Quantum Error Correction For Dummies

Avimita Chatterjee
CSE Department
Penn State University
PA, USA
amc8313@psu.edu

Koustubh Phalak
CSE Department
Penn State University
PA, USA
krp5448@psu.edu

Swaroop Ghosh
School of EECS
Penn State University
PA, USA
szg212@psu.edu

2023 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3503-4323-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/QCE57702.2023.00017

***Abstract*—In the current Noisy Intermediate Scale Quantum (NISQ) era of quantum computing, qubit technologies are prone to imperfections, giving rise to various errors such as gate errors, decoherence/dephasing, measurement errors, leakage, and crosstalk. These errors present challenges in achieving error-free computation within NISQ devices. A proposed solution to this issue is Quantum Error Correction (QEC), which aims to rectify the corrupted qubit state through a three-step process: (i) detection: identifying the presence of an error, (ii) decoding: pinpointing the location(s) of the affected qubit(s), and (iii) correction: restoring the faulty qubits to their original states. QEC is an expanding field of research that encompasses intricate concepts. In this paper, we aim to provide a comprehensive review of the historical context, current state, and future prospects of Quantum Error Correction, tailored to cater to computer scientists with limited familiarity with quantum physics and its associated mathematical concepts. In this work, we, (a) explain the foundational principles of QEC and explore existing Quantum Error Correction Codes (QECC) designed to correct errors in qubits, (b) explore the practicality of these QECCs with regard to implementation and error correction quality, and (c) highlight the challenges associated with implementing QEC within the context of the current landscape of NISQ computers.**

***Index Terms*—Quantum error correction, Quantum computing, Error correction codes, Repetition codes, Topological codes**

I. Introduction

In recent years, quantum computing has garnered substantial interest owing to its potential to revolutionize diverse industry sectors, including cybersecurity, pharmaceuticals, finance, and manufacturing [1]. Quantum computers employ qubits for the representation and computation of information. Qubits harness quantum-mechanical properties, such as superposition, entanglement, and interference, which theoretically endow quantum computers with a speed advantage over classical algorithms and computing systems. Quantum computing algorithms have been employed in various fields, including quantum machine learning [2], optimization [3] and quantum chemistry [4], [5]. Quantum computers are realized through a diverse range of qubit technologies, such as trapped ion qubits [6], [7], photonic qubits [8], [9], superconducting qubits [10], [11], quantum dots qubits [12] and many more. However, for all these technologies, it is a challenging task to entirely isolate qubits from external noise, making errors in quantum computers inevitable. Consequently, quantum computers necessitate some form of error correction.

The established classical error correction theories have resulted in a high error tolerance for classical computers [22]. However, the adaptation of existing classical error correction techniques for quantum computing is challenging due to the no-cloning theorem, which prohibits the duplication of qubits in contrast to classical bits that can be copied [23]. In addition, the measurement of qubits is subject to limitations as the act of measuring a qubit leads to the collapse of its wavefunction [24], resulting in the loss of its quantum state. The year 1995 saw the proposal of the first Quantum Error Correction (QEC) scheme by Peter Shor [13]. Fig.1 illustrates the chronology of significant advancements in the field of quantum error correction codes. For a comprehensive comparison of these significant developments, please refer to the Table IV.

Multiple review articles exist on QEC [25]–[27] and associated topics [28]–[30]. Nonetheless, these articles can be difficult to grasp, as they often involve complex mathematical concepts and implicitly assume that readers possess pre-existing knowledge about the domain. The aim of this work is to provide a comprehensive yet accessible introduction to the fundamental concepts of quantum error correction for researchers who may not have an extensive background in quantum physics or related mathematical fields. It is not a mandatory pre-requisite for readers to have prior knowledge of Quantum Error Correction (QEC) for this review. However, it is assumed that the readers are familiar with quantum circuit notations as described in [31], including basic measurement operations, controlled-NOT gate $(CNOT)$, and Hadamard gate (H). [1]

We commence by introducing the foundational concepts of quantum computing in Section II, including qubits, quantum gates, quantum circuits, types of errors, and the distinctions between classical and quantum error correction. Section III elucidates the essential principles of QEC, beginning with repetition codes and progressing to stabilizer formalism and topological codes. Sections IV and V addresses the practical applications of Quantum Error Correction Codes (QECC) and the challenges associated with their future development, respectively.

[1]As a notational shorthand, we often remove the tensor product sign, $\otimes$ when we denote the tensor product of multiple operators. For example, $X \otimes Y \otimes Z$ may be interchangeably written as XYZ.

979-8-3503-4323-6/23/$31.00 ©2023 IEEE
DOI 10.1109/QCE57702.2023.00017

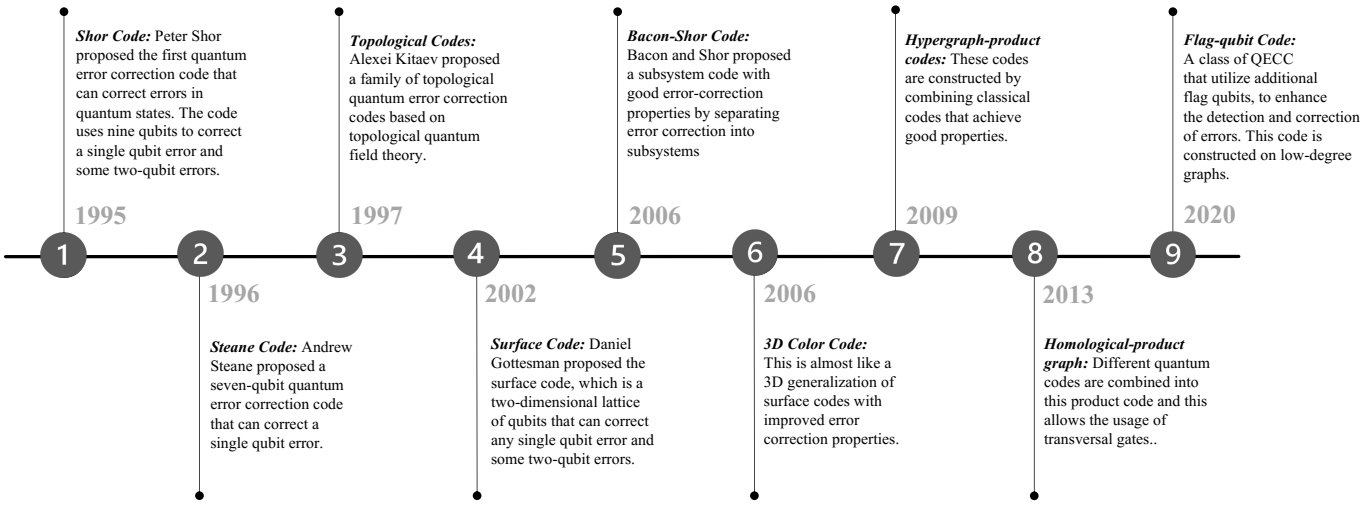

Fig. 1. Advancement and Evolution of Quantum Error Correction. ①: [13], ②: [14], ③: [15], ④: [16], ⑤: [17], ⑥: [18], ⑦: [19], ⑧: [20], ⑨: [21].

II. Preliminaries

A. An Overview of Quantum Computing

*a) **Qubits**:* Qubits are fundamental units of a quantum computer that are analogous to classical bits. In general, a qubit is represented by a quantum state $|\psi\rangle = \left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$, where α and β are complex amplitudes such that $|\alpha|^2$ represents the probability of qubit being measured to classical 0 and $|\beta|^2$ represents the probability of qubit being measured to classical 1 i.e. $|\alpha|^2 + |\beta|^2 = 1$. Every qubit has two fundamental basis states, $|0\rangle$ ($\alpha = 1$, $\beta = 0$) and $|1\rangle$ ($\alpha = 0$, $\beta = 1$). Qubit undergoes unitary gate operations that change its state and finally, a measurement operation is performed to collapse the qubit to either 0 or 1 values.

*b) **Quantum gates**:* Quantum gates are unitary matrix operations that operate on single or many qubits to change their states. They are realized using different methods based on the qubit technology such as using microwave pulses in superconducting qubits, laser pulses in trapped ion and quantum dots, and radio frequency pulses in Nuclear Magnetic Resonance (NMR) qubits. The gate operation speeds also vary with technology, ranging from picoseconds (photonic qubits) to a few seconds (NMR qubits) [32]. Single-qubit gates, for example, include the X (NOT) gate, H (Hadamard) gate, and rotation gates such as R_X, R_Y, R_Z, and U gate. Two-qubit gates include the CNOT (controlled-NOT) gate, Toffoli gate, controlled rotation gates, and Peres gate [33].

*c) **Quantum circuit**:* A quantum circuit is an ordered sequence of gate operations performed over time. A quantum circuit comprises of state initialization/preparation to prepare the initial state of the qubits, which are then transformed to the desired state using gate operations in the circuit and finally measured using a measurement gate. All quantum operations are performed in quantum Hilbert space [24] and the high-level gates in the circuit e.g., Tofolli are broken down into a native gate set of the quantum hardware prior to execution. This process is referred to as transpilation.

