QIsim: Architecting 10+K Qubit QC Interfaces Toward Quantum Supremacy

Dongmoon Min
dongmoon.min@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

Junpyo Kim
junpyo.kim@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

Junhyuk Choi
junhyuk.choi@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

Ilkwon Byun
ik.byun@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

Masamitsu Tanaka
masami_t@nagoya-u.jp
Nagoya University
Nagoya, Japan

Koji Inoue
inoue@ait.kyushu-u.ac.jp
Kyushu University
Fukuoka, Japan

Jangwoo Kim*
jangwoo@snu.ac.kr
Seoul National University
Seoul, Republic of Korea

ABSTRACT

A 10+K qubit Quantum-Classical Interface (QCI) is essential to realize the quantum supremacy. However, it is extremely challenging to architect scalable QCIs due to the complex scalability trade-offs regarding operating temperatures, device and wire technologies, and microarchitecture designs. Therefore, architects need a modeling tool to evaluate various QCI design choices and lead to an optimal scalable QCI architecture.

In this paper, we propose (1) *QIsim*, an open-source QCI simulation framework, and (2) novel architectural optimizations for designing 10+K qubit QCIs toward quantum supremacy. To achieve the goal, we first implement detailed QCI microarchitectures to model all the existing temperature and technology candidates. Next, based on the microarchitectures, we develop our scalability-analysis tool (QIsim) and thoroughly validate it using previous works, post-layout analyses, and real quantum-machine experiments. Finally, we successfully develop our 60,000+ qubit-scale QCI designs by applying eight architectural optimizations driven by QIsim.

CCS CONCEPTS

• **Computer systems organization** → **Quantum computing**; • **Hardware** → **Emerging tools and methodologies**.

KEYWORDS

Quantum Computing, Cryogenic Computing, Single Flux Quantum (SFQ), Modeling, Simulation, Quantum-Classical Interface

*Corresponding author

ISCA '23, June 17–21, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0095-8/23/06…$15.00
https://doi.org/10.1145/3579371.3589036

ACM Reference Format:
Dongmoon Min, Junpyo Kim, Junhyuk Choi, Ilkwon Byun, Masamitsu Tanaka, Koji Inoue, and Jangwoo Kim. 2023. QIsim: Architecting 10+K Qubit QC Interfaces Toward Quantum Supremacy. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA '23), June 17–21, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3579371.3589036

1 INTRODUCTION

Quantum computing is a new paradigm with a huge potential to solve many classically-intractable problems. However, the expected level of quantum supremacy requires hundreds of fault-tolerant logical qubits, in which each logical qubit consists of hundreds of noisy physical qubits. Therefore, architects aim to build a large-scale fault-tolerant quantum computer (FTQC) which consists of more than 10,000 qubits (10+K qubits).

To build such large-scale quantum computers, architects need a scalable quantum controller consisting of two key components: (1) quantum control processor (QCP), a classical digital processor to support FTQC operations, and (2) quantum-classical interface (QCI), an interfacing electronics to control and measure qubits. While recent works have focused on developing a scalable QCP to support up to 10+K qubits [7], it has remained as a critical challenge to develop a scalable QCI design to realize the quantum supremacy. Note that exploring scalable QCI architecture is not only the important but also the timely topic as shown in IBM's roadmap [33] (i.e., IBM targets to realize 10+K qubit quantum computers in 2026).

Despite the importance of developing a scalable QCI, architects suffer from two scalability constraints in the QCI design. First, in quantum computers using superconducting qubits, the limited cooling capacity of a dilution refrigerator constrains the QCI's scalability. For example, today's room-temperature QCI can connect only a few hundred electrical cables to the qubit chip due to the tight power budget of the qubit-located 20mK stage (i.e., 20μW). Second, a scalable QCI should ensure a low logical error rate to run practical FTQC applications. Even if a QCI can magically manage millions of qubits, it cannot realize quantum supremacy without satisfying the

required error rate. Although these constraints can change with the technology evolution, they are currently the major scalability bottlenecks in developing next-generation QCIs. Therefore, for the scalable QCI, architects should minimize both power dissipation inside the refrigerator and the logical error rate.

For this purpose, researchers and industries have recently proposed various ideas in operating temperatures, wire and device technologies, and microarchitecture designs. However, it is extremely challenging to develop a scalable QCI mainly due to its wide design space and complex scalability trade-offs. For example, using a QCI running at 4K can reduce the wire heat, but it can suffer from the significant device power consumption in the refrigerator. In addition, several wire (e.g., photonic link) and device (e.g., RSFQ) technologies have emerged, but they can degrade the scalability due to the additional heat at 20mK. As another example, microarchitectural decisions favoring lower power consumption (e.g., frequency multiplexing) can adversely increase the logical error rate.

Therefore, architects are now in dire need of a modeling tool to analyze the scalability trade-offs of various QCI designs and to develop a scalable QCI design successfully. The tool should accurately evaluate the scalability of various QCI designs for different operating temperatures, wire and device technologies, and microarchitectures. However, due to the lack of such tools, a scalable QCI design has not been proposed yet.

In this paper, we propose (1) *QIsim*, an open-source QCI simulation framework, (2) novel architectural optimizations to realize 10+K qubit QCI designs, and (3) our example 60,000+ qubit-scale QCI design which applies the optimizations.

To achieve the goal, we first implement the detailed QCI microarchitectures using all major candidates (i.e., 300K QCI with coaxial cable/microstrip/photonic link, 4K CMOS and SFQ-based QCI). To make our tool released, we implement both the currently absent microarchitecture components and the previously proposed but not open-sourced components. We validate their functionalities with RTL and Hamiltonian simulations.

Next, we develop a QCI simulation framework to accurately evaluate the QCI's scalability in four steps. (1) It estimates the frequency and power consumption for the target temperature, technology, and microarchitecture configurations. (2) It runs cycle-accurate simulations to obtain their gate timing and hardware activation information. (3) Based on the simulation results, it estimates the logical error rate and the power consumption in the refrigerator. (4) These results finally lead to the manageable qubit scale of the target design. We thoroughly validate QIsim using various previous works, post-layout analyses, and real quantum machine experiments.

Third, driven by the modeling and analysis, we identify that the scalability of existing QCI designs is severely limited. For example, the scalability of 300K QCI is limited to less than 700 qubits due to the wire's passive/active load. 4K CMOS or RSFQ-based QCIs might not suffer from the wire heat, but their high device power consumption limits the scalability to less than 700 qubits. From the analysis, we aim to improve the scalability of 4K QCIs, in which we can apply architecture-level innovations to resolve their device-power bottlenecks.

Finally, we successfully develop our example 60,000+ qubit QCIs by applying eight microarchitectural optimizations to resolve the observed bottlenecks as follows. (1) We propose five microarchitectural optimizations to reduce the power consumption of 4K CMOS and mK-4K RSFQ designs. These optimizations allow the QCI designs to achieve the near-term target scalability (i.e., 1,152 qubits). (2) We propose two error-minimizing and one instruction-compressing optimizations to lower-power CMOS (i.e., voltage and technology scaled) and SFQ (i.e., ERSFQ) devices available in the near future. As a result, we successfully present the scalable CMOS and SFQ-based QCI architectures managing over 60,000 qubits.

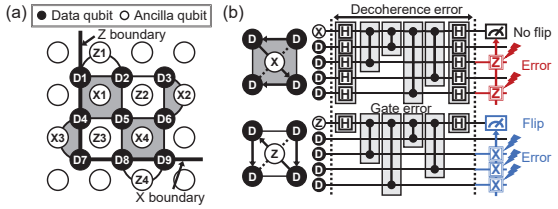


Figure 1: (a) Surface-code patch (d=3) and (b) ESM for X and Z-ancilla qubit

We believe that with the proposed 10+K qubit-scale QCIs available, our QCI modeling tool and novel architecture optimizations will contribute to realizing the true quantum supremacy.

In summary, this paper makes the following contributions:

- **QC interface modeling framework**: To the best of our knowledge, this is the first work to model, validate, and evaluate various QCIs for large-scale FTQC systems.
- **Holistic scalability analysis**: We show the importance of estimating the power consumption and error rate simultaneously to build a scalable FTQC system.
- **Novel architectural optimizations**: We provide eight insightful optimizations to improve the scalability of QCI.
- **Fully-implemented QCI designs**: This is also the first study to propose the fully-implemented 4K CMOS and SFQ-based QCIs to support 10+K qubits successfully.
- **Open-source tool release**: We release our QIsim framework to help architects to explore future QCI designs.[1]

2 BACKGROUND AND MOTIVATION

2.1 Fault-tolerant quantum computing (FTQC)

We briefly introduce the essential FTQC backgrounds with surface code [21, 51], one of the prominent QEC protocols.

2.1.1 Logical qubit overview. Fig. 1(a) shows a surface-code patch representing a logical qubit, which consists of two types of physical qubits: data qubit (solid circle; D) containing state information, and ancilla qubit (open circle; X or Z) used for extracting error information. We can make and control a logical qubit with error syndrome measurement (ESM) and quantum error correction (QEC).

2.1.2 Error syndrome measurement. The quantum circuit called ESM entangles the ancilla qubits with their adjacent data qubits, and then measures them (Fig. 1(b)). Even though data qubits are susceptible to gate and decoherence errors, the ESM discretizes them to X or Z errors. In addition, the ancilla qubit measurements (or error syndromes) provide hints to identify the types and locations

[1] https://github.com/SNU-HPCS/QIsim

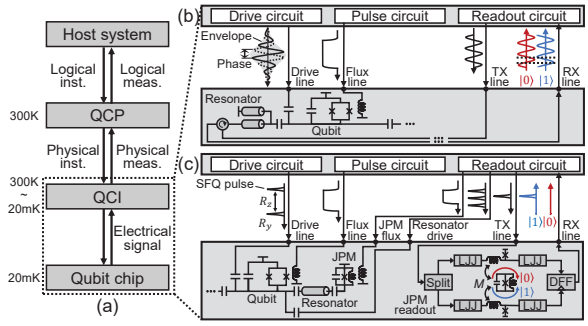


Figure 2: (a) Fault-tolerant quantum computer (FTQC) system with (b) CMOS and (c) SFQ-based QCIs

of these errors. Specifically, we can identify data-qubit errors based on the error syndromes because each X (or Z) error in data qubits flips their adjacent Z-ancilla (or X-ancilla) measurements.

2.1.3 Quantum error correction. A chain of errors flips the error syndromes in its two endpoints. For QEC, error decoding algorithms first pair the nearest flipped syndromes to find the error chain, and then apply the error-correcting X (or Z) gates to the data qubits on the identified error chains. However, the QEC can fail (i.e., logical error) when the physical error rate is too high. We can reduce the logical error rate by increasing the number of data qubits on the boundary (i.e., code distance, d), but this approach requires much more physical qubits for each logical qubit (i.e., $2 \cdot (d+1)^2$).

2.1.4 Fault-tolerant logical operation. We can apply any logical quantum operations with the iterative ESMs. Specifically, we can express arbitrary quantum circuits mainly with the sequence of multi-qubit Pauli-product measurements (i.e., PPM). Then, we can execute these PPMs with the equivalent lattice surgery, which dynamically merges and splits surface-code patches through the ESM operations [51].

2.2 Fault-tolerant quantum computer system

Fig. 2(a) shows the overview of our target FTQC system, where the quantum control processor (QCP) and quantum-classical interface (QCI) control superconducting qubits (i.e., flux-tunable Transmon) at 20mK [7, 23]. First, the QCP takes a logical-qubit level instruction (e.g., PPM) from the host system and translates it into the physical-qubit level instructions (e.g., H, CZ). Next, the QCI generates and sends the electrical signal corresponding to the received instructions. In addition, the QCI analyzes the reflected signal from the 20mK stage for the measurement. Lastly, QCP takes the physical measurements and performs QEC or logical measurement.

With the recent effort, previous works extensively explored QCP components [16, 17, 20, 22, 31, 36, 67, 69, 74] and proposed a scalable QCP architecture [7]. Therefore, it is a natural next step to design a scalable QCI, which has not been actively explored from the architect's perspective. For that reason, this work focuses on the QCI while assuming the QCP ideally operates at room temperature.

2.3 Technologies to implement QCIs

There are mainly two device technologies for the QCI: (1) CMOS and (2) superconductor single flux quantum (SFQ).

2.3.1 CMOS-based QCI. As the most natural approach, architects proposed a CMOS-based QCI that controls and measures qubits by using analog voltage signals (Fig. 2(b)) [44].

First, for the single-qubit gate, the drive circuit generates a microwave whose frequency corresponds to the target qubit's frequency. The microwave's envelope and phase determine the rotation angle and axis in the Bloch sphere. Second, for the two-qubit gate, the pulse circuit generates a DC voltage pulse to match one target qubit's frequency to that of the other qubit. Lastly, for the measurement, the readout circuit sends a microwave tuned to the coupled resonator's frequency (TX), interprets the reflected microwave, and differentiates the target qubit's state (RX).

2.3.2 SFQ-based QCI. The superconducting SFQ logic family is an ultra-fast and low-power digital technology that utilizes a voltage pulse as its information carrier. Recent studies have suggested the ideas to control and measure a superconducting qubit with the SFQ pulse (Fig. 2(c)) [32, 39, 50, 54].

First, the drive circuit realizes an arbitrary single-qubit gate by applying an SFQ pulse stream (or bitstream) to qubits. Specifically, each SFQ pulse incurs a small amount of y-axis rotation (R_y) in the Bloch sphere, while the qubit's state rotates around the z-axis (R_z) during the idle time. Thus, we can realize arbitrary single-qubit gates by carefully engineering the bitstream. Second, using SFQ devices for the two-qubit gate, the pulse circuit generates the same DC voltage signals as the CMOS-based approach. Lastly, the readout circuit measures qubit states with Josephson photomultiplier (JPM) and SFQ devices operating at 20mK. The readout circuit first converts the target qubit's state to JPM's state and then detects the converted JPM state by using the mK-located JPM readout circuit.

2.4 Scalability constraints of QCIs

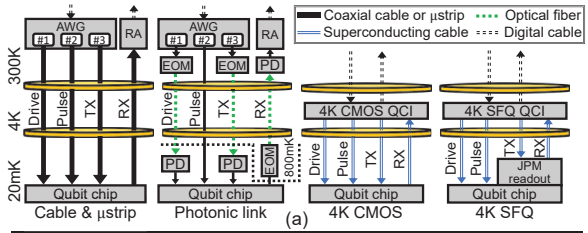
To successfully support the increasing number of qubits, the QCI should meet two scalability constraints.

