2025 38th International Conference on VLSI Design and 2025 24th International Conference on Embedded Systems (VLSID)

2025 38th International Conference on VLSI Design and 2025 24th International Conference on Embedded Systems (VLSID) | 979-8-3315-2244-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/VLSID64188.2025.00108

Investigating Impact of Bit-flip Errors in Control Electronics on Quantum Computation

Subrata Das
School of EECS
Pennsylvania State University
University Park, PA, USA.
sjd6366@psu.edu

Avimita Chatterjee
School of EECS
Pennsylvania State University
University Park, PA, USA.
amc8313@psu.edu

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

***Abstract*—In this paper, we investigate the impact of bit-flip errors in FPGA memories in control electronics on quantum computing systems. FPGA memories are integral in storing the amplitude and phase information pulse envelopes, which are essential for generating quantum gate pulses. However, these memories can incur faults due to physical and environmental stressors such as electromagnetic interference, power fluctuations, and temperature variations and adversarial fault injections, potentially leading to errors in quantum gate operations. To understand how these faults affect quantum computations, we conducted a series of experiments to introduce bit flips into the amplitude (both real and imaginary components) and phase values of quantum pulses using IBM's simulated quantum environments, FakeValencia, FakeManila, and FakeLima. We compare the sensitivity of floating-point and fixed-point representa- tions to these bit-flip errors. The findings reveal that bit flips in the exponent and initial mantissa bits of the real amplitude in floating- point representation cause substantial deviations in quantum gate operations, with TVD increases as high as $\sim$ 200%. Conversely, fixed-point representation shows reduced sensitivity to bit-flips, offering a more robust alternative for certain applications. These in- sights can guide the selection of data representations to enhance the robustness of quantum computing systems from hardware faults.**

***Index Terms*—FPGA Faults, Quantum Control Systems, Fault Analysis, Computational Reliability, Data Representation**

I. Introduction

Quantum computing represents a paradigm shift in computational capabilities, enabling the resolution of complex problems beyond the reach of classical computing methods. Unlike classical systems that use bits, quantum computing employs quantum bits, or qubits, capable of existing in multiple states simultaneously, thereby enhancing computational efficiency significantly [1], [2]. The operational synergy between quantum and classical computing systems is vital, particularly in control electronics which manage qubit operations via microwave pulses. This interaction is important for the functionality of quantum computers that operate based on superconducting qubits [3], [4]. An example is the Sycamore chip, which manages 53 qubits and involves an extensive array of over 200 digital-to-analog converters (DACs), 9 analog-to-digital converters (ADCs), and more than 30 field-programmable gate arrays (FPGAs) to ensure effective control and readout [5].

This work is supported in parts by NSF (CNS-1722557, CNS-2129675, CCF-2210963, CCF-1718474, OIA-2040667, DGE-1723687, DGE-2113839, and DGE-1821766) and Intel's gift.

A. Importance of Control Electronics in Quantum Computing

Quantum circuits, essential for manipulating qubit information, consist of sequences of quantum gates that perform complex computations through coherent manipulations. These operations are facilitated by FPGA-based control systems that generate and measure precise microwave pulses necessary for qubit manipulation, as depicted in Fig. 1 [3]. The integration of FPGA, DAC, and ADC into Radio Frequency System-on-Chips (RFSoCs) exemplifies the advancement in compact and efficient control systems like the 'QICK' and 'ICARUS-Q' platforms [6], [7]. The operational process involves submitting quantum circuit specifications to a cloud interface for optimization, followed by FPGA-based generation of control pulses, detailed in Fig. 1, which are important for the accurate execution of quantum operations [8], [9].

B. Understanding Bit Flip Errors in Classical Memory

FPGAs, being classical devices, are susceptible to bit-flip errors. Bit flip errors in the block RAMs (BRAMs) of an FPGA can occur due to several factors, often related to the physical and environmental conditions in which the FPGA operates such as radiation effects, electrical noise, and temperature variations [10], [11], [15]. The BRAMs in FPGAs used in quantum computer control electronics undergo further challenges like intense operations to meet the high-performance requirements which can create additional noise in the system (e.g., power supply droop). This, in turn, can cause read/write/retention failures. Such intense operations and subsequent errors can also be adversary-induced. For example, adversaries sharing the same quantum hardware as victims in multi-tenant computing environments can write a Trojan program to generate frequent pulses, increasing power consumption and subsequent supply noise. An insider adversary can also consider non-invasive tampering of the FPGA to inject bit flip errors by manipulating ambient conditions. The impact of bit-flip errors in BRAMs can be significant due to the precision and sensitivity required in quantum computing. Some specific reasons that could lead to bit flip errors or

