectively. The following procedure is called *weight parity error correction* (WPEC):

(1) If s_x is trivial, do nothing if w is even, or apply any logical Z if w is odd.

(2) If s_x is nontrivial, apply $E^z_{\mathrm{wt}-1}(s_x)$ if w is odd, or apply $E^z_{\mathrm{wt}-2}(s_x)$ if w is even.

WPEC always returns the original codewords because in each case, the error E and the correction operation have the same syndrome and weight parity, so by Claim 1, the correction is logically equivalent to E.

WPEC allows us to correct high-weight errors in the Steane code, but we need to know the weight parity of the error. The weight parity of a Z-type error is the outcome of measuring $X^{\otimes 7}$, so learning the weight parity is equivalent to a logical X measurement, which can destroy the superposition of the logical state. Fortunately, there is a setting in which the weight parity can be obtained without affecting the encoded data. Consider code concatenation in which each qubit of an error correcting code $\mathcal{C}_2$ is encoded into another quantum error correcting code $\mathcal{C}_1$. If $\mathcal{C}_1$ is chosen to be the Steane code, the weight parity of each codeblock can potentially be learnt from the syndrome of $\mathcal{C}_2$. We will develop WPEC for the concatenated Steane code in the rest of this section and show the advantage in the context of fault tolerance in the next section.

Consider code concatenation using two Steane codes in cyclic form. The resulting code which is a $[\![49, 1, 9]\!]$ code can be described by 48 stabilizer generators. The first group of 42 generators, called first-level generators, have the form $g^x_i \otimes I^{\otimes 42}$, $g^z_i \otimes I^{\otimes 42}$, $I^{\otimes 7} \otimes g^x_i \otimes I^{\otimes 35}$, $I^{\otimes 7} \otimes g^z_i \otimes I^{\otimes 35}, \ldots, I^{\otimes 42} \otimes g^x_i, I^{\otimes 42} \otimes g^z_i$ for $i = 1, 2, 3$. The remaining six of these generators, called second-level generators, have the form

$$\begin{aligned} \tilde{g}^x_1 &: \mathbf{X\,I\,X\,X\,X\,I\,I}, & \tilde{g}^z_1 &: \mathbf{Z\,I\,Z\,Z\,Z\,I\,I}, \\ \tilde{g}^x_2 &: \mathbf{I\,X\,I\,X\,X\,X\,I}, & \tilde{g}^z_2 &: \mathbf{I\,Z\,I\,Z\,Z\,Z\,I}, \\ \tilde{g}^x_3 &: \mathbf{I\,I\,X\,I\,X\,X\,X}, & \tilde{g}^z_3 &: \mathbf{I\,I\,Z\,I\,Z\,Z\,Z}, \end{aligned} \tag{2}$$

where $\mathbf{I} = I^{\otimes 7}$, $\mathbf{X} = X^{\otimes 7}$, and $\mathbf{Z} = Z^{\otimes 7}$. The logical X and logical Z operators can be chosen to be $\bar{X} = X^{\otimes 49}S$ and $\bar{Z} = Z^{\otimes 49}T$ for any stabilizer operators S, T. Relevant parts of the error syndrome corresponding to the first- and the second-level generators will be called first- and second-level syndromes, respectively.

Let us consider error correction on the $[\![49, 1, 9]\!]$ code and assume for now that error syndromes are reliable (which can be obtained from repetitive measurements). First, consider a simple motivating example, and suppose that a Z-type error E acts nontrivially on at most one subblock of seven-qubit code. In order to perform WPEC, the weight parity of E and the subblock in which E occurs must be known. Suppose that E has nontrivial first-level syndrome. The subblock in which E occurs is actually the subblock whose corresponding first-level syndrome is nontrivial, while the weight parity of E is a measurement result from a second-level generator which acts nontrivially on that subblock (note that the second-level generator must *act nontrivially on all qubits* in such a subblock; thus a choice of second-level generators is important). Now, suppose that E has trivial first-level syndrome. The subblock in which E occurs can no longer be identified by the first-level syndrome. Fortunately, since the second-level Steane code ($\mathcal{C}_2$) is a distance-3 code, it can identify if any of the seven subblocks of $[\![7, 1, 3]\!]$ code (the $\mathcal{C}_1$ subblocks) has a Z-type error logically equivalent to $Z^{\otimes 7}$, thus providing the weight parity for each subblock of $[\![7, 1, 3]\!]$ code. That is, if E has trivial first-level syndrome and its weight is odd, the weight parity of E and the subblock in which E occurs can be determined using only the second-level syndrome. (If E has trivial first-level syndrome and has even weight, it is a stabilizer and no error correction is needed.)

If the Z-type error is more general and may act on multiple subblocks, the second-level syndrome may not provide the weight parities of the subblocks. Instead, we consider only Z-type errors that arise from a small number of faults in specially designed generator measurements. We will show that for these errors, the weight parity for each subblock can be determined by the second-level syndrome along with the information whether each subblock has trivial first-level syndrome or not.

In particular, let *block parity* $p_x \in \mathbb{Z}_2^7$ be a bitstring, where each bit is the weight parity of the Z error in one subblock, and 0 and 1 represent even and odd weights, respectively. Also, define the *triviality* of a subblock to be 0 or 1 if the subblock has trivial or nontrivial first-level syndrome, and let *block triviality* $\tau_x \in \mathbb{Z}_2^7$ be a seven-bit string in which the ith bit represents the triviality of the ith subblock. If the block parity can be accurately determined using the second-level syndrome together with the block triviality (we will elaborate how this can be done later), then we can blockwisely perform WPEC as described in Definition 1 by using the first-level syndrome and the weight parity of each subblock.

In this work, we develop an FTEC protocol that uses WPEC to correct high-weight errors arising from up to three faults. As an example, consider the measurement of $\tilde{g}^z_1$ using the circuit depicted in Fig. 1(a). Here we assume that a fault from any two-qubit gate can cause any two-qubit Pauli errors on the qubits where the gate acts nontrivially, and X- and Z-type errors can be detected separately. Thus, we may assume that high-weight errors arising from a single controlled-NOT (CNOT) fault is of the form **PIZZZII**, **IIPZZII**, **IIIPZII**, or **IIIIPII**, where $\mathbf{Z} = Z^{\otimes 7}$, $\mathbf{P} = I^{\otimes 7-m} \otimes Z^{\otimes m}$, and $m \in \{1, \ldots, 7\}$ (see the analysis of possible errors in [19] for more details). It is not hard to find second-level syndrome, block triviality, and block parity corresponding to each possible error. For example, error **PIZZZII** with $m = 6$ anticommutes with g^x_1 and $\tilde{g}^x_1$, and commutes with the other generators. Thus, its corresponding second-level syndrome, block triviality, and block parity are (1, 0, 0), (1, 0, 0, 0, 0, 0, 0), and (0,0,1,1,1,0,0), respectively. Table I displays all possible high-weight errors arising from a single fault during $\tilde{g}^z_1$ measurement and their corresponding

