QcAssert: Quantum Device Testing with Concurrent Assertions

Hasini Witharana, Daniel Volya and Prabhat Mishra
University of Florida, Gainesville, Florida, USA

***Abstract*—Quantum devices are extremely noisy due to its inherent architecture. This can introduce errors or completely erase the information stored in qubits. High noise levels in a quantum device can lead to errors even when the quantum circuit is not buggy. Therefore, it is essential to verify that the noise level of the device is tolerable while running the quantum circuit. In this paper, we propose a quantum device testing framework using concurrent assertions. Specifically, we introduce a new type of assertion "QcAssert", which has the ability to run concurrently with the quantum circuit to ensure that the quantum device is working as expected. We demonstrate the effectiveness of the QcAssert in dynamic device testing using a suite of popular quantum benchmarks, including Shor's factoring algorithm and Grover's search algorithm.**

I. Introduction

Quantum computing has the potential to outperform its classical counterpart in solving many hard optimization problems. It is important to verify the correctness of quantum computers to ensure reliable and accurate performance. Since qubits rely on the principles of quantum mechanics, they possess an inherent small size and extreme susceptibility to environmental interactions. These interactions have the potential to introduce errors into the qubits or even completely erase the information they store. Hence, it is crucial to validate that the quantum device operates as intended when implementing the quantum circuits. Traditionally, users assess the device's condition by executing a small (known) circuit, before implementing the actual design, and analyzing the noise level. However, this approach does not provide a guarantee of the noise level during the actual execution.

Assertions are widely used for classical device testing such as post-silicon validation and in-field debugging [1], [2], [3]. Assertions provide a mechanism to describe desirable properties of a system that should be satisfied. In classical domain, there are two main types of assertions: concurrent and immediate assertions. Concurrent assertions are used to verify conditions that should hold true simultaneously during program execution. Immediate assertions, on the other hand, provide means of checking conditions or assumptions at specific points in the code, helping with self-checking and debugging purposes.

There are various types of quantum assertions [4], [5], [6] which can check a specific state of a quantum circuit. Specifically, these assertions consider three possible states: classical, superposition, and entanglement. Existing quantum assertions [4], [5], [6] can be considered as immediate assertion type due to the nature of these assertions. The classical quantum assertion has the ability to check whether the quantum state is classical at a specific breakpoint of the design. Similarly, superposition and entanglement assertions have the ability to check whether a quantum state is a uniform superposition and bell entangled at a given breakpoint, respectively. A major limitation of these assertions is that they can only check a very specific state at a given time. Moreover, these assertions do not give any guarantee about the quantum device during execution. *To the best of our knowledge, there are no concurrent type assertions defined for quantum computing*.

This work was partially supported by the NSF grant CCF-1908131.

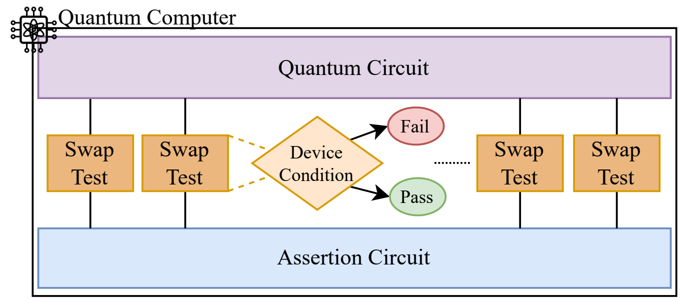


Fig. 1: Overview of quantum device testing using concurrent assertions that continuously checks validation conditions.

In this paper, we propose a dynamic quantum device testing framework using concurrent assertions. Specifically, we introduce a novel quantum assertion called "QcAssert", designed to run simultaneously with the quantum circuit. QcAssert will ensure that the quantum device is functioning as expected during execution. Figure 1 shows an overview of the proposed concurrent assertion-based validation framework. While the quantum circuit is getting executed, device noise is checked to see whether the device is working as expected. Our assertion circuit is designed to replicate the behaviour of the original gates in a single qubit. We use a swap test to compare the original circuit and the assertion circuit during execution. A swap test is a quantum algorithm that compares two quantum states to determine their similarity or distinguishability. This is used to check whether the noise level added by the quantum device is changing the functionality of the original circuit. Since the assertion circuit is running in parallel to the design, the results of the swap test can be used as an indicator of the quantum device condition while running the design. Specifically, this paper makes the following contributions:

- Identifies a single qubit representation of the original channel by approximating quantum channels via diamond norm minimization.
- Introduces a new quantum assertion "QcAssert", that runs concurrently with a quantum circuit.

