HetArch: Heterogeneous Microarchitectures for Superconducting Quantum Systems

HetArch: Heterogeneous Microarchitectures for Superconducting Quantum Systems

Samuel Stein*†
Pacific Northwest National Laboratory
Richland, Washington, USA

Sara Sussman*
Princeton University
Princeton, New Jersey, USA

Teague Tomesh*
Princeton University
Princeton, New Jersey, USA
Infleqtion
Chicago, Illinois, USA

Charles Guinn*
Princeton University
Princeton, New Jersey, USA

Esin Tureci*
Princeton University
Princeton, New Jersey, USA

Sophia Fuhui Lin
University of Chicago
Chicago, Illinois, USA

Wei Tang
Princeton University
Princeton, New Jersey, USA

James Ang
Pacific Northwest National Laboratory
Richland, Washington, USA

Srivatsan Chakram
Rutgers University
New Brunswick, New Jersey, USA

Ang Li
Pacific Northwest National Laboratory
Richland, Washington, USA

Margaret Martonosi
Princeton University
Princeton, New Jersey, USA

Fred T. Chong
University of Chicago
Chicago, Illinois, USA
Infleqtion
Chicago, Illinois, USA

Andrew A. Houck
Princeton University
Princeton, New Jersey, USA

Isaac L. Chuang
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Michael Austin DeMarco
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
Brookhaven National Laboratory
Upton, New York, USA

ABSTRACT

Noisy Intermediate-Scale Quantum Computing (NISQ) has dominated headlines in recent years, with the longer-term vision of Fault-Tolerant Quantum Computation (FTQC) offering significant potential albeit at currently intractable resource costs and quantum error correction (QEC) overheads. For problems of interest, FTQC will require millions of physical qubits with long coherence times, high-fidelity gates, and compact sizes to surpass classical systems. Just as heterogeneous specialization has offered scaling benefits in classical computing, it is likewise gaining interest in FTQC. However, systematic use of heterogeneity in either hardware or software elements of FTQC systems remains a serious challenge due to the vast design space and variable physical constraints.

This paper meets the challenge of making heterogeneous FTQC design practical by introducing HetArch, a toolbox for designing heterogeneous quantum systems, and using it to explore heterogeneous design scenarios. Using a hierarchical approach, we successively break quantum algorithms into smaller operations (akin to classical application kernels), thus greatly simplifying the design space and resulting tradeoffs. Specializing to superconducting systems, we then design optimized heterogeneous hardware composed of varied superconducting devices, abstracting physical constraints into design rules that enable devices to be assembled into *standard cells* optimized for specific operations. Finally, we provide a heterogeneous design space exploration framework which reduces the simulation burden by a factor of 10^4 or more and allows us to characterize optimal design points. We use these techniques to design superconducting quantum modules for entanglement distillation, error correction, and code teleportation, reducing error rates by 2.6×, 10.7×, and 3.0× compared to homogeneous systems.

*These authors contributed equally to this research.
†samuel.stein@pnnl.gov

MICRO '23, October 28–November 01, 2023, Toronto, ON, Canada

ACM ISBN 979-8-4007-0329-4/23/10…$15.00
https://doi.org/10.1145/3613424.3614300

CCS CONCEPTS

• **Computer systems organization → Quantum computing**; **Heterogeneous (hybrid) systems**.

KEYWORDS

Quantum Computing, Quantum Computing Architecture, Superconducting Quantum Systems

ACM Reference Format:

Samuel Stein, Sara Sussman, Teague Tomesh, Charles Guinn, Esin Tureci, Sophia Fuhui Lin, Wei Tang, James Ang, Srivatsan Chakram, Ang Li, Margaret Martonosi, Fred T. Chong, Andrew A. Houck, Isaac L. Chuang, and Michael Austin DeMarco. 2023. HetArch: Heterogeneous Microarchitectures for Superconducting Quantum Systems. In *56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '23), October 28–November 01, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3613424.3614300

1 INTRODUCTION

The pursuit of a large-scale fault-tolerant quantum computer (FTQC) that provides a significant advantage over classical computers [98, 108] has been in progress for nearly three decades. A FTQC would have high-impact applications in cryptography [109], physics [18, 38], chemistry [53, 103], and machine learning [49, 50] with numerous platforms under development. Some of these, particularly trapped ions, neutral atoms, and superconducting qubit systems, have begun to scale.

The realization of a FTQC has nonetheless been frustrated by high resource costs [8]. Quantum systems are highly sensitive to noise, requiring quantum error correction (QEC) [19, 90], which introduces an overhead that scales with the error rate of devices. In superconducting quantum systems, which are the focus of this paper, research efforts have produced a wide variety of interconnectable quantum devices [9] that each trade off between long and short coherence times, slow or fast gates, large or compact sizes, and multi-qubit or single-qubit storage capacity. These range from the transmon, a compact 2D qubit with fast gates but shorter coherence times ($O(500\mu s)$) [96, 124]) used in the largest systems today [6, 130] to long-lived resonators that store one or more qubits [20, 41, 63] with coherence times exceeding 25ms [82], but with relatively slow gates and large sizes. However, no platform to date has demonstrated — simultaneously in a single device — the required long coherence times, high-fidelity control, compact size, and scale needed for a FTQC, and current estimates for typical applications require hundreds of thousands to millions of physical qubits [8, 73].

On the other hand, a powerful method for reducing the overhead cost of FTQC has been the employment of heterogeneity at both hardware and software levels [118]. This has been established for some time at the software level [79], in terms of the heterogeneous operation of qubit arrays, where algorithm implementations [40] which vary the error correcting code or code depth across subroutines have yielded orders-of-magnitude improvements to state distillation [67] and factoring [43]. Recent work has also shown the power of adapting the operation of systems to natural hardware inhomogeneity in coherence times [27, 107] and gate sets [66, 76].

Experimental progress has also enabled heterogeneity at the hardware level, where devices can be optimized for specific functions and thus relax simultaneous demands for long coherence times, high-fidelity gates, scalable footprint, and effective topology placed by homogeneous architectures [8]. New error correction architectures have been proposed which leverage the differing coherence and gate times of quantum devices [23, 24, 48]. Composite

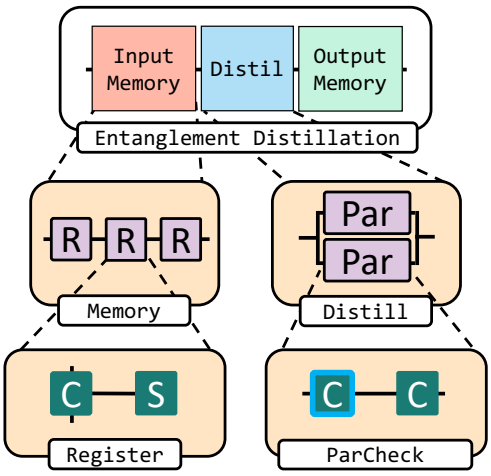


Figure 1: Hierarchical module design for entanglement distillation, a protocol for improving entangled pair fidelity. Storage devices (S), such as multimode resonators, provide multi-qubit memory, while compute devices (C), such as transmons, handle routing and two-qubit operations. These are grouped into standard cells and then into submodules for memory and distillation.

devices consisting of integrated qubit-resonator systems [20, 116] have been proposed and realized, with one experiment demonstrating the highest-ever coherence time improvement from QEC [110].

Several recent examples have found further advantages by leveraging software and hardware heterogeneity together. New QEC codes co-designed with novel hardware have found overhead reductions of 3-10× [36, 75] and can implement highly non-planar codes [75]. An approach to factoring utilizing multimode resonators found an overhead reduction of 1000× [45]. Novel hardware has also inspired quantum simulation approaches [123, 125], with theoretical results indicating cost reduction quadratic in system size [29].

However, despite the success of these ad-hoc examples, the systematic design and validation of heterogeneous quantum systems remains a serious challenge. On the one hand, heterogeneity leads to a combinatorially large design space at each abstraction level that must be navigated, with optimizations interacting across levels [43]. At the same time, the system must adhere to complex rules at multiple scales, from physical constraints on quantum device connectivity and topology [6, 25], to timing of quantum operations and stochastic processes [5], and these rules depend strongly on design choices of physical hardware and implementation. Compounding this, the complexity of quantum systems prohibits simulation beyond modest sizes [65], while models for overcoming this limitation rely on analyzing errors [31] as a system scales, with no clear generalization to situations where errors and their impact vary device-by-device. A framework is needed which simplifies the design process, addressing algorithm requirements, device selection, and ensuring compliance with experimental constraints to enable widespread design of heterogeneous quantum systems.

Of course, classical computing has long employed heterogeneous designs. Modern systems adopt heterogeneity from the level of

nodes [133], to processors [115], down to individual circuit designs [47]. Key to this design process is a hierarchical approach, with microarchitecture [115] organizing modules inside processors around critical subroutines or application kernels, while Very Large Scale Integration (VLSI) techniques [77, 128] create physical designs using a hierarchy to break down complex algorithmic needs into successively smaller pieces until a design can be created [106]. However, the adoption of these methods in quantum systems is challenging. Qubit entanglement underlies the advantage of quantum systems [5, 51, 86], but can often work in opposition to the modular approaches familiar in classical computing [115]. In addition, the extreme sensitivity of errors in quantum systems requires error correction, potentially at multiple levels [114] with varied codes [56]. Furthermore, communication in quantum systems admits fundamentally non-classical mechanisms including quantum teleportation [81, 90], given appropriate pre-distribution of Bell pairs [5].

Still, considerable literature has called for an analogue of large scale design for homogeneous quantum systems [61, 78, 100, 112, 129], with additional focus on the theoretical aspects of device and circuit verification [87, 126]. At the bottom of the stack, libraries such as scQubits [46, 92] and Qiskit Metal [84] support the detailed layout of the elements of devices used in homogeneous and heterogeneous systems. Hybrid systems [131], consisting of multiple qubit platforms such as ions and atoms or superconducting devices, have been proposed[28, 30, 55], but transduction challenges [121] have hindered coupling distinct platforms. In the nearer term, overarching visions for modular quantum systems [13, 51, 62, 85, 111, 135] have been proposed, but full-stack heterogeneous design, from software to hardware, remains to be tackled.

This paper introduces HetArch, a methodology and toolbox for the systematic design and simulation of heterogeneous quantum microarchitectures, and then uses HetArch to develop recommendations for the best uses of heterogeneity in key QC applications. Our contributions include:

- A **hierarchical hardware synthesis method** in which modules executing high-level quantum subroutines are recursively broken into basic operations for which heterogeneous physical architectures can be designed.
- **Quantum standard cells** as physical architectures optimized for these basic operations, and enumerated **design rules** to enable their systematic design in compliance with physical constraints while minimizing errors.
- A heterogeneous **design space exploration framework** that efficiently simulates performance, even with entanglement, to identify optimal design points among tradeoffs in coherence time, connectivity, and gate sets.

We specialize to the particular case of superconducting hardware, where we give an overview of present-day devices and lay out their design rules. We then use this framework to design heterogeneous architectures for **entanglement distillation, error correction, and code teleportation**, with reductions in error rates of up to **2.6×**, **10.7×**, and **3.0×** relative to homogeneous systems.

The following section presents an overview of our architectural framework. Section 3 outlines the available devices and abstracts the physical constraints into architectural design rules to assemble several quantum standard cells. Section 4 then presents the three example architectures and demonstrates how they may outperform homogeneous systems. Section 5 discusses how this framework can incorporate other quantum platforms and future experimental progress.

2 ARCHITECTURAL FRAMEWORK

In contrast to homogeneous "sea-of-qubits" architectures, heterogeneous QC architectures differentiate between the various functions that a device may be used for. For example, a device's role can be divided into "compute," when participating in a gate operation, and "storage," when idling. A device optimized for compute functionality requires fast, high-fidelity gates, high connectivity, and a diverse gate set. On the other hand, storage functionality requires long coherence times, thus allowing less connectivity and slower gates. By leveraging the natural performance tradeoffs in devices along these lines, a heterogeneous architecture expands the design space for quantum devices.

However, the increased flexibility offered by heterogeneity comes with a price: added complexity in design and operation. Building an effective heterogeneous system necessitates aligning the physical designs and inherent tradeoffs of its components with the performance needs of the subroutines to be executed. These subroutines are intricate, with even the simplest involving tens of 'program qubits' that need to be mapped to the physical devices, often placing diverse demands on the lifetimes, gate times, and topology of circuits. As a result, a method is required to break complex subroutines into smaller components that can be more easily managed by architects when specifying a physical design.

As outlined in the introduction, inspiration can be taken from VLSI design principles. A typical VLSI design process [106] begins with high-level functional specification followed by careful design of the logical commands and instruction set of the system, then chip design, and eventually transistor layout. This hierarchical approach allows designers to define key subroutines common to higher-level classical algorithms and break them into smaller subroutines and operations [120]. Once broken into smaller operations, a physical design optimized for each specific operation can be created, called a *standard cell* [4, 77, 122], which complies with *design rules* [106] imposed by the constraints of the physical system and the interconnection needs of modules. Standard cells can be combined into larger blocks, sometimes called modules, until a full system is designed. Finally, to verify that the designs perform as expected, simulation is performed in a 'simulation hierarchy' [128]. Because there is considerable complexity in the design of VLSI systems, the hierarchies are often treated quite flexibly. For instance, different parts of a system may be treated with a different number of levels, these levels may interface in complex ways, and the choice of levels in the hierarchies is a matter of the experience of and convenience for the designers.

