Classification of Quantum Computer Fault Injection Attacks

Chuanqi Xu
Yale University
New Haven, CT, USA
chuanqi.xu@yale.edu

Ferhat Erata
Yale University
New Haven, CT, USA
ferhat.erata@yale.edu

Jakub Szefer
Yale University
New Haven, CT, USA
jakub.szefer@yale.edu

ABSTRACT

The rapid growth of interest in quantum computing has brought about the need to secure these powerful machines against a range of physical attacks. As qubit counts increase and quantum computers achieve higher levels of fidelity, their potential to execute novel algorithms and generate sensitive intellectual property becomes more promising. However, there is a significant gap in our understanding of the vulnerabilities these computers face in terms of security and privacy attacks. Among the potential threats are physical attacks, including those orchestrated by malicious insiders within data centers where the quantum computers are located, which could compromise the integrity of computations and resulting data. This paper presents an exploration of fault-injection attacks as one class of physical attacks on quantum computers. This work first introduces a classification of fault-injection attacks and strategies, including the domain of fault-injection attacks, the fault targets, and fault manifestations in quantum computers. The resulting classification highlights the potential threats that exist. By shedding light on the vulnerabilities of quantum computers to fault-injection attacks, this work contributes to the development of robust security measures for this emerging technology.

1 INTRODUCTION

Quantum computing has accelerated in development in recent years. Many companies and universities are racing to build bigger and better machines. Among others, IBM unveiled a 433 qubit quantum computer in late 2022. A 4158 qubit IBM quantum computer is projected for 2025, and most recently IBM has announced plans for 100,000 qubit quantum computers by 2033 [13].

Presently, quantum computers are in the Nosy Intermediate Scale Quantum (NISQ) regime [19], where there are only quantum computers with qubits less than 1000 and are not able to support quantum error correction [9] and practical applications like Shor's algorithm [24]. Nevertheless, these machines have the potential to help accelerate drug discovery or find new materials. As the machines grow in size and fidelity they may reach quantum supremacy, which refers to a point in time at which a quantum computer can perform calculations that significantly surpass the capabilities of even the most powerful classical computers. Already, some studies have shown quantum supremacy before fault-tolerant [4, 15], though such an assertion may be controversial at this point. With the increase of qubits and improvement in fidelity, it will be possible to gradually move into the fault-tolerant quantum computing regime with techniques like quantum error correction. Optimistically, quantum computers and quantum algorithms promise to be applied to revolutionize many fields, such as Grover's [11] and Shor's algorithms that can be used to break some nowadays widely-used classical cryptographic algorithms like RSA [22].

As quantum computers grow in size, the data and information in the computing process may be sensitive and private. Further, the quantum programs themselves executed on quantum computers are also valuable intellectual properties. Integrity and confidentiality of the data or quantum programs can be compromised if there is a fault-injection attack.

1.1 Differences from Classical Computer Fault Injection

In classical computer fault injection, the faults mainly target the instructions executing on the processor or the data in registers. It is also possible to inject or cause faults in DRAM memory or on the memory bus or other parts of the system. The classical processor is typically encased in a single package, and in fault-injection attacks, the package is exposed to voltage glitching, clock glitching, EM, lasers, or other sources of disturbance. Section 6 provides a brief list of existing works on fault injection in classical computers.

One main difference in quantum computers is the extensive classical infrastructure that controls the qubits within the quantum computers. This infrastructure significantly extends the possible attack surface. Also, given the current and projected physical size of quantum computers (the size of a server room or at least a few server racks), physical access and opportunity to manipulate the equipment is much larger than with today's nanometer-sized transistors in CPUs. Further, there is an opportunity for attackers to either manipulate the qubits, or classical registers into which the qubit measurements are read, or the control signals (either digital signals going into the controller equipment, or analog signals going between controller equipment and the quantum computer itself). This extended attack surface compared to classical computers analyzed in this work sheds light on the possibly new perspective in quantum computer fault injection attacks.

1.2 Contributions

The contributions of this work are:

- We identify the *domain of quantum computer fault injection attacks*; this domain represents the attack surface that is distinct from classical computers, and at the same time it identifies the hardware components that may be subject to the fault injection attacks.
- We pinpoint 3 *fault targets* specific to quantum computers: quantum processing units, quantum computer controller, and classical co-processors; within the three targets, we present further 6 specific components that can be targeted for fault attacks.
- We present *fault model*, *fault bound*, and *fault lifespan* for the different fault targets.

- We propose the first classification of quantum computer fault injection attacks to help industry and researchers navigate the security of this emerging technology.

