Optimization of Quantum Circuits for Stabilizer Codes

Arijit Mondal and Keshab K. Parhi, *Fellow, IEEE*
Email: {monda109, parhi}@umn.edu
Department of Electrical and Computer Engineering, University of Minnesota

***Abstract*—Quantum computing is an emerging technology that has the potential to achieve exponential speedups over their classical counterparts. To achieve quantum advantage, quantum principles are being applied to fields such as communications, information processing, and artificial intelligence. However, quantum computers face a fundamental issue since quantum bits are extremely noisy and prone to decoherence. Keeping qubits error free is one of the most important steps towards reliable quantum computing. Different stabilizer codes for quantum error correction have been proposed in past decades and several methods have been proposed to import classical error correcting codes to the quantum domain. Design of encoding and decoding circuits for the stabilizer codes have also been proposed. Optimization of these circuits in terms of the number of gates is critical for reliability of these circuits. In this paper, we propose a procedure for optimization of encoder circuits for stabilizer codes. Using the proposed method, we optimize the encoder circuit in terms of the number of 2-qubit gates used. The proposed optimized eight-qubit encoder uses 18 CNOT gates and 4 Hadamard gates, as compared to 14 single qubit gates, 33 2-qubit gates, and 6 CCNOT gates in a prior work. The encoder and decoder circuits are verified using IBM Qiskit. We also present encoder circuits for the Steane code and a 13-qubit code that are optimized in terms of the number of gates used, leading to a reduction in number of CNOT gates by 1 and 8, respectively.**

***Index Terms*—Quantum ECCs, Quantum computation, Stabilizer codes, Eight-qubit code, 13-qubit code, Quantum encoders and decoders, Syndrome detection.**

I. Introduction

Quantum computing is a rapidly-evolving technology which exploits the fundamentals of quantum mechanics towards solving tasks which are too complex for current classical computers. In 1994, a quantum algorithm to find the prime factors of an integer in *polynomial* time was proposed by Shor [1] . In 1996, a quantum algorithm to search a particular element in an unsorted database with a high probability and significantly higher efficiency than any known classical algorithm was presented by Grover [2]. The realization of these powerful algorithms requires massive quantum computers with the capability of processing a large number of qubits.

The phenomenon through which quantum mechanical systems attain interference among each other is known as quantum coherence. Quantum coherence is essential to perform quantum computations on quantum information. However, quantum systems are inherently susceptible to noise and decoherence. Maintaining coherence and mitigating noise becomes increasingly challenging with the increase in the number of qubits in a quantum computer. Thus, quantum error correcting codes (ECCs) become essential for reliable quantum computing systems. There were various challenges in the process of designing a quantum ECC framework. It is well known that measurement destroys superposition in any quantum system. Also, since the quantum errors are continuous in nature, the design of an ECC for quantum systems was difficult. Furthermore, the no-go theorems in the quantum domain make it difficult to design an ECC system analogous to classical domain [3], [4], [5], [6], [7].

Quantum ECCs were believed to be impossible till 1995, when Shor demonstrated a 9-qubit ECC which was capable of correcting a single qubit error for the first time [8]. In 1996, Gottesman proposed a stabilizer framework which was widely used for construction of quantum ECCs from classical ECCs [9], [10]. Calderbank-Shor-Steane (CSS) codes were proposed independently by Calderbank-Shor [11] and Steane [12]. These codes were used to derive quantum codes from binary classical linear codes. Pre-shared entangled qubits were used to construct stabilizer codes over non-Abelian groups in [13]. These entanglement-assisted (EA) stabilizer codes contain qubits over the extended operators which are assumed to be at the receiver end throughout, and entangled with the transmitted set of qubits. It was later shown that EA stabilizer codes increase the error correcting capability of quantum ECCs [14].

An encoding procedure for EA stabilizer codes were proposed in [15]. Quantum low-density parity-check (LDPC) codes were constructed from classical quasi-cyclic binary LDPC codes by the authors in [16], [17], [18], [19]. Quantum analogs of Reed Solomon (RS) codes were constructed from self-orthogonal classical RS codes in [20], [21], [22]. Recent advances in this field include: hypergraph-product codes and homological product codes [23], [24], and tensor product of qunatum and classical codes [25], [26]. Purely quantum polar codes based on recursive channel combining and splitting construction were studied in [27]. EA stabilizer codes were extended to qudit systems in [28]. Recently, a universal decoding scheme was conceived for quantum stabilizer codes (QSCs) by adapting 'guessing random additive noise decoding' (GRAND) philosophy from classical domain codes [29]. However, it becomes necessary to design and simulate actual encoder and decoder circuits for these quantum ECCs, so that reliable quantum computing systems can be built. The CSS framework is of particular interest due to its simplicity as it

This work was supported in part by the National Science Foundation under grant number CCF-1954749..

is useful for importing classical codes to quantum domain if they satisfy certain properties [11], [12].

A systematic method for the construction of an encoder for stabilizer codes was demonstrated for a five qubit code in [10], which uses five physical qubits to encode a logical qubit. The authors in [30] reviewed this construction and presented it in the form of an algorithm for the construction of encoder circuits for stabilizer codes. We apply the algorithm to the eight-qubit code $[[8,3,3]]$ which encodes three logical qubits using eight physical qubits and is more efficient in terms of code rate than the 5-qubit code. CNOT gates play a vital role as building blocks for quantum circuits. It can be shown that any arbitrary unitary transformation on an n-qubit system can be performed as a combination of CNOT gates and other single qubit unitary gates [3], thus forming an universal set of gates for quantum circuits. The circuits designed in [10] contain 3 types of 2-qubit gates, the CNOT, CZ, and the CY gates. However, for practical quantum circuits, it is more convenient to have a single type of 2-qubit gates between any arbitrary pair of qubits. Similar to equivalence rules in classical digital circuits, several equivalencies also exist for quantum circuits. These were used to study quantum circuits in [31], [32], [33], [34]. The authors in [35] derived additional rules and compiled those into a set of equivalence rules. In the current experimental quantum circuits, the error rates of the CNOT gates are one of the major causes of circuit reliability [36], [37], [38]. Thus, it becomes extremely important to optimize the quantum circuits in terms of the number of CNOT gates. There exists a one-to-one correspondence between n-qubit systems consisting of CNOT gates and $n \times n$ non-singular matrices with coefficients in $\mathbb{F}_2$ as proved in [39], [40]. The authors in [38] described this correspondence in a formal way in terms of group isomorphism and group representation. They used this approach to propose an optimization algorithm for a system consisting of a few qubits, with CNOT gates interacting between them.

The contributions of this paper are two-fold. First, we propose a procedure for optimization of encoder circuits for stabilizer codes in terms of the number of gates used. Using the procedure, we design an optimized encoder circuit for an eight-qubit code. Compared to a prior construction in [41], our circuit requires half the number of gates. We use only CNOT and H gates for the circuit design. To the best of our knowledge, such an optimized circuit for stabilizer codes has not been explored before. Second, we design optimized encoder circuits for Steane code [42] and a 13-qubit code. For the Steane code, the number of CNOT gates was reduced by 1. However, for the 13-qubit code, 16% reduction in the number of gates was achieved. The method could potentially be useful for large quantum codes, leading to significant reductions in number of gates. Finally, we verify all the circuits we designed using IBM Qiskit [43].

The rest of the paper is organized as follows. In Section II, we present a brief review of the theory of the stabilizer framework which is essential for the design of the quantum ECCs. We also provide the theoretical background of the CSS framework. Subsequently, we derive the encoding and syndrome measurement circuits using the systematic method presented in [10]. In Section III, we propose a procedure for optimization of encoder circuits for stabilizer codes. The procedure is applied to the 8-qubit code designed in Section III. In Section IV, we present optimized encoder circuits for the Steane code and a 13-qubit code. We discuss the results and comparisons in Section V. Section VI concludes the paper.

II. Theoretical background

Shor's 9-qubit code [8] was the first ever quantum ECC capable of correcting a single qubit error. Gottesman [10] proposed a general methodology to construct quantum ECCs. This method is known as the stabilizer construction and the codes thus generated are known as stabilizer codes. Before going into the details of stabilizer codes, we briefly describe a quantum state and explain how the states evolve in a quantum circuit.

In two-level quantum systems, the two-dimensional unit of quantum information is called a quantum bit (qubit). The state of a qubit is represented by $|\psi\rangle = a|0\rangle + b|1\rangle$, where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. $|0\rangle$ and $|1\rangle$ are basis states of the state space. The evolution of a quantum mechanical system is fully described by a unitary transformation. State $|\psi_1\rangle$ of a quantum system at time t_1 is related to state $|\psi_2\rangle$ at time t_2 by a unitary operator U that depends only on the time instances t_1 and t_2, i.e., $|\psi_2\rangle = U|\psi_1\rangle$. The unitary operators or matrices which act on the qubit belong to $\mathbb{C}^{2\times 2}$. The Pauli group which represents the unitary matrices is given by

$$\Pi = \{\pm I_2, \pm iI_2, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\} \quad (1)$$

where $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$, $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

A quantum circuit consists of an initial set of qubits as inputs which evolve through time to a final state, comprising of the outputs of the quantum circuit. Quantum states evolve through unitary operations which are represented by quantum gates. Quantum gates can be single qubit gates which act on a single qubit, or they can be multiple qubit gates which act on multi-qubit states to produce a new multi-qubit state. The single qubit gates include the bit flip gate X, phase flip gate Z, Hadamard gate H, Y gate, and the phase gate S. The unitary operations related to the single qubit gates are described as follows:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (2)$$
$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

The multi-qubit gates include a controlled-X (CX, also referred as CNOT), controlled-Z (CZ), and controlled-Y (CY) gates. They act on 2-qubit states and are given by the following unitary transformations:

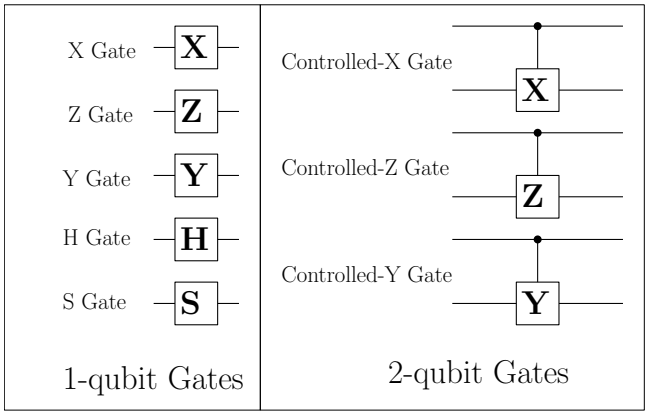

Fig. 1. Symbolic representations of various 1-qubit and 2-qubit gates.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \tag{3}$$

$$CY = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix} \tag{4}$$

Symbolic representations of various 1-qubit and 2-qubit gates are shown in Fig. 1. For further understanding on quantum circuits and introduction to quantum ECCs, interested readers are referred to [44], [45]. With the above background, we are ready to describe the stabilizer formalism for quantum ECCs.

A. Quantum code and Stabilizer formalism

An $[[n, k, d]]$ quantum code can be used for quantum error correction, where k logical qubits are encoded using n physical qubits, leading to a code rate of k/n analogous to classical error correction. The code has a minimum distance of d. It has 2^k basis codewords, and any linear combination of the basis codewords are also valid codewords. Let the space of valid codewords be denoted by T. If we consider the tensor product of Pauli operators (with possible overall factors of ± 1 or $\pm i$) in Equation 1, it forms a group G under multiplication. The stabilizer $\mathcal{S}$ (not to be confused with gate S) is an Abelian subgroup of G, such that the code space T is the space of vectors fixed by $\mathcal{S}$ [9], [10]. Stabilizer generators are a set of independent set of $n-k$ elements from the stabilizer group, in the sense that none of them is a product of any two other generators.

We know that the operators in the Pauli group act on single qubit states which are represented by 2-element vectors. The operators in Π have eigenvalues ± 1, and either commute or anti-commute with other elements in the group. The set Π^n is given by the n-fold tensor products of elements from the Pauli group Π as shown below,

$$\begin{aligned} \Pi^n = &\{e^{i\phi} A_1 \otimes A_2 \otimes \cdots \otimes A_n \\ &: \forall j \in \{1, 2, \cdots, n\} A_j \in \Pi, \phi \in \{0, \pi/2, \pi, 3\pi/2\}\} \end{aligned} \tag{5}$$

The stabilizer is a group with elements M such that $M|\psi\rangle = |\psi\rangle$. The stabilizer is Abelian, i.e., every pair of elements in the stabilizer group commute. This can be verified from the following observation. If $M|\psi\rangle = |\psi\rangle$ and $N|\psi\rangle = |\psi\rangle$, then $MN|\psi\rangle - NM|\psi\rangle = (MN - NM)|\psi\rangle = 0$. Thus, $MN - NM = 0$ or $MN = NM$, showing that every pair of elements in the stabilizer group commute.