2380-6923/25/$31.00 ©2025 IEEE
DOI 10.1109/VLSID64188.2025.00108

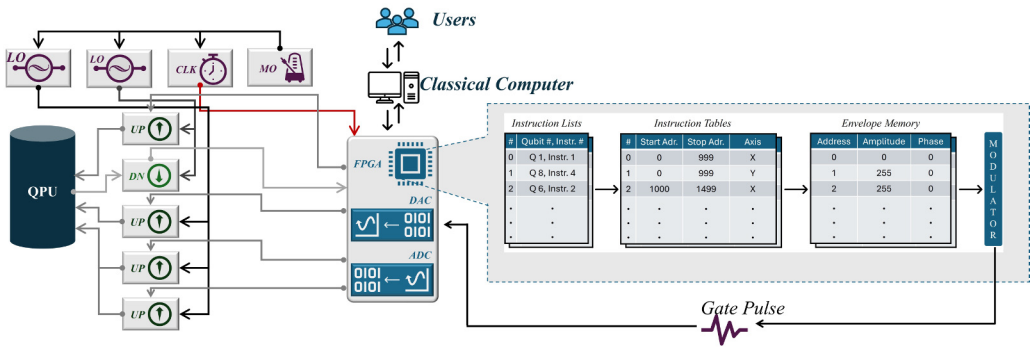

Fig. 1. **Qubit control system hardware and instruction Cycle in quantum computing using FPGA:** Key components include the master oscillator (MO), clock (CLK), local oscillator (LO), field-programmable gate array (FPGA), digital-to-analog converter (DAC), analog-to-digital converter (ADC), upconverter (UP) and downconverter (DN). The diagram also illustrates the instruction cycle, where FPGA memory processes amplitude and phase details for each quantum gate, converting these details into precise RF pulses directed at the quantum processor unit (QPU), enabling effective quantum operations.

TABLE I
SOURCES OF FAULTS IN FPGA MEMORY

Mode of Fault	**Fault Origins**	**Impact on FPGA Memory**
Electromagnetic Interference (EMI)	- External sources like nearby electronic devices, power lines [10]	- Induces errors in electronic components controlling qubits
Power Fluctuations	- Unstable power supply - Supply noise [11]	- Can lead to bit flips, disrupt operations
High-Precision Timing Requirements	- Inaccurate timing in qubit manipulation [12]	- Can cause memory faults like bit flips
Aging of Hardware	- Bias temperature instability [13] - Hot carrier injection, and electromigration	- Leads to memory faults like bit flips over time
Manufacturing Defects	- Imperfections in the manufacturing process [14] - Weak shorts or opens during manufacturing	- Might not be evident initially but can manifest as frequent bit flips occur under certain conditions
Adversarial Fault Injection	- Non-invasive methods such as elevating ambient temperatures, applying radiation - Overloading FPGA with intense computational demands [15] - Creating instability through power supply noise	- Allows adversaries to compromise the system without physical interaction - Specifically possible in shared environments, where quantum computing resources are utilized by multiple parties, enabling indirect interference with critical computing operations

similar memory disturbances in quantum computer control electronics are tabulated in Table I. To make the matters worse, current state-of-the-art FPGAs do not support ECC (Error Correcting Code) on BRAMs by default. For example, single port BRAM in Xilinx FPGAs is not ECC protected [16].

C. Data Representations in FPGA Memories

Traditionally, information in FPGA memories are stored using floating-point representation [17]. This format is favored for its ability to handle a wide dynamic range, making it suitable for storing information with arbitrary values. However, fixed-point representation is another efficient method for FPGA particularly when the data range is limited but precision is important. Fixed-point representation is easily synthesizable and offers higher precision for a given number of bits within a smaller range compared to floating-point representation.

