Exploration of Timing and Higher-Energy Attacks on Quantum Random Access Memory

Yizhuo Tan
Yale University
New Haven, Connecticut, USA

Chuanqi Xu
Yale University
New Haven, Connecticut, USA

Jakub Szefer
Yale University
New Haven, Connecticut, USA

ABSTRACT

This work presents the first evaluation of timing and higher-energy attacks on Quantum Random Access Memory (QRAM) circuits. By leveraging quantum principles, QRAM can efficiently store and manipulate both quantum and classical data, leading to potential significant speedups in a variety of quantum algorithms. However, as demonstrated in this work, when used in remote cloud-based quantum computers QRAM is vulnerable to different security attacks. The work demonstrates side-channel attacks, e.g., the timing attacks, as well as fault-injection-like attacks, e.g., the higher-energy attacks. This work evaluates the attacks on QRAM and different circuits that use QRAM. The work also proposes a set of defenses.

CCS CONCEPTS

• **Security and privacy** → **Security in hardware**; • **Hardware** → **Quantum technologies**.

KEYWORDS

Quantum Random Access Memory, Security, Higher-Energy Attacks, Timing Attacks

ACM Reference Format:
Yizhuo Tan, Chuanqi Xu, and Jakub Szefer. 2024. Exploration of Timing and Higher-Energy Attacks on Quantum Random Access Memory. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '24), October 27–31, 2024, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3676536.3689916

1 INTRODUCTION

Developments in quantum computing have accelerated rapidly in recent years. Many research labs, universities, and companies are racing to build bigger and more powerful quantum computers. Among others, IBM recently unveiled a 1121-qubit quantum computer in 2023, and 200-qubit IBM quantum computers with the ability to run 100 million gates are anticipated for 2029 [17]. Larger and larger quantum computers with improved fidelity promise to enable novel types of computation that are not possible with classical computers. Currently, quantum computers are in the Nosy Intermediate Scale Quantum (NISQ) regime [22], with less than 1000 qubits and no support for quantum error correction [10]. However, advances in error-corrected quantum computers have been demonstrated [7, 23] and we are on the cusp of having error-corrected quantum computers being deployed.

In parallel with work on quantum computation, research has advanced in the exploration of quantum memories and quantum databases. In particular, Quantum Random Access Memory (QRAM) holds the promise of transforming the field of quantum computing. By leveraging quantum principles, it can efficiently store and manipulate both quantum and classical data, leading to significant speedups in a variety of computational tasks.

Due to the expensive nature of quantum computers and their equipment, quantum computers are currently available as cloud-based systems. For example, cloud-based services such as IBM Quantum [16], Amazon Braket [1], and Azure Quantum [21] already provide access to Noisy Intermediate-Scale Quantum (NISQ) quantum computers remotely for users. In the future, these companies will provide access to error-corrected computers, as well as, to QRAM and other quantum data storage technologies.

The cloud setting in general allows for easy, on-demand access to computing resources, but in the case of quantum computing, it especially enables various users to connect to the expensive quantum computers which most users cannot even purchase themselves today. Although there are benefits to cloud-based classical and quantum computing, in the cloud setting, the user has no control over the classical servers nor quantum computers and their related equipment, such as quantum computer controllers or the cryogenic fridge. Also, users do not have control over other users who share the same equipment and computers.

Already, a number of security attacks have been demonstrated against quantum computers, especially against cloud-based quantum computers that are vulnerable to untrusted cloud providers and malicious co-tenants. Researchers have proposed reset attacks [20, 25], side-channel attacks [5, 11, 19, 28], higher-energy state attacks [27], and crosstalk attacks [2, 3, 8, 9]. All existing attacks focus on the computational part of the quantum algorithms.

Meanwhile, this work explores QRAM, or data storage, part. In particular, this work addresses the research gap in understanding potential security attacks on QRAM in a shared computing setting where different users share the quantum computer. We consider two types of attacks. Timing attacks are ones where the adversary is able to measure the execution time of the victim – here the execution time of how long it takes to run the quantum circuit and query the QRAM. The timing attacks are passive, as the adversary only measures time. We also consider higher-energy attacks [27]. The higher-energy attacks are ones where the adversary is able to set the qubits into higher energy states, such as $|2\rangle$ or $|3\rangle$, which causes superconducting quantum computer gates on these qubits not to operate correctly later [27]. The higher-energy state attacks

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICCAD '24, October 27–31, 2024, New York, NY, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1077-3/24/10.
https://doi.org/10.1145/3676536.3689916

are active, as they require the adversary to set qubits into specific higher-energy tates.

1.1 Paper Contribution

The contributions of the paper are as follows:

- Demonstration of the ability of an adversary to guess QRAM size from timing information.
- Demonstration of the ability of an adversary to guess the quantum circuit being used by the victim by leveraging timing information about the execution of the circuit and QRAM.
- Demonstration of higher-energy attacks on QRAM.
- Proposal and evaluation of defenses based for the timing and higher-energy attacks.

2 BACKGROUND

This section provides a brief background on Quantum Random Access Memory (QRAM). This section overviews the basic elements of QRAM and details two specific QRAM implementations: Fanout QRAM [12] and Bucket-Brigate QRAM [13].

2.1 QRAM

QRAM holds significant promise for executing comprehensive quantum queries and can be conceptualized as a generation of classical RAM. Classical RAM takes address i as the input to access the corresponding memory cell x_i. Similarly, QRAM operates by receiving a quantum superposition of various addresses $|\psi_{in}\rangle$ as input, and subsequently yields an entangled state $|\psi_{out}\rangle$ where each address is correlated with its corresponding memory element,

$$|\psi_{in}\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle_{\text{ad}} |0\rangle_{\text{bus}} \xrightarrow{\text{QRAM}} |\psi_{out}\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle_{\text{ad}} |x_i\rangle_{\text{bus}} \tag{1}$$

where N is the size of the data vector x, α_i is the amplitude of each address in the superposition, $|\cdot\rangle_{\text{ad}}$ ($|\cdot\rangle_{\text{bus}}$) is the address (bus) qubit.