Given an Abelian subgroup $\mathcal{S}$ of n-fold Pauli operators, code space is defined as

$$T(\mathcal{S}) = \{|\psi\rangle, s.t.\, M|\psi\rangle = |\psi\rangle, \forall M \in \mathcal{S}\} \tag{6}$$

Suppose $M \in \mathcal{S}$ and Pauli operator E anti-commutes with M. Then, $M(E|\psi\rangle) = -EM|\psi\rangle = -E|\psi\rangle$. Thus, $E|\psi\rangle$ has eigenvalue -1 for M. Conversely, if Pauli operator E commutes with M, $M(E|\psi\rangle) = EM|\psi\rangle = E|\psi\rangle$; thus $E|\psi\rangle$ has eigenvalue $+1$ for M. Therefore, eigenvalue of an operator M from a stabilizer group detects errors which anti-commute with M.

B. Binary vector space representation for stabilizers

The stabilizers can be written as binary vector spaces, which can be useful to bring connections with classical error correction theory [10]. For this, the stabilizers are written as a pair of $(n-k) \times n$ matrices. The rows correspond to the stabilizers and the columns correspond to the qubits. The first matrix has a 1 wherever there is a X or Y in the corresponding stabilizer, and 0 everywhere else. We will refer to this as the X-portion of the matrix. The second matrix has a 1 wherever there is a Z or Y in the corresponding stabilizer and 0 everywhere else. We will refer to this as the Z-portion of the matrix. It is often more convenient to write the two matrices as a single $(n-k) \times 2n$ matrix with a vertical line separating the two.

C. CSS framework

The CSS framework [11], [12] is a method to construct quantum ECCs from their classical counterparts. We can combine classical codes with parity check matrices P_1 and P_2 into a quantum code if the rows of P_1 and P_2 are orthogonal using the binary dot product. This implies that dual-containing codes can be imported to the quantum domain. Given two classical codes $C_1[n, k_1, d_1]$ and $C_2[n, k_2, d_2]$ which satisfy the dual containing criterion $C_1^{\perp} \subset C_2$, CSS framework can be used to construct quantum codes from such codes.

The CSS codes form a class of stabilizer codes. From the classical theory of error correction, let H_1 and H_2 be the check matrices of the codes C_1 and C_2. Since $C_1^{\perp} \subset C_2$, codewords of C_2 are basically the elements of $C_1^{\perp}$. Hence, we have, $H_2 H_1^T = 0$. The check matrix of a CSS code is given by:

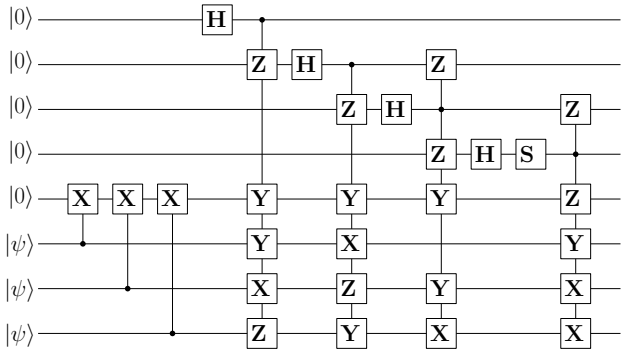

Fig. 2. Encoder for the eight-qubit code.

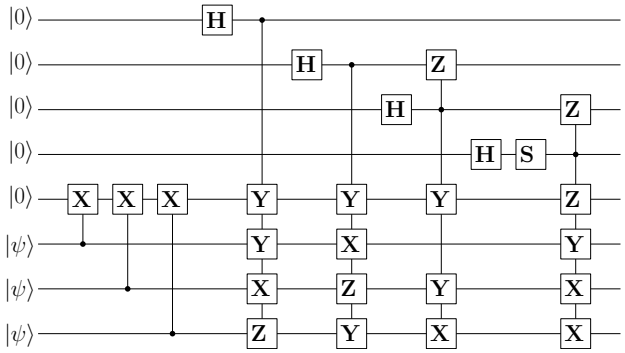

Fig. 3. Modified encoder for the eight-qubit code after removing redundant controlled-Z gates.

$$H_{C_1C_2} = \left[\begin{array}{c|c} H_1 & 0 \\ 0 & H_2 \end{array}\right] \tag{7}$$

D. Encoder and decoder circuit design for eight-qubit code [[8,3,3]]

A systematic method for the design of an encoder for a stabilizer code was presented in [10]. The authors in [30] used the key concepts in [10] to formulate an algorithm for the construction of encoder circuit for stabilizer codes. Details of the procedure are described in Section I of the Supplementary Information.

An eight-qubit code encodes $k = 3$ logical qubits using $n = 8$ physical qubits and can correct a single qubit error. Since $n - k = 8 - 3 = 5$, a total of 5 stabilizer generators are required as shown below [10]:

$$\begin{array}{c|cccccccc} M_1 & X & X & X & X & X & X & X & X \\ M_2 & Z & Z & Z & Z & Z & Z & Z & Z \\ M_3 & I & X & I & X & Y & Z & Y & Z \\ M_4 & I & X & Z & Y & I & X & Z & Y \\ M_5 & I & Y & X & Z & X & Z & I & Y \end{array} .$$

The encoder circuit can be designed using Algorithm 1 in [30], as shown in Fig. 2.

Observing the three CZ gates in Fig. 2 have one of their input qubits set at $|0\rangle$, we can remove those CZ gates to obtain the modified encoder circuit in Fig. 3.

The syndrome measurement circuit for an eight qubit code can also be designed as discussed in [30], as shown in Fig. 4.

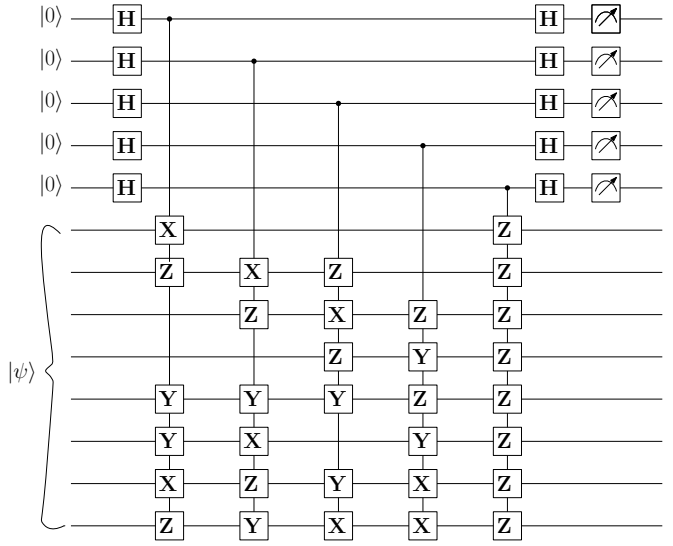

Fig. 4. Syndrome measurement circuit for the eight-qubit code.

The derivation of the encoder and syndrome measurement circuits for the eight qubit code is described in Section II of the Supplementary Information.

III. Optimization of eight-qubit encoder circuit

The encoder circuit designed in Fig. 3 uses 20 2-qubit gates and 5 single qubit gates. It requires 3 different types of 2-qubit gates, the CNOT, CY, and CZ gates. However, for most practical purposes, only a single type of 2-qubit gate may be available. Our goal is to optimize the circuit, such that it uses CNOT gates and H gates only. Since multiple qubit operations are a source of noise and decoherence, we also intend to minimize the number of CNOT gates. Our approach to optimizing the circuit consists of two steps. First, we use equivalence rules related to quantum gates for conversion between gates, and for moving the gates around in the circuit. We next use matrix equivalence to optimize a set of CNOT gates acting sequentially on a number of qubits. These two optimization strategies are described in more detail in the following sections.

A. Optimization using equivalence rules

Various equivalence rules related to quantum circuits have been presented and proven in [35]. These rules are illustrated in Fig. 5. Rules 1-3 relate to conversions between X and Z gates (or CNOT and CZ gates for 2-qubit gates). These rules also help a designer to switch control and target. Rules 4-5 are useful for circuits having multiple ancilla qubits at state $|0\rangle$, or state $|+\rangle$ in the Hadamard basis. Rules 6-10 are useful for manipulation of quantum circuits, enabling gates to move past each other. Additionally, we propose two new rules, referred as Rule 11 and Rule 12 in Fig. 5.

Rule 11 is useful for commutations between CNOT and CZ gates. We prove Rule 11 as follows. Let's consider the first pair of circuits. An arbitrary two-qubit initial state is given by:

$$\psi_1 = (\alpha_1|0\rangle + \alpha_2|1\rangle)(\beta_1|0\rangle + \beta_2|1\rangle) \quad (8)$$
$$= \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \alpha_2\beta_2|11\rangle \quad (9)$$

We analyze the circuit on the left first. After application of CNOT(1,2), we have

$$\psi_{l1} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|11\rangle + \alpha_2\beta_2|10\rangle \quad (10)$$

Next, CZ(1,2) is applied to get

$$\psi_{l2} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle - \alpha_2\beta_1|11\rangle + \alpha_2\beta_2|10\rangle \quad (11)$$

Now, we analyze the circuit on the right. The Z gate acting on the first qubit results in the following two-qubit state:

$$\psi_{r1} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle - \alpha_2\beta_1|10\rangle - \alpha_2\beta_2|11\rangle \quad (12)$$

Next, after application of CZ(1,2), we have

$$\psi_{r2} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle - \alpha_2\beta_1|10\rangle + \alpha_2\beta_2|11\rangle \quad (13)$$

Finally, after CNOT(1,2), we have

$$\psi_{r3} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle - \alpha_2\beta_1|11\rangle + \alpha_2\beta_2|10\rangle \quad (14)$$

We observe that the right hand side of equations 11 and 14 are the same. Thus output state of both the circuits are the same, thus proving the equivalency on the first pair. Similarly, it can also be proved that the circuits in the second pair are equivalent.

Rule 12 facilitates conversion between CY gates and a combination of CZ and CNOT gates. Additional single qubit S gate is required in the process. Rule 12 can be proved as follows.

An arbitrary two-qubit initial state is given by:

$$\psi_1 = (\alpha_1|0\rangle + \alpha_2|1\rangle)(\beta_1|0\rangle + \beta_2|1\rangle) \quad (15)$$
$$= \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \alpha_2\beta_2|11\rangle \quad (16)$$

For the circuit on the left, the output after CY gate is

$$\psi_{s1} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + i\alpha_2\beta_1|11\rangle - i\alpha_2\beta_2|10\rangle \quad (17)$$

For the circuit on the right, after S gate we have,

$$\psi_{p1} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + i\alpha_2\beta_1|10\rangle + i\alpha_2\beta_2|11\rangle \quad (18)$$

After CZ gate we have,

$$\psi_{p2} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + i\alpha_2\beta_1|10\rangle - i\alpha_2\beta_2|11\rangle \quad (19)$$

After CX gate we have,

$$\psi_{p3} = \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + i\alpha_2\beta_1|11\rangle - i\alpha_2\beta_2|10\rangle \quad (20)$$

We observe that the right hand side of equations 17 and 20 are the same. Thus, the output state of both circuits are the same, proving the equivalency.

B. Optimization using group-theoretic matrix equivalence

Consider an n-qubit quantum circuit consisting of only CNOT gates. Let the initial state of the circuit be represented by an $n \times n$ identity matrix. A CNOT gate acting between qubits k_1 and k_2 with control on k_1 and target on k_2 can be represented by the row transformation $R_{k_2} \rightarrow R_{k_1} + R_{k_2}$. Thus, the entire circuit can be represented by a series of elementary row transformations. Two n-qubit quantum circuits are equivalent if they result in the same matrix after their respective elementary row transformations. For example, let us consider Rule 8 (CNOT distribution) in Fig. 5.

The leftmost circuit consists of a single elementary row transformation $R_3 \rightarrow R_1 + R_3$, which results in

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

The middle circuit consists of four row transformations $R_2 \rightarrow R_1 + R_2$, $R_3 \rightarrow R_2 + R_3$, $R_2 \rightarrow R_1 + R_2$, and $R_3 \rightarrow R_2 + R_3$. This results in

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

The rightmost circuit consists of four row transformations $R_3 \rightarrow R_2 + R_3$, $R_2 \rightarrow R_1 + R_2$, $R_3 \rightarrow R_2 + R_3$, and $R_2 \rightarrow R_1 + R_2$. This results in

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

We observe that all the three circuits result in the same final matrix, and are thus equivalent. The authors in [38] note that for a n-qubit quantum circuit consisting of only CNOT gates, the input-output transformation of states can be represented by a series of elementary row transformations. Taking the final matrix after the transformations, and applying a Gaussian elimination method to that leads to a set of steps resulting in the identity matrix. These sets of steps applied in reverse to obtain an identity matrix leads to an equivalent circuit for the initial circuit. However, this circuit may not be optimal.

