

Quantum Circuits for Stabilizer Error Correcting Codes: A Tutorial

Arijit Mondal and Keshab K. Parhi, *Fellow, IEEE*

Email: {monda109, parhi}@umn.edu

Department of Electrical and Computer Engineering, University of Minnesota

Abstract—Quantum computers have the potential to provide exponential speedups over their classical counterparts. Quantum principles are being applied to fields such as communications, information processing, and artificial intelligence to achieve quantum advantage. However, quantum bits are extremely noisy and prone to decoherence. Thus, keeping the qubits error free is extremely important toward reliable quantum computing. Quantum error correcting codes have been studied for several decades and methods have been proposed to import classical error correcting codes to the quantum domain. Along with the exploration into novel and more efficient quantum error correction codes, it is also essential to design circuits for practical realization of these codes. This paper serves as a tutorial on designing and simulating quantum encoder and decoder circuits for stabilizer codes. We first describe Shor’s 9-qubit code which was the first quantum error correcting code. We discuss the stabilizer formalism along with the design of encoding and decoding circuits for stabilizer codes such as the five-qubit code and Steane code. We also design nearest neighbor compliant circuits for the above codes. The circuits were simulated and verified using IBM Qiskit.

Index Terms—Quantum ECCs, Quantum computation, Steane code, CSS code, Stabilizer codes, Quantum encoders and decoders, Syndrome measurement, Nearest neighbor compliant circuits.

I. INTRODUCTION

Quantum computing is a rapidly-evolving technology which exploits the fundamentals of quantum mechanics toward solving tasks which are too complex for current classical computers. Quantum computers have the potential to achieve exponential speedups over their classical counterparts [1], [2]. In 1981, Feynman suggested that a quantum computer would have the power to simulate systems which are not feasible for classical computers [3], [4]. In 1994, Shor proposed a quantum algorithm to find the prime factors of an integer in *polynomial* time [5]. Grover proposed an algorithm which was able to search a particular element in an unsorted database with a high probability, with significantly higher efficiency than any known classical algorithm in 1996 [6]. Subsequently, several quantum algorithms aimed at achieving better efficiencies than their classical counterparts were proposed. However, practical realization of these algorithms requires quantum computers, which are slowly evolving. IBM demonstrated a 433 qubit quantum computer in 2022 [7], and recently announced a 1121 qubit computer. The path towards a powerful quantum computer which can perform Shor’s factorization or Grover’s search algorithm may not be a distant reality. However, a fundamental issue needs to be addressed. As we pack more

number of qubits into quantum processors, we need to have a reliable method of processing to mitigate noise and quantum decoherence.

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 which necessitates building fault tolerant systems which can overcome noise and decoherence. Thus, quantum error correcting codes (ECCs) become a necessity for quantum computing systems. There were various challenges in the way of designing a quantum ECC framework. It is well known that measurement destroys superpositions in any quantum system. Additionally, since the quantum errors are continuous in nature, the design of an ECC for quantum systems was difficult. To make things more complicated, the no-go theorems in the quantum realm make it challenging to design an ECC system analogous to classical domain [8], [9], [10], [11], [12]. 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 [13]. In 1996, Gottesman proposed a stabilizer framework which was widely used for the construction of quantum ECCs from classical ECCs [14], [15]. Calderbank-Shor-Steane (CSS) codes were proposed independently by Calderbank-Shor [16] and Steane [17]. These codes were used to derive quantum codes from binary classical linear codes which satisfy a dual-containing criterion. The necessary and sufficient conditions for a quantum ECC to be able to recover from a set of errors were given in [18]. Topological quantum codes like toric code were constructed for applications on quantum circuits arranged in a torus [19]. Subsequently, surface codes were introduced using stabilizer formalism in [20].

Pre-shared entangled qubits were proposed toward constructing stabilizer codes over non-Abelian groups in [21]. This is done by extending the non-Abelian group into an Abelian group by using extended operators which commute with each other. 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 [22]. The advantage of the stabilizer framework lies in its ability to construct quantum ECCs from any classical binary ECC. The optimal number of pre-

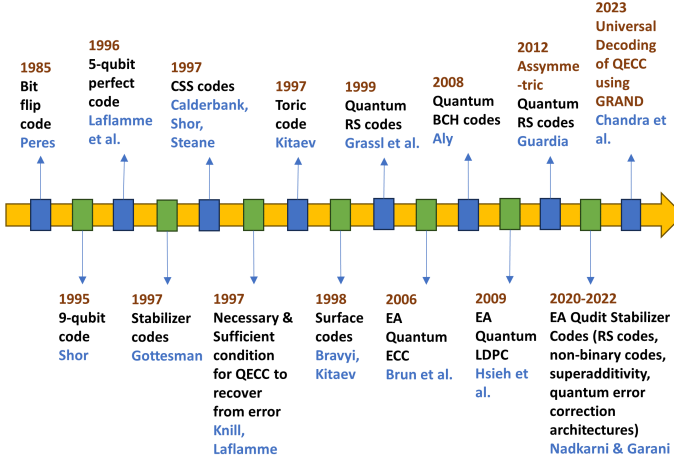


Fig. 1. Chronology of some of the primary advances in quantum ECC.

shared entangled qubits required for an EA stabilizer code was expressed analytically, along with an encoding procedure in [23]. Quantum analog of classical low-density parity-check (LDPC) codes were constructed using quasi-cyclic binary LDPC codes in [24]. Algebraic codes like Reed Solomon (RS) codes were also explored in the quantum domain by the authors in [25], [26], and [27] using self-orthogonal classical RS codes. Purely quantum polar codes based on recursive channel combining and splitting construction were studied in [28]. EA stabilizer codes were extended to qudit systems for different aspects of quantum ECCs in [29], [30], [31], [32]. Recently, a universal decoding scheme was conceived for stabilizer codes by adapting 'guessing random additive noise decoding' (GRAND) philosophy from classical domain codes [33]. However, it becomes necessary to design actual encoder and decoder circuits for these quantum ECCs, so that reliable quantum computing systems can be built. The CSS framework is particularly interesting due to its simplicity. It provides a method for importing any classical ECC into the quantum domain, as long as the dual-containing criterion is satisfied. A chronological list of some of the primary advances in quantum ECCs is shown in Fig. 1.

Along with the exploration of novel and efficient quantum ECCs, it is also important to design encoding and decoding circuits for quantum ECCs toward practical implementation on quantum computers. Quantum bits, unlike classical bits, are excessively prone to noise and decoherence. Also, there are other challenges related to the no-go theorems which make the design of quantum ECC circuits challenging. Systematic design of circuits for stabilizer codes was explored in [15]. There are challenges related to the reliability of these circuits. Multi-qubit gates are a source of noise since they involve interaction between qubits. Optimization of the circuits in terms of number of gates enhances reliability by reducing the number of 2-qubit gates [34]. In [35], a procedure to optimize encoding circuits for stabilizer codes was proposed. Also, nearest neighbor compliant (NNC) circuits are desirable to limit interactions to adjacent qubits, thus enhancing circuit reliability [36]. In this paper, we will focus on the NNC encoder and decoder circuits for stabilizer codes.

The contributions of this paper are as follows. This paper serves as a tutorial for the design of encoder and decoder circuits for stabilizer codes. First, we discuss the encoding and decoding circuits for Shor's 9-qubit code, followed by its treatment using stabilizer formalism. Second, we revisit the systematic method for construction of encoder and decoder circuits for stabilizer codes. We identify and analyze the key concepts for the construction of an encoder for stabilizer codes, demonstrated in [15] through a five-qubit code [37], [38]. The concepts are then used to formulate an algorithm for the construction of an encoder circuit for a general stabilizer code. For the decoder design, we use a syndrome measurement circuit, and depending on the measured syndromes, we may apply the appropriate error correction using suitable Pauli gates. Third, we present encoder and decoder circuits for two stabilizer codes: the five-qubit code and the Steane code. Fourth, we provide NNC circuits for the encoding and syndrome measurement for the above codes [36]. We also present simulation results using IBM Qiskit [39] and verify the circuits.

The rest of the paper is organized as follows. Section II presents a brief description of quantum gates and quantum circuits. Section III reviews Shor's 9-qubit code [13], stabilizer formalism, and CSS codes. In Section IV, we discuss a systematic method of construction of encoder circuits for general stabilizer codes. Using the above knowledge, we present design of encoding and syndrome measurement circuits for the five-qubit code and Steane code in Section V. In Section VI, we provide NNC circuits for the above codes. We discuss the results and comparisons in Section VII, followed by conclusions in Section VIII.

II. PAULI MATRICES AND QUANTUM GATES

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 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 $\mathbb{C}^{2 \times 2}$. We have a Pauli group which represents the unitary matrices given by

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

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

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 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 controlled- X (CNOT), controlled- Z (CZ), controlled- Y (CY) gates, and the controlled-controlled-NOT (CCNOT), also known as the Toffoli gate. These gates act on 2-qubit or 3-qubit states and are given by the following unitary transformations:

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

$$CY = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix} \quad (4)$$

$$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Symbolic representations of various 1-qubit, 2-qubit, and 3-qubit gates are shown in Fig. 2.

III. DEVELOPMENT OF THE FIRST QUANTUM ECC AND STABILIZER FORMALISM

Shor's 9-qubit code was the first ever quantum ECC capable of correcting a single qubit error [13]. Gottesman proposed a general methodology to construct quantum ECCs [15]. This method is known as the stabilizer construction and the codes thus generated are known as stabilizer codes. Calderbank-Shor [16] and Steane [17] proposed a method to derive quantum codes from binary classical linear codes which satisfy a dual-containing criterion. We will discuss the above in detail in this section.