2 BACKGROUND

This work focuses on superconducting quantum computers, such as those available from IBM, Rigetti, QCI, and others. The typical setup of a superconducting qubit quantum computer is shown in Figure 1. We consider today's cloud-based computers where users connect remotely to the machines. Figure 1 specifically depicts a superconducting qubit quantum computer setup. Other types of quantum computers may have different types of, for example, quantum computer controllers, but the same types of fault injection attacks can be applied.

2.1 Quantum Computing Basics

Analogous to the classical bit, a quantum bit, or *qubit*, is the fundamental computational unit in quantum computers. A qubit can be represented with the bra-ket representation. With $|0\rangle$ and $|1\rangle$ as the basis states, a qubit can be written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$. According to Born's rule, the results of measuring $|\psi\rangle$ is either $|0\rangle$ or $|1\rangle$, with probability $|\alpha|^2$ and $|\beta|^2$ respectively. Such a phenomenon that a qubit can be measured with two results is not seen in classical computing, and it is often called *superposition*. Also, the state after the measurement will *collapse* to the resulting state, no matter what the initial state is. Similarly, an n-qubit system is spanned by 2^n basis states. Surprisingly, some multi-qubit quantum states cannot be described independently by the state of their components, which is another phenomenon that is not shown in classical computing, and this is often referred to *entanglement*. Qubits are controlled and evolved by *quantum gates*, which are the building blocks of quantum circuits, like classical logic gates are for conventional digital circuits. We refer interested readers to [17] for details.

2.2 Cloud-based Access

Due to the expensive nature of quantum computing equipment, quantum computers are currently available as cloud-based systems. For example, cloud-based services such as IBM Quantum [14], Amazon Braket [3], and Azure Quantum [16] already provide access to Noisy Intermediate-Scale Quantum (NISQ) quantum computers remotely for users. In the cloud setting, the user has no control over the management server, quantum computer controllers, and the cryogenic fridge are not under the control of the user. A malicious insider or compromised cloud provider could try to perform fault injection attacks.

2.3 Users' Quantum Program

Quantum programs may be described using 'gates'. When using gate-level description, programs are composed of quantum gates. Quantum computers typically do not and it is also not necessary for them to support all kinds of quantum gates, because it is proved that any quantum gate can be approximated within a minor error using only a small number of quantum gates [7]. As a result, before running quantum circuits on a target computer, they are required to be transformed into quantum circuits that only contain the supported

quantum gates, usually called *native gates* or *basis gates*, in order to be compatible with target quantum computers. This process is often called *transpilation* or *compilation*. Besides to fulfill the native gates requirement, transpilers or compilers that do this step also need to satisfy other restrictions, such as that qubit connections need to be available on target quantum computers. Other processes such as optimizations may also be involved in transpilation.

In the end, quantum circuits only including native gates need to be further transformed into corresponding low-level operations and signals. This is a lower-level description of quantum circuits. For superconducting quantum computers, qubits are controlled by analog radiofrequency (RF) pulses, which are sent to the quantum computer. When using this *pulse-level* description, users directly specify the control pulses used to trigger the quantum gate operations. The pulses are very specific to each machine and typically need to be carefully designed to perform the desired operation. Advanced users and researchers may choose to use the pulse-level approach, rather than using the pre-defined native gates.

2.4 Management Server

The management server is a typical classical server that sits between the users and the quantum equipment. Management servers in the context of cloud computing oversee and control resources hosted on a cloud platform. Management servers for quantum computing commonly handle the receiving of quantum jobs, queuing, and dispatching jobs. Quantum jobs submitted by users are usually first pushed into priority queues, and based on the priority algorithms of the cloud platforms, these jobs wait in the queue, and then the information of jobs is processed and sent to quantum computer controllers after they finish waiting.

2.5 Classical Co-processor

A classical co-processor is a classical computer part of the quantum computer controller, or tightly coupled to the controller. The co-processor can perform classical computations based on the data readout from the quantum computer. It may contain user-defined code or application-specific code defining what operations to perform based on the readout data; as well as it can be used to determine what subsequent operations to execute on the quantum computer or to update the circuit executing on the quantum computer. In one example of quantum machine learning (QML) [6], based on the readout data, the co-processor can optimize the parameters of the quantum circuit and issue the next job with the updated circuit, similar to the classical machine learning.