Here we introduce a similar hierarchical architecture for heterogeneous systems. This architecture connects high-level quantum subroutines that are offered as fundamental operations to a user, such as state distillation, error correction, etc., with the low-level physical implementation. At the lowest level, such an architecture

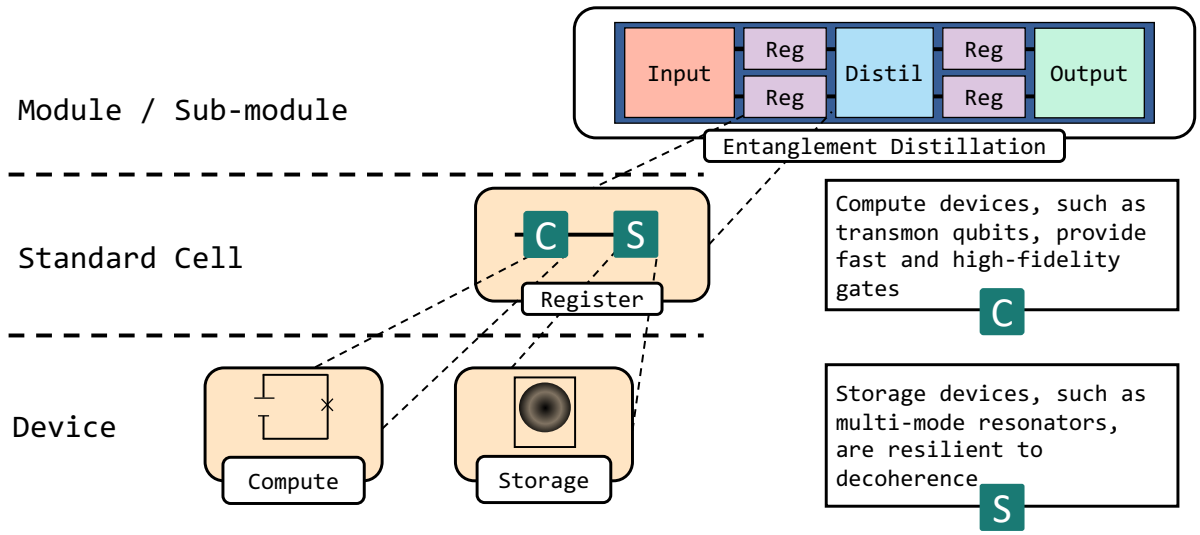


Figure 2: A hierarchical architectural approach that mitigates the exponential complexity of designing heterogeneous systems. High-level quantum subroutines are to be executed by modules, which are designed by breaking the subroutine into successively smaller components until reaching operations sufficiently small that physical architectures can be designed to execute them. This shows the framework for the distillation architecture of Figure 1.

can be mapped to an exact physical layout using existing frameworks such as Qiskit Metal [1]. The goal of this framework is to translate from the needs of a specific quantum subroutine to an abstract layout that can be denoted symbolically in terms of devices, with design rules constraining the resulting physical layout into one that could yield a high-fidelity system.

Figure 2 shows a high-level overview of our proposed framework for a hierarchical design of heterogeneous quantum computing architectures. Specifically, there are three layers of abstraction, namely *modules*, *standard cells ('cells')*, and *devices*:

- *Modules* are responsible for executing *subroutines* for quantum algorithms. Each module offers a specified set of operations on specific input states (if any). These operations are characterized in terms of the average execution time, logical error rate (or fidelity for non-error-corrected systems), and concurrency of operations. Note that I/O is often a key operation for performance [5]. Physically, the module inherits a control overhead and physical footprint from the layers below.
- *Standard cells* perform *operations* such as sequential entangling gates, readout, or syndrome operations. They are the elements which are typically tiled to scale up a system. Examples include a small memory register element, the unit cell of an error correcting code, or routing elements. Standard cells offer a specified set of operations, and are characterized by detailed simulation of these operations, extracting the time, fidelity, and concurrency. Standard cells inherit their control overhead and footprint from their constituent devices.
- *Devices* are the fundamental physical elements capable of storing and manipulating quantum information, such as transmon qubits or multimode resonators. Each device offers a wide array of potential gates and should be considered as acting on arbitrary input states. These operations are characterized by their speed and fidelity, while devices also should be labeled by their control overhead and footprint.

These elements form parallel software and hardware hierarchies: modules execute subroutines, cells execute operations, and devices hold qubits. While in VLSI the hierarchies for software, hardware, and simulation may differ due to system complexity, in these simpler quantum systems the three will be coincident. Furthermore, the hierarchy for systems and their interfaces should be flexible, with modules potentially becoming sub-modules, and standard cells appearing as sub-cells. The labels of module, standard cell, and device are guides to the operations performed and the level at which each of these layers is characterized. Devices, being the atomic unit, cannot be sub-devices. Note the use of standard cell at the physical level, in contrast to recent approaches at the error correction level [34].

Through the abstraction layers, this hierarchical design process provides the additional advantage of limiting exposure to exponential complexity during verification of correctness and evaluation of performance of a final design. At the standard cell level, detailed density matrix computations characterize fidelity and execution time. To characterize a module, multiple standard cells can be jointly simulated by exchanging density matrices. At higher levels, the final output of a module includes execution time and a logical error rate (or fidelity for non-error-corrected operations) relative to the intended output state. To mitigate exponentially growing simulation cost, the performance of several modules is modeled through phenomenological error analysis [31], evaluated in comparison to expected input and output states, ensuring efficient simulation performance.

Device	T_1/T_2	Readout time	Gate set	Gate error (time)	Connectivity	Control Overhead	Footprint	Notes
Fixed frequency qubit [124, 127]	300μs / 550μs	1μs	Arb. 1Q/2Q	1e-3 (100ns)	4	1 charge 1 readout	2 mm x 2 mm	e.g. Transmon
Flux tunable qubit [33, 35]	800μs / 200μs	1μs	Arb. 1Q/2Q	1e-3 (100ns)	4	1 charge 1 flux 1 readout	2 mm x 2 mm	e.g. Fluxonium
3D quantum memory [17, 83]	25ms / 30ms	N/A	SWAP	1e-2 (1μs)	1	N/A	50 mm x 0.5 mm x 1 mm	Requires 2D/3D integration
3D multimode resonator (10 modes)[20]	2ms / 2.5ms	N/A	SWAP	1e-2 (400ns)	1	N/A	100 mm x 100 mm x 10 mm	Requires 2D/3D integration
Future on-chip multimode resonator [20, 41, 64]	1ms / 1ms	N/A	SWAP	1e-2 (100ns)	1	N/A	5 mm x 5 mm	No demonstration

Table 1: Properties of near-term superconducting quantum devices (values estimated from Device column references). The best observed properties for each device were reported and these values have not been demonstrated at scale. For a discussion of the near-term viability of demonstrating a 1ms on-chip multimode resonator, see Section 3.1.

3 DEVICES AND STANDARD CELLS

Heterogeneous design begins by identifying primitives based on the 'top-down' software needs of the quantum algorithm [119]. Once an algorithm has been broken down into these basic operations, design then proceeds 'bottom-up,' assembling devices into standard cells optimized for those basic operations, with standard cells then grouping into modules.

3.1 Quantum Devices

Devices are the fundamental physical objects used for quantum information processing. Various superconducting devices have been created, ranging from compact 500μm x 500μm 2D transmon qubits with fast gates to 3D resonators with multiple-qubit storage capacity and a size of 100cm^3. Here we outline these devices, how to characterize them, and how we can group them for use in a heterogeneous system.

In this paper we focus on a central tradeoff in today's superconducting quantum devices: the competition between long coherence times required for quantum information storage and high connectivity desired for computation [42]. This tradeoff is present both within a type of device (for example, optimizing gate speed may require drive lines with higher couplings that lower coherence time), as well as between device types (as we will see for the longer coherence times and reduced connectivity of 3D resonators). Increasing coherence times while preserving connectivity is an ongoing engineering challenge.

Motivated by the needs of the quantum algorithm we consider in the following sections, we group devices into 'compute' and 'storage' functions, which can then be mapped onto the demands of quantum circuits. Compute devices have high connectivity and fast, high-fidelity gates, with single-qubit capacity. Storage devices have low connectivity, to preserve long coherence times, and multi-qubit capacity. Future tradeoffs we may consider, including potentially new hardware platforms, are discussed in Section 5.

Table 1 shows the key properties of devices. The coherence times T_1 and T_2 define the timescale for amplitude and phase damping errors, respectively. A device's readout time is the time required to measure the system (e.g. in the Z basis). Planar devices (transmons, etc.) are measured by coupling to readout circuitry while resonators are measured by coupling to a qubit and then readout. A device's gates are characterized by their typical durations and average gate fidelities. The connectivity of a device is the number of connections allowed. For the case of resonators, a single transmon is connected to the resonator which can then be connected to other devices. Control overhead is a measure of the extra I/O required to operate a device. For example, the fluxonium can achieve higher T_1 than a fixed frequency transmon but requires a dedicated flux bias line [89].

The primary compute device considered in this work is the planar transmon. The transmon is the only superconducting qubit for which high-fidelity gates have been successfully scaled up to a processor with over two devices [3, 59]. The transmon has known coherence limitations [42, 97, 101], with the highest demonstrated transmon coherence times currently well below 1ms [124]. While the transmon can be made with a tiny footprint of 1μm x 1μm [70, 71], this design has not yet been scaled up to a multi-qubit system.

The primary storage device considered here is the multimode resonator, which functions as a small multi-qubit quantum memory [20]. Resonators have already been experimentally implemented as quantum memories that store the state of planar superconducting qubits [72, 102]. Experimental demonstrations of high-coherence 3D multimode resonators showed coherence times of over 2ms for 8 modes accessible via a single transmon that can store and load qubits with 95% fidelity, 400ns long SWAP gates exchanging states between the transmon and resonator [20]. Moreover, the gate fidelity is expected to be limited only by SWAP gate time and the T_2 of the transmon, so further prototyping should result in fast, high-fidelity gates similar to those between two transmons. Experimental efforts to develop efficient multimode planar resonator designs are underway [74].

For ultra-high-coherence storage, 3D resonators are a promising candidate with coherence times as high as 25ms [83], but these have a large footprint and will be more challenging to precisely

couple to 2D devices [7, 17]. New experimental demonstrations allow us to consider storage devices that could be integrated on-chip. Single-mode planar resonators can now have coherence times of 1ms [41], and micromachined resonators can have coherence times of 5ms [63]. A future option may be nanomechanical resonators with coherence times exceeding 1 second [69] if they can be coupled to supercomputing qubits [93].

The compute-storage assignment is a major simplification of the full range of quantum devices, but in later sections will suffice to enable a wide range of new architectures. Importantly, a single storage device provides both long coherence times to and also many-to-one connectivity between the qubits stored in it and an attached compute device, akin to set associative access in a classical cache through a single port. Storage devices also reduce the overall control overhead relative to a homogeneous design: a multimode resonator with N modes is controlled via a single drive line [20], whereas it takes N drive lines to control N transmons. Furthermore, different qubit platforms will likely have different tradeoffs, and thus require a different approach to grouping devices, which we will return to in Section 5.

3.2 Quantum Standard Cells

Standard cells are functional units built from devices and optimized to perform a few quantum operations. Standard cells form the building blocks of a functional module or sub-module. However, standard cells must be assembled in compliance with physical design constraints, such as connectivity and available operations, in a way which best enables the operations they are to support. Here we list design rules which ensure that a standard cell can be physically implemented, and show how to craft several simple standard cells.

The physical constraints that systems must obey arise from both footprint constraints as well as the need to maximize coherence [9, 42]. The complex series of drive lines and couplings [12, 60] that are used to manipulate quantum information have a large footprint [59], with flip-chip architectures only somewhat alleviating 2D space constraints [15, 57]. On the other hand, every coupling forms a potential vector for quantum information to leak, thus harming coherence [124]. In particular, qubits with readout capabilities are expected to have lower coherence [105]. These considerations lead to the following empirically determined design rules (DRs) for planar devices:

(1) **Compute devices should be connected to at most 4 other devices.**
(2) **Storage devices should be connected to exactly 1 compute device to maximize coherence.**
(3) **Devices in standard cells should have connectivity that reflects their intended use.**
(4) **Compute devices with readout capabilities should be minimal without introducing extra `SWAP`s.**

The performance of a given standard cell is characterized through density matrix simulations at the device level, yielding an output density matrix that can be used to extract the relevant metrics such as operation fidelity. Fidelity computed as error rates and coherence are then used to model each standard cell as a quantum channel on its inputs/outputs, abstracting away all of the device level details, which is key to the scalability of the HetArch methodology. Note that, since standard cells exist one layer above the Device layer, it is possible to swap out different physical devices in a standard cell, e.g. changing the choice of storage unit. Doing so will change the performance of the standard cell as well as its footprint and control overhead.

To illustrate the process of designing a standard cell, we now examine two which will be used in an example application: a register cell, `Register`, and a cell optimized for a parity check operation needed for distillation, `ParCheck`. Entanglement distillation is a quantum application that consumes batches of noisy entangled pairs, performs some distillation protocol, and returns fewer higher purity pairs. We can break this into three key steps of input, distillation, and output (module). Focusing on the distillation portion (standard cell), we can then construct a hardware-efficient layout that targets the specific protocol (standard cell). Table 2 shows all four standard cells used in this paper, with `SeqOp` and `USC` described as they are used in subsequent sections.