A. Shor's 9-qubit Quantum ECC

Shor's 9-qubit code consists of a combination of 3-qubit bit flip and 3-qubit phase flip codes. First, we will provide a brief description of the working of these 3-qubit codes. From classical ECCs, we know about repetition codes. For a rate 1/3 repetition code, 0 is transmitted as 000 and 1 is transmitted as 111. The redundancies ensure that if a single error has

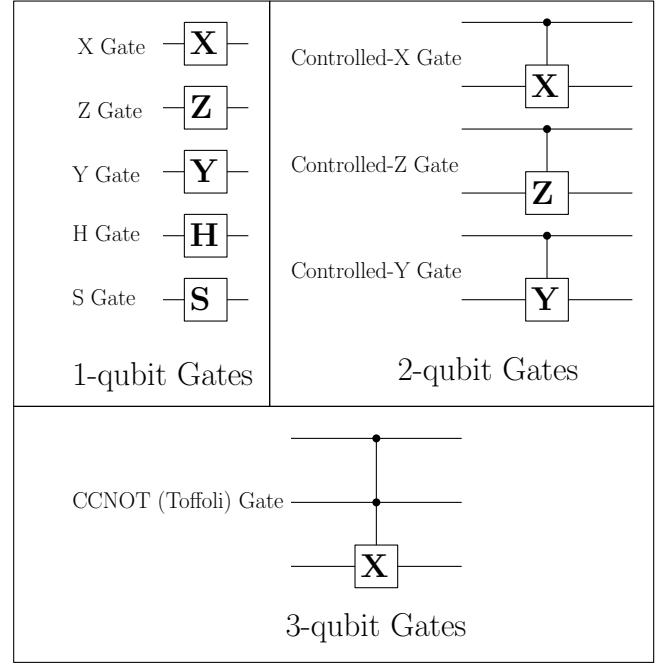


Fig. 2. Symbolic representations of various 1-qubit and 2-qubit gates.

occurred, a majority detector can detect and correct the error. Analogous to repetition code, we have a 3-qubit bit flip code and a 3-qubit phase flip code. However, due to the no-cloning theorem in quantum domain, we cannot create copies of a certain qubit state. Next, we will describe how this limitation is overcome toward the design of the 3-qubit bit flip and phase flip codes.

1) *3-qubit bit flip code*: We can design a 3-qubit quantum code [40] capable of correcting a single bit flip error as shown in Fig. 3. Two ancilla qubits are initialized to $|0\rangle$ analogous to redundant bits in a 3-bit repetition code. A single qubit is thus encoded into a 3-qubit state. The basis states $|0\rangle$ and $|1\rangle$ are encoded using encoding as shown below:

$$|0\rangle \xrightarrow{CNOT(3,2)CNOT(3,1)} |000\rangle, \quad (6)$$

$$|1\rangle \xrightarrow{CNOT(3,2)CNOT(3,1)} |111\rangle \quad (7)$$

For an arbitrary normalized state $|\phi\rangle = a|0\rangle + b|1\rangle$, where $a, b \in \mathbb{C}$, the encoding operation results in the state $|\psi\rangle$ given by

$$|\psi\rangle = a|000\rangle + b|111\rangle \quad (8)$$

The notation $CNOT(x, y)$ implies a $CNOT$ gate acting on qubits indexed x and y , with x as control and y as target qubit. The qubits are numbered from top to bottom. It should be noted that in $CNOT(3,2)CNOT(3,1)$, the rightmost operation is performed first and the leftmost operation is performed last.

The syndrome computation circuit is shown in Fig. 4. Two ancilla qubits initialized to $|0\rangle$ are used to compute the syndrome. We perform the operation $CNOT(1,4)CNOT(2,4)CNOT(1,5)CNOT(3,5)$ on the state $\mathcal{U}|\psi\rangle|00\rangle$ to obtain $\mathcal{U}|\psi\rangle|s\rangle$ as shown in Fig. 4. The two qubit syndrome state is given by $|s\rangle$.

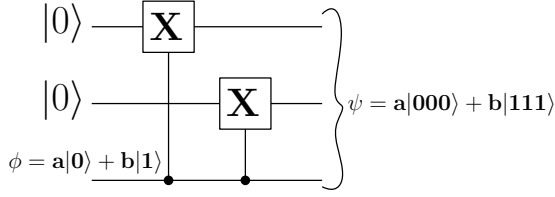


Fig. 3. 3-qubit bit flip encoder.

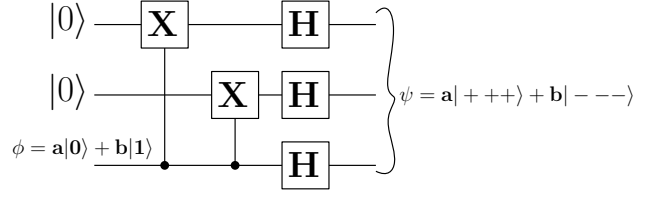


Fig. 5. 3-qubit phase flip encoder.

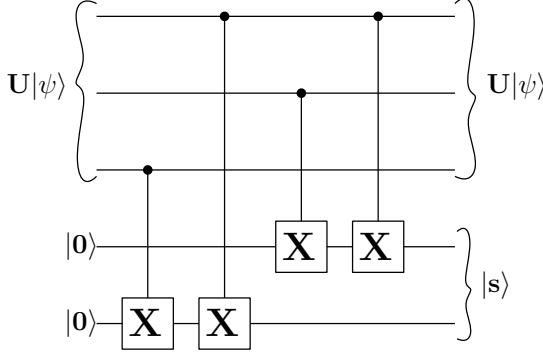


Fig. 4. 3-qubit bit flip syndrome computation.

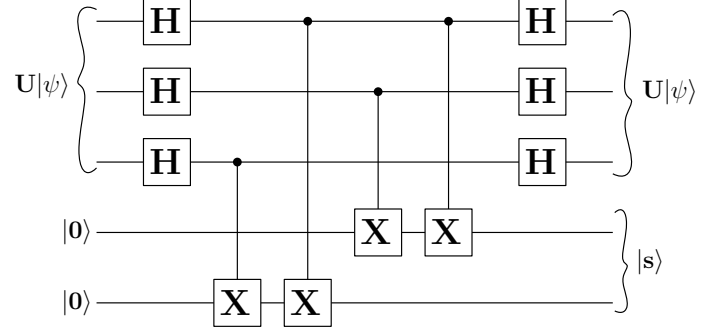


Fig. 6. 3-qubit phase flip syndrome computation.

Let's take an example to demonstrate the syndrome detection. Let the error be $\mathcal{U} = I_2 \otimes I_2 \otimes X$, leading to the erroneous state $\mathcal{U}|\psi\rangle = a|001\rangle + b|110\rangle$. Performing the operation $\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5)$ on $\mathcal{U}|\psi\rangle|00\rangle$, we have [41],

$$\begin{aligned}
 & (\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5)) \\
 & \mathcal{U}|\psi\rangle|00\rangle \\
 &= (\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5)) \\
 & (a|001\rangle + b|110\rangle)|00\rangle \\
 &= (\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5)) \\
 & (a|00100\rangle + b|11000\rangle) \\
 &= a|00101\rangle + b|11001\rangle \\
 &= \mathcal{U}|\psi\rangle|01\rangle \\
 &= \mathcal{U}|\psi\rangle|s\rangle
 \end{aligned}$$

Thus, the syndrome is $|s\rangle = |01\rangle$. The syndromes $|11\rangle$, $|10\rangle$, and $|01\rangle$ correspond to errors in the first, second and third qubits respectively. Here, since the syndrome is $|01\rangle$, the third qubit is in error.

2) *3-qubit phase flip code*: A 3-qubit phase flip code encodes a single qubit into a 3-qubit state as shown in Fig. 5. Basis states $|0\rangle$ and $|1\rangle$ are encoded as shown below:

$$|0\rangle \xrightarrow{H^{\otimes 3}\text{CNOT}(3,2)\text{CNOT}(3,1)} |++\rangle, \quad (9)$$

$$|1\rangle \xrightarrow{H^{\otimes 3}\text{CNOT}(3,2)\text{CNOT}(3,1)} |--\rangle \quad (10)$$

where $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$. Any arbitrary normalized state $|\phi\rangle = a|0\rangle + b|1\rangle$ gets encoded to state $|\psi\rangle$ using the above encoding operation as

$$|\psi\rangle = a|++\rangle + b|--\rangle \quad (11)$$

The unitary operator $H^{\otimes 3}\text{CNOT}(3,2)\text{CNOT}(3,1)$ is applied to the message state $|\phi\rangle$ along with two ancilla bits initially in the state $|0\rangle$ to perform the encoding.

The syndrome detection circuit is shown in Fig. 6. The syndrome is computed by performing the operation $(H^{\otimes 3} \otimes I_2^{\otimes 2})(\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5))(H^{\otimes 3} \otimes I_2^{\otimes 2})$ on the state $\mathcal{U}|\psi\rangle|00\rangle$ to obtain $\mathcal{U}|\psi\rangle|s\rangle$, where $|s\rangle$ is the two qubit syndrome state.

We now consider an example to demonstrate the syndrome detection. Let the error be $Z \otimes I_2 \otimes I_2$. Thus, the erroneous state is $\mathcal{U}|\psi\rangle = a|---\rangle + b|++\rangle$. The first step $(H^{\otimes 3} \otimes I_2^{\otimes 2})$ converts $\mathcal{U}|\psi\rangle|00\rangle$ to $U_1|\psi\rangle|00\rangle = a|10000\rangle + b|01100\rangle$. Next, the operation $(\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5))$ converts $U_1|\psi\rangle|00\rangle$ to $U_2|\psi\rangle|s\rangle$ as follows:

$$\begin{aligned}
 & (\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5)) \\
 & U_1|\psi\rangle|00\rangle \\
 &= (\text{CNOT}(1,4)\text{CNOT}(2,4)\text{CNOT}(1,5)\text{CNOT}(3,5)) \\
 & (a|10000\rangle + b|01100\rangle) \\
 &= a|10011\rangle + b|01111\rangle \\
 &= a|100\rangle|11\rangle + b|011\rangle|11\rangle \\
 &= (a|100\rangle + b|011\rangle)|11\rangle
 \end{aligned}$$

Next, the operation $(H^{\otimes 3} \otimes I_2^{\otimes 2})$ converts $(a|100\rangle + b|011\rangle)|11\rangle$ to $(a|++\rangle + b|--\rangle)|11\rangle = \mathcal{U}|\psi\rangle|11\rangle = \mathcal{U}|\psi\rangle|s\rangle$. Thus, the syndrome is $|s\rangle = |11\rangle$. The syndromes $|11\rangle$, $|10\rangle$, and $|01\rangle$ correspond to phase errors in the first, second and third qubits, respectively. Here, since the syndrome is $|11\rangle$, the first qubit has a phase error.