2.6 Quantum Computer Controller

In current small-scale quantum processors, each qubit or qubit pair is typically assigned dedicated control pulses with distinct parameter settings, including the pulse waveform, pulse duration, pulse frequency, pulse amplitude, and so on. Control pulses, both microwave and baseband flux, are generated at room temperature by classical equipment such as the arbitrary waveform generator (AWG) and IQ mixers. Then these pulses will be delivered to the qubits in the cryogenic system through a series of attenuators and filters designed to suppress harmful noises when the quantum programs reach the point to run the corresponding gates.
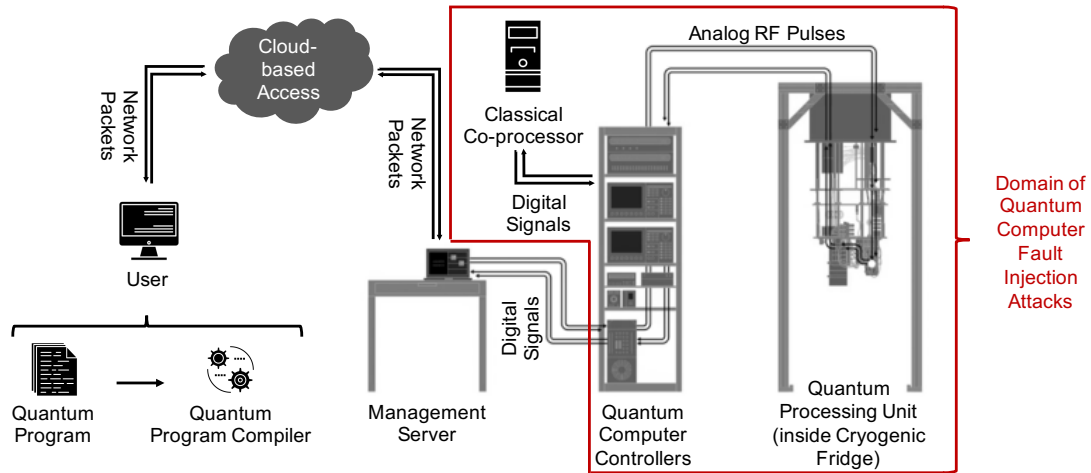
Figure 1: Typical setup of a superconducting qubit quantum computer, figure is based on figure provided by IBM.
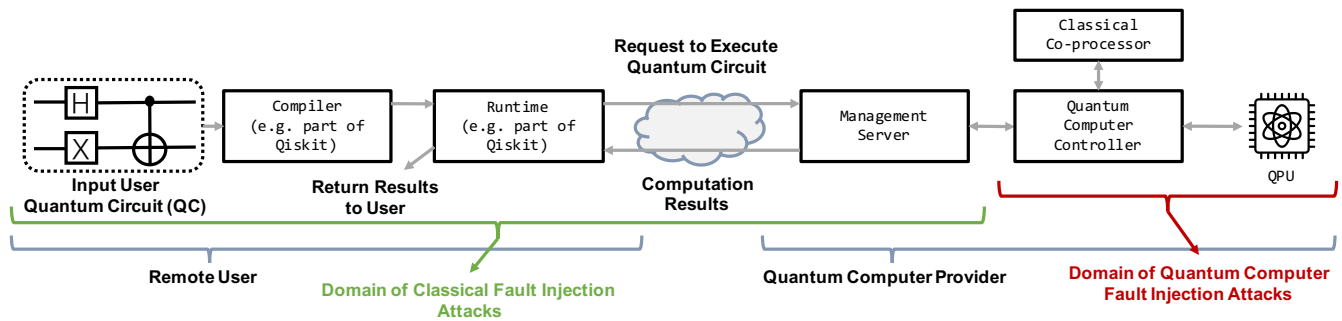

Figure 2: Typical quantum computer workflow.

Besides controlling the qubits, one important function of quantum computer controllers is to perform the measurement process and measurement readout results. The results from quantum computers may be stored in the controller and sent back to the management servers when jobs finish. In addition, for advanced features like dynamic circuits [12], it stores the middle-measurement results and controls future operations based on these results.

In the end, the qubit states are usually measured to get the job results. In superconducting quantum computers, measurement equipment often includes cryogenic amplifiers and analog-to-digital converters. These tools discern the quantum state of the qubits by monitoring microwave signals. The qubit-induced changes in these signals are amplified and converted into digital information, making quantum data readable to classical computers. Then this data is sent back to the quantum computer controllers, which then send it to the management server; and the users finally are forwarded the measurement data.
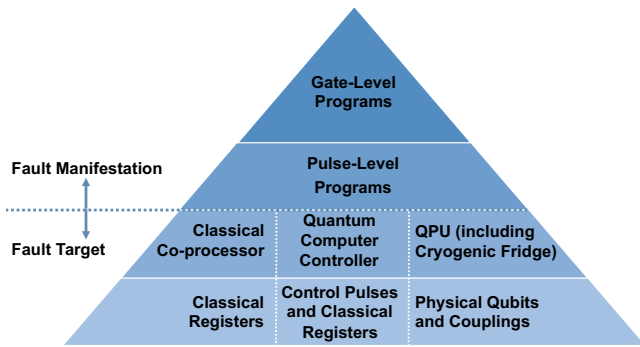
2.7 Quantum Processing Unit including Cryogenic Fridges