2.2 Quantum Router

One fundamental component of QRAM is the quantum router. As illustrated in Figure 1, the incident qubit $|i\rangle$ is directed either to the left or the right conditioned on the state $|r\rangle$ of this router. Specifically, when $|r\rangle = |0\rangle$, the incident qubit is routed to the left; when $|r\rangle = |1\rangle$, it is guided rightward. Figure 2 shows the quantum circuit of such a router, employing two CSWAP gates.

2.3 Fanout QRAM

Fanout QRAM constructed utilizing quantum routers is the first architecture to achieve an $O(logN)$-latency query, as depicted in Figure 3. These routers are organized in a binary tree structure, where the outputs of routers at one level cascade into the routers at the subsequent level below. Memory resides at the tree's bottom, with each of the N memory cells linked to a router at the base level. To initiate a query, the bus qubit and all routers are set to $|0\rangle$ state. The process has two stages: address loading and data retrieval.

During the address loading stage, a sequence of CX gates are implemented to entangle the input address with the routers and to distribute address qubits as multiple Greenberger–Horne–Zeilinger (GHZ) states. If the l-th address qubit is $|1\rangle$, all the routers at the level of l will be flipped into $|1\rangle$ state.

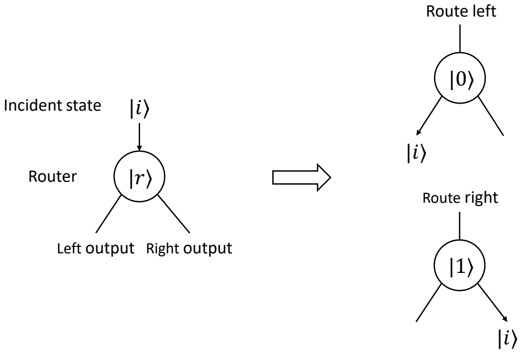

Figure 1: Example of a quantum router, the router qubit controls whether the incident state is propagated to the left or right output.

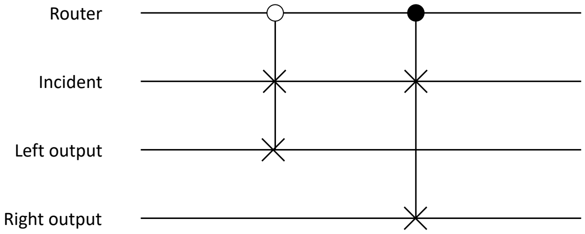

Figure 2: Example of a quantum router circuit. The two CSWAP gates operate based on different states of the router qubit. If the router qubit is $|0\rangle$, then the circuit will swap the incident qubit with the left output; otherwise, the incident qubit will be routed to the right.

In the data retrieval stage, the bus qubit is injected into the top node of the binary tree, as shown in Figure 3. Following the path outlined by the routers, the bus qubit finally reaches the bottom of this binary tree. Here, some classically-controlled Z gates copy the desired superposition of classical information $x_i = f(i)$ into the quantum state of the bus qubit. Upon completion of these two stages, the bus qubit retraces its path back to the top of the tree. All the GHZ states of address qubits are reverted to $|0\rangle$ through uncomputation.

2.4 Bucket-Brigade QRAM

The Bucket-Brigade QRAM, or BB QRAM for short, is a variant of the Fanout QRAM. In contrast to the Fanout QRAM's use of CX gates, the Bucket-Brigade QRAM swaps both the address and bus qubits into the QRAM. In this architecture shown in Figure 4, each router features an additional state denoted as $|W\rangle$ state, alongside the $|0\rangle$ and $|1\rangle$ states. Initially, all routers are set to the $|W\rangle$ state, indicating their inactive status. When the state of the router is $|W\rangle$, the action of the routing operation is trivial. When an address qubit encounters a $|W\rangle$ state router, the states of the address qubit and the router are swapped, which makes the address qubits route themselves into the binary tree in the address loading step. Consequently, this

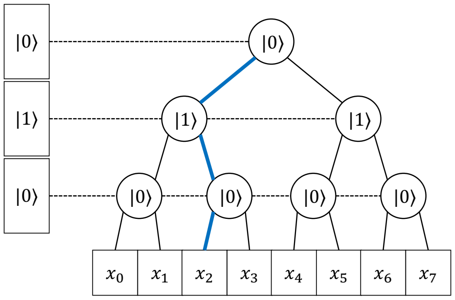

Figure 3: Fanout QRAM with 3 address qubits can store $2^3 = 8$ values.

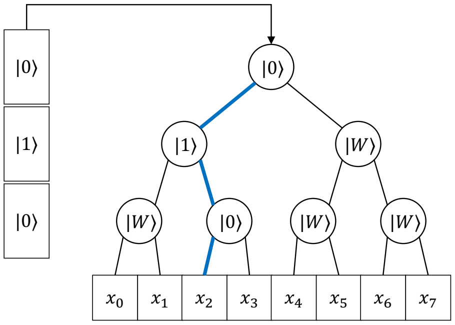

Figure 4: Bucket-Brigade QRAM with 3 address qubits can store $2^3 = 8$ values.

reduction in entanglement entropy compared to GHZ states in a Fanout architecture significantly mitigates the impact of errors within the circuit while maintaining latency $O(logN)$.

3 THREAT MODEL

This work considers the threat model of an honest-but-curious cloud provider, who is giving access to quantum computing resources to various users.