2.4.1 Power budget of the dilution refrigerator. First, the limited cooling capacity of the dilution refrigerator can constrain the scalability. The QCI's interconnect and hardware overhead proportionally increase with the qubit scale because it should provide the control signal to every physical qubit. However, the refrigerator's power budget is too tight to connect even hundreds of coaxial cables to the qubit-located 20mK stage (i.e., 20μW at 20mK [45]). Therefore, for the scalable QCI design, we should minimize its power dissipation inside the refrigerator.

2.4.2 Target logical-qubit error rate. Second, we should ensure a low logical error rate for the successful execution of target applications. When we scale a quantum algorithm to achieve quantum supremacy, the number of required logical qubits and logical operations increases, and thus the target logical error rate correspondingly decreases. Therefore, we should minimize the physical-qubit level errors (e.g., gate error, decoherence error) to satisfy the decreasing logical error target.

2.5 Various QCI designs with complex trade-offs

For scalable QCI development, recent studies have proposed various ideas with complex scalability trade-offs.



Technology		CMOS			SFQ
Temperature		300K		4K	4K
Interconnect		Coaxial cable & µstrip cable	Photonic interconnect	Superconducting cable	Superconducting cable
Power inside dilution refrigerator	+ Wire	-	Low wire passive load	Negligible wire heat load	Negligible wire heat load
	+ Device	No device power	No device power	No device power at 20mK	Lower device power at 4K
	− Wire	High wire passive load	High wire active load	-	-
	− Device	-	-	High device power at 4K	JPM readout power at 20mK
Logical error		Microarchitecture dependent			

(b)

Figure 3: (a) Various QCIs' overviews and (b) their complex scalability trade-offs

2.5.1 Temperature & Technology. Fig. 3(a) introduces the existing major temperature and technology candidates for the QCI design.

300K CMOS with electrical cable. Today's CMOS-based QCI operates at 300K and sends microwaves to the qubits through coaxial cables or microstrips [1]. Although it is straightforward, the passive heat of cables (i.e., wire passive load) limits the scalability.

300K CMOS with photonic link. The limitation of 300K-mK electrical cables motivates a new approach using photonic link [46, 80]. As the optical fiber's passive load is negligible, this approach can mitigate the passive heat of today's QCI. However, the photodetector (PD) to restore the microwave at 20mK can degrade the scalability due to its high active load (i.e., wire active load).

4K CMOS with superconducting electrical cable. To resolve the challenges of 300K QCIs, industry and academia have tried to run CMOS-based QCI inside the refrigerator, especially at 4K [5, 59]. When the QCI operates at 4K, we can meaningfully reduce wire heat by using superconducting electrical cables (e.g., superconducting coaxial cable [14] or microstrip [73]). However, its device power can limit the scalability because the power budget is also quite limited at 4K (i.e., 1.5W [45]).

4K SFQ with superconducting electrical cable. For the lower 4K device power, the SFQ-based QCI has emerged as another promising candidate [54]. In principle, SFQ devices can have lower power than conventional CMOS devices. However, as the SFQ-based readout requires the JPM readout circuits at 20mK (Fig. 2(c)), the SFQ-based QCI's scalability can be even lower than other approaches.

2.5.2 Microarchitecture. Orthogonal to the temperature and technology choices, numerous microarchitectural decisions can affect the trade-offs between power consumption and error. For example, recent QCI leverages the frequency multiplexing that enables 32 qubits to share a single drive circuit [79]. The multiplexing can reduce the device power and wire's passive load inside the refrigerator. However, we can suffer from severe decoherence errors due to the serialized single-qubit gates.

2.6 Research challenges toward 10+K qubit QCIs

Unfortunately, researchers still cannot find a clear direction due to the huge design spaces and their complex trade-offs (Fig. 3(b)). Therefore, we should resolve the following research challenges.

2.6.1 Absence of the full-microarchitecture implementation. Architects require the detailed and convincing microarchitectures to analyze the scalability of the major QCI technologies. However, it is impossible because there is no fully-implemented QCI microarchitecture for several major QCI technologies (i.e., 4K CMOS, SFQ).

2.6.2 Absence of a scalability analysis tool. To accurately analyze the pros and cons of each decision, architects need a scalability-analysis tool, which can evaluate QCIs with different operating temperatures, device and wire technologies, and microarchitectures. However, developing the tool is extremely challenging as it requires (1) cross-technology device models, (2) microarchitecture-dependent error models, and (3) cycle-accurate simulators. For this reason, there is no QCI analysis tool despite its critical importance.

2.6.3 Absence of design guidelines. Along with the tool, architects should clarify the directions for designing a 10+K qubit QCI. For this purpose, it is necessary to analyze the existing ideas' scalability bottlenecks and propose architectural solutions to resolve them. However, neither such analysis nor the guideline exists yet.

In this paper, we resolve these challenges as follows. First, we implement the detailed microarchitectures for every existing temperature and technology candidate (Section 3). Next, we develop a QCI simulation framework (*QIsim*) (Section 4) with thorough validations (Section 5). Finally, by using the tool, we propose architectural solutions and present 60,000+ qubit QCI designs (Section 6).

3 QCI MICROARCHITECTURE DESIGNS

In this section, we introduce our detailed microarchitecture setup and implemention of the existing QCI candidates.

3.1 300K CMOS-based QCI with electrical cable

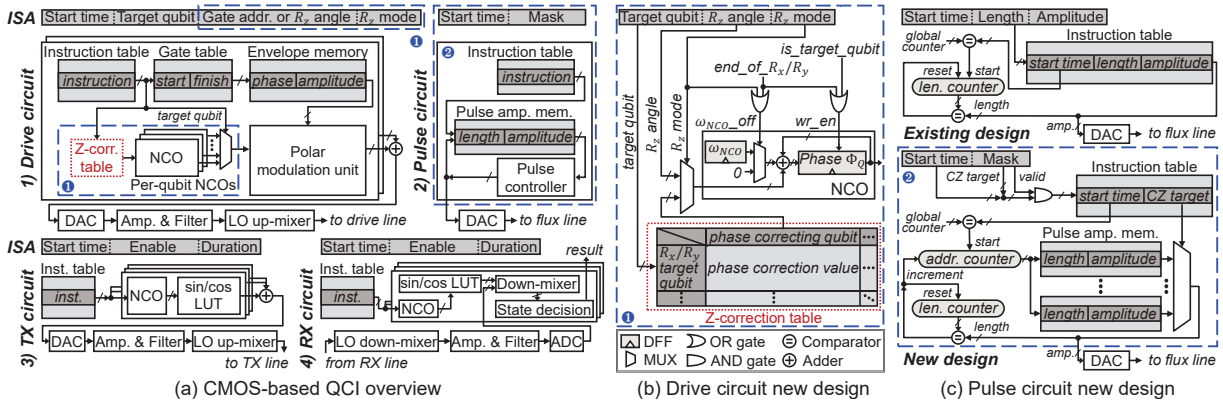
We set the 300K electrical-cable baseline by following today's quantum machines (Fig. 3(a)) [1]. For the drive, pulse, and TX, we adopt the 14-bit arbitrary waveform generator (AWG). In addition, we share a single AWG and cable with 32 and 8 qubits for drive and TX, respectively, following the state-of-the-art frequency-multiplexed drive and readout schemes [40, 76]. For RX, we use one cable for 8 qubits to forward the reflected signal to the readout analyzer (RA).

3.2 300K CMOS-based QCI with photonic link

We adopt the photonic-link architecture of previous works (Fig. 3(a)) [46, 80]. For the drive and TX, per-qubit 14-bit AWG generates a microwave, and the electro-optic modulator (EOM) converts it to an optical signal. Then, the optical fiber transmits the signal to the photodetector (PD) at 20mK, which restores the original microwave. With the reflected readout microwave, the mK-located EOM generates a modulated optical signal and forwards it to 300K. As there has been no two-qubit gate demonstration using photonic links, we use the per-qubit AWG and microstrip for the pulse circuit.

3.3 4K CMOS-based QCI

Even with many recent proposals, there has been no fully integrated 4K CMOS-based QCI for Transmon qubits. Therefore, we implement the full QCI microarchitecture for our 4K CMOS baseline (Fig. 4). For that purpose, we first reproduce the overall microarchitecture of Intel's Horse Ridge I & II [59, 76]. Next, we newly design the



(a) CMOS-based QCI overview (b) Drive circuit new design (c) Pulse circuit new design

Figure 4: (a) Our baseline 4K CMOS-based QCI overview with (b) newly-developed drive-circuit NCO and (c) pulse circuit

necessary but previously absent components and re-design the units without their detailed implementation (blue and red-colored in Fig. 4, respectively).

3.3.1 Drive circuit. The drive circuit generates a microwave whose frequency, envelope, and phase correspond to the target-qubit frequency, rotation angle, and rotation axis, respectively.

Existing design and limitation. Horse Ridge I's drive circuit can apply arbitrary x and y-axis rotations (i.e., $R_x(\phi)$, $R_y(\phi)$) to two of 32 qubits through one drive line simultaneously (i.e., frequency multiplexing; FDM). For the purpose, its digital part consists of two banks, and each bank can apply one gate at a time.

$$\begin{cases} I[n] = A[n]\cos(\omega_{\mathrm{NCO}} n + \Phi_{\mathrm{Q}} + \Phi_{\mathrm{G}}[n]) \\ Q[n] = A[n]\sin(\omega_{\mathrm{NCO}} n + \Phi_{\mathrm{Q}} + \Phi_{\mathrm{G}}[n]) \end{cases} \quad (1)$$

The drive circuit's ISA consists of *start time*, *target qubit*, and *gate table address* fields (Fig. 4(a)). At *start time*, the digital bank initiates generating I/Q digital samples ($I/Q[n]$ in Eq. (1)). First, the per-qubit NCO iteratively generates the target qubit's angle values ($\omega_{\mathrm{NCO}} n + \Phi_{\mathrm{Q}}$) based on the qubit-specific rotating frequency (ω_{NCO}) and accumulated phase (Φ_{Q}). Simultaneously, we obtain the target gate's address range (i.e., from *start* to *finish* in the gate table) and iteratively read the gate-specific phase ($\Phi_{\mathrm{G}}[n]$) and amplitude ($A[n]$) values from the envelope memory. Next, by using them, the polar modulation unit outputs the desired n-th I/Q samples ($I/Q[n]$). Finally, the analog part generates the final microwave by up-converting the digital signal to the qubit-frequency microwave.

However, this drive circuit and ISA cannot support virtual z-axis rotation ($R_z(\phi)$), which is required for low-error R_z operation. In addition, it does not provide a detailed microarchitecture for Z correction. Specifically, in FDM, the microwave targeting one qubit rotates other qubits' states around z-axis (i.e., AC-Stark shift [44]). The Z correction is necessary to handle the AC-Stark shift.

New design. To support the virtual R_z and Z correction, we propose the new NCO design with the extended ISA (Fig. 4(b)). The main idea of the virtual R_z is to realize $R_z(\phi)$ by accumulating the angle value (ϕ) to the target qubit's phase (Φ_{Q}). To implement the idea, we add 1-bit R_z *mode* to the ISA and reuse *gate table address* field as *target* R_z *angle*. We also add the new datapath to accumulate the angle value to the target qubit's NCO if R_z *mode* is 1.

To realize the Z correction, we further extend the NCO by adding the Z correction table. The table contains all the error-correcting phase values we should add to other qubits after applying $R_x(\phi)$ or $R_y(\phi)$ to the target qubit. We add the control path in NCO to take the Z-correction values at the end of $R_x(\phi)$ and $R_y(\phi)$ operations.

3.3.2 Pulse circuit. To entangle two flux-tunable Transmons with CZ gate, the pulse circuit generates a DC voltage pulse.

Existing design and limitation. Horse Ridge II's per-qubit pulse circuit can apply CZ gate with the unit-step voltage pulses (Existing design in Fig. 4(c)). For the purpose, its ISA provides *start time*, *length*, and *amplitude* information. From *start time*, the digital part iteratively reads the same target *amplitude* from the instruction table while increasing the length counter. Next, when the length counter reaches the target *length* value, it stops generating the digital samples. Meanwhile, the DAC converts the generated digital samples into the analog unit-step voltage signal.

However, this design suffers from severe errors because it cannot support arbitrary ramp-up/down waveform. In our detailed Hamiltonian simulation, we confirm that the unit-step voltage almost cannot realize the CZ gate.

New design. To efficiently support the arbitrary ramp-up/down, we newly design the pulse circuit with the custom ISA (new design in Fig. 4(c)). Our ISA includes *mask* and *start time*, where *mask* consists of the per-qubit 1-bit *valid* and 2-bit *CZ target*. *CZ target* indicates the target qubit among the four neighbors in the 2D lattice.

When *valid* is one, our pulse circuit takes *start time* and *CZ target* into the instruction table. From *start time*, it iteratively reads the target pulse amplitude memory, which includes a series of custom amplitude and length values for each neighbor. Specifically, we first obtain the initial amplitude and length values for the target CZ gate. Next, we forward the amplitude to the DAC every cycle while increasing the length counter. When the length counter reaches the current length value, we increase the address counter, obtain the next amplitude and length, and repeat the previous steps. Note that our memory overhead is negligible as we need an arbitrary waveform only for the short ramp-up/down period.

3.3.3 TX circuit. TX circuit generates resonator-frequency microwaves and applies them to resonators for a predefined duration.

Existing design. We reproduce Horse Ridge II's TX circuit, but changes its FDM level following state-of-the-art CMOS-based readout [40]. Thus, our TX circuit supports up to eight parallel readout by applying eight microwaves to a single TX line (Fig. 4(a)).