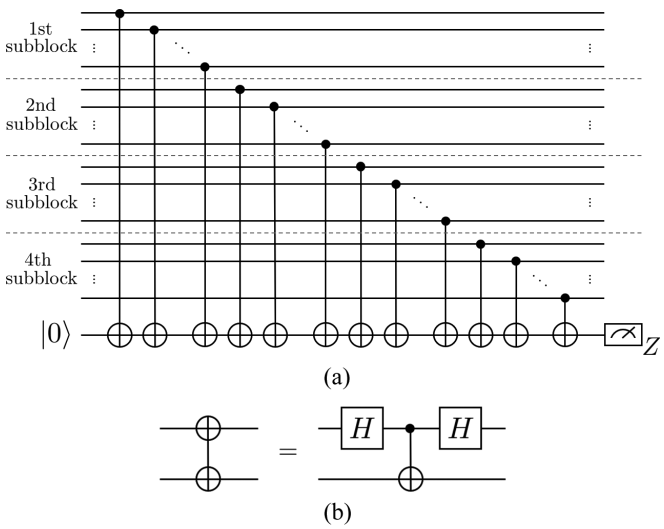


FIG. 1. (a) An example of circuit for measuring generator $\tilde{g}_1^z =$ **ZIZZZII**. Here we display only the subblocks in which the operator acts nontrivially (the first, second, third, and fourth subblocks in the figure correspond to the first, third, fourth, and fifth subblocks of $\tilde{g}_1^z$). A circuit for measuring X-type operator such as $\tilde{g}_1^x =$ **XIXXXII** can be obtained by replacing each controlled-NOT (CNOT) gate in (a) with the gate illustrated in (b).

second-level syndrome, block triviality, and block parity. Note that except for the first and the last row (with errors differing by multiplication of a stabilizer), each row has a unique combination of second-level syndrome and block triviality, so the block parity can be determined from the table. Since the second-level syndrome and the block triviality can in turn be obtained from the generator measurements, all possible errors arising from a single CNOT fault during the measurement of $\tilde{g}_1^z$ can be corrected using WPEC. In addition, observe that **ZIZZZII** and $\mathbf{I}^{\otimes 7}$ are equivalent up to a multiplication of $\tilde{g}_1^z$ but their block parities are different. Here we can see that multiplying an error with some second-level generators may change its block parity, but its second-level syndrome and block triviality (which is deduced from its first-level syndrome) remain the same. In this case, WPEC is still applicable. We say that block parities are equivalent whenever they can be transformed to one another by multiplying the corresponding errors with some stabilizer.

TABLE I. All possible forms of data errors arising from a single fault occurred during syndrome measurement using a circuit in Fig. 1(a) (where $\mathbf{P} = I^{\otimes 7-m} \otimes Z^{\otimes m}$). The block parity corresponding to each form of error can be determined by the second-level syndrome and the block triviality obtained from a full syndrome measurement. By knowing the block parity, high-weight errors can be corrected using WPEC.

Form of error	m	Second-level syndrome	Block triviality	Block parity
PIZZZII	7	(0,0,0)	(0,0,0,0,0,0,0)	(1,0,1,1,1,0,0)
	2,4,6	(1,0,0)	(1,0,0,0,0,0,0)	(0,0,1,1,1,0,0)
	1,3,5	(0,0,0)	(1,0,0,0,0,0,0)	(1,0,1,1,1,0,0)
IIPZZII	7	(1,0,0)	(0,0,0,0,0,0,0)	(0,0,1,1,1,0,0)
	2,4,6	(0,0,1)	(0,0,1,0,0,0,0)	(0,0,0,1,1,0,0)
	1,3,5	(1,0,0)	(0,0,1,0,0,0,0)	(0,0,1,1,1,0,0)
IIIPZII	7	(0,0,1)	(0,0,0,0,0,0,0)	(0,0,0,1,1,0,0)
	2,4,6	(1,1,1)	(0,0,0,1,0,0,0)	(0,0,0,0,1,0,0)
	1,3,5	(0,0,1)	(0,0,0,1,0,0,0)	(0,0,0,1,1,0,0)
IIIIPII	7	(1,1,1)	(0,0,0,0,0,0,0)	(0,0,0,0,1,0,0)
	2,4,6	(0,0,0)	(0,0,0,0,1,0,0)	(0,0,0,0,0,0,0)
	1,3,5	(1,1,1)	(0,0,0,0,1,0,0)	(0,0,0,0,1,0,0)
$\mathbf{I}^{\otimes 7}$	–	(0,0,0)	(0,0,0,0,0,0,0)	(0,0,0,0,0,0,0)

In an actual fault-tolerant protocol, we want to distinguish all possible high-weight errors arising from various types of faults up to three faults, including any gate faults, faults during the preparation and measurement of ancilla qubits, and faults during wait time. The circuit construction in Fig. 1(a), however, might not cause errors that can be distinguished. Note that possible errors arising from CNOT faults heavily depend on the ordering of CNOT gates being used in the measurement circuit. In Sec. III we will discuss conditions in which WPEC can be applied. We will also provide a family of circuits with specific CNOT ordering and an FTEC protocol for the $[\![49, 1, 9]\!]$ code which can correct high-weight errors arising from up to three faults.

III. FAULT-TOLERANT ERROR CORRECTION PROTOCOL FOR THE $[\![49, 1, 9]\!]$ CODE

Fault-tolerant error correction is one of the most essential gadgets for constructing large-scale quantum computers. Every FTEC protocol must satisfy the following conditions.

Definition 2. Fault-tolerant error correction [5]

For $t = \lfloor (d-1)/2 \rfloor$, an error correction protocol using a distance-d stabilizer code is *t-fault tolerant* if the following two conditions are satisfied:

(1) For any input codeword with error of weight v_1, if v_2 faults occur during the protocol with $v_1 + v_2 \leqslant t$, ideally decoding the output state gives the same codeword as ideally decoding the input state.

(2) If v faults happen during the protocol with $v \leqslant t$, no matter how many errors are present in the input state, the output state differs from any valid codeword by an error of at most weight v.

In this work, we develop an FTEC protocol for the $[\![49, 1, 9]\!]$ code that can correct up to three faults. The circuits for measuring first- and second-level generators are shown in Fig. 2. The types of faults being considered include faults that happen to the physical gates, faults during the preparation and measurement of ancilla qubits in the circuits, and faults during wait time.

Let *fault combination* be a collection of faults up to three faults (which may be of different types and can cause errors of weight much higher than three on the data qubits). Our goal is to distinguish all fault combinations that can be confusing and may cause WPEC to fail. Similar to an example of WPEC in Sec. II, we can categorize all possible fault combinations into subsets by their second-level syndrome and block triviality. The following sufficient condition can determine when the WPEC technique can be applied.