The Quantum Processing Unit (QPU) contains the actual physical qubits. The QPU is located in the cryogenic fridge, also known as the dilution refrigerator, which is an integral part of superconducting quantum computers. These qubits are sensitive to thermal noise, which is why the frigid environment provided by the dilution refrigerator is crucial. Once the qubits are in their superconducting state, they are manipulated using microwave pulses, generated by quantum computer controllers previously introduced. The pulses are delivered through coaxial cables that are also cooled within the refrigerator to minimize thermal noise.

2.8 Workflow of Executing Quantum Circuits on a Quantum Computer

The typical workflow of quantum computers is shown in Figure 2. In quantum computing, users can write gate-level programs using quantum programming languages such as Qiskit [20], Amazon Braket SDK [2], or Cirq [8]. These programs consist of sequences of quantum gates that operate on qubits. The programs are then transpiled to decompose the gates into elementary quantum gates supported by the hardware. The transpiler optimizes the program by reducing gate count and improving gate ordering. It also maps logical qubits to the physical qubits available in the hardware, considering connectivity constraints. The next step is *scheduling*, where

Figure 3: The fault target and fault manifestation security pyramid for superconducting quantum computers.

timing and control information are determined for each gate, specifying the precise microwave pulses required for their execution. When jobs are sent to quantum computer systems and start to execute, microwave electronics generate these pulses, corresponding to signals that manipulate the quantum state of the qubits. The pulses are applied to the physical qubits, implementing the desired gate operations. After execution, the resulting quantum state can be measured to obtain the computation's output. The specific details of the transpilation and scheduling process may vary depending on the programming language, hardware, and software stack used.

3 FAULT MANIFESTATION

In the context of fault injection in quantum computing, *fault manifestation* refers to the observable effect or consequence of an injected fault within the quantum system. This could include changes in the state of a qubit, alterations in the operation of a quantum gate, or eventually deviations in the outcome of a quantum algorithm. The study of fault manifestation is crucial in understanding the impact of errors on quantum computations and in developing strategies for error detection and correction. The fault manifestation can be:

3.1 Gate-level Program

The gate-level quantum circuit is a model used in quantum computing to describe qubit evolution and incidental operations. Computations in the gate-level quantum circuits are represented as a sequence of quantum gates acting on qubits, and other operations such as measurement, reset, and classical operations, from left to right to denote time steps. Each quantum gate, analogous to a logic gate in classical computing, performs a specific unitary operation or transformation on the quantum state of a qubit or a set of qubits. By arranging these gates in specific sequences and combinations, complex quantum algorithms can be implemented. This gate-level description is particularly useful for visualizing, designing, and analyzing quantum computations.

3.2 Pulse-level Program

This is one level lower abstraction of quantum circuits. Since superconducting qubits are controlled by microwave pulses, the exact physical actions of quantum gates and other operations in gate-level circuits are correspondingly predefined microwave pulses.

The pulse parameters such as frequency and amplitude are continuously changing due to the fluctuations in the environment and qubits. Therefore, the pulse parameters are frequently calibrated to reach high fidelity to the desired logic operations specified by the corresponding quantum gates. A pulse-level description provides a more granular view of quantum computation compared to the gate-level representation. It accounts for the physical implementation of quantum gates, offering insights into the precise control mechanisms and potential sources of error in quantum operations.

4 FAULT TARGET

Faults can occur or be injected at various locations or types of equipment within the quantum computing system. We focus here on the components within the domain of quantum computer fault injection attacks, defined in Figure 1: Quantum Processing Unit, Quantum Computer Controller, and Classical Co-processor.

4.1 Classical Co-processor

The classical co-processor is a classical equipment made of traditional digital computing hardware and peripherals, which is introduced in Section 2.6. The classical co-processor may be used in conjunction with the quantum computer. For example, in quantum machine learning (QML), there is an iterative process of running a circuit on a quantum computer, optimizing the circuit on a classical computer based on results, running it again on a quantum computer with updated parameters, etc.