The register cell `Register`, shown in Table 2 is designed to minimize errors during idle time while allowing high-fidelity movement to and from the compute device. It should accept incoming qubits, store them, and release them out as needed. To do so, it uses a storage element that is coupled to a single compute element (DR2), without readout (DR4). The compute device may be connected to up to three other devices (DR1), though the minimal number should be used (DR3).

A second simple standard cell that we will use is the `ParCheck`, a cell optimized for parity checks, which must be capable of doing single and two-qubit gates and qubit readout. It is made of two compute devices optimized for fast single-qubit gates and a two-qubit gate between them. One device has a readout resonator allowing parity checks (DR4). It has one connection between the compute devices, allowing each end to link with up to three other units (DR1, DR3).

4 THREE HETEROGENEOUS MICROARCHITECTURES

The architecture presented in Section 2 provides an overview of the heterogeneous design methodology, with the devices and standard cells following the constraints laid out in the design rules. We now employ the devices and standard cells of Section 3 to demonstrate the design process and evaluation of heterogeneous microarchitectures for three examples: entanglement distillation, error correction, and code teleportation, which leverage the long lifetimes of storage units, heterogeneity in compute units, and topology enhancements offered by heterogeneous design. We sweep design parameters of each and quantify the performance improvements in the heterogeneous systems.

To compare each heterogeneous architecture, consisting of compute and storage devices, fairly to the to the homogeneous case, we set as a baseline a system consisting of only compute devices arranged in a square lattice, representing the homogeneous "sea-of-qubits" comparison that current devices employ, as these underlie many 'software-only' heterogeneous approaches [43, 67]. While the homogeneous system lacks the long-lived, high capacity storage units, it is allowed to be as large as needed for maximally efficient

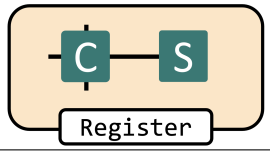
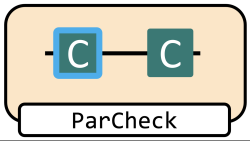
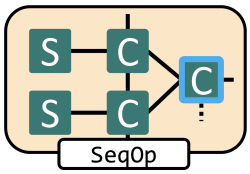
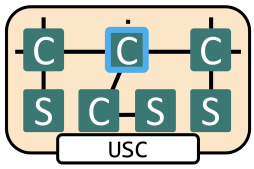
	**Register standard cell** `Register`: A high-capacity storage device coupled to a compute device which manages input/output, with up to three connections from the compute device. Characterized by the load/save time and fidelity to swap a qubit between compute and storage as well as the storage decay time T_S.
	**Parity check cell** `ParCheck`: Two compute devices coupled to read data in, perform one and two-qubit operations, and then measure one qubit, with up to three connections from each qubit. Characterized time and fidelity to move two qubits in and out, one and two-qubit gates, and readout time.
	**Sequential operations cell** `SeqOp`: Optimized for many sequential two-qubit operations and parity checks among a collection of qubits. Contains two `Register` standard cells as subcells, coupled to each other and a compute device with readout for parity checking. There are up to two connections from each `Register` compute device and an optional connection from the parity check compute. Characterized by the time and fidelity to execute a series of two-qubit gates among qubits stored in the `Register` subcells.
	**Universal stabilizer cell** `USC`: Contains three `Register` standard cells as subcells, with a central parity check compute device, with one connection available from each `Register` compute device and the parity check compute device. Characterized by the time and fidelity to execute stabilizer checks among qubits stored in the `Register` subcells, with the central parity check device holding the ancilla qubit.

Table 2: Quantum standard cells used in this paper. These are assembled in accordance with the design rules presented in Section 3.1, and optimized for particular operations. Devices with readout are outlined in blue.

transpilation, with the caveat that QEC will only be applied if it is applied in the heterogeneous system. If an optimal square lattice transpilation is known, as in the case of surface code, it will be used; otherwise the Qiskit transpiler [99] at the highest optimization setting is used.

Throughout this section, compute and storage coherence times are denoted as T_c and T_s, with $T_1 = T_2$. Unless otherwise specified, compute devices will have lifetimes $T_C = 0.5$ms, while the wide array of storage options allows T_S to vary from 0.5ms to 50ms. All gates are assumed to be coherence-limited, with two-qubit gate times (including `SWAP`s) of 100ns, single qubit gate times of 40ns, and 1μs error-free readout. Classical communication times are neglected.

4.1 Entanglement Distillation

The production and distribution of entanglement is a key subroutine within many quantum applications. This is typically achieved by preparing and distributing a Bell state entangled pair (EP), for example $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Current EP generation methods, including on-chip distribution, off-chip microwave connections, and hybrid microwave-optical schemes, suffer from noise and slow generation rates. Entanglement distillation protocols can correct noise by consuming multiple low-fidelity EPs to create a smaller number of higher fidelity EPs. Distillation protocols are expected to play a key rule in networked quantum systems [58], and here we focus on the DEJMPS protocol [32].

However, the effectiveness of distillation is severely impaired by low EP generation rates, as errors accumulate in stored EPs awaiting sufficient numbers for distillation [5]. Heterogeneous design can alleviate this by leveraging the differing lifetimes of compute and storage components to provide both a memory for storing EPs and a fast distillation protocol. Here, we design a module for entanglement distillation which purifies EPs until they reach a threshold fidelity and then offers them on demand to the rest of the system. We use our quantum standard cells, and focus on hardware parameters comparable to the case of microwave-to-optical conversion [5]: target fidelity is set to 99.5% and EP generation is random, with average times of 1-100μs and infidelities on the order of 0.01-0.1, 10-1000x slower and 10-100x noisier than compute operations.

The distillation operation comprises three subroutines. First, EPs are preserved in an input memory until enough are acquired for distillation. Next, the distillation protocol, including gates, measurement, and correction, is executed. Finally, EPs are preserved in an output memory until used by other modules. The entanglement distillation module contains a sub-module for each of these steps.

An example entanglement distillation module is shown in Figure 1. The input memory consists of one or more `Register` standard cells. The distillation sub-module is comprised of `ParCheck` standard cells, which can perform the parity checks needed for distillation. Finally, the output memory is also a collection of `Register` standard cells. In addition to the physical design of the distillation module, a complication arises in the design of its operation and scheduling. Because we assume the case of probabilistic EP generation, the distillation module must dynamically respond to

successful generations, requiring complex coordination. We implement a greedy scheduler with the following priorities: (1) re-distill existing pairs if it would yield improvement, (2) move distilled pairs to output memory (3) distill new pairs if available (4) store incoming pairs in memory.

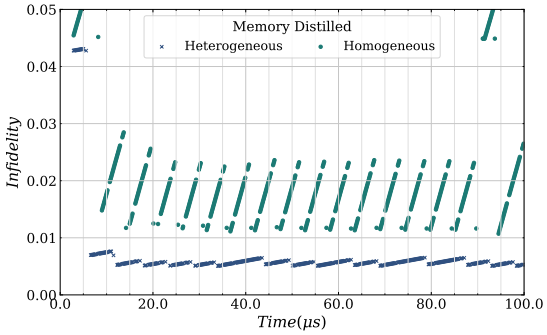


Figure 3: Entanglement distillation over time. Here we plot the best EP in the output register for heterogeneous (blue) and homogeneous (green) systems, with probabilistic EP generation. In the heterogeneous system, T_S = 12.5ms/mode. In the homogeneous system, $T_S = T_C$ = 0.5ms. The heterogeneous system preserves fidelity while waiting for lower-purity pairs to be received and purified.

Capacity demands for memories and the distillation module are determined by input EP rates, distillation times, desired output rates, and bottlenecks between these factors. Sweeping the parameter space with a single EP input line with generation rate of 0.1 − 100MHz, two `Register` cells for the input memory with three modes each, one `ParCheck` cell for distillation, and one output `Register` with three modes were found sufficient to achieve high fidelity distilled EPs without overflow in any sub-module.

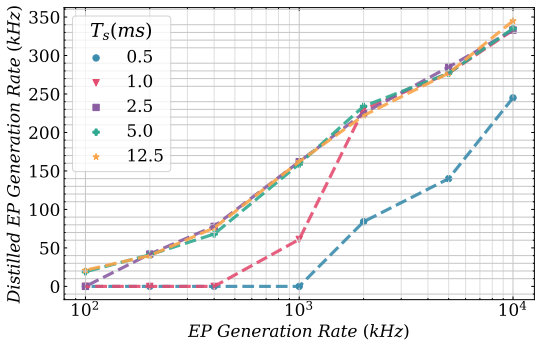


Figure 4: EP distillation rate to fidelity > .995 as a function of EP generation rate and storage coherence time T_S. This plot includes both heterogeneous and homogeneous systems; in the homogeneous system, $T_S = T_C$ = 0.5ms.

Simulation results for T_s = 12.5ms are shown in Fig. 3. When compared to a homogeneous architecture, using both input and output memories allows the heterogeneous system to achieve a lower infidelity when distilling, as shown by the lower infidelity minima, as well as the ability to preserve that infidelity for longer, as shown by the shallower fidelity decay per unit time. Fig. 4 shows the rate of production of distilled EPs with fidelity above 0.995 as a function of the generation rate of raw EPs and T_S. Distilled EP rates increase with generation rate both because more pairs are available to distill in a given time and because of the reduced idle errors, which are even further reduced in a heterogeneous system. Heterogeneous systems with T_S = 2.5ms and greater outperform the homogeneous system with $T_S = T_C$ = 0.5ms by a factor of 2× or more.

The storage device performance required varies with the generation rate. For generation rates in excess of 10kHz, storage lifetimes of T_S = 1ms are sufficient to achieve near-maximum performance, with diminishing returns for high lifetimes. For lower generation rates T_S = 2.5ms or higher is required. Notably, for generation rates below 1000kHz, the homogeneous system composed of only compute devices fails to distill any pairs to threshold fidelity due to idling errors, while heterogeneous systems still allow generation on the order of 100kHz.

4.2 Error Corrected Quantum Memory

In a fault-tolerant quantum computer, error correction will be constantly running on noisy physical qubits, making it one of the most important subroutines [73]. Programmatically, error correction requires two kinds of qubits, data and ancilla, with differing needs. Data qubits collectively store the state of the logical qubit, and so require higher coherence times, while ancilla qubits are used to perform error correction checks and hence need fast gates and readout.

In this section, we consider two heterogeneous error correction architectures. The first explores planar surface code architectures, leveraging heterogeneity in coherence times for data and ancilla qubits. The second stores data qubits in high-capacity storage devices, leveraging both their long coherence times and many-to-one topology to provide a universal error correction architecture capable of implementing many QEC codes efficiently in a single device.

QEC codes aim to perform 'below threshold' [39], wherein larger code distances correlate with a lower logical error rate or, for codes with only one distance, below 'pseudo-threshold' [22], when the error rate of the system is below that of the hardware. As two-qubit gate errors are a major limitation of experimental architectures, in this subsection two-qubit gates are taken to have an error rate of 1%, which scaled-up experiments have reported [3].

4.2.1 Planar Surface Code. Planar surface code error architectures have begun to be experimentally demonstrated and scaled in homogeneous systems [3, 39, 59], with several heterogeneous approaches being proposed as well [48, 91, 117]. In HetArch, the fundamental standard cell of the surface code is a data and ancilla qubit pair (Fig. 5). Since both undergo two-qubit gates too frequently to be storage devices, we can amend our approach and consider two classes of compute devices. The first class, optimized for high coherence, without readout, will store the data qubits; the second will be optimally coupled to a readout resonator for fast measurements and function as the ancilla. This directly maps to the parity check cell `ParCheck`. Because both are compute devices, we denote their coherence times

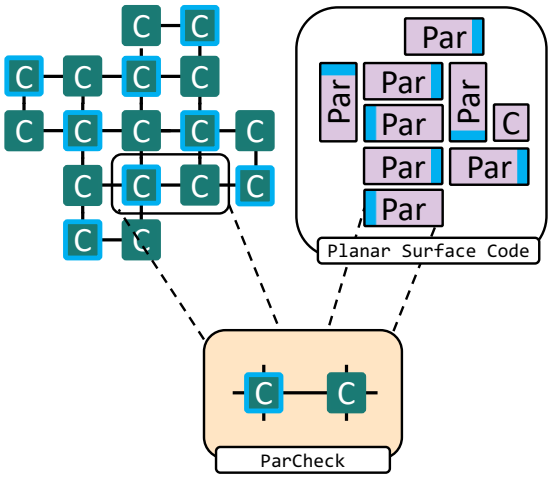


Figure 5: Hierarchical structure of a planar surface code quantum memory.

(with $T_1 = T_2$) as T_{CD}, T_{CA} for data and ancilla qubits, respectively. Note that due to the layout of the surface code, there will always be one more data than ancilla qubit.