D. Translating Classical Errors to Quantum Errors

Bit flip errors in FPGA memory can disrupt quantum operations by causing faulty pulse generation. Fig. 2 illustrates the pulse shape of a Hadamard Gate from IBM's FakeValencia Backend obtained using Qiskit. The bottom part of the figure shows the actual pulse shape, with the amplitude represented as a 32-bit floating-point number stored in the FPGA memory. If a bit flip occurs, such as a flip in the third bit, the amplitude undergoes a dramatic decrease from its original value of 0.09618851775276127+0.0008448724348311288j to a bit-flipped value of 2.239563395844968e-11+0.0008448724348311288j. This significant reduction (also shown in Fig. 2) can lead to substantial errors in the quantum computing outcomes.

E. Contributions

This paper provides a comprehensive analysis of the effects of bit flip errors in FPGA memory, specifically within the context of quantum computing control systems. First, we conducted a detailed and controlled sensitivity analyses considering floating-point and fixed-point data representation formats. We introduced single-bit flips in the amplitude (both real and imaginary components) and phase values of quantum pulses, using various IBM quantum fake backends that simulate real quantum computing environments. This approach allows for a precise understanding of how such errors affect quantum gate operations. And second, our analysis goes beyond general fault

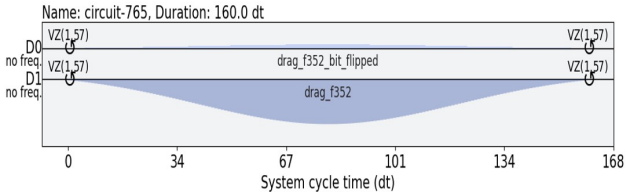

Fig. 2. Pulse shape impact from bit flip in Hadamard Gate using floating-point representation. The bottom graph shows the original pulse amplitude for IBM's FakeValencia Hadamard Gate, and the top graph illustrates the drastic reduction in amplitude caused by a third-bit flip in FPGA memory.

impacts by examining the specific effects of bit flips on the functionality and performance of various native quantum gates such as Hadamard, NOT, and Rotation-gates.

F. Paper Organization

In the remainder of this paper, Section II provides a background on the role of FPGAs in quantum control systems and explains the basics of quantum pulse specification and data representation. Section III details our experimental setup and findings on the bit flip errors. Finally, Section IV draws conclusions.

II. Background

A. FPGA in Quantum Control Systems

Field-Programmable Gate Arrays (FPGAs) generate high-frequency signals necessary for qubit manipulation through quantum gates. Their reconfigurability allows for rapid adaptation to evolving quantum algorithms and error correction codes without needing hardware replacement, unlike ASICs which are less flexible [3], [4]. Additionally, FPGAs play a vital role in error management by processing readout signals from qubits to detect and correct errors in real-time, critical for maintaining the accuracy of quantum computations. This functionality extends to managing the scalability challenges in quantum systems as they grow to include more qubits [5].

B. Quantum Pulse Specification and Data Representation in FPGAs

1) Complex Amplitude Components: Quantum computing relies on precisely controlled, complex waveforms to manipulate qubits effectively. Each quantum pulse is characterized by its amplitude, phase, and duration, with the amplitude typically represented as a complex number comprising real (I) and imaginary (Q) components. These components are fundamental to executing coherent quantum manipulations, with the I/Q representation mirroring the spectral efficiency methods used in digital communication. The real (I) part manages the pulse's basic strength and phase, while the imaginary (Q) part adds a critical phase shift for precise qubit state manipulation [2].

2) Phase Control: Phase control is essential in quantum computing for precise qubit state manipulation, facilitated by the `ShiftPhase` instruction in platforms like IBM's Qiskit Pulse. This instruction crucially updates the modulation phase of subsequent pulses on the same channel, allowing for fine adjustments necessary for accurate quantum operations. The `ShiftPhase` modifies the phase ϕ of the output signal, impacting all following pulses:

$$\mathrm{Re}\left[\exp\left(i2\pi f_j dt + \phi\right) d_j\right]. \tag{1}$$

This representation describes the real part of the pulse generated on a `PulseChannel`, emphasizing the importance of precise phase adjustments for effective quantum state rotations around the z-axis, thus ensuring nearly error-free Z-rotations.

