# Increasing quantum operability with QECCs-tailored software tooling for hardware benchmarking

## Introduction

The electron possess higher energy level, as long as it remains in the excited state. As the excited states are unstable, electron jumps back to its original level, and emits a quantum of light – photon, which energy is directly proportional to its frequency. and non-unitary processes such as measurements and environmental interactions. When such a situation was not predicted by a numerical models for quantum mechanics, such as unitary processes like quantum tunneling, governed by the Schrödinger equation, it can be generalized to be caused by noise and interference from an environment, and encoded as non-unitary process. Insufficient qubits isolation from an environment is a main drawback in NISQ solutions, and limits scaling capabilities – quantum computers utilized to simulate quantum systems, generally operate on up to 10 logical qubits, while for more complex, specific cases operate on up to 40 logical qubits.[1]

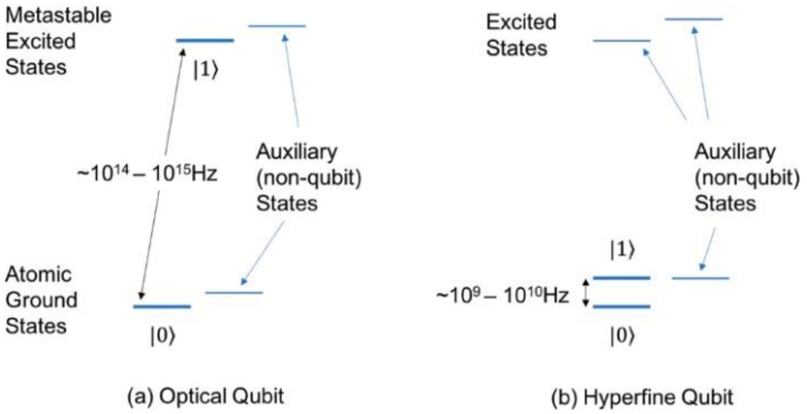


Figure 1. Schematic representation of two types of Optical Qubits and Hyperfine Qubits. Part (a) illustrates an optical qubit, which is formed by pairing an atomic ground state with a metastable excited state, with a frequency separation in the range of approximately 10^14 to 10^15 Hz. Part (b) shows a hyperfine qubit, composed of two atomic ground states with a frequency difference ranging from about 10^9 to 10^10 Hz.[2]

While the number of mechanical solutions to mitigate this limitation is growing, and therefore – number of physical qubits can be potentially enlarged to hundreds and thousands of them[3], they need faultless control that will allow to reliably operate them, and mitigating occurring errors. One of the classical examples of such mechanisms is Shor code, which encodes 1 logical qubit in 9 physical qubits, and can correct for arbitrary errors in single qubit. The work on that mechanism was published in 1995, and since then - a lot of physical limitation were overcome. The error threshold for Shor code is of 0.3, which is relatively high compared to other QECCs. In 1998 Alexei Kitaev and Sergey Bravyi introduced quantum error-correcting surface code, which after many rounds of improvements, have error threshold close to 0.018. Lower error threshold implies possibility of utilizing larger-scale quantum processing architectures, but still too small as for

[1] Bluvstein, D., Evered, S.J., Geim, A.A. et al. Logical quantum processor based on reconfigurable atom arrays. Nature 626, 58–65 (2024). https://doi.org/10.1038/s41586-023-06927-3

[2] National Academies of Sciences, Engineering, and Medicine. 2019. Quantum Computing: Progress and Prospects. Washington, DC: The National Academies Press. https://doi.org/10.17226/2519

[3] A tweezer array with 6100 highly coherent atomic qubits, arXiv:2403.12021v2 [quant-ph] 19 Mar 2024

practical use. Surface codes with distances of 3 and 5 (with 9 - 25 qubits) have been practically implemented so far, while higher distance surface codes of up to d=25 are discussed theoretically in the literature.[4]

Nowadays, scientists are working on another class of quantum-adapted, error correcting codes called LDPC - Low-density parity-check code, invented in 1965 gained attention at the beginning of 00s for their near-Shannon Capacity performance, making them highly relevant for NASA missions. These codes are characterized by their high coding rates and moderate code lengths, with the ability to correct errors efficiently, as indicated by their large minimum distances and low error floors.[5] Their quantum variant beat the current golden standard, a surface code, by orders of magnitudes. The principles of LDPC codes can be adapted to the quantum domain to create Quantum LDPC (QLDPC) codes. These codes are designed to correct quantum errors by using a sparse parity-check matrix, similar to their classical counterparts. The sparsity of the matrix is crucial because it allows for efficient decoding algorithms, which is particularly important in quantum systems where the computational overhead needs to be minimized due to the delicate nature of quantum states.