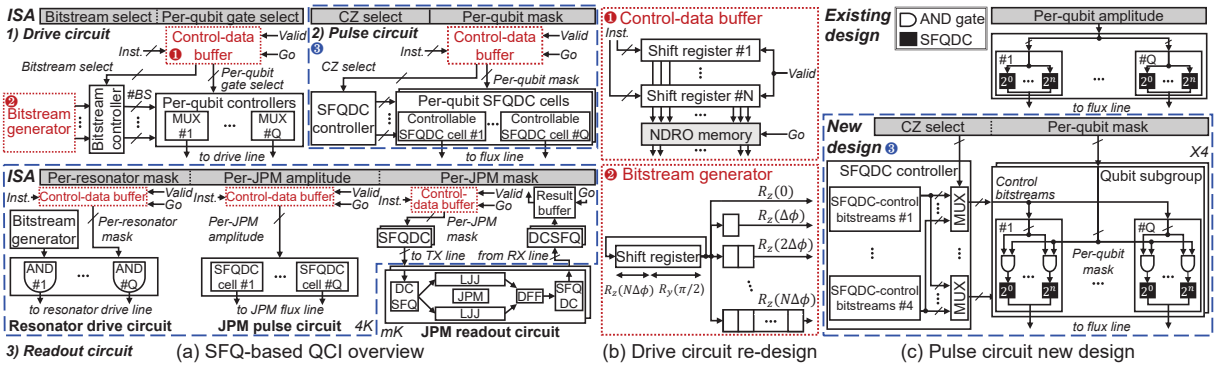


Figure 5: (a) Our baseline SFQ-based QCI overview with (b) re-designed drive circuit and (c) newly-designed pulse circuit

Specifically, by following the instruction, the enabled digital banks generate digital samples for the given *duration*. For this purpose, each bank consists of the NCO and sin/cos LUT where the ω_{NCO} and phase are tuned for the resonator. Then, the analog part up-converts the digital samples into the final multi-tone microwave.

3.3.4 RX circuit. The RX circuit determines the qubits' states by interpreting the reflected microwave from the 20mK stage.

Existing design. We reproduce the Horse Ridge II's RX circuit that supports parallel readout using eight per-qubit digital banks and one shared analog circuit (Fig. 4(a)). First, following the instruction, the analog part down-converts the reflected microwave into low-frequency digital samples for the specified *duration*. Next, from the digital samples, each enabled digital bank extracts the DC I/Q samples for the target qubit using the NCO, sin/cos LUT, and down mixer. Finally, the state-decision unit determines the qubit's state based on the obtained and accumulated DC I/Q samples.

We adopt Horse Ridge II's bin-counting method for our baseline state-decision unit because its error rate is the lowest among representative methods (as shown in Fig. 19). With the bin-counter memory, the decision unit counts the number of I/Q samples for each I/Q plane coordinate. At the end of readout, it divides the I/Q plane with the state-discriminating line, compares the number of samples in the $|0\rangle$ and $|1\rangle$ planes, and determines the readout value.

3.4 4K SFQ-based QCI

Fig. 5 shows our 4K SFQ-based QCI microarchitecture highlighting the newly-designed (blue-colored) and re-designed (red-colored) parts. We assume that all data (or ancilla) qubits have the same frequency and share the circuits in Fig. 5.

3.4.1 Drive circuit. To support arbitrary rotation, the SFQ drive circuit generates the SFQ pulse stream corresponding to $R_y(\frac{\pi}{2}) \cdot R_z(\phi)$ gate, which is well known SFQ-friendly basis gate [39, 50].

Existing design and limitation. We reproduce the representative SFQ-based QCI, DigiQ [39], whose drive ISA consists of *bitstream select* and *per-qubit gate select*. First, the control-data buffer takes instructions and forwards them to bitstream and per-qubit controllers every cycle. Next, among the gates from the bitstream generator (i.e., $R_y(\frac{\pi}{2}) \cdot R_z(\phi)$ with different ϕ), the bitstream controller selects #BS pulses using *bitstream select* and broadcasts to per-qubit controllers. Finally, based on *per-qubit gate select*, the per-qubit controllers select and apply a pulse stream to each qubit.

However, there exist no detailed control-data buffer and bitstream generator microarchitectures. To accurately evaluate the scalability, their detailed microarchitectures are necessary.

Re-design. We clarify the detailed control-data buffer and bitstream generator microarchitectures. As Fig. 5(b) shows, our control-data buffer consists of the shift registers and NDRO memory. First, by using the *Valid* bit as the clock signal, the shift registers collect the next-instruction bits while executing the current instruction. Next, when it receives *Go* signal, the NDRO memory takes the instruction bits from shift registers. Note that the NDRO memory can broadcast the signal every cycle as it does not lose the stored data. We also use this design in our pulse and readout circuits.

Next, Fig. 5(b) shows our bitstream generator design. Our key insight is that, as the idle time determines ϕ, we can realize various $R_z(\phi)$ just by inserting the different number of DFFs. Therefore, we design our bitstream generator to share a single shift register holding the $R_z(N\Delta\phi) \cdot R_y(\frac{\pi}{2})$ pulse and broadcast it to the output shift registers equipped with different number of DFFs.

3.4.2 Pulse circuit. The pulse circuit should generate the DC pulse with the accurate ramp-up and ramp-down voltage shapes [44].

Existing design and limitation. DigiQ proposed the SFQ-based pulse circuit [39] (Existing design in Fig. 5(c)). By taking the per-qubit amplitude bits, the per-qubit SFQDC cells turn on the corresponding numbers of SFQDC and generate the target DC pulse.

However, this pulse circuit can generate the unit-step voltage pulses only, which incurs significant CZ errors.

New design. We propose the novel SFQ-based pulse circuit supporting AWG (Fig. 5(c)). Our key design decision is to place the SFQDC-control bitstreams at 4K, which enables the AWG with the negligible 300K-4K bandwidth. For the parallel ESM, we assume four qubit subgroups with different CZ frequencies, and thus we should simultaneously provide four types of CZ waveforms to the subgroups. For that purpose, our ISA consists of the per-subgroup *CZ select* and *per-qubit mask*. First, by taking *CZ select*, the SFQDC controller selects the SFQDC-control bitstream for each subgroup and broadcasts the pulses to corresponding SFQDC cells. Next, based on *per-qubit mask* and control bitstreams, the appropriate SFQDC cells are turned on, and apply the AWG DC pulse to qubits.

3.4.3 Readout circuit. The readout circuit converts the qubit states to the JPM state and then readouts the JPM states. Based on the JPM readout result, we can guess the qubit states.

Existing design and limitation. There is no proposal for the full SFQ-based readout procedure and its required 4K SFQ circuits. The previous work only proposed the mK-located SFQ circuit for JPM-state readout (i.e., mK-located JPM readout circuit) [32].

New design. We propose the full SFQ readout procedure and the required 4K SFQ circuits (Fig. 5(a)). Our insight is that, if we follow the existing CMOS-based JPM readout while generating the equivalent SFQ pulse stream for each step, we enable full SFQ-based readout. Thus, we follow the stages of CMOS-based readout while designing the equivalent pulses and circuits. According to the CMOS-based studies [28, 58], the JPM-based readout consists of four steps: resonator driving, JPM tunneling, JPM readout, and reset. We first architect their required pulse streams and circuitries and then verify their functionality with Hamiltonian simulation.

i) Resonator driving. First, we convert the qubit state to the resonator state by applying enough resonator-frequency energy. Even though the previous works demonstrate this step with the microwave [58], we need an SFQ-pulse-based implementation for the full SFQ-based readout. At this point, our insight is that we can apply the required energy by delivering the SFQ pulse train with the resonator-rotating period (i.e., the reciprocal of the resonator frequency). The reason is that, when we apply Fourier Transform, the periodic pulse train is converted into the energy at the corresponding frequency in the frequency domain. We implement our resonator drive circuit by modifying our drive circuit (Fig. 5(a)).

ii) JPM tunneling. Second, we convert the resonator state to the JPM state. For this step, we should apply the DC pulse to match the JPM frequency with the resonator frequency. We implement the JPM pulse circuit by modifying our pulse circuit. Note that the JPM state determines the direction of the circulating current inside the JPM. We use it in the next step.

iii) JPM readout. Third, we read the JPM state by adopting the previously proposed mK-located JPM readout circuit (Fig. 5(a)) [32]. This circuit distinguishes the JPM state using the delay difference between the two LJJ trains. The LJJ train is an inductance-biased transmission line that slowly transports an SFQ pulse with zero static power. More importantly, the circulating JPM current reversely affects the pulse-transfer speed of each coupled LJJ train. As a result, in the JPM state $|1\rangle$ (or $|0\rangle$), the data (or clock) arrives at the connected DFF earlier, and thus the DFF generates an SFQ pulse (or no pulse). We also implement the 4K circuits to send (and receive) the SFQ pulse to (and from) the mK circuit.

iv) Reset. Lastly, we reset the JPM state for the next readout by turning off the JPM-pulse circuit's output DC signal.

4 QISIM: QCI SIMULATION FRAMEWORK

To evaluate QCIs with various temperatures, technologies, and microarchitectures, we develop a QCI scalability-analysis framework, QIsim. Fig. 6 shows the overview of QIsim.

4.1 Circuit model

The circuit model takes the architecture configuration and target technologies (e.g., 4K CMOS, RSFQ, ERSFQ) as its input, and predicts the static & dynamic powers of each QCI component.

4.1.1 Verilog code generator. The Verilog code generator generates the Verilog code of the target hardware following the input architecture configuration. For this purpose, we implement our CMOS QCI's digital parts and SFQ circuits in the fully parameterized Verilog. We validate their functionality with IVerilog (CMOS) [77] and Vivado simulation (SFQ).

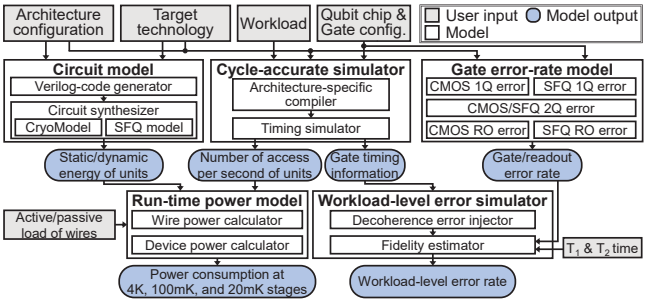


Figure 6: Our QCI simulation framework overview (QIsim)

4.1.2 Circuit synthesizer. With the Verilog code, the circuit synthesizer runs technology-aware synthesis and derives the static and dynamic power of each QCI component. To model the CMOS digital part, we use the validated 300K~4K CMOS modeling tool, CryoModel [7, 9, 47, 55]. CryoModel takes the target temperature, voltage level (V_{dd}, V_{th}), Verilog code, and memory specifications (e.g., capacity) as its inputs, synthesizes the input circuits with commercial tool (i.e., Design Compiler Topographical Mode), and predicts the frequency and power of logic and memory devices. For the CMOS analog, we use the analog power values of the previous works (i.e., [76] for drive and TX, [59] for pulse, [40] (LNA, mixer) and [59] (amplifier, ADC) for RX). We choose the Transmon-compatible analogs and confirm their seamless integration with the digital parts through digital-analog-Hamiltonian simulation (Section 4.4). We also confirm their power-level and frequency-range compatibility. For SFQ, we adopt XQsim's SFQ model [7]. The model predicts the frequency and power of RSFQ & ERSFQ circuits by synthesizing the Verilog code with Yosys [78] and then applying SFQ-specific optimizations to the generated netlist.

4.2 Cycle-accurate simulator

Our simulator generates gate-timing information and each unit's number of access per second (i.e., activity factor). First, we compile the input OpenQASM-based workload [15] to the architecture-specific executable and then run the cycle-accurate simulation. Specifically, the simulator stores all instructions in *per-qubit FIFO instruction queue* and then dequeues (or executes) the dependency-free heads while recording the execution timing. In addition, the simulator finds the activated hardware units for each instruction execution and calculates their activity factors.

We use *remaining time table* to calculate the execution timing. The table tracks the remaining times of the currently running instructions. For each instruction-executing trial, the simulator reads the table to check whether the target qubit (and counterpart for CZ gate) is idle (i.e., true dependency[64]), and whether the drive circuit is available for multiplexing and #BS (i.e., structure hazard).

In this manner, our cycle simulator accurately derives the gate-timing information and hardware activity factors.

4.3 Runtime-power model

The runtime-power model calculates the realistic power dissipation at major temperature domains (e.g., 4K, 100mK, 20mK). Specifically,

the model derives each domain's static power by multiplying the input per-unit static power and the per-wire passive load with the number of required units and wires, respectively. In addition, it derives each domain's dynamic power by multiplying the input activity factor to the per-access dynamic energy of each unit.

4.4 Gate error-rate model

Fig. 7 shows the overview of our gate error-rate model for CMOS and SFQ-based QCIs. The inserted digital waveforms, SFQ bitstreams, and data are generated by our model.

4.4.1 CMOS single-qubit gate error. With the architecture, gate, and qubit information (e.g., frequency, anharmonicity), the model predicts the error rate of the CMOS-based single-qubit gate. First, with the input gate time and waveform information, we generate realistic digital I/Q samples by following the target bit precision (❶). Second, the model generates a realistic and noisy analog microwave by inserting a Gaussian noise with the specified SNR value [76]. Third, our model performs Hamiltonian simulation by applying the noisy microwave to the qubit Hamiltonian and obtains the noisy unitary matrix (❷). Lastly, by comparing the ideal unitary with the noisy unitary matrix, our model derives the target gate's error rate.

4.4.2 SFQ single-qubit gate error. Based on the qubit information and QCI's frequency (f_{QCI}), we model the error rate of SFQ QCI's basis single-qubit gates (i.e., $R_y(\frac{\pi}{2}) \cdot R_z(\phi)$). First, we model the $R_z(\phi)$ error by considering the limited precision for the ϕ values. For $R_y(\frac{\pi}{2})$ error modeling, we follow the bitstream-optimizing method of the previous work [50]. Specifically, we optimize the bitstream by iteratively running Hamiltonian simulation while inserting (or removing) a pulse pair until its error rate does not decrease (❸,❹). As a result, we can obtain the low-error bitstream and its gate error.

4.4.3 CMOS/SFQ two-qubit gate error. Our model predicts the CZ error rate similar to the CMOS single-qubit gate. First, we obtain the ideal DC pulse for the CZ gate by using Baidu Quanlse framework [3] with the gate delay and target qubit information. Next, we derive the realistic noisy pulse by injecting the errors considering the amplitude's bit precision and thermal noise. Finally, by running Hamiltonian simulation for the coupled-qubit system, we obtain the noisy unitary matrix and compare it with the ideal matrix.