B. Types of Errors in Quantum Computing

In quantum computing, noise refers to any unwanted influences or external factors that can introduce errors and disrupt the quantum information stored in qubits. Primarily, there are two types of errors: bit-flip errors and phase-flip errors [31]. Bit flip error, also known as X error, occurs when the state of the qubit is flipped i.e. $|0\rangle$ changes to $|1\rangle$ and vice-versa. On the other hand phase flip error, also known as Z error, involves the sign of the qubit's phase i.e. $|1\rangle$ changes to $-|1\rangle$ but $|0\rangle$ remains $|0\rangle$. To sum up if we have a basis state, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then $X|\psi\rangle = \alpha|1\rangle + \beta|0\rangle$ and $Z|\psi\rangle = \alpha|0\rangle - \beta|1\rangle$. Both these errors can interact with each other and give rise to more complex errors in the system. A brief explanation of errors that can arise in the system is as follows:

*a) **Gate error**:* Gate error occurs when a quantum gate changes state of qubit(s) incorrectly. They are represented by fidelity, which denotes the probability of error in computation.

*b) **Decoherence error**:* Decoherence error occurs when a qubit interacts with the environment, resulting in the loss of its coherence and degradation of its quantum state.

*c) **Measurement error**:* Measurement error occurs when the classically measured output from a measurement operation is incorrect.

*d) **Crosstalk error**:* Crosstalk error occurs when a qubit interacts with a physically adjacent qubit, leading to an unwanted alteration to the qubit state.

C. Classical and Quantum Error Correction

*a) **Classical Error Correction**:* In classical computing, error correction is employed to maintain the integrity and precision of digital data by identifying and rectifying errors that may have arisen during transmission. This process utilizes Error Correction Codes (ECC) [22], [34]. The most commonly used ECC include Hamming codes [22], Bose-Chaudhuri-Hocquenghem (BCH) codes [35] and Reed-Solomon codes [36]. All of these codes detect and correct errors by adding redundant information to the original data. This allows the

reconstruction of the original message even if some parts of the data are corrupted or lost.

The $3-bit$ repetition code is the simplest example of a classical ECC, where the encoder expands the original binary information from a single bit to three bits i.e. $0 \rightarrow 000$ and $1 \rightarrow 111$. A $3-bit$ encoder can be formalized as a mapping from the original binary, Θ_b to the logical binary codeword, C_l. So, when a single-bit information $'0'$ is communicated the receiver will receive $'000'$.

$$\Theta_b = \{0,1\} \xrightarrow{\text{3-bit encoder}} C_l = \{000, 111\}$$

Error detection in a repetition code works by checking the bits at the receiving end. If they are not identical, the receiver knows that an error has occurred and resets the bits to their respective majority values. Therefore, in the case of a corrupted message, we can have three scenarios: ① *Single-bit flip error:* the receiver receives '010' instead of '000' considering that the second bit was flipped. In this case, the original codeword is generated using majority vote i.e. the corrupted message has '0' in two out of three bits, thus using majority vote the original message must be '000'. ② *Two-bit flip error:* the receiver receives '011' instead of '000' considering that the last two bits were flipped. The majority distance in this case will lead to the wrong result. ③ *Three-bit flip error:* the receiver receives '111' instead of '000' considering all the bits were flipped. In this case, the receiver will not even be able to detect the error.

The distance of a code is the smallest number of bits needed to transform one codeword to another. Formally, hamming distance [22] (or distance) between two codewords C_i and C_j is defined as $\delta(C_i, C_j) = 2t+1$, where $C_i, C_j \in C_l$ and t is the maximum number of errors the code can correct. The maximum number of errors that can be detected by a repetition code is $\delta - 1$ and the maximum number of errors that can be corrected is $\lfloor(\delta-1)/2\rfloor$, since the majority vote reset scheme will not work beyond this point. Therefore, for a $3-bit$ repetition code, $\delta = 3$. Traditionally, an ECC is described using the notation: $[n, k, \delta]$ where n is the number of bits in a codeword, k is the number of encoded bits or the original bitstring length, and δ is the code distance. For a $3-bit$ repetition code, the number of bits in the codeword is $n = 3$, the number of encoded bits is $k = 1$, and the distance is $\delta = 3$ as it requires a maximum of 3-bit flips to transform $'000'$ to $'111'$ and vice-versa. Therefore, the $3-bit$ repetition code is labeled as the $[3, 1, 3]$. In general, a classical $n-bit$ repetition code is labeled as $[n, 1, n]$.

*b) **Footsteps to Quantum Error Correction**:* There exist several reasons why the direct translation of classical Error Correction Codes (ECC) into the quantum domain is non-trivial. Firstly, quantum states cannot be duplicated similarly to classical information due to the no-cloning theorem [23]. Secondly, qubits are vulnerable to bit-flip errors as well as phase-flip errors as mentioned in subsection II-B, unlike the classical domain where bit-flip errors are the only kind of errors. Therefore, a QEC code (QECC) should be able to both

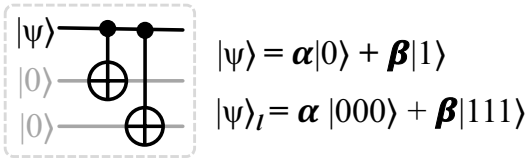

Fig. 2. Illustration of the quantum circuit utilized to create $|\psi\rangle_l$ from the original state $|\psi\rangle$ in a $3-qubit$ repetition code.

detect and correct phase-flip errors along with bit-flip errors. Finally, every time a qubit is measured, the wavefunction collapses [24] and the qubit loses its original state. Therefore, measuring a qubit directly is also not an option. An optimal QECC should possess the capability to identify and rectify both bit-flip and phase-flip errors while circumventing the direct duplication of the initial quantum state or the direct measurement of the qubits.

Classical $3-bit$ repetition code works by encoding a single bit into three bits i.e. $0 \rightarrow 000$ and $1 \rightarrow 111$. To explain QEC, we discuss the $3-qubit$ quantum repetition code first, which serves as the quantum counterpart to the classical $3-bit$ repetition code. The general idea of an $n-qubit$ repetition code is that the original state $|\psi\rangle$ is encoded with n qubits to form a logical state, $|\psi\rangle_l$ [37]. This distributes the quantum information, $|\psi\rangle$, across the encoded logical state, $|\psi\rangle_l$. Formally a $3-qubit$ quantum code is represented as,

$$\begin{aligned}
|\psi\rangle &= \alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{3-qubit encoder}} |\psi\rangle_l \\
|\psi\rangle_l &= \alpha|0\rangle_l + \beta|1\rangle_l \\
&= \alpha|0\rangle \otimes |0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle \otimes |1\rangle \\
&= \alpha|000\rangle + \beta|111\rangle \\
&\textit{where: } |0\rangle_l = |000\rangle\,; |1\rangle_l = |111\rangle
\end{aligned}$$

Due to the prohibition imposed by the no-cloning theorem [23], the encoding of the state $|\psi\rangle$ is done by applying CNOT gates to prepare the logical state, $|\psi\rangle_l$. Fig. 2 shows the quantum circuit that is used to expand the original state $|\psi\rangle$ into its logical state, $|\psi\rangle_l$.

Similar to classical ECC, QEC codes are implemented in a three-step process: error detection, error deduction, and error correction. Therefore, every QEC circuit must contain all of these three components. After preparing the logical state $|\psi\rangle_l$, we can move on to error detection, which in most QEC schemes, is done using the stabilizer codes. In simple terms, stabilizer codes check the parity of two or more qubits using CNOT gates. The output of the stabilizers is called the syndrome bits which is $+1$ (-1) for an error-free (erroneous) case. Error detection is done based on the syndrome bits as it provides the location of the error. Once we know the error location we can simply correct the error. Since the Pauli gates are self-inverse [38] applying a Pauli-gate twice returns the original state. Therefore, error correction, once we know where the error has occurred, is easy - we simply have to re-apply the gate on the affected qubit. Let us visualize this with an example: let there be a single bit-flip error, e on the encoded state, $|\psi\rangle_l$, such that $e = X \otimes I \otimes I$, and let there be a

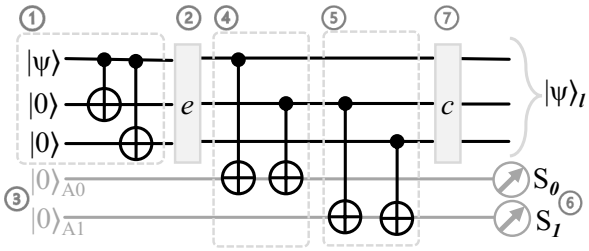

Fig. 3. Demonstrating the quantum circuit that implements a $3-qubit$ repetition code. The circuit comprises the following components: ①: The state preparation circuit represented in Fig. 2. ②: A single bit-flip error that may occur on any of the 3 qubits. ③: Two ancilla qubits that are initialized to the state, $|0\rangle$ are employed for parity checking purposes. ④: The first stabilizer circuit, $Z\otimes Z\otimes I$, is responsible for measuring the parity between the first two qubits. ⑤: The second stabilizer circuit, $I\otimes Z\otimes Z$, measures the parity between the last two qubits. ⑥: The ancilla qubits are utilized to obtain the syndrome bits, S_0, S_1, which enable the detection and deduction of errors. ⑦: To rectify the errors, a correction operator, c, comprising of a sequence of self-inverse Pauli-gates, is applied to the qubit that requires correction.

TABLE I
EIGENVALUES OF VARIOUS ERRORS ON A 3-QUBIT SYSTEM WITH RESPECT TO PARITY MEASUREMENT OPERATORS [39].