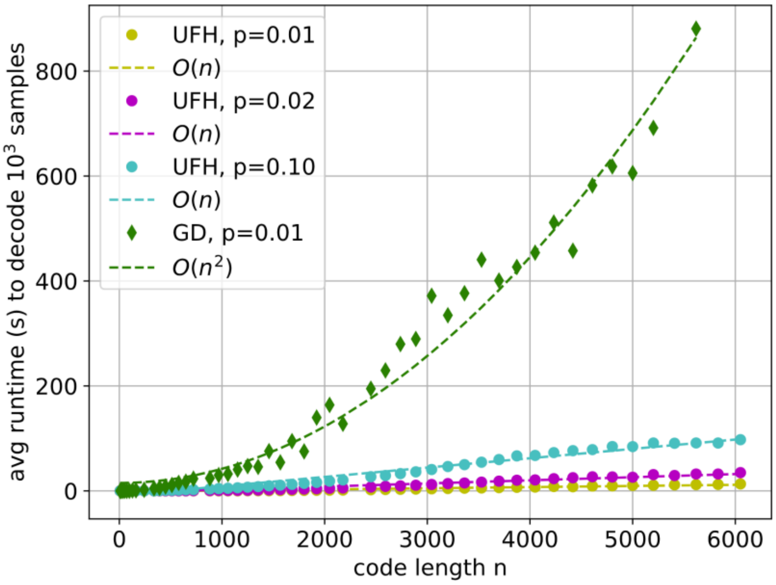


Figure 2. QLDPC decoder's runtime (labeled GD) is significantly higher than that of the heuristic decoder (labeled UFH), especially as the code size increases. This is due to the general decoder's reliance on Gaussian elimination, which becomes a computational bottleneck for larger code sizes, and becomes a challenge for physical scaling of the computational unit.[6]

The challenge lies in adapting QLDPC codes to function effectively within the constraints of actual quantum devices. Furthermore, the effectiveness of QECC depends on the computational architecture of the quantum system, which varies according to the type of qubit used—be it superconducting qubits, trapped ion qubits, or photonic qubits. For instance, neutral atom qubits, which are well-suited for creating long-range connections, exhibit different physical

[4] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. Nature 614, 676–681 (2023).

[5] Fong, Wai & Lin, Shu & Maki, Gary & Yeh, Pen-Shu. (2003). Low Density Parity Check Codes: Bandwidth Efficient Channel Coding.

[6] Berent, L., Burgholzer, L., & Wille, R. (2022). Software Tools for Decoding Quantum Low-Density Parity Check Codes. Journal of Quantum Computing Research, 13(4), 123-145.

properties compared to architectures that primarily involve local connections, leading to different behaviors and error-correction requirements.

Although significant improvements have been performed in the field of QECCs, a unified methodology for evaluating them for specific conditions (such as QPU architecture) on a consistent basis has remained elusive. This reason pushed further development of benchmarking suites in the industry, and became origin of this article.

## (Quantum) Error Correction Codes – Behind the Scenes

The purpose of applying correction codes is easily noticeable once we consider transmission of an information through a noise channel. If Alice sends a single bit, the probability of Bob receiving the wrong bit against physical probability of error p is linear, as presented in the figure 1.