Simulation of a heterogeneous surface code is implemented using the Stim package [2] which supports Monte Carlo experiments with a circuit-level noise model and allows for the performance of relatively large surface codes to be extrapolated from the properties of a standard cell: T_{CA}, T_{CD}, the single- and two-qubit gate durations and fidelities, and durations and fidelities of measurement and reset. These properties may be set to target the behavior of a future system, or, as in this case, extrapolate from current experimental performance. For this planar surface code example only, we take the baseline compute coherence to be 0.1ms (rather than 0.5ms). This is to make the example more relevant, as the planar surface code is

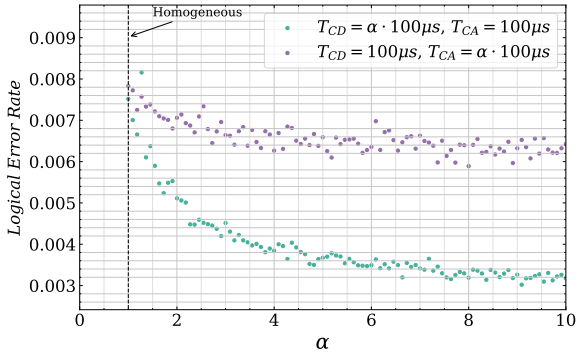


Figure 6: Logical error rate per cycle as a function of data and ancilla qubit coherence for distance d=13 surface code. In the homogeneous system, data qubit coherence T_{CD} equals ancilla qubit coherence $T_{CA} = 0.1$ms. Increasing T_{CD} by a factor of α leads to a greater improvement than increasing T_{CA} by a factor of α.

the only one of our examples that has already been implemented in a full experiment [3, 59].

The results of increasing the data and ancilla qubit coherence times are shown in Fig. 6. As expected, increasing the coherence times of ancilla qubits does not reduce the error rate as much as increasing the coherence times of data qubits, largely because a major source of error is data qubit idling during the $1\mu s$ ancilla measurement. However, increasing the coherence times of data qubits leads to a 2.5x reduction in logical error rate for a data qubit coherence time of approximately 0.5ms, within experimental reach [124], especially since data qubits do not require readout and can be designed with minimal leakage [94]. Note that increasing data coherence times further leads to diminishing performance returns.

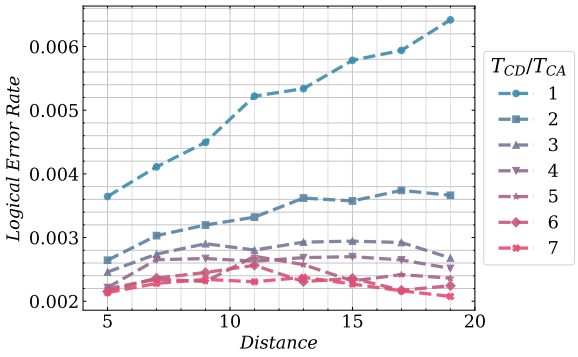


Figure 7: Logical error rate per cycle for code distances $d = 5$ to distance $d = 18$ as a function of the ratio T_{CD}/T_{CA}. When this ratio increases, the code moves below its threshold. Both heterogeneous and homogeneous system performance is plotted; in the homogeneous system, $T_{CD}/T_{CA} = 1$.

We also illustrate how the ratio T_{CD}/T_{CA} affects the performance of the overall surface code. In Fig. 7 we plot the logical error rate for code distances $d = 5$ to distance $d = 18$ as a function of the ratio T_{CD}/T_{CA}. When this ratio increases, the code moves below its threshold. Even with present-day two-qubit gate errors of 1% and coherences on the order of 0.1ms, the surface code can benefit from a heterogeneous design where data qubits are optimized for coherence and ancilla qubits are optimized for readout [3]. Diminishing performance returns beyond $T_{CD}/T_{CA} = 5$ suggest other limiting factors, such as two-qubit gate infidelity.

4.2.2 Universal Error Correction Module. For homogeneous systems, the performance of quantum error correcting codes is highly dependent on matching the required topology of code checks to the connectivity of hardware, leading to severe restrictions on the variety of codes that can be implemented. Here we leverage the effective topology of high-capacity (10-mode) storage devices to create heterogeneous systems which are agnostic to the topology of a stabilizer QEC code, functioning as universal error correction (UEC) modules at the cost of serializing QEC checks [36], unlike homogeneous systems which operate QEC checks in parallel.

The basic function of the memory is to execute stabilizer checks. We store the data qubits in `Register` cells with 10 modes per

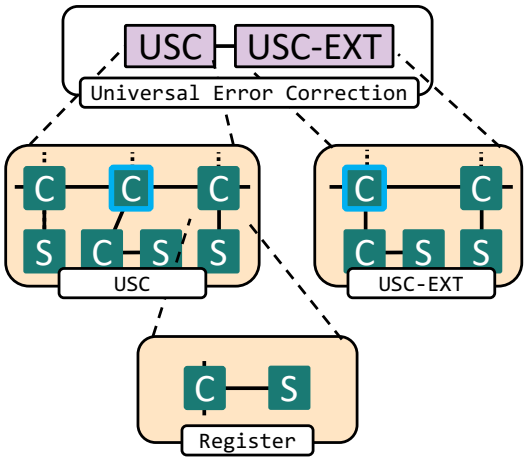


Figure 8: Hierarchical design of a universal error correction module.

storage element, keeping them in the multi-qubit storage devices when idling, and swapping them to compute elements only during stabilizer checks or logical gate operations. Because it contains many qubits, the storage device provides many-to-one coupling to compute device within the Register cell and also provides high-coherence while idling. Flag circuits may be used to ensure fault-tolerance [21, 25].

The central element of a general error correction module, called the universal stabilizer cell (USC), was originally described in Table 2 and we now describe further. The design features three Register sub-cells (satisfying DR2) arranged around a central compute device with readout which will function as the ancilla qubit that performs stabilizer measurements. This design represents the edge of the design space; USC designs with up to four Register cells were considered, and would allow even larger codes, but would exhaust the connectivity of the central ancilla compute (DR1), which is needed to connect to other cells or modules. Furthermore, this architecture allows for minimal number of internal connections (DR3) and readout qubits (DR4). Three additional outgoing connections are available if greater connectivity is required. As shown in Fig. 8, defining an extension cell with only two Registers, USC-EXT, allows multiple units to be chained together while still respecting DR1-5. Characterizing the USC as a sub-cell, rather than a sub-module, reflects the fact that detailed density matrix simulations will need to be carried out spanning a USC and one or more USC-EXTs.

The UEC module is able to adapt to any stabilizer code up to 30 qubits, and, with USC-EXTs added, to any code that can be partitioned in 1D for larger sizes. What enables this is the serialized error correction checks, in effect trading QEC code flexibility for long execution time and so requiring long-lived storage [44]. Qubit assignment to the various Register devices, as well as the gate schedule, can be highly optimized. This first study utilizes a brute-force search to optimize assignments, limiting the model to 30 data qubits, and then outputs a schedule of operations which seeks maximum possible parallelism while minimizing the time each data

Code	PT	Het.	Hom.	Red.
RM	0.0254	0.0353	0.1660	4.7x
17QCC	0.1608	0.0284	0.0990	3.5x
ST	0.1291	0.0097	0.1034	10.7x
SC3	-	0.0186	0.0061	.3x
SC4	-	0.0158	0.0092	.6x

Table 3: QEC code, pseudothreshold (PT), logical error rates, and error reduction (Red.) for the heterogeneous (Het.) architecture with $T_S = 50$ms and homogeneous (Hom.) architecture.

qubit spends outside storage devices. Replacing the brute-force search with scalable approaches may be the subject of future work.

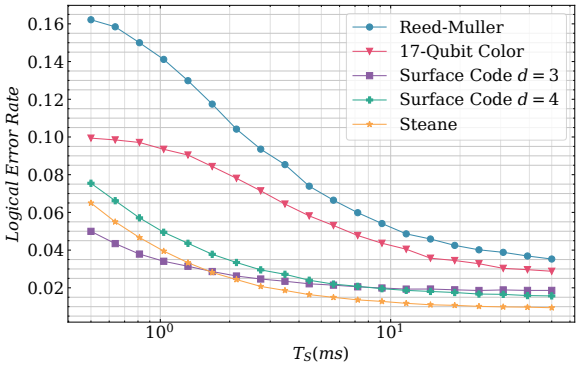


Figure 9: Performance of selected QEC codes on the universal error correction module.

Fig. 9 shows the performance of several QEC codes on the universal error correction module simulated using Stim [2]. These codes are the surface code with $d = 3, 4$ (SC3, SC4), the Steane 7-qubit code (ST), the 17-qubit color code (17QCC), and the 15-qubit Reed-Muller (RM) code [11]. The surface codes are designed for a square lattice, and so fit naturally on the homogeneous architecture, while the other three are non-square (and the Reed-Muller is non-planar). For these small codes, two register units, and hence a single USC cell, are sufficient to maximize the parallelism of the checks while minimizing the number of SWAP operations that are needed. Performance of the codes, along with their pseudothresholds [22] is listed in Table 3. Thresholds for the surface code assume parallel stabilizer checks, and so do not apply to this serial execution, but we can see that the $d = 4$ surface code outperforms the $d = 3$ code, suggesting that the system is below threshold. The 17-qubit color and Steane codes achieve error rates well below pseudo-threshold, while the Reed-Muller code misses pseudothreshold by approximately 50%. Because the surface code is native to the square lattice, the UEC module underperforms the homogeneous architecture by 3.0x and 1.7x for $d = 3$ and $d = 4$, respectively. On the other hand, the Reed-Muller, 17-qubit color, and Steane codes outperform the homogeneous architectures by 4.7x, 3.5x, and 10.7x, respectively.

4.3 Code Teleportation

No single QEC code can provide a fault-tolerant gate set which is simultaneously universal and transversal [37, 134]. Alongside state injection [14, 68] and stabilizer sequences [11], code teleportation (CT) [26] addresses this issue, while also allowing conversion between different depth codes [118]. CT functions by preparing CT resource states in the form of Bell states between logical codes, i.e. $|\Phi\rangle^+_{AB} = \frac{1}{\sqrt{2}}|\bar{0}_A\bar{0}_B\rangle + |\bar{1}_A\bar{1}_B\rangle$ for codes A and B, so that performing logical teleportation both teleports the state and changes the QEC code.

However, a module for CT state preparation must be able to adapt to the different needs of at least two codes; this is particularly true if one of the codes is to have transversal T gates, which have high-weight, non-planar checks [11] not easily realized in traditional arrays. In that case, not only must code teleportation adapt to two codes itself, but bridge between two physical architectures, likely introducing a weak link with additional noise.

In this section, we design a dedicated CT module that functions between *any* two arbitrary stabilizer codes up to 30 qubits. The CT module leverages both the entanglement distillation and universal error correction (UEC) designs as sub-modules, along with a new sub-module for CAT state generation. In particular, the flexibility of the UEC module allows the resulting CT module to act between any two codes within a single physical architecture, while the entanglement distillation module bridges the weak link between the two sides of the computer.

The six steps of CT are shown in Figure 10, following [26]. The key resources to create a CT state are logical $|\bar{+}\rangle$ states in the A and B codes, a shared CAT state of size $|A| + |B|$, and EPs that will be used to entangle and verify the CAT state. First, EPs are created (1) and used for remote gates [113] to create a full CAT state of size $|A| + |B|$ (2). Then, logical $|\bar{+}\rangle$ states are created (3), and parallel CNOT gates entangle the logical $|\bar{+}\rangle$ states with the CAT state (4). Finally a logical measurement is performed (5) and correction applied if needed (6). At the end of this process (dashed red line in Figure 10) the CT module will have successfully prepared the

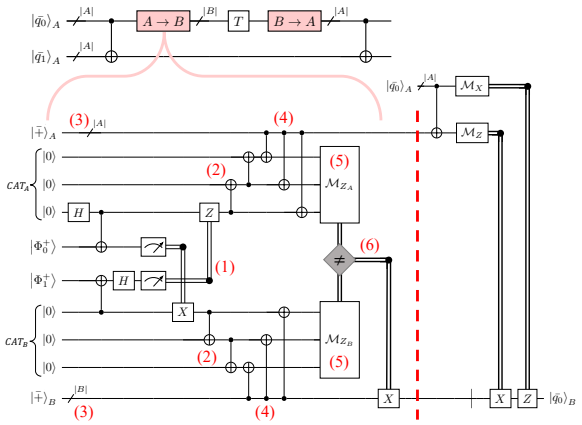


Figure 10: Example program (top) illustrating a compiler inserting code teleportation operations [26] (bottom).

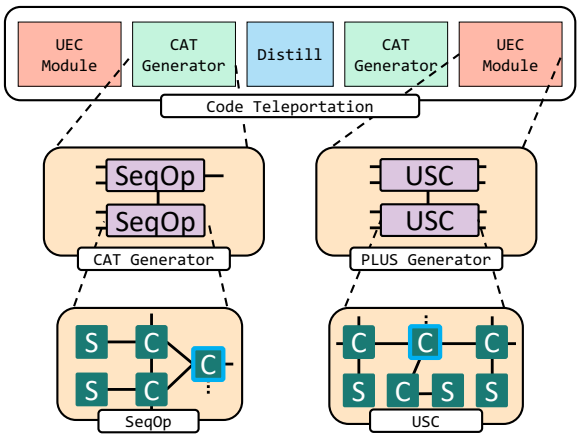


Figure 11: Hierarchical architecture of a code teleportation module.

code teleportation state Φ^+_{AB} which can be consumed to teleport and switch the QEC of an input state.

For creating a physical design, the creation of these resource states each constitutes a subroutine, and each receives a hardware module. As shown in Fig. 11, there are a total of five distinct sub-modules: an entanglement distillation module, two CAT generators, and two UEC modules, with the entanglement distillation and UEC modules described in Sections 4.1 and 4.2.2.

The new component of the CT module is the CAT generator sub-module. This subroutine requires many sequential CNOTs with the result verified by parity checks. The Sequential Operations Cell (SeqOp), shown in Fig. 11, is optimized for this purpose, with two 10-mode Register subcells connected by a triangle of compute units, one with readout. This SeqOp design was selected after many design cycles as it balances sequential operation performance and connectivity offered to other cells and modules. The CAT state generator enables a shortest depth CNOT between memories, while still allowing for ancilla parity readout. Relative to the USC, it offers direct two-qubit gates between qubits stored in the Register cells, while still allowing parity checks.