For the cloud provider, we assume that the provider can obtain timing information from the execution of the circuits. The threats we consider also include malicious insiders within the cloud provider. The insiders may not have access to the circuit submitted to the cloud provider, but may have enough access to controllers or other equipment and can measure the execution time of the user's circuits. With the timing information, the cloud provider or the malicious insiders can recover some details about the circuits or the QRAM as we demonstrate later in the paper.[1]

For the users, we assume they are mutually distrusting and that some users may attempt to attack other users. We assume the users have the ability to execute custom pulses on the quantum computer, necessary for the higher-energy attacks [27]. We also consider that users may have ways to measure the execution time of other users, although this has not yet been demonstrated in practice.

Table 1: Access duration for different sizes and types of QRAM. The time unit is the backend's cycle time, dt. The hardware used in this experiment is 127-qubit $IBM_{b}risbane$ machine. hlPlease specify which quantum computer and which qubits you are using

Size	Fanout QRAM Access Duration	BB QRAM Access Duration
1 addr. qubit	84560	84560
2 addr. qubit	185240	289680
3 addr. qubit	380120	596280

4 TIMING ATTACKS

Timing attacks exploit the time it takes for a system to perform certain operations to infer sensitive information [24]. For example, in classical computers performing cryptographic operations, variations in decryption time can reveal aspects of the private key. An attacker can measure how long different decryption attempts take and use this timing information to reconstruct the key [6]. In this section, in the context of quantum computers and QRAM, we explore what information could be learned from timing. We find that the most basic information that attackers can learn is the algorithm being used by the user.

If an attacker can monitor the timing of the activity of specific qubits, it could lead to information leakage concerning the quantum circuit. This is because different algorithms and QRAMs exhibit distinct running times, which could inadvertently reveal sensitive details about the operations being executed. In this section, we show we are able to measure the estimated duration of the algorithm portion and QRAM portion of the quantum circuit. From this timing information, the attacker can learn about which one of the n algorithms the victim executed, or the size of the QRAM.

4.1 Determining QRAM Size from Timing

As the QRAM size increases (i.e. there is a bigger number of address qubits), there are more routers used in the QRAM and the depth of the QRAM increases. As a result, it takes longer to query the QRAM as the size increases. Indeed, this is demonstrated by our measurements shown in Table 1. The experiments were performed with 1, 2, and 3 qubit QRAM on IBM quantum computer $IBM-brisbane$. Due to the limited size and qubit connections of the quantum computers we have access to, larger QRAMs could not be tested.

With the knowledge of the QRAM type used and the access duration, the attacker can thus guess the size of the QRAM and learn the size of the data set being used by the victim. Conversely, with knowledge of the access duration and the size of the QRAM, the attacker can learn the type of QRAM being used. This can enable further attacks that may be specific to the QRAM type and timing can reveal the QRAM type. It should be noted that the access time is not dependent on the address qubit, all addresses are accessed in the same duration.

4.2 Determining Algorithm from Timing

Timing can also be used to determine the type of algorithm used by the victim user. Assuming the attacker knows the set of possible

[1] This work ignores obvious attacks where the cloud provider can directly examine circuits that are submitted to the provider. Such attacks are not protected today, but may be protected in the future so this work focuses on more difficult attacks.

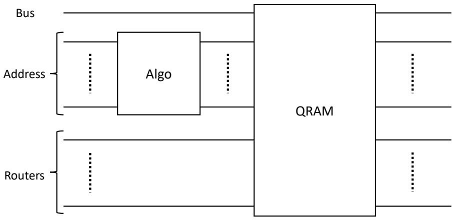

Figure 5: Example of quantum circuit where the algorithm executes first, to generate address qubits, which are later used by the QRAM.

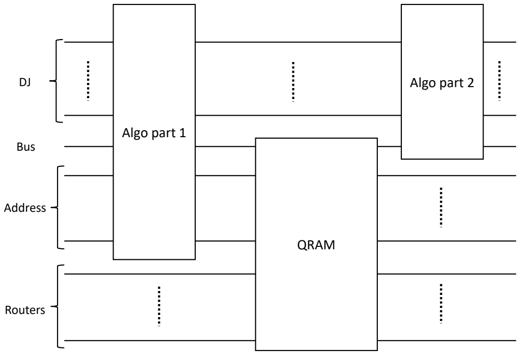

Figure 6: Example of the quantum circuit where the algorithm executes to generate address qubits (part 1), then QRAM is queried, and finally, the algorithm concludes its computation based on the output of the QRAM (part 2).

Table 2: Duration of different algorithms (t_{algo}) and QRAM (t_{QRAM}). The time unit is the backend's cycle time, dt.

Algorithm	Addr. Qubits	Fanout QRAM		BB QRAM	
		t_{algo}	t_{QRAM}	t_{algo}	t_{QRAM}
Grover	3	51600	369480	47400	700040
DJ	3	30600	375800	30600	779280
BV	3	3000	404680	3000	609720
QAOA	3	12120	359240	12120	642040
Simon	3	6449760	343760	6536880	626160
Shor	3	120	371200	120	693240

algorithms the victim is using, he or she can guess the specific one based on timing as we show.

4.2.1 Algorithms That Use QRAM at End. Different algorithms may use QRAM at different times within the execution of the algorithm. In one scenario, the algorithm is run to generate the qubit states that are used as the address, and then the QRAM is accessed. The structure of such a circuit is shown in Figure 5. As can be seen from the figure, the algorithm is run first to generate the address qubits, then use the output of algorithm as the input 3 address qubits of QRAM.

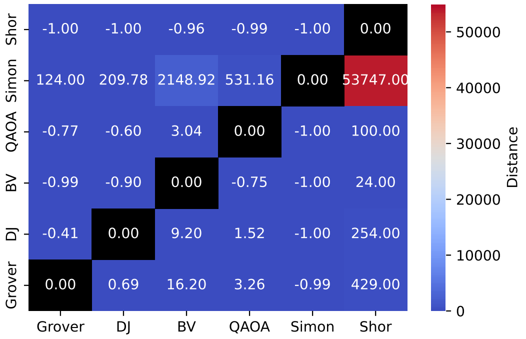

Figure 7: Confusion matrix heatmap for Fanout QRAM.