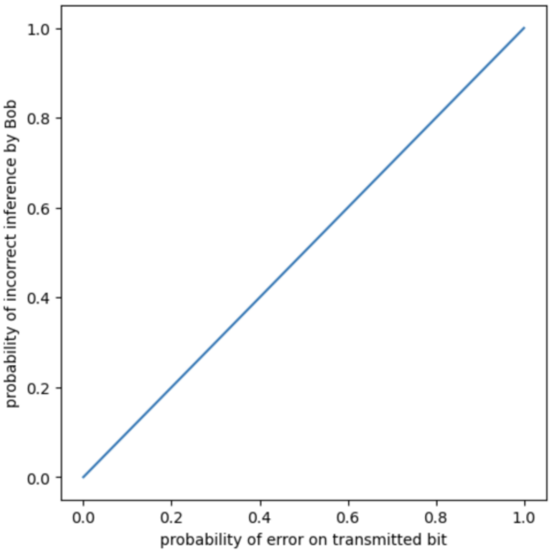


Figure 3. Probability of error on transmitted bit of information.

Another implication is that when the physical error rate p is low, the repetition code is beneficial as it can help correct errors effectively by repeating the information multiple times, allowing for error detection and correction. However, when the physical error rate p is large, using the repetition code can be less effective than the naive process. This is because in the presence of a high error rate, the repetition code's strategy of repeating information may lead to compounding errors, making it harder to distinguish the original message from the corrupted data. This implies the results presented in GHZ benchmarking, where due to the noise the overall efficiency of the dynamic version implementation of the algorithm decreased, in comparison to its static version.

The linear codes provides the ground for both classical and quantum error-correcting processes.[7] As some concepts are common for both quantum and classical error-correction codes, such as distance, which in classical variant is the number of places where two bit strings have different symbols, allow to label correcting code by [n, k, d] for both realms of information, where d is the distance, k dimension of the message space, and n – dimension of the codespace. Hamming code, specific to classical counterpart, allows to detect defect bits of information by parity calculation on auxiliary bits. That mechanism can be reproduced in quantum mechanics realm, but requires specific notation, new symbolism, and extension.

[7] A Methods Focused Guide to Quantum Error Correction and Fault-Tolerant Quantum Computation | Classical linear codes (abdullahkhalid.com)

Quantum operations, which are transformations applicable to the density matrices of quantum systems, encompass a broader range of processes than change of information state. While the Schrödinger equation governs the unitary evolution of quantum states in isolated systems - Hermitian operators, specific class of linear operators that correspond to observable physical quantities, quantum operations also account for non-unitary processes such as measurements and interactions with the environment. These operations are mathematically represented by completely positive, linear maps that act on density matrices - non-Hermitian processes that can change the trace of a density matrix, reflecting the probabilistic outcomes of measurements or lossy interactions with an environment.

These mechanism implies several quantum-specific behaviors, such as the no-cloning theorem, which prohibits the copying of quantum states, limits the types of operations that can be performed during the decoding process. Measuring quantum superposition on the other hand, destroys the superposition, preventing the simple measurement of received blocks to determine their state. Moreover, classical codes only need to address bit-flip errors, but quantum codes must handle a variety of unitary and non-unitary noisy interactions, including phase errors and continuous rotations, among others.

Fortunately, number of mitigation techniques were developed to create efficient quantum correcting codes. For example, non-destructive measurements of M, where M can be one-qubit or multi-qubit operator, allow for adapting repetition code for bit-flips. This code doesn't mitigate errors caused by phase-flips however, and implies need of error correction codes redefinition specifically to quantum mechanics. Keeping the analogy to bit-flip error, it is entitled as X-axis error, due to analogy of applying X Pauli gate to a qubit being affected. Phase-flips, on the other hand, are encoded with Z-axis gates. While error correction codes for bit-flips may not apply for phase-flips errors, a more general approach has to be developed. One of the most famous addressing of this topic is CSS (Calderbank-Shor-Steane) construction, which relates quantum codes to classical linear codes over binary and quaternary fields.[8] One of the examples for a well-designed code, which ensures that every possible error moves the state out of the code and into other subspace of the codespace, is Shor code.