Claim 2. Sufficient condition for WPEC

Let $\mathcal{F}$ be the set of all possible fault combinations during an FTEC protocol for the $[\![49, 1, 9]\!]$ code and let $\mathcal{F}_k \subseteq \mathcal{F}$

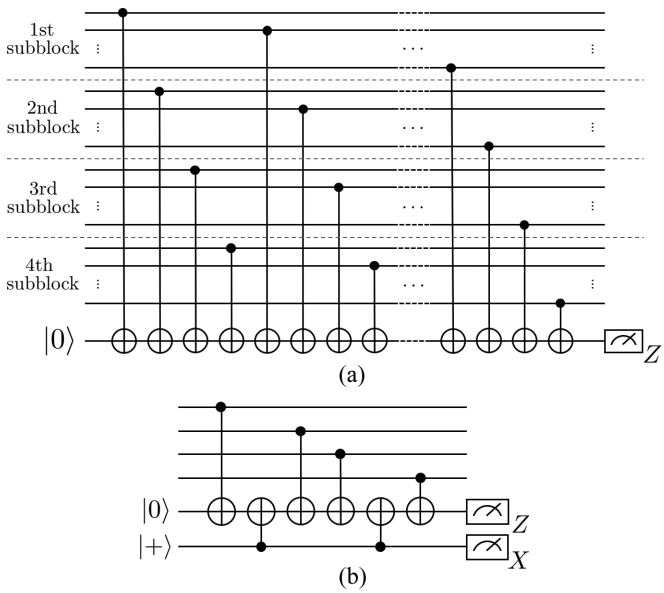


FIG. 2. Circuits for measuring second- and first-level generators being used in this work are shown in (a) and (b), respectively. With this gate permutation, the block parity corresponding to every possible high-weight error arising from up to three faults can be accurately determined. As such, our protocol can correct up to three faults.

be a subset of fault combinations with the same second-level syndrome and the same block triviality (where $\bigcup_k \mathcal{F}_k = \mathcal{F}$). WPEC is applicable in the FTEC protocol if each $\mathcal{F}_k$ satisfies one of the following conditions:

(1) Data errors from all fault combinations in $\mathcal{F}_k$ give equivalent block parities.

(2) Not every data error from a fault combination in $\mathcal{F}_k$ gives the same block parity (or its equivalence), but for each pair of fault combinations in $\mathcal{F}_k$ whose block parities of their data errors are not equivalent, their first-level syndromes or flag measurement results (or both) are different.

Proof. Whenever subset $\mathcal{F}_k$ satisfies the first condition in Claim 2, we can find a block parity that works for all fault combinations in $\mathcal{F}_k$ using only the second-level syndrome and the block triviality. A correction operator for each fault combination can be found following the definition of WPEC (Definition 1) using the first-level syndrome and the block parity. On the other hand, if $\mathcal{F}_k$ satisfies the second condition in Claim 2, a block parity cannot be accurately determined using only the second-level syndrome and the block triviality. Fortunately, with the assistance of the first-level syndrome and the flag measurement result, fault combinations that correspond to nonequivalent block parities can be distinguished and the block parity of each fault combination can be found. Similarly, a correction operator for each fault combination can be determined following Definition 1. ■

Whether possible fault combinations satisfy Claim 2 or not depends heavily on the ordering of the CNOT gates and the use of flag qubits in the circuits for syndrome measurements. In our FTEC protocol for the ⟦49, 1, 9⟧ code, the CNOT gates being used in the circuits for measuring second-level generator are applied in the following ordering:

$$(1, 8, 15, 22, 2, 9, 16, 23, \ldots, 7, 14, 21, 28), \qquad (3)$$

where the numbers 1 to 28 represent the qubits in which $\tilde{g}_i^z$ acts nontrivially. That is, CNOT gates are applied on the first qubit in each subblock for all subblocks, then on the second qubit in each subblock for all subblocks, and so on. The circuit for measuring $\tilde{g}_1^z$ is shown in Fig. 2(a). In addition, CNOT gates being used in the circuits for measuring first-level generator are in the normal ordering as shown in Fig. 2(b). Note that there is no flag qubit involved in the measurement of a second-level generator, and there is one flag qubit in the circuit for measuring a first-level generator.

Consider the case that there are some faults during Z-type generator measurements. Faulty circuits can produce nontrivial flag measurement results and cause error of any weight on the data qubits. Our goal is to detect and correct such an error using the flag measurement results from the faulty circuits, together with first- and second-level syndromes obtained from subsequent syndrome measurements. In particular, let the *flag vector* $\in \mathbb{Z}_2^{21}$ be a bitstring wherein each bit is the flag measurement result from each circuit for measuring g_i^z on each of the seven subblocks. We define the *cumulative flag vector* $f_x \in \mathbb{Z}_2^{21}$ to be the entry-wise sum of flag vectors (modulo 2) obtained from g_i^z measurements accumulated from the first round up until the current round of measurements (see the protocol described below for the definition of a round of measurements). Cumulative flag vector f_x together with first-level syndrome $s_x \in \mathbb{Z}_2^{21}$, second-level syndrome $\tilde{s}_x \in \mathbb{Z}_2^3$, and block triviality $\tau_x \in \mathbb{Z}_2^7$ from the latest round of measurements will be used for distinguishing all possible fault combinations that can occur during the syndrome measurements as described in Claim 2. Using the computer simulation described in Appendix A, we can verify that Claim 2 is satisfied when the number of input errors v_1 and the number of faults v_2 satisfy $v_1 + v_2 \leqslant 3$. A table of possible data errors and their corresponding s_x, $\tilde{s}_x$, τ_x, f_x, and block parity p_x (similar to Table I) can also be obtained from the simulation. Moreover, the subsets $\mathcal{F}_k$ can be deduced from this table (see Appendix A for more details).

Let the *outcome bundle* be the collection of first-level syndrome $s = (s_x|s_z)$, second-level syndrome $\tilde{s} = (\tilde{s}_x|\tilde{s}_z)$, block triviality $\tau = (\tau_x|\tau_z)$, and cumulative flag vector $f = (f_x|f_z)$ obtained during a single round of full syndrome measurement, where subscripts x and z denote results corresponding to X- and Z-type generator measurements. Using the simulation result together with the fact that X- and Z-type errors can be corrected separately, an FTEC protocol for the ⟦49, 1, 9⟧ code can be constructed as follows.

FTEC protocol for the ⟦49, 1, 9⟧ code

A full syndrome measurement, or a round of measurements, measure the generators in the following order: measure all $\tilde{g}_i^z$'s, then all $\tilde{g}_i^x$'s, then all g_i^z's, then all g_i^x's. Perform full syndrome measurements until the outcome bundles are repeated four times in a row. Afterwards, perform the following error correction procedure:

(1) Find the subset $\mathcal{F}_k$ corresponding to $\tilde{s}_x$ and τ_x from the table of possible errors (obtained from the simulation in Appendix A).

(a) If $\mathcal{F}_k$ satisfies Condition 1 in Claim 2, use a block parity of any fault combination in $\mathcal{F}_k$.

(b) If $\mathcal{F}_k$ satisfies Condition 2 in Claim 2, use a block parity of any combination in $\mathcal{F}_k$ that corresponds to s_x and f_x.

(c) If there is no $\mathcal{F}_k$ from the table of possible errors which corresponds to $\tilde{s}_x$ and τ_x, use the block parity with all 1's.

(2) Let $s_{x,i}$ be the first-level syndrome and w_i be the weight parity of the ith subblock. Apply Z-type error correction on each subblock as given by Definition 1. In particular:

(a) If $s_{x,i}$ is trivial, apply $ZZIZIII$ (logically equivalent to $Z^{\otimes 7}$) to the ith subblock when w_i is odd, or do nothing when w_i is even.