4.1.1 Faults in Classical Registers. Within the classical co-processor are of course the usual components such as ALU, registers, or memory, among others.[1] Faults can be injected in these classical components to, for example, affect the computations used in QML optimization routines between executions of a circuit on a quantum computer. For program specification at the gate-level, the faults can result in gates being added, removed, or modified by changing the digital bits that specify them in the program. For program specification at the pulse-level, the faults can affect the digital specification of the amplitude, duration, or phase of the control pulses to be generated.

4.2 Quantum Computer Controller

Quantum computer controller is typically made of equipment to generate microwave pulses to manipulate qubit states, and measurement equipment to translate quantum information into a classical format which is stored as the readout data.

4.2.1 Faults in Control Pulses. Faults can be injected into the control pulses generated by the quantum computer controller, for example, through EM radiation that affects the pulses generated by the controller, or more directly by affecting the operation of the controller itself causing it to generate wrong or modified pulses. Readout data is the classical data resulting from the measurements. Faults can also be injected into the readout control pulses through

---

[1]For simplicity, we specify the fault target here as "classical registers", but the physical faults could also be in ALU, memory, or other components. Since the faults will eventually occur in or enter registers, we use the simplification of calling the target just "classical registers".

EM, for example, or the readout data can itself be directly manipulated through faults in digital registers storing the data within the controller.

Faults can occur or be induced during the application of the control pulses that manipulate the quantum state of the qubits. In particular, faults can arise or be induced in the control electronics that generate and apply microwave pulses to manipulate the qubits. Issues with signal generation, calibration, timing, or stability of control signals can impact the accuracy of gate operations. Faults can affect unitary and non-unitary operations:

- *Unitary Operations* – Unitary operations refer to transformations that preserve the normalization and reversibility of quantum states. Quantum gates are unitary gates, and unitary operations are the typical computational operations on the qubits, such as different X, SX, CX, or other gates.
- *Non-Unitary Operations* – Non-unitary operations are all other operations. For instance, reset or measurement are not unitary, because they collapse the state of the qubits during the execution of the operation.

4.2.2 Faults in Classical Registers. Non-unitary operations such as reset or measurement utilize classical registers. In particular, when qubits are measured, the quantum state collapses to one of the eigenstates of the measurement, and the measurement result is stored in classical registers or memories inside the control electronics. The classical registers then can be victims of fault injection that affects the classical bits. These faults can affect operations in which the classical registers participate, such as mid-circuit measurement, and final measurement:

- *Mid-Circuit Measurement* – Mid-circuit measurement allows for measuring the qubit state in the middle of the execution. The results can then be used to determine what code to execute by analyzing the classical bit measurement results. If the classical bit is modified, the circuit execution can be affected, as the classical bit at each mid-circuit measurement determines the next set of operations that will be applied. One example is the reset instruction, where a measurement is performed to measure the qubit state, and if the state is measured to be 1 instead of 0, an X gate is applied to flip the qubit back to $|0\rangle$. If the classical register is modified, then the reset will reset to the other state.
- *Final Measurement* – The final measurement is performed at the end of each circuit. Usually, all qubits are measured, though sometimes ancilla qubits may not be measured. Injecting fault into the classical bits at this stage is effectively equivalent to manipulating the final circuit output.

4.3 Quantum Processing Unit

The quantum processing unit (QPU) refers to the hardware components in a superconducting quantum computer that operate based on the principles of quantum mechanics. This includes the QPU which implements qubits, such as the Josephson junction widely used to realize superconducting qubits.

4.3.1 Faults in Physical Qubits or Couplings. There are many ways to influence and thus inject faults into the qubits. For instance, superconducting qubits are susceptible to decoherence, which refers to the loss of coherence and information due to interactions with the environment. External noise sources, such as thermal fluctuations or electromagnetic radiation, can cause qubits to lose their quantum states and result in errors. Faults can be injected through external means such as EM radiation or thermal changes to the fridge holding the qubits.

5 CLASSIFICATION

Our classification of quantum computer fault injection attacks is now presented in this section. The classification is presented in Figure 4 and detailed below.[2]

5.1 Fault Targets

In the classification, we separate the three targets into six specific components vulnerable to faults and list them in more detail below.

Quantum Processing Unit:

- Target: *Qubits* are typically physical, two-level quantum-mechanical systems. A common type of qubit is built from a Josephson junction (but many others exist). As physical systems, they can be impacted by voltage changes, EM radiation, etc., that attackers can generate.
- Target: *Couplings* are typically intermediate electrical circuits used to connect qubits, they can be likewise impacted by voltage changes, EM radiation, etc., that attackers can generate.