Error	$Z\otimes Z\otimes I$	$Z\otimes I\otimes Z$	$I\otimes Z\otimes Z$
$I\otimes I\otimes I$	$+1$	$+1$	$+1$
$X\otimes I\otimes I$	-1	-1	$+1$
$I\otimes X\otimes I$	-1	$+1$	-1
$I\otimes I\otimes X$	$+1$	-1	-1

correction operator, c, such that $ce\,|\psi\rangle_l = |\psi\rangle_l$. Given that the Pauli-gates are self-inverse, this is satisfied when $c = e$. The following equations demonstrate the manner in which the correction operator transforms the erroneous state back to its accurate state.

$$|\psi\rangle_l \xrightarrow{error} e\,|\psi\rangle_l = (X\otimes I\otimes I)\,|\psi\rangle_l = \alpha\,|100\rangle + \beta\,|011\rangle$$
$$e\,|\psi\rangle_l \xrightarrow{correction} ce\,|\psi\rangle_l = (X\otimes I\otimes I)(X\otimes I\otimes I)\,|\psi\rangle_l$$
$$= \alpha\,|000\rangle + \beta\,|111\rangle = |\psi\rangle_l$$

Say, we want to make parity measurements on the encoded state, $|\psi\rangle_l = \alpha\,|000\rangle + \beta\,|111\rangle$, there are three possible parity measurement (or syndrome) operators : $(Z\otimes Z\otimes I)$, $(Z\otimes I\otimes Z)$ and $(I\otimes Z\otimes Z)$. All three of these will result in $+1$. Now suppose, we have a bit flip error on the second qubit, so $e\,|\psi\rangle_l = \alpha\,|010\rangle + \beta\,|101\rangle$, two out of three parity checkers measurement will return -1. Different errors on qubits respond to different combinations of pairwise parity measurement. Table I [39] shows the response of various errors on a 3-qubit system with all possible parity measurement operators. Similar to the context of multiplicative system, two operators o_i, o_j can be said to be commuting if $[o_i, o_j] = 0$, i.e. $(o_i\otimes o_j) = (o_j\otimes o_i)$ and they are anti-commuting if $[o_i, o_j] \neq 0$, i.e. $(o_i\otimes o_j) = -(o_j\otimes o_i)$. Therefore, if the outcome of Table I is $+1$, the error commutes with the parity operator and if the outcome is -1, the error anti-commutes with the parity operator. We will later use this concept in stabilizer formalism.

Fig. 3 shows the entire circuit of a $3-qubit$ repetition code which is divided into seven parts. ①: First, state preparation is done which produces the logical state $|\psi\rangle_l = |000\rangle$ from the original state, $|\psi\rangle = |0\rangle$. ②: A single qubit bit-flip error e occurs on one of the three qubits in the logical state, $|\psi\rangle_l$ and produces an erroneous state, $e\,|\psi\rangle_l$. ③: Two ancilla qubits A_0, A_1 are each initialized to state $|0\rangle$. ④: The first stabilizer measures $Z\otimes Z\otimes I$, which is equivalent to the parity between the first two qubits by using CNOT gates among the qubits and the first ancilla qubit. ⑤: The second stabilizer measures $I\otimes Z\otimes Z$, which is equivalent to the parity between the last two qubits by using CNOT gates among the qubits and the second ancilla qubit. ⑥: The syndrome bits S_0, S_1 measure the Pauli-Z expectation values of the ancilla qubits A_0, A_1 respectively. The value of S_i is either $+1$ or -1 depending on whether or not any error has occurred. Table II shows the value of the syndrome bits with respect to the erroneous state, $e\,|\psi\rangle_l$. ⑦: Once we know where and which error has occurred, we simply apply the same gate on the affected qubit as the Pauli-gates are self-inverse. Table II also shows how the Pauli-gates can be applied to correct the errors in the state $e\,|\psi\rangle_l$, for example, if the erroneous state is $|100\rangle$, then the correction operator is XII. Since the stabilizers are used to measure parity and the actual measurements are done on the ancilla qubits, the logical state $|\psi\rangle_l$ remains unharmed. Consequently, we have obtained a code capable of detecting and rectifying X-flip errors without adversely affecting the state of the system.

TABLE II
DETECTION, DEDUCTION, AND CORRECTION OF ERRORS WITH RESPECT TO THE SYNDROME MEASUREMENTS.

Detection		**Deduction**		**Correction**
S_0	S_1	Error Location	Erroneous State	Correction Operator
$+1$	$+1$	No error	$\|000\rangle$	III
-1	$+1$	Qubit 1	$\|100\rangle$	XII
-1	-1	Qubit 2	$\|010\rangle$	IXI
$+1$	-1	Qubit 3	$\|001\rangle$	IIX

In a manner analogous to single-qubit operators, such as X, Z, and others, which act on the state $|\psi\rangle$, there exist logical operators that act on the logically encoded state $|\psi\rangle_l$. These operators, known as the $logical-X$ operator $(\bar{X})$ and $logical-Z$ operator $(\bar{Z})$, execute bit-flip or phase-flip operations on the entire encoded state, rather than solely on individual physical qubits. By doing so, they preserve the error-correcting properties of the encoded system while facilitating the execution of logical operations as needed. The distance of a quantum code is the size of the logical Pauli operator that can transform one codeword into another. Therefore, by intuition the logical Pauli-X operator should be $\bar{X} = X\otimes X\otimes X$ i.e. $\bar{X}\,|000\rangle = |111\rangle\,; \bar{X}\,|111\rangle = |000\rangle$. If the quantum circuits have only been susceptible to Pauli-X errors, the distance of the 3-qubit repetition code would be 3. However, we also have to consider the logical Pauli-Z operator. Following the footsteps of the logical Pauli-X operator, we can write the logical Pauli-Z operator as $\bar{Z} = Z\otimes Z\otimes Z$ i.e. $\bar{Z}\,|000\rangle = |000\rangle\,; \bar{Z}\,|111\rangle = -\,|111\rangle$. However, the same can be achieved only by using one Z-operator i.e. by using $\bar{Z} = Z\otimes I\otimes I$. The proof below shows that the distance of a

3-qubit repletion code is 1, owing to the fact that the minimum number of operators required to transform one codeword to another is 1. Note that the two codewords of a $3-qubit$ QECC are: $\alpha\left|000\right\rangle \pm \beta\left|111\right\rangle$.

$$\begin{aligned}\bar{Z}\left|\psi\right\rangle_l &= (Z\otimes I\otimes I)(\alpha\left|000\right\rangle+\beta\left|111\right\rangle)\\ &= \alpha(\left|0\right\rangle\left|0\right\rangle\left|0\right\rangle)+\beta(-\left|1\right\rangle\left|1\right\rangle\left|1\right\rangle)\\ &= \alpha\left|000\right\rangle-\beta\left|111\right\rangle\end{aligned}$$

A QECC is labeled as $[[n,k,\delta]]$, where n is the total number of qubits, k is the original number of qubits and δ is the quantum code distance. Therefore, a $3-qubit$ repetition code is labelled as $[[3,1,1]]$. The 3-qubit repetition code does not constitute a comprehensive QEC solution, as it neither detects phase flip errors nor bit-flip errors occurring on multiple qubits. This code primarily serves as a means to emphasize the essential components necessary for constructing robust and comprehensive error correction codes.

III. Fundamentals of QEC

Most QEC circuits work in 5 steps: a state preparation, a stabilizer circuit, error detection, an error decoder, and finally error correction. So far, we have seen the basics of all of these five steps using the $3-qubit$ bit-flip error correcting code shown in Fig. 3. ① in the figure shows the state preparation where a given basis state is encoded with two arbitrary qubits. ④ and ⑤ show the sets of stabilizers that can be used in a $3-qubit$ ECC. Based on the output of the stabilizer circuits we perform: error detection, error deduction, and error correction. An error is detected if either of the syndromes measures -1. Based on the values of the syndrome, the error is deducted i.e. where the error has occurred and based on this deduction the correction operator, c is applied. This completes a full ECC. In this section, we concentrate on the fundamental building blocks required for the development of a foolproof QECC.

A. Stabilizer Formalism

Most QECC use stabilizers [37] to perform error detection. Therefore, a generalized idea of forming stabilizers irrespective of the code should be useful. A stabilizer is defined to be a set of operators that leave a quantum state unchanged i.e. $o\left|\psi\right\rangle = \psi$ where o is a set of stabilizer operators and $\left|\psi\right\rangle$ is a quantum state. Stabilizers are an important part of ECC primarily because they detect the error without harming or changing the original quantum state.

Stabilizer formalism is an efficient way to manipulate and describe quantum states when it comes to the context of QECCs and fault-tolerant quantum computing. It is a mathematical framework that is generated by tensor products of the Pauli matrices (shown below):

$$\mathbf{I} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}; \mathbf{X} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}; \mathbf{Y} = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}; \mathbf{Z} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

The Pauli group is a group of Pauli matrices that act on n qubits and is generated by the tensor products of the above-mentioned matrices. It has $2*4^n$ elements, which include a global phase factor of ± 1 or $\pm i$. For example, the Pauli group, P that acts on a single qubit will contain the following elements: $P = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$.