(b) If $s_{x,i}$ is nontrivial, apply $E^z_{\mathrm{wt}-1}(s_{x,i})$ to the ith subblock when w_i is odd, or apply $E^z_{\mathrm{wt}-2}(s_{x,i})$ when w_i is even.

(3) If there is no $\mathcal{F}_k$ from the table of possible errors which corresponds to $\tilde{s}_x$ and τ_x, further apply the following error correction procedure: find a Pauli operator from $\{\mathbf{ZIIIIII}, \mathbf{IZIIIII}, \ldots, \mathbf{IIIIIIZ}\}$ which corresponds to $\tilde{s}_x$, then apply such an operator (or its logically equivalent operator) to the data qubits.

(4) Repeat steps 1–3 but use $\tilde{s}_z$, s_z, τ_z, and f_z to deduce the X-type error correction ($E^x_{\mathrm{wt}-1}(s_{z,i})$, $E^x_{\mathrm{wt}-2}(s_{z,i})$, or XXIXIII) for each subblock.

Here we will assume that there are at most three faults during the protocol and the error is of Z type. The assumption on the number of faults guarantees that the outcome bundles must be repeated four times in a row within 16 rounds (the outcome bundle cannot keep changing forever since the number of faults is limited). To verify that the protocol above is three-fault tolerant, i.e., it satisfies the FTEC conditions in Definition 2 with $t = 3$ (the $[\![49, 1, 9]\!]$ code acts as a distance-7 code), first let us consider the case that there are no faults during the last round of full syndrome measurement. In this case, the outcome bundle corresponds to the actual data error. From the simulation in Appendix A, we know that whenever $v_1 + v_2 \leqslant 3$, one of the conditions in Claim 2 is satisfied and the block parity can be accurately determined. The operation in Step 2 will give the correct output state; thus both of the FTEC conditions are satisfied. On the other hand, if $v_1 + v_2 > 3$ but $v_2 \leqslant 3$, $\tilde{s}_x$ and τ_x may not correspond to any error in the table of possible errors. By using the block parity with all 1's, the operation in Step 2 will project the state in each subblock back to the subspace of the $[\![7, 1, 3]\!]$ code, where each subblock has an error equivalent to either $\mathbf{I}$ or $\mathbf{Z}$ after the operation. Afterwards, the operation in Step 3 will project the output state back to the subspace of the $[\![49, 1, 9]\!]$ code. Thus, the second condition in Definition 2 is satisfied.

Now let us consider the case that there are some faults during the last round of full syndrome measurement. The outcome bundle we obtained from the last round may not correspond to the data error since some errors arising during the last round may be undetectable. Since we perform full syndrome measurements until the outcome bundles are repeated four times in a row and there are at most three faults during the whole protocol, at least one of the last four rounds of full syndrome measurement must be correct. From the simulation result in Appendix A, the outcome bundle obtained from the last round (which is equal to that obtained from any correct round in the last four rounds) can definitely correct the error occurred before the last correct round. Here we want to verify that whenever the last four rounds have v faults (where $v \leqslant 3$), after the last round, the weight of the data error is increased by no more than v. This can be verified using the computer simulation described in Appendix B. By applying operation in Step 2 (and possibly Step 3) as previously discussed, the output state differs from a valid codeword by an error of weight at most v, regardless of the number of input errors. Thus, the second condition in Definition 2 is satisfied. Furthermore, whenever $v_1 + v_2 \leqslant 3$, we will obtain an output state which differs from a correct output state by an error of weight at most 3. Therefore, the first condition in Definition 2 is also satisfied.

The analysis for X-type errors is similar to that of Z-type errors. Note that during the measurement of Z-type generators, a single gate fault can cause an X-type error of weight 1 on the data qubits. This error can be considered as an input error for the measurement of X-type generators; thus the same analysis is applicable.

IV. WEIGHT PARITY ERROR CORRECTION FOR OTHER CODES

Besides the Steane code, we find that the WPEC technique is applicable to the $[\![23, 1, 7]\!]$ Golay code [6], which is a perfect CSS code of distance 7. The $[\![23, 1, 7]\!]$ Golay code can correct up to three errors and can be constructed from the parity check matrix of the classical [23,12,7] Golay code [36]. In cyclic form, the $[\![23, 1, 7]\!]$ Golay code can be constructed from the parity check matrix,

$$\begin{pmatrix}
1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0\\
0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0\\
0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0\\
0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1&0\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0&1
\end{pmatrix},$$

which can be generated from the check polynomial $h(x) = x^{12} + x^{10} + x^7 + x^4 + x^3 + x^2 + x + 1$ [36]. The ith Z-type (or X-type) generator of this code will be denoted as g_i^z (or g_i^x) where $i = 1, \ldots, 11$. The logical X and logical Z operators of this code can be chosen to be $\bar{X} = X^{\otimes 23}S$ and $\bar{Z} = Z^{\otimes 23}T$ for any stabilizer operators S, T.

Similar to the $[\![7, 1, 3]\!]$ code, we can prove the equivalence of errors with the same syndrome and the same weight parity as follows.

Claim 3. Logical equivalence of errors with the same syndrome for the $[\![23, 1, 7]\!]$ Golay code

Suppose E_1, E_2 are arbitrary Z-type errors (of any weights) on the $[\![23, 1, 7]\!]$ code with the same syndrome. Then E_1 and E_2 have the same weight parity iff $E_1 = E_2 S$ for some stabilizer S.

Proof. We can verify that every Z-type stabilizer in the stabilizer group of the $[\![23, 1, 7]\!]$ code has even weight, and every logical Z operator has odd weight. The rest of the proof follows the proof of Claim 1. ■

Let us consider Z-type error correction for the $[\![23, 1, 7]\!]$ code. Since the code is a perfect CSS code of distance 7, for each $s_x \in \mathbb{Z}_2^{11}$, $(s_x|0\ldots0)$ is the syndrome of a unique Z-type error of weight $\leqslant 3$. Suppose that a codeword is corrupted by a Z-type error with syndrome s_x. If we apply the minimal weight error correction corresponding to s_x, we sometimes obtain the codeword with undesirable logical Z operator. Fortunately, by knowing the weight parity of the error, the WPEC technique can be applied. The error correction procedure for the $[\![23, 1, 7]\!]$ code is defined as follows.

Definition 3. Weight parity error correction for the $[\![23, 1, 7]\!]$ Golay code

Suppose a Z-type error E occurs to a codeword of the $[\![23, 1, 7]\!]$ code. Let s_x and w be the syndrome and the weight of E, and let $E^z_{\min}(s_x)$ be the unique minimal weight error correction corresponding to the syndrome s_x. The following procedure is called *weight parity error correction* (WPEC):

(1) If $E^z_{\min}(s_x)$ has even weight (0 or 2), apply $E^z_{\min}(s_x)$ to the data qubits whenever w is even, or apply any Z-type operator P that has odd weight and corresponds to s_x to the data qubits whenever w is odd.

(2) If $E^z_{\min}(s_x)$ has odd weight (1 or 3), apply $E^z_{\min}(s_x)$ to the data qubits whenever w is odd, or apply any Z-type operator P that has even weight and corresponds to s_x to the data qubits whenever w is even.