3) Floating Point Representation in FPGA Memory: Typically, the real and imaginary parts of pulse amplitude, along with values used in phase control instructions like `ShiftPhase`, are represented as floating-point numbers. Floating point numbers in FPGA-based quantum control systems are typically stored using the IEEE 754 standard where a floating-point number is represented through three distinct parts: (a) **Sign bit:** A single bit that indicates if the number is positive or negative. (b) **Exponent:** An 8-bit field (for single precision) that scales the number by powers of two. (c) **Mantissa (or significand):** A 23-bit field that contains the significant digits of the number. For example, the floating-point representation of the number -248.75 in IEEE 754 format is as follows:

- The sign bit is 1, indicating a negative number.
- The exponent is encoded with a bias of 127; for the actual exponent of 7, it is stored as 135, or `10000110` in binary.
- The mantissa is derived from the non-integer part of the number, stored as `11110001100000000000000` in binary.

This encoding results in the binary representation `11000011011110001100000000000000`.

4) Fixed Point Representation in FPGA Memory: In some FPGA memories, fixed-point representation is used instead of floating-point representation as it can provide higher precision for a given number of bits within a smaller range. A fixed-point number is represented by three main components: (a) Sign bit: A single bit that indicates if the number is positive or negative. (b) Integer part: The bits that represent the integer portion of the number. (c) Fractional part: The bits that represent the fractional portion of the number. For example, consider a 32-bit fixed-point representation with 1 sign bit, 15 bits for the integer part, and 16 bits for the fractional part. The number -7.9375 would be represented as follows:

- The sign bit is 1, indicating a negative number.
- The integer part is represented by the next 15 bits. The binary representation of -7 is 111111111111000 (in two's complement format).
- The fractional part is represented by the next 16 bits. the binary representation of 0.9375 is 0001000000000000.

This encoding results in the binary representation of 1111111111111000 0001000000000000. Fixed-point representation in FPGAs can be advantageous in quantum control systems where the range of pulse amplitude and phase values is limited but precision is important. Unlike floating-point

representation, fixed-point representation allows for more bits to be dedicated to the fractional part, thus providing finer granularity.

III. Analysis of Bit Flip Errors

A. Experimental Setup

We utilized the Qiskit framework to simulate quantum circuits and manipulate quantum pulse sequences on IBM's fake quantum backends.

1) Quantum Gate and Pulse Configuration: We start by constructing a basic quantum circuit for different single and two qubit gates using the QuantumCircuit module. The circuit is then transpiled and scheduled for the specific backend to obtain the corresponding control pulse sequence. This step translates high-level quantum operations into the low-level pulses that physically manipulate the qubits. Each pulse in the sequence, particularly the Drag pulse associated with the X gate, is analyzed to extract its amplitude and phase components. The Drag pulse which is designed to minimize leakage to non-computational states consists of a complex amplitude described by real (I) and imaginary (Q) parts. These components are subsequently converted to both 32-bit floating-point and 32-bit fixed-point formats. Bit flip experiments are performed on both binary representations.

2) Bit Flip Simulation and Analysis: For floating-point representation, each bit, from the least significant bit of the mantissa to the sign bit, is flipped at a time to observe the resultant effect on the quantum operation. Same approach is adopted for fixed-point representation. After flipping a bit, the modified binary string is converted back to its respective numerical format, which is then used to adjust the amplitude of the Drag pulse in a custom-built pulse sequence.

3) Measurement and Comparison Metrics: The primary metric used to evaluate the impact of the bit flips is the Total Variation Distance (TVD) which is a statistical measure used to quantify the difference between two probability distributions. In our experiments, TVD is calculated between the probability distributions of the quantum state measurements obtained from the ideal (unmodified) and the perturbed (bit-flipped) pulse sequences. Each experiment is run multiple times to ensure statistical relevance.

B. Results with Floating-Point Representation

In this subsection, we evaluate the sensitivity of floating-point representation in FPGA memories to bit-flip errors, assessing their impact on quantum gate operations through TVD (Total Variation Distance) measurements between ideal and perturbed quantum states.

We conducted experiments across three simulated backends—FakeValencia, FakeManila, and FakeLima—manipulating the real part of the amplitude of pulse envelopes for X and H gates. During these experiments, flipping certain exponent bits (notably bits 1 and 6) led to invalid pulses, exceeding the maximum pulse amplitude norm of 1.0. We assume that such extreme conditions would be immediately recognized by the system's error detection proto- cols to halt the transmission of the pulse. In our experiments, to provide an uninterrupted visual trend, we employed a linear inter- polation method to estimate the missing TVD values by connecting the adjacent data points directly. The observed trends were consistent across the backends for both X and H gates, as shown in Fig. 3. TVD measurements indicated that early exponent bits (1 to 8) and certain mantissa bits (9 to 17) significantly influenced the pulse's amplitude, causing substantial changes in quantum gate behavior. Conversely, later mantissa bits (18 to 31) showed minimal impact, demonstrating their limited role in the operational characteristics of the gates [2].