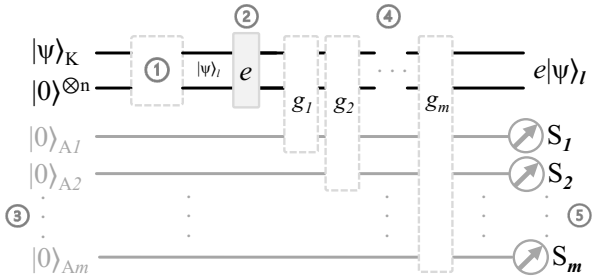

Fig. 4. Illustration of a generalized stabilizer circuit that encodes k physical qubits with n logical qubits and employs m stabilizers to detect potential errors in the logical codeword, $\left|\psi\right\rangle_l$, where $m = n-k$. The circuit comprises the following components: ①: The encoder circuit that encodes k physical qubits with n logical qubits. ②: A bit-flip or phase-flip or both errors that may occur on one or more qubits within the logical codeword, $\left|\psi\right\rangle_l$. ③: m ancilla qubits that are initialized to $\left|0\right\rangle$ to facilitate the projection of the measurements of the stabilizer generators. ④: m stabilizer generators, designated as $g_1, g_2, \ldots, g_m$, are employed. ⑤: Each stabilizer g_i projects its measurements onto the ancilla qubit $\left|0\right\rangle_{Ai}$. The ancilla qubits are subsequently measured as syndrome measurements, S_i. Based on these syndrome values, error deduction and correction mechanisms are executed. It is essential to highlight that the provided diagram primarily illustrates a generalized stabilizer circuit designed for detecting errors. This representation does not encompass the execution of correction operators; consequently, the final basis state persists as the erroneous state, $e\left|\psi\right\rangle_l$.

A stabilizer generator (or stabilizer) that acts on n qubits is a product of a maximal set of n commuting elements of the Pauli group, P with an eigenvalue of $+1$. These n operators are denoted as $g_1, g_2, \ldots, g_n$. Therefore, an element, g, in an abelian group [40] of self-inverse Pauli operators should satisfy the following relation: $g\left|\psi\right\rangle = \Pi g_i\left|\psi\right\rangle = \left|\psi\right\rangle; i = 1, 2, \ldots, n$. Stabilizer generators must obey the following relations:

- Each generator is an element of the Pauli group.
- They commute with each other i.e. $g_i \otimes g_j = g_j \otimes g_i; \forall i, j$
- All generators are independent of each other meaning, no product of a subset of the generators can be equal to another generator. In simpler terms, we cannot create one by multiplying others.

With a basic mathematical background on Pauli operators and stabilizer generators, we can define a stabilizer code which is a quantum error-detecting code that is formed by a set of stabilizer generators. This code executes subsequent to the encoding of k qubits into n logical qubits. For each stabilizer code, there are encoded logical operators that depict operations on the encoded qubits. These logical operators are denoted as $\bar{X}_i$ and $\bar{Z}_i$ for $i = 1, 2, \ldots, k$. It is important to note that these logical operators do not change the stabilizer generators, they only have an effect on the encoded qubits and they follow the same rules as that of the standard X and Z operators.

Theoretically, every stabilizer is dividing the Hilbert space based on the eigenvalue (± 1). The size of the Hilbert space at the beginning is 2^n as we are using n physical qubits. If we apply m stabilizer generators, the Hilbert space is getting divided and finally, it boils down to the size of $\frac{2^n}{2^m}$. We already know that we are encoding n logical qubits with k physical qubits, therefore: $\frac{2^n}{2^m} = 2^k \Rightarrow m = n-k$ [39].

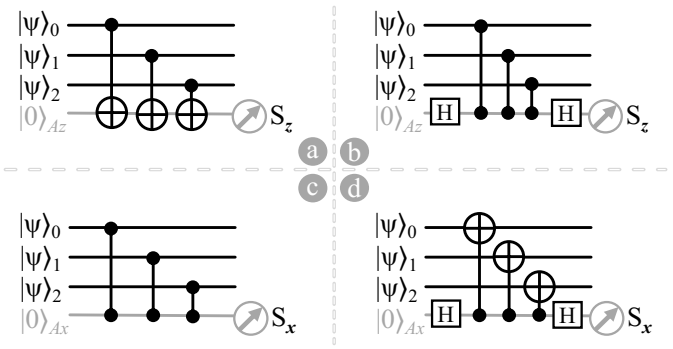

Fig. 5. Different approaches to generate a $Z-stabilizer$ and $X-stabilizer$ operating on 3 qubits, namely $|\psi\rangle_0$, $|\psi\rangle_1$ and $|\psi\rangle_2$. The ancilla qubit, on which the stabilizer output is being projected, is represented as $|0\rangle_{Az}$ and $|0\rangle_{Ax}$ for $Z-stabilizers$ and $X-stabilizers$, respectively. The circuits are presented as follows: ⓐ: This circuit exhibits a $Z-stabilizer$ similar to the one depicted in Fig. 3. The ancilla qubit, $|0\rangle_{Az}$ measures $Z|\psi\rangle_0 \otimes Z|\psi\rangle_1 \otimes Z|\psi\rangle_2$, commonly denoted as $Z_0 \otimes Z_1 \otimes Z_2$. ⓑ: This circuit demonstrates the same $Z-stabilizer$ as in (a). The stabilizer is regenerated using H gates and CZ gates instead of $CNOT$ gates. The ancilla qubit, $|0\rangle_{Az}$ also measures $Z_0 \otimes Z_1 \otimes Z_2$. ⓒ: This circuit represents an $X-stabilizer$. The ancilla qubit, $|0\rangle_{Ax}$ measures $X|\psi\rangle_0 \otimes X|\psi\rangle_1 \otimes X|\psi\rangle_2$, commonly denoted as $X_0 \otimes X_1 \otimes X_2$. ⓓ: This circuit portrays the same $X-stabilizer$ as mentioned in (c). The stabilizer is recreated using H gates and $CNOT$ gates instead of CZ gates. The ancilla qubit, $|0\rangle_{Ax}$ also measures $X_0 \otimes X_1 \otimes X_2$.

From the above equation it is important to note that if one increases the number of stabilizer generators, the number of logical qubits may decrease. Let us now see what a generalized stabilizer circuit looks like: there is basis state, $|\psi\rangle_K$ with k qubits i.e. $|\psi\rangle_K = |\psi_1\psi_2\ldots\psi_k\rangle$ that will be encoded with n physical qubits, $|0\rangle^{\otimes n} = |000\ldots\rangle$ and m stabilizers will be applied on the final logical codeword, $|\psi\rangle_l$. Fig. 4 shows a generalized stabilizer circuit that detects errors.

B. Structure of the Stabilizer Generator

For a stabilizer generator (or stabilizer) to be able to detect an error, the stabilizer has to anti-commute with the error (see Table I). Therefore, we need $Z-stabilizers$ to detect the X or $bit-flip$ errors and $X-stabilizers$ to detect Z or $phase-flip$ errors. A $Z-stabilizer$ will anti-commute with a X or $bit-flip$ error and produce an eigenvalue of -1. In Fig. 3: ④ and ⑤ depict two $Z-stabilizers$ which can detect X errors only. Let's take a quantum system with three qubits $|\psi\rangle_0$, $|\psi\rangle_1$ and $|\psi\rangle_2$, with a $Z-stabilizer$ and an $X-stabilizer$ acting on these three qubits. Finally, these stabilizers will project their values onto the ancilla qubits $|0\rangle_{Az}$ and $|0\rangle_{Ax}$, respectively. In Fig. 5 we note that $|0\rangle_{Az}$ measures $Z_0 \otimes Z_1 \otimes Z_2$ which will anti-commute with $X-errors$ to produce an eigenvalue of -1. Similarly, $|0\rangle_{Ax}$ measures $X_0 \otimes X_1 \otimes X_2$ which will anti-commute with $Z-errors$ to produce an eigenvalue of -1. Fig. 5 also shows different ways of creating the two different types of stabilizers [41]. A generalized circuit of a stabilizer generator is shown in Fig. 6 where P is a Pauli-gate operation and the circuit measures $P|\psi\rangle_0 \otimes P|\psi\rangle_1 \otimes P|\psi\rangle_2 \equiv P_0 \otimes P_1 \otimes P_2$. An ancilla qubit, initialized to $|0\rangle$ is a control for an arbitrary state $|\psi\rangle$, using a unitary operator, P i.e., operator P is applied when the control

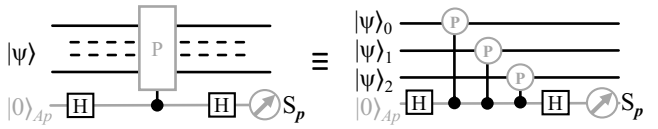

Fig. 6. Illustration of a universal circuit of a stabilizer generator, in which P represents a Pauli-gate operation. This circuit measures $P|\psi\rangle_0 \otimes P|\psi\rangle_1 \otimes P|\psi\rangle_2 \equiv P_0 \otimes P_1 \otimes P_2$. Usually for a stabilizer generator, $P \in \{X, Z\}$.

is $|1\rangle$ and nothing happens when the control is $|0\rangle$. We explain the state of the circuit (Fig. 6: *left*), one step at a time:

$$\textit{First H gate} \Rightarrow \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)|\psi\rangle$$

$$\textit{Controlled P} \Rightarrow \frac{1}{\sqrt{2}}\big(|0\rangle|\psi\rangle + |1\rangle P|\psi\rangle\big)$$

$$\textit{Second H gate} \Rightarrow \frac{1}{2}\Big(\big(|0\rangle + |1\rangle\big)|\psi\rangle + \big(|0\rangle - |1\rangle\big)P|\psi\rangle\Big)$$

$$= |0\rangle\left(\frac{1+P}{2}\right)|\psi\rangle + |1\rangle\left(\frac{1-P}{2}\right)|\psi\rangle$$

$$\textit{Finally, } |0\rangle_{Ap} = |0\rangle\lambda_+^P|\psi\rangle + |1\rangle\lambda_-^P|\psi\rangle$$

$$\textit{where the eigenstate projectors are } \Rightarrow \lambda_+^P; \lambda_-^P$$

The primary importance of the above equations is to show that the stabilized state, $|\psi\rangle$ will always hold the form $\alpha'|0\rangle + \beta'|1\rangle$. Consequently, even if an error occurs at an arbitrary rotational angle, the stabilizer circuit will enforce a non-superposed state for the error. As a result, when the error detection circuit is applied, the stabilizers will consistently return an eigenstate of either ± 1 [31].