4.4.4 CMOS readout error. Our model predicts the CMOS readout error in three steps. First, the model obtains the reflected microwave from the stochastic Hamiltonian simulation of the qubit-resonator-coupled system [6, 38] by using the realistic square-envelope TX microwave and resonator decay (❺). Second, for RX modeling, we down-convert the reflected microwave to the DC I/Q components while injecting various RX hardware noises (e.g., TWPA [61], HEMT [11], digital/analog noise [59]) (❺). Finally, with the obtained DC I/Q samples, our state-decision model derives the qubit state and predicts the readout error (❻). We support two state-of-the-art state-decision methods: bin-counting method [59] and single-point method [1, 44]. The single-point method averages I/Q samples and determines the qubit state by comparing the single-average point's location with the discriminating line.

4.4.5 SFQ readout error. By taking the qubit, resonator, and JPM information, our model predicts the latency and error of each step.

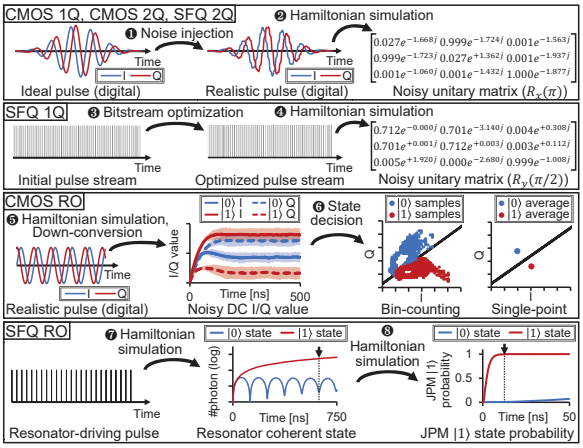


Figure 7: Gate error-rate model overview

i) Resonator driving. By simulating the coupled resonator's Hamiltonian, our model first obtains the drive time at the error-saturating point [28] (❼). Then, it derives the resonator's coherent state for each initial qubit state (i.e., $|0\rangle$, $|1\rangle$) at the point.

ii) JPM tunneling. Second, the model derives the JPM's $|1\rangle$ state probability for each resonator coherent state. For that purpose, we run a detailed Hamiltonian simulation using the Lindblad master equation of resonator-JPM-coupled systems [27] (❽).

iii) JPM readout. We faithfully model the mK JPM-readout circuit and its interaction with JPM, run the JoSIM simulation [19], and obtain the latency and failure rate under realistic thermal noise of AIST fabrication. For the circuit parameters, we adopt the values of the previously developed mK JPM-readout circuits [30, 32].

iv) Reset. For the reset stage, we use the error rate and delay of the previous CMOS-based experiment [58] because the reset is independent of the technology (i.e., just turning off the JPM flux).

4.5 Workload-level error simulator

Our workload-level error simulator predicts the workload-level fidelity based on the input gate-timing information and gate & readout error rates. First, by using the timing information from the cycle-accurate simulator, the decoherence error injector inserts the identity gate to all qubits for every specified period (e.g., 100ns). Then, the fidelity estimator runs Qiskit error simulation [68] by applying the decoherence error rate to the injected identity gates, and applying the gate & readout error rates to other operations. We use Pauli channel for both errors [21, 70, 72] because it ensures high accuracy with fast simulation in FTQC regime [24].

5 QISIM VALIDATION

5.1 4K CMOS and SFQ power validation

5.1.1 4K CMOS circuit power. We validate our 4K CMOS model by comparing its results with the peak power of 22nm Intel Horse Ridge I & II. For the fair comparison with Horse Ridge I, we exclude our newly-implemented circuits. We use the analog power of Horse Ridge II for the RX and TX circuits for the same reason.

$$P_{dyn} = \frac{1}{2}CV_{dd}^2 f \propto \frac{1}{2}(C_g wl)V_{dd}^2(\frac{1}{\tau_{tr}}) \quad (2)$$

As CryoModel supports 45nm technology only, we scale the frequency and dynamic power results to those of 22nm technology

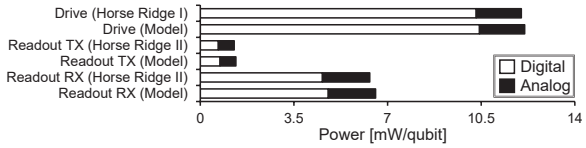


Figure 8: CMOS validation result (vs. Horse Ridge I & II)

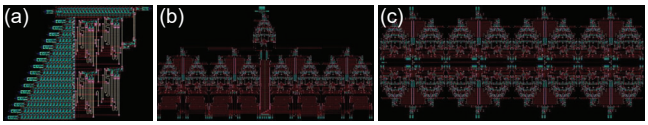


Figure 9: Layout of (a) bitstream generator, (b) bitstream controller, and (c) per-qubit controller with AIST process

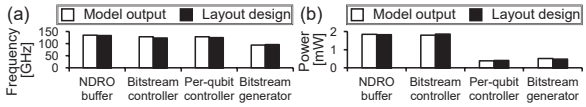


Figure 10: RSFQ validation result: (a) frequency and (b) power

following Eq. (2) [41] and ITRS roadmap [10]. We exclude the pulse circuit as Intel does not disclose its power. Fig. 8 shows the validation results with 5.1% of maximum error in the RX. Note that our model frequency is the same as that of the Horse Ridges (2.5GHz) because we give the frequency as the design objective of synthesis.

5.1.2 SFQ circuit power. We validate our 4K SFQ model by comparing its results with the post-layout analysis. With the AIST Nb 9-layer fabrication process [57], we design the layouts of four most power-hungry circuits (Fig. 9). Their specifications correspond to the drive circuit using 21-bit bitstream (i.e., 5-bit $R_y(\pi/2)$, 16-bit $R_z(\phi)$) to support eight qubits with the #*BS* of eight. Fig. 10 shows our model's accuracy for the frequency and power estimation with 6.7% and 7.2% of maximum error, respectively.

5.2 Gate error validation

We validate our gate-error model by comparing it with state-of-the-art IBMQ machines and previous works, which report the minimum error rate for each category. We validate the model with two approaches. First, for SFQ 1Q gate, 2Q gate, and SFQ readout validations, we compare our gate-error model with the reported gate-error rate excluding the decoherence error [50, 58, 66]. On the other hand, as the references only report the decoherence-included error rate for CMOS 1Q gate and CMOS readout [34], we extend our error models to include the decoherence errors. For CMOS 1Q gate, we first run our gate-error model with the noises induced by QCI and microwave components, and then inject the decoherence error to the obtained qubit state following Bloch-Redfield model in [44]. For CMOS readout, we include the decoherence effect in the Lindblad operator while following the same methodology in Section 4. Lindblad operator, used in Hamiltonian simulation of qubit-resonator coupled systems [6], is the operator for reflecting the interaction between environment and target quantum system. For the Bloch-Redfield model and Lindblad operator setups, we use the T1 and T2 times reported in the references to reflect the target systems' environmental noises.

Table 1: Gate error rate validation result

Type	Reference	Error (Ref.)	Error (Model)
CMOS 1Q ★	*ibm_peekskill (Q21)*	6.59E-5	6.07E-5
SFQ 1Q	[50]	1.37E-5	1.51E-5
2Q	[66]	9.00E-4	1.09E-3
CMOS readout ★▲	*ibm_washington (Q117)*	1.50E-3	1.47E-3
SFQ readout ▲	[58]	6.00E-3	6.10E-3

★Include decoherence ▲Without state preparation

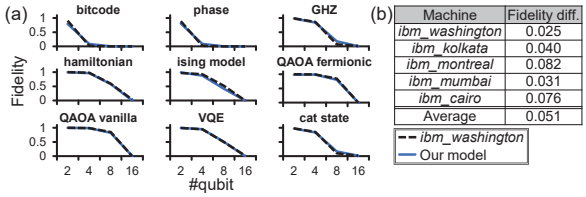


Figure 11: Workload-level fidelity validation result: (a) *ibm_washington* and (b) five IBM machines on average

Even for the component without demonstration (i.e., SFQ 1Q and SFQ readout), we compare our model with the best references. For example, we validate our SFQ readout error with the previous CMOS microwave-based experiment [58] because our SFQ pulses are equivalent to the microwave in the frequency domain. Meanwhile, we exclude the JPM readout validation because neither our results nor the previous studies observe any error. We also skip the JPM reset as our model directly adopts previous work's error rate.

Table 1 summarizes our validation results with the referred quantum computers and literature. As Table 1 shows, our model well matches the reference results with 10.2% of maximum error (SFQ 1Q). Note that our model's CZ error is within the experimental error range ($9 \cdot 10^{-4} \pm 7 \cdot 10^{-4}$) reported in the previous work [66].

5.3 Workload-level error validation

We validate our workload-level error simulator with end-to-end workload fidelity on various IBMQ machines [34]. We run seven SupermarQ [71] and two ScaffCC benchmarks [37] with up to 16 qubits, as they show zero fidelity in larger scales. For T1 & T2 times and gate & readout errors, we use those from the target IBMQ machines. We validate our physical-error prediction as there is no fault-tolerant machine. We validate the model's fidelity prediction rather than the state distribution itself because the realistic error-channel and experimental-noise modelings for state distribution are more complex and less scalable compared to our approach [29, 44]. Due to their complex effect to qubit state, developing the accurate error-channel and noise models is an active research area orthogonal to this work. Note that the main purpose of our model is to predict the high-level metrics (e.g., logical error, workload-level fidelity) to derive the QCI scalability.

Fig. 11 shows the average-fidelity difference between IBMQ machines and our model with 5.1% of low average-fidelity difference. This result also indirectly validates the cycle-accurate simulator, which provides the gate-timing information used in the decoherence-error modeling.

6 ARCHITECTING A SCALABLE QCI

In this section, driven by QIsim, we provide the design directions to develop the 10+K qubit QCI. We introduce our analysis setup (Section 6.1), identify the current bottlenecks (Section 6.2), and propose 10+K qubit QCIs with eight optimizations (Section 6.3).

Table 2: Scalability analysis setup

Quantum operation specification									
Tech.	300K/4K CMOS			SFQ					
Operation type	1Q	2Q	RO	1Q	2Q	Resonator driving	JPM tunneling	JPM readout	Reset
Error*	8.17E-7	7.8E-4	1.00E-3	1.18E-4	1.09E-3	7.8E-3		0	7.00E-3
Latency (ns)	25	50	517	25	50	578.2	12.8	4	70

Wiring power of QCIs (passive/active▲ per cable under 100% activation)				
	Coaxial cable [45]◇	Microstrip [18]	Photonic link [46, 80]	4K microstrip [73]
4K	1mW / 7.9μW	315μW / 7.9μW	250nW★ / -	315μW / 7.9μW
100mK	400nW / 7.9nW	210nW / 7.9nW	0.1nW★ / -	0.1nW★ / 7.9nW
20mK	13nW / 0.79nW	4.3nW / 0.79nW	0.003nW / 790nW (PD)	0.003nW / 0.79nW

Cooling capacity [45]			Clock frequency		Decoherence time●	
4K	100mK	20mK	4K CMOS	SFQ	T1	T2
1.5W	200μW	20μW	2.5GHz	24GHz	122μs	118μs

*Without decoherence ★Project from the 20mK data by following the ratio of coaxial cables
▲-76dBm of peak-driving power [5] with the active-load minimizing attenuator setup (i.e., 0-20-10-10-20dB attenuation for 50K-4K-1K-100mK-20mK stages) proposed in [45]
●*ibm_mumbai* (2022.11.03) which has the highest quantum volume [34]
◇For 4K-mK superconducting coaxial cable, we use 7.4 times lower passive load than 300K coaxial cable, by comparing [14] with [13] of similar attenuation.

6.1 Scalability analysis setup

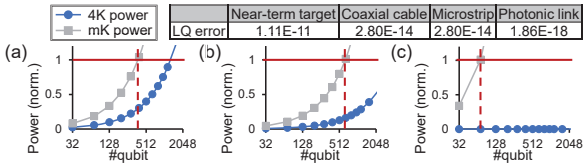
Target scalability & logical error. Following Google's FTQC roadmap [4], our near-term goal is to increase the logical-qubit distance (i.e., d) until 1+K qubits (i.e., d = 23; 1,152 qubits), and the long-term goal is to increase the number of logical qubits with the same d. For the near term, our target logical error is $1.11 \cdot 10^{-11}$, enough to run the smallest quantum chemistry algorithm (i.e., Jellium N=2) with a 99% success rate. For the long-term, we target to support 54 logical qubits (62,208 qubits) with the logical error of $1.69 \cdot 10^{-17}$, enough to run Jellium N=54 with a 99% success rate. We calculate the target errors based on the number of T gates to run Jellium [43] and the well-known target logical error equation [51]. Note that Jellium N=54 is classically intractable and thus its execution indicates quantum supremacy [2].

Workload. We run ESM for all logical-qubit patches because it shows the peak power consumption when running arbitrary FTQC workloads (as shown in Section 2.1). After running ESM, we derive the logical error rate by generating physical X & Z errors from QIsim and applying them to the well-known error model [25].

Technology & physical error setup. We summarize our setup in Table 2. For 4K CMOS, we use 14nm technology to show its full potential because the 14nm node is the latest node demonstrated at 4K [12]. We obtain the 4K CMOS results with FreePDK 45nm [65] and scale with the same methodology as Section 5.1.1. We also apply the power gating to 4K CMOS for the same reason. For CMOS envelope memories, we used sufficiently large memory to support non-identical pulse envelopes. Specifically, we adopt Intel's specification (7.65KB/qubit in total [59, 76]), which is enough to support eight drive, four pulse, and one TX envelopes per qubit of our setup (for 2.5GHz sampling with 25ns, 50ns, and 517ns duration). For SFQ, we use MITLL SFQ5ee library [63] to make our artifact an open-source, and apply mK-enabled I_c scaling (i.e., $0.01 \cdot I_c$) to mK devices following the previous works [32, 54]. We set every gate delay to its error-saturating point.

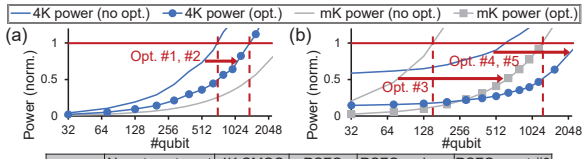
6.2 Scalability analysis of the existing QCIs

6.2.1 Scalability of 300K QCI. We first evaluate the scalability of 300K QCIs. Fig. 12 shows their scalabilities with 4K and mK powers and logical-qubit errors (LQ error). We normalize the 4K & mK powers to the cooling capacity of each temperature stage. Among



	Near-term target	Coaxial cable	Microstrip	Photonic link
LQ error	1.11E-11	2.80E-14	2.80E-14	1.86E-18

Figure 12: Scalability of 300K QCI using (a) coaxial cable, (b) microstrip, and (c) photonic link



	Near-term target	4K CMOS	RSFQ	RSFQ + share	RSFQ + opt.#3
LQ error	1.11E-11	1.56E-16	4.13E-16	3.50E-7	1.34E-13

Figure 13: Scalability of 4K QCI using (a) 4K CMOS and (b) RSFQ technology

the two mK powers (i.e., 100mK, 20mK), we only show the higher value in each QCI for better presentation.