Given the consistent trends across different backends and to enhance resource efficiency, we focused subsequent detailed experiments on only FakeValencia backend. Additional experiments included quantum gates such as the SX, CNOT, and rotation gates Rx at $\pi/4$ and $\pi/3$ angles, focusing on how bit flips in the real part of the amplitude affect gate functionality, as illustrated in Fig 4. The experiments confirmed that exponent bits significantly affect the amplitude, with a notable TVD increase when altered. The Rx and CNOT gates displayed resilience to these bit flips, highlighting their robust error handling capabilities. The Rx gates, which perform rotations around the x-axis by specific angles, might be less affected by small amplitude changes because their primary function is to change the phase of the qubits rather than their state. As for the CNOT gate, its operation as a two-qubit control gate might inherently buffer it against the impact of amplitude changes in single-qubit control lines. Its functionality relies more on the relative states of two qubits rather than on the precise amplitude of a control pulse [2].

Furthermore, we analyzed the effects of bit flips in the imaginary part of the pulse amplitude (except for the X gate, which lacks an imaginary component). As depicted in Fig. 5, these bit flips showed no consistent impact across different bits. This diminished sensitivity can be attributed to the relatively smaller magnitude of the imaginary part of the pulse amplitude compared to its real counterpart. For example, the amplitude for the H gate on FakeValencia backend is recorded as 0.09618851775276127+0.0008448724348311288j, where the imaginary part is much smaller than the real part. Next, we evaluated the impact of bit flips on phase shift values using the `ShiftPhase` instruction. These findings, shown in Fig. 6, revealed that bit flips in phase shift values do not follow a consistent pattern across different bits. The TVD impacts were generally lower compared to those from amplitude bit flips. `ShiftPhase` instruction does not alter the pulse amplitude but adjusts the phase angle ϕ of the output signal, impacting all following pulses in the same channel. The fact that they do not directly change the amplitude means that errors in phase values may have a less disruptive impact on the overall quantum state compared to amplitude errors.

C. Results with Fixed-Point Representation

We introduce bit-flips across both the integer and fractional parts of the fixed-point format and examine their impact on quantum operations using TVD. Utilizing a fixed-point format

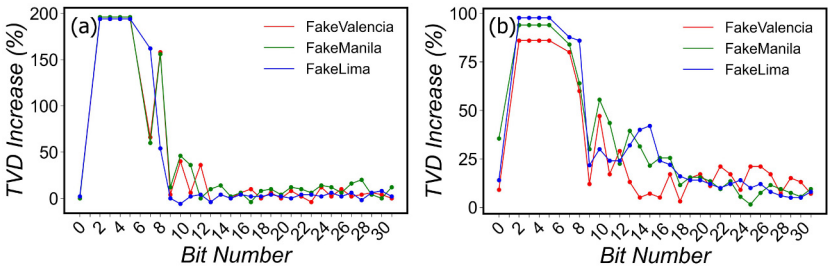

Fig. 3. TVD increase from bit flips in the real part of amplitude (floating-point representation) for (a) X and, (b) H gates across three simulated backends: FakeValencia, FakeManila, and FakeLima demonstrating consistent sensitivity trend across backends.

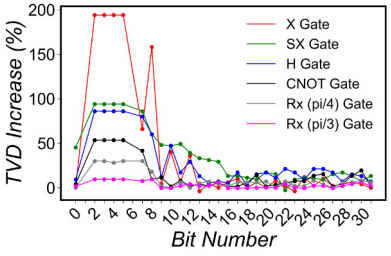

Fig. 4. TVD increase from bit flips in the real part of amplitude (floating-point representation) for various quantum gates simulated in FakeValencia backend.