Note that the $[\![23, 1, 7]\!]$ Golay code can be made cyclic; thus it can distinguish high-weight errors in consecutive form [19]. Claim 3 together with the cyclic property give us some possibilities to construct an FTEC protocol for the $[\![529, 1, 49]\!]$ concatenated Golay code in the same way as what we have done for the $[\![49, 1, 9]\!]$ code. We expect that our technique can lead to a protocol which can correct a large number of faults and will compare well with other FTEC schemes. To reach this goal, syndrome extraction circuits with appropriate permutation of gates (and possibly with flag qubits) must be found so that conditions similar to those in Claim 2 are satisfied. The search for such circuits with careful numerical verification of fault tolerance is a challenging and interesting future research direction.

The WPEC technique may also apply to the code obtained from concatenating the Steane code to the kth level, e.g., the $[\![7^k, 1, 3^k]\!]$ code. Since the kth-level Steane code is a distance-3 code, we expect that a block of errors in the $(k-1)$-th level can be determined and corrected using the syndrome and the block parity defined at the kth level. Again, however, appropriate syndrome extraction circuits must be found, which is beyond the scope of this work.

V. DISCUSSION AND CONCLUSIONS

In this work, we prove the logical equivalence between errors of any weight on seven qubits which have the same weight parity and correspond to the same error syndrome when error detection is performed by the $[\![7, 1, 3]\!]$ code in Claim 1. From this result, we introduce the WPEC technique in Definition 1, which can correct errors of any weight on seven qubits whenever their weight parity is known. We show that the WPEC technique can be extended to error correction in subblocks of the $[\![49, 1, 9]\!]$ code, and we prove the sufficient condition for WPEC in Claim 2. Afterwards, we provide a family of circuits and an FTEC protocol for the $[\![49, 1, 9]\!]$ code which can correct up to three faults. We also point out that the WPEC technique seems applicable to FTEC schemes for other codes such as the concatenated Golay code and concatenated Steane code with more than two levels of concatenation.

Since the FTEC protocol provided in this work satisfies the definition of FTEC in Definition 2 with $t = 3$, we can guarantee that the logical error rate is suppressed to $O(p^4)$ whenever the physical error rate is p under the random Pauli noise model. Note that we did not use the full ability of a code with distance 9, which, in principle, can correct up to four errors. In terms of error suppression, our FTEC protocol is as good as typical FTEC protocols for a concatenated code which are constructed by replacing each physical qubit with a code block and replacing each physical gate with the corresponding logical gate [5].

One major advantage of our FTEC protocol is that only two ancillas are needed: one ancilla for a syndrome measurement result and another ancilla for a flag measurement result (assuming that the qubit preparation and measurement are fast compared to the gate operation time). As a result, our protocol requires 51 qubits in total. The number of required qubits is very low compared to other FTEC protocols for the $[\![49, 1, 9]\!]$ code; the FTEC schemes in [15,16] extended to the $[\![49, 1, 9]\!]$ code require 63 qubits in total (the minimum number of required ancillas is 14 assuming that they are recyclable). Meanwhile, the FTEC protocol in [33] which extracts multiple syndromes at once encodes two logical qubits and requires no ancilla, but needs to work on two code blocks of 98 qubits in total. Our protocol might not have the fewest total number of qubits compared with other protocols for a different code which can correct up to three faults; for example, the flag FTEC protocol in [17] applying to the $[\![37, 1, 7]\!]$ 2D color code requires 45 qubits in total. Nevertheless, our work provides a substantial improvement over other FTEC protocols for a *concatenated* code, an approach that is still advantageous in some circumstances. Furthermore, the use of weight parities in error correction may be extended to other families of codes as well [37]. We believe that if the protocol requires fewer ancillas, the number of total locations will decrease,

which can result in higher accuracy threshold. However, a simulation with careful analysis is required for the accuracy threshold calculation; thus we leave this for future work.

The protocol in Sec. III which can correct up to three faults exploits two techniques: the flag technique which partitions set of possible errors using flag measurement results, and the WPEC technique which corrects errors of any weight using their syndromes and weight parities. It should be emphasized that flag ancillas are not necessarily required for a protocol exploiting WPEC technique; we find that a protocol which uses circuits similar to a circuit in Fig. 2(a) for second-level syndrome measurements and uses nonflag circuits for first-level measurements can correct up to two faults.

We point out that the permutation of CNOT gates in the syndrome extraction circuits that make the protocol satisfies Claim 2 is not unique. We choose the permutation in Eq. (3) by using the fact that a CSS code constructed from classical cyclic codes can distinguish high-weight errors in the consecutive form [19]. In particular, the circuit is designed in the way that high-weight errors arising in each subblock can be determined by the underlying $[\![7, 1, 3]\!]$ code in cyclic form. We did not prove the optimality of the choice of gate permutation in our protocol, so an FTEC protocol for the $[\![49, 1, 9]\!]$ code with only one ancilla or a protocol that can correct up to four faults might be possible.

Last, we note that the WPEC technique introduced in this work is not limited to the $[\![49, 1, 9]\!]$ code. In Sec. IV we prove the logical equivalence of errors with the same syndrome and weight parity for the $[\![23, 1, 7]\!]$ Golay code in Claim 3 and provide a WPEC scheme in Definition 3, which shows that WPEC can correct some high-weight errors in a subblock of the $[\![529, 1, 49]\!]$ concatenated Golay code. In addition, we expect that WPEC can be applied to any concatenated Steane code with more than two levels of concatenation in a similar fashion. However, circuits and a protocol must be carefully designed so that the full error correction ability of the code can be achieved. Another interesting future direction would be extending the WPEC technique to other families of quantum codes.

ACKNOWLEDGMENTS

We thank Christopher Chamberland, Rui Chao, Narayanan Rengaswamy, and Michael Vasmer for interesting comments and suggestions. T.T. acknowledges the support of The Queen Sirikit Scholarship under The Royal Patronage of Her Majesty Queen Sirikit of Thailand. D.L. is supported by an NSERC Discovery grant. The Perimeter Institute is supported in part by the Government of Canada and the Province of Ontario.

APPENDIX A: SIMULATION OF POSSIBLE FAULTS DURING THE FTEC PROTOCOL ASSUMING THAT THE LAST ROUND OF FULL SYNDROME MEASUREMENT HAS NO FAULTS

As discussed in Sec. III, in order to verify that the FTEC protocol for the $[\![49, 1, 9]\!]$ code satisfies the FTEC conditions in Definition 2, we consider two separate cases: the case that there are some faults during the last round of full syndrome measurement, and the case that there are not. In this section, we provide details of a simulation to show that whenever the number of faults is at most three and none of the faults occurs during the last round, all possible fault combinations satisfy Claim 2 and our protocol can correct errors on the data qubits.

In our protocol, we will perform full syndrome measurements until the outcome bundles are repeated four times in a row. Since there are at most three faults, the repetition condition will be satisfied within 16 rounds of full syndrome measurement. In this simulation, we assume that the last round of measurement has no faults; thus the high-weight error on the data qubits arising from at most three faults is accumulated from up to 15 rounds. We will use the outcome bundle (syndromes and flag vector) obtained from the last round to determine the fault combination that cause the error so that the corresponding weight parity can be found and the WPEC can be done.