C. Barriers towards optimization

Optimization of quantum circuits is a challenging problem. Global optimization of arbitrary quantum circuits is QMA-hard as noted in [46]. Considering the complexity of the problem, our goal is to reduce the number of gates in the stabilizer encoder circuits in less time. The results may be sub-optimal; however, the procedure follows a specific route that will work for large encoder circuits with a low time complexity. To ensure that only one type of 2-qubit gates is used in the circuit, all the CZ gates need to be converted to CNOT gates. However, doing this increases the number of H gates significantly. To solve this, we flip control and target for all the CZ gates before converting those to CNOT gates. As we will see later, this keeps the number of H gates constant. Also, it gives the circuit a definite structure, dividing it into

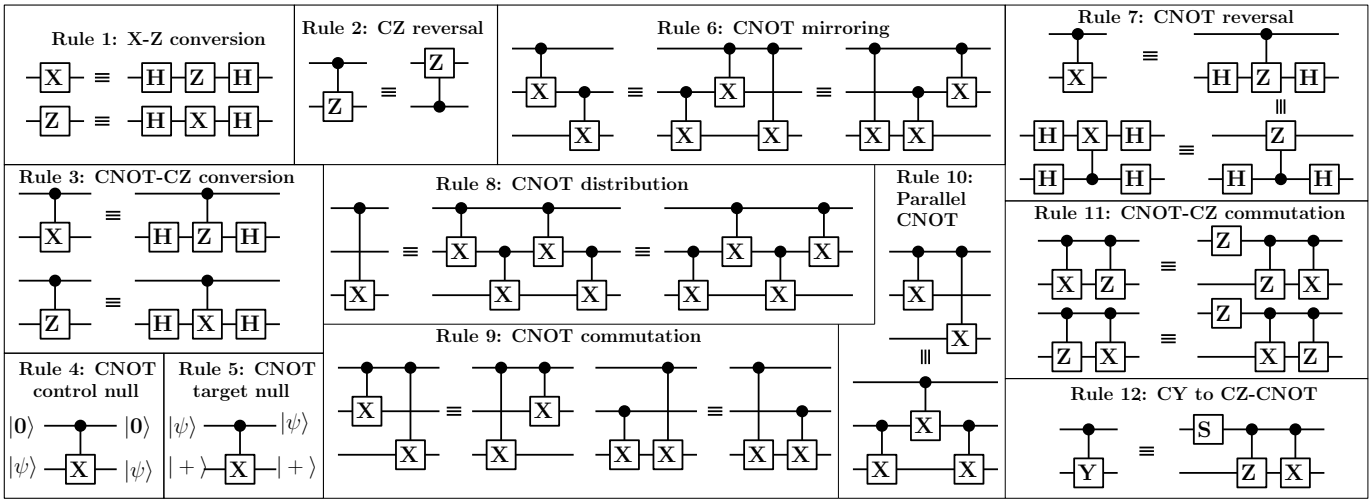

Fig. 5. Equivalence rules related to quantum circuits (partly based on [35]).

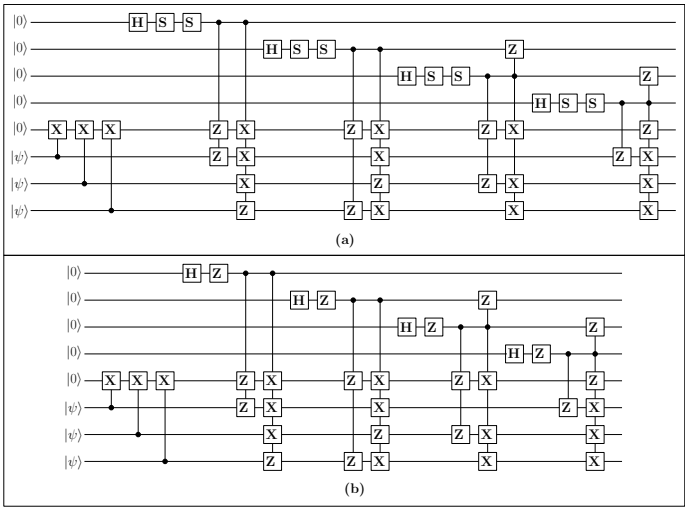

Fig. 6. Encoder for the eight-qubit code using controlled-ZX gates instead of controlled-Y gates.

sub-circuits containing H gates and CNOT gates. Using divide and conquer strategy, we can then try to optimize each of the sub-circuit separately. Later, we can rearrange the CNOT gates to create bigger sub-circuits of CNOT gates on which the matrix equivalence strategy may be applied.

D. Optimization procedure for stabilizer encoder circuits

We now illustrate how the above rules can be used to find an equivalent circuit for the 8-qubit encoder shown in Fig. 3. First, we replace all the CY gates with CZ and CNOT gates by using Rule 12 in 5, as shown in Fig. 6 (a). Since $S^2 = Z$, the circuit gets reduced to Fig. 6 (b).

Since most practical quantum computing systems use one type of 2-qubit gates, we will convert all CZ gates to CNOT gates using Rule 3 in Fig. 5. However, this step will result in the increase of H gates significantly. Thus, we propose a systematic way where each stabilizer is considered separately in a sequential fashion to achieve this goal. The steps for the

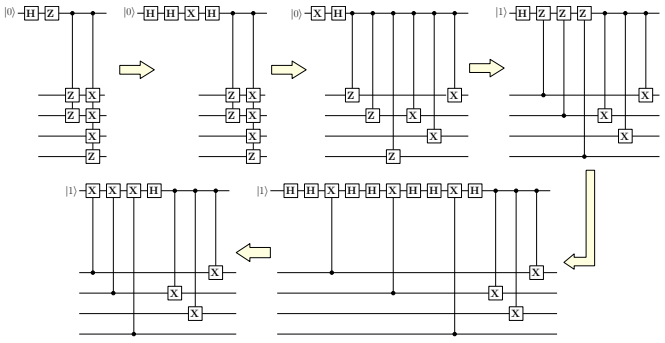

Fig. 7. Conversion of CZ gates to CNOT gates for the first stabilizer, keeping the number of H gates constant.

first stabilizer are illustrated in Fig. 7. First, control and target qubits of the CZ gates are reversed using Rule 2. Subsequently, the CZ gates are converted to CNOT gates using Rule 3. This leads to an introduction of X gate at the left of the H gate. This X gate can be removed by initializing the ancilla qubit to $|1\rangle$ instead of $|0\rangle$. Similar steps can be applied to all the stabilizers to obtain the circuit shown in Fig. 8.

Next, we will use the equivalence rules in Fig. 5 to reduce the number of CNOT gates in the encoder circuit as follows:

- The CNOT gates are rearranged to obtain the circuit in Fig. 9(a).
- Observe the green shaded regions in Fig. 9(a). Using CNOT mirroring (Rule 6), we can reduce the CNOT gates in each shaded region by 1. After slight rearrangement, we obtain the circuit as shown in Fig. 9(b).
- Using CNOT mirroring on the shaded region in Fig. 9(b), we obtain the circuit in Fig. 9(c).
- Next, we apply CNOT mirroring to the shaded region in Fig. 9(c). After slight rearrangement, we obtain the circuit in Fig. 9(d).
- We observe that the CNOT gate in the shaded region in Fig. 9(d) has the control qubit set as $|0\rangle$. Using Rule 4, we can remove the CNOT gate to obatin the circuit in Fig. 9(e).

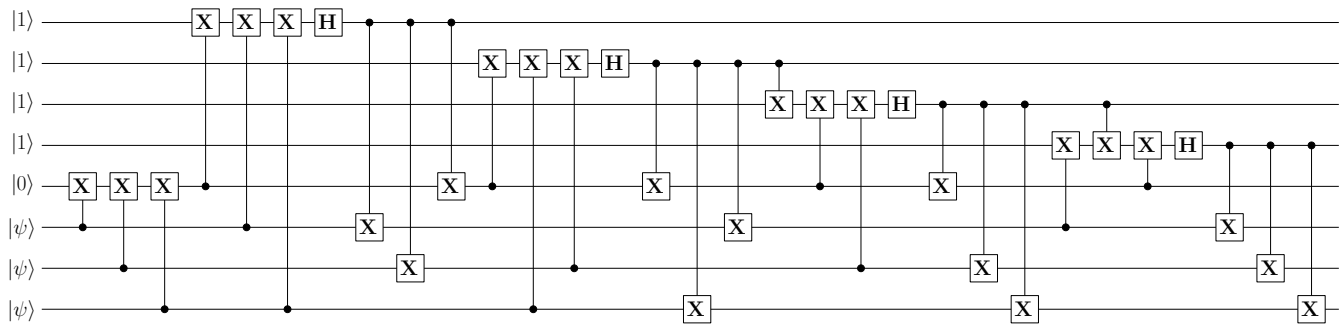

Fig. 8. Encoder for the eight-qubit code after converting CZ to CNOT gates.

- Applying Rule 6 to the CNOT mirror in Fig. 9(e), we reduce the number of CNOT gates by 1 as shown in Fig. 9(f).
- In Fig. 9(f), we apply the CNOT mirror to the shaded region. Subsequently, the CNOT gates are rearranged to transform it to Fig. 9(g).
- In Fig. 9(g), we observe three CNOT mirrors. Applying Rule 6 to the three shaded regions, we can reduce the total number of CNOT gates by 3 as shown in Fig. 9(h).
- We apply CNOT mirror to the green region in Fig. 9(h) to transform it to Fig. 9(i). This increases the number of CNOT gates by 1. However it leads us to the next reduction step.
- In Fig. 9(i), the control qubit on the CNOT gate in shaded region is $|0\rangle$. Applying Rule 4, we can remove that CNOT gate to produce the circuit in Fig. 9(j).

As discussed above, we could optimize the circuit to 19 CNOT and 4 Hadamard gates as shown in Fig. 9 (j). After we have optimized the circuit using the equivalence rules, it may still be possible to optimize the circuit further. However, it may be difficult to do so using visual inspection. To accomplish this goal, we use some approaches described in [38]. We use matrix equivalence for quantum circuits containing CNOT gates to optimise the circuit further, as we described next.

We apply this procedure to the shaded region in Fig. 9 (j), which consists of 11 CNOT gates. We start with an 8×8 identity matrix I, and apply the 11 sequential row transformations as follows:

$$\begin{aligned}
R_5 &\to R_5 + R_6 \\
R_7 &\to R_7 + R_1 \\
R_6 &\to R_6 + R_1 \\
R_5 &\to R_5 + R_8 \\
R_3 &\to R_3 + R_5 \\
R_5 &\to R_5 + R_7 \\
R_5 &\to R_5 + R_2 \\
R_8 &\to R_8 + R_2 \\
R_6 &\to R_6 + R_2 \\
R_4 &\to R_4 + R_6 \\
R_4 &\to R_4 + R_5
\end{aligned} \tag{21}$$

At the end of the row transformations, we obtain the matrix

$$T = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{22}$$

Gaussian elimination on T requires 15 steps (equivalent to 15 CNOT gates), which is less optimal than the original circuit. However, applying the following steps sequentially to T yields the 8×8 identity matrix.

$$\begin{aligned}
R_5 &\to R_5 + R_6 \\
R_4 &\to R_4 + R_5 \\
R_7 &\to R_7 + R_1 \\
R_8 &\to R_8 + R_2 \\
R_5 &\to R_5 + R_7 \\
R_3 &\to R_3 + R_5 \\
R_6 &\to R_6 + R_1 \\
R_6 &\to R_6 + R_2 \\
R_5 &\to R_5 + R_8 \\
R_3 &\to R_3 + R_6
\end{aligned} \tag{23}$$

Thus, the operations in Equation (23) applied in reverse order is equivalent to the operations in Equation (21). This is more optimal since it requires 10 transformations, and thus corresponds to 10 CNOT gates. The resulting encoder circuit consisting of 18 CNOT and 4 H gates is shown in Fig. 10.

E. Generalized optimization procedure for stabilizer encoder circuits

In the last section, we discussed the optimization procedure for the eight-qubit encoder circuit. If one observes carefully, the procedure is not specific to any particular encoder circuit. Any encoder circuit generated through Algorithm 1 in [30] has a similar structure to the eight-qubit encoder. It contains a certain number of H gates on the ancilla qubits (initialized to $|0\rangle$). After each of the H gates there is a combination of CNOT and CZ gates. We consider each of the H gates along