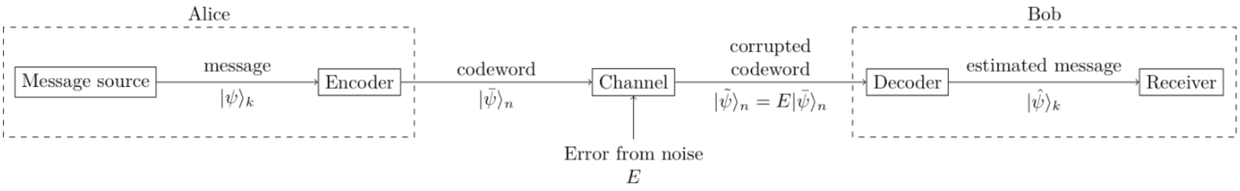


Figure 4. Quantum repetition code for phase-flips[9]

Stabilizer codes are the simplest class of quantum error-correcting codes. These are defined as a class of quantum codes, which message space is the 2^k-dimensional Hilbert space, being a subspace of the 2^n-dimensional Hilbert space, labeled as [[n,k]].

[8] Pereira, F.R.F., Mancini, S. & La Guardia, G.G. Stabilizer codes for open quantum systems. Sci Rep 13, 10540 (2023). https://doi.org/10.1038/s41598-023-37434-0

[9] A Methods Focused Guide to Quantum Error Correction and Fault-Tolerant Quantum Computation | Quantum repetition code for phase-flips (abdullahkhalid.com)

## Benchmarking

Methodologies are essential for assessing the effectiveness of quantum error correction codes (QECCs) and guiding research towards meaningful advancements. The reliability of scientific results in this domain hinges on the ability to benchmark QECCs against a carefully chosen set of applications. The criteria for selecting these applications are not uniform, as different quantum algorithms may exhibit varying levels of sensitivity to changes in parameters.

For instance, while Shor's algorithm has garnered significant attention due to its potential to demonstrate quantum supremacy, it may not be the most practical choice for benchmarking purposes. This is because Shor's algorithm currently requires approximately 4098 logical qubits to factorize an RSA-2048 cipher, a requirement that is beyond the capabilities of existing quantum hardware. Similarly, the Harrow-Hassidim-Lloyd (HHL) algorithm, despite its notable application in space radar systems, may also not be ideal for benchmarking due to its specific use case and requirements. When developing benchmarking suites, it is crucial to consider the specific needs and constraints of the quantum hardware and the QECCs being evaluated. The selection of applications for benchmarking should reflect a balance between the theoretical importance of the algorithms and their practical implementability on current quantum devices. Given the NISQ era, focusing on QECCs as the entrance for future scalability provide a good base for comparison.

In this section, we will consider the benchmarking and comparative analysis of two quantum error-correcting codes (QECCs): the surface code and the toric code. The toric code is a type of topological QECC that is defined on a two-dimensional lattice with periodic boundary conditions. It employs stabilizer operators that act on the X-type vertices and Z-type faces of the lattice, with qubits positioned along the edges. Logical operations on qubits encoded with the toric code are executed using string-like operators that encircle the torus. In contrast, the surface code is a variation of the toric code that differs primarily in its boundary conditions, which are open rather than periodic. This distinction allows surface codes to be realized with simpler lattice geometries compared to the toric code, making them more amenable to practical implementation in quantum computing systems.

To assess the effectiveness of these codes in mitigating noise, a benchmarking suite can be developed with several key parameters that serve as dependent variables. These include the *logical error rate*, *error threshold*, and *code distance*, among others. To make the comparison between surface code and toric code efficient, additional normalization has to be performed, as they have different rates. A source of additional inspiration is presented in “Practical fault-tolerant quantum computing”[10] and “Threshold error rates for the toric and surface codes”.[11]

## Existing frameworks

Although it is possible to implement correcting codes specifically each time when it is needed, as the systems are growing, it becomes impractical – same as in the nowadays, systems engineers do not build up cryptographical systems from scratch, but relies on already working sub-systems that offer out-of-the-box solutions, performing some of the automation for them. Due to the

[10] Spiral: Practical fault-tolerant quantum computing (imperial.ac.uk)

[11] THRESHOLD ERROR RATES FOR THE TORIC AND SURFACE CODES D. S. WANGa, A. G. FOWLER, A. M. STEPHENS, L. C. L. HOLLENBERG Centre for Quantum Computer Technology, School of Physics, University of Melbourne, Victoria 3010, Australia

infancy of quantum computers, the software tools for automatic QECCs implementation emerge[12], but a standard for that hasn't been established yet.