The 3-qubit bit flip code is good at correcting a single bit flip. However, it cannot correct phase errors. It is in fact more prone to phase flip errors since phase flips in any of the qubits are indistinguishable from each other. Similarly, the 3-qubit phase flip code cannot correct bit flip errors. Hence, it was believed for a long time that a general quantum ECC capable of correcting both type of errors was not feasible, until Shor [13] proposed a 9 qubit code capable of correcting a bit flip and a phase flip simultaneously. The encoding and decoding circuits for Shor's 9-qubit code are shown in Fig. 7 and Fig. 8, respectively.

The encoding process consists of the following steps:

Step 1: Phase flip coding: After applying the CNOT gates we have the following state

$$|\psi_1\rangle = a|000\rangle + b|111\rangle \quad (12)$$

Next, we have three Hadamard gates resulting in the state

$$|\psi_2\rangle = a \left[\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \right] + b \left[\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \right]$$

Step 2: Bit flip coding: After adding the ancillas, we have the state

$$|\psi_3\rangle = \frac{a}{2\sqrt{2}} [((|0\rangle + |1\rangle)|00\rangle)((|0\rangle + |1\rangle)|00\rangle) \\ ((|0\rangle + |1\rangle)|00\rangle)] + \frac{b}{2\sqrt{2}} [((|0\rangle - |1\rangle)|00\rangle) \\ ((|0\rangle - |1\rangle)|00\rangle)((|0\rangle - |1\rangle)|00\rangle)]$$

Next the CNOT gates are applied to achieve the encoded state

$$|\psi_t\rangle = \frac{a}{2\sqrt{2}} [(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)] \\ + \frac{b}{2\sqrt{2}} [(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)]$$

Decoding process: Let us assume that there is a bit and a phase flip on the 4th qubit. Thus the combined state of the received qubits can be represented as:

$$|\psi_r\rangle = a \left[\left(\frac{|000\rangle + |111\rangle}{\sqrt{2}} \right) \left(\frac{|100\rangle - |011\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}} \right) \right] + b \left[\left(\frac{|000\rangle - |111\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{|100\rangle + |011\rangle}{\sqrt{2}} \right) \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}} \right) \right]$$

The evolution of states for the decoding can be described in the following steps:

Step 1: After the application of the first two CNOT gates, we have

$$|\psi_{s1}\rangle = a \left[\left(\frac{|000\rangle + |100\rangle}{\sqrt{2}} \right) \left(\frac{|111\rangle - |011\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{|000\rangle + |100\rangle}{\sqrt{2}} \right) \right] + b \left[\left(\frac{|000\rangle - |100\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{|111\rangle + |011\rangle}{\sqrt{2}} \right) \left(\frac{|000\rangle - |100\rangle}{\sqrt{2}} \right) \right]$$

Step 2: Next, the CCNOT (Toffoli) gates are applied resulting in the state

$$|\psi_{s2}\rangle = a \left[\left(\frac{|000\rangle + |100\rangle}{\sqrt{2}} \right) \left(\frac{|011\rangle - |111\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{|000\rangle + |100\rangle}{\sqrt{2}} \right) \right] + b \left[\left(\frac{|000\rangle - |100\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{|011\rangle + |111\rangle}{\sqrt{2}} \right) \left(\frac{|000\rangle - |100\rangle}{\sqrt{2}} \right) \right] \\ = a \left[\left(\frac{(|0\rangle + |1\rangle)|00\rangle}{\sqrt{2}} \right) \left(\frac{(|0\rangle - |1\rangle)|11\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{(|0\rangle + |1\rangle)|00\rangle}{\sqrt{2}} \right) \right] + b \left[\left(\frac{(|0\rangle - |1\rangle)|00\rangle}{\sqrt{2}} \right) \right. \\ \left. \left(\frac{(|0\rangle + |1\rangle)|11\rangle}{\sqrt{2}} \right) \left(\frac{(|0\rangle - |1\rangle)|00\rangle}{\sqrt{2}} \right) \right]$$

Step 3: Applying Hadamard gate on 1st, 4th, and 7th qubits, we have

$$|\psi_{s3}\rangle = a|0\rangle_1|1\rangle_4|0\rangle_7 + b|1\rangle_1|0\rangle_4|1\rangle_7 \quad (13)$$

Step 4: Next, applying CNOT gates, we have

$$|\psi_{s4}\rangle = a|0\rangle_1|1\rangle_4|0\rangle_7 + b|1\rangle_1|1\rangle_4|0\rangle_7 \quad (14)$$

Step 5: Finally, applying the CCNOT (Toffoli) gates, the state is

$$|\psi_{s5}\rangle = |0\rangle_1|1\rangle_4|0\rangle_7 + b|1\rangle_1|1\rangle_4|0\rangle_7 \\ = (a|0\rangle + b|1\rangle)|1\rangle|0\rangle$$

As we can see, the first qubit is restored to the $a|0\rangle + b|1\rangle$ state. This is true independent of the index of the qubit on which the error has occurred.

B. Shor's 9-qubit code in stabilizer framework

Now, we analyze the 9-qubit code and try to reason why it works, and then we generalize it toward a systematic method of error correction using the idea used in the 9-qubit code. From Fig. 8, we observe that for detecting bit flips in each group of three, we compare the first and third qubit, followed by the first two qubits. A correctly encoded state has the property that the first two qubits have even parity. Equivalently, a codeword is a +1 eigen vector of ZZI , and a state with an

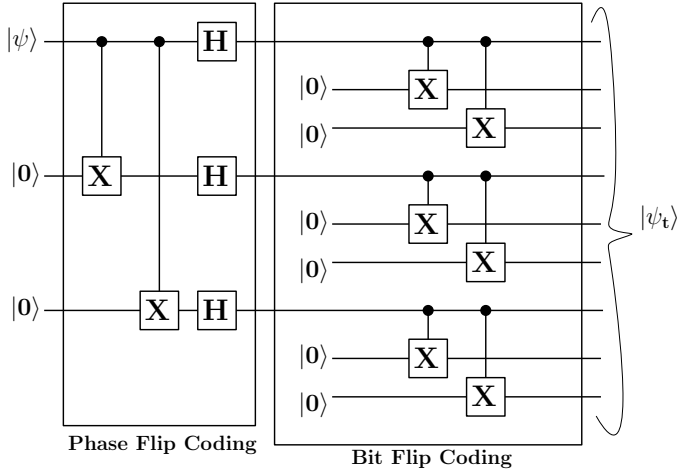


Fig. 7. Encoder for Shor's 9-qubit code.

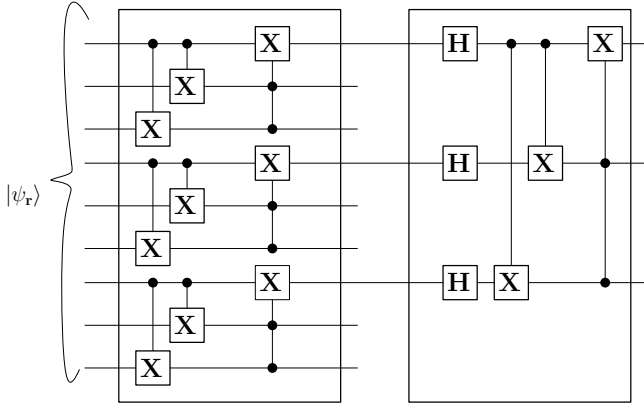


Fig. 8. Decoder for Shor's 9-qubit code.

error on first or second qubit is a -1 eigenvector of ZZI . Similarly, first and third qubit should have even parity. Thus a codeword is also $+1$ eigenvector of ZIZ .

For detecting phase errors, we compare the signs of first and second blocks of three, and the signs of first and third blocks of three. Thus, a correctly encoded codeword is a $+1$ eigenvector of $XXXXXXIII$ and $XXXIIIXXX$. Thus, to correct the code, we need to measure the eigenvalues of the eight operators as shown in the following table.

M_1	Z	Z	I	I	I	I	I	I
M_2	Z	I	Z	I	I	I	I	I
M_3	I	I	I	Z	Z	I	I	I
M_4	I	I	I	Z	I	Z	I	I
M_5	I	I	I	I	I	I	Z	Z
M_6	I	I	I	I	I	I	Z	I
M_7	X	X	X	X	X	X	I	I
M_8	X	X	X	I	I	I	X	X

The two valid codewords in Shor's code are eigenvectors of all these operators M_1 through M_8 with eigenvalues $+1$. These generate a group, the stabilizer of the code, which consists of all Pauli operators M with the property that $M|\psi\rangle = |\psi\rangle$ for all encoded states $|\psi\rangle$.

C. 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 [15]. 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. The second matrix has a 1 wherever there is a Z or Y in the corresponding stabilizer and 0 everywhere else. 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.

D. Stabilizer formalism

An $[[n, k]]$ 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. 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 S is an Abelian subgroup of G , such that the code space T is the space of vectors fixed by S [14], [15]. Stabilizer generators are a set of independent $n - k$ 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-bit vectors. The operators in Π have eigen values ± 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,

$$\begin{aligned} \Pi^n = \{ & e^{i\phi} A_1 \otimes A_2 \otimes \dots \otimes A_n \\ & : \forall j \in \{1, 2, \dots, n\} A_j \in \Pi, \phi \in \{0, \pi/2, \pi, 3\pi/2\} \} \end{aligned} \quad (15)$$

The stabilizers form 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 S of Pauli operators, the code space is defined as

$$T(S) = \{ |\psi\rangle, s.t. M|\psi\rangle = |\psi\rangle, \forall M \in S \} \quad (16)$$

Suppose $M \in S$ 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 . Thus, eigenvalue of an operator M from a stabilizer group detects errors which anti-commute with M .

Single qubit operators X , Y , and Z commute with themselves while they anti-commute with each other. For two multiple qubit operators, we need to evaluate how many anti-commutations happen. If the number is odd, the operators anti-commute; else, they commute.

Examples:

- X commutes with X , and anti-commutes with Y and Z .
- $X \otimes X \otimes Z$ commutes with $X \otimes Y \otimes X$ since there are two anti-commuting qubit positions, 2 and 3.
- $Y \otimes Z \otimes X$ anti-commutes with $Y \otimes X \otimes X$, since there is a single anti-commutation at position 2.

E. CSS framework

The CSS framework [16], [17] is a method to construct quantum ECCs from their classical counterparts. 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:

$$H_{C_1 C_2} = \left[\begin{array}{c|c} H_1 & 0 \\ \hline 0 & H_2 \end{array} \right] \quad (17)$$

IV. SYSTEMATIC PROCEDURE FOR ENCODER DESIGN FOR A STABILIZER CODE

A systematic method for the design of an encoder for a stabilizer code was presented in [15]. Taking the key concepts from the above method, the complete procedure for the design of an encoder circuit for a stabilizer code can be summarized as follows:

Step 1: The stabilizers are written in a matrix form using binary vector space formalism as mentioned in Section III-C. Let the parity-check matrix thus obtained be H_q .