Quantum Computer Controller:

- Target: *Control Pules (Analog RF Signals)* are often microwave pulses sent to an antenna or transmission line coupled to the qubit with a frequency resonant with that qubit to realize an operation. The attacker can interfere with or modify analog properties of the signals to induce faults in the qubits or gate operations, e.g., by changing the frequency, phase, or envelope.
- Target: *Control Pulses (Digital Specification)* are generated by arbitrary waveform generators from digital specification, e.g. by an FPGA. The attacker performs attacks on classical bits or classical operations that read, modify, or write the digital information, thus resulting in wrong pulses being sent.
- Target: *Classical Registers* are used, for example, to store measurement readout information during mid-circuit or final measurement. In particular, the mid-circuit measurement may be used to determine subsequent operations in dynamic circuits [12]. The attacker can induce faults in these classical registers.

Classical Co-processor:

- Target: *Classical Registers* are also used in classical co-processors used to perform computations on the output. For example in quantum machine learning (QML), parts of the input circuit are optimized based on the results of computation,

---

[2]The terminology used in this section focuses on superconducting qubit machines, but this classification can be equally applied to other types of quantum computers by replacing certain terms. For example, control microwave pulses can be replaced by laser pulses if ion-trap computers are considered.
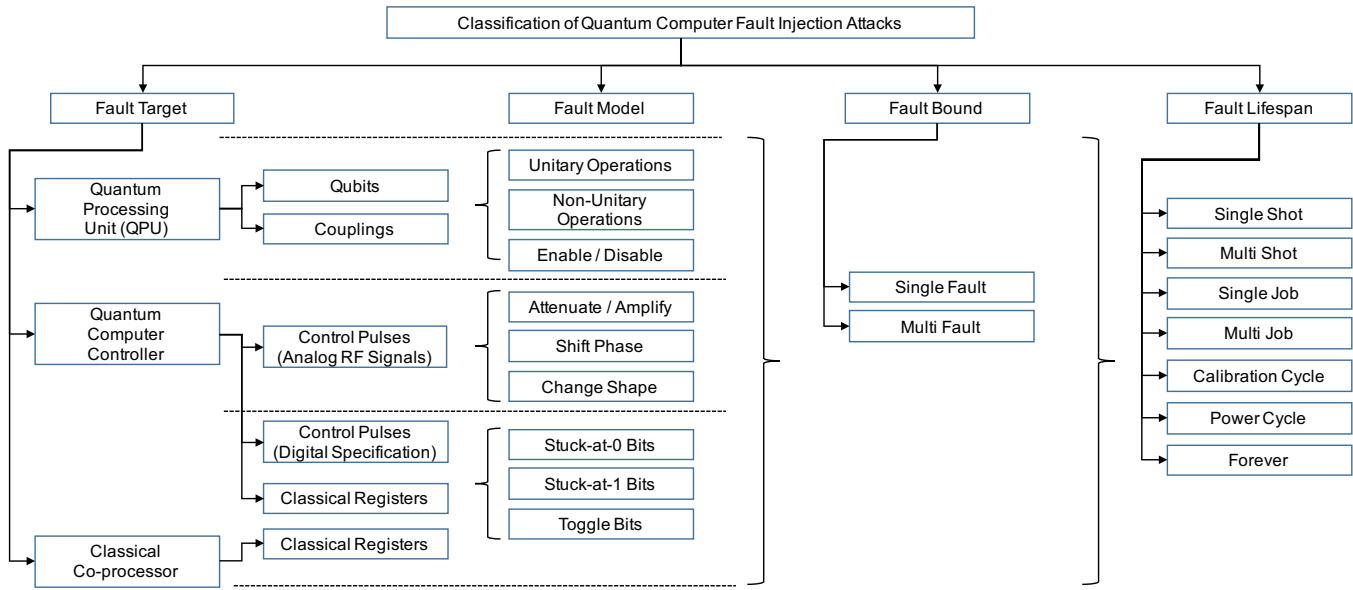
Figure 4: Classification of quantum computer fault injection attacks.

and the circuit is run again. The attacker can induce faults in these classical registers.

5.2 Fault Model

The fault model is a theoretical representation or framework that predicts or describes the types of faults that may occur in a system, their causes, and their potential effects. We have three fault models, corresponding to different targets.

Quantum Processing Unit: The qubits and couplings are vulnerable to three types of novel faults not found in classical computers: Faults can result in unitary type operations, which are effectively faults inducing a change in qubit state that can be reversed like any other (non-malicious) unitary gate. Faults can result in non-unitary operations, which are usually hard to reverse. Faults can result in enabling / disabling of qubits or couplings, which may be similar to instruction skip faults in classical computers if a coupling is disabled, for example.