- Uses swap testing in "QcAssert" to identify whether the quantum device is noisy.
- Demonstrates the utility of these assertions in verifying quantum devices using popular quantum benchmarks in both simulation and IBM quantum environments.

This paper is organized as follows. Section II surveys related efforts. Section III describes our assertion-based validation framework. Section IV presents experimental results. Finally, Section V concludes the paper.

II. Related Work

There are two types of assertions in classical domain [1]: immediate and concurrent assertions. Concurrent assertions serve the purpose of validating conditions that must hold true simultaneously while the program is running. On the contrary, immediate assertions offer a way to inspect conditions or assumptions at a particular point. Classical assertions cannot be directly used in quantum circuits since output is deterministic for a given input for classical computing, while output values are a result of destructive measurements and come with a probability distribution in quantum computing. There are early efforts to discuss the importance and applicability of quantum assertions [5], [6], [7]. Recent approaches explored different assertion generation methods such as ancilla-based methods [4], statistical methods [5], and projection-based methods [6]. All these assertions can be considered as immediate type assertions since they only check a specific state at a specific point during the execution.

There are related efforts in quantum error correction [8] and formal verification of quantum circuits. Error correcting codes assume a certain noise model and provide special state encoding that can correct a state if an error is detected. Quantum assertions, although similar, are not concerned with correcting a state and only seek to assert a given property of the state. There are also recent efforts to check the correctness in the output of a quantum circuit, such as through formal verification of quantum circuits [9], [10], or by assuming domain-specific knowledge (e.g., post-selection rules) to ignore incorrect outputs of a quantum computation [11], [12]. Simulation-based testing such as switch test [13] is also used in quantum device testing. To the best of our knowledge, our proposed approach is the first attempt in generating concurrent assertions that can verify the quantum device during execution.

III. Quantum Concurrent Assertion Generation

Figure 2 shows an overview of our proposed concurrent assertion based device testing framework that consists of three major phases: circuit approximation, assertion generation, and assertion-based device testing. In the first step, we need to find a unitary gate representation of the original quantum channel by approximation. This unitary gate is used in the next step to generate the concurrent assertion circuit. Since one qubit is used in the assertion circuit, the overhead of adding the assertion can be minimized. In the second step, we generate the assertion circuit by combining the original circuit and the unitary circuit. In the last step, we perform the device testing using the modified circuit. The results of the assertion circuit can be used to identify the noise level during execution. We first define two important concepts. Next, we describe the three majors steps in our proposed framework.

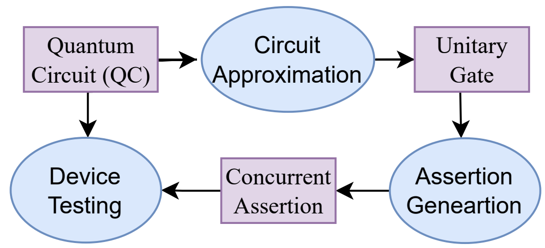


Fig. 2: Our proposed framework for quantum device testing using concurrent assertions that consists of three major tasks.

A. Definitions

Quantum Channel: A quantum channel, denoted as $\mathcal{E}$, refers to a specialized map defined as

$$\mathcal{E} : \mathcal{H}(\rho) \mapsto \mathcal{H}'(\rho') \tag{1}$$

This map transfers density states from a certain Hilbert space $\mathcal{H}$ of dimension N to density states on another (possibly different) Hilbert space $\mathcal{H}'$ of dimension M. To resemble a physical process, the channel $\mathcal{E}$ must maintain the properties of complete positivity and trace preservation. An example of a quantum channel can be the unitary operators, such as quantum circuits or quantum gates, which enable coherent evolution of states, expressed as $\rho' = U\rho U^\dagger$.

