FPGA-Based Emulation of Quantum Computing Systems: A Survey of Architectures, Noise Modelling, And Digital Twin Approaches

Mr. Akash Ananda Mendke[1], Sukhdev K Thorat[2], Manojkumar Vilas Kukade[3], Dr. Kailash Baliram Sapnar[4]

[1]*Assistant Professor, Department of Electronics, Fergusson College (Autonomous), Pune*

[2]*Adv. M. N. Deshmukh College Rajur*

[3]*Associate Professor, Department of Computer Science, Fergusson College (Autonomous), Pune*

[4]*Associate Professor, Department of Physics, Fergusson College (Autonomous), Pune*

doi.org/10.64643/ijirt.208360-459

Abstract*—Classical simulation of quantum circuits is fundamentally constrained by the exponential growth of state-vector memory and computation with qubit count, motivating a long-running effort to accelerate quantum-circuit emulation using reconfigurable hardware. Field-programmable gate arrays (FPGAs) offer fine-grained parallelism, deterministic timing, and reprogrammability, making them an attractive substrate for such emulation. A substantial body of work spanning nearly two decades has explored architectures for mapping quantum gate operations onto FPGA fabric. This survey traces that evolution, from early circuit-model emulators through serial-parallel and high-precision architectures to recent iteration-scheduling optimisations that improve energy efficiency relative to CPU and GPU baselines. It then examines a parallel and largely separate thread of research on noise and decoherence modelling, where calibration-driven "digital twins" of physical qubit devices have been constructed primarily in software, using device-specific parameters extracted from hardware calibration data rather than generic noise assumptions. We argue that the convergence of these two threads — real-time-capable FPGA emulation architectures and calibration-driven noise models — remains largely unexplored, and we outline this convergence as a concrete research gap: a hardware-native, real-time noise-adaptive digital twin of a quantum processor. Beyond consolidating scattered literature from VLSI, reconfigurable-computing, and quantum-information venues into a single narrative, this survey is written to directly motivate and ground an ongoing low-cost FPGA-based prototyping effort, positioning it within the broader trajectory of the field. The survey concludes with a discussion of open challenges, including standardised fidelity metrics, scalability beyond small qubit counts, integration with cryogenic control electronics, and hybrid quantum-classical algorithm testbeds, each identified as a concrete direction for future hardware-native digital-twin research.

Index Terms*—FPGA, quantum computing emulation, quantum circuit simulation, decoherence modelling, digital twin, noise-adaptive hardware, reconfigurable computing

I. INTRODUCTION

Quantum computers hold promise for solving certain computational problems beyond the reach of classical machines, yet large-scale, fault-tolerant quantum hardware is not yet routinely available. As a result, most present-day algorithm design, debugging, and testing still happens through classical simulation. This approach runs into a well-known wall: representing an n-qubit state exactly requires 2^n complex amplitudes, so both the memory footprint and the computational cost of a full state-vector simulation scale exponentially with the number of qubits. Overcoming this bottleneck has occupied simulation researchers for decades, producing several competing acceleration strategies multi-core and distributed computing, GPU acceleration, tensor-network methods, and, the focus of this survey, FPGA-based hardware emulation.

FPGAs suit this problem well for a structural reason: quantum gates act as matrix-vector products on pairs of state-vector amplitudes, an operation pattern that is inherently regular and parallel. Because FPGA fabric can be configured into a datapath purpose-built for exactly this pattern, rather than routed through a general instruction pipeline, it can exploit that

regularity far more directly than a CPU or GPU can. This translates into a favourable energy profile for the kind of repetitive, high-throughput arithmetic that dominates quantum-circuit emulation an advantage that becomes more consequential as the circuits being emulated grow larger. This survey pursues two goals. The first is to draw together, in one continuous narrative, an FPGA-emulation literature that is otherwise dispersed across VLSI, reconfigurable-computing, and quantum-information venues. The second is to place this hardware-emulation literature alongside a separate and more recent strand of work on calibration-driven noise modelling and device-specific "digital twins," which to date has been built almost entirely as a software layer (for instance, through Qiskit Aer-style noise models). Our position is that moving this noise modelling onto FPGA fabric so that it becomes part of the emulation itself rather than a step applied afterward is a natural next step that current literature has not yet taken, and it is this direction that motivates the survey as a whole.