First, even though we adopt the state-of-the-art FDM (i.e., 32 for drive, 8 for readout), the scalabilities of coaxial cable and microstrip are limited to 400 and 650 qubits due to their huge wire passive load at 100mK. Second, even with the negligible passive load of optical fibers, 300K QCI with photonic link can support only 70 qubits due to the huge active load of photodetectors at 20mK.

In summary, the 300K QCIs cannot support even 1,000 qubits due to the huge wire passive or active load. From this perspective, 300K QCIs have little room for architectural innovations because their scalabilities entirely depend on the wire technology and material. Therefore, we aim to investigate the 4K QCIs as they can utilize more scalable superconducting electrical cables [14, 73].

6.2.2 Scalability of 4K QCI. Fig. 13 shows the scalabilities of 4K QCIs with the 4K & 20mK powers and logical error rate. For the near-term case studies, we use the commercial superconducting-coaxial cable [14] as 4K-mK interconnects. Thanks to the much thinner shape enabled by lower resistivity, its passive load is 7.4 times lower than the 300K cable [13] with similar attenuation.

However, the 4K CMOS QCI cannot support more than 700 qubits due to the significant device power at 4K. In this case, the mK power does not limit the scalability thanks to the low heat load of the superconducting cable. Meanwhile, the RSFQ QCI's scalability is limited to less than 160 qubits due to the huge power consumption at 4K and 20mK.

In summary, different from the intuition, the 4K QCIs achieve similar or even lower scalability than 300K QCIs mainly due to their huge 4K & mK device power consumption. However, we have many opportunities to reduce the device power with effective architectural optimizations. In the remaining sections, we target to increase their scalabilities with various architectural innovations.

6.3 4K QCIs for near-term target scalability

6.3.1 1+K qubit QCI with 4K CMOS. We design the 4K CMOS QCI for our near-term target, 1,152 qubits. From our analysis, we observe that the digital parts of the RX and drive circuits account for a significant portion of the 4K device power (54.7% and 13.3%).

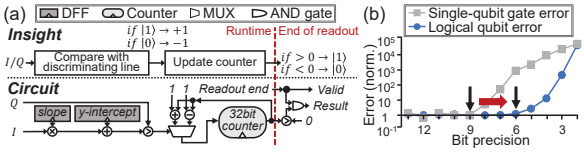


Figure 14: Opt-#1, #2: (a) Decision unit without bin-count memory and (b) single-qubit gate error and logical-qubit error for various bit precisions

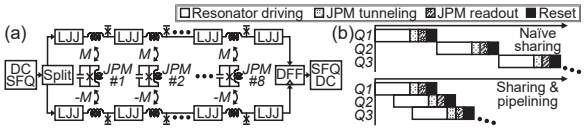


Figure 15: Opt-#3: (a) readout sharing and (b) pipelining

As the 4K device power is the scalability bottleneck, we propose two optimizations to reduce each of them.

Opt-#1. Decision unit without bin-counter memory. Our first optimization is to remove the power-hungry bin-counter memory of the RX state-decision unit. Intel Horse Ridge II and our baseline accumulate the number of I/Q samples for each coordinate in the per-qubit memory and compare them at the end of readout. However, this method consumes significant power because it requires iteratively accessing the per-qubit 32KB memory twice per cycle. Note that 32KB memory setup (i.e., $(2^7 \cdot 2^7$ I/Q coordinates)·16-bit counter [59]) is the error-saturating point (i.e., 7-bit I/Q value) derived from our simulation, and Horse Ridge II utilizes even larger I/Q plane memories than ours (i.e., 128KB in maximum) [59].

Our insight here is that we do not need such a large memory if we compare each I/Q sample's location with the state-discriminating line every cycle. Fig. 14(a) shows our idea for the decision unit without memory, which requires only a 32-bit counter for the same precision and functionality. With this optimization, we reduce the RX and total 4K device power by 88.4% and 48.3%, respectively.

Opt-#2. Lower bit precision for drive circuit. Our second optimization is reducing the digital-bit precision of the drive circuit. Fig. 14(b) shows the single-qubit gate error without decoherence and logical-qubit errors for various bit precisions. As the 9-bit precision is enough to saturate the single-qubit gate error, the previous works use 8∼10 bits for their digital output [5, 12, 75, 79]. However, our insight here is that we can further reduce the precision for lower power consumption, as the single-qubit gate error affects the logical error much less than others (e.g., CZ error). Specifically, Fig. 14(b) shows the logical-qubit error is saturated at the 6-bit precision. Therefore, we adopt the 6-bit precision and reduce the drive digital and total power by 30.9% and 4.1%, respectively.

In summary, our two optimizations resolve the 4K device power bottleneck, and the 4K CMOS QCI achieves the 1+K qubit scalability (1,399 qubits in Fig. 13(a)). Note that the novelty of Opt-#1, #2 comes from breaking the widely-adopted QCI design conventions. With holistic analysis enabled by QIsim, we identify that the conventions in fact are wrong design guidelines which limit the scalability.

6.3.2 1+K qubit QCI with RSFQ. Next, we develop the RSFQ-based QCI supporting 1+K qubits with three optimizations.

Opt-#3. Shared and pipelined JPM readout. We should reduce the mK device's static power because it dominates the mK power

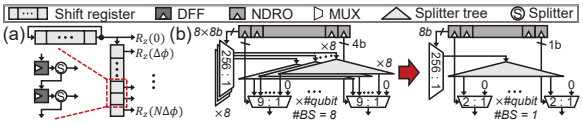


Figure 16: Opt-#4, #5: (a) Low-power bitstream generator and (b) drive-circuit controllers

(99.7%). In this point, our third optimization is sharing the single readout circuit with many JPMs. Fig. 15(a) shows our JPM readout circuit shared by eight JPMs. We maintain the number of LJJs for each JPM to minimize the interactions between adjacent JPMs, and reduce the LJJ's inductance (from 40pH to 4pH) for the low readout delay. Note that 4pH is the widely-used inductance scale in MITLL and AIST libraries. In our JoSIM simulation under thermal noise, our circuit operates without error with 13ns latency. As LJJs do not consume static power thanks to the inductance biasing [30], our readout sharing greatly reduces the mK static power by eight times.

However, the naïve sharing can incur significant logical errors due to the serialized readout. Specifically, the eight serialized readouts take 5,320ns (Naïve sharing in Fig. 15(b)) and increase the logical-qubit error from $4.13 \cdot 10^{-16}$ to $3.50 \cdot 10^{-7}$ (Fig. 13). To resolve this problem, we propose pipelining the readout (Sharing & pipelining in Fig. 15(b)). Our key insight is that we can overlap multiple readouts in a pipelined manner if we do not overlap the JPM readout stages with the JPM writing stages (i.e., JPM tunneling, reset) as shown in Fig. 15(b). With this scheme, we resolve the mK power bottleneck while achieving comparable readout latency (1,255ns) and logical error ($1.34 \cdot 10^{-13}$) to the baseline.

Opt-#4, #5. Low-power bitgen. and controllers. To reduce the 4K power, we optimize the bitstream generator (i.e., bitgen.) and drive-circuit controllers because the drive circuit dominates the 4K power (71.7%). First, we propose the low-power bitgen. by replacing its power-hungry 256 output shift registers (i.e., for $R_z(\phi)$ with 256 different ϕ values) with one splitter-equipped shift register (Fig. 16(a)). As a result, we reduce the bitgen. and 4K total power by 98.2% and 23.2%. Second, we propose the low-power controller design (Fig. 16(b)). From our analysis, we observe that FTQC workloads require only a few types of single-qubit gates in parallel at the same time, different from NISQ workloads (*#BS*=8 in [39]). Specifically, even though we reduce *#BS* from 8 to 1, the execution time remains almost the same when we run ESM and PPM. As the RSFQ-based multiplexer consumes significant power, we can reduce the 4K power by 43.8% with the reduced *#BS*.

With the three optimizations, we successfully achieve our near-term scalability with RSFQ-based QCI (1,248 qubits in Fig. 13(b)).

6.4 4K QCIs for long-term target scalability

Finally, we design 4K QCIs supporting 10+K qubits. For the long-term case studies, we use superconducting microstrip by assuming that the prototyped microstrip becomes quite mature [73].

6.4.1 10+K qubit QCI with advanced 4K CMOS. First, we design a 10+K qubit QCI with 4K CMOS. Fig. 17(a) shows the scalability of 4K CMOS with our long-term target (i.e., 62,208 qubits), where the 4K device power limits the scalability again. To further reduce the power, we apply the technology (i.e., from 14nm to 7nm) and voltage scalings to the 4K CMOS QCI. The minimum technology node

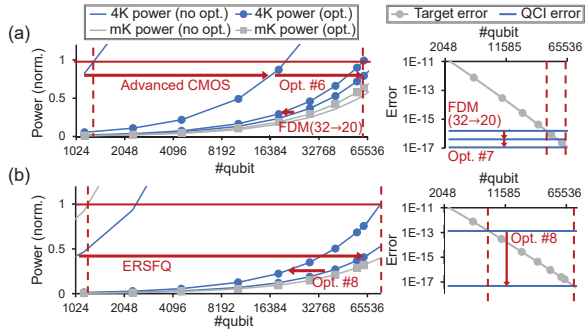


Figure 17: Scalability of (a) advanced 4K CMOS and (b) ERSFQ

operating at 4K has been rapidly reduced (from 28nm (2019)[5] to 14nm (2022) [12]), and the nearly-eliminated leakage current at 4K correspondingly enables the threshold (V_{th}) and operating voltage (V_{dd}) scaling. When we minimize the power without increasing frequency, the technology and voltage scalings reduce the 4K power by 4.15 and 16 times [7], respectively. Fig. 17(a) shows the potential power reduction and scalability with the advanced 4K CMOS. However, even if we successfully scale the 4K device power, the 4K power and logical-qubit error still limit the scalability. To resolve these bottlenecks, we propose two optimizations.

Opt-#6. FTQC-friendly instruction masking. First, to reduce the 4K wire power, we propose the FTQC-friendly instruction masking (Fig. 18). As shown in Fig. 18(a), the wire power dominates the overall power (81.2%) due to the inefficient single-qubit gate ISA of the baseline and Horse Ridge (i.e., 42 bits per operation [59]). Our instruction masking reduces the required bit by compressing the existing ISA to *instruction select* and *per-qubit mask* (Fig. 18(b)). More importantly, to further reduce the bandwidth, we change the basis-gate set from $R_x(\frac{n\pi}{4})$, $R_y(\frac{n\pi}{4})$, R_z of Horse Ridge I to $R_y(\frac{\pi}{2})\cdot R_z(\frac{n\pi}{4})$. Our key insight here is that, in the surface code with lattice surgery, all the two adjacent single-qubit gates are always $H\cdot R_z(\frac{n\pi}{4})$ form, which can be compressed into one $R_y(\frac{\pi}{2})\cdot R_z(\frac{n\pi}{4})$ instruction [21, 51, 52]. As a result, we reduce the wire bandwidth and power by 93% and resolve the 4K power bottleneck.

Opt-#7. Fast multi-round readout. Next, to reduce the logical-qubit error, the simplest but powerful way is reducing the FDM degree to minimize the gate serialization. By reducing the drive circuit's FDM degree within the 4K power budget (from 32 to 20), we achieve 3.85 times lower logical-qubit error (Fig. 17(a)). However, as the logical error is still much higher than our target, we need an effective error-minimizing scheme with little power consumption.

To reduce the error, we propose the fast multi-round readout based on two observations. First, in most cases, we can accurately determine the qubit state in the short readout time (e.g., 98.6% accuracy within 267ns). Second, most errors occur when the difference between the number of two states' samples (i.e., diff.) is near zero.

Inspired by these observations, our scheme iteratively tries short readout rounds for the faster readout (Fig. 19(a) and algorithm in Fig. 19). First, after the resonator ring-up stage (117ns), we collect the readout I/Q samples during 50ns (one readout round). Next, if the diff. is outside of the predefined range, we determine the qubit state immediately. However, if the diff. is within the predefined range (i.e., $threshold - range \sim threshold + range$), we try one more readout round (50ns) as the current round can be vulnerable to the

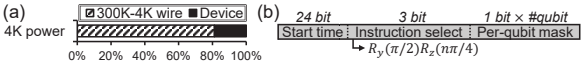


Figure 18: Opt-#6: (a) 4K-power breakdown and (b) FTQC-friendly instruction masking

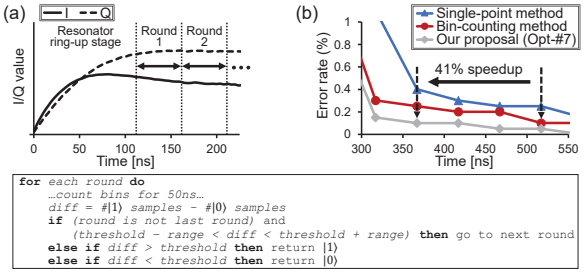


Figure 19: Opt-#7: (a) Multi-round readout and (b) error rate of various methods

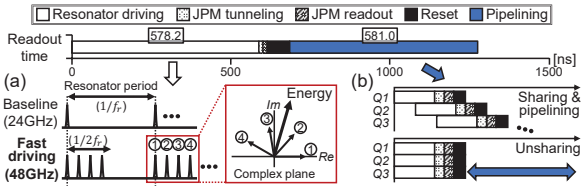


Figure 20: Opt-#8: (a) Fast resonator driving and (b) unsharing

error. By using this method, we achieve 40.9% faster readout with the same readout error (Fig. 19(b)). As the fast readout significantly suppresses the decoherence error during ESM, we reduce the logical-qubit error by 3.62 times and successfully achieve our long-term scalability target (i.e., 63,883 qubits in Fig. 17(a)).

6.4.2 10+K qubit QCI with ERSFQ. Next, we aim to achieve our long-term target with SFQ-based QCI. When we target 10+K qubits, the 4K and mK powers again limit the scalability (Fig. 17(b)).