Quantum Computer Controller: The analog control pulses are also vulnerable to novel types of faults not found in classical computers: Faults can attenuate / amplify the analog pulses, causing different gate operations to be effectively performed. Faults can also shift the phase of the pulses, likewise resulting in different gate operations being effectively performed. The faults can also change the shape of the envelope of the pulse, again changing the gate operation performed. If the pulses are attenuated or otherwise sufficiently distorted, a gate operation may be effectively disabled. Conversely, amplifying or otherwise injecting an analog signal can create or insert a gate operation not part of the original circuit.

Quantum Computer Controller and Classical Co-processor: The controller and the co-processor also contain digital classical information, specifying the pulses (before they are generated as analog microwave signals) and other registers. These are vulnerable to well-known stuck-at faults or bit toggling faults.

5.3 Fault Bound

The fault bound is a limit or threshold that defines the maximum number of faults that a system can tolerate without significant degradation in its performance or functionality. Regardless of the fault target, there is either a single or multiple fault threat.

5.4 Fault Lifespan

The fault lifespan refers to the duration for which a fault persists in a system. For quantum systems, this could refer to the period during which a qubit remains in an erroneous state before it is corrected or resets to its original state. In quantum computers, there are many more different lifespans compared to classical computers.

- Single Shot – each circuit is divided into one or more shots that are executed on a quantum computer; most short-lived faults would affect single shots. Most faults on analog pulses would fit in this category.
- Multi Shot – faults can persist through the execution of multiple shots of a circuit. Modification of the digital specification of the pulses would fit in this category.
- Single Job – multi-shot faults that last for all shots of a circuit would be Single Job faults.
- Multi Job – faults across multiple jobs of the same or different users would be multi-job faults. Faults in classical co-processor registers could fit in this category.
- Calibration Cycle – each quantum computer is calibrated frequently. Calibration can correct for changes in the environment or noise. Unitary operation-type faults in qubits could be in this category.
- Power Cycle – periodically, a quantum computer fridge has to be warmed up to replace or modify hardware, this is effectively a power cycle. Changes to control pulses which cause rapid heating and then cooling of the qubits could result in flux trapping, requiring power cycling the fridge.

- Forever – faults that permanently alter the hardware would be faults that last forever. Disable faults on couplings could fit in this category.

6 RELATED WORK

There are only a few studies on fault injection attacks in quantum computers. Most of them are based on the hardware-induced faults in qubits [1, 18, 25]. Therefore, we drew inspiration from the fault injection literature in classical computing. Our decomposition includes Fault Target, Fault Model, Fault Bound, and Lifespan [5, 23, 26]. However, our classification represents the attack surface that is distinct from classical computers and, at the same time, identifies the hardware components that may be subject to fault injection attacks in quantum computers.

Giraud et al. [10] classify fault injection attacks in classical computing as transient vs. permanent and invasive vs. non-invasive. However, for our study, we focused solely on non-invasive attacks and classified them as transient or permanent under the Fault Lifespan category. In a recent study by Ravi et al. [21], fault injection attacks specific to post-quantum cryptography algorithms (Kyber and Dilithium) were classified. This attack-specific classification examined characteristics such as the need for profiling, the number of required traces, and the ability to observe or communicate with the victim device. Baksi et al. [5] conducted a survey on fault attacks on symmetric key cryptosystems, consolidating existing attacks under fault models, data alteration methods, sources of fault injection, and analysis methods. Although our fault injection attack classification for quantum computers shares similar categories, it is tailored specifically to the quantum computing domain. Notably, our classification does not cover fault injection analysis methods, as no such methods have been reported in the literature for quantum computers.

Furthermore, the fault target and fault manifestation security pyramid for superconducting quantum computers (Figure 3) presented in our work is the quantum computing counterpart of the one introduced by Verbauwhede et al. [26].

7 CONCLUSION

This paper presented the first classification of fault-injection attacks on quantum computers. This work first introduced the domain of quantum computer fault injection attacks. It then proceeded to present fault targets and fault manifestations for quantum computers. The resulting classification also specifies fault models unique to quantum computers, along with fault bounds and fault lifespans that should be considered. By shedding light on the vulnerabilities of quantum computers to fault-injection attacks, this work contributes to the development of secure quantum computer systems.

ACKNOWLEDGMENTS

REFERENCES

[1] 2021. Exponential suppression of bit or phase errors with cyclic error correction. *Nature* 595, 7867 (2021), 383–387.