C. Topological Codes

Repetition codes exhibit limitations as QECCs because of their restricted error correction capacities and diminished error thresholds. Moreover, they do not possess fault tolerance and encounter scalability issues. These codes are also unsuitable for rectifying errors that transpire on multiple qubits or are complex. Therefore, a class of QEC codes, known as the Topological Codes are proposed that use the properties of topological order [42] to protect the quantum information from errors. Topological codes [15] such as, toric codes are designed to process and store quantum information by exploiting the global features of lattice-like structures, which make them naturally resistant to local errors. Topological codes have gained significant popularity due to their fault-tolerant properties and ability to correct quantum errors with relatively low overhead. Some key features of topological codes include:

- *Spatial separation:* All logical qubits are encoded using non-local degrees of freedom, meaning that the probability of local errors corrupting the information is really low. Simply put, the logical qubits are spread out (i.e., not close to each other) making it harder for local errors to affect the information encoded.
- *Anyonic excitation:* So far topological codes are built using anyons, which have a unique braiding property to manipulate and store quantum information
- *Error correction:* Stabilizer checks [19], [43]–[45] are an important part of error correcting codes but stabilizers in

topological codes are non-local operators and they can detect the anyonic excitation created by errors.
- *Fault tolerant quantum computation:* Topological codes support fault-tolerant quantum computing with a low error rate even in the presence of gate errors.

There exist multiple types of Topological codes, which can include, but are not limited to, the following:

- *Toric codes:* These are one of the earliest known topological codes and are defined in a two-dimensional lattice with periodic boundary conditions. The robustness of detecting and correcting errors is increased by the structure of the lattice [46].
- *Surface codes:* These are also one of the earliest known topological codes and are defined in a two-dimensional lattice, but unlike the toric codes, without periodic boundary conditions. They are currently the most promising candidate for a large-scale, fault-tolerant quantum computer due to their high error threshold and low overhead [16]. Realization of a surface code has been the primary goal for multiple research articles [47]–[51].
- *Color codes:* These are another type of topological codes that can be either on a two or three-dimensional lattice. They possess similar error-correcting properties as surface codes but offer additional advantages like the ability to apply certain logical gates transversely [18], [52].

The overarching structure of a topological code entails the construction of the complete code by assembling repetitive elements. In this subsection, we primarily discuss the functioning of topological codes, with particular emphasis on toric and surface codes. The operation of color codes and other topological codes is beyond the scope of this paper.

The toric code is defined in terms of a square lattice with periodic boundary conditions, meaning if we have an L^*L square lattice wrapped around a torus, then the right-most edge is equivalent to the left-most edge and the top-most edge is equivalent to the lower-most edge. For surface codes such boundary conditions do not exist and hence it is often referred to as the planar code [53], [54]. Fig. 7 depicts a torus and how it is used to model the boundary conditions of a toric code lattice. It is important to keep in mind that the lattice for a surface code would look exactly like the toric code lattice but without the boundary conditions. A comparison between toric codes and surface codes is shown in Table III.

Every edge of a lattice corresponds to one qubit. Therefore, in a L^*L lattice there are $2L^2$ edges, i.e., $2L^2$ physical qubits. Every plaquette is a $Z-stabilizer$ generator and every vertex is a $X-stabilizer$ generator. Fig. 8 showcases the following aspects: ⓐ demonstrates that there exists a $Z-stabilizer$ that operates on the four physical qubits located on the four edges of a plaquette, while ⓑ depicts an $X-stabilizer$ that operates on the four physical qubits located on the four edges of a vertex. This implies that each physical qubit would have one $Z-stabilizer$ and one $X-stabilizer$ operating on it. Hence, it is essential to note that plaquettes and vertices overlap with each other, and as a result, these two operators commute. Upon

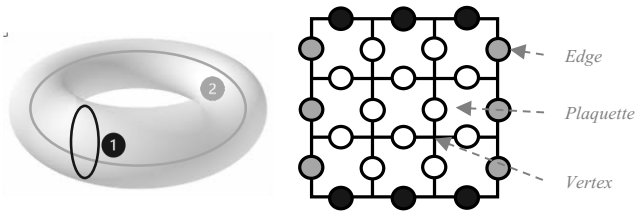

Fig. 7. ***Left:*** A torus that is used to model a toric code. ***Right:*** A toric code in the form of a lattice. The lattice comprises of two loops that run through the torus. Various components of the lattice, such as the edge, plaquette, and vertex, are designated in the figure. ① represents the vertical loop that corresponds to the top-most and the lower-most edge of the lattice, whereas ② represents the horizontal loop that corresponds to the left-most and the right-most edge of the lattice. It is noteworthy that a surface code would possess a lattice that appears exactly like the one depicted here but would not have any boundary conditions. This means that a surface code does not model on a torus and is therefore referred to as a planar code.

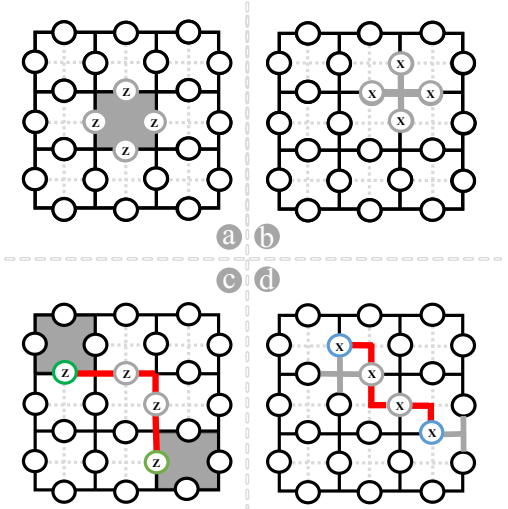

Fig. 8. Depiction of Z and $X-stabilizers$ acting on qubits in a toric code, and showcasing of the erroneous qubits that trigger plaquettes and vertices at the end of the error strings. ⓐ: A plaquette that represents $Z-stabilizers$ acting on qubits. ⓑ: A vertex that represents $X-stabilizers$ acting on qubits. ⓒ: The red line exhibits the error string that triggers the marked $Z-operators$, However, only the $plaquette-operators$ that are marked in grey at the end of the error string will produce an eigenstate -1. The green color denotes the erroneous qubits that activate the grey plaquettes at the end of the error string. ⓓ: The red line illustrates the error string that activates the marked $X-operators$. Nevertheless, only the $vertex-operators$ that are marked in grey at the end of the error string will generate an eigenstate of -1. The blue color is used to identify the erroneous qubits that initiate the grey vertices at the end of the error string.

multiplying a set of plaquette operators, the resultant operator would be equal to the $Z-operators$ that operate on the boundary of the consolidated plaquettes. The same principle applies to vertices.

In the toric code, the lattice has periodic boundary conditions, meaning that it is modeled around a torus. Such a structure leads to two independent loops: one in the horizontal direction and the other in the vertical direction, as shown in Fig. 7, and this gives rise to two logical qubits. Mathematically, the toric code will have four logical operators: two $X-operators$ on the horizontal and vertical loops $(\bar{X}1, \bar{X}2)$ and two $Z-operators$ on the horizontal and vertical loops $(\bar{Z}1, \bar{Z}2)$. These operators must commute with the stabilizer generator but never with each other. Given that there are two pairs of non-commuting logical operators $(\bar{X}1, \bar{X}2$ and

TABLE III
A COMPARATIVE STUDY BETWEEN TORIC CODES AND SURFACE CODES

Property	Toric Code	Surface Code
Dimensionality	2D in a L^*L lattice	2D in a L^*L lattice
Lattice Structure	Regular lattice with periodic boundary conditions	Regular lattice without periodic boundary conditions
Logical Qubits	2 (two independent logical qubits)	1 (one logical qubit)
Error Correction	Detects and corrects any single-qubit error	Detects and corrects any single-qubit error
Stabilizer Generators	Two types: vertex and plaquette operators	Two types: vertex and plaquette operators
Boundary Conditions	Periodic (closed topology)	Open (open topology)
Logical Gates	Braiding anyons	Lattice surgery or code deformation
Implementation Complexity	More complex due to periodic boundary conditions	Simpler due to open boundary conditions
$[[n, k, \delta]]$	$[[2L^2, 2, L]]$	$[[2L^2 + 2L, 1, L]]$

$\bar{Z}1, \bar{Z}2$), the toric code has 2 logically encoded qubits. On the contrary, the surface code lattice has open boundary conditions, meaning there are no loops. In this case, there are two logical operators: one $X - operator$ line that goes in one direction ($\bar{X}$) and another $Z - operator$ line that goes in the direction orthogonal ($\bar{Z}$) to the previously mentioned line. These operations must commute with the stabilizers but never with each other. Given there is only one pair of non-commuting logical operators ($\bar{X}1$ and $\bar{Z}1$), the surface code has only 1 logically encoded qubit.