To reduce the power, we assume the energy-efficient RSFQ (ERSFQ [42]) technology becomes quite mature. With its inductance biasing, the ERSFQ consumes no static power and resolves the power bottlenecks (Fig. 17(b)). However, as the logical error becomes the bottleneck again, we should further reduce the error.

Opt-#8. Fast resonator driving and unsharing. To reduce the error, we propose the fast SFQ-based readout scheme. We target to reduce the resonator-driving and pipelining overheads, which occupy huge portions in readout latency (46.1% and 46.3% in Fig. 20).

First, we propose the fast resonator driving (Fig. 20(a)). In the resonator's frequency domain, each SFQ pulse corresponds to a vector, whose amplitude means the resonator-driving energy. The vector rotates 2π for a resonator period ($\frac{1}{f_r}$). Our insight is that we can boost the resonator driving by pushing more SFQ pulses within a half resonator period. To put more pulses, we selectively increase the resonator-driving circuit's frequency to its maximum value (48GHz). As a result, we reduce the driving time from 578.2ns to 230.9ns with the same error. Next, we unshare the JPM readout circuit again, because the sharing is no more required in zero-static-power ERSFQ. With the fast driving and unsharing, we reduce the logical error by 28,355 times, and successfully support 82,413 qubits.

Note that our optimizations are the first QCI-level solutions which (1) exploit the FTQC regime's characteristics (Opt-#2,#5,#6),

Table 3: Current status and maturity of QCI technologies

	Technology type			Interconnect type		
Gate type	300K CMOS	4K CMOS	4K SFQ	300K cable	4K μstrip	Photonic
1Q gate	E [1, 34]	D [40, 76]	D [49]	E [1, 34]	C [73]	D [46, 80]
2Q gate (CZ)	E [1]	C [59]	C [39]	E [1]	C [73]	A
Readout	E [1, 34]	C [40, 59]	A [54]	E [1, 34]	C [73]	D [46, 80]

A: The full approach does not exist / B: Purely theoretical / C: Circuit-level implementation
D: Qubit control demonstration / E: Large-scale system demonstration (>50 qubits)

(2) improve the readout with the generally-applicable architectural idea (Opt-#3,#7), and (3) break the widely-adopted design convention (Opt-#1,#2). Along with QIsim and our 10+K qubit QCI designs, our design insights and optimizations will be the great references to realize quantum supremacy.

7 DISCUSSION

7.1 Maturity of current QCI technologies

We model QIsim to cover all the currently-available and promising QCI technologies even though their maturity is different from each other. Thus, based on up-to-date previous works, we clarify the maturity of each technology in Table 3. In the current status shown in Table 3, we use the best reference data for each technology.

As our analysis is based on the current technology, the detailed scalability number can vary with the technology evolution and maturity. Even though we cannot provide the exact future scenarios, QIsim is still useful as architects can analyze those future systems by changing the simulation parameters (e.g., wire passive and active load, gate delay). In addition, our optimizations also will be effective in many future cases because they holistically mitigate the error, bandwidth, and power issues.

7.2 Composability, modularity, and scalability

To provide more details of our model, we describe the composability, modularity, and scalability of QIsim.

Composability. We can easily integrate QIsim with other tools (e.g., qubit and QCP simulators) to simulate the entire quantum-computer system. For example, by passing the output gate-timing information and gate-error rate to qubit simulators (e.g., Qiskit [68], Cirq [26]), QIsim runs along with the qubit simulators. QIsim is also composable with QCP simulators (e.g., XQsim [7]) by taking the output gates of XQsim as the input of cycle-accurate simulator.

Modularity/Extensibility. QIsim is designed in fully modular and parameterized way to easily support future architecture and technology evolution. For example, to evaluate the new 4K CMOS drive circuit microarchitecture, users just edit the Verilog code of the circuit model and CMOS single-qubit gate-error model without modifying other modules. In addition, architects just should change the input simulation parameters to analyze the future system with longer T1 & T2 times, lower wiring overhead, higher cooling capacity, and lower-noise analog circuits & microwave components (e.g., attenuator, amplifier).

Scalability. QIsim is scalable to simulate more than 10+K qubits. Specifically, QIsim's time complexity is sublinear when evaluating the FTQC systems because we analyze the power/error complexity of QCIs rather than accurately simulate the quantum states.

7.3 QIsim's limitations

QIsim has several limitations due to its limited coverage. First, QIsim cannot support QCI technologies other than CMOS and SFQ (e.g., AQFP and adiabatic CMOS design family). Second, QIsim does not support other popular two qubit gates (e.g., iSWAP and CR gates) and leakage-state measurement. Third, although QIsim accurately models the noise from QCI and microwave components, it simply aggregates the complex environmental noises (e.g., 1/f noise, pink noise, and correlated noise) into the relaxation and decoherence errors using T1 and T2 times, respectively. Fourth, for the microwave component, QIsim only supports a single fixed attenuator setup (0-20-10-10-20dB attenuation for 50K-4K-1K-100mK-20mK stages). Finally, QIsim does not yet support temperature domains with higher power budgets (e.g., 30W [45] at 70K) at which we may further improve scalability by moving power-hungry components. To support these (and more) design choices, we keep extending QIsim's coverage in various directions.

8 RELATED WORK

Scalable QCI. Lecocq et al. [46] and Amir et al. [80] proposed QCIs with photonic link. Intel [59, 76], IBM [12], Google [5], Microsoft [60], and researchers in academia [40, 53, 62] proposed 4K QCIs for spin and Transmon qubits. McDermott et al. [54] and Jokar et al. [39] proposed RSFQ QCIs. However, no previous work fully implemented 4K CMOS and SFQ-based QCIs targeting Transmon, holistically analyzed scalability, and proposed a 10+K qubit QCI.

Cryogenic CMOS and SFQ modeling. Lee et al. [47, 48], Min et al. [55, 56], Byun et al. [8, 9] developed cryogenic CMOS models for memory and logic devices. Ishida et al. [35] proposed SFQ-based NPU model. By extending them, Byun et al. [7] developed 4K CMOS and RSFQ models for QCP. However, no previous work developed a scalability analysis tool for QCIs.

To our knowledge, this work is the first to build the QCI scalability-analysis tool and propose 10+K qubit QCI with optimizations.

9 CONCLUSION

This paper proposed (1) *QIsim*, an open-source QCI simulation framework, and (2) 10+K qubit QCIs along with eight architectural optimizations. For that purpose, we implemented the detailed QCI microarchitectures and developed our scalability-analysis tool (QIsim) with thorough validation. Then, driven by QIsim, we proposed eight architectural optimizations with 60,000+ qubit-scale QCIs. As QIsim is the first open-source QCI simulator, it will contribute to initiating active research on QCI.

ACKNOWLEDGMENTS

This work was supported in part by National Research Foundation of Korea (NRF) funded by the Korean Government under Grants NRF-2019R1A5A1027055, NRF-2021R1A2C3014131, NRF-2021M3F3A2A02037893, and NRF-2019-Global Ph.D. Fellowship Program; in part by the Institute of Information & Communications Technology Planning & Evaluation funded by the Korean Government under Grants 2021000853; in part by the Creative Pioneering Researchers Program through Seoul National University; and in part by the Automation and Systems Research Institute (ASRI) and Inter-university Semiconductor Research Center at Seoul National University. We also appreciate the support from JSPS KAKENHI under Grant JP22H05000. The circuit was designed with the support by VDEC of the University of Tokyo in collaboration with Cadence Design Systems, Inc.

REFERENCES

[1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (01 Oct 2019), 505–510. https://doi.org/10.1038/s41586-019-1666-5

[2] Ryan Babbush, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. 2018. Encoding electronic spectra in quantum circuits with linear T complexity. *Physical Review X* 8, 4 (2018), 041015.

[3] Baidu. 2023. Baidu Quanlse. https://quanlse.baidu.com

[4] Joseph Bardin. 2022. Beyond-Classical Computing Using Superconducting Quantum Processors. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 65. IEEE, 422–424.

[5] Joseph C. Bardin, Evan Jeffrey, Erik Lucero, Trent Huang, Sayan Das, Daniel Thomas Sank, Ofer Naaman, Anthony Edward Megrant, Rami Barends, Ted White, Marissa Giustina, Kevin J. Satzinger, Kunal Arya, Pedram Roushan, Benjamin Chiaro, Julian Kelly, Zijun Chen, Brian Burkett, Yu Chen, Andrew Dunsworth, Austin Fowler, Brooks Foxen, Craig Gidney, Rob Graff, Paul Klimov, Josh Mutus, Matthew J. McEwen, Matthew Neeley, Charles J. Neill, Chris Quintana, Amit Vainsencher, Hartmut Neven, and John Martinis. 2019. Design and Characterization of a 28-nm Bulk-CMOS Cryogenic Quantum Controller Dissipating Less Than 2 mW at 3 K. *IEEE Journal of Solid-State Circuits* 54, 11 (2019), 3043–3060. https://doi.org/10.1109/JSSC.2019.2937234

[6] R Bianchetti, Stefan Filipp, M Baur, JM Fink, M Göppl, Peter J Leek, L Steffen, Alexandre Blais, and Andreas Wallraff. 2009. Dynamics of dispersive single-qubit readout in circuit quantum electrodynamics. *Physical Review A* 80, 4 (2009), 043840.

[7] Ilkwon Byun, Junpyo Kim, Dongmoon Min, Ikki Nagaoka, Kosuke Fukumitsu, Iori Ishikawa, Teruo Tanimoto, Masamitsu Tanaka, Koji Inoue, and Jangwoo Kim. 2022. XQsim: modeling cross-technology control processors for 10+ K qubit quantum computers. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 366–382.

[8] Ilkwon Byun, Dongmoon Min, Gyuhyeon Lee, Seongmin Na, and Jangwoo Kim. 2021. A Next-Generation Cryogenic Processor Architecture. *IEEE Micro* 41, 3 (2021), 80–86.

[9] Ilkwon Byun, Dongmoon Min, Gyu-hyeon Lee, Seongmin Na, and Jangwoo Kim. 2020. CryoCore: A fast and dense processor architecture for cryogenic computing. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 335–348.

[10] Juan-Antonio Carballo, Wei-Ting Jonas Chan, Paolo A Gargini, Andrew B Kahng, and Siddhartha Nath. 2014. ITRS 2.0: Toward a re-framing of the Semiconductor Technology Roadmap. In *2014 IEEE 32nd international conference on computer design (ICCD)*. IEEE, 139–146.

[11] Eunjung Cha, Niklas Wadefalk, Giuseppe Moschetti, Arsalan Pourkabirian, Jörgen Stenarson, and Jan Grahn. 2020. A 300-μW Cryogenic HEMT LNA for quantum computing. In *2020 IEEE/MTT-S International Microwave Symposium (IMS)*. IEEE, 1299–1302.

[12] Sudipto Chakraborty, David J. Frank, Kevin Tien, Pat Rosno, Mark Yeck, Joseph A. Glick, Raphael Robertazzi, Ray Richetta, John F. Bulzacchelli, Devin Underwood, Daniel Ramirez, Dereje Yilma, Andrew Davies, Rajiv V. Joshi, Shawn D. Chambers, Scott Lekuch, Ken Inoue, Dorothy Wisnieff, Christian W. Baks, Donald S. Bethune, John Timmerwilke, Thomas Fox, Peilin Song, Blake R. Johnson, Brian P. Gaucher, and Daniel J. Friedman. 2022. A Cryo-CMOS Low-Power Semi-Autonomous Transmon Qubit State Controller in 14-nm FinFET Technology. *IEEE Journal of Solid-State Circuits* 57, 11 (2022), 3258–3273. https://doi.org/10.1109/JSSC.2022.3201775

[13] COAX. 2023. Coaxial Cable (SC-086/50-SS-SS). http://www.coax.co.jp/en/product/sc/086-50-ss-ss.html

[14] COAX. 2023. Superconducting Coaxial Cable (SC-033/50-NbTi-CN). http://www.coax.co.jp/en/product/sc/033-50-nbti-cn.html

[15] Andrew Cross, Ali Javadi-Abhari, Thomas Alexander, Niel De Beaudrap, Lev S. Bishop, Steven Heidel, Colm A. Ryan, Prasahnt Sivarajah, John Smolin, Jay M. Gambetta, and Blake R. Johnson. 2022. OpenQASM 3: A Broader and Deeper Quantum Assembly Language. *ACM Transactions on Quantum Computing* 3, 3, Article 12 (sep 2022), 50 pages. https://doi.org/10.1145/3505636

[16] Poulami Das, Aditya Locharla, and Cody Jones. 2022. LILLIPUT: a lightweight low-latency lookup-table decoder for near-term Quantum error correction. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 541–553.

[17] Poulami Das, Christopher A Pattison, Srilatha Manne, Douglas M Carmean, Krysta M Svore, Moinuddin Qureshi, and Nicolas Delfosse. 2022. AFS: Accurate, Fast, and Scalable Error-Decoding for Fault-Tolerant Quantum Computers. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 259–273.

[18] DelftCircuit. 2023. Flexible multi channel cable (CrioFlex3). https://delft-circuits.com/product-overview/cf3/

[19] Johannes Arnoldus Delport, Kyle Jackman, Paul Le Roux, and Coenrad Johann Fourie. 2019. Josim—superconductor spice simulator. *IEEE Transactions on Applied Superconductivity* 29, 5 (2019), 1–5.

[20] Yongshan Ding, Adam Holmes, Ali Javadi-Abhari, Diana Franklin, Margaret Martonosi, and Frederic Chong. 2018. Magic-state functional units: Mapping and scheduling multi-level distillation circuits for fault-tolerant quantum architectures. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 828–840.

[21] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. 2012. Surface codes: Towards practical large-scale quantum computation. *Physical Review A* 86, 3 (2012), 032324.

[22] X. Fu, L. Riesebos, M. A. Rol, Jeroen van Straten, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, V. Newsum, K. K. L. Loh, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudever, L. DiCarlo, and K. Bertels. 2019. eQASM: An Executable Quantum Instruction Set Architecture. In *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 224–237. https://doi.org/10.1109/HPCA.2019.00040

[23] X. Fu, M. A. Rol, C. C. Bultink, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudever, L. DiCarlo, and K. Bertels. 2017. An Experimental Microarchitecture for a Superconducting Quantum Processor. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture* (Cambridge, Massachusetts) *(MICRO-50 '17)*. Association for Computing Machinery, New York, NY, USA, 813–825. https://doi.org/10.1145/3123939.3123952

[24] Michael R Geller and Zhongyuan Zhou. 2013. Efficient error models for fault-tolerant architectures and the Pauli twirling approximation. *Physical Review A* 88, 1 (2013), 012314.