We first define mathematical objects being used in our simulation. Let *fault* be an object with two associated variables: *Pauli error* defined on the code block of 49 qubits arising from the fault, and *flag vector* $\in \mathbb{Z}_2^{21}$ which indicates the flag measurement results associated with the fault. There are four types of possible faults: faults during wait time (denoted by W), faults arising from the measurement of first- and second-level generators (denoted by G_1 and G_2, respectively), and flag measurement faults (denoted by F). A *fault combination* can be constructed by combining faults of same or different types up to three faults, i.e., multiplying their Pauli errors and adding their flag vectors. The errors on the input codeword can be considered as wait time faults in which associated Pauli errors do not propagate to other data qubits during the FTEC protocol. In addition, the X-type errors on the data qubits arising from the faults during the measurement of Z-type generators can be considered as wait time faults during the measurement of subsequent X-type generators, in which our simulation is also applicable. (Since the last round of measurement has no faults, we can assume that the syndromes obtained from the last round are correct and the syndrome measurement faults can be neglected.)

Next, we define *fault set* as follows: for faults of type G_1 (or type G_2), we denote $F_{i,j}^{G_1}$ (or $F_{i,j'}^{G_2}$) to be sets of possible G_1 (or G_2) faults arising from a circuit for measuring g_j^z, $j = 1, \ldots, 21$ (or $\tilde{g}_{j'}^z$, $j' = 1, 2, 3$) where the number of faults is $i \in \{0, 1, 2, 3\}$ (g_j^z refers to the generator $g_{(j-1) \bmod 3+1}^z$ on the $\lceil j/3 \rceil$-th subblock). Also, we denote F_i^W and F_i^F to be sets of possible faults of type W and F, respectively, where the number of faults is $i \in \{0, 1, 2, 3\}$. In addition, we define *fault set combination* to be a set of fault sets up to three sets.

Last, let $v_{G_1}, v_{G_2}, v_W, v_F$ be the number of faults of type G_1, G_2, W, and F, respectively. $(v_{G_1}, v_{G_2}, v_W, v_F)$ that satisfies $v_{G_1} + v_{G_2} + v_W + v_F \leqslant 3$ is called *fault number combination*.

With the definitions of fault, fault combination, fault set, fault set combination, and fault number combination, now we are ready to describe the simulation.

Pseudocode for a simulation of possible faults assuming that the last round of full syndrome measurement has no faults

(1) Construct fault sets $F_{i,j}^{G_1}, F_{i,j'}^{G_2}, F_i^W$, and F_i^F for all $i = 0, 1, 2, 3$, $j = 1, \ldots, 21$, $j' = 1, 2, 3$.

(2) Construct all possible fault number combinations that satisfy $v_{G_1} + v_{G_2} + v_W + v_F \leqslant 3$.

(3) For each $(v_{G_1}, v_{G_2}, v_W, v_F)$, find all possible fault set combinations from $v_{G_1}, v_{G_2}, v_W, v_F$. Note that if v_{G_1} is 2, the fault set combination can have $F^{G_1}_{i,j}$ and $F^{G_1}_{i',j'}$ with $i = i' = 1$, or have $F^{G_1}_{i,j}$ with $i = 2$. Also, if v_{G_1} is 3, the fault set combination can have $F^{G_1}_{i,j}$, $F^{G_1}_{i',j'}$, and $F^{G_1}_{i'',j''}$ with $i = i' = i'' = 1$, or have $F^{G_1}_{i,j}$ and $F^{G_1}_{i',j'}$ with $i = 2, i' = 1$, or have $F^{G_1}_{i,j}$ with $i = 3$. The same goes for v_{G_2}.

(a) For each fault set combination, find all possible fault combinations. Each fault combination can be found by picking one fault from each fault set (up to three sets) in the fault set combination, then combine the faults to get the Pauli error E and the cumulative flag vector f_x associated with the fault combination.

(b) For each fault combination, find first-level syndrome s_x, second-level syndrome $\tilde{s}_x$, block triviality τ_x, and block parity p_x from the associated Pauli error E. Store $(s_x, \tilde{s}_x, \tau_x, f_x, p_x)$ for each fault combination in a lookup table.

(4) After the lookup table is complete, categorize fault combinations by their second-level syndromes and block trivialities in order to get $\mathcal{F}_k$'s as in Claim 2.

(5) For each $\mathcal{F}_k$, verify whether Condition 1 or 2 in Claim 2 is satisfied.

From the simulation above, we find that all possible fault combinations satisfy Claim 2. That is, for each fault combination, we can determine the weight parity from the outcome bundles obtained from the last round of full syndrome measurement by looking at the table constructed in Step 3b. The weight parity can be later used to perform WPEC on the code block. With this simulation result, we can verify our FTEC protocol for the $[\![49, 1, 9]\!]$ code satisfies FTEC conditions as previously discussed in Sec. III.

APPENDIX B: SIMULATION OF POSSIBLE FAULTS DURING THE FTEC PROTOCOL ASSUMING THAT THE LAST ROUND OF FULL SYNDROME MEASUREMENT HAS SOME FAULTS

In Appendix A, we describe the simulation of possible faults during the FTEC protocol for the $[\![49, 1, 9]\!]$ code which is applicable to the case that there are no faults during the last round of full syndrome measurement. In this Appendix, we will extend the ideas and construct a simulation of possible faults for the case that some faults occur during the last round.

As previously described, we will perform full syndrome measurements in the protocol until the outcome bundles are repeated four times in a row. Now, suppose that the last round of full syndrome measurement has some faults. In this case, we cannot be sure whether the outcome bundle from the last round exactly corresponds to the error in the data qubits. Fortunately, since there are at most three faults during the whole protocol, at least one outcome bundle obtained from the last four rounds must be correct. Note that the outcome bundles from the last four rounds are identical. From the simulation result discussed in Appendix A, the outcome bundle from the last round can be used to correct the data error occurred before *any* correct round using the WPEC technique (see Fig. 3 for more details). The goal of the simulation in this section is to verify that all possible fault combinations which can happen after the last correct round give data error of weight no more than 3.

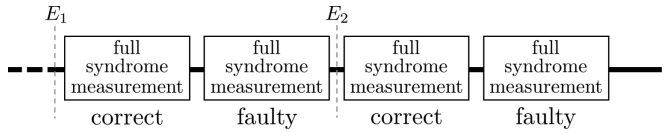
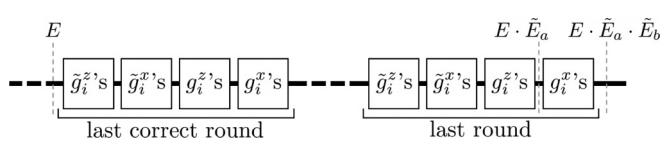


FIG. 3. At least one of the last four rounds of full syndrome measurement is correct since there are at most three faults. Because the outcome bundles from the last four rounds are identical, the outcome bundle from the last round can be used in WPEC to correct both errors E_1 and E_2 (even though E_1 and E_2 may not be equal).