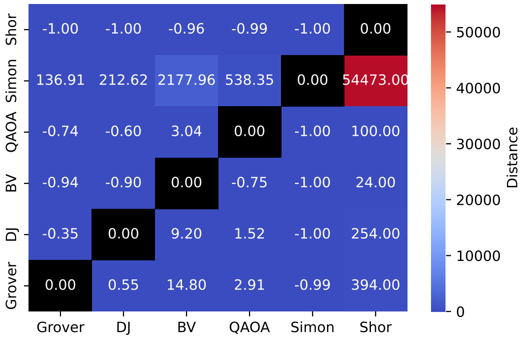

Figure 8: Confusion matrix heatmap for BB QRAM.

Table 3: Duration of DJ algorithm before (t_{pre}) and after (t_{after}) QRAM access, along with QRAM access time (t_{QRAM}). The time unit is the backend's cycle time, dt.

QRAM	Duration		
	t_{pre}	t_{QRAM}	t_{after}
Fanout	120	345840	273440
BB	120	567720	433000

As we can observe in Table 2, the different algorithms and QRAMs have different durations. In this experiment, the Grover and Simon algorithms we use have random input states and we can find that this leads to duration difference of the same algorithm. Note, Qiskit implements *SabreSwap*, a stochastic heuristic algorithm, to compute the swap mapping using random seeds [18], which may cause small differences in QRAM's execution time. Nominally, for the same QRAM size, the access duration should be the same regardless of the algorithm. Further Figures 7 and 8 show the confusion matrix depicting the similarity or distance among the execution times of the different algorithms. As it can be seen, each algorithm is similar to itself (diagonal), while dissimilar to the others. Detailed evaluation of what is the timing resolution in practice for timing attacks is left as future work. But already it can be seen that algorithms with large distances will likely be more easily distinguished in practice.

4.2.2 Algorithms That Use QRAM in the Middle. In this subsection, we consider a different scenario where the QRAM is used in the middle of the algorithm. For this scenario, we adapted the example for the Deutsch-Jozsa (DJ) algorithm from Qiskit.

For the portion before QRAM, in DJ there is a state preparation step (apply X and H gates) to generate the address qubits. If the output of QRAM is 0, then DJ function is considered "constant". Otherwise, DJ is considered "balanced". The goal is to determine whether a DJ function is "constant" or "balanced", thus after querying the QRAM, the rest of the circuit builds and distinguishes what is the DJ function. From Table 3 we can see clear differences among the duration of these three parts of the whole circuit, which means we may be able to distinguish what part of the circuit was executed.

5 HIGHER-ENERGY ATTACKS

Higher-energy attacks abuse the ability of qubits in superconducting quantum computers to be excited into higher-energy levels [15]. Most quantum computers give abstraction of two possible states $|0\rangle$ and $|1\rangle$. However, IBM provides pulse-level access and a function that allows the user to excite selected qubit to higher-energy states, such as $|2\rangle$ and $|3\rangle$. Prior work has demonstrated that the quantum gates and reset operations in superconducting quantum computers are not effective on qubits set in higher-energy states[27].

5.1 Setting Qubits into Higher-Energy States

In order to set a qubit into a higher-energy gate, a custom gate (with custom control pulses) is needed. Users using IBM quantum computers are able to configure such custom gates today [14].

In order to implement a custom gate, a frequency sweep and Rabi experiment are needed [15]. One can build a $|1\rangle \rightarrow |2\rangle$ π custom gate using the corresponding frequency and amplitude obtained from these experiments. Such a gate can be used to excite $|1\rangle$ state into $|2\rangle$. Existing X gate is equivalent to $|0\rangle \rightarrow |1\rangle$ π custom gate. Starting in $|0\rangle$ state, one can apply X gate, followed by a $|1\rangle \rightarrow |2\rangle$ π custom gate to excite the qubit into final $|2\rangle$ state. This can be further extended, and custom pulses can be chained to get $|3\rangle$ and even higher energy levels. The parameters are specific to a target qubit based on its required frequency and amplitude of the control pulses and a custom gate is needed for each qubit on each superconducting quantum computer.

On most IBM machines, $|0\rangle$ and $|1\rangle$ are distinguished on the I-Q planes, and $|2\rangle$ is interpreted as $|1\rangle$ for most discriminators. As a result, if we measure $|2\rangle$ state, the measured result is still 1 and the output is plausible and not the correct higher-energy state.

5.2 Bypassing Reset Operation with Higher-Energy States

In order to evaluate how higher-energy attacks could be deployed, we first tested how the reset gate will not properly reset higher-energy states in qubits. Our experiments confirm that findings from prior work [27] about the ineffectiveness of rest are also applicable to the newer IBM Brisbane machine.

Figure 9 shows a test circuit where the qubit is set to $|2\rangle$, then it is reset as denoted by the $|0\rangle$ box, and finally, it is measured. After the measurement, the output should be 0, but as can be seen from Figure 10 it is actually 1. We evaluated an additional scenario where the qubit is measured before the reset, as shown in Figure 11. Again, we observe that the qubit state is not correctly reset, and the final measurement results in 1 instead of 0.

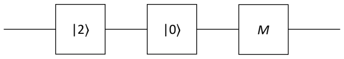

Figure 9: Single qubit reset-measure test circuit.

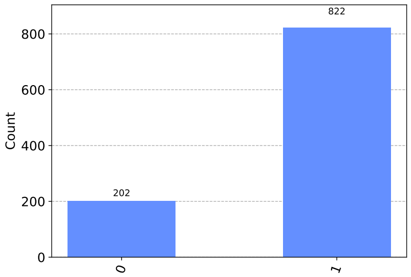

Figure 10: Attack results in reset-measure scenario.