Every stabilizer measurement is projected onto an ancilla qubit and the syndrome measurement tells us where the error has occurred on the lattice. A stabilizer generator circuit can be found in Fig. 5. If in a plaquette there are four qubits, then their respective $Z - stabilizer$ or $plaquette - operator$ measurement will be $Z_1 \otimes Z_2 \otimes Z_3 \otimes Z_4$ and similarly, the $X - stabilizer$ or $vertex - operator$ will measure $X_1X_2X_3X_4$. If there is an error on any one of the qubits, depending on the type of error, either the $X - stabilizer$ or the $Z - stabilizer$ linked with that qubit will return an eigenvalue of -1. This is the case when only one qubit has an error.

Now suppose, there is a string of errors on the lattice, represented by a product of X operators along the string. Mathematically, when we measure the plaquette operators, we are taking the tensor product of the $Z - operators$ on the edges forming the plaquette. If the plaquette is not at the end of the string, the product of $Z - operators$ for the plaquette will cancel out and will always result in a $+1$ eigenstate. This is due to the fact that $Z - operators$ square to the identity i.e. $ZZ = I$ and the tensor product of an even number of $Z - operators$ will result in identity. However, if the plaquette is at the end of the error string, the product of $Z - operators$ will not cancel out thus, resulting in a -1 eigenstate. This is because there is an odd number of $Z - operators$ at the end plaquette, making the tensor products of $Z - operators$ unequal to the identity. In simple terms, when there is a string of errors on the lattice, the only plaquette or vertex with -1 eigenstate will be at the end of the error string. Fig. 8: ⓒ displays that $plaquette - operators$ situated at the end of an error chain produce an eigenvalue of -1 and similarly, ⓓ demonstrates that $vertex - operators$ located at the end of an error chain solely produce an eigenvalue of -1.

Given that solely the *'end of the error string'* generates a signal that an error has transpired, it is imperative to address the concern of the existence of multiple paths between the two endpoints. Therefore, a fundamental query arises, which is how to identify the precise location of the error string between two given endpoints. *How do we know which is the correct path?* To solve this problem large-scale QECCs use approximate inference algorithms to determine the most likely error that might have occurred given a specific syndrome value. These methods allow for application in real-time between successive stabilizer code cycles. To date, there is no particular algorithm that does this job efficiently for all ECC. For toric and surface codes an algorithm known as the Minimum Weight Perfect Matching (MWPM) is often used for decoding [55], [56]. The effectiveness of a QECC's error rate is highly dependent on the type of decoder utilized, as some approximate algorithms perform better than others. To combat this problem, error-correcting codes are simulated over multiple cycles and the syndrome is sampled to better understand a noise model. Surface codes are also simulated over multiple cycles in the hope of eventually creating fault-tolerant quantum computers [50], [57], [58].

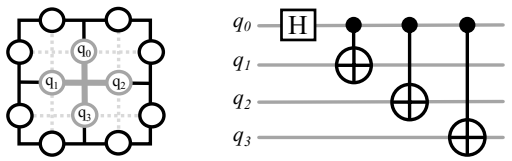

Fig. 9. ***Left:*** A $vertex - operator$ acting on four qubits, q_0, q_1, q_2, q_3 on a lattice. ***Right:*** A state preparation encoding circuit operates on qubits q_0, q_1, q_2, q_3, within the vertex-operator, all initialized to $|0\rangle$. When applied to $|0000\rangle$, the circuit yields $\frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$.

D. A General Encoding Circuit

In quantum error correction, before employing stabilizer circuits for error detection, an encoding circuit is needed to map logical qubits to physical qubits, thereby protecting quantum information. The encoding process, often termed state preparation, is represented by a unitary transformation U that encodes k qubits with n logical qubits. Two methods exist for state preparation: stabilizer generator circuits and unitary circuits [59], [60]. This paper focuses on the former. Encoding of logical states has been experimentally demonstrated in small-scale error correction protocols using various codes [61]–[68].

It is well-established that a logical codeword commutes with a stabilizer generator, meaning they have $+1$ eigenstates for each stabilizer generator in a group of stabilizers. To prepare the logical state from an arbitrary input state, it is essential to

project qubits into the $+1$ eigenstate of the relevant stabilizer operators. When initializing the quantum circuit with n qubits, all set to $|0\rangle$, we obtain a stabilized state at $|0\rangle^{\otimes n}$. This condition is valid when applying $Z-stabilizers$; however, when applying $X-stabilizers$, nearly half of the resultant eigenstate will be $+1$, and the rest will be -1 due to superposition. The primary goal of the state preparation circuit is to ensure that the $X-stabilizers$ consistently return a $+1$ eigenstate in the absence of errors [16]. Fig. 9 illustrates an $X-stabilizer$ or $vertex-stabilizer$ acting on four qubits (q_0, q_1, q_2, q_3) and the corresponding encoding circuit. This circuit is inspired by the Greenberger–Horne–Zeilinger (GHZ) state [69]. Assuming the circuit depicted in Fig. 9 can be represented by U, a product of operators, and all qubits (q_0, q_1, q_2, q_3) initialized to $|0\rangle$, then $U|0000\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$. The decomposition is valid only when the initial Hadamard qubit is in the $|0\rangle$ state; thus, the choice of Hadamard's qubit must be made cautiously. The superposition generated by the state preparation circuit counteracts the superposition induced by the $X-stabilizer$. Consequently, in the absence of errors, the X-stabilizer eigenstate yields a $+1$ result.

IV. Practicality of QECC

In this section, we show a comparative analysis of existing QECCs in literature (Table IV). We also examine diverse qubit types and the corresponding suitable QECCs for them.

A. Superconducting Qubits

a) ***Technological Details****:* Superconducting qubits are LC oscillator circuits maintained at cryogenic temperatures. Typically, the inductor is implemented as a Josephson Junction (for introducing non-linearity in the circuit) using superconducting material such as niobium or aluminum and the capacitor is implemented as either an inter-digitated capacitor or parallel plate capacitor. Additionally, the Josephson Junction possesses its own intrinsic capacitance. The overall LC circuit forms a harmonic oscillator that creates different energy levels out of which the lowest two energy levels are selected as the basis states of the qubit [70].

b) ***Feasibility****:* Surface codes are implemented in [71] on seven qubits (four data qubits, three ancilla qubits). It has been observed that repeated error correction results in longer coherence times of qubits than no error correction at a high 96.1% average local fidelity. [72] implements a distance three surface code on 17 qubits to achieve fidelity up to 0.9 after error correction. A [[5,1,3]] error correcting code is produced in [73] using 92 gates overall that corrects single-qubit gate errors at $\sim$75% fidelity. There are older works like [62] that incorporate a nine-qubit code to rectify bit-flip errors, [74] that propose a quantum error detection-only circuit on N-qubit systems and two-qubit error correction schemes.

B. Trapped-ion Qubits

a) ***Technological Details****:* Trapped-ion quantum computers use ions as qubits, that are trapped in electromagnetic traps such as the Penning trap that provides confinement in up to two directions or the more widely used RF Paul trap that confines in two or three directions. The ions selected, are usually of alkaline earth metals, such as Be^+, Mg^+, and Ca^+, or of those used in atomic clocks such as Al^+, In^+, Lu^+. These trapped ions are maintained at low temperatures and in vacuum chambers for long coherence times [75].

b) ***Feasibility****:* Bacon-Shor logical qubit is implemented on 13 trapped ion qubits in [76] for error of up to 0.6% and $> 99\%$ fidelity after error correction. [[7,1,3]] Steanne code is built on 10 qubits in [77] that provided fidelity of up to 93%. Fault-tolerant parity readout has been incorporated in [78] with 93% parity measurement fidelity. A non-traditional work includes dissipative processing to incorporate a three-qubit code on trapped ions in [79]. Older works such as [68] correct single-qubit errors and improve the fidelity of computation by roughly 1%.

C. Photonic Qubits

a) ***Technological Details****:* A photon is used as a qubit, where the life of the qubit starts from the generation of the photon and ends at the detection. The photon is generated usually via processes like spontaneous parametric down-conversion (SPDC) or spontaneous four-wave mixing (SFWM). Cryogenic methods like superconducting nanowire single-photon detectors (SNSPD) are used to detect photons with up to $\geq 95\%$ detection efficiency. The state of photonic qubits is given by different degrees of freedom such as, polarization (vertical: $|0\rangle$, horizontal: $|1\rangle$), and spatial modes (such as Orbital Angular Momentum a.k.a OAM), while quantum gates are implemented using different optical devices such as, beam splitters, and phase shifters (more information can be found in [80]).

b) ***Feasibility****:* OAM-based photons are error corrected in [81] that provide up to 20% fidelity in a noisy channel as opposed to less than 1% fidelity in the uncorrected scenarios. Silicon-based photonic qubits were error corrected, leading to a 30% increase in overall qubit fidelity in [82]. Bosonic logical qubits have been proposed in [83] that improve fidelity up to 97%. Older works such as, [84] demonstrate error correction on a two-qubit system and achieve up to 98% fidelity and [85] implements photonic quantum memories to perform quantum error correction that provides around 95% fidelity.