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

$$H_s = \left[\begin{array}{ccc|ccc} I_1 & A_1 & A_2 & B & C_1 & C_2 \\ 0 & 0 & 0 & D & I_2 & E \end{array} \right] \quad (18)$$

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 [15]. The logical operators $\bar{X}$ and $\bar{Z}$ can be written as

$$\bar{X} = [\quad 0 \quad U_2 \quad U_3 \quad | \quad V_1 \quad 0 \quad 0 \quad] \quad (19)$$

$$\bar{Z} = [\quad 0 \quad 0 \quad 0 \quad | \quad V'_1 \quad 0 \quad V'_3 \quad] \quad (20)$$

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 $\bar{X}$, the encoding operation for a stabilizer code can be written as,

$$|c_1 c_2 \cdots c_k\rangle = \bar{X}_1^{c_1} \bar{X}_2^{c_2} \cdots \bar{X}_k^{c_k} \left(\sum_{M \in S} M \right) |00 \cdots 0\rangle \quad (21)$$

$$= \bar{X}_1^{c_1} \bar{X}_2^{c_2} \cdots \bar{X}_k^{c_k} (I + M_1)(I + M_2) \cdots (I + M_{n-k}) |00 \cdots 0\rangle. \quad (22)$$

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 $\bar{X}$:

- We know that a particular logical operator $\bar{X}_i$ is applied only if the qubit at i^{th} position is $|1\rangle$. Thus, applying $\bar{X}_i$ controlled at i^{th} qubit encodes $\bar{X}_i$.
- The $\bar{X}$ operators consist of products of only Z s for the first r qubits. For the rest of the qubits, $\bar{X}$ consists of products of X 's 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 Z s in $\bar{X}$.
- 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 from 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 can be used to devise an algorithm as shown in Algorithm 1 to design the encoding circuit.

V. ENCODER AND SYNDROME MEASUREMENT CIRCUIT DESIGN FOR STABILIZER CODES

A. Error model

The quantum error correction system consists of an encoder, the channel, and a decoder as shown in Fig 9. The encoder transforms k logical qubits to n physical qubits. Subsequently, the channel introduces errors. The errors can be of three different types, namely a bit flip (X), a phase flip (Z), and a Y error. We will assume that errors act independently on individual qubits. Additionally, it is assumed that the error acting on a qubit is equally likely to be a X , Y or Z error. If the number of errors introduced is within the error correction limit, the syndrome measurement circuit generates unique syndromes for each of the possible error types. Depending on the measured syndromes, the error corrector uses the appropriate Pauli (X , Z , or Y) gate to correct the error.

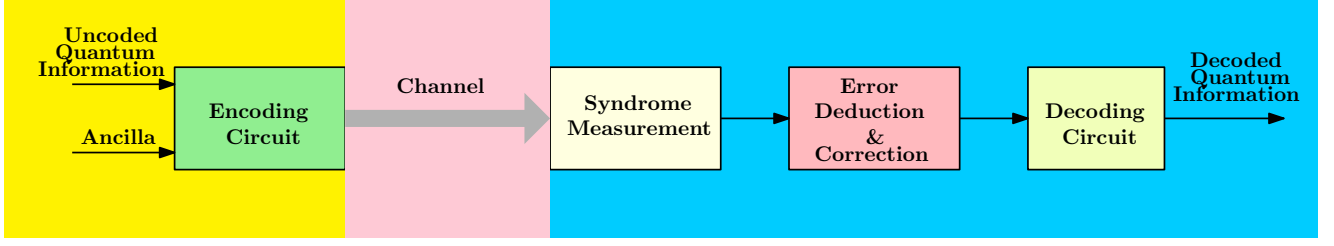


Fig. 9. A quantum error correction system consisting of an encoder, channel, and a decoder.

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

Data: $H_s, \bar{X}$

Result: Encoding circuit

for $i = 1$ **to** k **do**

if $\bar{X}_{i,i+n-k} == 1$ **then**

 Place controlled dot at qubit $i + n - k$

end

for $j = 1$ **to** n **do**

if $i + n - k \neq j$ **then**

if $\bar{X}_{i,j} == 1$ **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 $H_{s(i,j)} == 1 \ \&\& \ H_{i,j+n} == 0$ **then**

 Place X gate on qubit j with control at qubit i

end

if $H_{s(i,j)} == 0 \ \&\& \ H_{i,j+n} == 1$ **then**

 Place Z gate on qubit j with control at qubit i

end

if $H_{s(i,j)} == 1 \ \&\& \ H_{i,j+n} == 1$ **then**

 Place Y gate on qubit j with control at qubit i

end

end

end

end

B. Encoder and syndrome measurement circuit design for 5-qubit code

The five-qubit [37], [38] ECC is the smallest quantum ECC with the ability to correct a single qubit error. It is a cyclic code with a distance of 3. The treatment of the 5-qubit code in the stabilizer formalism was provided in [15]. We will revisit the concept in brief. The stabilizers $M_1 - M_4$ along with the logical $\bar{X}$ and $\bar{Z}$ operators for a 5-qubit ECC are given as follows:

M_1	X	Z	Z	X	I
M_2	I	X	Z	Z	X
M_3	X	I	X	Z	Z
M_4	Z	X	I	X	Z
$\bar{X}$	X	X	X	X	X
$\bar{Z}$	Z	Z	Z	Z	Z

1) *Extended parity-check matrix and encoder design:* In [15], the parity check matrix of the five-qubit code using binary formalism was given. Using the binary formalism as described in Section III-C, we can write the extended parity-check matrix as follows:

$$H_q = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right] \quad (23)$$

For the encoder design, H_q is converted to standard form using Gaussian elimination. The standard form was given in [15] directly; however, we describe the steps in detail. Our aim is to bring the above parity check matrix in the standard form through Gaussian elimination as described in Section IV. Applying $R_3 \rightarrow R_3 + R_1$ and $R_4 \rightarrow R_4 + R_2$

$$H_q = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array} \right] \quad (24)$$

Applying $R_1 \rightarrow R_1 + R_4$ and $R_3 \rightarrow R_3 + R_4$, we get the standard form of the parity check matrix as

$$H_s = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array} \right] \quad (25)$$

We observe that $A_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$, and

$$C_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The logical operators can be evaluated using the steps mentioned in Section IV. We get,

$$\bar{X} = [00001 \mid 10010] \quad (26)$$

$$\bar{Z} = [00000 \mid 11111] \quad (27)$$

From the standard parity-check matrix H_s and the logical operators $\bar{X}$ and $\bar{Z}$, we have,

M_1	Y	Z	I	Z	Y
M_2	I	X	Z	Z	X
M_3	Z	Z	X	I	X
M_4	Z	I	Z	Y	Y
$\bar{X}$	Z	I	I	Z	X
$\bar{Z}$	Z	Z	Z	Z	Z

The basis codewords for this code can be written as

$$|\bar{0}\rangle = \sum_{M \in S} M|00000\rangle \quad (28)$$

$$|\bar{1}\rangle = \bar{X}|\bar{0}\rangle \quad (29)$$

which gives us the encoded $|\bar{0}\rangle$ as

$$\begin{aligned} |\bar{0}\rangle &= |00000\rangle + M_1|00000\rangle + M_2|00000\rangle + M_3|00000\rangle \\ &+ M_4|00000\rangle + M_1M_2|00000\rangle + M_1M_3|00000\rangle \\ &+ M_1M_4|00000\rangle + M_2M_3|00000\rangle + M_2M_4|00000\rangle \\ &+ M_3M_4|00000\rangle + M_1M_2M_3|00000\rangle \\ &+ M_1M_2M_4|00000\rangle + M_1M_3M_4|00000\rangle \\ &+ M_2M_3M_4|00000\rangle + M_1M_2M_3M_4|00000\rangle \\ &= \frac{1}{4}(|00000\rangle + |10010\rangle + |01001\rangle + |10100\rangle + |01010\rangle \\ &- |11011\rangle - |00110\rangle - |11000\rangle - |11101\rangle - |00011\rangle \\ &- |11110\rangle - |01111\rangle - |10001\rangle - |01100\rangle - |10111\rangle \\ &+ |00101\rangle) \end{aligned} \quad (30)$$

and encoded $|\bar{1}\rangle$ as

$$\begin{aligned} |\bar{1}\rangle &= \bar{X}|\bar{0}\rangle \\ &= \frac{1}{4}(-|11111\rangle - |01101\rangle - |10110\rangle - |01011\rangle - |10101\rangle \\ &+ |00100\rangle + |11001\rangle + |00111\rangle + |00010\rangle + |11100\rangle \\ &+ |00001\rangle + |10000\rangle + |01110\rangle + |10011\rangle + |01000\rangle \\ &- |11010\rangle) \end{aligned} \quad (31)$$

Following the procedure in IV, we put the input qubit $|\psi\rangle$ at the 5th spot followed by $n - 1$ qubits initialized to $|0\rangle$ state. Next, the logical operators are encoded according to

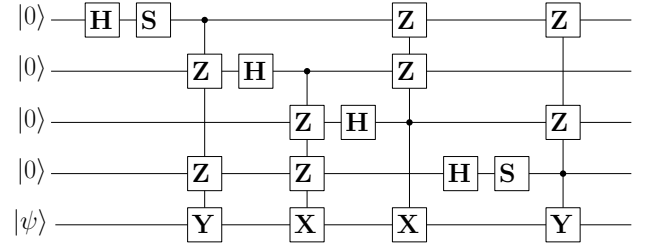


Fig. 10. Encoder for the five-qubit code.

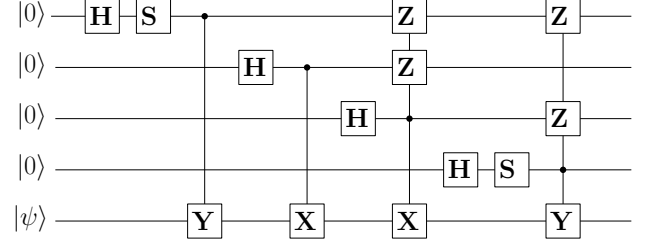


Fig. 11. Modified encoder for the five-qubit code.