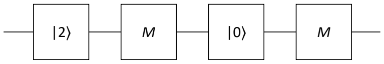

Figure 11: Single qubit measure-reset-measure test circuit.

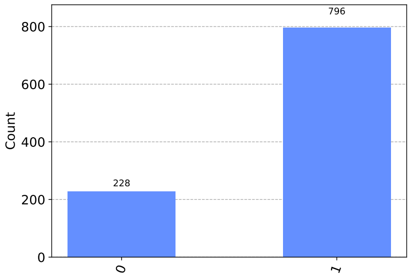

Figure 12: Attack results in measure-reset-measure scenario.

Existing work has also demonstrated that higher-energy states are not affected by regular gate operations [27]. Thus if the qubit is set to a higher-energy state, the gate operations do not affect it (except for decoherence that occurs continuously) and the qubit is read out to $|1\rangle$ no matter the operation applied to it.

5.3 Attack on QRAM Bus Qubit

We now evaluate an attack on the bus qubit. In this scenario, we assume the attacker is able to run his or her circuit right before the victim on the same qubits. This is akin to how time-shared quantum computers can operate, where users share access to the same qubits in a time-sliced fashion. In this attack, the attacker sets the bus qubit to $|2\rangle$.

5.3.1 Results with Unchanged Qubit Mapping. In theory, if the bus is initialized to $|2\rangle$ state, then whatever data is stored in QRAM, the measured output of the bus should always be 1. This is because all the gates used in QRAM are designed for $|0\rangle$ and $|1\rangle$ states and their amplitudes and frequencies do not match $|2\rangle$. As a result, $|2\rangle$ remains in bus qubit and cannot be swapped into QRAM. Figure 13 demonstrates that setting qubit into $|2\rangle$ state, results in a dominant number of measurements to return 1 state.

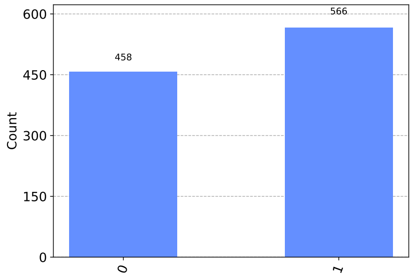

Figure 13: Attack results on QRAM bus qubit when it is set to $|2\rangle$ initially.

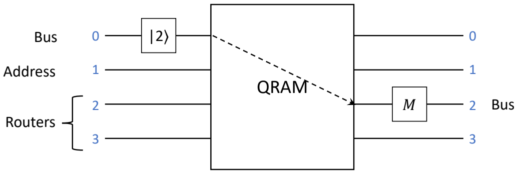

Figure 14: QRAM diagram with changed mapping due to transpilation and optimization. The physical qubit 0 that was the bus qubit initially is no longer the bus qubit after the mapping changes.

5.3.2 Results with Routing Problem. In practice, post-processing of the circuits can change the qubit mapping, or can introduce swaps that swap qubits, and the initial bus qubit which is set to higher-energy, is not the bus qubit that is eventually measured. Indeed, before submitting a QRAM circuit to an IBM machine, we need to compile the circuit and submit a transpiled circuit which is constructed through only a set of basic gates. During transpilation, there are six stages in Qiskit SDK: init, layout, routing, translation, optimization, and scheduling. The routing stage inserts SWAP gates into the circuit to move wanted physical qubits together so that two-qubit gates can be performed between these adjacent qubits. Qiskit uses a stochastic heuristic algorithm called *SabreSwap* to compute such swap mapping.

For our test circuits, we define the initial mapping using a trivial layout method as shown in Figure 14. The bus is mapped to physical qubit 0. QRAM qubits are mapped to physical qubits 1, 2, 3. We use 1 address qubit Fanout QRAM and combine the address qubit with one of the router qubits. However, a SWAP gate is inserted between the logical bus and other QRAM qubit. Because the normal SWAP gate cannot correctly operate on $|2\rangle$, the state of physical qubit 0 remains $|2\rangle$ until the end of the circuit, while logically the qubit should be swapped due to the SWAP gate.

We measure all qubits in order to show that such a mapping problem can affect the whole circuit. As we can see in Figure 16, the measured result is 1101 instead of ideal 0000 in Figure 15 when the bus is initialized to $|2\rangle$. There should be at least one 1 in the measured result because $|2\rangle$ is mistaken for $|1\rangle$. The reason why there are more 1s is that after swapping the bus with the qram qubit, the following gates on these two qubits will also be swapped to their new corresponding physical qubits. But these gates are actually performed in the wrong states because the correct states are still kept on old physical qubits and cannot be swapped together with these gates.

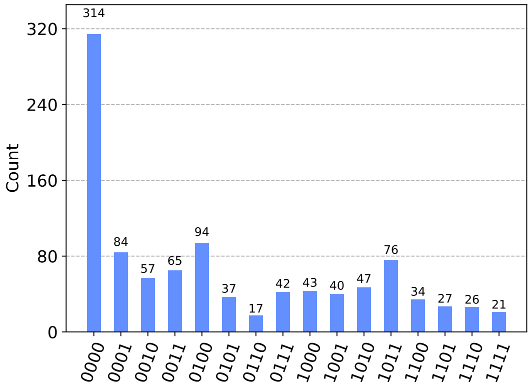

Figure 15: Expected output with bus qubit set $|0\rangle$, i.e. no attack is performed.