Diamond Norm: The diamond norm (completely bounded trace norm) provides a measure of the maximum distance between the behavior of two quantum channels, thereby quantifying their difference in a physically meaningful way. Mathematically, for two quantum channels $\mathcal{E}$ and $\mathcal{F}$, the diamond norm is defined as:

$$\|\mathcal{E} - \mathcal{F}\|_\diamond = \sup_\rho \|(\mathcal{E} \otimes \mathcal{I})(\rho) - (\mathcal{F} \otimes \mathcal{I})(\rho)\|_1 \tag{2}$$

Here, the supremum is taken over all input states ρ that can be operated upon by the channels. The operator $\mathcal{I}$ stands for the identity map on an ancillary system, and $||.||_1$ denotes the trace norm. The diamond norm thereby measures the worst-case effect of the difference between the two channels on any input state, accounting for possible correlations with an auxiliary system. This norm is especially important when considering error bounds and the robustness of quantum systems. A small diamond norm difference between two quantum channels implies that they have almost the same effect on all input states, which can be vital in assessing the quality of quantum gates and their implementations.

B. Circuit Approximation

We approach assertion generation by approximating quantum channels via diamond norm minimization. The aim is to efficiently approximate a complex quantum channel with a simpler one that ensures accurate and reliable quantum computations.

Let $\mathcal{E}$ denote the original quantum circuit (channel) and the approximated circuit (channel) as $\mathcal{F}$. The approximated channel is selected from a set of simpler channels, often determined by the constraints of a particular quantum system or programming framework. The objective is to ensure that the diamond norm difference $||\mathcal{E}-\mathcal{F}||_\diamond$ is minimized, thereby asserting that the actual quantum operation (implemented by $\mathcal{F}$) closely mimics the desired operation (defined by $\mathcal{E}$). The approximated circuit can thus be formally written as:

$$||\mathcal{E}-\mathcal{F}||_\diamond \leq \epsilon \tag{3}$$

where ϵ is a pre-determined tolerance level. This assertion verifies the degree to which the implemented channel approximates the desired one. If the diamond norm difference exceeds the tolerance ϵ, the approximation fails, which suggests the quantum computation is not performing the desired operation with sufficient accuracy, hence triggering an error or exception.

We approach the minimization problem through gradient descent. The first step involves defining a differentiable parameterization for the approximating quantum channel $\mathcal{F}(\theta)$, where θ represents the set of tunable parameters. The function $\mathcal{F}(\theta)$ should be designed such that it can represent a wide range of possible quantum channels, subject to any constraints specific to the problem at hand (such as a circuit implementable by a given set of quantum gates). The diamond norm $||\mathcal{E}-\mathcal{F}(\theta)||_\diamond$ is then calculated. This norm, as a function of θ, is generally non-convex and may have many local minima. Nonetheless, a local minimum can often provide a good enough approximation in practice.

The gradient of the diamond norm with respect to the parameters θ, denoted by $\nabla_\theta||\mathcal{E}-\mathcal{F}(\theta)||_\diamond$, is computed. This gradient points in the direction of the steepest increase of the diamond norm, and therefore, by moving in the opposite direction, we can reduce the diamond norm. The parameters are then updated iteratively using the rule:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta\nabla_\theta||\mathcal{E}-\mathcal{F}(\theta_{\text{old}})||_\diamond \tag{4}$$

where η is the learning rate, which determines the step size in each iteration. This process is repeated until convergence is achieved, that is, until the change in the diamond norm falls below a specified threshold, or a maximum number of iterations is reached.

C. Assertion Generation

After obtaining an approximate quantum channel $\mathcal{F}$ that closely mimics the functionality of an original quantum channel $\mathcal{E}$, our next concern is assertion generation – to monitor whether any noise or other perturbations may cause $\mathcal{F}$ to deviate from its desired behavior. To facilitate this monitoring, we propose the use of the swap test, a well-established technique in quantum computing that measures the overlap or similarity between two quantum states. If we consider two states $|\phi\rangle$ and $|\psi\rangle$, the protocol begins with an auxiliary qubit yielding an overall system state as $|\Psi\rangle = |0\rangle|\phi\rangle|\psi\rangle$. After a Hadamard gate on the auxiliary qubit, a controlled swap gate, and another Hadamard gate on the auxiliary qubit, the total state is

$$|\Psi'\rangle = \frac{1}{2}|0\rangle(|\phi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle) + \frac{1}{2}|1\rangle(|\phi\rangle|\psi\rangle - |\psi\rangle|\phi\rangle). \tag{5}$$