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. 10.

Next, we observe that there are four Z gates which are acting on state $|0\rangle$, making those Z gates redundant. After removing those Z gates, the modified encoding circuit is shown in Fig. 11.

On observing carefully, we notice that this circuit is slightly different from the encoder provided in [15]. In the circuit in [15], there is a subtle error (later addressed in an errata [42]) due to which the stabilizers don't commute. To be specific, the H gate (or H followed by S gate) should appear just before the control dots, else the stabilizer operators don't commute. Also, the circuit in [15] uses Z gates instead of S after the H gates when required. However, if one intends to use Z gate, one has to use controlled- Z followed by controlled- X instead of controlled- Y gates. The evolution of the output state of the encoder is mathematically derived for the circuit in Fig. 10 in Appendix A.

2) *Syndrome measurement circuit and error corrector*: The syndrome measurement circuit measures the four stabilizers using four ancilla qubits initialized to the $|0\rangle$ state. There are 5 qubits and each qubit can be affected by X , Y , or Z errors. So, 15 unique error syndromes are possible, which are represented by the final state of the ancilla qubits. The syndromes are shown in Table I.

The syndrome measurement circuit is shown in Fig. 12. Depending on the syndrome, appropriate error correction can be performed by using suitable X , Z , or Y gate on the appropriate qubit.

C. Encoder and syndrome measurement circuit design for Steane code

Hamming codes [43] are linear error correcting codes which have the property that they can detect 1- and 2-bit errors,

TABLE I
SYNDROME TABLE FOR THE 5-QUBIT CODE.

	M_1	M_2	M_3	M_4	Decimal
X I I I I	0	0	0	1	1
Z I I I I	1	0	1	0	10
Y I I I I	1	0	1	1	11
I X I I I	1	0	0	0	8
I Z I I I	0	1	0	1	5
I Y I I I	1	1	0	1	13
I I X I I	1	1	0	0	12
I I Z I I	0	0	1	0	2
I I Y I I	1	1	1	0	14
I I I X I	0	1	1	0	6
I I I Z I	1	0	0	1	9
I I I Y I	1	1	1	1	15
I I I I X	0	0	1	1	3
I I I I Z	0	1	0	0	4
I I I I Y	0	1	1	1	7
I I I I I	0	0	0	0	0

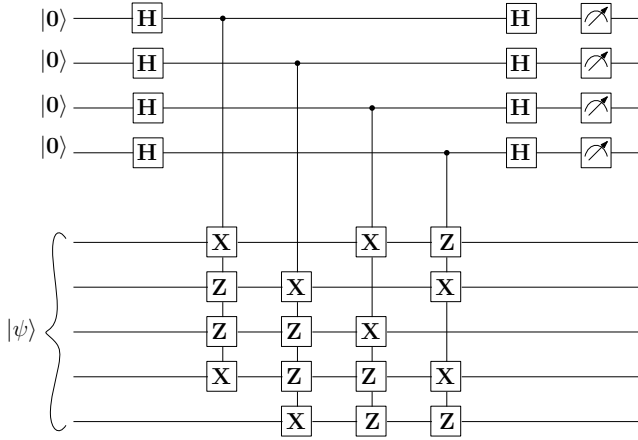


Fig. 12. Syndrome measurement circuit for 5-qubit code.

and can correct 1-bit errors. The $[[7, 4, 3]]$ Hamming code was introduced by Hamming. It encodes 4 bits of data into 7 bits, such that the 3 parity bits provide the ability to detect and correct single bit errors. The generator matrix G and the parity check matrix H of the Hamming code are given as,

$$G = \begin{bmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{bmatrix}, H = \begin{bmatrix} 1101100 \\ 1011010 \\ 0111001 \end{bmatrix} \quad (32)$$

1) *Steane code as the quantum analog of classical Hamming code:* Steane code [44] 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:

$$\begin{array}{l|lllllll} M_1 & X & X & X & X & I & I & I \\ M_2 & X & X & I & I & X & X & I \\ M_3 & X & I & X & I & X & I & X \\ M_4 & Z & Z & Z & Z & I & I & I \\ M_5 & Z & Z & I & I & Z & Z & I \\ M_6 & Z & I & Z & I & Z & I & Z \end{array}.$$

Each of the above generators is a tensor product of 7 Pauli matrices. It should however be noted that tensor product symbols $\otimes$ are often omitted for brevity. The logical operators are $X_L = XXXXX$ and $Z_L = ZZZZZ$. Thus, the two codewords for the Steane code are,

$$\begin{aligned} |0\rangle_L &= \frac{1}{2\sqrt{2}}(|0000000\rangle + |1111000\rangle + |1100110\rangle + |1010101\rangle \\ &\quad + |0011110\rangle + |0101101\rangle + |0110011\rangle + |1001011\rangle) \end{aligned} \quad (33)$$

$$\begin{aligned} |1\rangle_L &= X_L|0\rangle \\ &= \frac{1}{2\sqrt{2}}(|0000111\rangle + |1111111\rangle + |1100001\rangle + |1010010\rangle \\ &\quad + |0011001\rangle + |0101010\rangle + |0110100\rangle + |1001100\rangle) \end{aligned} \quad (34)$$

2) *Encoder design for Steane code:* The parity check matrix and generator matrix for the $(7,4)$ Hamming code are described as follows:

$$H = \begin{bmatrix} 1101100 \\ 1011010 \\ 0111001 \end{bmatrix}, G = \begin{bmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{bmatrix} \quad (35)$$

We can verify that H is contained in G . Thus, it satisfies the dual-containing criterion for construction of CSS codes. In the binary formalism, the parity check matrix for the augmented parity check matrix can be written as

$$H_q = \left[\begin{array}{ccc|ccc} 1101100 & & & 0000000 & & \\ 1011010 & & & 0000000 & & \\ 0111001 & & & 0000000 & & \\ 0000000 & & & 1101100 & & \\ 0000000 & & & 1011010 & & \\ 0000000 & & & 0111011 & & \end{array} \right] \quad (36)$$

Our aim is to transform the above parity check matrix to the standard form as described in Section IV. First, some columns are swapped, which is equivalent to swapping qubit positions. The columns (or equivalently the qubit positions) are swapped in following order $1 \leftarrow 5$, $2 \leftarrow 6$, $3 \leftarrow 7$, $4 \leftarrow 1$, $5 \leftarrow 2$, $6 \leftarrow 4$, $7 \leftarrow 3$. This gives us the new augmented H matrix

$$H_q = \left[\begin{array}{ccc|ccc} 1001110 & & & 0000000 & & \\ 0101011 & & & 0000000 & & \\ 0010111 & & & 0000000 & & \\ 0000000 & & & 1001110 & & \\ 0000000 & & & 0101011 & & \\ 0000000 & & & 0010111 & & \end{array} \right] \quad (37)$$

Performing the operation $R_5 \rightarrow R_5 + R_4$

$$H_q = \left[\begin{array}{ccc|ccc} 1001110 & & & 0000000 & & \\ 0101011 & & & 0000000 & & \\ 0010111 & & & 0000000 & & \\ 0000000 & & & 1001110 & & \\ 0000000 & & & 1100101 & & \\ 0000000 & & & 0010111 & & \end{array} \right]$$

Performing the operation $R_6 \rightarrow R_6 + R_5$

$$H_q = \left[\begin{array}{ccc|ccc} 1001110 & & & 0000000 & & \\ 0101011 & & & 0000000 & & \\ 0010111 & & & 0000000 & & \\ 0000000 & & & 1001110 & & \\ 0000000 & & & 1100101 & & \\ 0000000 & & & 1110010 & & \end{array} \right] \quad (39)$$

Performing the operation $R_4 \rightarrow R_4 + R_5$

$$H_q = \left[\begin{array}{ccc|ccc} 1001110 & & & 0000000 & & \\ 0101011 & & & 0000000 & & \\ 0010111 & & & 0000000 & & \\ 0000000 & & & 0101011 & & \\ 0000000 & & & 1100101 & & \\ 0000000 & & & 1110010 & & \end{array} \right] \quad (40)$$

Performing the operation $R_4 \rightarrow R_4 + R_6$ we get the standard form H_s as

$$H_s = \left[\begin{array}{ccc|ccc} 1001110 & & & 0000000 & & \\ 0101011 & & & 0000000 & & \\ 0010111 & & & 0000000 & & \\ 0000000 & & & 1011001 & & \\ 0000000 & & & 1100101 & & \\ 0000000 & & & 1110010 & & \end{array} \right] \quad (41)$$

We have the following from H_s .

$$\begin{aligned} I_1 = I_2 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, B = C_1 = 0_{3 \times 3}, C_2 = 0_{3 \times 1}, \\ D &= \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, E = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. \end{aligned} \quad (42)$$

The stabilizers of the code can be written as

$$\begin{array}{l|cccccc} M_1 & X & I & I & X & X & X & I \\ M_2 & I & X & I & X & I & X & X \\ M_3 & I & I & X & I & X & X & X \\ M_4 & Z & I & I & Z & Z & Z & I \\ M_5 & I & Z & I & Z & I & Z & Z \\ M_6 & I & I & Z & I & Z & Z & Z \end{array}$$

The logical operators can be evaluated as described in Section IV, producing

$$\bar{X} = [0001101 \mid 0000000] \quad (43)$$

$$\bar{Z} = [0000000 \mid 0110001] \quad (44)$$

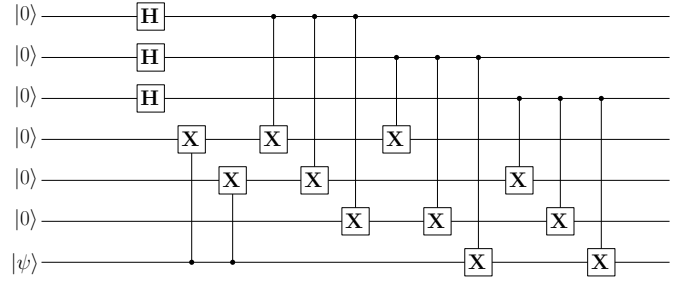


Fig. 13. Encoder for the Steane code.

The encoding circuit can be generated from $\bar{X}$ and H_s by applying Algorithm 1. The qubit to be encoded is placed at the 7th position, followed by 6 qubits initialized to $|0\rangle$. First, the $\bar{X}|0000000\rangle$ state is obtained by applying $\bar{X}$ conditioned on the last qubit. Applying Algorithm 1, the encoder circuit thus obtained is shown in Fig. 13.