[25] Joydip Ghosh, Austin G Fowler, and Michael R Geller. 2012. Surface code with decoherence: An analysis of three superconducting architectures. *Physical Review A* 86, 6 (2012), 062318.

[26] Google. 2023. Cirq. 10.5281/zenodo.4062499

[27] Luke CG Govia, Emily J Pritchett, Seth T Merkel, Deanna Pineau, and Frank K Wilhelm. 2012. Theory of Josephson photomultipliers: Optimal working conditions and back action. *Physical Review A* 86, 3 (2012), 032311.

[28] Luke CG Govia, Emily J Pritchett, Canran Xu, BLT Plourde, Maxim G Vavilov, Frank K Wilhelm, and R McDermott. 2014. High-fidelity qubit measurement with a microwave-photon counter. *Physical Review A* 90, 6 (2014), 062307.

[29] Mauricio Gutiérrez and Kenneth R Brown. 2015. Comparison of a quantum error-correction threshold for exact and approximate errors. *Physical Review A* 91, 2 (2015), 022335.

[30] Anna Herr, Arkady Fedorov, Alexander Shnirman, Evgeny Il'Ichev, and Gerd Schön. 2007. Design of a ballistic fluxon qubit readout. *Superconductor Science and Technology* 20, 11 (2007), S450.

[31] Adam Holmes, Mohammad Reza Jokar, Ghasem Pasandi, Yongshan Ding, Massoud Pedram, and Frederic T Chong. 2020. NISQ+: Boosting quantum computing power by approximating quantum error correction. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 556–569.

[32] Caleb Howington, Alex Opremcak, Robert McDermott, Alex Kirichenko, Oleg A Mukhanov, and Britton LT Plourde. 2019. Interfacing superconducting qubits with cryogenic logic: Readout. *IEEE Transactions on Applied Superconductivity* 29, 5 (2019), 1–5.

[33] IBM. 2023. Expanding the IBM Quantum roadmap to anticipate the future of quantum-centric supercomputing. https://research.ibm.com/blog/ibm-quantum-roadmap-2025

[34] IBM. 2023. IBMQ. https://quantum-computing.ibm.com

[35] Koki Ishida, Ilkwon Byun, Ikki Nagaoka, Kosuke Fukumitsu, Masamitsu Tanaka, Satoshi Kawakami, Teruo Tanimoto, Takatsugu Ono, Jangwoo Kim, and Koji Inoue. 2020. SuperNPU: An extremely fast neural processing unit using superconducting logic devices. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 58–72.

[36] Ali Javadi-Abhari, Pranav Gokhale, Adam Holmes, Diana Franklin, Kenneth R Brown, Margaret Martonosi, and Frederic T Chong. 2017. Optimized surface code communication in superconducting quantum computers. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. 692–705.

[37] Ali JavadiAbhari, Shruti Patil, Daniel Kudrow, Jeff Heckey, Alexey Lvov, Frederic T Chong, and Margaret Martonosi. 2014. ScaffCC: A framework for compilation

and analysis of quantum computing programs. In *Proceedings of the 11th ACM Conference on Computing Frontiers*. 1–10.

[38] J Robert Johansson, Paul D Nation, and Franco Nori. 2012. QuTiP: An open-source Python framework for the dynamics of open quantum systems. *Computer Physics Communications* 183, 8 (2012), 1760–1772.

[39] Mohammad Reza Jokar, Richard Rines, Ghasem Pasandi, Haolin Cong, Adam Holmes, Yunong Shi, Massoud Pedram, and Frederic T Chong. 2022. DigiQ: A Scalable Digital Controller for Quantum Computers Using SFQ Logic. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 400–414.

[40] Kiseo Kang, Donggyu Minn, Seongun Bae, Jaeho Lee, Seokhyeong Kang, Moonjoo Lee, Ho-Jin Song, and Jae-Yoon Sim. 2022. A 40-nm Cryo-CMOS Quantum Controller IC for Superconducting Qubit. *IEEE Journal of Solid-State Circuits* 57, 11 (2022), 3274–3287.

[41] Nam Sung Kim, Todd Austin, David Baauw, Trevor Mudge, Krisztián Flautner, Jie S Hu, Mary Jane Irwin, Mahmut Kandemir, and Vijaykrishnan Narayanan. 2003. Leakage current: Moore's law meets static power. *computer* 36, 12 (2003), 68–75.

[42] DE Kirichenko, Saad Sarwana, and AF Kirichenko. 2011. Zero static power dissipation biasing of RSFQ circuits. *IEEE Transactions on Applied Superconductivity* 21, 3 (2011), 776–779.

[43] Ian D. Kivlichan, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Wei Sun, Zhang Jiang, Nicholas Rubin, Austin Fowler, Alán Aspuru-Guzik, Hartmut Neven, and Ryan Babbush. 2020. Improved Fault-Tolerant Quantum Simulation of Condensed-Phase Correlated Electrons via Trotterization. *Quantum* 4 (July 2020), 296. https://doi.org/10.22331/q-2020-07-16-296

[44] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. 2019. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews* 6, 2 (2019), 021318.

[45] Sebastian Krinner, Simon Storz, Philipp Kurpiers, Paul Magnard, Johannes Heinsoo, Raphael Keller, Janis Luetolf, Christopher Eichler, and Andreas Wallraff. 2019. Engineering cryogenic setups for 100-qubit scale superconducting circuit systems. *EPJ Quantum Technology* 6, 1 (2019), 2.

[46] Florent Lecocq, Franklyn Quinlan, Katarina Cicak, Jose Aumentado, SA Diddams, and JD Teufel. 2021. Control and readout of a superconducting qubit using a photonic link. *Nature* 591, 7851 (2021), 575–579.

[47] Gyu-hyeon Lee, Dongmoon Min, Ilkwon Byun, and Jangwoo Kim. 2019. Cryogenic computer architecture modeling with memory-side case studies. In *Proceedings of the 46th International Symposium on Computer Architecture*. 774–787.

[48] Gyu-Hyeon Lee, Seongmin Na, Ilkwon Byun, Dongmoon Min, and Jangwoo Kim. 2021. CryoGuard: A near refresh-free robust DRAM design for cryogenic computing. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 637–650.

[49] Edward Leonard Jr, Matthew A Beck, J Nelson, Brad G Christensen, Ted Thorbeck, Caleb Howington, Alexander Opremcak, Ivan V Pechenezhskiy, Kenneth Dodge, Nicholas P Dupuis, et al. 2019. Digital coherent control of a superconducting qubit. *Physical Review Applied* 11, 1 (2019), 014009.

[50] Kangbo Li, R McDermott, and Maxim G Vavilov. 2019. Hardware-efficient qubit control with single-flux-quantum pulse sequences. *Physical Review Applied* 12, 1 (2019), 014044.

[51] Daniel Litinski. 2019. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum* 3 (2019), 128.

[52] Daniel Litinski and Felix von Oppen. 2018. Lattice surgery with a twist: simplifying clifford gates of surface codes. *Quantum* 2 (2018), 62.

[53] Satvik Maurya and Swamit Tannu. 2022. COMPAQT: Compressed Waveform Memory Architecture for Scalable Qubit Control. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 1059–1077.

[54] R McDermott, MG Vavilov, BLT Plourde, FK Wilhelm, PJ Liebermann, OA Mukhanov, and TA Ohki. 2018. Quantum–classical interface based on single flux quantum digital logic. *Quantum science and technology* 3, 2 (2018), 024004.

[55] Dongmoon Min, Ilkwon Byun, Gyu-Hyeon Lee, Seongmin Na, and Jangwoo Kim. 2020. CryoCache: A fast, large, and cost-effective cache architecture for cryogenic computing. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. 449–464.

[56] Dongmoon Min, Yujin Chung, Ilkwon Byun, Junpyo Kim, and Jangwoo Kim. 2022. CryoWire: wire-driven microarchitecture designs for cryogenic computing. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 903–917.

[57] Shuichi Nagasawa, Kenji Hinode, Tetsuro Satoh, Mutsuo Hidaka, Hiroyuki Akaike, Akira Fujimaki, Nobuyuki Yoshikawa, Kazuyoshi Takagi, and Naofumi Takagi. 2014. Nb 9-layer fabrication process for superconducting large-scale SFQ circuits and its process evaluation. *IEICE Transactions on Electronics* 97, 3 (2014), 132–140.

[58] A. Opremcak, C. H. Liu, C. Wilen, K. Okubo, B. G. Christensen, D. Sank, T. C. White, A. Vainsencher, M. Giustina, A. Megrant, B. Burkett, B. L. T. Plourde, and R. McDermott. 2021. High-Fidelity Measurement of a Superconducting Qubit Using an On-Chip Microwave Photon Counter. *Phys. Rev. X* 11 (Feb 2021), 011027. Issue 1. https://doi.org/10.1103/PhysRevX.11.011027

[59] Jongseok Park, Sushil Subramanian, Lester Lampert, Todor Mladenov, Ilya Klotchkov, Dileep J. Kurian, Esdras Juarez-Hernandez, Brando Perez Esparza, Sirisha Rani Kale, Asma Beevi K. T., Shavindra P. Premaratne, Thomas F. Watson, Satoshi Suzuki, Mustafijur Rahman, Jaykant B. Timbadiya, Saksham Soni, and Stefano Pellerano. 2021. A Fully Integrated Cryo-CMOS SoC for State Manipulation, Readout, and High-Speed Gate Pulsing of Spin Qubits. *IEEE Journal of Solid-State Circuits* 56, 11 (2021), 3289–3306. https://doi.org/10.1109/JSSC.2021.3115988

[60] S. J. Pauka, K. Das, R. Kalra, A. Moini, Y. Yang, M. Trainer, A. Bousquet, C. Cantaloube, N. Dick, G. C. Gardner, M. J. Manfra, and D. J. Reilly. 2021. A cryogenic CMOS chip for generating control signals for multiple qubits. *Nature Electronics* 4, 1 (01 Jan 2021), 64–70. https://doi.org/10.1038/s41928-020-00528-y

[61] Arpit Ranadive, Martina Esposito, Luca Planat, Edgar Bonet, Cécile Naud, Olivier Buisson, Wiebke Guichard, and Nicolas Roch. 2022. Kerr reversal in Josephson meta-material and traveling wave parametric amplification. *Nature communications* 13, 1 (2022), 1–9.

[62] Andrea Ruffino, Yatao Peng, Tsung-Yeh Yang, John Michniewicz, Miguel Fernando Gonzalez-Zalba, and Edoardo Charbon. 2021. 13.2 a fully-integrated 40-nm 5-6.5 GHz cryo-CMOS system-on-chip with I/Q receiver and frequency synthesizer for scalable multiplexed readout of quantum dots. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 64. IEEE, 210–212.

[63] Lieze Schindler, Johannes A Delport, and Coenrad J Fourie. 2021. The ColdFlux RSFQ cell library for MIT-LL SFQ5ee fabrication process. *IEEE Transactions on Applied Superconductivity* 32, 2 (2021), 1–7.

[64] Zheng Shan, Yu Zhu, and Bo Zhao. 2022. A high-performance compilation strategy for multiplexing quantum control architecture. *Scientific Reports* 12, 1 (2022), 1–10.

[65] James E. Stine, Ivan Castellanos, Michael Wood, Jeff Henson, Fred Love, W. Rhett Davis, Paul D. Franzon, Michael Bucher, Sunil Basavarajaiah, Julie Oh, and Ravi Jenkal. 2007. FreePDK: An Open-Source Variation-Aware Design Kit. In *2007 IEEE International Conference on Microelectronic Systems Education (MSE'07)*. 173–174. https://doi.org/10.1109/MSE.2007.44

[66] Youngkyu Sung, Leon Ding, Jochen Braumüller, Antti Vepsäläinen, Bharath Kannan, Morten Kjaergaard, Ami Greene, Gabriel O. Samach, Chris McNally, David Kim, Alexander Melville, Bethany M. Niedzielski, Mollie E. Schwartz, Jonilyn L. Yoder, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. 2021. Realization of High-Fidelity CZ and ZZ-Free iSWAP Gates with a Tunable Coupler. *Phys. Rev. X* 11 (Jun 2021), 021058. Issue 2. https://doi.org/10.1103/PhysRevX.11.021058

[67] Yasunari Suzuki, Takanori Sugiyama, Tomochika Arai, Wang Liao, Koji Inoue, and Teruo Tanimoto. 2022. Q3DE: A fault-tolerant quantum computer architecture for multi-bit burst errors by cosmic rays. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 1110–1125.