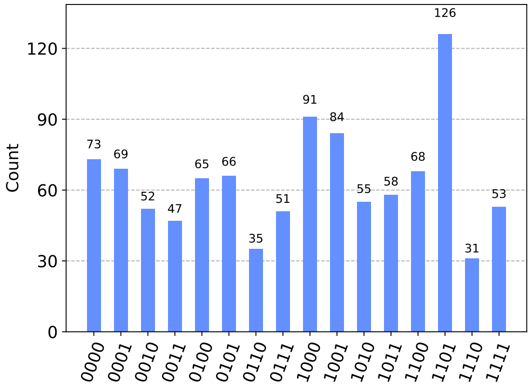

Figure 16: Output with bus qubit set to $|2\rangle$ due to attack and with changed mapping.

5.4 Attack on QRAM Address Qubit

We now evaluate the attack on the address qubit. In this scenario, we set address qubit to $|2\rangle$ and data stored in QRAM to $|1\rangle$ so the expected output should be 1 no matter which memory cell is retrieved. The 0 output in Figure 17 shows when address qubit is initialized to $|2\rangle$, then the bus can not retrieve data from QRAM because CX gates during the address loading stage in Fanout QRAM are invalid for $|2\rangle$.

5.5 Attack Across Multiple QRAM Accesses

In Figure 18, we show an attack involving two QRAM accesses. We initialize bus qubit to $|2\rangle$, query QRAM, measure all qubits and reset them after first access, query QRAM again, measure bus. The ideal output should be all zero state because we set all the memory cells in QRAM to 0. However, the results in Figure 19 show that a higher energy state cannot be reset simply through a regular reset gate between the two QRAM access, which means if the attacker

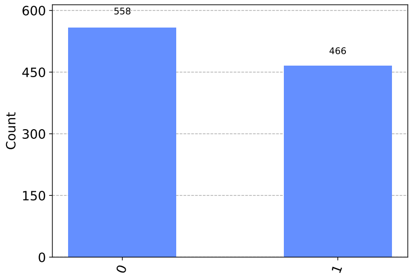

Figure 17: Result of attack on QRAM address qubit. The attacker sets the initial address qubit to $|2\rangle$. All data stored in QRAM are set to 1.

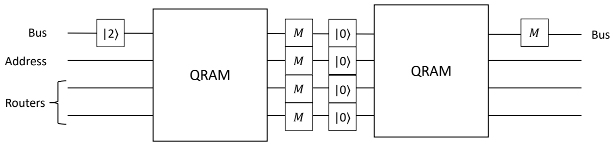

Figure 18: Test circuit with two QRAM circuits. The attacker sets the bus qubit to $|2\rangle$, and then QRAM is accessed, results are measured, the bus qubit is reset and QRAM is accessed again. Results show that even is such a setting, the higher-energy state persists across multiple QRAM accesses and the final measurement of bus qubit is not correct.

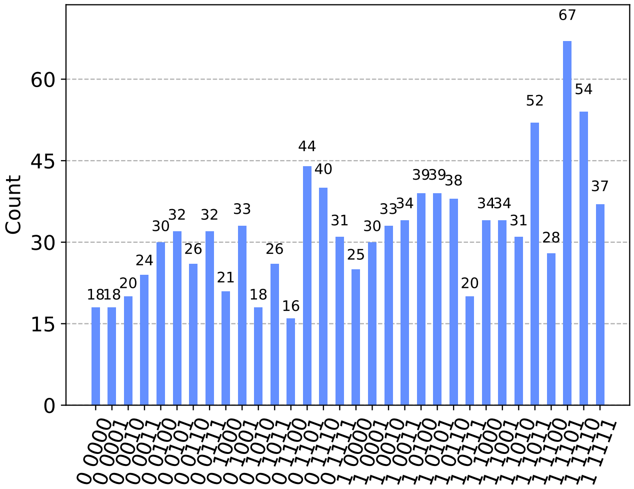

Figure 19: Results of attack on QRAM with initial bus qubit being set to $|2\rangle$ and QRAM being accessed twice. The upper four bits are measurements of the first QRAM. The bottom number is bus output of the second QRAM.

accesses QRAM first and injects such a higher energy state, the following user will query the QRAM that has not been successfully initialized.

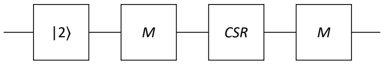

Figure 20: Single qubit measure measure-CSR-measure test circuit.

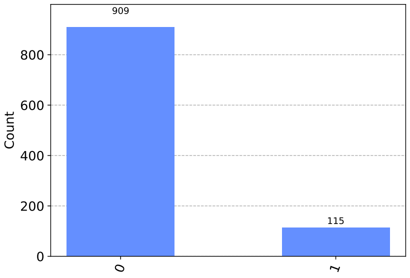

Figure 21: Attack results in measure-CSR-measure scenario.

6 DEFENSES

This section presents possible defenses against the timing and higher-energy attacks. The proposed defenses can be realized without modification to the hardware while allowing existing quantum computer functionality to be unchanged.

6.1 Defense for Timing Attack

Timing-based attacks are easiest to defend by ensuring that the execution time of the algorithm and the QRAM access are constant. By design, QRAM access does not depend on the address qubits, thus for a fixed QRAM size, it is the algorithm portion that needs to be made constant. To extend the duration of any algorithm, sequences of two X gates can be added. Simply adding delays will cause the qubits to decohere, but sequences of two X gates in sequence act to limit the decoherence while preserving the qubit state. Each X gate is akin to a classical NOT gate, thus two X gates logically cancel each other out and preserve the qubit state. When considering a set of algorithms, each can be extended by a different amount so that the total execution of the algorithms is the same and they cannot be distinguished by timing changes. Detailed evaluation of this defense for timing attacks is left as future work.

6.2 Defense for Higher-Energy Attacks

Higher-energy attacks can be defended if $|2\rangle$ and higher-energy level states are prohibited. This, however, may limit the utility of future quantum computers because higher-energy states have positive use in quantum machine learning [26]. Another approach is then to allow the higher-energy states, but correctly reset them so that bus or address qubits are not affected.