3) *Syndrome measurement circuit and error corrector for Steane code*: The syndrome measurement circuit measures all the six stabilizers using six ancilla qubits. The syndromes are unique as shown in Table II. Each qubit in the 7-qubit Steane code can be affected by three kind of errors, namely X , Y , and Z errors. So, there are 21 different types of single qubit errors possible, each of which gives a different syndrome as shown in Table II. The M_1 - M_6 values in the table can be explained by the following example. Let us take the fifth row of the table for example, i.e., $IZIIIII$, which implies that a Z error has occurred on the second qubit. It is easy to observe that $IZIIIII$ anti-commutes with M_1 and M_2 , while it commutes with M_3 , M_4 , M_5 , and M_6 . Thus, we get a syndrome of 110000. It can be observed that each syndrome is unique as shown in Table II. Since this code uses only 21 different syndromes for various single qubit errors, the rest of the syndromes are unused, unlike the 5-qubit code where all the syndromes are used.

The syndrome measurement circuit is shown in Fig. 14. Six 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 000000 implies that no error has occurred. It should also be noted that any 6 bit syndrome other than the syndromes mentioned in Table II implies the occurrence of more than a single qubit error, which cannot be corrected using the Steane code.

VI. NNC CIRCUIT DESIGN

In the circuits we discussed in the previous sections, we assume that any particular qubit can interact with any other qubit. This implies that there can be a 2-qubit gate between any two arbitrary qubits. If interacting qubits are not close to each other, the underlying qubit interactions become increasingly prone to noise. Many of the fault-tolerant technologies thus rely on the nearest neighbour compliance, where the interacting qubits need to be adjacent to each other before they can interact via 2-qubit gates. Many techniques have been

TABLE II
SYNDROME TABLE FOR STEANE CODE.

	M_1	M_2	M_3	M_4	M_5	M_6	Decimal value
X I I I I I I	0	0	0	1	0	0	4
Z I I I I I I	1	0	0	0	0	0	32
Y I I I I I I	1	0	0	1	0	0	36
I X I I I I I	0	0	0	0	1	0	2
I Z I I I I I	0	1	0	0	0	0	16
I Y I I I I I	0	1	0	0	1	0	18
I I X I I I I	0	0	0	0	0	1	1
I I Z I I I I	0	0	1	0	0	0	8
I I Y I I I I	0	0	1	0	0	1	9
I I I X I I I	0	0	0	1	1	0	6
I I I Z I I I	1	1	0	0	0	0	48
I I I Y I I I	1	1	0	1	1	0	54
I I I I X I I	0	0	0	1	0	1	5
I I I I Z I I	1	0	1	0	0	0	40
I I I I Y I I	1	0	1	1	0	1	45
I I I I I X I	0	0	0	1	1	1	7
I I I I I Z I	1	1	1	0	0	0	56
I I I I I Y I	1	1	1	1	1	1	63
I I I I I I X	0	0	0	0	1	1	3
I I I I I I Z	0	1	1	0	0	0	24
I I I I I I Y	0	1	1	0	1	1	27
I I I I I I I	0	0	0	0	0	0	0

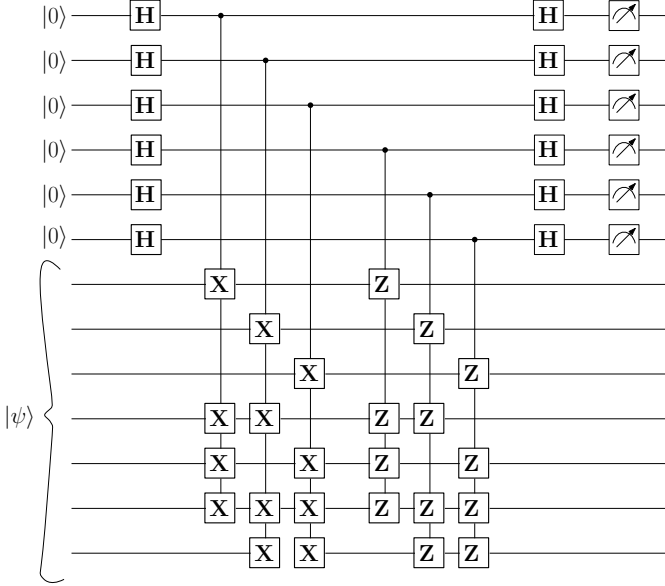


Fig. 14. Syndrome measurement circuit for Steane code.

proposed in the past to make 1-D and 2-D NNC architectures [45], [46], [47], [48], [49], [50], [51], [36]. In a 2-D array of qubits in Fig. 15, the qubits at the corners and edges can interact with 2 or 3 qubits, while the rest of the qubits can interact with their 4 closest neighbors.

To design a NNC circuit, we need to use swap gates to bring the qubits adjacent to each other [36]. It should be noted that the qubits are not moved physically. Their states are swapped which is equivalent to moving them to adjacent positions without doing it physically. A swap gate requires 3 CNOT gates. Thus, it is important to position the qubits and perform the operations in such a way that the number of swap gates is minimized. A swap gate is shown in Fig. 16.

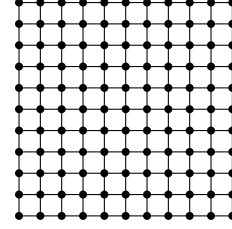


Fig. 15. 2-D array of qubits (represented by black dots). The qubits can only interact with their nearest neighbours. The qubits on corners and edges can interact with 2 and 3 qubits respectively. The rest of the qubits can interact with 4 qubits each.

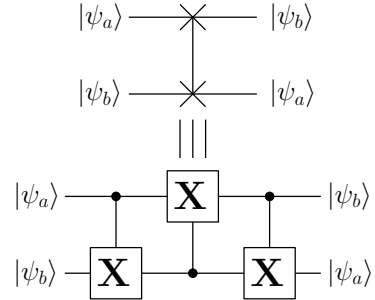


Fig. 16. Symbol of a swap gate (top). A swap gate circuit implemented using 3 CNOT gates (bottom).

The procedure to design a NNC circuit is described as follows:

- The qubits in the original circuit are indexed from 1 to n .
- Next, we design a 2-D configuration with the indexed qubits such that for maximum number of 2-qubit gates, the interacting qubits are already neighbors.
- Subsequently, swap gates are introduced to bring the set of qubits adjacent to each other, mindful of the next set

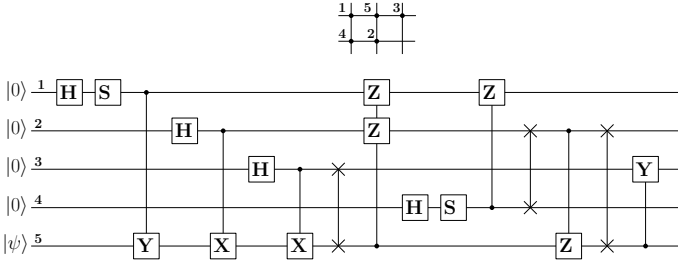


Fig. 17. NNC circuit for the five-qubit encoder. The positions of the qubits in a 2-D array is shown at the top. The circuit requires 3 swap gates.

of 2-qubit gates in the sequence.

- The process goes on till the whole circuit has been designed.

It should be noted that the above process may not be optimum, especially if the number of qubits in the circuit is large. For circuits containing a small number of qubits, we can expect optimum results.

A NNC circuit for the five-qubit encoder is shown in Fig. 17. The initial qubit position is shown at the top. Three swap gates are required to implement the circuit. We also designed a NNC circuit for the syndrome measurement circuit for the five-qubit code as shown in Fig. 18. The initial qubit configuration in the 2-D array is shown at the top. Eight swap gates are required for the circuit; these require 24 CNOT gates.

Similar to the five-qubit code, NNC circuit can be designed for the Steane code encoder and syndrome measurement circuit as well. NNC circuits for Steane code encoder and syndrome measurement circuits are shown in Figs. 19 and 20, respectively.

The encoder circuits for the 5-qubit code and Steane code, which contain 5 qubits and 7 qubits, respectively, were designed using a small number of swap gates. However, the syndrome measurement circuits for the above codes contain 9 qubits and 13 qubits, respectively, and thus require a large number of swap gates. NNC circuits for those, as shown in Figs. 18 and 20, may not be optimal.

VII. RESULTS

The combined encoder and decoder circuits were simulated using IBM Qiskit. Errors were introduced at different positions to test for correctability. X , Y , and Z errors were introduced in the circuit by adding the corresponding Pauli gate after the encoder circuit. The measured syndromes varied depending on the type of error introduced. An exact match was found between the syndromes measured and the corresponding syndromes shown in Tables I and II for the five-qubit and Steane code, respectively. Depending on the syndromes, the corresponding Pauli gate can be used to correct the error.

Another important parameter to measure the efficiency of the quantum circuits is the number of single and multiple qubit gates used in the quantum circuits. We list the number of gates used in the quantum circuits presented in this paper in Table III. We observe that all these circuits were designed using H , S , CNOT, CY , and CZ gates. The NNC circuit for the 5-qubit encoder requires 3 swap gates, or equivalently 9 extra CNOT

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

Parameters	H gate	S gate	CNOT gates	CY gates	CZ gates
Five qubit encoder	4	2	2	2	4
Five qubit encoder NNC	4	2	11	2	4
Five qubit syndrome measurement	8	0	8	0	8
Five qubit syndrome measurement NNC	8	0	32	0	8
Steane code encoder	3	0	11	0	0
Steane code encoder NNC	3	0	17	0	0
Steane code syndrome measurement	12	0	12	0	12
Steane code syndrome measurement NNC	12	0	60	0	12

gates. Similarly, the NNC circuit for the Steane code encoder requires 2 swap gates, or 6 extra CNOT gates. However, the syndrome measurement circuits for both the 5-qubit code and 7-qubit code require a lot more swap gates. This is because these circuits contain 9 and 13 qubits, respectively. Thus, on a 2-D array, it requires a larger number of steps to bring the required qubits adjacent to each other.