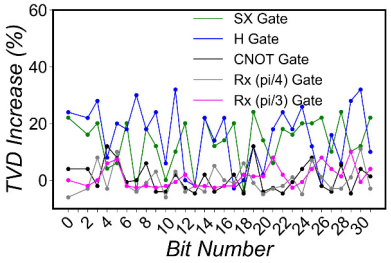

Fig. 5. TVD increase from bit flips in the imaginary part of amplitude (floating-point representation) for various quantum gates simulated in FakeValencia backend.

with 1 sign bit, 0 bits for the integer part, and 30 bits for the fractional part, we conducted sensitivity tests for the X and H gates across the FakeValencia, FakeAthens, and FakeLima backends. We observed that the fixed-point representation offers a more stable and resilient behavior against bit-flip errors, particularly beyond the initial bits (Fig. 7). After bit 5, the impact of bit flips was considerably lower, with the TVD never exceeding 20% for bits 6 to 31. In the FakeValencia backend, we extended our experiments to additional quantum gates such as SX, CNOT, and rotation gates Rx at $\pi/4$ and $\pi/3$ angles. These tests demonstrated that the impact of bit flips on the gates' amplitude was significantly less pronounced in fixed-point representation compared to floating-point, particularly beyond bit 5 where TVD increases did not exceed 20% for any of the gates (Fig. 8).

Further analysis of the effects of bit flips in the imaginary part of the pulse amplitude showed that for fixed-

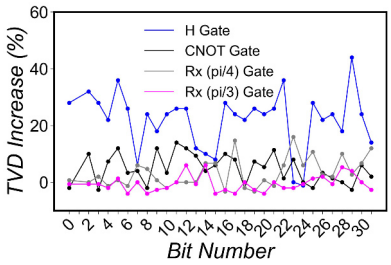

Fig. 6. TVD increase from bit flips in the phase (floating-point representation) for various quantum gates simulated in FakeValencia backend.

point representation, TVD values remain below 25% across all bit positions, whereas for floating-point representation, the TVD values are around 40%. whereas for floating-point representation, the TVD values are around 40%. (Fig. 9).

Lastly, our examination of bit flips on the phase shift values using the ShiftPhase instruction demonstrated that similar to the floating-point representation, the TVD increase from bit flips in the phase values does not follow a consistent pattern across different bit positions (Fig. 10). However, for fixed-point representation, the TVD values never exceeded 20%, whereas in floating-point representation, the TVD values were as high as 50% for some gates.

IV. Conclusion

We conducted a comprehensive analysis of the impact of bit-flip errors in FPGA memories that store amplitude and phase information used to generate quantum gate pulses. This analysis covered both floating-point and fixed-point representations. Our findings identified significant disruptions in quantum operations due to faults in the real part of the amplitude, particularly within the exponent and initial mantissa bits of floating-point representation. Since the pulse amplitude and phase values in quantum gate operations do not vary significantly, most of the bits can be allocated for precision in a fixed-point representation. This makes fixed-point representation more suitable for quantum application, as it offers greater resilience against bit-flip errors within a known range.

References

[1] R. LaPierre, *Introduction to quantum computing*. Springer Nature, 2021.

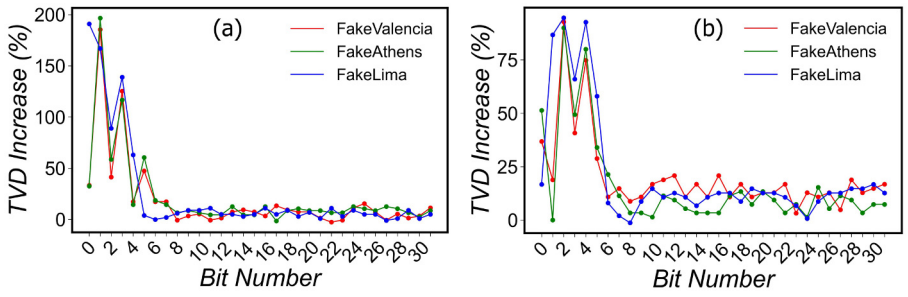

Fig. 7. TVD increase from bit flips in the real part of amplitude (fixed-point representation) for (a) X and, (b) H gates across three simulated backends: FakeValencia, FakeAthens, and FakeLima demonstrating consistent sensitivity trend across backends.

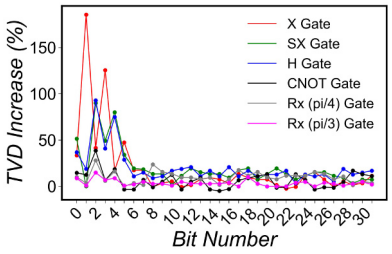

Fig. 8. TVD increase from bit flips in the real part of amplitude (fixed-point representation) for various quantum gates simulated in FakeValencia backend.