II. BACKGROUND: CLASSICAL SIMULATION OF QUANTUM CIRCUITS

Classical approaches to simulating quantum circuits fall into two broad families. State-vector methods keep track of the entire 2^n-dimensional complex amplitude vector and update it gate by gate; the approach is exact and works for any circuit, but its memory and time cost both scales exponentially, which in practice caps simulation at a few tens of qubits on ordinary hardware. Tensor-network methods take a different route, representing the quantum state as a network of lower-rank tensors. Where circuits have limited entanglement true of many circuits of practical interest this representation can bring memory and computation well below the worst-case exponential bound, though at the cost of generality for circuits with dense entanglement.

The FPGA-emulation literature reviewed here has overwhelmingly targeted the state-vector family. The reason is architectural: state-vector updates follow a fixed, predictable memory-access pattern that maps cleanly onto static hardware, whereas tensor-network contraction involves data-dependent structure that is much harder to express efficiently as fixed RTL. Accordingly, the remainder of this survey is scoped to state-vector-based FPGA emulation.

III. EVOLUTION OF FPGA-BASED QUANTUM EMULATION ARCHITECTURES

3.1. Early Circuit-Model Emulation

The earliest efforts in this space focused on demonstrating, at the hardware-description level, that qubit amplitudes and gate matrices could be represented directly in fixed-point logic on FPGA fabric. Building on that groundwork, Aminian et al. developed an automated flow capable of generating register-transfer-level descriptions of a quantum co-processor straight from a circuit specification removing much of the hand-coded RTL effort that earlier approaches required, and in doing so making FPGA emulation practical for larger circuits than manual design would allow.

3.2. Serial-Parallel and Pipelined Architectures

FPGA emulator design is shaped by a persistent trade-off: fully parallel data paths deliver the highest throughput but consume resources that scale exponentially with qubit count, while serial designs conserve resources at the expense of latency. Lee, Khalil-Hani, and Marsono tackled this trade-off head-on with a serial-parallel architecture tested on Quantum Fourier Transform and Grover's search circuits, reporting a linear reduction in resource use compared with earlier fully pipelined designs, without giving up a meaningful speed advantage over equivalent software simulation.

What sets this work apart is its explicit treatment of resource efficiency not just raw speed as a primary design goal, a priority that carries over directly to low-cost FPGA platforms such as the one anticipated for the present author's own prototype work.

3.3. High-Precision and Scalable Architectures

A separate concern in FPGA emulation is numerical accuracy: fixed-point arithmetic can accumulate error over long sequences of gates, a problem that grows more serious precisely as circuits become large or deep enough to be practically interesting. Mahmud and El-Araby addressed this with a scalable, high-precision, high-throughput architecture for unitary-transform computation, later extended by Mahmud et al. with further hardware techniques for unitary transformations more generally.

3.4. Toward Real-Time and General-Purpose Emulators

Pilch and Długopolski took a different framing, presenting their FPGA-based emulator explicitly as a substitute for access to physical quantum hardware rather than purely as a simulation accelerator. This reframes emulation as a learning and testing platform for situations where real quantum devices are scarce, costly, or otherwise inaccessible a motivation that anticipates the digital-twin framing developed later in this survey, even though it predates the term's explicit use in the FPGA-emulation literature.

3.5. Recent Advances: Scheduling and Energy Efficiency

The most recent architectural advance covered here is due to Moawad et al., who observed that conventional FPGA implementations of full state-vector simulation schedule a fixed number of memory-access iterations per gate irrespective of how many control qubits are actually imposed wasting cycles whenever a control condition goes unsatisfied. Their scheduling technique instead precomputes exactly which iterations require a memory access, eliminating that waste. Tested on Quantum Fourier Transform, squaring, and streaming circuits of up to 29 qubits, the approach left resource utilisation essentially unchanged from the baseline while improving both execution time and energy consumption — with the largest gains, close to sevenfold, appearing on circuits whose gates involve many control qubits. Notably, once this optimisation was applied, FPGA energy efficiency in the same study surpassed both CPU and GPU baselines, strengthening the case for energy efficiency as FPGA emulation's central advantage at scale.

IV. COMPARATIVE SUMMARY OF FPGA EMULATION ARCHITECTURES

Table I lays out this architectural lineage in chronological order, tracing the shift from early feasibility demonstrations toward designs optimised in turn for resource use, numerical precision, and energy efficiency.