VIII. CONCLUSIONS

In this paper, we discussed a detailed procedure for the construction of encoding and decoding circuits for stabilizer codes. We started with Shor's 9-qubit code and analyzed the code using stabilizer formalism. For using stabilizer codes in practical quantum computers, it is essential to design efficient encoding and decoding circuits. We reviewed a systematic procedure for encoder circuit design for stabilizer codes, and using that we formulated an algorithm to generate encoding circuits for general stabilizer code. For the decoder, we used a syndrome measurement circuit, and depending on the measured syndromes, we corrected the error using suitable Pauli gates. To enhance reliability of the designed circuits, it is important that the circuit is NNC, i.e., for multiple qubit operations, the qubits must be adjacent to each other. We explored NNC circuits for both the five-qubit code and Steane code.

Design of quantum ECCs for codes such as RS codes [25], [26], [27], LDPC codes [52], [53], [54], hypergraph-product codes and homological product codes [55], [56], tensor product of quantum and classical codes [57], [58], and polar codes [28] remains an active area of research. Furthermore, design of quantum circuits for these codes remains a topic

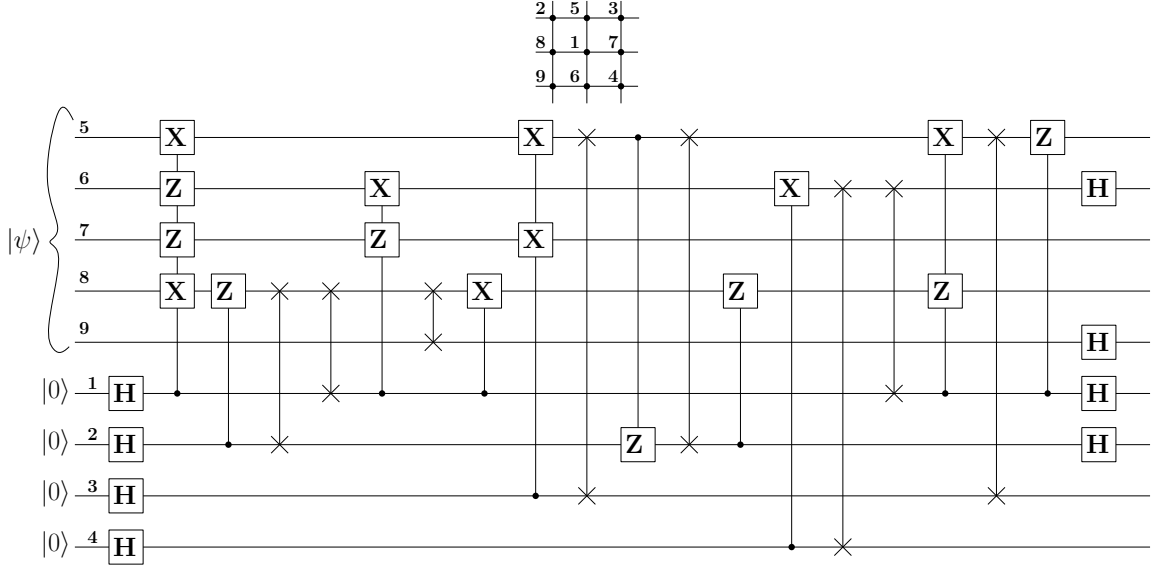


Fig. 18. NNC circuit for the five-qubit syndrome measurement circuit. The positions of the qubits in a 2-D array is shown at the top. The circuit requires 8 swap gates.

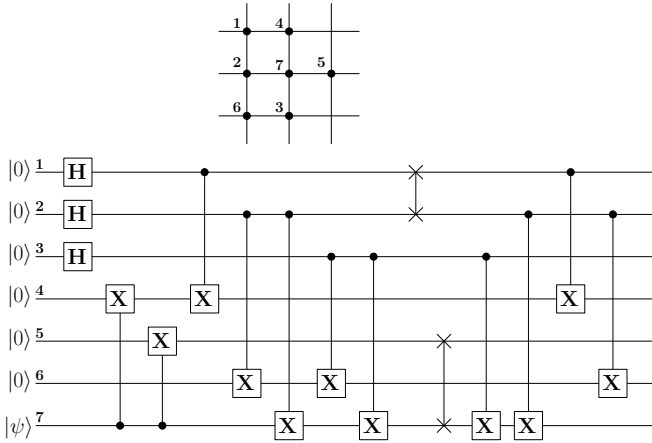


Fig. 19. Nearest neighbour compliant circuit for the Steane code encoder. The positions of the qubits in a 2-D array is shown at the top. The circuit requires 2 swap gates.

of future research. Future work needs to be directed towards optimization of quantum circuits for these codes with respect to the number of gates subject to NNC constraint, and towards development of automated tools for design of these circuits.

IX. ACKNOWLEDGMENT

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

APPENDIX A

EVALUATION OF THE OUTPUT STATE OF THE 5-QUBIT ENCODER CIRCUIT

A good exercise would be to evaluate the output state of the encoder circuit and verify if it matches with $|\bar{0}\rangle$ and $|\bar{1}\rangle$ states in equations 30 and 31.

The initial state ψ_0 when the fifth qubit is set to $|0\rangle$ is $\psi_0 = |00000\rangle$.

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

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|00000\rangle + i|10000\rangle) \quad (45)$$

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

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2}}(|00000\rangle + (i \cdot i)|10001\rangle) \\ &= \frac{1}{\sqrt{2}}(|00000\rangle - |10001\rangle) \end{aligned} \quad (46)$$

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

$$|\psi_3\rangle = \frac{1}{2}(|00000\rangle + |01000\rangle - |10001\rangle - |11001\rangle) \quad (47)$$

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

$$|\psi_4\rangle = \frac{1}{2}(|00000\rangle + |01001\rangle - |10001\rangle - |11000\rangle) \quad (48)$$

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

$$\begin{aligned} |\psi_5\rangle &= \frac{1}{2\sqrt{2}}(|00000\rangle + |00100\rangle + |01001\rangle + |01101\rangle \\ &\quad - |10001\rangle - |10101\rangle - |11000\rangle - |11100\rangle) \end{aligned} \quad (49)$$

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

$$\begin{aligned} |\psi_6\rangle &= \frac{1}{2\sqrt{2}}(|00000\rangle + |00101\rangle + |01001\rangle - |01100\rangle \\ &\quad - |10001\rangle + |10100\rangle - |11000\rangle - |11101\rangle) \end{aligned} \quad (50)$$

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

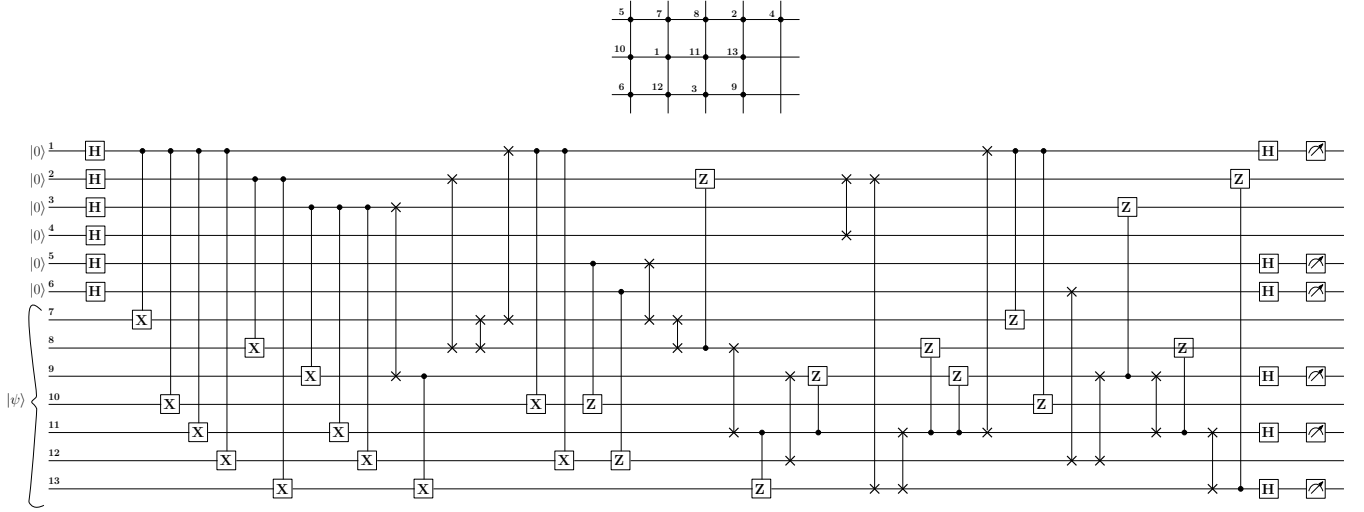


Fig. 20. NNC circuit for the Steane code syndrome measurement circuit. The positions of the qubits in a 2-D array is shown at the top. The circuit requires 16 swap gates.

$$\begin{aligned}
 |\psi_7\rangle = & \frac{1}{4}(|00000\rangle + i|00010\rangle + |00101\rangle + i|00111\rangle \\
 & + |01001\rangle + i|01011\rangle - |01100\rangle - i|01110\rangle \\
 & - |10001\rangle - i|10011\rangle + |10100\rangle + i|10110\rangle \\
 & - |11000\rangle - i|11010\rangle - |11101\rangle - i|11111\rangle) \quad (51)
 \end{aligned}$$

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

$$\begin{aligned}
 |\psi_8\rangle = & \frac{1}{4}(|00000\rangle + i \cdot i|00011\rangle + |00101\rangle \\
 & + i(-i \cdot -1)|00110\rangle + |01001\rangle + i(-i)|01010\rangle \\
 & - |01100\rangle - i(-1 \cdot i)|01111\rangle - |10001\rangle \\
 & - i(-1 \cdot -i)|10010\rangle + |10100\rangle + i(i \cdot -1 \cdot -1)|10111\rangle \\
 & - |11000\rangle - i(-1 \cdot i)|11011\rangle - |11101\rangle \\
 & - i(-1 \cdot -1 \cdot -i)|11110\rangle) \quad (52) \\
 = & \frac{1}{4}(|00000\rangle - |00011\rangle + |00101\rangle - |00110\rangle \\
 & + |01001\rangle + |01010\rangle - |01100\rangle - |01111\rangle \\
 & - |10001\rangle + |10010\rangle + |10100\rangle - |10111\rangle \\
 & - |11000\rangle - |11011\rangle - |11101\rangle - |11110\rangle) \quad (53)
 \end{aligned}$$

We observe that $|\psi_8\rangle$ matches with state $|\bar{0}\rangle$ in equation (30).