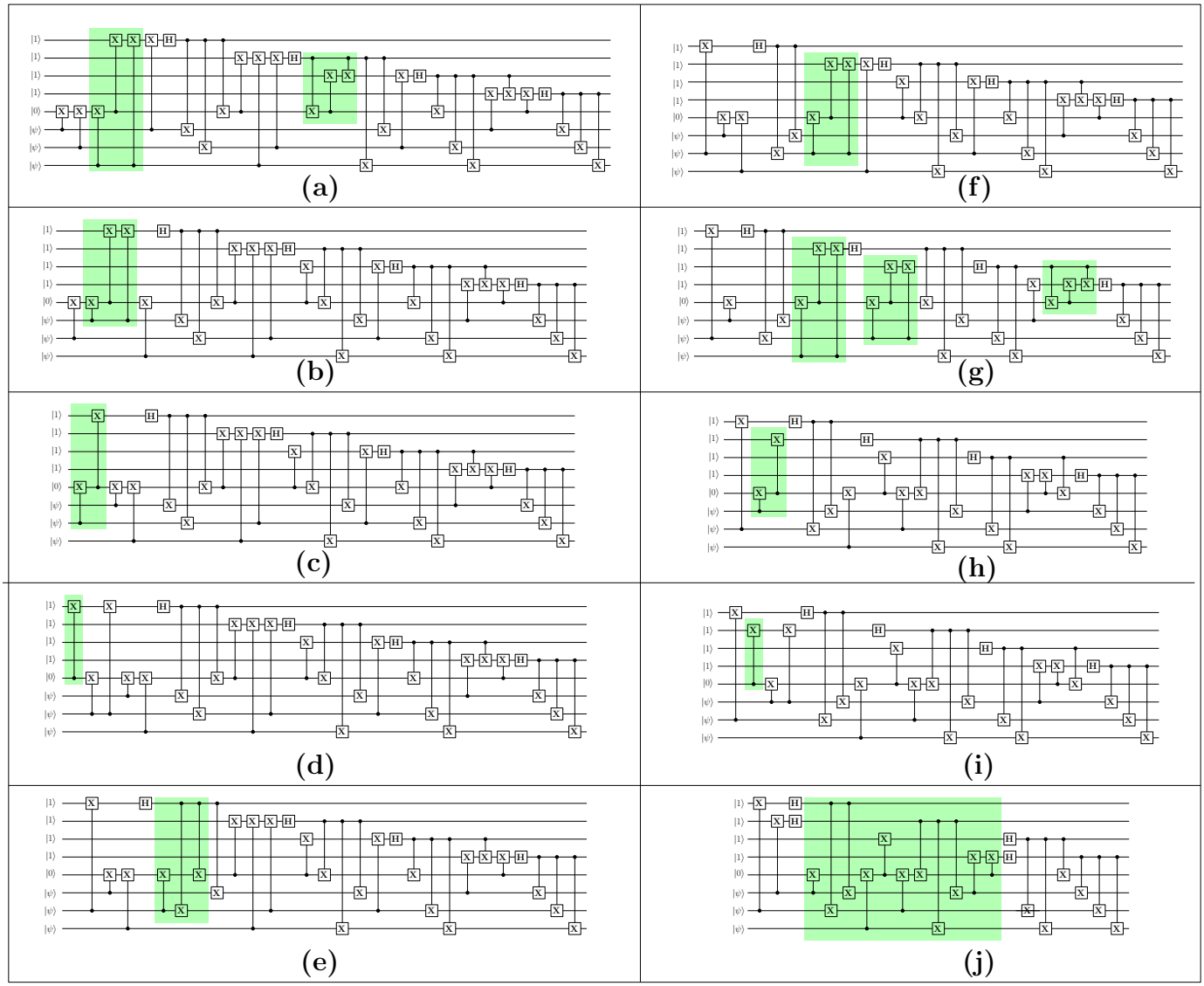

Fig. 9. Stepwise optimization of encoder circuit using equivalence rules in Fig. 5.

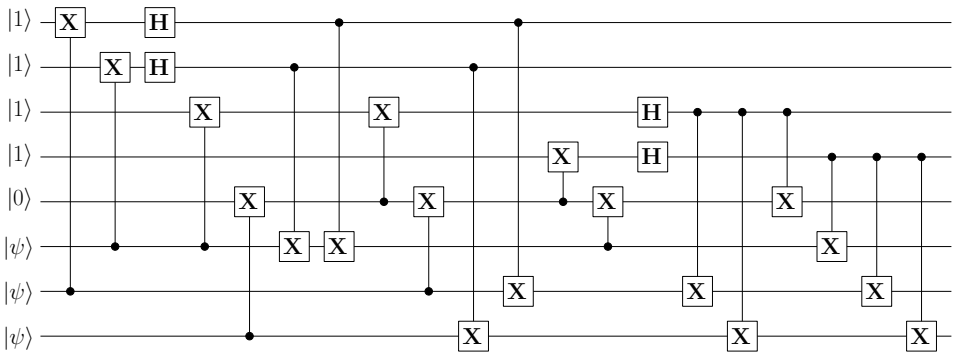

Fig. 10. Final optimized encoder circuit for the eight-qubit code after optimizing shaded region in Fig. 9 (j) using matrix equivalence.

with the sequence of CNOT and CZ gates appearing after it separately. Before the optimization process starts, we need to convert it to sub-circuits consisting of only CNOT gates. Each of the sub-circuits is separated by H gates. Next, the two strategies of optimization, i.e., equivalence rules and matrix equivalence can be applied sequentially to optimize the circuit. The complete optimization procedure for a stabilizer encoder circuit can be represented as follows:

- **Step 1:** First, we divide the circuit into a number of sub-circuits, each containing a H gate and a sequence of CNOT and CZ gates, corresponding to the stabilizer it encodes. We ensure that the CZ gates are to the left, mindful of introducing a Z gate after H if the number of CNOT-CZ commutations required is odd.
- **Step 2:** Next, we flip control and targets of each of the CZ gates. This will ensure that when the CZ gates are converted to CNOT gates, the number of H gates remains constant.
- **Step 3:** Subsequently, each of the CZ gates are converted to CNOT gates using Rule 3 in Fig. 5. Each of the new H gate introduced in the process is annihilated by an already existing H gate, keeping the count of CNOT gates same.
- **Step 4:** Now, our encoder contains CNOT gates as the only multi-qubit gates. Each of the sub-circuits of CNOT gates is inspected separately and equivalence rules are applied wherever possible.
- **Step 5:** Next, each of the sub-circuit is optimized using matrix equivalence by following a greedy algorithm, where the row transformations leading to reduction in number of 1's in the matrix by at least two are favoured.

IV. Optimized encoder circuit design for Steane code and a 13-qubit code

Steane code [42] is a CSS code which uses the Hamming $[7, 4, 3]$ code and the dual of the Hamming code, i.e., the $[7, 3, 4]$ code to correct bit flip and phase flip errors respectively. The $[7, 4, 3]$ Hamming code contains its dual, and thus can be used in the CSS framework to obtain a quantum ECC. One logical qubit is encoded into seven physical qubits, thus enabling the Steane code to detect and correct a single qubit error. In stabilizer framework, the Steane code is represented by six generators as shown below:

M_1	X	X	X	X	I	I	I
M_2	X	X	I	I	X	X	I
M_3	X	I	X	I	X	I	X
M_4	Z	Z	Z	Z	I	I	I
M_5	Z	Z	I	I	Z	Z	I
M_6	Z	I	Z	I	Z	I	Z

We designed a Steane code encoder using Algorithm 1 in [30] as shown in Fig. 11 (a). Using the equivalence rules the total number of CNOT gates was reduced by 1 as shown in Fig. 11 (b).

New stabilizer codes can be generated by merging available stabilizer codes. Such a code can be created by pasting a five-qubit code [[5,1,3]] and an eight-qubit code [[8,3,3]] as described in [10]. The new 13-qubit code [[13,7,3]] has a rate of $7/13$. It can be represented by the following stabilizers:

M_1	X	X	X	X	X	X	X	X	I	I	I	I	I
M_2	Z	Z	Z	Z	Z	Z	Z	Z	I	I	I	I	I
M_3	I	I	I	I	I	I	I	I	X	Z	Z	X	I
M_4	I	X	I	X	Y	Z	Y	Z	I	X	Z	Z	X
M_5	I	X	Z	Y	I	X	Z	Y	X	I	X	Z	Z
M_6	I	Y	X	Z	X	Z	I	Y	Z	X	I	X	Z

Using Algorithm 1 in [30], the encoder circuit for the 13-qubit code was designed using CNOT, CZ, and CY gates as shown in Fig. 12. Details of the construction of the encoder circuit is given in Section IV of the Supplementary Information.

Converting the CY gates to CNOT and CZ gates according to Rule 12, we get the circuits in Fig. 13. It uses 50 2-qubit gates and 5 single qubit gates.

We optimized the encoder circuit according to the procedure described in Section IV to reduce the number of gates to 42 CNOT gates and 5 H gates. The resulting circuit is presented in Fig. 14.

V. Results

The combined encoder and decoder circuits for the 8-qubit code, Steane code, and the 13-qubit code were simulated using IBM Qiskit. Circuits designed from Algorithm 1 in [30] as well as the optmized circuits derived from those were simulated using IBM Qiskit. Errors were introduced at different positions to test for correctability. The syndrome table was constructed in a similar way as in [30]. The syndromes were found to be an exact match depending on the type and location of error introduced. The syndrome tables for 13-qubit code and Steane code can also be derived using the same procedure as in [30].

Another important parameter to measure the efficiency of the quantum circuits is the number of single and multiple qubit gates used by the quantum circuits. We list the number of gates used in the quantum circuits presented in this paper in Table I. In the second column, we observe that the eight-qubit encoder designed using Algorithm 1 in [30] requires 8 CNOT gates, 7 CY gates, and 5 CZ gates, thus using a total of 20 2-qubit gates. In practical applications, one may need to use a combination of CZ and CNOT gates to obtain CY gates. The third column shows the number of gates used in a circuit using CZ and CNOT gates. We observe that it requires a total of 27 2-qubit gates (15 CNOT and 12 CZ). For practical considerations, only one type of 2-qubit gate may be available. To achieve this, the circuit was further optimized using the equivalence rules to reduce the number of gates to 18 CNOT gates and 4 H gates, as reported in column 6 of Table I. The reduction in 2-qubit gates from 27 gates to 18 gates is quite significant[1]. Columns 5 and 6 show the number of gates used in Steane code [42] encoder and its optimized version. The number of CNOT gates is reduced from 11 to 10. We also report the number of gates used in the 13-qubit encoder and its optimized version. The proposed approach achieves a 16

[1]It should be noted that CY gates are usually designed as a combination of CNOT and CZ gates. Thus, a single CY gate is equivalent to two 2-qubit gates.

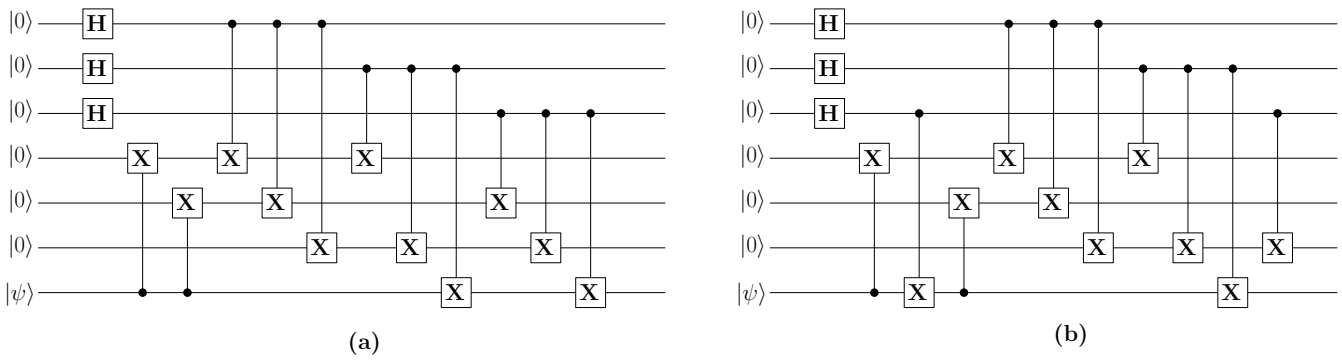

Fig. 11. (a) Encoder for Steane code using Algorithm 1 in [30]. (b) Optimized encoder for the Steane code using equivalence rules.

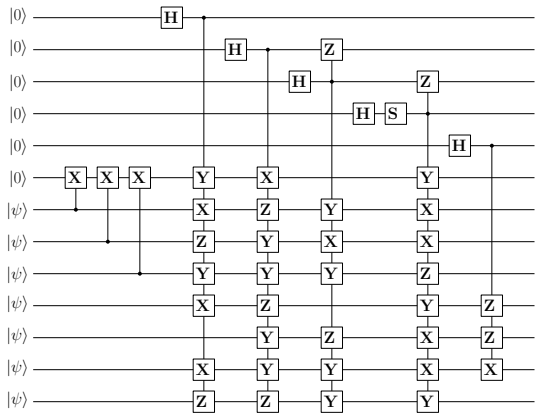

Fig. 12. 13-qubit encoder designed using Algorithm 1 in [30]. CNOT and CZ gates were used as 2-qubit gates.

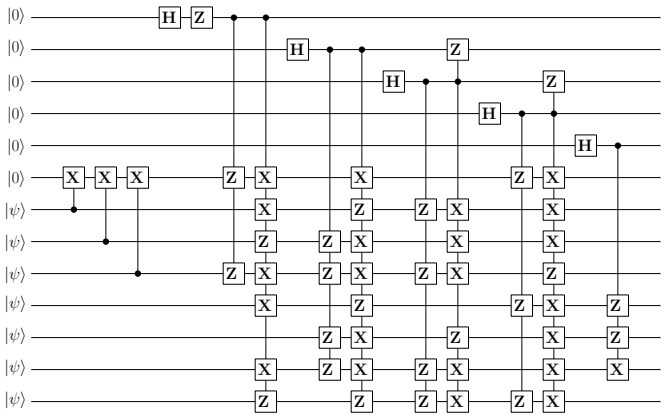

Fig. 13. 13-qubit encoder designed using Algorithm 1 in [30]. CNOT and CZ gates were used as 2-qubit gates.

% reduction in the number of 2-qubit gates, from 26 CNOT and 24 CZ gates to 42 CNOT gates.

In [41], an encoding rule was formulated by observing all the 2^k different codewords for the eight-qubit code. For quantum codes where k is significantly larger, the method in [41] would be very complex because one has to figure out an encoding rule by observing 2^k codewords. Additionally, the method in [41] requires more than twice the number of gates than the eight-qubit code encoder circuit proposed in this paper. Furthermore, the circuit in [41] uses an array of different types of multi-qubit gates which may not be desirable for practical considerations. Compared to [41] which requires 14 single qubit gates, 33 2-qubit gates, and 6 3-qubit gates, the proposed eight-qubit encoder requires only 18 CNOT gates and 4 H gates. The syndrome measurement circuit in this paper requires slightly less number of gates than [41].