Then, the probability of measuring the auxiliary qubit as 0 is given as $\frac{1}{2}+\frac{1}{2}|\langle\psi|\phi\rangle|^2$. If the states are equal, then $|\langle\psi|\phi\rangle|^2 = 1$ and the probability is unity. Depending on the outcome of the auxiliary qubit, the post-measurement state is either of the following:

- $0 \rightarrow |\Psi'\rangle = \frac{1}{\sqrt{2}}(|\phi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle)$ (symmetric)
- $1 \rightarrow |\Psi'\rangle = \frac{1}{\sqrt{2}}(|\phi\rangle|\psi\rangle - |\psi\rangle|\phi\rangle)$ (antisymmetric)

While the original states are changed, these can be useful for further symmetry tests [14], particularly in a multipartite system.

In our case, we employ the swap test to detect any significant deviation in the functionality of the approximated channel $\mathcal{F}$ due to noise. The procedure is summarized as follows:

1) Start by preparing a set of input states, denoted as ρ_i, where i spans the selected set. In principle, the states may be generated as part of the execution of the quantum circuit.
2) Apply both $\mathcal{E}$ and $\mathcal{F}$ to these input states, resulting in output states referred to as $\mathcal{E}(\rho_i)$ and $\mathcal{F}(\rho_i)$, respectively.
3) Implement the swap test to measure the degree of similarity between the output states from each pair of input states.

The resulting assertion can be framed as:

$$\frac{1}{N}\sum_{i=1}^{N}\text{SWAP}(\mathcal{E}(\rho_i), \mathcal{F}(\rho_i)) \geq \xi. \tag{6}$$

Here, $\text{SWAP}(\mathcal{E}(\rho_i), \mathcal{F}(\rho_i))$ indicates the result of the swap test on the i-th pair of states, N is the total number of prepared input states, and ξ is a predetermined confidence threshold. This assertion checks whether the average overlap of the output states from channels $\mathcal{E}$ and $\mathcal{F}$ exceeds the confidence threshold ξ. A failure of this assertion, indicated by the average overlap falling below ξ, would suggest that the functionality of the approximated channel $\mathcal{F}$ diverges from the original channel $\mathcal{E}$ and results in triggering the assertion.

D. Device Testing

Figure 3 shows an example circuit which combines both original ($\mathcal{E}$) and assertion ($\mathcal{F}$) circuit. We have used swap test to combine the two circuits. Measurement is conducted to get the swap test results. We are using chi-squared testing to analyze the results of the swap test. The chi-squared test is a statistical hypothesis test that is used to identify significant differences between expected frequencies with observed frequencies [15]. Without noise, the measurement for the auxiliary qubit should follow a unimodal distribution as shown in Figure 3. For the concurrent assertion, the hypothesis of