A straightforward way to verify the claim above is to find all possible fault combinations and check the weight of their associated Pauli errors. Unfortunately, this process requires many computational resources. Thus, we will use "relaxed conditions" for the verification instead; for each fault combination, if the associated Pauli error and flag vector satisfy all relaxed conditions, the fault combination will be marked (indicating that the fault combination might cause the protocol to fail). We want to make sure that for all fault combination that can cause the protocol to fail (i.e., its associated error has weight more than 3), the fault combination will be marked. Note that some fault combinations may be marked by the relaxed conditions but will not cause the protocol to fail. For this reason, all of the marked fault combinations must be examined after the simulation is done.

We should note that the order of generator measurements is important for the fault tolerance of our FTEC protocol. Consider the protocol description in Sec. III in which we measure generator measurements in the following order during a single round of full syndrome measurement: measuring all $\tilde{g}^z_i$'s, then all $\tilde{g}^x_i$'s, then all g^z_i's, then all g^x_i's. Let us first consider the errors that can be caught by the g^x_i measurements of the last round. Observe that all Z-type data errors that arise before the g^x_i measurements of the last round will be evaluated by the first-level syndrome s_x. However, some faults during g^x_i measurements of the last round may cause X- or Z-type errors that will not be caught by any syndrome. Without loss of generality, we will construct a simulation using an assumption that faults before the g^x_i measurements of the last round can cause only Z-type errors, and faults during or after the g^x_i measurements can cause X- or Z-type errors. The simulation is also applicable to the case of g^z_i measurements.

Let E, $\tilde{E}_a$, and $\tilde{E}_b$ be data errors arising from faults occurred before the last correct round among the last four rounds, faults occurred after the last correct round but before the g^x_i measurements of the last round, and faults occurred during or after the g^x_i measurements of the last round. The errors can be illustrated as follows:



The outcome bundle obtained from the last round is equal to the outcome bundle obtained from the correct round and can be used to correct E. Thus, we would like to mark every fault combination that can occur after the correct round, corresponds to the trivial outcome bundle (since the outcome bundle obtained from the last round is the same as that obtained from the correct round), and corresponds to a Pauli error of weight more than 3. In particular, our relaxed conditions will examine three objects for each fault combination: the first-level syndrome, the cumulative flag vector, and the weight of the Pauli error.

The mathematical objects being used in this simulation are similar to those defined in Appendix A. In addition, we will consider syndrome measurement faults (denoted by S) as another type of faults in this simulation since we will assume that the syndrome measurement during the last four rounds can be faulty. Also, let $v_{G_{1a}}$ be the number of G_1 faults that occur before the g_i^x measurements of the last round, and let $v_{G_{1b}}$ be the number of G_1 faults that occur during or after the g_i^x measurements of the last round. *Fault number combination* is a tuple $(v_{G_{1a}}, v_{G_{1b}}, v_{G_2}, v_W, v_F, v_S)$ that satisfies $v_{G_{1a}} + v_{G_{1b}} + v_{G_2} + v_W + v_F + v_S \leqslant 3$.

For the first relaxed condition, let us first assume that none of the faults of type W occurs before or during the g_i^x measurements of the last round. For each $(v_{G_{1a}}, v_{G_{1b}}, v_{G_2}, v_W, v_F, v_S)$, error $\tilde{E}_a$ will be constructed from possible fault combinations that correspond to $v_{G_{1a}}$ and v_{G_2}. We will mark every fault combination whose associated $\tilde{E}_a$ gives a first-level syndrome that has Hamming weight no more than v_S (where the Hamming weight is the number of 1's in a bitstring). This is because each fault of type S can alter at most one syndrome bit. Now let us consider the case that some faults of type W occurs before or during the g_i^x measurements. Each W fault (which corresponds to error of weight 1) can change at most three bits of s_x, but the change will affect only the subblock in which the fault acts nontrivially. We will define function $\sigma(\tilde{E}_a, v_W)$ by the following calculation:

(1) Find the first-level syndrome of $\tilde{E}_a$ and calculate the Hamming weight of the syndrome for each subblock.

(2) Sort the Hamming weights from all subblocks. The function value is the the sum of the $7 - v_W$ smallest Hamming weights.

The value of $\sigma(\tilde{E}_a, v_W)$ is the minimum Hamming weight of the first-level syndrome when v_W faults of type W occur. Taking all fault types into account, our first relaxed condition becomes

$$\sigma(\tilde{E}_a, v_W) \leqslant v_S. \tag{B1}$$

For the second relaxed condition, we will consider the cumulative flag vector associated with each fault combination. Note that a flag measurement result will be obtained during any g_i^x or g_i^z measurement. Let $f = (f_x|f_z)$ denote the cumulative flag vector associated with each fault combination, and let $h(f)$ denote the Hamming weight of f. Since each fault of type F can alter at most one bit of f, our second relaxed condition becomes

$$h(f) \leqslant v_F. \tag{B2}$$

For the third relaxed condition, we will consider the weight of the Pauli error associated with each fault combination. The weight is evaluated at the end of the protocol where the resulting error is caused by all faults of type G_1, G_2, and W (errors arising during or after the g_i^x measurements of the last round can be X- or Z-type). If W faults do not occur before or during the g_i^x measurements at the last round, the weight of the resulting error is the weight of $\tilde{E}_a \cdot \tilde{E}_b$. If they do, each W fault can increase the total weight by at most 1. Hence, our third condition becomes

$$\mathrm{wt}(\tilde{E}_a \cdot \tilde{E}_b) + v_W > 3. \tag{B3}$$

Note that the weight of $\tilde{E}_a \cdot \tilde{E}_b$ can be reduced by multiplication of some stabilizer, and the fault combination will not be marked unless (B3) is satisfied for all choice of stabilizer.

Using the relaxed conditions in Eqs. (B1) to (B3), our simulation to verify that all possible data errors arising after the correct round have weight no more than three can be constructed as follows.

Pseudocode for a simulation of possible faults assuming that the last round of full syndrome measurement has some faults

(1) Construct fault sets $F_{i,j}^{G_1}, F_{i,j'}^{G_2}$ for all $i = 0, 1, 2, 3$, $j = 1, \ldots, 21$, $j' = 1, 2, 3$.

(2) Construct all possible fault number combinations that satisfies $v_{G_{1a}} + v_{G_{1b}} + v_{G_2} + v_W + v_F + v_S \leqslant 3$.

(3) For each $(v_{G_{1a}}, v_{G_{1b}}, v_{G_2}, v_W, v_F, v_S)$, construct all possible fault set combinations from only $v_{G_{1a}}$, $v_{G_{1b}}$, and v_{G_2}. During the construction of each fault set combination, label fault sets that come from $v_{G_{1a}}$ or v_{G_2} with letter a, and label fault sets that come from $v_{G_{1b}}$ with letter b. Note that if $v_{G_{1a}}$ is 2, the fault set combination can have $F_{i,j}^{G_1}$ and $F_{i',j'}^{G_1}$ with $i = i' = 1$, or have $F_{i,j}^{G_1}$ with $i = 2$. Also, if $v_{G_{1a}}$ is 3, the fault set combination can have $F_{i,j}^{G_1}$, $F_{i',j'}^{G_1}$, and $F_{i'',j''}^{G_1}$ with $i = i' = i'' = 1$, or have $F_{i,j}^{G_1}$ and $F_{i',j'}^{G_1}$ with $i = 2, i' = 1$, or have $F_{i,j}^{G_1}$ with $i = 3$. The same goes for $v_{G_{1b}}$ and v_{G_2}.