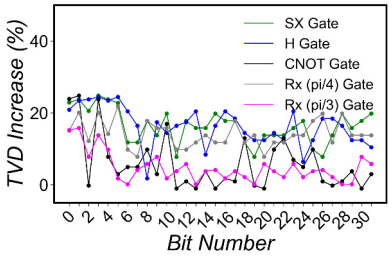

Fig. 9. TVD increase from bit flips in the imaginary part of amplitude (fixed-point representation) for various quantum gates simulated in FakeValencia backend.

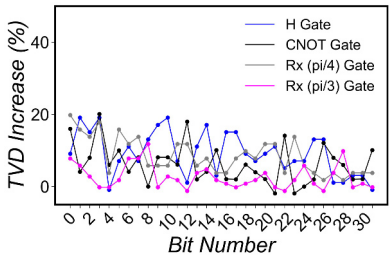

Fig. 10. TVD increase from bit flips in the phase (fixed-point representation) for various quantum gates simulated in FakeValencia backend.

[2] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[3] Y. Xu, G. Huang, J. Balewski, R. Naik, A. Morvan, B. Mitchell, K. Nowrouzi, D. I. Santiago, and I. Siddiqi, "Qubic: An open-source fpga-based control and measurement system for superconducting quantum information processors," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–11, 2021.

[4] S. Maurya and S. Tannu, "Compaqt: Compressed waveform memory architecture for scalable qubit control," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2022, pp. 1059–1077.

[5] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[6] S. Sussman, L. Stefanazzi, K. Treptow, N. Wilcer, C. Stoughton, S. Montella, C. Bradford, G. Cancelo, S. Saxena, H. Arnaldi *et al.*, "The qick (quantum instrumentation control kit): Readout and control for qubits and detectors," in *APS March Meeting Abstracts*, vol. 2022, 2022, pp. B35–005.

[7] K. H. Park, L. H. Nguyen, R. Dumke, S. Efthymiou, C. Hufnagel, Y. P. Tan, K. H. Lau, P. Bore, S. Carrazza, R. P. Budoyo *et al.*, "arxiv: Icarus-q: A scalable rfsoc-based control system for superconducting quantum computers," Tech. Rep., 2021.

[8] R. Vijay, C. Macklin, D. Slichter, S. Weber, K. Murch, R. Naik, A. N. Korotkov, and I. Siddiqi, "Stabilizing rabi oscillations in a superconducting qubit using quantum feedback," *Nature*, vol. 490, no. 7418, pp. 77–80, 2012.

[9] D. Ristè, C. Bultink, K. W. Lehnert, and L. DiCarlo, "Feedback control of a solid-state qubit using high-fidelity projective measurement," *Physical review letters*, vol. 109, no. 24, p. 240502, 2012.

[10] N. Rollins, M. Fuller, and M. J. Wirthlin, "A comparison of fault-tolerant memories in sram-based fpgas," in *2010 IEEE Aerospace Conference*. IEEE, 2010, pp. 1–12.

[11] B. Salami, O. Unsal, and A. Cristal, "Fault characterization through fpga undervolting," in *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2018, pp. 85–853.

[12] C. Carmichael, M. Caffrey, and A. Salazar, "Correcting single-event upsets through virtex partial configuration," *Xilinx Application Notes*, vol. 216, p. v1, 2000.

[13] M. Wirthlin, E. Johnson, N. Rollins, M. Caffrey, and P. Graham, "The reliability of fpga circuit designs in the presence of radiation induced configuration upsets," in *11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2003. FCCM 2003.* IEEE, 2003, pp. 133–142.

[14] Microsemi Corporation, "Single event effects in fpgas," https://www.microsemi.com/document-portal/doc_view/125149-single-event-effects-in-fpgas, 2017, accessed: 2024-05-05.

[15] A. S. Rakin, Y. Luo, X. Xu, and D. Fan, "Deep-dup: An adversarial weight duplication attack framework to crush deep neural network in multi-tenant fpga," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1919–1936.

[16] Xilinx, "Bram ecc instantiate infer," https://support.xilinx.com/s/question/0D52E00006iHmIlSAK/bram-ecc-instantiate-infer?language=en_US, 2020, accessed: May 5, 2024.

[17] C. H. Ho, C. W. Yu, P. Leong, W. Luk, and S. J. Wilton, "Floating-point fpga: Architecture and modeling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 12, pp. 1709–1718, 2009.