A major source of design complexity within the code teleportation module is the need to support long range interactions between many of the sub-components, but this is mitigated in this architecture by the built-in storage capabilities of the entanglement distillation, UEC, and CAT generation sub-modules. At the same time, the universal capabilities of the UEC module allow the CT module to teleport between any two codes, regardless of their underlying topology, with an output from the ancilla compute of the UEC module allowing for transversal logical gates with an external module. The price of this flexibility is the serialization of stabilizer checks, which can only be afforded with long storage coherence times.

To evaluate the CT module, a target fidelity of 0.995 is set for the entanglement distillation sub-module. Simulation then proceeds as in Section 4.1 for the entanglement distillation sub-module and Section 4.2.2 for the UEC sub-module. To simulate CAT state generation with realistic noise, a modular approach is also taken, modeling one

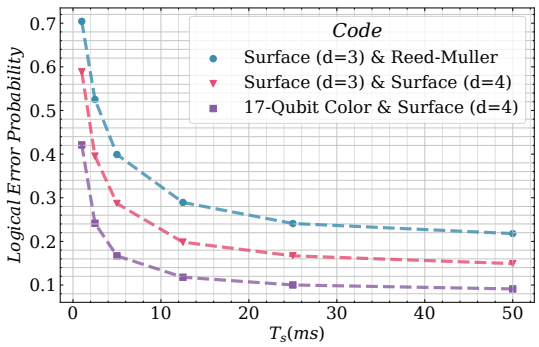


Figure 12: Code teleportation performance using a heterogeneous architecture between two codes. EP generation rate is 1000kHz, with a target distillation fidelity of 99.5%. Homogeneous results are presented in Table 4.

code at a time, abstracting the remote CNOT gate [5], and breaking the modeling of a single CAT generation into smaller CAT states, with multiplicative compounding fidelities. The parallel CNOT gate from step (4) between the CAT and $|\mp\rangle$ states is treated by similarly modeling errors in the CAT generator and UEC modules. For the logical $|\mp\rangle$ state, the same approach described in Section 4.2 yields logical error probability. To create the module level error model, independent error rates [31] are summed.

Figure 12 shows the logical error probability in the prepared CT state, where the error probability decreases significantly as storage lifetime increases. The logical error rate performance is best for the largest codes with low (pseudo-) thresholds, namely the $d = 4$ surface code and 17-qubit color code. The demand for high lifespan entanglement distillation, idling errors from CAT state parity checks, and errors during $|\mp\rangle$ state stabilizer measurements reveal the substantial benefits of long-lived storage for code teleportation. For the simpler surface codes, T_s times above 10ms lead to diminishing returns, while for the more complicated codes, T_s times above 50ms, beyond what we consider for near-term devices, may be advantageous.

Furthermore, Table 4 shows that heterogeneous CT systems outperform their homogeneous counterparts for every pair of codes studied. The expected advantage for non-planar codes is due to the UEC module's higher efficiency. Surprisingly, even for planar codes, heterogeneous systems outperform homogeneous ones, thanks to the enhanced performance of the entanglement distillation and CAT generation modules. Notably, some homogeneous experiments were unable to achieve the 99.5% fidelity EP target, as the entanglement distillation sub-module failed to reach the desired fidelity with EP generation rates of 1000kHz. The most significant reduction occurs between Reed-Muller and surface code (d=3), yielding a 2.96x reduction in logical error probability from 0.500 (essentially mixed) to 0.169. On average, error probabilities are reduced by 2.33x, with a minimum reduction of 1.60x, showcasing the remarkable potential of heterogeneous architectures.

	RM	17Q CC	ST	SC (d=3)	SC (d=4)
RM	-	0.284	0.189	0.169	0.202
17Q CC	0.700	-	0.221	0.202	0.234
ST	0.548	0.516	-	0.096	0.132
SC (d=3)	0.500	0.465	0.192	-	0.111
SC (d=4)	0.540	0.507	0.256	0.178	-

Table 4: Logical error probabilities of code teleportation, for heterogeneous (top right) and homogeneous systems (bottom left). Codes abbreviated as in Section 4.2.2.

5 CONCLUSION

In this paper, we presented a novel study of heterogeneous quantum microarchitectures. By developing a hierarchical approach for aligning software and hardware needs, elucidating design rules for assembling devices into standard cells, and exploring design spaces for three heterogeneous quantum applications, we systematically designed highly efficient systems tailored to their applications, delivering error rate reductions for entanglement distillation, error correction, and code teleportation of up to 10.7×.

As we look to the future, we anticipate that experimental progress will yield even greater hardware heterogeneity. In atomic and ion systems, the use of multiple species or spin states presents tradeoffs involving multi-qubit gates, global control, and communication bottlenecks. Hybrid systems comprising atoms [10, 132], ions [16, 54, 95], and superconducting devices, may be challenging to achieve experimentally but would combine high-speed superconducting compute capabilities with the long lifetimes of ions and atoms [28, 30, 88, 104], albeit with a memory-compute bottleneck [55]. Even within superconducting systems, progress in routers [76, 136] opens new connectivity, while variability within superconducting devices [42] in fact offers functionality more like p-cells [52] in classical systems.

Through these future developments, we envision that the hierarchical HetArch methodology, breaking algorithms into subroutines to be executed by modules, and subroutines into operations to be executed by standard cells, will produce systems that could scale upwards of 1000 qubits. However, the resulting architectures and standard cells will likely be completely different. Future quantum systems may encompass many levels of memory [118], with communication devices [80], caches, and buffers. Compute regions may contain specialized hardware for Fourier transforms, modular multiplication, and local registers. In the burgeoning field of networked quantum systems, dedicated designs for both distillation modules and repeaters are likely to play an integral role. Through integrated heterogeneous design of software and hardware, the community is poised to explore a quantum computing design space as vast as that of classical computing.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-design Center for Quantum Advantage (C2QA) under contract number DE-SC0012704. This research used resources of the Oak Ridge Leadership Computing Facility (OLCF),

which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830. This work is funded in part by EPiQC, an NSF Expedition in Computing, under award CCF-1730449; in part by STAQ under award NSF Phy-1818914; in part by the US Department of Energy Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing Program; and in part by the NSF Quantum Leap Challenge Institute for Hybrid Quantum Architectures and Networks (NSF Award 2016136) and in part based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers. F. T. Chong is Chief Scientist for Quantum Software at Infleqtion and an advisor to Quantum Circuits, Inc. S. Sussman is supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. This material is based upon work supported by (while Martonosi was serving at) the National Science Foundation. The authors express gratitude to Kaitlin N. Smith, Lev Krayzman, Alec Eickbusch, and Volodymyr Sivak for insightful discussions.

REFERENCES

[1] 2023. Qiskit Metal. https://qiskit.org/metal/.

[2] 2023. Stim Github repository. https://github.com/quantumlib/Stim.

[3] Rajeev Acharya, Igor Aleiner, Richard Allen, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Juan Atalaya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Joao Basso, Andreas Bengtsson, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, Leon Brill, Michael Broughton, Bob B. Buckley, David A. Buell, Tim Burger, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Ben Chiaro, Josh Cogan, Roberto Collins, Paul Conner, William Courtney, Alexander L. Crook, Ben Curtin, Dripto M. Debroy, Alexander Del Toro Barba, Sean Demura, Andrew Dunsworth, Daniel Eppens, Catherine Erickson, Lara Faoro, Edward Farhi, Reza Fatemi, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, William Giang, Craig Gidney, Dar Gilboa, Marissa Giustina, Alejandro Grajales Dau, Jonathan A. Gross, Steve Habegger, Michael C. Hamilton, Matthew P. Harrigan, Sean D. Harrington, Oscar Higgott, Jeremy Hilton, Markus Hoffmann, Sabrina Hong, Trent Huang, Ashley Huff, William J. Huggins, Lev B. Ioffe, Sergei V. Isakov, Justin Iveland, Evan Jeffrey, Zhang Jiang, Cody Jones, Pavol Juhas, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Tanuj Khattar, Mostafa Khezri, Mária Kieferová, Seon Kim, Alexei Kitaev, Paul V. Klimov, Andrey R. Klots, Alexander N. Korotkov, Fedor Kostritsa, John Mark Kreikebaum, David Landhuis, Pavel Laptev, Kim-Ming Lau, Lily Laws, Joonho Lee, Kenny Lee, Brian J. Lester, Alexander Lill, Wayne Liu, Aditya Locharla, Erik Lucero, Fionn D. Malone, Jeffrey Marshall, Orion Martin, Jarrod R. McClean, Trevor Mccourt, Matt McEwen, Anthony Megrant, Bernardo Meurer Costa, Xiao Mi, Kevin C. Miao, Masoud Mohseni, Shirin Montazeri, Alexis Morvan, Emily Mount, Wojciech Mruczkiewicz, Ofer Naaman, Matthew Neeley, Charles Neill, Ani Nersisyan, Hartmut Neven, Michael Newman, Jiun How Ng, Anthony Nguyen, Murray Nguyen, Murphy Yuezhen Niu, Thomas E. O'Brien, Alex Opremcak, John Platt, Andre Petukhov, Rebecca Potter, Leonid P. Pryadko, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Negar Saei, Daniel Sank, Kannan Sankaragomathi, Kevin J. Satzinger, Henry F. Schurkus, Christopher Schuster, Michael J. Shearn, Aaron Shorter, Vladimir Shvarts, Jindra Skruzny, Vadim Smelyanskiy, W. Clarke Smith, George Sterling, Doug Strain, Marco Szalay, Alfredo Torres, Guifre Vidal, Benjamin Villalonga, Catherine Vollgraff Heidweiller, Theodore White, Cheng Xing, Z. Jamie Yao, Ping Yeh, Juhwan Yoo, Grayson Young, Adam Zalcman, Yaxing Zhang, and Ningfeng Zhu. 2023. Suppressing quantum errors by scaling a surface code logical qubit. *Nature* 614 (2023), 676–681. Issue 7949. https://doi.org/10.1038/s41586-022-05434-1

[4] S. Alagic and M.A. Arbib. 1978. *The Design of Well-Structured and Correct Programs.* Springer New York. https://books.google.com/books?id=vscmAAAAMAAJ

[5] James Ang, Gabriella Carini, Yanzhu Chen, Isaac Chuang, Michael Austin DeMarco, Sophia E. Economou, Alec Eickbusch, Andrei Faraon, Kai-Mei Fu, Steven M. Girvin, Michael Hatridge, Andrew Houck, Paul Hilaire, Kevin Krsulich, Ang Li, Chenxu Liu, Yuan Liu, Margaret Martonosi, David C. McKay, James Misewich, Mark Ritter, Robert J. Schoelkopf, Samuel A. Stein, Sara Sussman, Hong X. Tang, Wei Tang, Teague Tomesh, Norm M. Tubman, Chen Wang, Nathan Wiebe, Yong-Xin Yao, Dillon C. Yost, and Yiyu Zhou. 2022. Architectures for Multinode Superconducting Quantum Computers. (2022). https://doi.org/10.48550/ARXIV.2212.06167

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

[7] C. Axline, M. Reagor, R. Heeres, P. Reinhold, C. Wang, K. Shain, W. Pfaff, Y. Chu, L. Frunzio, and R. J. Schoelkopf. 2016. An architecture for integrating planar and 3D cQED devices. *Applied Physics Letters* 109 (2016). Issue 4. https://doi.org/10.1063/1.4959241

[8] Michael E. Beverland, Prakash Murali, Matthias Troyer, Krysta M. Svore, Torsten Hoefler, Vadym Kliuchnikov, Guang Hao Low, Mathias Soeken, Aarthi Sundaram, and Alexander Vaschillo. 2022. Assessing requirements to scale to practical quantum advantage. (2022). https://doi.org/10.48550/ARXIV.2211.07629

[9] Alexandre Blais, Arne L. Grimsmo, S. M. Girvin, and Andreas Wallraff. 2021. Circuit quantum electrodynamics. *Reviews of Modern Physics* 93, 2, Article 025005 (April 2021), 025005 pages. https://doi.org/10.1103/RevModPhys.93.025005 arXiv:2005.12667 [quant-ph]

[10] Dolev Bluvstein et al. 2022. A quantum processor based on coherent transport of entangled atom arrays. *Nature* 604, 7906 (2022), 451–456. https://doi.org/10.1038/s41586-022-04592-6 arXiv:2112.03923 [quant-ph]

[11] Sergey Bravyi and Andrew Cross. 2015. Doubled color codes. *arXiv preprint arXiv:1509.03239* (2015).

[12] Sergey Bravyi, Oliver Dial, Jay M. Gambetta, Darió Gil, and Zaira Nazario. 2022. The future of quantum computing with superconducting qubits. *Journal of Applied Physics* 132 (2022). Issue 16. https://doi.org/10.1063/5.0082975

[13] Sergey Bravyi, Oliver Dial, Jay M. Gambetta, Dario Gil, and Zaira Nazario. 2022. The Future of Quantum Computing with Superconducting Qubits. *arXiv e-prints*, Article arXiv:2209.06841 (Sept. 2022), arXiv:2209.06841 pages. arXiv:2209.06841 [quant-ph]

[14] Sergey Bravyi and Jeongwan Haah. 2012. Magic-state distillation with low overhead. *Physical Review A* 86, 5 (2012), 052329.