(a) For each fault set combination, find all possible fault combinations. Each fault combination can be found by picking one fault from each fault set (up to three sets) in the fault set combination. $\tilde{E}_a$ associated with each fault combination can be found by combining only faults from fault sets with label a, while f and $\tilde{E}_a \cdot \tilde{E}_b$ can be found by combining faults from all fault sets.

(i) For each fault combination, if Eqs. (B1) to (B3) are all satisfied, the fault combination will be marked. Note that for (B3), the weight of $\tilde{E}_a \cdot \tilde{E}_b$ must be minimized by stabilizer multiplication.

From the simulation above, we find that there are six fault combinations which are marked by the relaxed conditions in Eqs. (B1) to (B3). All of them correspond to the case that $v_{G_2} = 1$, $v_W = 2$, $v_{G_{1a}}, v_{G_{1b}}, v_F, v_S = 0$, and their associated Pauli errors are trivial on five subblocks and have either $IIIIIIZ$ or $ZIIIIII$ on two subblocks. We find that $IIIIIIZ$ and $ZIIIIII$ correspond to first-level syndrome (001) and (100), respectively. Since $v_S = 0$, the associated first-level syndrome must be trivial whenever errors from W faults are taken into account. This can happen only when errors from W faults cancel with the aforementioned Pauli error, which means that the resulting error has weight 0. As a result, we

find that all of the marked fault combinations cannot cause data error of weight higher than 3. Similar simulations can be done to show that whenever v faults occur where $v = 0, 1, 2,$ the weight of the output error is at most v. This result verifies that the FTEC protocol for the $[\![49, 1, 9]\!]$ code satisfies FTEC conditions as discussed in Sec. III.

[1] P. W. Shor, Fault-tolerant quantum computation, in *Proceedings of 37th Conference on Foundations of Computer Science* (IEEE Computer Society Press, Los Alamitos, California, 1996), pp. 56–65.

[2] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, SIAM J. Comput. **38**, 1207 (2008).

[3] J. Preskill, Reliable quantum computers, Proc. R. Soc. London A **454**, 385 (1998).

[4] E. Knill, R. Laflamme, and W. H. Zurek, Threshold accuracy for quantum computation, arXiv:quant-ph/9610011 (1996).

[5] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Inf. Comput. **6**, 97 (2006).

[6] A. M. Steane, Overhead and noise threshold of fault-tolerant quantum error correction, Phys. Rev. A **68**, 042322 (2003).

[7] A. Paetznick and B. W. Reichardt, Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit Golay code, Quantum Inf. Comput. **12**, 1034 (2012).

[8] C. Chamberland, T. Jochym-O'Connor, and R. Laflamme, Overhead analysis of universal concatenated quantum codes, Phys. Rev. A **95**, 022313 (2017).

[9] R. Takagi, T. J. Yoder, and I. L. Chuang, Error rates and resource overheads of encoded three-qubit gates, Phys. Rev. A **96**, 042302 (2017).

[10] D. P. DiVincenzo and P. Aliferis, Effective Fault-Tolerant Quantum Computation with Slow Measurements, Phys. Rev. Lett. **98**, 020501 (2007).

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

[12] A. M. Steane, Active Stabilization, Quantum Computation, and Quantum State Synthesis, Phys. Rev. Lett. **78**, 2252 (1997).

[13] A. M. Steane, Fast fault-tolerant filtering of quantum codewords, arXiv:quant-ph/0202036 (2002).

[14] A. W. Steane, Multiple-particle interference and quantum error correction, Proc. Roy. Soc. Lond. **452**, 2551 (1996).

[15] T. J. Yoder and I. H. Kim, The surface code with a twist, Quantum **1**, 2 (2017).

[16] R. Chao and B. W. Reichardt, Quantum Error Correction with Only Two Extra Qubits, Phys. Rev. Lett. **121**, 050502 (2018).

[17] R. Chao and B. W. Reichardt, Flag fault-tolerant error correction for any stabilizer code, PRX Quantum **1**, 010302 (2020).

[18] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, Quantum **2**, 53 (2018).

[19] T. Tansuwannont, C. Chamberland, and D. Leung, Flag fault-tolerant error correction, measurement, and quantum computation for cyclic Calderbank-Shor-Steane codes, Phys. Rev. A **101**, 012342 (2020).

[20] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu, Triangular color codes on trivalent graphs with flag qubits, New J. Phys. **22**, 023019 (2020).

[21] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Topological and subsystem codes on low-degree graphs with flag qubits, Phys. Rev. X **10**, 011022 (2020).

[22] R. Chao and B. W. Reichardt, Fault-tolerant quantum computation with few qubits, npj Quantum Inf. **4**, 42 (2018).

[23] C. Chamberland and A. W. Cross, Fault-tolerant magic state preparation with flag qubits, Quantum **3**, 143 (2019).

[24] Y. Shi, C. Chamberland, and A. Cross, Fault-tolerant preparation of approximate GKP states, New J. Phys. **21**, 093007 (2019).

[25] P. Baireuther, M. Caio, B. Criger, C. W. Beenakker, and T. E. O'Brien, Neural network decoder for topological color codes with circuit level noise, New J. Phys. **21**, 013003 (2019).

[26] A. Bermudez, X. Xu, M. Gutiérrez, S. C. Benjamin, and M. Müller, Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources, Phys. Rev. A **100**, 062307 (2019).

[27] C. Vuillot, Is error detection helpful on IBM 5Q chips? Quantum Inf. Comput. **18**, 0949 (2018).

[28] M. Gutiérrez, M. Müller, and A. Bermúdez, Transversality and lattice surgery: Exploring realistic routes toward coupled logical qubits with trapped-ion quantum processors, Phys. Rev. A **99**, 022330 (2019).

[29] L. Lao and C. G. Almudever, Fault-tolerant quantum error correction on near-term quantum processors using flag and bridge qubits, Phys. Rev. A **101**, 032333 (2020).

[30] C. Chamberland and K. Noh, Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits, npj Quantum Inf. **6**, 91 (2020).

[31] D. M. Debroy and K. R. Brown, Extended flag gadgets for low-overhead circuit verification, Phys. Rev. A **102**, 052409 (2020).

[32] A. Rodriguez-Blanco, A. Bermudez, M. Müller, and F. Shahandeh, Efficient and robust certification of genuine multipartite entanglement in noisy quantum error correction circuits, PRX Quantum **2**, 020304 (2021).

[33] B. W. Reichardt, Fault-tolerant quantum error correction for Steane's seven-qubit color code with few or no extra qubits, Quantum Sci. Tech. **6**, 015007 (2020).

[34] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, Phys. Rev. A **54**, 1098 (1996).

[35] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, California Institute of Technology, 1997.

[36] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes* (North-Holland, New York, 1977).

[37] T. Tansuwannont and D. Leung, Achieving fault tolerance on capped color codes with few ancillas, arXiv:2106.02649 (2021).