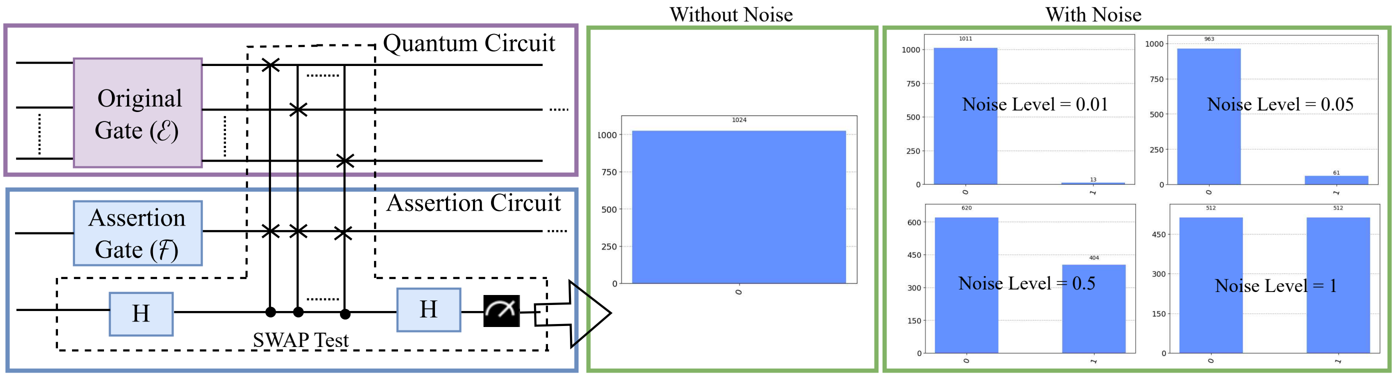


Fig. 3: Device testing with concurrent assertions. The original circuit and concurrent assertion are combined using the swap test (left). The null hypothesis accepted (middle). The null hypothesis is rejected (right), which indicates the presence of noise.

the chi-square test is selected such that the expected distribution should be a unimodal distribution with one peak value. After comparing the expected distribution with the observed distribution, if the $p-value$ of the test is less than 0.05, the null hypothesis is rejected. This means that the observed distribution is not unimodal. If the $p-value$ is higher and closer to 1, the null hypothesis is accepted, which implies that the observed distribution is unimodal, as shown in Figure 3 without noise.

The hypothesis will be rejected when the device is noisy. As presented in Figure 3, for different noise levels the distribution is not following a unimodal distribution. However, this will not fail the assertion. When the hypothesis is rejected, the assertion measurement is calculated using Equation 6. The probabilities of states $|0\rangle$ and $|1\rangle$ change for different noise levels. The user can define the confidence threshold ξ which will determine how confident the results should be when the noise is present. If the assertion measurement is less than the confidence threshold ξ, the assertion will fail.

QcAssert can be used as either a noise monitor or as an assertion for quantum device testing. If QcAssert is applied without a confidence threshold ξ, it can be used as a noise monitor while the quantum circuit is getting executed. If the user wants to stop the execution of the quantum circuit when the noise level is high, our framework can be applied with the confidence threshold. Moreover, our framework can be applied gate-wise and circuit-wise as well. In gate wise, we can add the assertion circuits to different gates. This can be used to monitor how the noise level dynamically scales with the depth of the design. When applying the framework circuit-wise, we can generate a unitary circuit that represents the whole execution of the circuit and apply it to the original circuit. Both of these methods can be used to dynamically test the noise level in the quantum device.

For the existing quantum assertions [4], [5], [6], prior knowledge of the quantum state is necessary before proceeding with the execution. This allows the continuation of the execution after measuring the assertion. Even when dealing with randomized states, knowledge of the expected state remains essential when employing these quantum assertions. Unlike classical systems where states can be copied and maintained separately for execution, the no-cloning theory in quantum mechanics prevents such duplication of states. However, with the introduction of QcAssert, it becomes possible to apply assertions without prior knowledge of the quantum state. This is a significant advantage as it overcomes the need for state information, which can be challenging to obtain in certain cases. It is important to note that the assertion process may sometimes alter the original state based on the original circuit. To address this limitation, a possible approach is to apply the assertion batch-wise and iteratively check the device accuracy.

IV. Experiments

This section demonstrates the effectiveness of our framework for quantum device testing using concurrent assertions. First, we describe our experimental setup. Next, we present the quality of the generated assertions for noise detection.

A. Experimental Setup

For experimental evaluation, we have selected quantum circuits that are widely used in the quantum assertions community. The assertion generation framework is implemented using Python and Qiskit. The classical simulation is performed using an Aer simulator. We ran our experiments on Intel i7-5500U @ 3.0GHz CPU with 16GB RAM machine. For execution on a quantum device, we used *ibmq_quito*, a 5 qubit machine with an estimated quantum volume of 16.

B. Experimental Results

We present our experimental results in three avenues: accuracy of unitary matrix representation, quantum circuit stability with assertion circuit and the quality of assertion.