[2] Amazon Braket SDK. 2023. https://docs.aws.amazon.com/braket/latest/developerguide/api-and-sdk-reference.html.

[3] Amazon Web Services. 2023. Amazon Braket. https://aws.amazon.com/braket/

[4] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[5] Anubhab Baksi, Shivam Bhasin, Jakub Breier, Dirmanto Jap, and Dhiman Saha. 2022. A survey on fault attacks on symmetric key cryptosystems. *Comput. Surveys* 55, 4 (2022), 1–34.

[6] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. *Nature* 549, 7671 (2017), 195–202.

[7] David Elieser Deutsch, Adriano Barenco, and Artur Ekert. 1995. Universality in quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 449, 1937 (1995), 669–677. https://doi.org/10.1098/rspa.1995.0065 arXiv:https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1995.0065

[8] Cirq Developers. 2022. Cirq. (Dec 2022). https://doi.org/10.5281/zenodo.7465577 See full list of authors on Github: https://github.com/quantumlib/Cirq/graphs/contributors.

[9] Simon J Devitt, William J Munro, and Kae Nemoto. 2013. Quantum error correction for beginners. *Reports on Progress in Physics* 76, 7 (jun 2013), 076001. https://doi.org/10.1088/0034-4885/76/7/076001

[10] Christophe Giraud and Hugues Thiebeauld. 2004. A survey on fault attacks. In *Smart Card Research and Advanced Applications VI: IFIP 18th World Computer Congress TC8/WG8. 8 & TC11/WG11. 2 Sixth International Conference on Smart Card Research and Advanced Applications (CARDIS) 22–27 August 2004 Toulouse, France*. Springer, 159–176.

[11] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Philadelphia, Pennsylvania, USA) *(STOC '96)*. Association for Computing Machinery, New York, NY, USA, 212–219. https://doi.org/10.1145/237814.237866

[12] IBM. 2022. Bringing the full power of dynamic circuits to Qiskit Runtime. https://research.ibm.com/blog/quantum-dynamic-circuits

[13] IBM. 2023. Charting the course to 100,000 qubits. https://research.ibm.com/blog/100k-qubit-supercomputer

[14] IBM Quantum. 2023. https://quantum-computing.ibm.com/

[15] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout Van Den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. 2023. Evidence for the utility of quantum computing before fault tolerance. *Nature* 618, 7965 (2023), 500–505.

[16] Microsoft Azure. 2023. Azure Quantum. https://azure.microsoft.com/en-us/products/quantum

[17] Michael A Nielsen and Isaac L Chuang. 2001. Quantum computation and quantum information. *Phys. Today* 54, 2 (2001), 60.

[18] D. Oliveira, E. Giusto, E. Dri, N. Casciola, B. Baheri, Q. Guan, B. Montrucchio, and P. Rech. 2022. QuFI: a Quantum Fault Injector to Measure the Reliability of Qubits and Quantum Circuits. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE Computer Society, Los Alamitos, CA, USA, 137–149. https://doi.org/10.1109/DSN53405.2022.00025

[19] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (Aug. 2018), 79. https://doi.org/10.22331/q-2018-08-06-79

[20] Qiskit contributors. 2023. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505

[21] Prasanna Ravi, Anupam Chattopadhyay, Jan Pieter D'Anvers, and Anubhab Baksi. 2022. Side-channel and fault-injection attacks over lattice-based post-quantum schemes (Kyber, Dilithium): Survey and new results. *ACM Transactions on Embedded Computing Systems* (2022).

[22] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* 21, 2 (feb 1978), 120–126. https://doi.org/10.1145/359340.359342

[23] Carlton Shepherd, Konstantinos Markantonakis, Nico van Heijningen, Driss Aboulkassimi, Clément Gaine, Thibaut Heckmann, and David Naccache. 2021. Physical fault injection and side-channel attacks on mobile devices: A comprehensive analysis. *Computers & Security* 111 (2021), 102471.

[24] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (1997), 1484–1509. https://doi.org/10.1137/S0097539795293172 arXiv:https://doi.org/10.1137/S0097539795293172

[25] Antti P Vepsäläinen, Amir H Karamlou, John L Orrell, Akshunna S Dogra, Ben Loer, Francisca Vasconcelos, David K Kim, Alexander J Melville, Bethany M Niedzielski, Jonilyn L Yoder, et al. 2020. Impact of ionizing radiation on superconducting qubit coherence. *Nature* 584, 7822 (2020), 551–556.

[26] Ingrid Verbauwhede, Dusko Karaklajic, and Jorn-Marc Schmidt. 2011. The fault attack jungle-a classification model to guide you. In *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 3–8.