[15] Teres A. Brecht, Wolfgang Pfaff, Chen Wang, Yiwen Chu, Luigi Frunzio, Michel H. Devoret, and Robert J. Schoelkopf. 2016. Multilayer microwave integrated quantum circuits for scalable quantum computing. *npj Quantum Information* 2 (2016). Issue 1. https://doi.org/10.1038/npjqi.2016.2

[16] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. 2019. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews* 6, 2, Article 021314 (June 2019), 021314 pages. https://doi.org/10.1063/1.5088164 arXiv:1904.04178 [quant-ph]

[17] Luke D. Burkhart, James D. Teoh, Yaxing Zhang, Christopher J. Axline, Luigi Frunzio, M. H. Devoret, Liang Jiang, S. M. Girvin, and R. J. Schoelkopf. 2021. Error-Detected State Transfer and Entanglement in a Superconducting Quantum Network. *PRX Quantum* 2 (2021). Issue 3. https://doi.org/10.1103/PRXQuantum.2.030321

[18] Tim Byrnes and Yoshihisa Yamamoto. 2006. Simulating lattice gauge theories on a quantum computer. *Physical Review A* 73, 2 (2006), 022328.

[19] A. R. Calderbank and Peter W. Shor. 1996. Good quantum error-correcting codes exist. *Phys. Rev. A* 54 (Aug 1996), 1098–1105. Issue 2. https://doi.org/10.1103/PhysRevA.54.1098

[20] Srivatsan Chakram, Andrew E. Oriani, Ravi K. Naik, Akash V. Dixit, Kevin He, Ankur Agrawal, Hyeokshin Kwon, and David I. Schuster. 2021. Seamless High- Q Microwave Cavities for Multimode Circuit Quantum Electrodynamics. *Physical Review Letters* 127 (2021). Issue 10. https://doi.org/10.1103/PhysRevLett.127.107701

[21] Christopher Chamberland and Michael E. Beverland. 2018. Flag fault-tolerant error correction with arbitrary distance codes. *Quantum* 2 (Feb. 2018), 53. https://doi.org/10.22331/q-2018-02-08-53 arXiv:1708.02246 [quant-ph]

[22] Christopher Chamberland and Tomas Jochym-O'Connor. 2017. Error suppression via complementary gauge choices in Reed-Muller codes. *Quantum Science and Technology* 2, 3 (Sept. 2017), 035008. https://doi.org/10.1088/2058-9565/aa7c4a arXiv:1705.00010 [quant-ph]

[23] Christopher Chamberland, Kyungjoo Noh, Patricio Arrangoiz-Arriola, Earl T. Campbell, Connor T. Hann, Joseph Iverson, Harald Putterman, Thomas C. Bohdanowicz, Steven T. Flammia, Andrew Keller, Gil Refael, John Preskill, Liang Jiang, Amir H. Safavi-Naeini, Oskar Painter, and Fernando G. S. L. Brandão. 2020. Building a fault-tolerant quantum computer using concatenated cat codes. *arXiv e-prints*, Article arXiv:2012.04108 (Dec. 2020), arXiv:2012.04108 pages. https://doi.org/10.48550/arXiv.2012.04108 arXiv:2012.04108 [quant-ph]

[24] Christopher Chamberland, Kyungjoo Noh, Patricio Arrangoiz-Arriola, Earl T. Campbell, Connor T. Hann, Joseph Iverson, Harald Putterman, Thomas C. Bohdanowicz, Steven T. Flammia, Andrew Keller, Gil Refael, John Preskill, Liang Jiang, Amir H. Safavi-Naeini, Oskar Painter, and Fernando G.S.L. Brandão. 2022.

Building a Fault-Tolerant Quantum Computer Using Concatenated Cat Codes. *PRX Quantum* 3 (2022). Issue 1. https://doi.org/10.1103/PRXQuantum.3.010329

[25] Christopher Chamberland, Guanyu Zhu, Theodore J. Yoder, Jared B. Hertzberg, and Andrew W. Cross. 2020. Topological and Subsystem Codes on Low-Degree Graphs with Flag Qubits. *Phys. Rev. X* 10 (Jan 2020), 011022. Issue 1. https://doi.org/10.1103/PhysRevX.10.011022

[26] Byung-Soo Choi. 2013. Cost comparison between code teleportation and stabilizer sequence methods for quantum code conversion. In *2013 International Conference on ICT Convergence (ICTC)*. 1083–1087. https://doi.org/10.1109/ICTC.2013.6675561

[27] Jahan Claes, J. Eli Bourassa, and Shruti Puri. 2023. Tailored cluster states with high threshold under biased noise. *npj Quantum Information* 9, 1 (2023), 9. https://doi.org/10.1038/s41534-023-00677-w

[28] Nikos Daniilidis and Hartmut Häffner. 2013. Quantum Interfaces Between Atomic and Solid-State Systems. *Annual Review of Condensed Matter Physics* 4, 1 (2013), 83–112. https://doi.org/10.1146/annurev-conmatphys-030212-184253 arXiv:https://doi.org/10.1146/annurev-conmatphys-030212-184253

[29] Zohreh Davoudi, Norbert M. Linke, and Guido Pagano. 2021. Toward simulating quantum field theories with controlled phonon-ion dynamics: A hybrid analog-digital approach. *Phys. Rev. Res.* 3 (Oct 2021), 043072. Issue 4. https://doi.org/10.1103/PhysRevResearch.3.043072

[30] D. De Motte, A. R. Grounds, M. Rehák, A. Rodriguez Blanco, B. Lekitsch, G. S. Giri, P. Neilinger, G. Oelsner, E. Il'ichev, M. Grajcar, and W. K. Hensinger. 2016. Experimental system design for the integration of trapped-ion and superconducting qubit systems. *Quantum Information Processing* 15, 12 (Dec. 2016), 5385–5414. https://doi.org/10.1007/s11128-016-1368-y arXiv:1510.07298 [quant-ph]

[31] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. 2002. Topological quantum memory. *J. Math. Phys.* 43, 9 (08 2002), 4452–4505. https://doi.org/10.1063/1.1499754 arXiv:https://pubs.aip.org/aip/jmp/article-pdf/43/9/4452/8171926/4452_1_online.pdf

[32] David Deutsch, Artur Ekert, Richard Jozsa, Chiara Macchiavello, Sandu Popescu, and Anna Sanpera. 1996. Quantum Privacy Amplification and the Security of Quantum Cryptography over Noisy Channels. *Physical Review Letters* 77, 13 (sep 1996), 2818–2821. https://doi.org/10.1103/physrevlett.77.2818

[33] Leon Ding, Max Hays, Youngkyu Sung, Bharath Kannan, Junyoung An, Agustin Di Paolo, Amir H. Karamlou, Thomas M. Hazard, Kate Azar, David K. Kim, Bethany M. Niedzielski, Alexander Melville, Mollie E. Schwartz, Jonilyn L. Yoder, Terry P. Orlando, Simon Gustavsson, Jeffrey A. Grover, Kyle Serniak, and William D. Oliver. 2023. High-Fidelity, Frequency-Flexible Two-Qubit Fluxonium Gates with a Transmon Coupler. arXiv:2304.06087 [quant-ph]

[34] Evan E. Dobbs, Joseph S. Friedman, and Alexandru Paler. 2022. Efficient Quantum Circuit Design with a Standard Cell Approach. *arXiv e-prints*, Article arXiv:2206.04990 (June 2022), arXiv:2206.04990 pages. https://doi.org/10.48550/arXiv.2206.04990 arXiv:2206.04990 [quant-ph]

[35] Ebru Dogan, Dario Rosenstock, Loïck Le Guevel, Haonan Xiong, Raymond A. Mencia, Aaron Somoroff, Konstantin N. Nesterov, Maxim G. Vavilov, Vladimir E. Manucharyan, and Chen Wang. 2022. Demonstration of the Two-Fluxonium Cross-Resonance Gate. arXiv:2204.11829 [quant-ph]

[36] Casey Duckering, Jonathan M. Baker, David I. Schuster, and Frederic T. Chong. 2020. Virtualized logical qubits: A 2.5d architecture for error-corrected quantum computing. *Proceedings of the Annual International Symposium on Microarchitecture, MICRO* 2020-October. https://doi.org/10.1109/MICRO50266.2020.00026

[37] Bryan Eastin and Emanuel Knill. 2009. Restrictions on transversal encoded quantum gate sets. *Physical review letters* 102, 11 (2009), 110502.

[38] Richard P. Feynman. 1982. Simulating physics with computers. *International Journal of Theoretical Physics* 21 (1982). Issue 6-7. https://doi.org/10.1007/BF02650179

[39] 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.

[40] X. Fu, L. Riesebos, L. Lao, C. G. Almudever, F. Sebastiano, R. Versluis, E. Charbon, and K. Bertels. 2016. A Heterogeneous Quantum Computer Architecture. In *Proceedings of the ACM International Conference on Computing Frontiers* (Como, Italy) *(CF '16)*. Association for Computing Machinery, New York, NY, USA, 323–330. https://doi.org/10.1145/2903150.2906827

[41] Suhas S Ganjam, Yanhao Wang, Yao Lu, Archan Banerjee, Chan U Lei, Lev Krayzman, Kim Kisslinger, Chenyu Zhou, Yichen Jia, Mingzhao Liu, Luigi Frunzio, and Robert J Schoelkopf. 2023. Characterizing microwave losses in superconducting quantum circuits: Part 2. *Bulletin of the American Physical Society* (2023).

[42] Yvonne Y. Gao, M. Adriaan Rol, Steven Touzard, and Chen Wang. 2021. Practical Guide for Building Superconducting Quantum Devices. *PRX Quantum* 2 (2021). Issue 4. https://doi.org/10.1103/PRXQuantum.2.040202

[43] Craig Gidney and Martin Ekerå. 2019. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *arXiv e-prints*, Article arXiv:1905.09749 (May 2019), arXiv:1905.09749 pages. https://doi.org/10.48550/arXiv.1905.09749 arXiv:1905.09749 [quant-ph]

[44] Daniel Gottesman. 2022. Opportunities and Challenges in Fault-Tolerant Quantum Computation. arXiv:2210.15844 [quant-ph]

[45] Élie Gouzien and Nicolas Sangouard. 2021. Factoring 2048-bit RSA Integers in 177 Days with 13 436 Qubits and a Multimode Memory. *Phys. Rev. Lett.* 127, 14, Article 140503 (Oct. 2021), 140503 pages. https://doi.org/10.1103/PhysRevLett.127.140503 arXiv:2103.06159 [quant-ph]

[46] Peter Groszkowski and Jens Koch. 2021. Scqubits: a Python package for superconducting qubits. *Quantum* 5 (Nov. 2021), 583. https://doi.org/10.22331/q-2021-11-17-583 arXiv:2107.08552 [quant-ph]

[47] D.V. Hall. 1992. *Microprocessors and Interfacing: Programming and Hardware*. Glencoe. https://books.google.com/books?id=FslhmAEACAAJ

[48] Connor T Hann, Christopher Chamberland, Harald Putterman, Joseph Iverson, Arne Grimsmo, Oskar Painter, Fernando Brandao, and Kyungjoo Noh. 2023. Hybrid cat-transmon architecture for scalable, hardware-efficient quantum error correction. *Bulletin of the American Physical Society* (2023).

[49] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature* 567, 7747 (2019), 209–212.

[50] Hsin-Yuan Huang, Richard Kueng, and John Preskill. 2020. Predicting many properties of a quantum system from very few measurements. *Nature Physics* 16, 10 (2020), 1050–1057.

[51] IBM. [n. d.]. IBM Quantum Roadmap. https://research.ibm.com/blog/ibm-quantum-roadmap-2025 Accessed: Sep. 20, 2022.

[52] S.M. Kang, Y. Leblebici, and C.W. Kim. 2014. *CMOS Digital Integrated Circuits Analysis & Design*. McGraw-Hill Education. https://books.google.com/books?id=8wSTygAACAAJ

[53] Ivan Kassal, James D Whitfield, Alejandro Perdomo-Ortiz, Man-Hong Yung, and Alán Aspuru-Guzik. 2011. Simulating chemistry using quantum computers. *Annual review of physical chemistry* 62 (2011), 185–207.