1) Circuit Approximation Results: The initial phase in our assertion generation framework involves identifying a channel that can effectively optimize the diamond norm relative to the original circuit. An illustration of the optimization landscape of the diamond norm between the genuine quantum circuit and the unitary circuit representation of the CNOT circuit is provided in Figure 4. The landscape's non-convex character

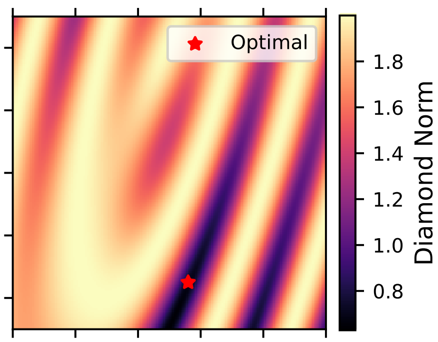


Fig. 4: Optimization landscape for CNOT

results in several local minima, a challenge we successfully navigate by employing stochastic gradient descent. Upon discovering an optimal solution, it is important to note that the solution demonstrates stability within its defined region. This implies that the solution maintains its stability even in the face of minor perturbations or deviations.

2) Impact of Noise in Circuits: When we run a quantum circuit, noise in the quantum processor can lead to deviations from the intended quantum state. Figure 5 shows how a quantum circuit (CX) can be effected by noise. Infidelity is used as a measurement to quantify the effect of noise in quantum channels. Infidelity is computed as 1 - fidelity. Fidelity measures the overlap between the ideal and the actual quantum channels. An infidelity of 0 represents a perfect match between the desired and actual quantum states, while an infidelity close to 1 suggests a greater deviation between the desired and actual quantum states. Figure 5 shows that that noise can significantly impact quantum channels. Specifically, when the noise level is low, infidelity is close to 0. Similarly, when the noise level is high, infidelity is close to 1.

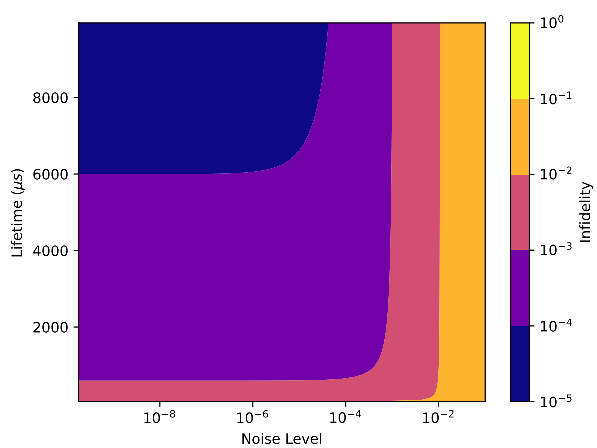
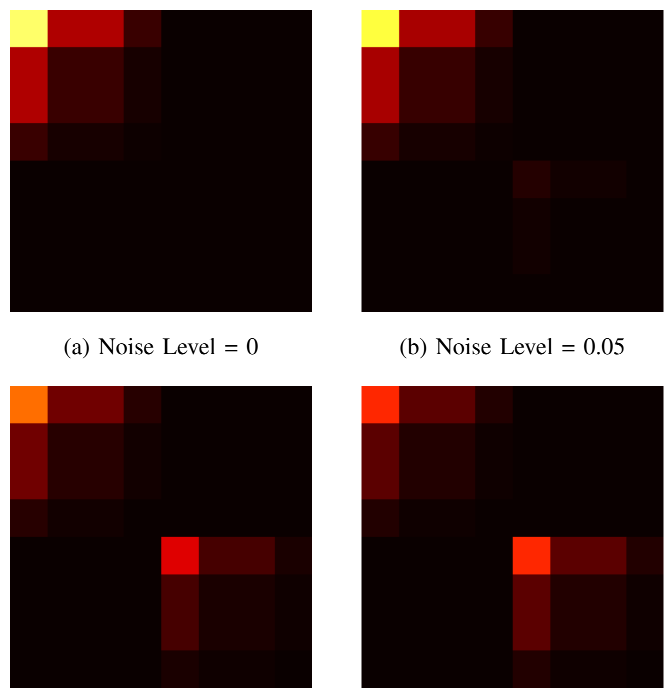