TABLE I *Comparative overview of representative FPGA-based quantum emulation architectures.*

Work	Architecture / Design Approach	Target Platform	Qubit Scale Demonstrated	Reported Strength
Khalid, Zilic & Radecka (2004)	Direct RTL circuit-model emulation of quantum gates	FPGA (HDL)	Small circuits (proof of concept)	Established feasibility of gate-level FPGA emulation
Aminian et al. (2008)	Automated RTL generation for a quantum co-processor	HPRC / FPGA	Small-to-medium circuits	Reduced manual RTL design effort
Lee, Khalil-Hani & Marsono (2016)	Serial-parallel state-vector architecture	FPGA	QFT and Grover's search case studies	Linear reduction in resource utilisation vs. pipelined designs
Mahmud & El-Araby (2018); Mahmud et al. (2020)	High-precision, high-throughput unitary-transform architecture	FPGA	Scaled multi-qubit circuits	Improved arithmetic precision and throughput
Pilch & Długopolski (2019)	Real-time quantum computer emulator	FPGA	General-purpose circuits	Positioned emulator as a stand-in for physical hardware access
Moawad et al. (2022, 2024)	Iteration-scheduling optimisation for full state-vector simulation	FPGA (OpenCL), compared against CPU/GPU	Up to 29 qubits (application circuits)	Up to ~7x energy-efficiency gain over baseline scheduling
Mueller et al. (2025)	Calibration-driven parametric noise model (software digital twin)	Classical software (validated against a 5-qubit QPU)	5-qubit transmon device	Captured device-specific, time-varying noise behaviour

V. NOISE MODELLING AND THE EMERGENCE OF DIGITAL TWINS

The architectures discussed in Section 3 all emulate ideal, error-free gate operations. Physical quantum processors behave quite differently: finite qubit relaxation time (T1) and dephasing time (T2), together with gate and readout errors, shape circuit outcomes in ways that matter a great deal on today's noisy, intermediate-scale (NISQ-era) devices. Capturing this behaviour accurately has become its own research thread, pursued so far almost entirely through software. Mueller and colleagues, for example, built a parametric error model for a superconducting transmon qubit fitting its parameters from device calibration data and benchmark circuits rather than assuming generic noise and reported that predictions from this model matched measured output distributions from a real five-qubit device closely enough to describe the model as a digital twin of that hardware. What stands out about this and comparable efforts is not just the calibration-driven approach fitting device-specific parameters instead of assuming a generic noise profile, at the cost of needing access to a physical device for that calibration but where they are implemented: entirely within classical software noise-simulation frameworks, with no real-time hardware component. The upshot is that the two literatures surveyed here have grown up largely apart. FPGA-based emulation (Section 3) has optimised for the speed and resource cost of ideal-gate execution; digital-twin noise modelling (this section) has optimised for fidelity within a software model. The two rarely cite, borrow from, or build on one another.

VI. TOWARD A HARDWARE-NATIVE DIGITAL TWIN OF A QUANTUM PROCESSOR

Moving calibration-driven noise modelling onto FPGA fabric would, in principle, combine what each thread does well: the deterministic, real-time, energy-efficient execution that FPGA emulation already offers, paired with the device-specific fidelity that calibration-driven noise models provide. The resulting platform would depart from existing FPGA emulators in one important respect — instead of emulating an idealized, noiseless processor, it would emulate the timing and error behaviour of one particular physical device, updating as that device's calibration data changes, and doing so in real time on reconfigurable hardware rather than as an offline software pass.

This direction fits a pattern visible elsewhere in quantum technology. FPGAs have separately been proposed as a natural substrate for digital twins of quantum-networking hardware, on the strength of their high operating rate and their ability to handle pipelined, parallel control-layer processing. In quantum sensing, digital twins of the noise process itself have been used to drive adaptive control strategies that claw back some of the performance lost to decoherence. Taken together, these threads suggest that "digital twin" is becoming something of a unifying idea across quantum-technology sub-fields and a hardware-native FPGA twin of a gate-based processor's timing and noise behaviour remains one of the more conspicuous gaps in that emerging picture.

VII. OPEN CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Taken together, the preceding sections point to several concrete directions where further work is still needed:

- Embedding noise directly in FPGA hardware: current FPGA emulators run idealised gates; folding parametric T1/T2 and gate-error behaviour directly into the FPGA datapath, rather than layering it on afterward in software, has not yet been demonstrated.
- Scaling past small qubit counts: published FPGA emulation results generally cap out at a few tens of qubits before single-device resource limits become binding, pointing to multi-FPGA and resource-sharing architectures as an open area.
- Agreeing on fidelity metrics: studies that validate emulation or digital-twin accuracy against physical hardware tend to use their own metric of choice (total variation distance being one example), and no shared benchmark yet exists for how close is close enough.
- Linking to cryogenic control electronics: as cryogenic qubit-controller ICs mature, an FPGA-based digital twin could double as a pre-fabrication testbed for controller logic and timing — an application the surveyed literature has not yet taken up.
- Hybrid quantum-classical testbeds: algorithms like VQE and QAOA alternate between classical optimisation and quantum-circuit evaluation; an

FPGA platform able to run both the optimiser and the noise-aware emulator with minimal round-trip delay would extend current single-purpose emulators in a direction not yet explored.