As we have shown, the injection of $|2\rangle$ state can disturb the QRAM circuit even after resetting. One possible defense is to make use of the recently proposed Cascading Secure Reset, or CSR, gate [27]. This gate has been developed to help reset higher-energy states. In CSR gate, if we only consider $|1\rangle$ state, it uses a basic reset gate to reset it to $|0\rangle$. If we consider $|2\rangle$ state, CSR gate applies $1 \longrightarrow 2$ π pulse first to switch it to $|1\rangle$, then use reset gate to reset

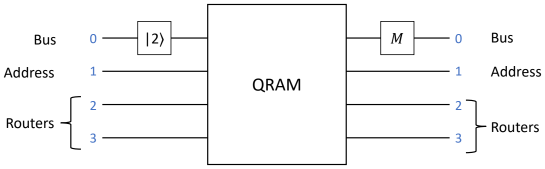

Figure 22: Test circuit for testing defenses for higher-energy attacks on QRAM bus qubit. The circuit illustrates the bus qubit being set to |2⟩, the reset or CSR gate used after the bus qubit is not shown.

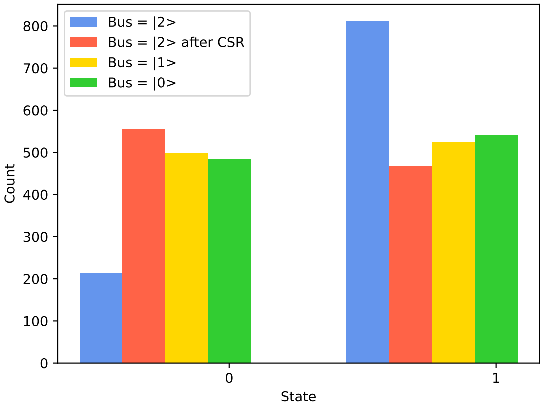

Figure 23: Results of a test circuit for testing defenses for higher-energy attacks on QRAM bus qubit.

back to |0⟩. The CSR gate can be implemented to protect against arbitrary levels of the higher-energy states.

We have evaluated the CSR gate on the newer IBM Brisbane machine where the gate has not been tested before in detail. We used a test circuit with a single qubit as shown in Figure 20. The test attack results are shown in Figure 21. Based on the data, it can be observed that indeed the CSR gate is able to bring the qubits back into their ground states. Following the CSR gate resetting of the qubits, the measurement results return predominantly 0. Thus if bus and address qubits are reset with CSR gate before execution of the algorithm, the attack should be mitigated.

To illustrate the defense, we implemented such defense with the circuit illustrated in Figure 22. The bus qubit is set to |2⟩ to simulate an attack. The other qubits are set to |0⟩. We ensured that the remapping and routing changes did not occur in this test.

As shown in Figure 23, if bus qubit is set to |2⟩ and regular reset is used (blue bars) the state output is unbalanced and incorrect. When bus qubit is set to |2⟩ and CSR gate is used, the output states are balanced (red bars). When bus qubit is set to |1⟩ or |0⟩ and regular reset is used, the output states are also balanced (yellow and green bars). In summary, if the attacker injects |2⟩ state to QRAM bus qubit during, the CSR gate can reset all possible states before QRAM executes and the attack is mitigated.

7 RELATED WORK

Recently, some research has been done on side-channel attacks on quantum computers. In classical computing, timing and power side-channels are widely studied. As a counterpart, the timing and power side-channel attacks on quantum computer controllers were proposed, and how they can be used to reconstruct quantum circuits are demonstrated [28]. However, the power side-channel data required in [28] is per-channel data. Following this work, a novel method requiring only the aggregate power side-channel data was proposed [11]. Similarly, a more detailed study on the timing side-channel attacks was done in [19].

Much other research has been done to explore the security and privacy issues raised due to imperfections in current quantum computers. Malicious users might exploit the shared quantum computing environments to infer the quantum states of other users' qubits by examining leaked data from computation results. One key source of this data leakage is the noise and errors introduced during operations like qubit resetting, which is crucial between circuit executions. These imperfect resets can inadvertently transmit information to subsequent executions, creating a vulnerability that has been leveraged in various types of attacks, such as reset attacks [20, 25], side-channel attacks [5], and higher-energy state attacks [27]. This phenomenon is referred to as "horizontal" leakage in [29], where information flows sequentially from one execution to the next. Conversely, "vertical" leakage occurs across qubits at the same time, representing another form of vulnerability, as shown in crosstalk attacks [2, 3, 8, 9] and qubit sensing [4].

The horizontal leakage was thoroughly studied in [29], and the one-time pad was proposed as the countermeasure to mitigate such attacks. However, this work still examines only the two-level computing systems in quantum computers, without considering higher-energy state attacks [27], though similar one-time pad schemes may be developed by replacing the X gate with the π gate between different energy levels. This work extends the high-energy state attacks by applying them to QRAM.

8 CONCLUSION AND FUTURE WORK

This work presented the first evaluation of timing and higher-energy attacks on QRAM circuits, and on algorithms that use QRAM circuits. The work demonstrated side-channel attacks, e.g., the timing attacks, as well as fault-injection-like attacks, e.g., the higher-energy attacks. We have validated that previously demonstrated higher-energy attacks still work on more recent superconducting quantum computers and then leveraged the ideas to attack QRAM. The proposed defenses are simple and can be deployed today. The simple defenses can be implemented in software, such as by adding the X gates or incorporating CSR gate.

Future work can explore in-depth the benefits (and overheads) of the defenses. For example, timing attack defenses will result in fidelity degradation of the circuits. The attacks can be also evaluated against other types of QRAM, beyond Fanout QRAM and BB QRAM explored in this work. As QRAM becomes more practical, more circuits will use it and these circuits and algorithms can be analysed.