V. Challenges and Future Direction

Quantum error correction encounters various hurdles that require resolution to facilitate a fault-tolerant, large-scale quantum computer. Addressing these challenges is critical to executing complex computations with a low error rate. Some of the prevalent challenges include:

- *Scalability:* To implement QEC, it is necessary to employ a significant number of qubits to encode a small number of logical qubits. Hence, scaling towards a large number of logical qubits is a substantial challenge.
- *Complexity of error decoding algorithms:* Sometimes the complexity of decoding algorithms can be remarkably high, making it challenging to perform real-time QEC.

TABLE IV
A COMPARATIVE ANALYSIS AMONG THE VARIOUS QUANTUM ERROR CORRECTION CODES AVAILABLE IN THE EXISTING LITERATURE

Code	Year	# Qubits	Description	Complexity	Decoding Algorithm	Advantage	Disadvantage
Shor's 9-qubit code [13]	1995	9	First quantum error-correcting code, correcting 1 arbitrary error. Simple example of an error-correcting code.	Moderate (9 physical qubits for 1 logical qubit)	Syndrome measurement, lookup table	Good for understanding basic error correction concepts.	Requires 9 qubits, not very resource-efficient.
Steane's 7-qubit code [14]	1996	7	Corrects single error, example of Calderbank-Shor-Steane code. Exploits classical error-correcting codes.	Moderate (7 physical qubits for 1 logical qubit)	Syndrome measurement, lookup table	More resource-efficient than Shor's code, easy to implement.	Only corrects single error, not suitable for larger systems.
Toric codes [15]	1997	Varies(2D lattice)	Topological codes defined on 2D lattice, robust against local errors. High threshold and fault-tolerant.	High (2D lattice)	Minimum weight perfect matching (MWPM)	High error threshold, robust against local errors.	Requires complex decoding algorithms.
Surface codes [16]	2002	Varies(2D lattice)	Topological codes defined on 2D lattice, most studied topological code. High threshold and fault-tolerant.	High (2D lattice)	Minimum weight perfect matching (MWPM)	High error threshold, well-studied, many efficient decoders.	Requires large qubit overhead.
Bacon-Shor codes [17]	2006	Varies	Subsystem codes with good error-correction properties. Separates error correction into subsystems.	Moderate to High (depends on specific code)	Syndrome measurement, lookup table	Simple structure, transversal gates for some operations.	Lower error threshold compared to topological codes.
3D Color codes [18]	2006	Varies(3D lattice)	3D generalization of surface codes, improved error-correction properties. Combines features of toric and color codes.	High (3D lattice)	Minimum weight perfect matching (MWPM)	Higher error threshold than 2D codes, transversal gates.	Requires 3D lattice structure, more complex to implement.
Homological Product codes [20]	2013	Varies	Product codes combining different quantum codes, allowing transversal gates. Exploits the structure of different codes	Varies (depends on specific codes used)	Syndrome measurement, classical error-correction algorithms	Enables transversal gates, versatile, can combine various codes.	Complexity and error threshold depend on the specific codes used.
Flag-qubit codes [21]	2020	Varies	This code is constructed on low degree graphs.	Varies (depends on specific parameters)	Classical maximum likelihood decoding	Reduces overhead and maintains high error threshold.	The decoding algorithm can be computationally demanding.

- *Resource overhead:* Error correction necessitates a substantial amount of physical overhead, including gates, qubits, and time. As the code scales towards a higher number of qubits, the overhead also intensifies. Therefore, optimizing the resource overhead is a challenging task.
- *Fault-tolerant gates:* The implementation of fault-tolerant quantum gates that can function reliably in the presence of errors is one of the major challenges. While some ECCs permit the use of transversal gates, they do not cover all the gates required for universal quantum computation. [86].
- *Noise modeling and error characterization:* Although the development and comprehension of precise noise models are paramount for ensuring the optimal performance of error-correcting codes, it remains a challenging task in quantum systems.

Future research will aim to enhance quantum error correction by developing efficient codes, improving decoding algorithms, and refining hardware for fault tolerance and performance. Investigations into hybrid codes [87], combining the best aspects of various error-correcting types, are underway. Additionally, machine learning is being utilized to optimize error correction code performance and scalability [88]. As quantum computing progresses, addressing these challenges is critical for achieving fault-tolerant computation.

VI. CONCLUSION

This paper presents a simplified perspective on quantum error correction aimed at researchers who may not possess comprehensive knowledge of quantum physics and mathematics. We cover classical error correction and its application in the quantum domain using a $3-qubit$ error correction code. The three essential components of error-correcting principles: detection, deduction, and correction are emphasized. We explore the stabilizer formalism to understand the necessary properties for the effective detection of quantum errors. Additionally, we discuss topological codes, including toric and surface codes, highlighting their distinct features within the shared lattice-based structure for error correction. The requirements and general mathematical framework of an encoding circuit are also addressed. Finally, we discuss current qubit technologies and their relevance to quantum error-correcting codes. This article aims to give readers a fundamental understanding of quantum error correction, its current state, and potential future advancements.

ACKNOWLEDGMENT

The work is supported in parts by the National Science Foundation (NSF) (CNS-1722557, CCF-1718474, OIA-2040667, DGE-1723687 and DGE-1821766).

REFERENCES

[1] F. Bova, A. Goldfarb, and R. G. Melko, "Commercial applications of quantum computing," *EPJ quantum technology*, vol. 8, no. 1, p. 2, 2021.

[2] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[3] Y. Li, M. Tian, G. Liu, C. Peng, and L. Jiao, "Quantum optimization and quantum learning: A survey," *Ieee Access*, vol. 8, pp. 23 568–23 593, 2020.

[4] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya *et al.*, "Quantum chemistry in the age of quantum computing," *Chemical reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.

[5] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated quantum computation of molecular energies," *Science*, vol. 309, no. 5741, pp. 1704–1707, 2005.

[6] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Demonstration of a small programmable quantum computer with atomic qubits," *Nature*, vol. 536, no. 7614, pp. 63–66, 2016.

[7] M. Brandl, M. Van Mourik, L. Postler, A. Nolf, K. Lakhmanskiy, R. Paiva, S. Möller, N. Daniilidis, H. Häffner, V. Kaushal *et al.*, "Cryogenic setup for trapped ion quantum computing," *Review of Scientific Instruments*, vol. 87, no. 11, p. 113103, 2016.

[8] X.-L. Wang, L.-K. Chen, W. Li, H.-L. Huang, C. Liu, C. Chen, Y.-H. Luo, Z.-E. Su, D. Wu, Z.-D. Li *et al.*, "Experimental ten-photon entanglement," *Physical review letters*, vol. 117, no. 21, p. 210502, 2016.

[9] X. Qiang, X. Zhou, J. Wang, C. M. Wilkes, T. Loke, S. O'Gara, L. Kling, G. D. Marshall, R. Santagati, T. C. Ralph *et al.*, "Large-scale silicon quantum photonics implementing arbitrary two-qubit processing," *Nature photonics*, vol. 12, no. 9, pp. 534–539, 2018.

[10] J. M. Chow, J. M. Gambetta, A. D. Corcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. R. Rozen *et al.*, "Universal quantum gate set approaching fault-tolerant thresholds with superconducting qubits," *Physical review letters*, vol. 109, no. 6, p. 060501, 2012.

[11] G. Wendin, "Quantum information processing with superconducting circuits: a review," *Reports on Progress in Physics*, vol. 80, no. 10, p. 106001, 2017.

[12] P. Recher and B. Trauzettel, "Quantum dots and spin qubits in graphene," *Nanotechnology*, vol. 21, no. 30, p. 302001, 2010.

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

[14] A. M. Steane, "Simple quantum error-correcting codes," *Physical Review A*, vol. 54, no. 6, p. 4741, 1996.

[15] A. Y. Kitaev, "Quantum computations: algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, p. 1191, 1997.

[16] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4452–4505, 2002.

[17] D. Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," *Physical Review A*, vol. 73, no. 1, p. 012340, 2006.

[18] H. Bombin and M. A. Martin-Delgado, "Topological quantum distillation," *Physical review letters*, vol. 97, no. 18, p. 180501, 2006.

[19] 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.

[20] J. Haah, "Local stabilizer codes in three dimensions without string logical operators," *Physical Review A*, vol. 83, no. 4, p. 042330, 2011.

[21] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, "Topological and subsystem codes on low-degree graphs with flag qubits," *Physical Review X*, vol. 10, no. 1, p. 011022, 2020.

[22] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[23] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, pp. 802–803, 1982.

[24] J. Von Neumann, *Mathematical foundations of quantum mechanics: New edition*. Princeton university press, 2018, vol. 53.

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

[26] S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum error correction for beginners," *Reports on Progress in Physics*, vol. 76, no. 7, p. 076001, 2013.

[27] R. Matsumoto and M. Hagiwara, "A survey of quantum error correction," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 104, no. 12, pp. 1654–1664, 2021.

[28] J. Preskill, "Reliable quantum computers," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 385–410, 1998.

[29] F. Gaitan, *Quantum error correction and fault tolerant quantum computing*. CRC Press, 2008.

[30] D. A. Lidar and T. A. Brun, *Quantum error correction*. Cambridge university press, 2013.

[31] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[32] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *nature*, vol. 464, no. 7285, pp. 45–53, 2010.

[33] A. Peres, "Reversible logic and quantum computers," *Physical review A*, vol. 32, no. 6, p. 3266, 1985.

[34] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[35] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

[36] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

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

[38] E. Witten, "Quantum field theory and the jones polynomial," *Communications in Mathematical Physics*, vol. 121, no. 3, pp. 351–399, 1989.