VI. Conclusions

This paper has presented a procedure for optimization of encoder circuits for stabilizer codes in terms of the number of gates used. The procedure was used to design optimized encoder and decoder circuits for an eight-qubit code. The circuits were verified using IBM Qiskit. The encoder circuits designed using the proposed procedure result in a reduction in the number of gates by a factor of 2 for the eight-qubit code, compared to prior designs. We also present optimized encoder circuits for the Steane code and a 13-qubit code.

In this paper, we optimized the encoder circuits such that they contain CNOT gates as the only 2-qubit gates. However, some practical quantum computers may have CZ gates as the native 2-qubit gates. For translating the encoder circuits to such computers, the optimization procedure would be more challenging. Since CNOT gates are not symmetric, new H gates would be required at the target, and thus the number of H gates may increase significantly. This problem does not arise for CZ to CNOT conversion, since CZ gates are symmetric. Investigation into optimization approaches to design quantum circuits that contain only CZ gates is a topic of further research.

Another important research area is the investigation into fault-tolerant circuits. For example, in [47], the authors investigate design of fault-tolerant circuits for the Steane code. Design of optimized circuits that are fault-tolerant is a topic that requires further research. Design of near-optimal quantum circuits that satisfy physical constraints of the quantum computer, such as nearest neighbor compliance or geometry of the placement of the qubits, is also a topic that needs to be explored further.

For larger code lengths involving significantly higher number of qubits, it is difficult to optimize the circuits manually. Therefore, future efforts will be directed towards developing an automated tool to optimize the quantum circuits. Designing encoder-decoder circuits for EA stabilizer codes is also a topic

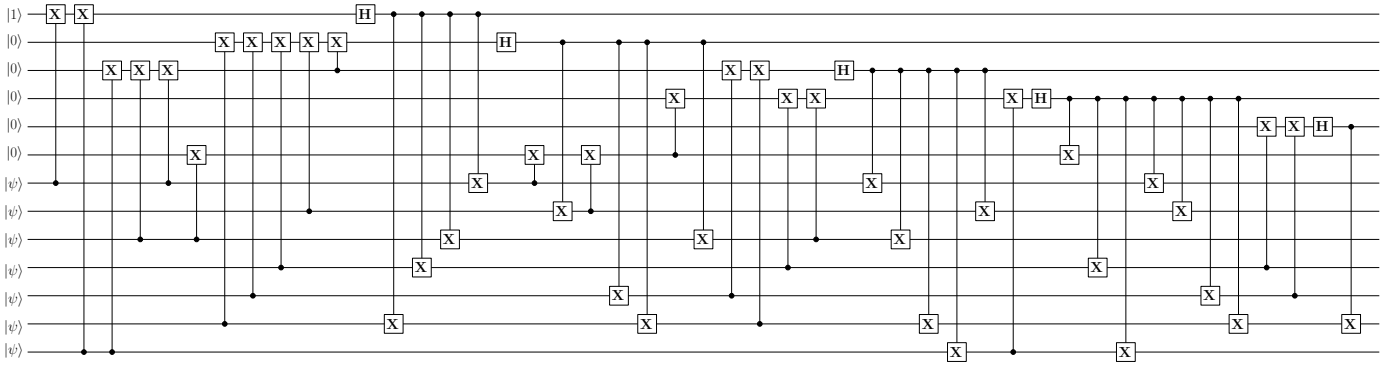

Fig. 14. Optimized 13-qubit encoder using 42 CNOT gates and 5 H gates.

TABLE I
RESOURCE UTILIZATION SUMMARY FOR THE VARIOUS DESIGNED QUANTUM CIRCUITS IN TERMS OF NUMBER OF GATES USED.

Parameters	H gates	X gates	S gates	Z gates	CNOT gates	CY gates	CZ gates	CCNOT gates
Eight qubit encoder [41]	4	10	0	0	24	0	9	6
Eight qubit encoder using [10] (using CNOT and CZ gates)	4	0	0	4	15	0	12	0
Proposed eight qubit encoder (optimized using equivalence rules and matrix equivalence)	**4**	**0**	**0**	**0**	**18**	**0**	**0**	**0**
Steane code encoder	3	0	0	0	11	0	0	0
Optimized Steane code encoder	**3**	**0**	**0**	**0**	**10**	**0**	**0**	**0**
13-qubit code encoder	5	0	0	1	26	0	24	0
Optimized 13-qubit code encoder	**5**	**0**	**0**	**0**	**42**	**0**	**0**	**0**

of further research. Additionally, future efforts need to be directed towards design of quantum circuits for more complex quantum ECCs such as BCH, LDPC, and polar codes.

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. IEEE, 1994, pp. 124–134.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[4] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[5] D. Dieks, "Communication by EPR devices," *Physics Letters A*, vol. 92, no. 6, pp. 271–272, 1982.

[6] A. Kumar Pati and S. L. Braunstein, "Impossibility of deleting an unknown quantum state," *Nature*, vol. 404, no. 6774, pp. 164–165, 2000.

[7] H. Barnum, C. M. Caves, C. A. Fuchs, R. Jozsa, and B. Schumacher, "Noncommuting mixed states cannot be broadcast," *Physical Review Letters*, vol. 76, no. 15, p. 2818, 1996.

[8] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical review A*, vol. 52, no. 4, p. R2493, 1995.

[9] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum hamming bound," *Physical Review A*, vol. 54, no. 3, p. 1862, 1996.

[10] ——, *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

[11] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, p. 1098, 1996.

[12] A. M. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, no. 5, p. 793, 1996.

[13] T. Brun, I. Devetak, and M.-H. Hsieh, "Correcting quantum errors with entanglement," *science*, vol. 314, no. 5798, pp. 436–439, 2006.

[14] C.-Y. Lai and T. A. Brun, "Entanglement increases the error-correcting ability of quantum error-correcting codes," *Physical Review A*, vol. 88, no. 1, p. 012320, 2013.

[15] M. M. Wilde, "Quantum coding with entanglement," Ph.D. dissertation, University of Southern California, 2008.

[16] M.-H. Hsieh, T. A. Brun, and I. Devetak, "Entanglement-assisted quantum quasicyclic low-density parity-check codes," *Physical Review A*, vol. 79, no. 3, p. 032340, 2009.

[17] M. M. Wilde, *Quantum information theory*. Cambridge university press, 2013.

[18] L. Gyongyosi, S. Imre, and H. V. Nguyen, "A survey on quantum channel capacities," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1149–1205, 2018.

[19] N. Delfosse and G. Zémor, "Upper bounds on the rate of low density stabilizer codes for the quantum erasure channel," *arXiv preprint arXiv:1205.7036*, 2012.

[20] M. Grassl, W. Geiselmann, and T. Beth, "Quantum Reed Solomon codes," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: 13th International Symposium, AAECC-13 Honolulu, Hawaii, USA, November 15–19, 1999 Proceedings 13*. Springer, 1999, pp. 231–244.

[21] S. A. Aly, "Asymmetric quantum BCH codes," in *2008 International Conference on Computer Engineering & Systems*. IEEE, 2008, pp. 157–162.

[22] G. G. La Guardia, "Asymmetric quantum Reed-Solomon and generalized Reed-Solomon codes," *Quantum Information Processing*, vol. 11, pp. 591–604, 2012.

[23] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, 2013.

[24] S. Bravyi and M. B. Hastings, "Homological product codes," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014, pp. 273–282.

[25] M. B. Hastings, "Weight reduction for quantum codes," *arXiv preprint arXiv:1611.03790*, 2016.

[26] S. Evra, T. Kaufman, and G. Zémor, "Decodable quantum LDPC codes beyond the n distance barrier using high-dimensional expanders," *SIAM Journal on Computing*, no. 0, pp. FOCS20–276, 2022.

[27] F. Dupuis, A. Goswami, M. Mhalla, and V. Savin, "Purely quantum polar codes," in *2019 IEEE Information Theory Workshop (ITW)*. IEEE, 2019, pp. 1–5.

[28] P. J. Nadkarni, "Entanglement-assisted additive qudit stabilizer codes," Ph.D. dissertation, Indian Institute of Science, 2021.

[29] D. Chandra, Z. B. K. Egilmez, Y. Xiong, S. X. Ng, R. G. Maunder, and L. Hanzo, "Universal decoding of quantum stabilizer codes via classical guesswork," *IEEE Access*, vol. 11, pp. 19 059–19 072, 2023.

[30] A. Mondal and K. K. Parhi, "Quantum circuits for stabilizer error correcting codes: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 24, no. 1, pp. 33–51, 2024.

[31] X. Zhou, D. W. Leung, and I. L. Chuang, "Methodology for quantum logic gate construction," *Physical Review A*, vol. 62, no. 5, p. 052316, 2000.

[32] N. D. Mermin, "From classical state swapping to quantum teleportation," *Physical Review A*, vol. 65, no. 1, p. 012320, 2001.

[33] ——, "Deconstructing dense coding," *Physical Review A*, vol. 66, no. 3, p. 032308, 2002.

[34] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, "Quantum circuit simplification and level compaction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 436–444, 2008.

[35] J. C. Garcia-Escartin and P. Chamorro-Posada, "Equivalent quantum circuits," *arXiv preprint arXiv:1110.2998*, 2011.

[36] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3305–3310, 2017.

[37] K. Wright, K. M. Beck, S. Debnath, J. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. Pisenti, M. Chmielewski, C. Collins *et al.*, "Benchmarking an 11-qubit quantum computer," *Nature communications*, vol. 10, no. 1, p. 5464, 2019.

[38] M. Bataille, "Quantum circuits of cnot gates: optimization and entanglement," *Quantum Information Processing*, vol. 21, no. 7, p. 269, 2022.

[39] G. Alber, T. Beth, M. Horodecki, P. Horodecki, R. Horodecki, M. Rotteler, H. Weinfurter, R. Werner, A. Zeilinger, T. Beth *et al.*, "Quantum algorithms: Applicable algebra and quantum physics," *Quantum information: an introduction to basic theoretical concepts and experiments*, pp. 96–150, 2001.

[40] K. N. Patel, I. L. Markov, and J. P. Hayes, "Optimal synthesis of linear reversible circuits." *Quantum Inf. Comput.*, vol. 8, no. 3, pp. 282–294, 2008.

[41] P. Dong, J. Liu, and Z.-L. Cao, "Efficient quantum circuit for encoding and decoding of the [[8, 3, 5]] stabilizer code," *International Journal of Theoretical Physics*, vol. 52, pp. 1274–1281, 2013.

[42] A. Steane, "Multiple-particle interference and quantum error correction," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 452, no. 1954, pp. 2551–2577, 1996.

[43] "Ibm quantum," *https://quantum-computing.ibm.com/*.

[44] A. M. Steane, "A tutorial on quantum error correction," *Quantum Computers, Algorithms and Chaos*, pp. 1–32, 2006.

[45] J. Roffe, "Quantum error correction: an introductory guide," *Contemporary Physics*, vol. 60, no. 3, pp. 226–245, 2019.

[46] D. Janzing, P. Wocjan, and T. Beth, ""identity check" is qma-complete," *arXiv: Quantum Physics*, 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:118913526

[47] H. Goto, "Minimizing resource overheads for fault-tolerant preparation of encoded states of the steane code," *Scientific reports*, vol. 6, no. 1, p. 19578, 2016.

Arijit Mondal received the B.Tech. degree from National Institute of Technology, Durgapur, followed by M.E. and Ph.D from Indian Institute of Science. His research interests include quantum error correction code circuits and VLSI architectures for error correction codes. He is currently working as a Posdoctoral Associate at the University of Minnesota.

Keshab K. Parhi (Fellow, IEEE) received the B.Tech. degree from the Indian Institute of Technology (IIT), Kharagpur, in 1982, the M.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1984, and the Ph.D. degree from the University of California, Berkeley, in 1988. He has been with the University of Minnesota, Minneapolis, since 1988, where he is currently the Erwin A. Kelen Chair and a Distinguished McKnight University Professor in the Department of Electrical and Computer Engineering. He has published over 700 papers, is the inventor of 36 patents, and has authored the textbook VLSI Digital Signal Processing Systems (Wiley, 1999). His current research addresses VLSI architecture design of machine learning and signal processing systems, hardware security, and data-driven neuroengineering and neuroscience. Dr. Parhi is the recipient of numerous awards including the 2017 Mac Van Valkenburg award and the 2012 Charles A. Desoer Technical Achievement award from the IEEE Circuits and Systems Society, the 2003 IEEE Kiyo Tomiyasu Technical Field Award, and a Golden Jubilee medal from the IEEE Circuits and Systems Society in 2000. He served as the Editor-in-Chief of the IEEE Trans. Circuits and Systems, Part-I during 2004 and 2005. He currently serves as the Editor-in-Chief of the IEEE Circuits and Systems Magazine. He is a Fellow of the American Association for the Advancement of Science (AAAS), the Association for Computing Machinery (ACM), the American Institute of Medical and Biological Engineering (AIMBE), and the National Academy of Inventors (NAI).