[68] A tA v, MD SAJID ANIS, Abby-Mitchell, Héctor Abraham, AduOffei, Rochisha Agarwal, Gabriele Agliardi, Merav Aharoni, Vishnu Ajith, Ismail Yunus Akhalwaya, Gadi Aleksandrowicz, Thomas Alexander, Matthew Amy, Sashwat Anagolum, Anthony-Gandon, Israel F. Araujo, Eli Arbel, Abraham Asfaw, Anish Athalye, Artur Avkhadiev, Carlos Azaustre, PRATHAMESH BHOLE, Vishal Bajpe, Abhik Banerjee, Santanu Banerjee, Will Bang, Aman Bansal, Panagiotis Barkoutsos, Ashish Barnawal, George Barron, George S. Barron, Luciano Bello, Yael Ben-Haim, M. Chandler Bennett, Daniel Bevenius, Dhruv Bhatnagar, Prakhar Bhatnagar, Arjun Bhobe, Paolo Bianchini, Lev S. Bishop, Carsten Blank, Sorin Bolos, Soham Bopardikar, Samuel Bosch, Sebastian Brandhofer, Brandon, Sergey Bravyi, Bryce-Fuller, David Bucher, Artemiy Burov, Fran Cabrera, Padraic Calpin, Lauren Capelluto, Jorge Carballo, Ginés Carrascal, Adam Carriker, Ivan Carvalho, Rishabh Chakrabarti, Adrian Chen, Chun-Fu Chen, Edward Chen, Jielun (Chris) Chen, Richard Chen, Franck Chevallier, Kartik Chinda, Rathish Cholarajan, Jerry M. Chow, Spencer Churchill, CisterMoke, Christian Claus, Christian Clauss, Caleb Clothier, Romilly Cocking, Ryan Cocuzzo, Jordan Connor, Filipe Correa, Zachary Crockett, Abigail J. Cross, Andrew W. Cross, Simon Cross, Juan Cruz-Benito, Chris Culver, Antonio D. Córcoles-Gonzales, Navaneeth D, Sean Dague, Tareq El Dandachi, Animesh N Dangwal, Jonathan Daniel, Marcus Daniels, Matthieu Dartiailh, Abdón Rodríguez Davila, Faisal Debouni, Anton Dekusar, Amol Deshmukh, Mohit Deshpande, Delton Ding, Jun Doi, Eli M. Dow, Patrick Downing, Eric Drechsler, Marc Sanz Drudis, Eugene Dumitrescu, Karel Dumon, Ivan Duran, Kareem EL-Safty, Eric Eastman, Grant Eberle, Amir Ebrahimi, Pieter Eendebak, Daniel Egger, ElePT, Iman Elsayed, Emilio, Alberto Espiricueta, Mark Everitt, Davide Facoetti, Farida, Paco Martín Fernández, Samuele Ferracin, Davide Ferrari, Axel Hernández Ferrera, Romain Fouilland, Albert Frisch, Andreas Fuhrer, Bryce Fuller, MELVIN GEORGE, Julien Gacon, Borja Godoy Gago, Claudio Gambella, Jay M. Gambetta, Adhisha Gammanpila, Luis Garcia, Tanya Garg, Shelly Garion, James R. Garrison, Jim Garrison, Tim Gates, Gian Gentinetta, Hristo Georgiev, Leron Gil, Austin Gilliam, Aditya Giridharan, Glen, Juan Gomez-Mosquera, Gonzalo, Salvador de la Puente González, Jesse Gorzinski, Ian Gould, Donny Greenberg, Dmitry Grinko, Wen Guan, Dani Guijo, Guillermo-Mijares-Vilarino, John A. Gunnels, Harshit Gupta, Naman Gupta, Jakob M. Günther, Mikael Haglund, Isabel Haide, Ikko Hamamura, Omar Costa Hamido, Frank Harkins, Kevin Hartman, Areeq Hasan, Vojtech Havlicek, Joe Hellmers, Łukasz

Herok, Ryan Hill, Stefan Hillmich, Colin Hong, Hiroshi Horii, Connor Howington, Shaohan Hu, Wei Hu, Chih-Han Huang, Junye Huang, Rolf Huisman, Haruki Imai, Takashi Imamichi, Kazuaki Ishizaki, Ishwor, Raban Iten, Toshinari Itoko, Alexander Ivrii, Ali Javadi, Ali Javadi-Abhari, Wahaj Javed, Qian Jianhua, Madhav Jivrajani, Kiran Johns, Scott Johnstun, Jonathan-Shoemaker, JosDenmark, JoshDumo, John Judge, Tal Kachmann, Akshay Kale, Naoki Kanazawa, Jessica Kane, Kang-Bae, Annanay Kapila, Anton Karazeev, Paul Kassebaum, Tobias Kehrer, Josh Kelso, Scott Kelso, Hugo van Kemenade, Vismai Khanderao, Spencer King, Yuri Kobayashi, Kovi11Day, Arseny Kovyrshin, Rajiv Krishnakumar, Pradeep Krishnamurthy, Vivek Krishnan, Kevin Krsulich, Prasad Kumkar, Gawel Kus, Ryan LaRose, Enrique Lacal, Raphaël Lambert, Haggai Landa, John Lapeyre, Joe Latone, Scott Lawrence, Christina Lee, Gushu Li, Tan Jun Liang, Jake Lishman, Dennis Liu, Peng Liu, Lolcroc, Abhishek K M, Liam Madden, Yunho Maeng, Saurav Maheshkar, Kahan Majmudar, Aleksei Malyshev, Mohamed El Mandouh, Joshua Manela, Manjula, Jakub Marecek, Manoel Marques, Kunal Marwaha, Dmitri Maslov, Paweł Maszota, Dolph Mathews, Atsushi Matsuo, Farai Mazhandu, Doug McClure, Maureen McElaney, Joseph McElroy, Cameron McGarry, David McKay, Dan McPherson, Srujan Meesala, Dekel Meirom, Corey Mendell, Thomas Metcalfe, Martin Mevissen, Andrew Meyer, Antonio Mezzacapo, Rohit Midha, Declan Millar, Daniel Miller, Hannah Miller, Zlatko Minev, Abby Mitchell, Nikolaj Moll, Alejandro Montanez, Gabriel Monteiro, Michael Duane Mooring, Renier Morales, Niall Moran, David Morcuende, Seif Mostafa, Mario Motta, Romain Moyard, Prakash Murali, Daiki Murata, Jan Müggenburg, Tristan NEMOZ, David Nadlinger, Ken Nakanishi, Giacomo Nannicini, Paul Nation, Edwin Navarro, Yehuda Naveh, Scott Wyman Neagle, Patrick Neuweiler, Aziz Ngoueya, Thien Nguyen, Johan Nicander, Nick-Singstock, Pradeep Niroula, Hassi Norlen, NuoWenLei, Lee James O'Riordan, Oluwatobi Ogunbayo, Pauline Ollitrault, Tamiya Onodera, Raul Otaolea, Steven Oud, Dan Padilha, Hanhee Paik, Soham Pal, Yuchen Pang, Ashish Panigrahi, Vincent R. Pascuzzi, Simone Perriello, Eric Peterson, Anna Phan, Kuba Pilch, Francesco Piro, Marco Pistoia, Christophe Piveteau, Julia Plewa, Pierre Pocreau, Clemens Possel, Alejandro Pozas-Kerstjens, Rafał Pracht, Milos Prokop, Viktor Prutyanov, Sumit Puri, Daniel Puzzuoli, Pythonix, Jesús Pérez, Quant02, Quintiii, Rafey Iqbal Rahman, Arun Raja, Roshan Rajeev, Isha Rajput, Nipun Ramagiri, Anirudh Rao, Rudy Raymond, Oliver Reardon-Smith, Rafael Martín-Cuevas Redondo, Max Reuter, Julia Rice, Matt Riedemann, Rietesh, Drew Risinger, Pedro Rivero, Marcello La Rocca, Diego M. Rodríguez, RohithKarur, Ben Rosand, Max Rossmannek, Mingi Ryu, Tharrmashastha SAPV, Nahum Rosa Cruz Sa, Arijit Saha, Abdullah Ash-Saki, Arfat Salman, Sankalp Sanand, Martin Sandberg, Hirmay Sandesara, Ritvik Sapra, Hayk Sargsyan, Aniruddha Sarkar, Ninad Sathaye, Niko Savola, Bruno Schmitt, Chris Schnabel, Zachary Schoenfeld, Travis L. Scholten, Eddie Schoute, Mark Schulterbrandt, Joachim Schwarm, Paul Schweigert, James Seaward, Sergi, Ismael Faro Sertage, Kanav Setia, Freya Shah, Nathan Shammah, Will Shanks, Rohan Sharma, Polly Shaw, Yunong Shi, Jonathan Shoemaker, Adenilton Silva, Andrea Simonetto, Deeksha Singh, Divyanshu Singh, Parmeet Singh, Phattharaporn Singkanipa, Yukio Siraichi, Siri, Jesus Sistos, Jesús Sistos, Iskandar Sitdikov, Seyon Sivarajah, Slavikmew, Magnus Berg Sletfjerding, John A. Smolin, Mathias Soeken, Igor Olegovich Sokolov, Igor Sokolov, Vicente P. Soloviev, SooluThomas, Starfish, Dominik Steenken, Matt Stypulkoski, Adrien Suau, Shaojun Sun, Kevin J. Sung, Makoto Suwama, Oskar Słowik, Rohit Taeja, Hitomi Takahashi, Tanvesh Takawale, Ivano Tavernelli, Charles Taylor, Pete Taylour, Soolu Thomas, Kevin Tian, Mathieu Tillet, Maddy Tod, Miroslav Tomasik, Caroline Tornow, Enrique de la Torre, Juan Luis Sánchez Toural, Kenso Trabing, Matthew Treinish, Dimitar Trenev, TrishaPe, Felix Truger, Georgios Tsilimigkounakis, Davindra Tulsi, Doğukan Tuna, Wes Turner, Yotam Vaknin, Carmen Recio Valcarce, Francois Varchon, Adish Vartak, Almudena Carrera Vazquez, Prajjwal Vijaywargiya, Victor Villar, Bhargav Vishnu, Desiree Vogt-Lee, Christophe Vuillot, WQ, James Weaver, Johannes Weidenfeller, Rafal Wieczorek, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, WinterSoldier, Jack J. Woehr, Stefan Woerner, Ryan Woo, Christopher J. Wood, Ryan Wood, Steve Wood, James Wootton, Matt Wright, Lucy Xing, Jintao YU, Yaiza, Bo Yang, Unchun Yang, Jimmy Yao, Daniyar Yeralin, Ryota Yonekura, David Yonge-Mallo, Ryuhei Yoshida, Richard Young, Jessie Yu, Lebin Yu, Yuma-Nakamura, Christopher Zachow, Laura Zdanski, Helena Zhang, Iulia Zidaru, Bastian Zimmermann, Christa Zoufal, aeddins ibm, alexzhang13, b63, bartek bartlomiej, bcamorrison, brandhsn, nick bronn, chetmurthy, choerst ibm, comet, dalin27, deeplokhande, dekel.meirom, derwind, dime10, dlasecki, ehchen, ewinston, fanizzamarco, fs1132429, gadial, galeinston, georgezhou20, georgios ts, gruu, hhorii, hhyap, hykavitha, itoko, jeppevinkel, jessica angel7, jezerjojo14, jliu45, johannesgreiner, jscott2, kUmezawa, klinvill, krutik2966, ma5x, michelle4654, msuwama, nico lgrs, nrhawkins, ntgiwsvp, ordmoj, sagar pahwa, pritamsinha2304, rithikaadiga, ryancocuzzo, saktar unr, saswati qiskit, sebastian mair, septembrr, sethmerkel, sg495, shaashwat, smturro2, sternparky, strickroman, tigerjack, tsura crisaldo, upsideon, vadebayo49, welien, willhbang, wmurphy collabstar, yang.luh, yuri@FreeBSD, and Mantas Čepulkovskis. 2021. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505

[69] Swamit S Tannu, Zachary A Myers, Prashant J Nair, Douglas M Carmean, and Moinuddin K Qureshi. 2017. Taming the instruction bandwidth of quantum computers via hardware-managed error correction. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. 679–691.

[70] Barbara M Terhal. 2015. Quantum error correction for quantum memories. *Reviews of Modern Physics* 87, 2 (2015), 307.

[71] Teague Tomesh, Pranav Gokhale, Victory Omole, Gokul Subramanian Ravi, Kaitlin N Smith, Joshua Viszlai, Xin-Chuan Wu, Nikos Hardavellas, Margaret R Martonosi, and Frederic T Chong. 2022. Supermarq: A scalable quantum benchmark suite. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 587–603.

[72] Yu Tomita and Krysta M Svore. 2014. Low-distance surface codes under realistic quantum noise. *Physical Review A* 90, 6 (2014), 062320.

[73] David B Tuckerman, Michael C Hamilton, David J Reilly, Rujun Bai, George A Hernandez, John M Hornibrook, John A Sellers, and Charles D Ellis. 2016. Flexible superconducting Nb transmission lines on thin film polyimide for quantum computing applications. *Superconductor Science and Technology* 29, 8 (2016), 084007.

[74] Yosuke Ueno, Masaaki Kondo, Masamitsu Tanaka, Yasunari Suzuki, and Yutaka Tabuchi. 2022. QULATIS: A Quantum Error Correction Methodology toward Lattice Surgery. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 274–287.

[75] Jeroen PG Van Dijk, Bishnu Patra, Stefano Pellerano, Edoardo Charbon, Fabio Sebastiano, and Masoud Babaie. 2020. Designing a DDS-based SoC for high-fidelity multi-qubit control. *IEEE Transactions on Circuits and Systems I: Regular Papers* 67, 12 (2020), 5380–5393.

[76] Jeroen Petrus Gerardus Van Dijk, Bishnu Patra, Sushil Subramanian, Xiao Xue, Nodar Samkharadze, Andrea Corna, Charles Jeon, Farhana Sheikh, Esdras Juarez-Hernandez, Brando Perez Esparza, Huzaifa Rampurawala, Brent R. Carlton, Surej Ravikumar, Carlos Nieva, Sungwon Kim, Hyung-Jin Lee, Amir Sammak, Giordano Scappucci, Menno Veldhorst, Lieven M. K. Vandersypen, Edoardo Charbon, Stefano Pellerano, Masoud Babaie, and Fabio Sebastiano. 2020. A Scalable Cryo-CMOS Controller for the Wideband Frequency-Multiplexed Control of Spin Qubits and Transmons. *IEEE Journal of Solid-State Circuits* 55, 11 (2020), 2930–2946. https://doi.org/10.1109/JSSC.2020.3024678

[77] Stephen Williams and Michael Baxter. 2002. Icarus verilog: open-source verilog more than a year later. *Linux Journal* 2002, 99 (2002), 3.

[78] Clifford Wolf. 2016. Yosys open synthesis suite.

[79] Xiao Xue, Bishnu Patra, Jeroen P. G. van Dijk, Nodar Samkharadze, Sushil Subramanian, Andrea Corna, Brian Paquelet Wuetz, Charles Jeon, Farhana Sheikh, Esdras Juarez-Hernandez, Brando Perez Esparza, Huzaifa Rampurawala, Brent Carlton, Surej Ravikumar, Carlos Nieva, Sungwon Kim, Hyung-Jin Lee, Amir Sammak, Giordano Scappucci, Menno Veldhorst, Fabio Sebastiano, Masoud Babaie, Stefano Pellerano, Edoardo Charbon, and Lieven M. K. Vandersypen. 2021. CMOS-based cryogenic control of silicon quantum circuits. *Nature* 593, 7858 (01 May 2021), 205–210. https://doi.org/10.1038/s41586-021-03469-4

[80] Amir Youssefi, Itay Shomroni, Yash J Joshi, Nathan R Bernier, Anton Lukashchuk, Philipp Uhrich, Liu Qiu, and Tobias J Kippenberg. 2021. A cryogenic electro-optic interconnect for superconducting devices. *Nature Electronics* 4, 5 (2021), 326–332.