Fig. 5: Quantum circuit infidelity with varying noise levels

3) SWAP Test Stability with Noise: Since our assertion uses swap circuit to combine the original ($\mathcal{E}$) and assertion ($\mathcal{F}$) circuit, it is important to check the impact of the noise on the swap test circuit. We are using the density matrix of the swap circuit to show the similarity of the quantum design. Density matrix can be used to quantify how close the actual state of a quantum system is to the intended or target state. Figure 6 shows the density comparison of the swap circuit for different noise levels such as 0 (without noise), 0.05, 0.5, and 1 (extreme noise). When the noise increases, the density changes from the original density. This indicates that the swap circuit also gets impacted by the noise. Since we are only focusing on the measurement of swap being a non-unimodal distribution to identify the noise, the impacted swap test can still be used. When the noise level is high, the user can identify this behaviour by observing the assertion measurement and change the confidence threshold ξ accordingly.



(a) Noise Level = 0

(b) Noise Level = 0.05

(c) Noise Level = 0.5

(d) Noise Level = 1

Fig. 6: SWAP circuit density at different noise levels

4) Quantum Device Testing using Assertions: This section presents the quality of our assertion to detect the device noise. The experiment is conducted both in a simulation environment (Aer/Qiskit) and using an actual quantum computer (ibmq_quito). For the simulation environment, a custom noise model with depolarizing error was applied. Then the noise level was increased to check the quality of assertion in identifying the noise levels. Table I shows the assertion measurement results for simulation environment for different noise levels. The noise level rages from 0 to 1. The assertion measurement indicates the similarity between the expected measurement and actual measurement. For example, the CNOT obtained a 98.53% assertion measurement when noise level is 0.01 (less noisy). This means that we have a confidence of 98.53% that the actual outcome and the expected outcome of the design is similar. When noise level is 1 (extreme noise), we have less confidence and are unable to discern the correctness (49.6%).

TABLE I: Assertion measurement for different noise levels created by simulation environment.

Circuit	Assertion Measurement for Different Noise Levels				
	0.01	**0.05**	**0.1**	**0.5**	**1**
CNOT	98.53%	94.62%	88.57%	57.61%	49.60%
CU1	98.92%	93.46%	87.98%	58%	51.95%
CH	74.90%	52.14%	51.46%	50.20%	49.51%
CX	98.43%	93.26%	88.86%	50.09%	49.51%
GHZ	91.02%	89.07%	65.78%	55.89%	45.34%
Adder	89.06%	66.60%	54.49%	51.90%	47.65%
Grover	81%	76.92%	50.58%	51.85%	48.92%
Shor's	84.87%	79.32%	76.23%	52.40%	48.33%

The concurrent assertion can be used as either a noise monitor or a run-time verifier. If used as a noise monitor, we can get the confidence level during execution. This can be used to get an idea of how noise affected our design during run time. If user needs to stop execution when the noise level is too high, the assertion circuit can be applied using a threshold. For example, if the threshold is set to 50% for the CNOT benchmark, the assertion will fail when the noise level is 1.

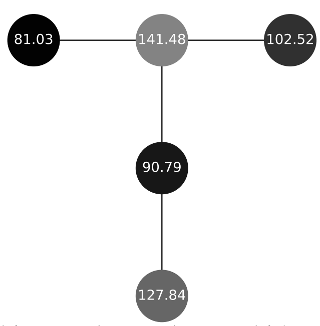
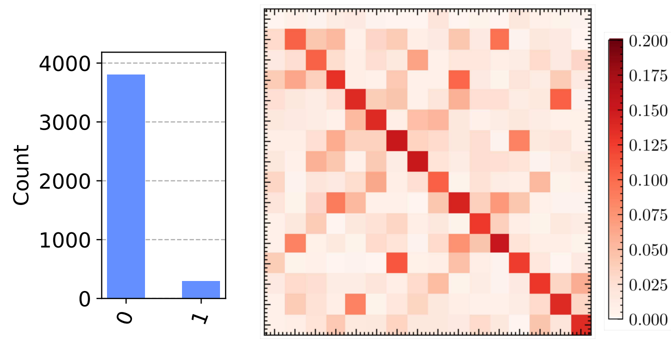


Fig. 7: IBM qubits topology where qubit's color is scaled by their T1 coherence time (ns).