Supplementary Information: Optimization of Quantum Circuits for Stabilizer Codes

Arijit Mondal and Keshab K. Parhi, *Fellow, IEEE*
Email: {monda109, parhi}@umn.edu
Department of Electrical and Computer Engineering, University of Minnesota

I. Systematic procedure for encoder design for a stabilizer code

The procedure to design encoding circuits for stabilizer codes is as follows [1]:

Step 1: The stabilizers are written in a matrix form using binary vector space formalism. Let the parity chek matrix thus obtained be H_q.

Step 2: Our aim is to bring H_q to the standard form H_s below:

$$H_s = \left[\begin{array}{ccc|ccc} I_1 & A_1 & A_2 & B & C_1 & C_2 \\ 0 & 0 & 0 & D & I_2 & E \end{array}\right] \quad (1)$$

where, I_1 and B are $r \times r$ matrices. 'r' is the rank of the X portion of H_s. A_1 and C_1 are $r \times (n-k-r)$ matrices. A_2 and C_2 are $r \times k$ matrices. D is a $(n-k-r) \times r$ matrix. I_2 is a $(n-k-r) \times (n-k-r)$ matrix. E is a $(n-k-r) \times k$ matrix. I_1 and I_2 are identity matrices.

H_q is converted to standard form H_s using Gaussian elimination [1]. The logical operators $\overline{X}$ and $\overline{Z}$ can be written as

$$\overline{X} = \left[\begin{array}{cccc|ccc} 0 & U_2 & U_3 & & V_1 & 0 & 0 \end{array}\right] \quad (2)$$

$$\overline{Z} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & & V_1' & 0 & V_3' \end{array}\right] \quad (3)$$

where $U_2 = E^T$, $U_3 = I_{k\times k}$, $V_1 = E^T C_1^T + C_2^T$, $V_1' = A_2^T$, and $V_3' = I_{k\times k}$.

Given the parity check matrix in standard form H_s and $\overline{X}$, the encoding operation for a stabilizer code can be written as,

$$|c_1 c_2 \cdots c_k\rangle = \overline{X}_1^{c_1} \overline{X}_2^{c_2} \cdots \overline{X}_k^{c_k} \left(\sum_{M\in S} M\right) |00\cdots 0\rangle \quad (4)$$

$$= \overline{X}_1^{c_1} \overline{X}_2^{c_2} \cdots \overline{X}_k^{c_k} (I+M_1)(I+M_2)\cdots (I+M_{n-k})|00\cdots 0\rangle. \quad (5)$$

There are a total of n qubits. Place qubits initialized to $|0\rangle$ at qubit positions $i = 1$ to $i = n-k$. Place the qubits to be encoded at positions $i = n-k+1$ to $i = n$.

We observe the following from H_s and $\overline{X}$:

- We know that a particular logical operator $\overline{X}_i$ is applied only if the qubit at i^{th} position is $|1\rangle$. Thus, applying $\overline{X}_i$ controlled at i^{th} qubit encodes $\overline{X}_i$.
- The $\overline{X}$ operators consist of products of only Zs for the first r qubits. For the rest of the qubits, $\overline{X}$ consists of products of Xs only. We know that Z acts trivially on $|0\rangle$. Since the first r qubits are initialized to $|0\rangle$, we can ignore all the Zs in $\overline{X}$.
- The first r generators in H_s apply only a single bit flip to the first r qubits. This implies that when $I + M_i$ is applied, the resulting state would be a sum of $|0\rangle$ and $|1\rangle$ for the i^{th} qubit. This corresponds to applying H gates to the first r qubits, which puts each of the r qubits in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- If we apply M_i conditioned on qubit i, it implies the application of $I + M_i$. The reason is as follows. When the control qubit i is $|1\rangle$, M_i needs to be applied to the combined qubit state. Since the qubit i suffers a bit flip X only by the stabilizer M_i, it is already in flipped state when it is $|1\rangle$. Thus, only the rest of the operators in M_i need to be applied. However, there would be an issue if $H_{s_{(i,i+n)}}$ is not 0, i.e., there is a Y instead of X. In that case, adding an S gate after the H gate resolves the issue.

Step 3: The observations in Step 2 were used by authors in [2] to devise an algorithm as shown in Algorithm 1 to design the encoding circuit.

To design an encoder circuit that uses CNOT and CZ gates only as multiple qubit gates, we can use rule 12 (Fig. 5, main paper) to convert the CY gates to CNOT and CZ gates. This would lead to introduction of single qubit S gates.

II. Encoder and syndrome measurement circuit design for eight-qubit code

A. Encoder design for the eight-qubit code

The stabilizers for the eight-qubit code can be written in binary vector space formalism. This corresponds to Step 1 in Section I.

$$H_q = \left[\begin{array}{c|c} 11111111 & 00000000 \\ 00000000 & 11111111 \\ 01011010 & 00001111 \\ 01010101 & 00110011 \\ 01101001 & 01010101 \end{array}\right] \quad (6)$$

Step 2 involves converting H_q into standard form H_s as in Equation 1. First, we push the 2^{nd} row to the 5^{th} position and push the rows below up by one step. We also swap the 4^{th} column with the 5^{th} column (equivalent to swapping fourth and fifth qubit position), which would require the operation of swapping 12^{th} column with 13^{th} column as well. Performing the above operations, we get

Algorithm 1: Algorithm to generate encoding circuit from H_s and $\overline{X}$ (n = number of physical qubits, k = number of logical qubits, r = rank of X-portion of H_s).

Data: H_s, $\overline{X}$
Result: Encoding circuit
for $i = 1$ **to** k **do**
  if $\overline{X}_{i,i+n-k} == 1$ **then**
    Place controlled dot at qubit $i + n - k$
  end
  for $j = 1$ **to** n **do**
    if $i + n - k \neq j$ **then**
      if $\overline{X}_{i,j} == 1$ **then**
        Place X gate at qubit j controlled at qubit $i + n - k$
      end
    end
  end
end
for $i = 1$ **to** r **do**
  if $H_{s_{(i,i+n)}} == 0$ **then**
    Place H gate followed by controlled dot at qubit i
  else
    Place H gate followed by S gate followed by controlled dot at qubit i
  end
  for $j = 1$ **to** n **do**
    if $i \neq j$ **then**
      if $H_{s_{(i,j)}} == 1$ && $H_{s_{i,j+n}} == 0$ **then**
        Place X gate on qubit j with control at qubit i
      end
      if $H_{s_{(i,j)}} == 0$ && $H_{s_{i,j+n}} == 1$ **then**
        Place Z gate on qubit j with control at qubit i
      end
      if $H_{s_{(i,j)}} == 1$ && $H_{s_{i,j+n}} == 1$ **then**
        Place Y gate on qubit j with control at qubit i
      end
    end
  end
end

$$H_q = \left[\begin{array}{cc} 11111111 & 00000000 \\ 01011010 & 00010111 \\ 01001101 & 00101011 \\ 01110001 & 01001101 \\ 00000000 & 11111111 \end{array}\right] \tag{7}$$

We perform the operation $R_4 \rightarrow R_4 + R_2$, followed by $R_2 \rightarrow R_2 + R_3$. Next, we swap R_2 with R_3, and R_3 with R_4. Finally, performing the operation $R_1 \rightarrow R_1 + R_2 + R_3 + R_4$, we get the standard form as

$$H_s = \left[\begin{array}{cc} 10001110 & 01001101 \\ 01001101 & 00101011 \\ 00101011 & 01011010 \\ 00010111 & 00111100 \\ 00000000 & 11111111 \end{array}\right] \tag{8}$$

From the standard form H_s, the stabilizers which will be used for syndrome measurement are as follows.

$$\begin{array}{c|cccccccc} M_1 & X & Z & I & I & Y & Y & X & Z \\ M_2 & I & X & Z & I & Y & X & Z & Y \\ M_3 & I & Z & X & Z & Y & I & Y & X \\ M_4 & I & I & Z & Y & Z & Y & X & X \\ M_5 & Z & Z & Z & Z & Z & Z & Z & Z \end{array}.$$

From H_s, we can evaluate the following:

$$I_1 = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}, A_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, A_2 = \begin{bmatrix} 110 \\ 101 \\ 011 \\ 111 \end{bmatrix},$$
$$B = \begin{bmatrix} 0100 \\ 0010 \\ 0101 \\ 0011 \end{bmatrix}, C_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, C_2 = \begin{bmatrix} 101 \\ 011 \\ 010 \\ 100 \end{bmatrix},$$
$$D = [1111], I_2 = 1, E = [111] \tag{9}$$

From the above,

$$\overline{X} = \left[\begin{array}{cc} 00001100 & 01100000 \\ 00001010 & 10010000 \\ 00001001 & 00110000 \end{array}\right] \tag{10}$$

$$\overline{Z} = \left[\begin{array}{cc} 00000000 & 11010100 \\ 00000000 & 10110010 \\ 00000000 & 01110001 \end{array}\right] \tag{11}$$

Thus, the logical $\overline{X}$ and $\overline{Z}$ operators are

$$\begin{array}{c|cccccccc} \overline{X_1} & I & Z & Z & I & X & X & I & I \\ \overline{X_2} & Z & I & I & Z & X & I & X & I \\ \overline{X_3} & I & I & Z & Z & X & I & I & X \\ \overline{Z_1} & Z & Z & I & Z & I & Z & I & I \\ \overline{Z_2} & Z & I & Z & Z & I & I & Z & I \\ \overline{Z_3} & I & Z & Z & Z & I & I & I & Z \end{array}.$$

Using H_s and the $\overline{X}$ operators, the various encoded states can be evaluated using the following equation

$$|c_1c_2c_3\rangle = \overline{X}_1^{c_1}\overline{X}_2^{c_2}\overline{X}_3^{c_3}\left(\sum_{M\in S} M\right)|00000000\rangle \tag{12}$$
$$= \overline{X}_1^{c_1}\overline{X}_2^{c_2}\overline{X}_3^{c_3}(I + M_1)(I + M_2)$$
$$(I + M_3)(I + M_4)|00000000\rangle \tag{13}$$

Since M_5 consists of tensor products of Z Pauli operators, and since Z acts trivially on $|0\rangle$, $I + M_5$ does not change the state $|00000000\rangle$. Thus, we can ignore M_5.

Following the procedure in Section I, the 3 qubits to be encoded are placed at positions $n-2$, $n-1$, and n, followed

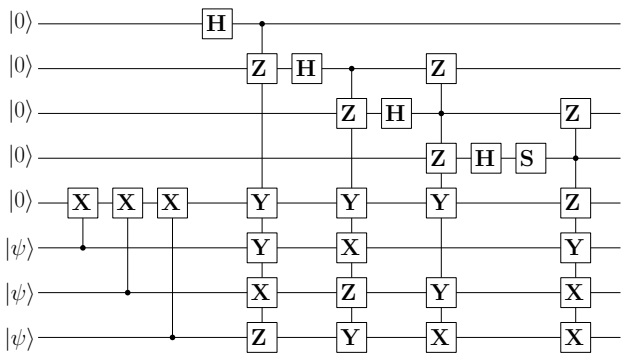

Fig. S1. Encoder for the eight-qubit code.

by the rest of the qubits initialized to the state $|0\rangle$. Next, the logical operators are encoded according the Algorithm 1. Thereafter, the stabilizers corresponding to the rows of standard form of the parity check matrix H_s are applied according the Algorithm 1. The encoder circuit thus designed is shown in Fig. S1.

B. Syndrome measurement circuit and error corrector

The syndromes are unique as shown in Table S1. Each qubit in the eight-qubit code can be affected by three kind of errors, namely X, Y, and Z errors. The qubit indices have been represented as $Q_1 - Q_8$. So, there are 24 different types of single qubit errors possible, each of which gives a different syndrome as shown in Table S1. Each bit in the 5-bit syndrome represents whether the corresponding stabilizer commutes with the error. If it commutes, the bit is 0, else it is 1. It should also be observed that each syndrome is unique as shown in Table S1.

The syndrome measurement circuit is shown in Fig. S2. Five ancilla qubits are used to measure each of the six stabilizers. Measurement of the ancilla qubits gives the syndrome. Depending on the syndrome, appropriate error correction can be performed by using suitable X, Z, or Y gate on the appropriate qubit. A syndrome measurement of 00000 implies that no error has occurred. It should also be noted that any 5 bit syndrome other than the syndromes mentioned in Table S1 signifies more than a single qubit error which cannot be corrected.

III. Mathematical verification of Algorithm 1 through eight-qubit code

In this Appendix, we verify that Algorithm 1 results in the same state as given by Equation (13) for the eight-qubit code. Out of the eight possible states, we only verify the application of the algorithm for the state $\overline{000}$. The algorithm can be verified for other states using suitable $\overline{X}$ operators. operators

From Equation (13) we have,

$$|\overline{000}\rangle = (I + M_1)(I + M_2)(I + M_3)(I + M_4)|00000000\rangle \tag{14}$$

Expanding, we get

TABLE S1
Syndrome table for the eight-qubit code.

Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	M_1	M_2	M_3	M_4	M_5	Decimal value
X	I	I	I	I	I	I	I	0	0	0	0	1	1
Z	I	I	I	I	I	I	I	1	0	0	0	0	16
Y	I	I	I	I	I	I	I	1	0	0	0	1	17
I	X	I	I	I	I	I	I	1	0	1	0	1	21
I	Z	I	I	I	I	I	I	0	1	0	0	0	8
I	Y	I	I	I	I	I	I	1	1	1	0	1	29
I	I	X	I	I	I	I	I	0	1	0	1	1	11
I	I	Z	I	I	I	I	I	0	0	1	0	0	4
I	I	Y	I	I	I	I	I	0	1	1	1	1	15
I	I	I	X	I	I	I	I	0	0	1	1	1	7
I	I	I	Z	I	I	I	I	0	0	0	1	0	2
I	I	I	Y	I	I	I	I	0	0	1	0	1	5
I	I	I	I	X	I	I	I	1	1	1	1	1	31
I	I	I	I	Z	I	I	I	1	1	1	0	0	28
I	I	I	I	Y	I	I	I	0	0	0	1	1	3
I	I	I	I	I	X	I	I	1	0	0	1	1	19
I	I	I	I	I	Z	I	I	1	1	0	1	0	26
I	I	I	I	I	Y	I	I	0	1	0	0	1	9
I	I	I	I	I	I	X	I	0	1	1	0	1	13
I	I	I	I	I	I	Z	I	1	0	1	1	0	22
I	I	I	I	I	I	Y	I	1	1	0	1	1	27
I	I	I	I	I	I	I	X	1	1	0	0	1	25
I	I	I	I	I	I	I	Z	0	1	1	1	0	14
I	I	I	I	I	I	I	Y	1	0	1	1	1	23
I	I	I	I	I	I	I	I	0	0	0	0	0	0

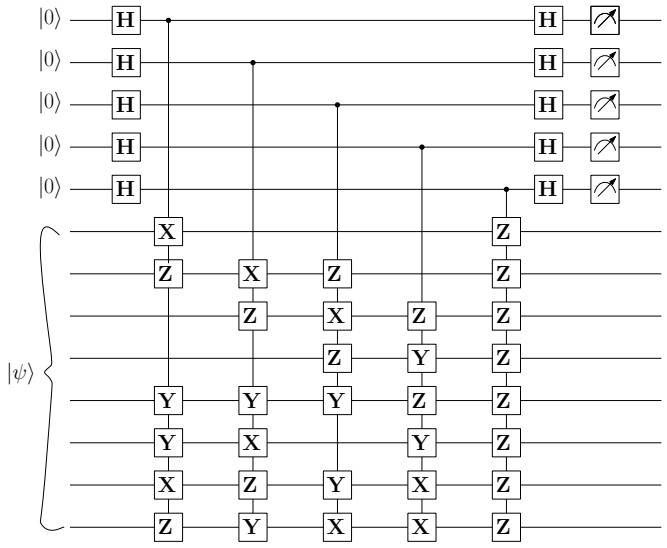

Fig. S2. Syndrome measurement circuit for the eight-qubit code.

$$
\begin{aligned}
|\overline{000}\rangle =&|00000000\rangle + M_1|00000000\rangle + M_2|00000000\rangle \\
&+ M_3|00000000\rangle + M_4|00000000\rangle + M_1M_2|00000000\rangle \\
&+ M_1M_3|00000000\rangle + M_1M_4|00000000\rangle \\
&+ M_2M_3|00000000\rangle + M_2M_4|00000000\rangle \\
&+ M_3M_4|00000000\rangle + M_1M_2M_3|00000000\rangle \\
&+ M_1M_2M_4|00000000\rangle + M_1M_3M_4|00000000\rangle \\
&+ M_2M_3M_4|00000000\rangle + M_1M_2M_3M_4|00000000\rangle
\end{aligned} \tag{15}
$$

$$
\begin{aligned}
=&\frac{1}{4}(|00000000\rangle - |00010111\rangle - |00101011\rangle \\
&+ |00111100\rangle - |01001101\rangle + |01011010\rangle \\
&+ |01100110\rangle - |01110001\rangle - |10001110\rangle \\
&+ |10011001\rangle + |10100101\rangle - |10110010\rangle \\
&+ |11000011\rangle - |11010100\rangle - |11101000\rangle \\
&+ |11111111\rangle)
\end{aligned} \tag{16}
$$

We now verify that Algorithm 1, and the circuit in Fig. S1 result in the same state as in Equation (16). We have the initial state $\psi = |00000000\rangle$.

Step A: Applying H gate on qubit 1, we have

$$
|\psi_1\rangle = \frac{1}{\sqrt{2}}(|00000000\rangle + |10000000\rangle) \tag{17}
$$

Step B: Applying M_1 controlled at qubit 1 we have,

$$
\begin{aligned}
|\psi_2\rangle =&\frac{1}{\sqrt{2}}(|00000000\rangle + (i \cdot i)|10001110\rangle) \\
=&\frac{1}{\sqrt{2}}(|00000000\rangle - |10001110\rangle)
\end{aligned} \tag{18}
$$

Step C: Applying H gate on qubit 2, we have

$$
\begin{aligned}
|\psi_3\rangle =&\frac{1}{2}(|00000000\rangle + |01000000\rangle - |10001110\rangle \\
&- |11001110\rangle)
\end{aligned} \tag{19}
$$

Step D: Applying M_2 controlled at qubit 2 we have,

$$
\begin{aligned}
|\psi_4\rangle =&\frac{1}{2}(|00000000\rangle + (i \cdot i)|01001101\rangle - |10001110\rangle \\
&- (-i \cdot -1 \cdot i)|11000011\rangle) \\
=&\frac{1}{2}(|00000000\rangle - |01001101\rangle - |10001110\rangle \\
&+ |11000011\rangle)
\end{aligned} \tag{20}
$$

Step E: Applying H gate on qubit 3, we have

$$
\begin{aligned}
|\psi_5\rangle =&\frac{1}{2\sqrt{2}}(|00000000\rangle + |00100000\rangle - |01001101\rangle \\
&- |01101101\rangle - |10001110\rangle - |10101110\rangle \\
&+ |11000011\rangle + |11100011\rangle)
\end{aligned} \tag{21}
$$

Step F: Applying M_3 controlled at qubit 3 we have,

$$
\begin{aligned}
|\psi_6\rangle =&\frac{1}{2\sqrt{2}}(|00000000\rangle + (i \cdot i)|00101011\rangle - |01001101\rangle \\
&- (-1 \cdot -i \cdot i)|01100110\rangle - |10001110\rangle \\
&- (-i \cdot -i)|10100101\rangle + |11000011\rangle \\
&+ (-1 \cdot i \cdot -i)|11101000\rangle) \\
=&\frac{1}{2\sqrt{2}}(|00000000\rangle - |00101011\rangle - |01001101\rangle \\
&+ |01100110\rangle - |10001110\rangle + |10100101\rangle \\
&+ |11000011\rangle - |11101000\rangle)
\end{aligned} \tag{22}
$$

Step G: Applying H gate followed by S gate on qubit 4, we have

$$
\begin{aligned}
|\psi_7\rangle =&\frac{1}{4}(|00000000\rangle + i|00010000\rangle - |00101011\rangle \\
&- i|00111011\rangle - |01001101\rangle - i|01011101\rangle \\
&+ |01100110\rangle + i|01110110\rangle - |10001110\rangle \\
&- i|10011110\rangle + |10100101\rangle + i|10110101\rangle \\
&+ |11000011\rangle + i|11010011\rangle - |11101000\rangle \\
&- i|11111000\rangle)
\end{aligned} \tag{23}
$$

Step H: Applying M_4 controlled at qubit 4 we have,

$$
\begin{aligned}
|\psi_8\rangle =&\frac{1}{4}(|00000000\rangle + (i \cdot i)|00010111\rangle - |00101011\rangle \\
&(-i \cdot -1 \cdot -1 \cdot i)|00111100\rangle - |01001101\rangle \\
&- (i \cdot -1 \cdot -i)|01011010\rangle + |01100110\rangle \\
&+ (i \cdot -1 \cdot -i)|01110001\rangle - |10001110\rangle \\
&- (i \cdot -1 \cdot -i)|10011001\rangle + |10100101\rangle \\
&+ (i \cdot -1 \cdot -i)|10110010\rangle + |11000011\rangle \\
&+ (i \cdot i)|11010100\rangle - |11101000\rangle \\
&- (i \cdot -1 \cdot -1 \cdot i)|11111111\rangle) \\
=&\frac{1}{4}(|00000000\rangle - |00010111\rangle - |00101011\rangle \\
&+ |00111100\rangle - |01001101\rangle + |01011010\rangle \\
&+ |01100110\rangle - |01110001\rangle - |10001110\rangle \\
&+ |10011001\rangle + |10100101\rangle - |10110010\rangle \\
&+ |11000011\rangle - |11010100\rangle - |11101000\rangle \\
&+ |11111111\rangle)
\end{aligned} \tag{24}
$$

We can clearly observe that the right hand side of Equation (16) is same as the Equation (24). Thus, $|\psi_8\rangle = |\overline{000}\rangle$, implying that the final state given by Algorithm 1 or Fig. S1 is $|\overline{000}\rangle$. The algorithm can be verified for the remaining states by applying suitable $\overline{X}$ operators to state $|\overline{000}\rangle$.

IV. Higher encoding rate using 13-qubit code

New stabilizer codes can be generated by pasting together available stabilizer codes. Such a code can be created by pasting a five-qubit code [[5,1,3]] and a eight-qubit code [[8,3,3]]. The new 13-qubit code [[13,7,3]] has a rate of $7/13$. It can be represented by the following stabilizers [1]:

$$\begin{array}{c|cccccccccccc} M_1 & X & X & X & X & X & X & X & X & I & I & I & I & I \\ M_2 & Z & Z & Z & Z & Z & Z & Z & Z & I & I & I & I & I \\ M_3 & I & I & I & I & I & I & I & I & X & Z & Z & X & I \\ M_4 & I & X & I & X & Y & Z & Y & Z & I & X & Z & Z & X \\ M_5 & I & X & Z & Y & I & X & Z & Y & X & I & X & Z & Z \\ M_6 & I & Y & X & Z & X & Z & I & Y & Z & X & I & X & Z \end{array}.$$

A. Encoder design

The stabilizers for the 13-qubit code can be written in binary vector space formalism. This corresponds to Step 1 in Section I.

$$H_q = \left[\begin{array}{c|c} 1111111100000 & 0000000000000 \\ 0000000000000 & 1111111100000 \\ 0000000010010 & 0000000001100 \\ 0101101001001 & 0000111100110 \\ 0101010110100 & 0011001100011 \\ 0110100101010 & 0101010110001 \end{array}\right] \quad (25)$$

Step 2 involves converting H_q into standard form H_s as in Equation 1. First, we swap rows 2 and 3 with rows 6 and 5, respectively. Next, we swap columns 4 and 5 (equivalent to swapping 4^{th} and 5^{th} qubits), which would also require swapping columns 17 and 18. We then swap columns 5 and 9, which requires swapping columns 18 and 22 as well. Performing the above operations, we get

$$H_q = \left[\begin{array}{c|c} 1111011110000 & 0000000000000 \\ 0111000101010 & 0100110110001 \\ 0100110110100 & 0010001110011 \\ 0101001011001 & 0001011100110 \\ 0000100000010 & 0000000001100 \\ 0000000000000 & 1111011110000 \end{array}\right] \quad (26)$$

Now H_q is ready for the Gaussian elimination process. For this, we perform the following steps sequentially,

$$\begin{aligned} R_3 &\to R_3 + R_2 \\ R_4 &\to R_4 + R_2 + R_3 \\ R_1 &\to R_1 + R_2 \\ R_2 &\to R_2 + R_3 + R_5 \\ R_3 &\to R_3 + R_4 \\ R_4 &\to R_4 + R_5 \end{aligned}$$

to produce the standard parity check matrix H_s given by

$$H_s = \left[\begin{array}{c|c} 1000011011010 & 0100110110001 \\ 0100010110110 & 0010001111111 \\ 0010001110011 & 0101101010111 \\ 0001011101111 & 0011010011001 \\ 0000100000010 & 0000000001100 \\ 0000000000000 & 1111011110000 \end{array}\right]. \quad (27)$$

From the standard form H_s, the stabilizers in standard form are as follows:

$$\begin{array}{c|cccccccccccc} M_1 & X & Z & I & I & Z & Y & X & Z & Y & X & I & X & Z \\ M_2 & I & X & Z & I & I & X & Z & Y & Y & Z & Y & Y & Z \\ M_3 & I & Z & X & Z & Z & I & Y & X & Y & I & Z & Y & Y \\ M_4 & I & I & Z & Y & I & Y & X & X & Z & Y & X & X & Y \\ M_5 & I & I & I & I & X & I & I & I & I & Z & Z & X & I \\ M_6 & Z & Z & Z & Z & I & Z & Z & Z & Z & I & I & I & I \end{array}.$$

From H_s, we have

$$\begin{aligned} I_1 &= \begin{bmatrix} 10000 \\ 01000 \\ 00100 \\ 00010 \\ 00001 \end{bmatrix}, A_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, A_2 = \begin{bmatrix} 1011010 \\ 0110110 \\ 1110011 \\ 1101111 \\ 0000010 \end{bmatrix}, \\ B &= \begin{bmatrix} 01001 \\ 00100 \\ 01011 \\ 00110 \\ 00000 \end{bmatrix}, C_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0110001 \\ 1111111 \\ 1010111 \\ 0011001 \\ 0001100 \end{bmatrix}, \\ D &= [11110], I_2 = 1, E = [1110000] \end{aligned} \quad (28)$$