[54] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger. 2020. Shuttling-based trapped-ion quantum information processing. *AVS Quantum Science* 2, 1, Article 014101 (Feb. 2020), 014101 pages. https://doi.org/10.1116/1.5126186 arXiv:1912.04712 [quant-ph]

[55] D. Kielpinski, D. Kafri, M. J. Woolley, G. J. Milburn, and J. M. Taylor. 2012. Quantum Interface between an Electrical Circuit and a Single Atom. *prl* 108, 13, Article 130504 (March 2012), 130504 pages. https://doi.org/10.1103/PhysRevLett.108.130504 arXiv:1111.5999 [quant-ph]

[56] Emanuel Knill and Raymond Laflamme. 1996. Concatenated Quantum Codes. *arXiv e-prints*, Article quant-ph/9608012 (Aug. 1996), quant-ph/9608012 pages. https://doi.org/10.48550/arXiv.quant-ph/9608012 arXiv:quant-ph/9608012 [quant-ph]

[57] Sandoko Kosen, Hang Xi Li, Marcus Rommel, Daryoush Shiri, Christopher Warren, Leif Grönberg, Jaakko Salonen, Tahereh Abad, Janka Biznárová, Marco Caputo, Liangyu Chen, Kestutis Grigoras, Göran Johansson, Anton Frisk Kockum, Christian KriÅ3/4an, Daniel Pérez Lozano, Graham J. Norris, Amr Osman, Jorge Fernández-Pendás, Alberto Ronzani, Anita Fadavi Roudsari, Slawomir Simbierowicz, Giovanna Tancredi, Andreas Wallraff, Christopher Eichler, Joonas Govenius, and Jonas Bylander. 2022. Building blocks of a flip-chip integrated superconducting quantum processor. *Quantum Science and Technology* 7 (2022). Issue 3. https://doi.org/10.1088/2058-9565/ac734b

[58] Stefan Krastanov, Hamza Raniwala, Jeffrey Holzgrafe, Kurt Jacobs, Marko Lončar, Matthew J. Reagor, and Dirk R. Englund. 2021. Optically Heralded Entanglement of Superconducting Systems in Quantum Networks. *Phys. Rev. Lett.* 127 (Jul 2021), 040503. Issue 4. https://doi.org/10.1103/PhysRevLett.127.040503

[59] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, Graham J Norris, Christian Kraglund Andersen, Markus Müller, Alexandre Blais, Christopher Eichler, and Andreas Wallraff. 2022. Realizing repeated quantum error correction in a distance-three surface code. *Nature* 605, 7911 (2022), 669–674. https://doi.org/10.1038/s41586-022-04566-8

[60] S. Krinner, S. Storz, P. Kurpiers, P. Magnard, J. Heinsoo, R. Keller, J. Lütolf, C. Eichler, and A. Wallraff. 2019. Engineering cryogenic setups for 100-qubit scale superconducting circuit systems. *EPJ Quantum Technology* 6 (2019). Issue 1. https://doi.org/10.1140/epjqt/s40507-019-0072-0

[61] Thi Ha Kyaw, Tim Menke, Sukin Sim, Abhinav Anand, Nicolas P. D. Sawaya, William D. Oliver, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. 2021. Quantum Computer-Aided Design: Digital Quantum Simulation of Quantum Processors. *Phys. Rev. Applied* 16, 4 (2021), 044042. https://doi.org/10.1103/PhysRevApplied.16.044042 arXiv:2006.03070 [quant-ph]

[62] Nicholas LaRacuente, Kaitlin N. Smith, Poolad Imany, Kevin L. Silverman, and Frederic T. Chong. 2022. Short-Range Microwave Networks to Scale Superconducting Quantum Computation. (1 2022). arXiv:2201.08825 [quant-ph]

[63] Chan U Lei, Lev Krayzman, Suhas Ganjam, Luigi Frunzio, and Robert J. Schoelkopf. 2020. High coherence superconducting microwave cavities with indium bump bonding. *Applied Physics Letters* 116 (4 2020), 154002. Issue 15. https://doi.org/10.1063/5.0003907

[64] N. Leung, Y. Lu, S. Chakram, R. K. Naik, N. Earnest, R. Ma, K. Jacobs, A. N. Cleland, and D. I. Schuster. 2019. Deterministic bidirectional communication and

remote entanglement generation between superconducting qubits. *npj Quantum Information* 5 (2019). Issue 1. https://doi.org/10.1038/s41534-019-0128-0

[65] Ang Li, Omer Subasi, Xiu Yang, and Sriram Krishnamoorthy. 2020. Density Matrix Quantum Circuit Simulation via the BSP Machine on Modern GPU Clusters. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–15. https://doi.org/10.1109/SC41405.2020.00017

[66] Sophia Fuhui Lin, Sara Sussman, Casey Duckering, Pranav S. Mundada, Jonathan M. Baker, Rohan S. Kumar, Andrew A. Houck, and Frederic T. Chong. 2022. Let Each Quantum Bit Choose Its Basis Gates. *arXiv:2208.13380; In MICRO 2022: 55th IEEE/ACM International Symposium on Microarchitecture, 17 pages, 7 figures* (2022).

[67] Daniel Litinski. 2019. Magic State Distillation: Not as Costly as You Think. *Quantum* 3 (Dec. 2019), 205. https://doi.org/10.22331/q-2019-12-02-205 arXiv:1905.06903 [quant-ph]

[68] Daniel Litinski. 2019. Magic state distillation: Not as costly as you think. *Quantum* 3 (2019), 205.

[69] Gregory S. MacCabe, Hengjiang Ren, Jie Luo, Justin D. Cohen, Hengyun Zhou, Alp Sipahigil, Mohammad Mirhosseini, and Oskar Painter. 2020. Nano-acoustic resonator with ultralong phonon lifetime. *Science* 370 (2020). Issue 6518. https://doi.org/10.1126/science.abc7312

[70] H.J. Mamin, E. Huang, S. Carnevale, C.T. Rettner, N. Arellano, M.H. Sherwood, C. Kurter, B. Trimm, M. Sandberg, R.M. Shelby, M.A. Mueed, B.A. Madon, A. Pushp, M. Steffen, and D. Rugar. 2021. Merged-Element Transmons: Design and Qubit Performance. *Phys. Rev. Appl.* 16 (Aug 2021), 024023. Issue 2. https://doi.org/10.1103/PhysRevApplied.16.024023

[71] Harry J Mamin. 2022. The Merged Element Transmon. *Bulletin of the American Physical Society* (2022).

[72] Matteo Mariantoni, H. Wang, T. Yamamoto, M. Neeley, Radoslaw C. Bialczak, Y. Chen, M. Lenander, Erik Lucero, A. D. O'Connell, D. Sank, M. Weides, J. Wenner, Y. Yin, J. Zhao, A. N. Korotkov, A. N. Cleland, and John M. Martinis. 2011. Implementing the quantum von Neumann architecture with superconducting circuits. *Science* 334 (2011). Issue 6052. https://doi.org/10.1126/science.1208517

[73] John M. Martinis. 2015. Qubit metrology for building a fault-tolerant quantum computer. *npj Quantum Information* 1 (2015). https://doi.org/10.1038/npjqi.2015.5

[74] Aleksei R. Matanin, Konstantin I. Gerasimov, Eugene S. Moiseev, Nikita S. Smirnov, Anton I. Ivanov, Elizaveta I. Malevannaya, Victor I. Polozov, Eugeny V. Zikiy, Andrey A. Samoilov, Ilya A. Rodionov, and Sergey A. Moiseev. 2023. Toward Highly Efficient Multimode Superconducting Quantum Memory. *Phys. Rev. Appl.* 19 (Mar 2023), 034011. Issue 3. https://doi.org/10.1103/PhysRevApplied.19.034011

[75] Evan McKinney, Mingkang Xia, Chao Zhou, Pinlei Lu, Michael Hatridge, and Alex K. Jones. 2022. Co-Designed Architectures for Modular Superconducting Quantum Computers. *arXiv e-prints*, Article arXiv:2205.04387 (May 2022), arXiv:2205.04387 pages. https://doi.org/10.48550/arXiv.2205.04387 arXiv:2205.04387 [quant-ph]

[76] Evan McKinney, Chao Zhou, Mingkang Xia, Michael Hatridge, and Alex K. Jones. 2023. Parallel Driving for Fast Quantum Computing Under Speed Limits. *arXiv e-prints*, Article arXiv:2302.01252 (Feb. 2023), arXiv:2302.01252 pages. https://doi.org/10.48550/arXiv.2302.01252 arXiv:2302.01252 [quant-ph]

[77] C. Mead and L. Conway. 1980. *Introduction to VLSI systems*.

[78] Tim Menke, Florian Häse, Simon Gustavsson, Andrew J. Kerman, William D. Oliver, and Alán Aspuru-Guzik. 2021. Automated design of superconducting circuits and its application to 4-local couplers. *npj Quantum Information* 7, 1 (2021), 49. https://doi.org/10.1038/s41534-021-00382-6

[79] T.S. Metodi and F.T. Chong. 2006. *Quantum Computing for Computer Architects*. Morgan & Claypool Publishers. https://books.google.com/books?id=8sBQAAAAMAAJ

[80] T.S. Metodi, D.D. Thaker, A.W. Cross, F.T. Chong, and I.L. Chuang. 2005. A quantum logic array microarchitecture: scalable quantum data movement and computation. In *38th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'05)*. 12 pp.–318. https://doi.org/10.1109/MICRO.2005.9

[81] Tzvetan S. Metodi, Darshan D. Thaker, Andrew W. Cross, Isaac L. Chuang, and Frederic T. Chong. 2008. High-Level Interconnect Model for the Quantum Logic Array Architecture. *J. Emerg. Technol. Comput. Syst.* 4, 1, Article 1 (apr 2008), 28 pages. https://doi.org/10.1145/1330521.1330522

[82] Ofir Milul, Barkay Guttel, Uri Goldblatt, Sergey Hazanov, Lalit M. Joshi, Daniel Chausovsky, Nitzan Kahn, Engin Çiftyürek, Fabien Lafont, and Serge Rosenblum. 2023. A superconducting quantum memory with tens of milliseconds coherence time. *arXiv e-prints*, Article arXiv:2302.06442 (Feb. 2023), arXiv:2302.06442 pages. https://doi.org/10.48550/arXiv.2302.06442 arXiv:2302.06442 [quant-ph]

[83] Ofir Milul, Barkay Guttel, Uri Goldblatt, Sergey Hazanov, Lalit M. Joshi, Daniel Chausovsky, Nitzan Kahn, Engin Çiftyürek, Fabien Lafont, and Serge Rosenblum. 2023. A superconducting quantum memory with tens of milliseconds coherence time. https://doi.org/10.48550/ARXIV.2302.06442

[84] Zlatko K Minev, Thomas G McConkey, Jeremy Drysdale, Priti Shah, Dennis Wang, Marco Facchini, Grace Harper, John Blair, Helena Zhang, Nick Lanzillo, Sagarika Mukesh, Will Shanks, Chris Warren, and Jay M Gambetta. 2021. Qiskit Metal: An Open-Source Framework for Quantum Device Design & Analysis. https://doi.org/10.5281/zenodo.4618153

[85] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L. M. Duan, and J. Kim. 2012. Large Scale Modular Quantum Computer Architecture with Atomic Memory and Photonic Interconnects. *arXiv e-prints*, Article arXiv:1208.0391 (Aug. 2012), arXiv:1208.0391 pages. arXiv:1208.0391 [quant-ph]

[86] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L. M. Duan, and J. Kim. 2014. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *pra* 89, 2, Article 022317 (Feb. 2014), 022317 pages. https://doi.org/10.1103/PhysRevA.89.022317

[87] Yehuda Naveh, Elham Kashefi, James R. Wootton, and Koen Bertels. 2018. Theoretical and practical aspects of verification of quantum computers. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 721–730. https://doi.org/10.23919/DATE.2018.8342103

[88] Tomáš Neuman, Matt Eichenfield, Matthew E. Trusheim, Lisa Hackett, Prineha Narang, and Dirk Englund. 2021. A phononic interface between a superconducting quantum processor and quantum networked spin memories. *npj Quantum Information* 7, Article 121 (Jan. 2021), 121 pages. https://doi.org/10.1038/s41534-021-00457-4

[89] Long B. Nguyen, Yen-Hsiang Lin, Aaron Somoroff, Raymond Mencia, Nicholas Grabon, and Vladimir E. Manucharyan. 2019. High-Coherence Fluxonium Qubit. *Phys. Rev. X* 9 (Nov 2019), 041041. Issue 4. https://doi.org/10.1103/PhysRevX.9.041041

[90] M.A. Nielsen and I.L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. https://books.google.com/books?id=-s4DEy7o-a0C

[91] Kyungjoo Noh, Christopher Chamberland, and Fernando G.S.L. Brandão. 2022. Low-Overhead Fault-Tolerant Quantum Error Correction with the Surface-GKP Code. *PRX Quantum* 3 (Jan 2022), 010315. Issue 1. https://doi.org/10.1103/PRXQuantum.3.010315

[92] Sai Pavan Chitta, Tianpu Zhao, Ziwen Huang, Ian Mondragon-Shem, and Jens Koch. 2022. Computer-aided quantization and numerical analysis of superconducting circuits. *arXiv e-prints*, Article arXiv:2206.08320 (June 2022), arXiv:2206.08320 pages. https://doi.org/10.48550/arXiv.2206.08320 arXiv:2206.08320 [quant-ph]

[93] Marek Pechal, Patricio Arrangoiz-Arriola, and Amir H. Safavi-Naeini. 2019. Superconducting circuit quantum computing with nanomechanical resonators as storage. *Quantum Science and Technology* 4 (2019). Issue 1. https://doi.org/10.1088/2058-9565/aadc6c

[94] M. Pechal, G. Salis, M. Ganzhorn, D. J. Egger, M. Werninghaus, and S. Filipp. 2021. Characterization and Tomography of a Hidden Qubit. *Phys. Rev. X* 11 (Nov 2021), 041032. Issue 4. https://doi.org/10.1103/PhysRevX.11.041032

[95] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis. 2021. Demonstration of the trapped-ion quantum CCD computer architecture. *nat* 592, 7853 (Jan. 2021), 209–213. https://doi.org/10.1038/s41586-021-03318-4 arXiv:2003.01293 [quant-ph]

[96] Alexander PM Place, Lila VH Rodgers, Pranav Mundada, Basil M Smitham, Mattias Fitzpatrick, Zhaoqi Leng, Anjali Premkumar, Jacob Bryon, Andrei Vrajitoarea, Sara Sussman, et al. 2021. New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds. *Nature communications* 12, 1 (2021), 1–6.