Now, we will verify state $|\bar{1}\rangle$. The initial state ψ_0 when the fifth qubit is set to $|1\rangle$ is $\psi_0 = |00001\rangle$.

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

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|00001\rangle + i|10001\rangle) \quad (54)$$

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

$$\begin{aligned}
 |\psi_2\rangle = & \frac{1}{\sqrt{2}}(|00001\rangle + (i \cdot -i)|10000\rangle) \\
 = & \frac{1}{\sqrt{2}}(|00001\rangle + |10000\rangle) \quad (55)
 \end{aligned}$$

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

$$|\psi_3\rangle = \frac{1}{2}(|00001\rangle + |01001\rangle + |10000\rangle + |11000\rangle) \quad (56)$$

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

$$|\psi_4\rangle = \frac{1}{2}(|00001\rangle + |01000\rangle + |10000\rangle + |11001\rangle) \quad (57)$$

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

$$\begin{aligned}
 |\psi_5\rangle = & \frac{1}{2\sqrt{2}}(|00001\rangle + |00101\rangle + |01000\rangle + |01100\rangle \\
 & + |10000\rangle + |10100\rangle + |11001\rangle + |11101\rangle) \quad (58)
 \end{aligned}$$

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

$$\begin{aligned}
 |\psi_6\rangle = & \frac{1}{2\sqrt{2}}(|00001\rangle + |00100\rangle + |01000\rangle - |01101\rangle \\
 & + |10000\rangle - |10101\rangle + |11001\rangle + |11100\rangle) \quad (59)
 \end{aligned}$$

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

$$\begin{aligned}
 |\psi_7\rangle = & \frac{1}{4}(|00001\rangle + i|00011\rangle + |00100\rangle + i|00110\rangle \\
 & + |01000\rangle + i|01010\rangle - |01101\rangle - i|01111\rangle \\
 & + |10000\rangle + i|10010\rangle - |10101\rangle - i|10111\rangle \\
 & + |11001\rangle + i|11011\rangle + |11100\rangle + i|11110\rangle) \quad (60)
 \end{aligned}$$

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

$$\begin{aligned} |\psi_8\rangle &= \frac{1}{4}(|00001\rangle + i \cdot (-i)|00010\rangle + |00100\rangle \\ &\quad + i(-1 \cdot i)|00111\rangle + |01000\rangle + i(i)|01011\rangle \\ &\quad - |01101\rangle - i(-1 \cdot -i)|01110\rangle + |10000\rangle \\ &\quad + i(-1 \cdot i)|10011\rangle - |10101\rangle - i(-1 \cdot -1 \cdot -i)|10110\rangle \\ &\quad + |11001\rangle + i(-1 \cdot -i)|11010\rangle + |11100\rangle \\ &\quad + i(-1 \cdot -1 \cdot i)|11111\rangle) \end{aligned} \quad (61)$$

$$\begin{aligned} &= \frac{1}{4}(|00001\rangle + |00010\rangle + |00100\rangle + |00111\rangle \\ &\quad + |01000\rangle - |01011\rangle - |01101\rangle + |01110\rangle \\ &\quad + |10000\rangle + |10011\rangle - |10101\rangle - |10110\rangle \\ &\quad + |11001\rangle - |11010\rangle + |11100\rangle - |11111\rangle) \end{aligned} \quad (62)$$

We observe that $|\psi_8\rangle$ matches with state $|\bar{1}\rangle$ in equation (31).

REFERENCES

- [1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
- [2] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme *et al.*, “Evidence for the utility of quantum computing before fault tolerance,” *Nature*, vol. 618, no. 7965, pp. 500–505, 2023.
- [3] R. P. Feynman, “Quantum mechanical computers,” *Optics news*, vol. 11, no. 2, pp. 11–20, 1985.
- [4] R. P. Feynman *et al.*, “Simulating physics with computers,” *Int. J. Theor. phys.*, vol. 21, no. 6/7, 1982.
- [5] 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.
- [6] 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.
- [7] J. Gambetta, “Quantum-centric supercomputing: The next wave of computing,” *IBM Research Blog*, 2022.
- [8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [9] W. K. Wootters and W. H. Zurek, “A single quantum cannot be cloned,” *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.
- [10] D. Dieks, “Communication by EPR devices,” *Physics Letters A*, vol. 92, no. 6, pp. 271–272, 1982.
- [11] A. Kumar Pati and S. L. Braunstein, “Impossibility of deleting an unknown quantum state,” *Nature*, vol. 404, no. 6774, pp. 164–165, 2000.
- [12] 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.
- [13] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Physical review A*, vol. 52, no. 4, p. R2493, 1995.
- [14] D. Gottesman, “Class of quantum error-correcting codes saturating the quantum hamming bound,” *Physical Review A*, vol. 54, no. 3, p. 1862, 1996.
- [15] —, *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.
- [16] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Physical Review A*, vol. 54, no. 2, p. 1098, 1996.
- [17] A. M. Steane, “Error correcting codes in quantum theory,” *Physical Review Letters*, vol. 77, no. 5, p. 793, 1996.
- [18] E. Knill and R. Laflamme, “Theory of quantum error-correcting codes,” *Physical Review A*, vol. 55, no. 2, p. 900, 1997.
- [19] A. Y. Kitaev, “Fault-tolerant quantum computation by anyons,” *Annals of physics*, vol. 303, no. 1, pp. 2–30, 2003.
- [20] S. B. Bravyi and A. Y. Kitaev, “Quantum codes on a lattice with boundary,” *arXiv preprint quant-ph/9811052*, 1998.
- [21] T. Brun, I. Devetak, and M.-H. Hsieh, “Correcting quantum errors with entanglement,” *science*, vol. 314, no. 5798, pp. 436–439, 2006.
- [22] 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.
- [23] M. M. Wilde, “Quantum coding with entanglement,” Ph.D. dissertation, University of Southern California, 2008.
- [24] 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.
- [25] 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.
- [26] S. A. Aly, “Asymmetric quantum BCH codes,” in *2008 International Conference on Computer Engineering & Systems*. IEEE, 2008, pp. 157–162.
- [27] G. G. La Guardia, “Asymmetric quantum Reed-Solomon and generalized Reed-Solomon codes,” *Quantum Information Processing*, vol. 11, pp. 591–604, 2012.
- [28] 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.
- [29] P. J. Nadkarni and S. S. Garani, “Non-binary entanglement-assisted stabilizer codes,” *Quantum Information Processing*, vol. 20, no. 8, p. 256, 2021.
- [30] —, “Encoding of nonbinary entanglement-unassisted and assisted stabilizer codes,” *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–22, 2021.
- [31] —, “Entanglement-assisted reed-solomon codes over qudits: theory and architecture,” *Quantum Information Processing*, vol. 20, pp. 1–68, 2021.
- [32] —, “Coding analog of superadditivity using entanglement-assisted quantum tensor product codes over $\mathbb{F}_{p^k}$,” *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–17, 2020.
- [33] 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.
- [34] J. C. Garcia-Escartin and P. Chamorro-Posada, “Equivalent quantum circuits,” *arXiv preprint arXiv:1110.2998*, 2011.
- [35] A. Mondal and K. K. Parhi, “Systematic design and optimization of quantum circuits for stabilizer codes,” *arXiv preprint arXiv:2309.12373*, 2023.
- [36] J. Ding and S. Yamashita, “Exact synthesis of nearest neighbor compliant quantum circuits in 2-d architecture and its application to large-scale circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 5, pp. 1045–1058, 2019.
- [37] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, “Mixed-state entanglement and quantum error correction,” *Physical Review A*, vol. 54, no. 5, p. 3824, 1996.
- [38] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, “Perfect quantum error correcting code,” *Physical Review Letters*, vol. 77, no. 1, p. 198, 1996.
- [39] “Ibm quantum,” <https://quantum-computing.ibm.com/>.
- [40] A. Peres, “Reversible logic and quantum computers,” *Physical review A*, vol. 32, no. 6, p. 3266, 1985.
- [41] P. J. Nadkarni, “Entanglement-assisted additive qudit stabilizer codes,” Ph.D. dissertation, Indian Institute of Science, 2021.
- [42] D. Gottesman, *Errata in Stabilizer Codes and Quantum Error Correction*. <https://thesis.library.caltech.edu/2900/1/Errata.pdf>, 1997.
- [43] R. W. Hamming, “Error detecting and error correcting codes,” *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [44] 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.
- [45] M. Saeedi, R. Wille, and R. Drechsler, “Synthesis of quantum circuits for linear nearest neighbor architectures,” *Quantum Information Processing*, vol. 10, pp. 355–377, 2011.
- [46] A. Chakrabarti, S. Sur-Kolay, and A. Chaudhury, “Linear nearest neighbor synthesis of reversible circuits by graph partitioning,” *arXiv preprint arXiv:1112.0564*, 2011.
- [47] A. Matsuo and S. Yamashita, “Changing the gate order for optimal lnn conversion,” in *Reversible Computation: Third International Workshop*,

RC 2011, Gent, Belgium, July 4-5, 2011. Revised Papers 3. Springer, 2012, pp. 89–101.

- [48] A. Shafaei, M. Saeedi, and M. Pedram, “Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures,” in *Proceedings of the 50th annual design automation conference*, 2013, pp. 1–6.
- [49] P. Pham and K. M. Svore, “A 2D nearest-neighbor quantum architecture for factoring in polylogarithmic depth,” *Quantum Inf. Comput.*, vol. 13, no. 11-12, pp. 937–962, 2013.
- [50] M. G. Alfaiakawi, I. Ahmad, and S. Hamdan, “Harmony-search algorithm for 2d nearest neighbor quantum circuits realization,” *Expert Systems with Applications*, vol. 61, pp. 16–27, 2016.
- [51] C.-C. Lin, S. Sur-Kolay, and N. K. Jha, “Paqcs: Physical design-aware fault-tolerant quantum circuit synthesis,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 7, pp. 1221–1234, 2014.
- [52] M. M. Wilde, *Quantum information theory*. Cambridge university press, 2013.
- [53] 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.
- [54] 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.
- [55] 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.
- [56] 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.
- [57] M. B. Hastings, “Weight reduction for quantum codes,” *arXiv preprint arXiv:1611.03790*, 2016.
- [58] 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.



Keshab K. Parhi (Fellow, IEEE) received the B.Tech. degree from the Indian Institute of Technology (IIT), Kharagpur, in 1982, the M.S. 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).



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