From the above, we obtain the logical $\overline{X}$ and $\overline{Z}$ operators.

$$\overline{X} = \left[\begin{array}{c|c} 0000011000000 & 1111000000000 \\ 0000010100000 & 0101000000000 \\ 0000010010000 & 0110000000000 \\ 0000000001000 & 0101100000000 \\ 0000000000100 & 0110100000000 \\ 0000000000010 & 0110000000000 \\ 0000000000001 & 1111000000000 \end{array}\right] \quad (29)$$

$$\overline{Z} = \left[\begin{array}{c|c} 0000000000000 & 1011001000000 \\ 0000000000000 & 0111000100000 \\ 0000000000000 & 1110000010000 \\ 0000000000000 & 1001000001000 \\ 0000000000000 & 0101000000100 \\ 0000000000000 & 1111100000010 \\ 0000000000000 & 0011000000001 \end{array}\right] \quad (30)$$

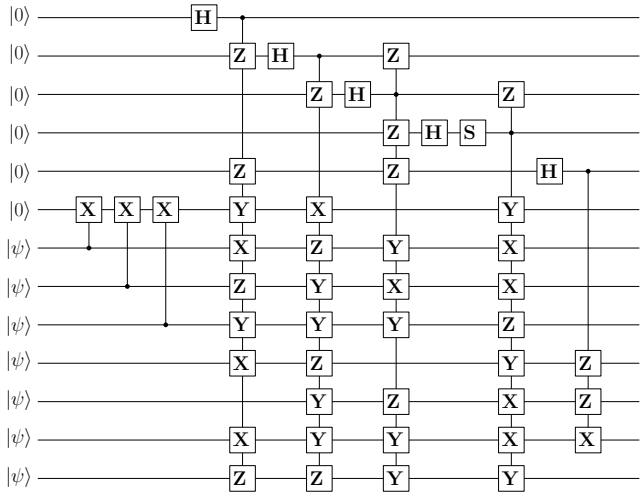

Fig. S3. Encoder for the thirteen qubit code.

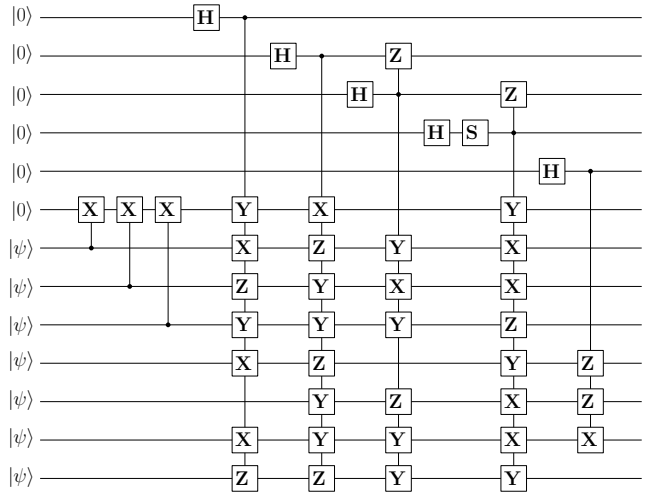

Fig. S4. Modified encoder for the thirteen qubit code after removing redundant controlled-Z gates.

$\overline{X_1}$	Z	Z	Z	Z	I	X	X	I	I	I	I	I	I
$\overline{X_2}$	I	Z	I	Z	I	X	I	X	I	I	I	I	I
$\overline{X_3}$	I	Z	Z	I	I	X	I	I	X	I	I	I	I
$\overline{X_4}$	I	Z	I	Z	Z	I	I	I	I	X	I	I	I
$\overline{X_5}$	I	Z	Z	I	Z	I	I	I	I	I	X	I	I
$\overline{X_6}$	I	Z	Z	I	I	I	I	I	I	I	I	X	I
$\overline{X_7}$	Z	Z	Z	Z	I	I	I	I	I	I	I	I	X
$\overline{Z_1}$	Z	I	Z	Z	I	I	Z	I	I	I	I	I	I
$\overline{Z_2}$	I	Z	Z	Z	I	I	I	Z	I	I	I	I	I
$\overline{Z_3}$	Z	Z	Z	I	I	I	I	I	Z	I	I	I	I
$\overline{Z_4}$	Z	I	I	Z	I	I	I	I	I	Z	I	I	I
$\overline{Z_5}$	I	Z	I	Z	I	I	I	I	I	I	Z	I	I
$\overline{Z_6}$	Z	Z	Z	Z	Z	I	I	I	I	I	I	Z	I
$\overline{Z_7}$	I	I	Z	Z	I	I	I	I	I	I	I	I	Z

Using H_s and the $\overline{X}$ operators, the various encoded states can be evaluated using the following equation

$$|c_1c_2c_3c_4c_5c_6c_7\rangle = \overline{X}_1^{c_1}\overline{X}_2^{c_2}\overline{X}_3^{c_3}\overline{X}_4^{c_4}\overline{X}_5^{c_5}\overline{X}_6^{c_6}\overline{X}_7^{c_7} \left(\sum_{M\in S} M\right)|0000000000000\rangle \quad (31)$$

$$= \overline{X}_1^{c_1}\overline{X}_2^{c_2}\overline{X}_3^{c_3}\overline{X}_4^{c_4}\overline{X}_5^{c_5}\overline{X}_6^{c_6}\overline{X}_7^{c_7} (I+M_1)(I+M_2)(I+M_3)(I+M_4) (I+M_5)(I+M_6)|00000000\rangle \quad (32)$$

Since M_6 consists of tensor products of Z Pauli operators, and since Z acts trivially on $|0\rangle$, $I+M_6$ which does not change the state $|0000000000000\rangle$. Thus, we can ignore M_6.

Following the procedure in Section I, The 7 qubits to be encoded are placed at positions $n-6$, $n-5$, $n-4$, $n-3$, $n-2$, $n-1$, and n, followed by the rest of the qubits initialized to the state $|0\rangle$. Next, the logical operators are encoded according the Algorithm 1. Thereafter, the stabilizers corresponding to the rows of standard form of the parity check matrix H_s are applied according the Algorithm 1. The encoder circuit thus designed is shown in Fig. S3.

From Fig. S3, we observe that there are 5 controlled-Z gates which act on $|0\rangle$. Since Z acts trivially on $|0\rangle$, those controlled-Z gates can be ignored. The modified encoding circuit after removing the redundant controlled-Z gates is shown in Fig. S4.

B. Syndrome measurement circuit and error corrector

The syndrome measurement circuit measures all the six stabilizers using six ancilla qubits. The syndromes are unique and can be calculated similar to the eight-qubit code. Each qubit in the 13-qubit code can be affected by three kinds of errors, namely X, Y, and Z errors. So, there are 39 different types of single qubit errors possible, each of which gives a different syndrome as shown in Table S2. The M_1-M_6 values in the table can be derived similar to the eight qubit code.

The syndrome measurement circuit is shown in Fig. S5. Six ancilla qubits are used to measure each of the six stabilizers. Measurement of the ancilla qubits produces the syndrome. Depending on the syndrome, appropriate error correction can be performed by using suitable X, Z, or Y gate on the appropriate qubit. A syndrome of '000000' implies that no error has occurred.

REFERENCES

[1] D. Gottesman, *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

[2] A. Mondal and K. K. Parhi, "Quantum circuits for stabilizer error correcting codes: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 24, no. 1, pp. 33–51, 2024.

TABLE S2
SYNDROME TABLE FOR THE 13-QUBIT CODE.

													M_1	M_1	M_1	M_1	M_1	M_1	Decimal value
X	I	I	I	I	I	I	I	I	I	I	I	I	0	0	0	0	0	1	1
Y	I	I	I	I	I	I	I	I	I	I	I	I	1	0	0	0	0	1	33
Z	I	I	I	I	I	I	I	I	I	I	I	I	1	0	0	0	0	0	32
I	X	I	I	I	I	I	I	I	I	I	I	I	1	0	1	0	0	1	41
I	Y	I	I	I	I	I	I	I	I	I	I	I	1	1	1	0	0	1	57
I	Z	I	I	I	I	I	I	I	I	I	I	I	0	1	0	0	0	0	16
I	I	X	I	I	I	I	I	I	I	I	I	I	0	1	0	1	0	1	21
I	I	Y	I	I	I	I	I	I	I	I	I	I	0	1	1	1	0	1	29
I	I	Z	I	I	I	I	I	I	I	I	I	I	0	0	1	0	0	0	8
I	I	I	X	I	I	I	I	I	I	I	I	I	0	0	1	1	0	1	13
I	I	I	Y	I	I	I	I	I	I	I	I	I	0	0	1	0	0	1	9
I	I	I	Z	I	I	I	I	I	I	I	I	I	0	0	0	1	0	0	4
I	I	I	I	X	I	I	I	I	I	I	I	I	1	0	1	0	0	0	40
I	I	I	I	Y	I	I	I	I	I	I	I	I	1	0	1	0	1	0	42
I	I	I	I	Z	I	I	I	I	I	I	I	I	0	0	0	0	1	0	2
I	I	I	I	I	X	I	I	I	I	I	I	I	1	0	0	1	0	1	37
I	I	I	I	I	Y	I	I	I	I	I	I	I	0	1	0	0	0	1	17
I	I	I	I	I	Z	I	I	I	I	I	I	I	1	1	0	1	0	0	52
I	I	I	I	I	I	X	I	I	I	I	I	I	0	1	1	0	0	1	25
I	I	I	I	I	I	Y	I	I	I	I	I	I	1	1	0	1	0	1	53
I	I	I	I	I	I	Z	I	I	I	I	I	I	1	0	1	1	0	0	44
I	I	I	I	I	I	I	X	I	I	I	I	I	1	1	0	0	0	1	49
I	I	I	I	I	I	I	Y	I	I	I	I	I	1	0	1	1	0	1	45
I	I	I	I	I	I	I	Z	I	I	I	I	I	0	1	1	1	0	0	28
I	I	I	I	I	I	I	I	X	I	I	I	I	1	1	1	1	0	1	61
I	I	I	I	I	I	I	I	Y	I	I	I	I	0	0	0	1	0	1	5
I	I	I	I	I	I	I	I	Z	I	I	I	I	1	1	1	0	0	0	56
I	I	I	I	I	I	I	I	I	X	I	I	I	0	1	0	1	1	0	22
I	I	I	I	I	I	I	I	I	Y	I	I	I	1	1	0	0	1	0	50
I	I	I	I	I	I	I	I	I	Z	I	I	I	1	0	0	1	0	0	36
I	I	I	I	I	I	I	I	I	I	X	I	I	0	1	1	0	1	0	26
I	I	I	I	I	I	I	I	I	I	Y	I	I	0	0	1	1	1	0	14
I	I	I	I	I	I	I	I	I	I	Z	I	I	0	1	0	1	0	0	20
I	I	I	I	I	I	I	I	I	I	I	X	I	0	1	1	0	0	0	24
I	I	I	I	I	I	I	I	I	I	I	Y	I	1	0	0	1	1	0	38
I	I	I	I	I	I	I	I	I	I	I	Z	I	1	1	1	1	1	0	62
I	I	I	I	I	I	I	I	I	I	I	I	X	1	1	1	1	0	0	60
I	I	I	I	I	I	I	I	I	I	I	I	Y	1	1	0	0	0	0	48
I	I	I	I	I	I	I	I	I	I	I	I	Z	0	0	1	1	0	0	12
I	I	I	I	I	I	I	I	I	I	I	I	I	0	0	0	0	0	0	0

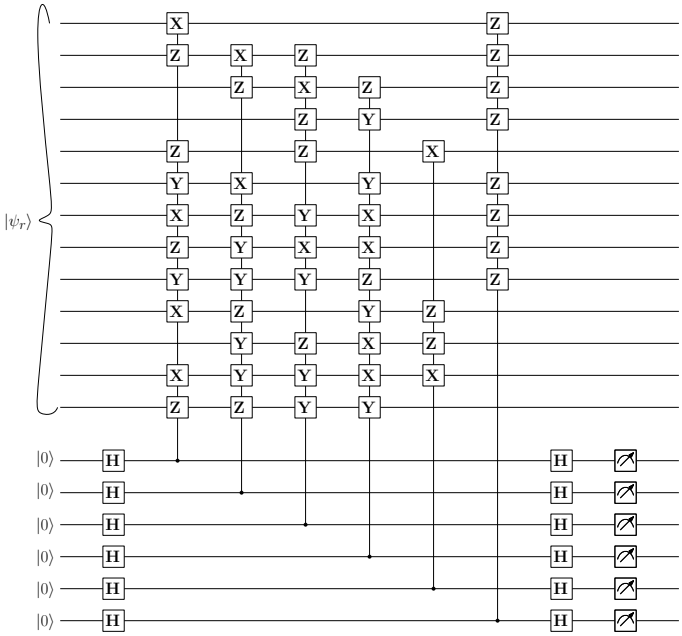

Fig. S5. Syndrome measurement circuit for a thirteen qubit code.