[97] Alexander Patrick McCormick Place. 2022. *Increasing Lifetimes of Superconducting Qubits*. Ph. D. Dissertation. Princeton University.

[98] John Preskill. 2021. Quantum computing 40 years later. (2021). https://doi.org/10.48550/ARXIV.2106.10522

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

[100] Shesha Raghunathan and Leon Stok. 2020. EDA and Quantum Computing: a symbiotic relationship? *IEEE Design & Test* 37, 6 (2020), 71–78. https://doi.org/10.1109/MDAT.2020.3016586

[101] Alexander P. Read, Benjamin J. Chapman, Chan U Lei, Jacob C. Curtis, Suhas Ganjam, Lev Krayzman, Luigi Frunzio, and Robert J. Schoelkopf. 2022. Precision measurement of the microwave dielectric loss of sapphire in the quantum regime with parts-per-billion sensitivity. (2022). https://doi.org/10.48550/ARXIV.2206.14334

[102] Matthew Reagor, Wolfgang Pfaff, Christopher Axline, Reinier W. Heeres, Nissim Ofek, Katrina Sliwa, Eric Holland, Chen Wang, Jacob Blumoff, Kevin Chou, Michael J. Hatridge, Luigi Frunzio, Michel H. Devoret, Liang Jiang, and Robert J. Schoelkopf. 2016. Quantum memory with millisecond coherence in circuit QED. *Phys. Rev. B* 94 (Jul 2016), 014506. Issue 1. https://doi.org/10.1103/PhysRevB.94.014506

[103] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. 2017. Elucidating reaction mechanisms on quantum computers. *Proceedings of the national academy of sciences* 114, 29 (2017), 7555–7560.

[104] C. Schuck, X. Guo, L. Fan, X. Ma, M. Poot, and H. X. Tang. 2016. Quantum interference in heterogeneous superconducting-photonic circuits on a silicon

chip. *Nature Communications* 7, Article 10352 (Jan. 2016), 10352 pages. https://doi.org/10.1038/ncomms10352 arXiv:1511.07081 [quant-ph]

[105] Eyob A. Sete, John M. Martinis, and Alexander N. Korotkov. 2015. Quantum theory of a bandpass Purcell filter for qubit readout. *Phys. Rev. A* 92 (Jul 2015), 012325. Issue 1. https://doi.org/10.1103/PhysRevA.92.012325

[106] R. Shankar, E.B. Fernandez, and N.G. Einspruch. 2014. *VLSI and Computer Architecture*. Elsevier Science. https://books.google.com/books?id=jDGoBQAAQBAJ

[107] Yunong Shi, Pranav Gokhale, Prakash Murali, Jonathan M. Baker, Casey Duckering, Yongshan Ding, Natalie C. Brown, Christopher Chamberland, Ali Javadi Abhari, Andrew W. Cross, David I. Schuster, Kenneth R. Brown, Margaret Martonosi, and Frederic T. Chong. 2020. Resource-Efficient Quantum Computing by Breaking Abstractions. *arXiv e-prints*, Article arXiv:2011.00028 (Oct. 2020), arXiv:2011.00028 pages. https://doi.org/10.48550/arXiv.2011.00028 arXiv:2011.00028 [quant-ph]

[108] P.W. Shor. 1996. Fault-tolerant quantum computation. In *Proceedings of 37th Conference on Foundations of Computer Science*. 56–65. https://doi.org/10.1109/SFCS.1996.548464

[109] Peter W. Shor. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* 26 (1997). Issue 5. https://doi.org/10.1137/S0097539795293172

[110] V. V. Sivak, A. Eickbusch, B. Royer, S. Singh, I. Tsioutsios, S. Ganjam, A. Miano, B. L. Brock, A. Z. Ding, L. Frunzio, S. M. Girvin, R. J. Schoelkopf, and M. H. Devoret. 2023. Real-time quantum error correction beyond break-even. *Nat.* 616, 7955 (April 2023), 50–55. https://doi.org/10.1038/s41586-023-05782-6

[111] Kaitlin N. Smith, Gokul Subramanian Ravi, Jonathan M. Baker, and Frederic T. Chong. 2022. Scaling Superconducting Quantum Computers with Chiplet Architectures. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 1092–1109. https://doi.org/10.1109/MICRO56248.2022.00078

[112] Leon Stok. 2021. EDA and Quantum Computing: The Key Role of Quantum Circuits. In *Proceedings of the 2021 International Symposium on Physical Design* (Virtual Event, USA) *(ISPD '21)*. Association for Computing Machinery, New York, NY, USA, 111. https://doi.org/10.1145/3439706.3446900

[113] Thomas Szkopek, P Oscar Boykin, Heng Fan, Vwani P Roychowdhury, Eli Yablonovitch, Geoffrey Simms, Mark Gyure, and Bryan Fong. 2006. Threshold error penalty for fault-tolerant quantum computation with nearest neighbor communication. *IEEE transactions on nanotechnology* 5, 1 (2006), 42–49.

[114] Andrew K. Tan, Yuan Liu, Minh C. Tran, and Isaac L. Chuang. 2023. Error Correction of Quantum Algorithms: Arbitrarily Accurate Recovery Of Noisy Quantum Signal Processing. *arXiv e-prints*, Article arXiv:2301.08542 (Jan. 2023), arXiv:2301.08542 pages. https://doi.org/10.48550/arXiv.2301.08542 arXiv:2301.08542 [quant-ph]

[115] A.S. Tanenbaum and J.R. Goodman. 1999. *Structured Computer Organization*. Prentice Hall. https://books.google.com/books?id=6oVQAAAAMAAJ

[116] James D. Teoh, Patrick Winkel, Harshvardhan K. Babla, Benjamin J. Chapman, Jahan Claes, Stijn J. de Graaf, John W. O. Garmon, William D. Kalfus, Yao Lu, Aniket Maiti, Kaavya Sahay, Neel Thakur, Takahiro Tsunoda, Sophia H. Xue, Luigi Frunzio, Steven M. Girvin, Shruti Puri, and Robert J. Schoelkopf. 2022. Dual-rail encoding with superconducting cavities. *arXiv e-prints*, Article arXiv:2212.12077 (Dec. 2022), arXiv:2212.12077 pages. https://doi.org/10.48550/arXiv.2212.12077 arXiv:2212.12077 [quant-ph]

[117] James D. Teoh, Patrick Winkel, Harshvardhan K. Babla, Benjamin J. Chapman, Jahan Claes, Stijn J. de Graaf, John W. O. Garmon, William D. Kalfus, Yao Lu, Aniket Maiti, Kaavya Sahay, Neel Thakur, Takahiro Tsunoda, Sophia H. Xue, Luigi Frunzio, Steven M. Girvin, Shruti Puri, and Robert J. Schoelkopf. 2022. Dual-rail encoding with superconducting cavities. (2022). https://doi.org/10.48550/ARXIV.2212.12077

[118] Darshan D Thaker, Tzvetan S Metodi, Andrew W Cross, Isaac L Chuang, and Frederic T Chong. 2006. Quantum memory hierarchies: Efficient designs to match available parallelism in quantum computing. *ACM SIGARCH Computer Architecture News* 34, 2 (2006), 378–390.

[119] Darshan D. Thaker, Tzvetan S. Metodi, Andrew W. Cross, Isaac L. Chuang, and Frederic T. Chong. 2006. Quantum memory hierarchies: Efficient designs to match available parallelism in quantum computing. *Proceedings - International Symposium on Computer Architecture* 2006. https://doi.org/10.1109/ISCA.2006.32

[120] S. Trimberger and R. F. W. Pease. 1983. Reaching for the million-transistor chip. *IEEE Spectrum* 20, 11 (1983), 100–105. https://doi.org/10.1109/MSPEC.1983.6370030

[121] J. Verdu, H. Zoubi, Ch. Koller, J. Majer, H. Ritsch, and J. Schmiedmayer. 2009. Strong Magnetic Coupling of an Ultracold Gas to a Superconducting Waveguide Cavity. *Phys. Rev. Lett.* 103 (Jul 2009), 043603. Issue 4. https://doi.org/10.1103/PhysRevLett.103.043603

[122] K.N. Vijeyakumar, V. Sumathy, M.Gayathri Devi, S. Tamilselvan, and Remya.R. Nair. 2012. Design Of Hardware Efficient High Speed Multiplier Using Modified Ternary Logic. *Procedia Engineering* 38 (2012), 2186–2195. https://doi.org/10.1016/j.proeng.2012.06.263 INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING.

[123] Christopher Wang. 2022. Bosonic Quantum Simulation in Circuit Quantum Electrodynamics.

[124] Chenlu Wang, Xuegang Li, Huikai Xu, Zhiyuan Li, Junhua Wang, Zhen Yang, Zhenyu Mi, Xuehui Liang, Tang Su, Chuhong Yang, Guangyue Wang, Wenyan Wang, Yongchao Li, Mo Chen, Chengyao Li, Kehuan Linghu, Jiaxiu Han, Yingshan Zhang, Yulong Feng, Yu Song, Teng Ma, Jingning Zhang, Ruixia Wang, Peng Zhao, Weiyang Liu, Guangming Xue, Yirong Jin, and Haifeng Yu. 2022. Towards practical quantum computers: transmon qubit with a lifetime approaching 0.5 milliseconds. *npj Quantum Information* 8 (2022), 3. Issue 1. https://doi.org/10.1038/s41534-021-00510-2

[125] Christopher S. Wang, Jacob C. Curtis, Brian J. Lester, Yaxing Zhang, Yvonne Y. Gao, Jessica Freeze, Victor S. Batista, Patrick H. Vaccaro, Isaac L. Chuang, Luigi Frunzio, Liang Jiang, S. M. Girvin, and Robert J. Schoelkopf. 2020. Efficient Multiphoton Sampling of Molecular Vibronic Spectra on a Superconducting Bosonic Processor. *Phys. Rev. X* 10 (Jun 2020), 021060. Issue 2. https://doi.org/10.1103/PhysRevX.10.021060

[126] Shiou-An Wang, Chin-Yung Lu, I-Ming Tsai, and Sy-Yen Kuo. 2008. An XQDD-Based Verification Method for Quantum Circuits. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* E91-A, 2 (feb 2008), 584–594. https://doi.org/10.1093/ietfec/e91-a.2.584

[127] K. X. Wei, E. Magesan, I. Lauer, S. Srinivasan, D. F. Bogorin, S. Carnevale, G. A. Keefe, Y. Kim, D. Klaus, W. Landers, N. Sundaresan, C. Wang, E. J. Zhang, M. Steffen, O. E. Dial, D. C. McKay, and A. Kandala. 2022. Hamiltonian Engineering with Multicolor Drives for Fast Entangling Gates and Quantum Crosstalk Cancellation. *Phys. Rev. Lett.* 129 (Aug 2022), 060501. Issue 6. https://doi.org/10.1103/PhysRevLett.129.060501

[128] Neil HE Weste and Kamran Eshraghian. 1985. *Principles of CMOS VLSI design: a systems perspective*. Addison-Wesley Longman Publishing Co., Inc.

[129] Robert Wille, Austin Fowler, and Yehuda Naveh. 2018. Computer-Aided Design for Quantum Computation. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–6. https://doi.org/10.1145/3240765.3267469

[130] Sau Lan Wu, Jay Chan, Wen Guan, Shaojun Sun, Alex Wang, Chen Zhou, Miron Livny, Federico Carminati, Alberto Di Meglio, Andy CY Li, et al. 2021. Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM quantum computer simulator and hardware with 10 qubits. *Journal of Physics G: Nuclear and Particle Physics* 48, 12 (2021), 125003.

[131] Ze-Liang Xiang, Sahel Ashhab, JQ You, and Franco Nori. 2013. Hybrid quantum circuits: Superconducting circuits interacting with other quantum systems. *Reviews of Modern Physics* 85, 2 (2013), 623.

[132] C. B. Young, A. Safari, P. Huft, J. Zhang, E. Oh, R. Chinnarasu, and M. Saffman. 2022. An architecture for quantum networking of neutral atom processors. *Appl. Phys. B* 128, 8 (2022), 151. https://doi.org/10.1007/s00340-022-07865-0 arXiv:2202.01634 [quant-ph]

[133] Mohamed Zahran. 2016. Heterogeneous Computing: Here to Stay: Hardware and Software Perspectives. *Queue* 14, 6 (dec 2016), 31–42. https://doi.org/10.1145/3028687.3038873

[134] Bei Zeng, Andrew Cross, and Isaac L Chuang. 2011. Transversality versus universality for additive quantum codes. *IEEE Transactions on Information Theory* 57, 9 (2011), 6272–6284.

[135] Kuan Zhang, Jayne Thompson, Xiang Zhang, Yangchao Shen, Yao Lu, Shuaining Zhang, Jiajun Ma, Vlatko Vedral, Mile Gu, and Kihwan Kim. 2019. Modular quantum computation in a trapped ion system. *Nature Communications* 10, Article 4692 (Oct. 2019), 4692 pages. https://doi.org/10.1038/s41467-019-12643-2 arXiv:1907.12171 [quant-ph]

[136] Chao Zhou, Pinlei Lu, Matthieu Praquin, Tzu-Chiao Chien, Ryan Kaufman, Xi Cao, Mingkang Xia, Roger Mong, Wolfgang Pfaff, David Pekker, and Michael Hatridge. 2021. A modular quantum computer based on a quantum state router. (2021). https://doi.org/10.48550/ARXIV.2109.06848