VIII. CONCLUSION

Over roughly two decades, FPGA-based quantum-circuit emulation has moved from early feasibility demonstrations through resource-conscious serial-parallel designs and high-precision arithmetic, arriving most recently at scheduling optimisations that give FPGA platforms a real energy-efficiency edge over CPU and GPU baselines on suitable circuits. Running in parallel, calibration-driven digital-twin models have shown that device-specific noise behaviour can be captured with real fidelity — but almost always in software. Bringing these two threads together a real-time, resource-efficient FPGA architecture paired with calibration-driven, device-specific noise modelling remains a concrete and largely open research direction, and one that a low-cost FPGA-based digital-twin platform is well placed to pursue.

REFERENCES

[1] U. Khalid, Z. Zilic, and K. Radecka, "FPGA emulation of quantum circuits," in *Proc. IEEE Int. Conf. Computer Design: VLSI in Computers and Processors (ICCD 2004)*, San Jose, CA, USA, 2004, pp. 310–315.

[2] M. Aminian, M. Saeedi, M. Saheb Zamani, and M. Sedighi, "FPGA-based circuit model emulation of quantum algorithms," in Proc. IEEE Computer Society Annual Symp. on VLSI, Montpellier, France, 2008, pp. 399–404.

[3] O. Segal, N. Nasiri, M. Margala, and W. Vanderbauwhede, "High level programming of FPGAs for HPC and data centric applications," in Proc. IEEE High Performance Extreme Computing Conf. (HPEC), Waltham, MA, USA, 2014, pp. 1–3.

[4] C. Conceicao and R. Reis, "Efficient emulation of quantum circuits on classical hardware," in Proc. IEEE 6th Latin American Symp. on Circuits & Systems (LASCAS), Montevideo, Uruguay, 2015, pp. 1–4.

[5] Y. H. Lee, M. Khalil-Hani, and M. N. Marsono, "An FPGA-based quantum computing emulation framework based on serial-parallel architecture," Int. J. Reconfigurable Computing, vol. 2016, pp. 1–18, 2016.

[6] A. Kelly, "Simulating quantum computers using OpenCL," arXiv preprint arXiv:1805.00988, May 2018.

[7] N. Mahmud and E. El-Araby, "A scalable high-precision and high-throughput architecture for emulation of quantum algorithms," in Proc. 2018 31st IEEE Int. System-on-Chip Conf. (SOCC), Arlington, VA, USA, 2018, pp. 206–212.

[8] J. Pilch and J. Długopolski, "An FPGA-based real quantum computer emulator," J. Computational Electronics, vol. 18, no. 1, pp. 329–342, Mar. 2019.

[9] T. Bonny and A. Haq, "Emulation of high-performance correlation-based quantum clustering algorithm for two-dimensional data on FPGA," Quantum Information Processing, vol. 19, no. 6, p. 179, Jun. 2020.

[10] N. Mahmud, B. Haase-Divine, A. Kuhnke, A. Rai, A. MacGillivray, and E. El-Araby, "Efficient computation techniques and hardware architectures for unitary transformations in support of quantum algorithm emulation," J. Signal Processing Systems, vol. 92, no. 9, pp. 1017–1037, Sep. 2020.

[11] M. Khalid, N. ul Islam Muhammad, U. M. Khokhar, A. Jafri, and H. Choi, "FPGA based hardware abstraction of quantum computing system," preprint, in review, Aug. 2021.

[12] Y. Moawad, W. Vanderbauwhede, and R. Steijl, "Investigating hardware acceleration for simulation of CFD quantum circuits," Frontiers in Mechanical Engineering, vol. 8, Art. no. 925637, Oct. 2022.

[13] Y. Moawad, A. Brown, R. Steijl, and W. Vanderbauwhede, "Optimising iteration scheduling for full-state vector simulation of quantum circuits on FPGAs," arXiv preprint arXiv:2411.18354, Nov. 2024.

[14] R. Mueller, M. Zanner, M. Schielein, M. Rüfenacht, E. Jennings, and C. Gustiani, "Towards a digital twin of noisy quantum computers: Calibration-driven emulation of transmon qubits," *arXiv preprint arXiv:2504.08313*, 2025.