We have run our experiment in IBM quantum computer to identify the quality of our concurrent assertion when presented with the actual noise of a quantum computer. Figure 7 shows the topology and T1 coherence time – the lifetime of a qubit to remain in $|1\rangle$ of the IBM quantum computer we used for the experiment. The qubit with the T1 coherence value 81.03 is the noisiest qubit and we mapped the assertion circuit to it. For real environments, some of the unrelated qubits which are used by other users can also increase the noise of the used qubits due to the topology. Our assertion circuit can be used as a monitor to check whether such noise is influencing the circuit throughout the design execution. Figure 8 compares the result of our QcAssert in comparison with a full characterization via quantum process tomography. As shown in the figure, the CNOT gate has some error and our assertion was able to monitor the noise deviations.

V. Conclusion

Quantum devices are susceptible to entangle with various sources of noise. When running a quantum circuit, the state can be perturbed resulting in error. As a practice, to identify



(a) SWAP test (b) CNOT process error

Fig. 8: Comparison of the (a) QcAssert output testing CNOT gate and (b) the error process $|\mathcal{R} - \mathcal{I}|$ of CNOT gate on *ibmq_quito*. $\mathcal{R}$ is the error of the CNOT extracted via process tomography in the Pauli basis, and $\mathcal{I}$ is the identity (no error.)

the noise levels in a device, users are running known circuits prior to running the actual circuit. However, this method does not guarantee how much the noise actually affects the circuit during run-time. In this paper, we proposed an assertion circuit ('QcAssert') that has the ability to run concurrently with the device and provide insight into the noise influence during run-time. QcAssert can be used as a mechanism to test the noise level of the quantum device during execution. Extensive experimental evaluation demonstrated our assertion's effectiveness in identifying the noise in a quantum device.

References

[1] Hasini Witharana et al. A Survey on Assertion-based Hardware Verification. *ACM Computing Surveys*, January 2022.
[2] Hasini Witharana, Yangdi Lyu, and Prabhat Mishra. Directed test generation for activation of security assertions in rtl models. *ACM Transactions on Design Automation of Electronic Systems*, 26(4), 2021.
[3] Hasini Witharana, Aruna Jayasena, Andrew Whigham, and Prabhat Mishra. Automated generation of security assertions for rtl models. *ACM Journal on Emerging Technologies in Computing Sys.*, 19(1), 2023.
[4] Huiyang Zhou et al. Quantum Circuits for Dynamic Runtime Assertions in Quantum Computation. *IEEE Computer Architecture Letters*, 2019.
[5] Y. Huang and M. Martonosi. Statistical Assertions for Validating Patterns and Finding Bugs in Quantum Programs. In *ISCA*, 2019.
[6] Gushu Li et al. Proq: Projection-based runtime assertions for debugging on a quantum computer. *arXiv preprint arXiv:1911.12855*, 2019.
[7] Mingsheng Ying et al. Invariants of quantum programs: characterisations and generation. *ACM SIGPLAN Notices*, 52(1):818–832, 2017.
[8] Daniel Volya and Prabhat Mishra. Quantum steering of surface error correcting codes. In *QCE*, 2023.
[9] Robert Rand et al. Qwire practice: Formal verification of quantum circuits in coq. *arXiv preprint arXiv:1803.00699*, 2018.
[10] Lukas Burgholzer, Richard Kueng, and Robert Wille. Random stimuli generation for the verification of quantum circuits. In *ASPDAC*, 2021.
[11] Daniel Volya and Prabhat Mishra. Quantum Spectral Clustering of Mixed Graphs. In *DAC*, pages 463–468, December 2021.
[12] N. Nguyen et al. Digital Quantum Simulation of the Schwinger Model and Symmetry Protection with Trapped Ions. *PRX Quantum*, 2022.
[13] Pedro Chamorro-Posada and Juan Carlos Garcia-Escartin. The switch test for discriminating quantum evolutions. *arXiv preprint arXiv:1706.06564*, 2017.
[14] Igor Jex et al. Antisymmetric multi-partite quantum states and their applications. *Fortschritte der Physik: Progress of Physics*, 2003.
[15] Mary L McHugh. The chi-square test of independence. *Biochemia medica*, 23(2):143–149, 2013.