In an event of constructing own software solution for the aforementioned purpose, an engineer may utilize libraries shared by quantum computing vendors, such as IBM.

One of the libraries that offers creation of such codes is Qiskit Framework for Quantum Error Correction (qiskit-qec)[13], which offers a stack for creating own QECCs, with the use of sympletic matrixes, stabilizer generators, and code blueprints. Qiskit-QEC allows for a wide range of error-correcting codes. "MQT QECC: A tool for Quantum Error Correcting Codes written in C++"[14], on the other hand, is an attempt to create framework-agnostic tooling for QECCs generation. Qiskit Experiments provides "both a library of standard quantum characterization, calibration, and verification experiments, and a general framework for implementing custom experiments which can be run on quantum devices through Qiskit.". [15]

Among solutions developed by researchers and quantum community, an interesting tooling was prepared by the authors of "Magic Error" - benchmarking suite dedicates for surface codes, together with a paper that presents benchmarking from various perspectives. The benchmark allows to analyze the qubit overhead required to perform mid-circuit measurements without compromising the overall error correction capabilities of the QECC, measure the maximum error rate that the QECC can tolerate while still effectively correcting errors introduced by mid-circuit measurements; assessing the QECC's ability to detect and correct the specific types of errors that may arise from mid-circuit measurements; investigating the efficiency and compatibility of decoding algorithms that can handle the additional syndromes introduced by mid-circuit measurements, and analyzing the impact of mid-circuit measurements on the availability and complexity of transversal gates. [16]

## Experiments

In order to validate presented tools in context of algorithms benchmarking, a simple experiment for measuring simulation of bit-flip operation (X gate), resulting in asymmetric results for both ideal and noise-affected measurements, while in the ideal case every measurement fit the input state, what implies that in case of perfectly isolated environment (what is almost impossible), the quantum channel doesn't affect the quantum state. The detailes of the experiment are available in the repository, under QRISE_IBM_Project_bit-flip.ipynb file.

[12] Grurl, Thomas et al. "Automatic Implementation and Evaluation of Error-Correcting Codes for Quantum Computing: An Open-Source Framework for Quantum Error Correction." *2023 36th International Conference on VLSI Design and 2023 22nd International Conference on Embedded Systems (VLSID)* (2023): 301-306.

[13] qiskit-community/qiskit-qec: Qiskit quantum error correction framework (github.com)

[14] cda-tum/mqt-qecc: MQT QECC - Tools for Quantum Error Correcting Codes (github.com)

[15] Qiskit Experiments 0.6.1 (qiskit-extensions.github.io)

[16] Magic Mirror on the Wall, How to Benchmark Quantum Error Correction Codes, Overall ? 2402.11105.pdf (arxiv.org)