[39] D. Browne. Lectures on topological codes and quantum computation. [Online]. Available: https://sites.google.com/site/danbrowneucl/teaching/lectures-on-topological-codes-and-quantum-computation

[40] I. R. Shafarevich and M. Reid, *Basic algebraic geometry*. Springer, 1994, vol. 2.

[41] "Suppressing quantum errors by scaling a surface code logical qubit," *Nature*, vol. 614, no. 7949, pp. 676–681, 2023.

[42] X.-G. Wen, "Colloquium: Zoo of quantum-topological phases of matter," *Reviews of Modern Physics*, vol. 89, no. 4, p. 041004, 2017.

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

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

[45] A. A. Kovalev and L. P. Pryadko, "Quantum kronecker sum-product low-density parity-check codes with finite rate," *Physical Review A*, vol. 88, no. 1, p. 012311, 2013.

[46] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of physics*, vol. 303, no. 1, pp. 2–30, 2003.

[47] J. Kelly, R. Barends, A. Fowler, A. Megrant, E. Jeffrey, T. White, D. Sank, J. Mutus, B. Campbell, Y. Chen *et al.*, "Scalable in situ qubit calibration during repetitive error detection," *Physical Review A*, vol. 94, no. 3, p. 032321, 2016.

[48] E. A. Sete, W. J. Zeng, and C. T. Rigetti, "A functional architecture for scalable quantum computing," in *2016 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2016, pp. 1–6.

[49] J. O'Gorman, N. H. Nickerson, P. Ross, J. J. Morton, and S. C. Benjamin, "A silicon-based surface code quantum computer," *npj Quantum Information*, vol. 2, no. 1, pp. 1–14, 2016.

[50] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann *et al.*, "Realizing repeated quantum error correction in a distance-three surface code," *Nature*, vol. 605, no. 7911, pp. 669–674, 2022.

[51] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, "Experimental demonstration of fault-tolerant state preparation with superconducting qubits," *Physical review letters*, vol. 119, no. 18, p. 180501, 2017.

[52] H. Bombin and M.-A. Martin-Delgado, "Topological computation without braiding," *Physical review letters*, vol. 98, no. 16, p. 160502, 2007.

[53] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," *arXiv preprint quant-ph/9811052*, 1998.

[54] M. H. Freedman and D. A. Meyer, "Projective plane and planar quantum codes," *Foundations of Computational Mathematics*, vol. 1, pp. 325–332, 2001.

[55] J. Edmonds, "Paths, trees, and flowers," *Canadian Journal of mathematics*, vol. 17, pp. 449–467, 1965.

[56] V. Kolmogorov, "Blossom v: a new implementation of a minimum cost perfect matching algorithm," *Mathematical Programming Computation*, vol. 1, pp. 43–67, 2009.

[57] D. P. DiVincenzo and P. Aliferis, "Effective fault-tolerant quantum computation with slow measurements," *Physical review letters*, vol. 98, no. 2, p. 020501, 2007.

[58] C. Gidney, "Stim: a fast stabilizer circuit simulator," *Quantum*, vol. 5, p. 497, 2021.

[59] O. Higgott, M. Wilson, J. Hefford, J. Dborin, F. Hanif, S. Burton, and D. E. Browne, "Optimal local unitary encoding circuits for the surface code," *Quantum*, vol. 5, p. 517, 2021.

[60] S. Bravyi, M. B. Hastings, and F. Verstraete, "Lieb-robinson bounds and the generation of correlations and topological quantum order," *Physical review letters*, vol. 97, no. 5, p. 050401, 2006.

[61] J. Roffe, D. Headley, N. Chancellor, D. Horsman, and V. Kendon, "Protecting quantum memories using coherent parity check codes," *Quantum Science and Technology*, vol. 3, no. 3, p. 035010, 2018.

[62] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen *et al.*, "State preservation by repetitive error detection in a superconducting quantum circuit," *Nature*, vol. 519, no. 7541, pp. 66–69, 2015.

[63] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer *et al.*, "Realization of quantum error correction," *Nature*, vol. 432, no. 7017, pp. 602–605, 2004.

[64] J. Cramer, N. Kalb, M. A. Rol, B. Hensen, M. S. Blok, M. Markham, D. J. Twitchen, R. Hanson, and T. H. Taminiau, "Repeated quantum error correction on a continuously encoded qubit by real-time feedback," *Nature communications*, vol. 7, no. 1, p. 11526, 2016.

[65] M. Gong, X. Yuan, S. Wang, Y. Wu, Y. Zhao, C. Zha, S. Li, Z. Zhang, Q. Zhao, Y. Liu *et al.*, "Experimental exploration of five-qubit quantum error-correcting code with superconducting qubits," *National Science Review*, vol. 9, no. 1, p. nwab011, 2022.

[66] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, "Fault-tolerant quantum error detection," *Science advances*, vol. 3, no. 10, p. e1701074, 2017.

[67] C.-Y. Lu, W.-B. Gao, J. Zhang, X.-Q. Zhou, T. Yang, and J.-W. Pan, "Experimental quantum coding against qubit loss error," *Proceedings of the National Academy of Sciences*, vol. 105, no. 32, pp. 11 050–11 054, 2008.

[68] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt, "Experimental repetitive quantum error correction," *Science*, vol. 332, no. 6033, pp. 1059–1061, 2011.

[69] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond bell's theorem," *Bell's theorem, quantum theory and conceptions of the universe*, pp. 69–72, 1989.

[70] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Applied physics reviews*, vol. 6, no. 2, p. 021318, 2019.

[71] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff, "Repeated quantum error detection in a surface code," *Nature Physics*, vol. 16, no. 8, pp. 875–880, 2020.

[72] Y. Zhao, Y. Ye, H.-L. Huang, Y. Zhang, D. Wu, H. Guan, Q. Zhu, Z. Wei, T. He, S. Cao *et al.*, "Realization of an error-correcting surface code with superconducting qubits," *Physical Review Letters*, vol. 129, no. 3, p. 030501, 2022.

[73] M. Gong, X. Yuan, S. Wang, Y. Wu, Y. Zhao, C. Zha, S. Li, Z. Zhang, Q. Zhao, Y. Liu *et al.*, "Experimental verification of five-qubit quantum error correction with superconducting qubits," *Preprint at https://arxiv.org/abs/1907.04507*, 2019.

[74] K. Keane and A. N. Korotkov, "Simplified quantum error detection and correction for superconducting qubits," *Physical Review A*, vol. 86, no. 1, p. 012333, 2012.

[75] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, "Trapped-ion quantum computing: Progress and challenges," *Applied Physics Reviews*, vol. 6, no. 2, p. 021314, 2019.

[76] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina *et al.*, "Fault-tolerant operation of a quantum error-correction code," *arXiv preprint arXiv:2009.11482*, 2020.

[77] C. Ryan-Anderson, J. Bohnet, K. Lee, D. Gresh, A. Hankin, J. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. Brown *et al.*, "Realization of real-time fault-tolerant quantum error correction," *Physical Review X*, vol. 11, no. 4, p. 041058, 2021.

[78] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. Poschinger, "Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer," *Physical Review X*, vol. 12, no. 1, p. 011032, 2022.

[79] F. Reiter, A. S. Sørensen, P. Zoller, and C. Muschik, "Dissipative quantum error correction and application to quantum sensing with trapped ions," *Nature communications*, vol. 8, no. 1, p. 1822, 2017.

[80] S. Slussarenko and G. J. Pryde, "Photonic quantum information processing: A concise review," *Applied Physics Reviews*, vol. 6, no. 4, p. 041303, 2019.

[81] K. Zhu, L. Yin, C. Wang, and G. Long, "Protecting the orbital angular momentum of photonic qubits using quantum error correction," *Europhysics Letters*, vol. 132, no. 5, p. 50005, 2021.

[82] C. Vigliar, S. Paesani, Y. Ding, J. C. Adcock, J. Wang, S. Morley-Short, D. Bacco, L. K. Oxenløwe, M. G. Thompson, J. G. Rarity *et al.*, "Error-protected qubits in a silicon photonic chip," *Nature Physics*, vol. 17, no. 10, pp. 1137–1143, 2021.

[83] L. Hu, Y. Ma, W. Cai, X. Mu, Y. Xu, W. Wang, Y. Wu, H. Wang, Y. Song, C.-L. Zou *et al.*, "Quantum error correction and universal gate set operation on a binomial bosonic logical qubit," *Nature Physics*, vol. 15, no. 5, pp. 503–508, 2019.

[84] T. Pittman, B. Jacobs, and J. Franson, "Demonstration of quantum error correction using linear optics," *Physical Review A*, vol. 71, no. 5, p. 052332, 2005.

[85] J. Kerckhoff, H. I. Nurdin, D. S. Pavlichin, and H. Mabuchi, "Designing quantum memories with embedded control: photonic circuits for autonomous quantum error correction," *Physical Review Letters*, vol. 105, no. 4, p. 040502, 2010.

[86] P. Aliferis, D. Gottesman, and J. Preskill, "Quantum accuracy threshold for concatenated distance-3 codes," *arXiv preprint quant-ph/0504218*, 2005.

[87] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, "Hybrid quantum-classical algorithms and quantum error mitigation," *Journal of the Physical Society of Japan*, vol. 90, no. 3, p. 032001, 2021.

[88] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, "Optimizing quantum error correction codes with reinforcement learning," *Quantum*, vol. 3, p. 215, 2019.