ACKNOWLEDGMENTS

This work was supported in part by NSF grant 2312754.

REFERENCES

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

[2] Abdullah Ash-Saki, Mahabubul Alam, and Swaroop Ghosh. 2020. Analysis of Crosstalk in NISQ Devices and Security Implications in Multi-Programming Regime. In *International Symposium on Low Power Electronics and Design (ISLPED)*. Association for Computing Machinery, 25–30.

[3] Abdullah Ash-Saki, Mahabubul Alam, and Swaroop Ghosh. 2020. Experimental Characterization, Modeling, and Analysis of Crosstalk in a Quantum Computer. *IEEE Transactions on Quantum Engineering* 1 (2020), 1–6.

[4] Abdullah Ash-Saki and Swaroop Ghosh. 2021. Qubit Sensing: A New Attack Model for Multi-programming Quantum Computing. arXiv:2104.05899

[5] Brennan Bell and Andreas Trügler. 2022. Reconstructing quantum circuits through side-channel information on cloud-based superconducting quantum computers. In *Conference on Quantum Computing and Engineering (QCE)*. 259–264.

[6] David Brumley and Dan Boneh. 2005. Remote timing attacks are practical. *Computer Networks* 48, 5 (2005), 701–716.

[7] Christopher Chamberland, Kyungjoo Noh, Patricio Arrangoiz-Arriola, Earl T Campbell, Connor T Hann, Joseph Iverson, Harald Putterman, Thomas C Bohdanowicz, Steven T Flammia, Andrew Keller, et al. 2022. Building a fault-tolerant quantum computer using concatenated cat codes. *PRX Quantum* 3, 1 (2022), 010329.

[8] Sanjay Deshpande, Chuanqi Xu, Theodoros Trochatos, Yongshan Ding, and Jakub Szefer. 2022. Towards an Antivirus for Quantum Computers. In *International Symposium on Hardware Oriented Security and Trust (HOST)*. 37–40.

[9] Sanjay Deshpande, Chuanqi Xu, Theodoros Trochatos, Hanrui Wang, Ferhat Erata, Song Han, Yongshan Ding, and Jakub Szefer. 2023. Design of Quantum Computer Antivirus. In *International Symposium on Hardware Oriented Security and Trust (HOST) (HOST)*. 260–270.

[10] 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

[11] Ferhat Erata, Chuanqi Xu, Ruzica Piskac, and Jakub Szefer. 2024. Quantum Circuit Reconstruction from Power Side-Channel Attacks on Quantum Computer Controllers. arXiv:2401.15869 [cs.CR] https://arxiv.org/abs/2401.15869

[12] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. 2008. Architectures for a quantum random access memory. *Physical Review A—Atomic, Molecular, and Optical Physics* 78, 5 (2008), 052310.

[13] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. 2008. Quantum random access memory. *Physical review letters* 100, 16 (2008), 160501.

[14] IBM. 2024. Pulse schedules documentation. https://docs.quantum.ibm.com/guides/pulse#pulse-schedules.

[15] IBM. 2024. Quantum hardware pulses documentation. https://github.com/Qiskit/textbook/tree/main/notebooks/quantum-hardware-pulses.

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

[17] IBM Quantum. 2023. The hardware and software for the era of quantum utility is here. https://research.ibm.com/blog/quantum-roadmap-2033.

[18] Gushu Li, Yufei Ding, and Yuan Xie. 2019. Tackling the qubit mapping problem for NISQ-era quantum devices. In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*. 1001–1014.

[19] Chao Lu, Esha Telang, Aydin Aysu, and Kanad Basu. 2024. Quantum Leak: Timing Side-Channel Attacks on Cloud-Based Quantum Services. arXiv:2401.01521 [cs.ET] https://arxiv.org/abs/2401.01521

[20] Allen Mi, Shuwen Deng, and Jakub Szefer. 2022. Securing Reset Operations in NISQ Quantum Computers. In *Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, 2279–2293.

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

[22] 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

[23] Ciaran Ryan-Anderson, Justin G Bohnet, Kenny Lee, Daniel Gresh, Aaron Hankin, JP Gaebler, David Francois, Alexander Chernoguzov, Dominic Lucchetti, Natalie C Brown, et al. 2021. Realization of real-time fault-tolerant quantum error correction. *Physical Review X* 11, 4 (2021), 041058.

[24] Jakub Szefer. 2018. Principles of Secure Processor Architecture Design. *Synthesis Lectures on Computer Architecture* 13, 3 (2018), 1–173.

[25] Jerry Tan, Chuanqi Xu, Theodoros Trochatos, and Jakub Szefer. 2023. Extending and Defending Attacks on Reset Operations in Quantum Computers. arXiv:2309.06281 [cs.AR] https://arxiv.org/abs/2309.06281

[26] Themistoklis Valtinos, Aikaterini Mandilara, and Dimitris Syvridis. 2024. The Gell-Mann feature map of qutrits and its applications in classification tasks. In *Quantum Computing, Communication, and Simulation IV*, Vol. 12911. SPIE, 229–247.

[27] Chuanqi Xu, Jessie Chen, Allen Mi, and Jakub Szefer. 2023. Securing NISQ Quantum Computer Reset Operations Against Higher Energy State Attacks. In *Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS)*. Association for Computing Machinery.

[28] Chuanqi Xu, Ferhat Erata, and Jakub Szefer. 2023. Exploration of Power Side-Channel Vulnerabilities in Quantum Computer Controllers. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 579–593. https://doi.org/10.1145/3576915.3623118

[29] Chuanqi Xu, Jamie Sikora, and Jakub Szefer. 2024. A Thorough Study of State Leakage Mitigation in Quantum Computing with One-Time Pad. In *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 55–65. https://doi.org/10.1109/HOST55342.2024.10545386