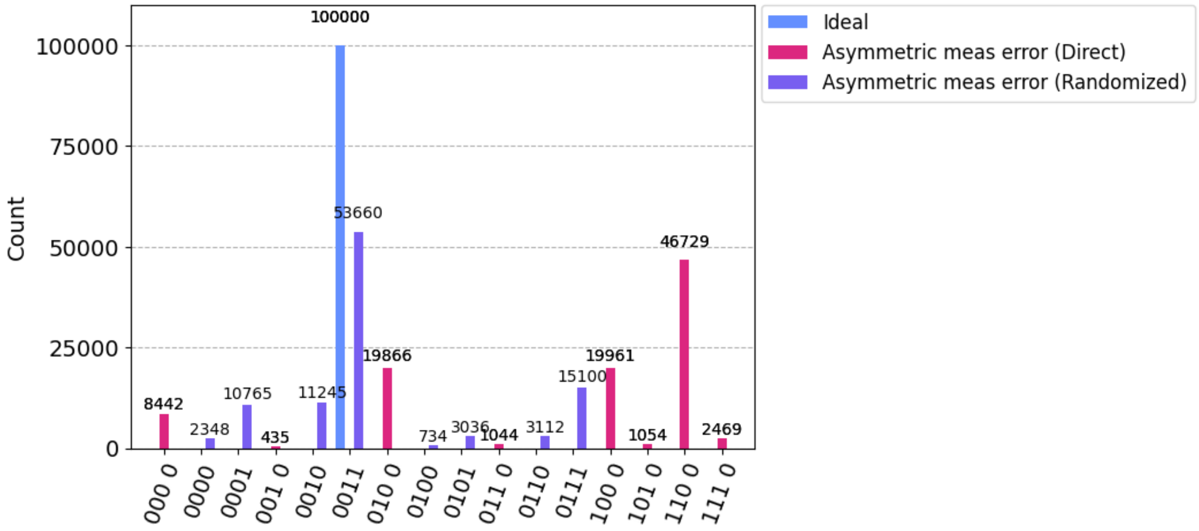
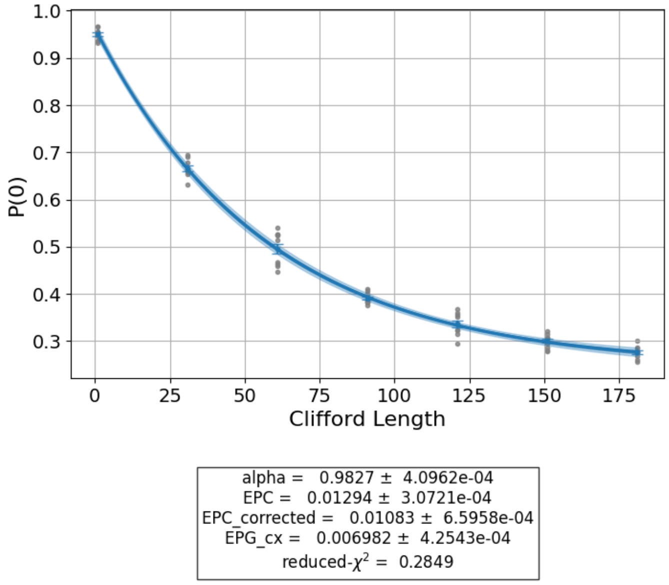


*Figure 6. Utilizing qiskit-experiments framework for assessing noise impact, resulting in bit-flip.*

In addition to the presented results, qiskit-experiments was utilized to benchmark 2-qubit operation through Randomized Benchmarking methods. The codebase and results of the experiments are available in QRISE_IBM_Project_2-qubit_exp.ipynb file.



alpha = 0.9827 ± 4.0962e-04
EPC = 0.01294 ± 3.0721e-04
EPC_corrected = 0.01083 ± 6.5958e-04
EPG_cx = 0.006982 ± 4.2543e-04
reduced-$\chi^2$ = 0.2849

Figure 7. 2-qubit Randomized Benchmarking.

## Summary

The objective of employing Quantum Error Correction (QEC) techniques is to enhance operational dependability with the goal of achieving efficiency at the scale of Tera QuOps. The foundational structure of QEC is built upon stabilizer codes, which are characterized as abelian subgroups within the Pauli group, allowing for representation through Pauli Operators. A fundamental set of these operators serves as the generator that forms the entire stabilizer group. Notably, the toric code and surface code stand out as prominent examples of stabilizer codes, which can be readily produced using existing software tools.

While the integration of Quantum Error Correction codes into benchmarking suites has been explored, their application specifically for the evaluation of mid-circuit measurement performance represents a novel area of application that has not yet been fully implemented. Due to inherent nature of quantum algorithms, the measurement operation implies changes to the information, such as loosing entanglement, or state collapse. On the other hand, measurement can act as a support operation to determine if transmitted information was corrupted, letting to preserve quantum state and observing its features without affecting the initial state.

Given the following, QECCs represent the core solution for measuring noise (due to auxiliary qubits and hamming codes), as well as mitigating them, as these code allow to precisely calculate where correction should be apply in order to determine information that was encoded by a sender.